SIAM J. OPTIM.
Vol. 9, No. 1, pp. 1–13

AN INTERIOR PROXIMAL ALGORITHM AND THE EXPONENTIAL MULTIPLIER METHOD FOR SEMIDEFINITE PROGRAMMING*

MOSHE DOLJANSKY† AND MARC TEBOULLE†

Abstract. We introduce an interior proximal algorithm for semidefinite optimization problems and establish its convergence properties. We also study the corresponding dual algorithm leading to an exponential multiplier method for semidefinite programs. Potential applications and extensions are also discussed.

Key words. semidefinite optimization, proximal-like methods, exponential penalty, augmented Lagrangian/multiplier methods

AMS subject classifications. 90C25, 90C30

PII. S1052623496309405

1. Introduction.

Semidefinite programming (SDP) has recently attracted the attention of many researchers with a focus on the development of interior point methods similar to those used in linear programming. Semidefinite programs arise naturally in a wide range of applications in engineering, optimal control, statistics, and combinatorial optimization. We refer the reader to the recent survey articles of Alizadeh [1] and Vandenberghe and Boyd [21] and references therein.

Semidefinite programs are in fact a special class of convex programs with either real symmetric matrices as decision variables and cone constraints and/or decision variables in the n-dimensional Euclidean space R^n with convex nonsmooth linear matrix inequality constraints. Thus they can in principle be solved via general convex programming methods. One such method is the proximal point algorithm (see, e.g., [15], [18]) for minimizing a closed proper convex function over R^n. Recently, several generalizations of the proximal algorithm have been considered, where the usual quadratic proximal term is replaced by nonquadratic distance-like functions; see, for example, [2], [9], [12], [19] and references therein. One key feature of these nonquadratic proximal methods is that they can handle simple bounds (such as nonnegativity constraints) in such a way that they automatically generate iterates which stay in the interior of the feasible set and therefore eliminate the combinatorial nature of the problem. One of the main applications of these proximal methods is to the dual of smooth convex programs, leading to twice continuously differentiable nonquadratic augmented Lagrangians (in contrast with the usual once differentiable quadratic Lagrangian) and thus allowing the use of Newton's method; see, e.g., [6], [16]. Several recent implementations of these methods have been reported with good numerical results, particularly for large scale problems; see, e.g., [4], [5], [7].

It is thus natural to consider the possibility of developing similar proximal-type algorithms and their corresponding dual augmented Lagrangian methods (also called multiplier methods) for solving SDP. The motivation of this paper is to explore such

*Received by the editors September 16, 1996; accepted for publication (in revised form) October 8, 1997; published electronically October 30, 1998.

http://www.siam.org/journals/siopt/9-1/30940.html

†School of Mathematical Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel (teboulle@math.tau.ac.il). The second author was partially supported by grant 9636-1-96 of the Israeli Ministry of Science.

a possibility. We introduce an interior proximal-type algorithm for convex optimization problems over the cone of positive semidefinite matrices, outline the main tools needed to develop such a proximal method for SDP, state a basic algorithm and its corresponding dual multipliers method, and establish convergence properties of the proposed methods.

The next section gives some preliminary results on positive semidefinite matrices and corresponding matrix functions needed in the paper. Section 3 introduces a distance-like functional measuring closeness between two positive semidefinite matrices and summarizes some of its basic properties. The basic algorithm and its convergence analysis is studied in section 4. In section 5, we consider a dual application of the algorithm, leading to a multiplier method for solving SDP which is shown to possess properties similar to the exponential multiplier method used in convex programming; see, e.g., [6], [20]. Our last section briefly discusses further results and potential extensions within the proximal framework for solving SDP.

2. Convex minimization over the cone of positive semidefinite matrices. In this paper, we study convex optimization problems of the form

$$(P) \quad \inf\{f(X) : X \succeq 0\}.$$

Here $f : S_n \to (-\infty, +\infty]$ is a closed proper function on the space S_n of $n \times n$ real symmetric matrices. For $A \in S_n$, the notation $A \succeq 0$ $(A \succ 0)$ means that the matrix A is positive semidefinite (positive definite). We will also use the notation S_n^+, S_n^{++} to denote the space of positive semidefinite and positive definite matrices, respectively. The inner product on S_n is defined as $\langle A, B\rangle := \mathrm{tr}(AB) = \sum_{i,j} A_{i,j}B_{i,j}$, where tr stands for the trace operator.

Every matrix $X \in S_n$ can be written in the form

$$X = V^T \Lambda V,$$

where $\Lambda = \mathrm{diag}\ (\lambda_1(X), \ldots, \lambda_n(X))$, $\{\lambda_i(X)\}_{i=1}^m$ being the eigenvalues of X, and V is the corresponding orthonormal matrix of the eigenvectors of X. Hence, for any analytic scalar valued function g, we can define a function of a matrix $X \in S_n$ by the matrix

$$g(X) := V^T g(\Lambda) V \tag{2.1}$$

whenever the scalar functions $g(\lambda_i(X))$ are well defined; see, e.g., [11, Section 6.2].

The trace function plays an important role in this paper. From (2.1) we have

$$\mathrm{tr}\, g(X) \;=\; \mathrm{tr}\, V^T g(\Lambda) V \;=\; \mathrm{tr}\, V^T V g(\Lambda) \;=\; \mathrm{tr}\, g(\Lambda) \;=\; \sum_{i=1}^n g(\lambda_i(X)).$$

All the necessary material on continuity and differentiability of matrix functions and properties of traces of matrix functions used in this paper can be found in Chapter 6 of Horn and Johnson [11].

Standard convex analysis results on R^n have their counterparts on the space S_n. Following Rockafellar [17] we denote the domain of f by $dom\, f := \{X \in S_n : f(X) < \infty\}$. For a closed proper convex function f on S_n, the subdifferential of f at $M \in S_n$ is the convex set

$$\partial f(M) \;=\; \{B \in S_n \;:\; f(A) \;\geq\; f(M) \;+\; \mathrm{tr}\,(A - M)B \;\; \forall A \in S_n\}. \tag{2.2}$$

We say that $B \in S_n$ is normal to a convex set $C \subset S_n$ at M if

$$\mathrm{tr}\,(AB) \le \mathrm{tr}\,(MB) \quad \forall A \in C.$$

We conclude this preliminary section by recalling a special case of a fundamental and useful result on the trace of the product of two matrices from von Neumann (see, e.g., Marshall and Olkin [14, page 514]). Let $\lambda(A) = (\lambda_1(A), \dots, \lambda_n(A))^T$ denote the vector of eigenvalues of an $n \times n$ matrix A. We assume that the eigenvalues are ordered, e.g., $\lambda(A_1) \ge \lambda(A_2) \ge \cdots \ge \lambda_n(A)$.

Lemma 2.1. *For any $A, B \in S_n$, $\mathrm{tr}\,(AB) \le \lambda(A)^T \lambda(B)$.*

3. An interior proximal minimization algorithm. To solve the minimization problem,

$$(P) \quad \inf\{f(X) : X \succeq 0\},$$

we suggest the following basic algorithm.

Interior Proximal Method (IPM). Starting with $X^0 \in S_n^{++}$, generate the sequence $\{X^k\}$ satisfying

$$X^k = \arg\min_{X \in S_n^+} \{f(X) + \mu_k^{-1} H(X, X^{k-1})\}, \tag{3.1}$$

where $\mu_k > 0$ and $H : S_n \times S_n \to (-\infty, \infty]$ is defined by

$$H(X,Y) = \begin{cases} \mathrm{tr}\,(X \log X - X \log Y + Y - X) & \forall\, X, Y \in S_n^+ \times S_n^{++}, \\ +\infty & \text{otherwise.} \end{cases} \tag{3.2}$$

Here we adopt the convention that $0 \log 0 \equiv 0$, the zero matrix, and we recall that for any $B \in S_n^{++}$, $\log B = A$ is equivalent to $B = e^A$, where e denotes the exponential function (see [11]).

The above algorithm is a proximal-type algorithm [18], except that here, instead of using the classical quadratic term $||X - X^k||^2 = \mathrm{tr}(X - X^k)^2$, we use the nonquadratic functional H, which will guarantee that the generated sequence of matrices $\{X^k\}$ will be positive definite, thus leading to an "interior" proximal method (see section 4, Lemma 4.1).

The functional H can be seen as a natural extension of the so-called Kullback–Leibler relative entropy used to measure the "distance" between two matrices in S_n^+ (see Lemma 3.1 below), and the proposed algorithm extends recent entropy-like proximal algorithms developed for standard convex programs (see, e.g., [12], [20]) to semidefinite programs. It should be noted that many other distance-like functions could also be considered natural candidates in the algorithm (3.1). This will be briefly discussed in the last section. Our present work is limited to using the functional H. This avoids technical difficulties, makes the presentation more transparent, and reveals the potential usefulness of proximal-type algorithms for solving semidefinite programs.

It turns out that most of the important properties (but unfortunately not all—see Remark 4.1) which hold for the relative entropy defined on R_n^+ by

$$d(a,b) := \sum_{i=1}^{n} a_i \log a_i - a_i \log b_i + b_i - a_i \tag{3.3}$$

also hold for the functional H defined in (3.2). The next two results collect the properties of H relevant to the analysis of the algorithm (3.1).

Lemma 3.1. *Let H be defined by* (3.2). *Then the following hold.*

(i) *H is a continuous function on $S_n^+ \times S_n^{++}$ and $X \to H(X,Y)$ is strictly convex for any $Y \in S_n^{++}$.*

(ii) *$H(X,Y) \geq 0 \ \forall X,Y \in S_n^+ \times S_n^{++}$ and $H(X,Y)=0$ if and only if $X=Y$.*

(iii) *$H(X,Y) \geq d(\lambda(X),\lambda(Y)) \ \forall X,Y \in S_n^+ \times S_n^{++}$.*

(iv) *The level sets of $H(\cdot,Y)$ and $H(X,\cdot)$ are bounded $\forall Y \in S_n^{++}$, $\forall X \in S_n^+$, respectively.*

Proof.

(i) The continuity of H over $S_n^+ \times S_n^{++}$ follows from the fact that $X \log X$ and $\log Y$ are continuous over S_n^+ and S_n^{++}, respectively. The strict convexity of $X \to H(X,Y) \ \forall Y \in S_n^{++}$ follows from the strict convexity of $\mathrm{tr} X \log X$ on S_n^+; see [11].

(ii) Using the gradient inequality for the strictly convex function $\psi(X) = \mathrm{tr} X \log X$ we have

$$\mathrm{tr} X \log X - \mathrm{tr} Y \log Y \geq \mathrm{tr}(X-Y)(\log Y + I).$$

Here we use the fact that for any $X \in S_n^{++}, \bigtriangledown\psi(X) = \log X + I$, which can be verified by direct computation (see also [11, Sections 6.5–6.6]), where $\bigtriangledown$ denotes the gradient with respect to X. The latter inequality can be rewritten as

$$\mathrm{tr}(X\log X - X\log Y + Y - X) \geq 0 \ \forall X,Y \in S_n^+ \times S_n^{++},$$

proving the first part of (ii). The second part follows from the strict convexity of ψ, i.e., strict inequality above holds when $X \neq Y$.

(iii) Invoking Lemma 2.1 and using the fact that $\lambda_i(\log Y) = \log \lambda_i(Y)$ (cf. (2.1)) we have

$$\mathrm{tr}(X\log Y) \leq \sum_{i=1}^n \lambda_i(X) \log \lambda_i(Y) \ \forall X,Y \in S_n^+ \times S_n^{++},$$

and hence

$$\begin{array}{rcl} H(X,Y) & = & \mathrm{tr}(X\log X - X\log Y + Y - X) \\ & \geq & \mathrm{tr}(X\log X + Y - X) - \sum_{i=1}^n \lambda_i(X)\log\lambda_i(Y) \\ & = & \sum_{i=1}^n \lambda_i(X)\log\lambda_i(X) + \lambda_i(Y) - \lambda_i(X) - \lambda_i(X)\log\lambda_i(Y) \\ & = & d(\lambda(X),\lambda(Y)). \end{array}$$

(iv) Let us show that $L := \{X \in S_n^+ : H(X,Y) \leq \nu\}$ are bounded for any fixed $Y \in S_n^{++}$ and $\nu \geq 0$. Using (iii) we have

$$L \subseteq \{X \in S_n^+ : d(\lambda(X),\lambda(Y)) \leq \nu\},$$

and since $d(\cdot,b)$ has bounded level sets (see, e.g., [12]), the latter set is bounded and hence L is bounded. The proof that $H(X,\cdot)$ has bounded level sets for every fixed $X \in S_n^+$ is similar and thus omitted. □

Lemma 3.2. *Let $\{X^k\} \in S_n^+$, $\{Y^k\} \in S_n^{++}$ be bounded sequences of matrices satisfying $H(X^k,Y^k) \to 0$. Then the following hold.*

(i) $\lambda(X^k) - \lambda(Y^k) \to 0$.

(ii) $\mathrm{tr}(X^k - Y^k) \to 0$.

Proof.

(i) From Lemma 3.1(iii) we have

$$H(X^k, Y^k) \geq d(\lambda(X^k), \lambda(Y^k)) \geq 0,$$

and hence, since $H(X^k, Y^k) \to 0$, it follows that $d(\lambda(X^k), \lambda(Y^k)) \to 0$. Moreover, since $d(\lambda(X^k), \lambda(Y^k)) = \sum_{i=1}^n \varphi(\lambda_i(X^k), \lambda_i(Y^k))$ (recall that $\varphi(s,t) = s \log s - s \log t + t - s$ with $\varphi(\cdot,\cdot) \geq 0$), it follows that $\varphi(\lambda_i(X^k), \lambda_i(Y^k)) \to 0$ $\forall i = 1, \ldots, n$. Since $\{\lambda_i(X^k)\}$, $\{\lambda_i(Y^k)\}$ are bounded for every $i = 1, 2, \ldots, n$ (which follows from the boundedness of $\{X^k\}$, $\{Y^k\}$), invoking Lemma A.1 given in the Appendix with $a_k := \lambda_i(X^k)$, $b_k := \lambda_i(Y^k)$ gives the desired result.

(ii) Since $\mathrm{tr} X^k = \sum \lambda_i(X^k)$ the result follows immediately from (i). □

We conclude this section by noting two useful relations for the functional H, which can be easily verified by direct substitution using the definition of H given in (3.2) and recalling that $H \geq 0$ (Lemma 3.1(ii)).

LEMMA 3.3. *For all $A, B \in S_n^{++}$ and $C \in S_n^+$ we have*

(i) $H(C, A) - H(C, B) = \mathrm{tr}(C \log B - C \log A + A - B)$;

(ii) $\mathrm{tr}(C - B)(\log B - \log A) = H(C, A) - H(C, B) - H(B, A) \leq H(C, A) - H(C, B)$.

4. Convergence analysis of IPM. The analysis of IPM for solving SDP is similar to the analysis developed for convex minimization problems by Güler [10] and Lemaire [13] for quadratic proximal methods and its extension derived in Chen and Teboulle [8] for nonquadratic proximal methods. We derive a global convergence rate estimate for IPM in terms of function values.

We make the following assumptions for problem (P):

A0. $\inf\{f(X) \,:\, X \succeq 0\} = f_* > -\infty$,

A1. *dom* $f \cap S_n^{++} \neq \emptyset$.

We denote the set of minimizers of f by $X_* = \{X : f(X) = f_*\}$. Our first result shows that algorithm (3.1) is well defined, i.e., it generates a sequence $\{X^k\} \in S_n^{++}$.

LEMMA 4.1. *For any $Y \in S_n^{++}$ and $\mu > 0$ we have the following.*

(i) *Under* A0, *the function $X \to f(X) + \mu^{-1} H(X, Y)$ has bounded level sets.*

(ii) *If, in addition,* A1 *holds, then there exists a unique $X(Y) \in S_n^{++}$ satisfying*

$$X(Y) = \mathrm{argmin}\{f(X) + \mu^{-1} H(X, Y)\}, \tag{4.1}$$

the minimum being attained at $X(Y) \in S_n^{++}$ satisfying

$$\mu^{-1}(\log Y - \log X(Y)) \in \partial f(X(Y)), \tag{4.2}$$

where ∂f is the subdifferential of f (see (2.2)*).*

Proof. Fix $Y \in S_n^{++}$ and $\mu > 0$. By Lemma 3.1(iv) the level sets $L := \{X \in S_n^+ \,:\, H(X, Y) \leq \nu\}$ are bounded, from which it follows that, under A0, $X \to F_Y(X) := f(X) + \mu^{-1} H(X, Y)$ has bounded level sets, and hence $X(Y)$ exists. The uniqueness is implied by the strict convexity of $F_Y(X)$. Under assumption A1, writing the optimality condition for (4.1), we have, using the gradient formula $\nabla_X H(X, Y) = \log X - \log Y, X \in S_n^{++}$,

$$0 \in \partial f(X(Y)) + \mu^{-1}(\log X(Y) - \log Y) + \partial \delta(X(Y) | S_n^+), \tag{4.3}$$

where $\delta(M | S_n^+) = 0$ if $M \succeq 0$ and $+\infty$ otherwise. Note that for any sequence $X^k \in S_n^{++}$ with $X^k \to X \in S_n^+$, we have $|| \log X^k || \to \infty$; then $H(\cdot, Y)$ is essentially smooth (see Rockafellar [17, Section 26]), and we thus have $\partial_X H(X,Y) = \emptyset$ for any singular positive semidefinite matrix X. Thus from (4.3) we must have $X(Y) \succ 0$. Since $\partial\delta(M|S_n^+) = \{N \in S_n \ : \ N \preceq 0, \mathrm{tr} NM = 0\}$ (see Rockafellar [17, page 226]), then using the fact that for any real $n \times n$ matrices A, B,

$$A \succeq 0, B \succ 0, \ \mathrm{tr}(AB) = 0 \ \Rightarrow A = 0,$$

we obtain $\partial\delta(X(Y)|S_n^+) = \{0\}$, the zero matrix, and the proof is completed. □

The next result provides the key properties of IPM from which our convergence result will follow.

LEMMA 4.2. *Let $\{X^k\}$ be the sequence generated by algorithm* (3.1)*, and let $\sigma_m := \sum_{k=1}^m \mu_k$. Then we have the following.*

(i) *$f(X^k)$ is nonincreasing.*

(ii) $\mu_k(f(X^k) - f(X)) \le H(X, X^{k-1}) - H(X, X^k) \ \ \forall\, X \ \succeq 0.$

(iii) $\sigma_m(f(X^m) - f(X)) \le H(X, X^0) - H(X, X^m) \ \ \forall X \succeq 0.$

Moreover, if $X_* \neq \emptyset$, then

(iv) *$H(X, X^k)$ is nonincreasing $\forall X \in X_*$;*

(v) *$H(X^k, X^{k-1}) \to 0$ and* $\mathrm{tr}(X^k - X^{k-1}) \to 0$.

Proof.

(i) Using the definition of X^k it follows that

$$f(X^k) + \mu_k^{-1} H(X^k, X^{k-1}) \le f(X^{k-1}) + \mu_k^{-1} H(X^{k-1}, X^{k-1}),$$

and since $H(X^k, X^{k-1}) \ge 0$, $H(X^{k-1}, X^{k-1}) = 0$, we get $f(X^k) \le f(X^{k-1})$.

(ii) Using (4.2) in (2.2) we have

$$\begin{aligned} \mu_k(f(X^k) - f(X)) &\le \mathrm{tr}(X - X^k)(\log X^k - \log X^{k-1}) \\ &= H(X, X^{k-1}) - H(X, X^k) - H(X^k, X^{k-1}) \\ &\le H(X, X^{k-1}) - H(X, X^k), \end{aligned}$$

where the equality and last inequality above follow from Lemma 3.3(ii).

(iii) Summing in (ii) over $k = 1, \dots, m$, we get

$$\sum_{k=1}^m \mu^k f(X^k) - \sigma_m f(X) \le H(X, X^0) - H(X, X^m). \tag{4.4}$$

On the other hand, since by (i) $f(X^{k-1}) - f(X^k) \ge 0$, multiplying the latter by σ_{k-1} and using $\sigma_k = \mu_k + \sigma_{k-1}$ (with $\sigma_0 \equiv 0$) we get $\sum_{k=1}^m \mu^k f(X^k) \ge \sigma_m f(X^m)$, which combined with (4.4) implies the desired result.

(iv) Since $\forall\, X \in X_*$, $f(X^k) - f(X) \ge 0$, the result follows from (ii).

(v) From (iv) we have that $H(X, X^k)$ is nonincreasing $\forall X \in X_*$. Since, on the other hand, $H(X, X^k) \ge 0 \ \forall k$, we have that $H(X, X^k)$ is converging, and therefore $H(X, X^{k-1}) - H(X, X^k) \to 0$. On the other hand, we have from (ii) and (iv) that $H(X^k, X^{k-1}) \le H(X, X^{k-1}) - H(X, X^k)$. Since $H(X^k, X^{k-1}) \ge 0$, the result follows. The second part of (v) then follows immediately from Lemma 3.2(ii). □

We can now state the main convergence result for IPM.

THEOREM 4.1. *Let $\{X^k\}$ be generated by algorithm* (3.1) *and suppose that* A0 *and* A1 *are satisfied. Then*

(i) $f(X^m) - f(X) \le \sigma_m^{-1} H(X, X^0) \ \ \forall\, X \in S_n^+$;

(ii) *if $\sigma_m \to \infty$, then* $\lim_{k\to\infty} f(X^m) = f_*$;

(iii) *moreover, if the optimal set X_* is nonempty, the sequence $\{X^k\}$ is bounded and every one of its limit points is a solution of problem (P).*

Proof.

(i) This proof follows immediately from Lemma 4.2(iii) (since $H(X, X^k) \geq 0$).

(ii) Passing to the limit in (i), since $\sigma_m \to \infty$, we get $\limsup_{m\to\infty} f(X^m) \leq f(X)\ \forall X \in S_n^+$. This, combined with the fact that $f(X^m) \geq f_*$, proves the desired result.

(iii) Let $X_* \neq \emptyset$. The boundedness of the sequence $\{X^k\}$ follows from the fact that $H(X, \cdot)$ has bounded level sets for every X (Lemma 3.1(iv)) and from Lemma 4.2(iv). If $Y \in S_n^+$ is a limit point of $\{X^k\}$, and $\{X^{k_j}\}$ converges to Y, then we have $f(Y) \leq \liminf f(X^{k_j})$ by the lower semicontinuity of f. On the other hand, since we also have $f(X^{k_j}) \to f^*$, the result follows. □

Remark 4.1. The global rate of convergence estimate established in Theorem 4.1(i) is similar to the one obtained for proximal-type methods in standard convex minimization problems. Global convergence of the sequence $\{X^k\}$ itself to an optimal solution of (P) can be obtained under the assumption that the sequence $\{X^k\} \in S_n^{++}$ has a limit point $X^\infty \in S_n^{++}$. This assumption is rather stringent and we do not know at the moment if it can be removed. The difficulty came from the fact that the basic property

$$R_{++}^n \ni \{a^k\} \to a \in R_+^n \ \Rightarrow\ d(a, a^k) \to 0,$$

which is used in the global convergence proof of the corresponding entropic proximal method in R_+^n, does not hold in general in the space S_n^+ for the functional H, as shown in the following example.[1]

Example 4.1. Here we shall bring a sequence $\{Y^k\}_{k=1}^\infty \subset S_2^{++}$, which converges to $Y \in S_2^+$ and with $H(Y, Y^k) \to \infty$. Let

$$Y^k = k^{-1} \begin{pmatrix} k - 1 + e^{-k^2} & (1 - e^{-k^2})\sqrt{k-1} \\ (1 - e^{-k^2})\sqrt{k-1} & (k-1)e^{-k^2} + 1 \end{pmatrix},$$

$$Y = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

The eigenvalues of Y^k are $1, e^{-k^2}$ (and therefore $Y^k \in S_2^{++}$), and the corresponding eigenvectors are $(\sqrt{k^{-1}(k-1)}, k^{-1/2}), (-k^{-1/2}, \sqrt{k^{-1}(k-1)})$. It is obvious that $Y^k \to Y$. Now, we compute $H(Y, Y^k)$. We have

$$Y \log Y = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

$$\log Y^k = \begin{pmatrix} -k & k\sqrt{k-1} \\ k\sqrt{k-1} & -k(k-1) \end{pmatrix},$$

$$Y \log Y^k = \begin{pmatrix} -k & k\sqrt{k-1} \\ 0 & 0 \end{pmatrix}.$$

[1] We thank a referee for pointing out to us the recent work which also gives a similar example, [3, Example 7.29].

Substituting the above quantities in the definition of H (cf. (3.2)) we get

$$\begin{aligned} H(Y, Y^k) &= \mathrm{tr}(Y \log Y - Y \log Y^k + Y^k - Y) \\ &= k + \mathrm{tr}(Y^k - Y). \end{aligned}$$

Since $Y^k \to Y$, we have $\mathrm{tr}(Y^k - Y) \to 0$, and therefore $H(Y, Y^k) \to \infty$.

5. The exponential multiplier method for SDP.

The entropic proximal algorithm introduced in this paper provides the basis for developing multiplier-type methods for solving semidefinite programs, much in the same way that it has been done for standard convex programs. Here we give a dual application of algorithm (3.1) leading to an exponential method of multipliers for solving semidefinite programs of the form

$$(\mathrm{SDP}) \quad \min\{c^T x \,:\, A(x) \succeq 0\,,\; x \in R^m\},$$

where $c \in R^m$, $A(x) = A_0 + \sum_{i=1}^m x_i A_i$, $\{A_i\}_{i=0}^m \in S_n$.

We make the following assumptions regarding problem (SDP):

S1. The optimal set for (SDP) is nonempty and bounded.

S2. Slater's condition holds, i.e., $\exists \hat{x} \in R^m \,:\, A(\hat{x}) \succ 0$.

The Lagrangian associated with (SDP) is

$$L(x, U) = \begin{cases} c^T x \;-\; \mathrm{tr} U A(x) & \text{if } U \succeq 0, \\ -\infty & \text{otherwise,} \end{cases}$$

and the dual problem corresponding to (SDP) is given by

$$(\mathrm{DSDP}) \quad \max_{U \succeq 0} \inf_{x \in R^m} L(x, U), \tag{5.1}$$

which can be explicitly written as

$$(\mathrm{DSDP}) \quad \begin{cases} \max & -\mathrm{tr}\, U A_0, \\ \text{s.t.} & \mathrm{tr}\, U A_i = c_i,\; i = 1, \dots, m, \\ & U \succeq 0. \end{cases} \tag{5.2}$$

By standard convex analysis arguments (see Rockafellar [17]), under assumption S2 there is no duality gap between (SDP) and (DSDP) and, moreover, the set of optimal solutions of (DSDP) is nonempty and compact.

To solve (SDP) we propose the following multiplier-type algorithm.

Exponential Multiplier Method. Let $U^0 \succ 0$, $\mu_k \geq \mu > 0$, $\forall k$. Generate the sequence $\{x^k\} \in R^m$, $\{U^k\} \in S_n^{++}$ by

$$x^{k+1} = \arg \min_{x \in R^m} \{c^T x \;+\; \mu_k^{-1} \mathrm{tr}\, e^{-\mu_k A(x) \,+\, \log\, U_k}\}, \tag{5.3}$$

$$U^{k+1} = e^{-\mu_k A(x^{k+1}) \,+\, \log\, U_k}. \tag{5.4}$$

The above algorithm can be seen as a natural extension of the exponential multiplier method used in convex programs (see, e.g., [20]) to the SDP case. We prove below that algorithm (5.3)–(5.4) shares similar properties.

First, we have to show that the sequence generated by (5.3) is well defined. For that purpose, it is enough to prove that $F(x)$ is coercive, i.e., that $\lim_{\|x\| \to +\infty} F(x) = +\infty$, where for any fixed $\mu > 0$, $c \in R^m$, $B \in S_n$ we define

$$F(x) := c^T x + \mu^{-1} \mathrm{tr} e^{-\mu A(x) + B}.$$

LEMMA 5.1. *Under assumption* S1, *the minimum set of F is a nonempty and bounded set.*

Proof. We prove the result by contradiction. Suppose that F is not coercive, i.e., some level set of F is not bounded, then $\exists\{x^k\} \subset R^m$ such that

$$\| x^k \| \to \infty, \lim_{k\to\infty} x^k / \| x^k \| = d \neq 0$$

with $F(x^k) \le \delta$ for some $\delta \in R$. Since $\mathrm{tr} e^V > 0 \,\forall V \in S_n$, then $F(x^k) \le \delta$ implies that

$$c^T x^k < \delta, \tag{5.5}$$

$$\mathrm{tr} e^{-\mu A(x^k)+B} = \sum_{i=1}^m e^{\lambda_i(-\mu A(x^k)+B)} \le \mu(\delta - c^T x^k), \tag{5.6}$$

where $\lambda_i(\cdot)$ denotes the eigenvalues of $-\mu A(x^k) + B$. From (5.5) we obtain $c^T d \le 0$, while from (5.6) we get

$$e^{\lambda_i(-\mu A(x^k)+B)} \le \mu(\delta - c^T x^k) \;\; \forall i = 1, \ldots, m.$$

Since $\lambda_i(\cdot)$ is homogeneous, then taking the log on both sides, dividing by $\| x^k \|$, and passing to the limit, we obtain $\lambda_i(A(d) - A_0) \ge 0 \;\; \forall i = 1, \ldots, m$, namely,

$$A(d) - A_0 \succeq 0, \tag{5.7}$$

which is precisely the recession cone of $\{x : \; A(x) \succeq 0\}$. Since we showed earlier that $c^T d \le 0$, we obtain a contradiction to the boundedness of the optimal set of (SDP) assumed in S1. □

Next, we show that the sequence $\{U^k\}$ generated above is nothing but the sequence produced by algorithm (3.1) when applied to the dual problem (DSDP).

LEMMA 5.2. *The sequence $\{U^k\}$ generated by the multiplier method* (5.3)–(5.4) *is obtained via the iteration*

$$U^{k+1} = \arg\max_{U \succ 0}\{h(U) - \mu_k^{-1} H(U, U^k)\}, \tag{5.8}$$

where $h(U) := \inf_{x\in R^n} L(x, U)$ is the dual objective of (SDP).

Proof. First we show that

$$-A(x^{k+1}) \in \partial h(U^{k+1}).$$

Indeed, the optimality condition for (5.3) yields

$$\begin{aligned} 0 &= \frac{\partial}{\partial x_i}(c^T x \, + \, \mu_k^{-1} \mathrm{tr}\, e^{-\mu_k A(x) \, + \, \log \, U_k}) \,|_{x=x^{k+1}} \\ &= c_i - \mathrm{tr} A_i e^{-\mu_k A(x^{k+1}) \, + \, \log \, U_k} \\ &= c_i - \mathrm{tr} A_i U^{k+1} \;\; \forall i = 1, \ldots, m, \end{aligned}$$

where in the last equality we use (5.4). Therefore, for each $i = 1, \ldots, m$ we have

$$\mathrm{tr} A_i U^{k+1} = c_i,$$

showing that x^{k+1} is also minimizing the Lagrangian $L(x, U^{k+1})$, and hence that $h(U^{k+1}) = L(x^{k+1}, U^{k+1})$. Now we have

$$\begin{aligned} h(U) &= \inf_x \{c^T x - \mathrm{tr} UA(x)\} \\ &\le c^T x^{k+1} - \mathrm{tr} UA(x^{k+1}) \\ &= c^T x^{k+1} - \mathrm{tr} U^{k+1} A(x^{k+1}) - \mathrm{tr}(U - U^{k+1})A(x^{k+1}) \\ &= h(U^{k+1}) - \mathrm{tr}(U - U^{k+1})A(x^{k+1}), \end{aligned}$$

and hence, recalling that $h(U)$ is concave, $-A(x^{k+1}) \in \partial h(U^{k+1})$. Using (5.4) we also have

$$\log U^{k+1} - \log U^k = -\mu_k A(x^{k+1}),$$

which combined with the above inclusion gives

$$\mu_k^{-1}(\log U^k - \log U^{k+1}) \in \partial h(U^{k+1}),$$

and which by Lemma 4.1 is precisely the optimality condition for (5.8). ∎

We can then prove the following convergence result.

THEOREM 5.1. *Let* $\{x^k\}$, $\{U^k\}$ *be generated by* (5.3)–(5.4) *and assume that* S1 *and* S2 *hold. Then we have the following.*

(a) *The dual sequence* $U^k \in S_n^{++}$ *is bounded, and all of its limit points are optimal dual solutions.*

(b) $\mathrm{tr}\, U^k A(x^k) \to 0$ *as* $k \to +\infty$.

(c) *Let* $\overline{x}^k = \sum_{l=1}^k \eta_l x^l$ *with* $\eta_l := \mu_l/\nu_k > 0$ *and* $\nu_k := \sum_{l=1}^k \mu_l$. *Then* $\liminf_{k\to\infty} \lambda_{\min}(A(\overline{x}^k)) \succeq 0$.

(d) *Denote by* h^* *the optimal value of the dual problem* (DSDP). *Then* $c^T x^k \to h^*$ *(and thus* $c^T \overline{x}^k \to h^*$*).*

(e) $\{\overline{x}^k\}$ *is bounded.*

(f) *Every limit point of* $\{\overline{x}^k\}$ *is an optimal solution* x^* *of* (SDP).

(g) $\lim_{k\to\infty} c^T x^k = \lim_{k\to\infty} h(U^k) = c^T x^*$.

Proof. (a) From Lemma 5.1 we have that $\{U^k\}$ is the sequence generated by the proximal-like algorithm (3.1) applied to (DSDP). Since by S2 the set of optimal solutions of (DSDP) is nonempty and compact, invoking Theorem 4.1(iii) proves the statement.

(b) Equation (5.4) implies that $-\mu_k A(x^{k+1}) = \log U^{k+1} - \log U^k$. On the other hand, we have

$$\begin{aligned} H(U^{k+1}, U^k) &= \mathrm{tr}(U^{k+1}(\log U^{k+1} - \log U^k) + U^k - U^{k+1}) \\ &= -\mu_k \mathrm{tr} U^{k+1} A(x^{k+1}) + \mathrm{tr}(U^k - U^{k+1}). \end{aligned}$$

But from Lemma 4.2(v) we have $H(U^{k+1}, U^k) \to 0$, $\mathrm{tr}(U^k - U^{k+1}) \to 0$. Therefore

$$\mu_k \mathrm{tr} U^{k+1} A(x^{k+1}) \to 0.$$

Since $\mu_k \ge \mu > 0$, we get $\mathrm{tr} U^{k+1} A(x^{k+1}) \to 0$.

(c) Since $A(x)$ is affine we have, with $\overline{x}^k = \sum_{l=1}^k \eta_l x^l$, $\eta_l = \mu_l/\nu_k$, and $\nu_k := \sum_{l=1}^k \mu_l$,

$$\begin{aligned} A(\overline{x}^k) = \sum_{l=1}^k \eta_l A(x^l) &= \sum_{l=1}^k (\log U^{l-1} - \log U^l)/\nu_k \\ &= (\log U^0 - \log U^k)/\nu_k. \end{aligned} \tag{5.9}$$

The second equality in (5.9) is from (5.4). Since $\lambda_{\min}(X)$ is a super additive function, we have $\lambda_{\min}(A(\overline{x}^k)) \geq \lambda_{\min}(\log U^0)/\nu_k + \lambda_{\min}(-\log U^k)/\nu_k$. Since $\nu_k \to \infty$ (recall that $\mu_k \geq \mu > 0$), the first term tends to zero, and thus it remains to prove that $\liminf \lambda_{\min}(-\log U^k)/\nu_k \geq 0$. Note that

$$\begin{aligned} \liminf \lambda_{\min}(-\log U^k)/\nu_k &= -\limsup \nu_k^{-1} \lambda_{\max}(\log U^k) \\ &= -\limsup \nu_k^{-1} \log(\lambda_{\max}(U^k)). \end{aligned} \tag{5.10}$$

Since $\{U^k\} \subset S_n^{++}$ is bounded, $\lambda_{\max}(U^k) \leq \lambda$ for some $\lambda > 0$, and thus $\log(\lambda_{\max}(U^k)) \leq \log \lambda$. Therefore

$$-\limsup \nu_k^{-1} \log(\lambda_{\max}(U^k)) \geq -\limsup \nu_k^{-1} \log \lambda.$$

Since $\nu_k \to \infty$, we get the desired result from (5.10).

(d) From (ii) we have $\mathrm{tr} U^k A(x^k) \to 0$. On the other hand, since $\{U^k\}$ is feasible for (DSDP), $\mathrm{tr} U^k A(x^k) = c^T x^k + \mathrm{tr} U^k A_0 = c^T x^k - h(U^k)$. From Theorem 4.1(ii), we know that $h(U^k) \to h^*$, and the result follows.

(e) Denote $y^k := \overline{x}^k$. Suppose by contradiction that y^k is unbounded. Since the optimal set of (SDP) is bounded, denote by x its element with the maximal norm. Define $\alpha_k = 1 - 3 \parallel x \parallel /(\parallel y^k - x \parallel)$. Since $\parallel y^k \parallel \to \infty$, $\exists k_0$ s.t. $0 < \alpha_k < 1$ $\forall k \geq k_0$. Let $z^k = \alpha_k x + (1-\alpha_k) y^k$. By the triangle inequality, $2 \parallel x \parallel \leq \parallel z^k \parallel \leq 4 \parallel x \parallel$, which means that z^k is bounded. If we can prove that a limit point of $\{z^k\}$ is an optimal solution to (SDP), we are done, since this will be a contradiction to the maximality of the norm of x. Let $z^k \to z$. Since $A(z^k) = \alpha_k A(x) + (1-\alpha_k) A(y^k)$, $\alpha_k \to 1$, and $\liminf \lambda_{\min}(A(y^k)) \succeq 0$, we get that $A(z) \succeq 0$, and therefore z is a feasible point for (SDP), and thus $c^T z \geq c^T x$. On the other hand, since $c^T y^k \to h^* \leq c^T x$ (by weak duality), we get $c^T z^k = \alpha_k c^T x + (1-\alpha_k) c^T y^k \leq c^T x$. Therefore, $c^T x = c^T z$, and z is in the optimal set of (SDP).

(f) Let $\overline{x}$ be a limit point of $\{\overline{x}^k\}$. Since $\liminf \lambda_{\min}(A(\overline{x}^k)) \succeq 0$, we have that $\overline{x}$ is a feasible point of (SDP), and therefore

$$c^T \overline{x} \geq c^T x^*. \tag{5.11}$$

On the other hand, we know from (iv) that $c^T \overline{x}^k \to h^*$, and therefore

$$c^T \overline{x} = h^* \leq c^T x^*. \tag{5.12}$$

Combining (5.11) and (5.12), we have

$$c^T \overline{x} = h^* = c^T x^*. \tag{5.13}$$

(g) The fact that $\lim c^T x^k = \lim h(U^k)$ is due to (d), and the last equality is given in (5.13). □

Note that the above convergence properties are very similar to the ones derived for standard convex programs [20], except that here global convergence of the dual sequence to an optimal dual solution is not guaranteed. However, one still has convergence in function values, and by applying Theorem 4.1, one has the global convergence rate estimate

$$\mathrm{tr}(U^k - U^*)A_0 \leq \left(\sum_{p=1}^{k-1} \mu_p \right)^{-1} H(U^*, U^0),$$

where U^* is an optimal solution of (DSDP).

6. Conclusions and extensions.

The motivation of this paper was to explore the possibility of developing proximal and multiplier methods for solving SDPs, along the lines of proximal-like methods used for standard convex programs. The convergence properties established for the IPM and the related dual exponential multiplier method demonstrate that, theoretically, algorithms of these types appear viable, and we hope that the present work will contribute to future developments and implementations of augmented Lagrangian methods for solving SDPs.

In the present study, we restricted our analysis to the special but important distance-like functional H. This allowed us to avoid some technical difficulties and make the presentation more transparent. Moreover, even for more general distances, it should be recalled that this functional plays a key role in the convergence analysis.

Several other distance-like functions could be used by mimicking the general class of nonquadratic proximal maps developed in [19] in the space R^n_+ to the space S^+_n, and many of our results, and our analysis, can be extended with these distance-like functionals to develop a general approach for multiplier/penalty methods for semidefinite programs. As an illustration of another interesting possible choice for the functional H that could be used in IPM, and for which it can be shown that our convergence results applied, consider the functional

$$H_1(X,Y) \;=\; \mathrm{tr}XY^{-1} - \log\det XY^{-1} - n \;\; \forall X \succ 0,\; Y \succ 0, \tag{6.1}$$

where det$A$ denotes the determinant of $n \times n$ matrix A. It can be shown that the corresponding multiplier method used to solve (SDP) obtained by applying IPM with H_1 on the dual of (SDP) leads to a logarithmic-barrier-type method of the following form.

Log-barrier multiplier method. Let $U^0 \succ 0$, $\mu_k \geq \mu > 0 \forall k$. Generate the sequence $\{x^k\} \in R^m$, $\{U^k\} \in S^{++}_n$ by

$$x^{k+1} = \arg\min_{x \in R^m} \{c^T x \;-\; \mu_k^{-1} \log\det(I + \mu_k U^k A(x))\}, \tag{6.2}$$

$$U^{k+1} = (I + \mu_k U^k A(x^{k+1}))^{-1} U^k = ((U^k)^{-1} + \mu_k A(x^{k+1}))^{-1}. \tag{6.3}$$

Appendix.

LEMMA A.1. *Let $\{a_k\}$, $\{b_k\}$ be bounded sequences in R_{++}, and let $\varphi(s,t) = s\log s - s\log t + t - s,\; s \geq 0, t > 0$. Suppose that $\varphi(a_k, b_k) \to 0$. Then $c_k := a_k - b_k \to 0$.*

Proof. $\{c_k\}$ is bounded (because $\{a_k\}$, $\{b_k\}$ are such). Argue by contradiction; suppose $\{c_k\}$ has a subsequence converging to $c \neq 0$. Without loss of generality, assume $c_k \to c$. Let a_{k_j} be a subsequence of a_k converging to a. Let $b_{k_{j_l}}$ be a subsequence of b_{k_j} converging to b. Denote $n := k_{j_l}$. We have $c \leftarrow c_n = a_n - b_n \to a - b$. There are four possibilities for the values of a and b: (1) $a = b = 0$. (2) $a \neq 0$, $b \neq 0$. (3) $a = 0$, $b \neq 0$. (4) $a \neq 0$, $b = 0$. Let us split our discussion into the four possible cases.

(i) $a = b = 0$. This is impossible, because $c = a - b \neq 0$.

(ii) $a \neq 0, b \neq 0$. φ is continuous on $R_+ \times R_{++}$, and therefore $\varphi(a_n, b_n) \to \varphi(a,b)$. We know that $\varphi(a,b) = 0$ if and only if $a = b$. Thus we get $c = a - b = 0$ in contradiction to the assumption that $c \neq 0$.

(iii) $a = 0$, $b \neq 0$. This is the same argument as in (ii).

(iv) $a \neq 0$, $b = 0$. Writing down the explicit expression for $\varphi(a_n, b_n)$, we get $\varphi(a_n, b_n) \to \infty$, a contradiction to the assumption $\varphi(a_n, b_n) \to 0$.

Therefore every limit of a subsequence of $\{c_k\}$ must be 0. This, together with the fact that $\{c_k\}$ is bounded, gives $c_k \to 0$. □

REFERENCES

[1] F. Alizadeh, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.

[2] A. Auslender and M. Haddou, *An interior-proximal method for convex linearly constrained problems and its extension to variational inequalities*, Math. Programming, 71 (1995), pp. 77–100.

[3] H. H. Bauschke and J. M. Borwein, *Legendre functions and the method of random Bregman projections*, J. Convex Anal., 4 (1997), pp. 27–67.

[4] A. Ben-Tal, I. Yuzefovich, and M. Zibulevsky, *Penalty/Barrier Multiplier Methods for Minimax and Constrained Smooth Convex Programs*, Research report 9-92, Optimization Laboratory, Technion, Israel, 1992.

[5] A. Ben-Tal and M. Zibulevsky, *Penalty/barrier multiplier methods for convex programming problems*, SIAM J. Optim., 7 (1997), pp. 347–366.

[6] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[7] M. G. Breitfeld and D. F. Shanno, *Computational Experience with Modified Log-barrier Methods for Nonlinear Programming*, Rutcor research report 17-93, Rutgers University, New Brunswick, NJ, 1993.

[8] G. Chen and M. Teboulle, *Convergence analysis of a proximal-like minimization algorithm using Bregman functions*, SIAM J. Optim., 3 (1993), pp. 538–543.

[9] J. Eckstein, *Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming*, Math. Oper. Res., 18 (1993), pp. 202–226.

[10] O. Güler, *On the convergence of the proximal point algorithm for convex minimization*, SIAM J. Control Optim., 29 (1991), pp. 403–419.

[11] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.

[12] A. Iusem and M. Teboulle, *Convergence rate analysis of nonquadratic proximal methods for convex and linear programming*, Math. Oper. Res., 20 (1995), pp. 657–677.

[13] B. Lemaire, *About the convergence of the proximal method*, in Advances in Optimization, Lecture Notes in Econom. and Math. Systems 382, D. Pallaschke, ed., Springer-Verlag, Berlin, 1992, pp. 39–51.

[14] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and its Applications*, Academic Press, New York, 1979.

[15] B. Martinet, *Perturbation des méthodes d'optimisation. Applications*, RAIRO Anal. Numér., 12 (1978), pp. 153–171.

[16] R. A. Polyak, *Modified barrier functions (theory and methods)*, Math. Programming, 54 (1992), pp. 177–222.

[17] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[18] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[19] M. Teboulle, *Entropic proximal mappings with application to nonlinear programming*, Math. Oper. Res., 17 (1992), pp. 670–690.

[20] P. Tseng and D. Bertsekas, *On the convergence of the exponential multiplier method for convex programming*, Math. Programming, 60 (1993), pp. 1–19.

[21] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

SIAM J. Optim.
Vol. 9, No. 1, pp. 14–32

ON THE ACCURATE IDENTIFICATION OF ACTIVE CONSTRAINTS*

FRANCISCO FACCHINEI[†], ANDREAS FISCHER[‡], AND CHRISTIAN KANZOW[§]

Abstract. We consider nonlinear programs with inequality constraints, and we focus on the problem of identifying those constraints which will be active at an isolated local solution. The correct identification of active constraints is important from both a theoretical and a practical point of view. Such an identification removes the combinatorial aspect of the problem and locally reduces the inequality constrained minimization problem to an equality constrained problem which can be more easily dealt with. We present a new technique which identifies active constraints in a neighborhood of a solution and which requires neither complementary slackness nor uniqueness of the multipliers. We also present extensions to variational inequalities and numerical examples illustrating the identification technique.

Key words. constrained optimization, variational inequalities, active constraints, degeneracy, identification of active constraints

AMS subject classifications. 90C30, 65K05, 90C33, 90C31

PII. S1052623496305882

1. Introduction.

In this paper we consider the problem of identifying the constraints which are active at an isolated stationary point of the nonlinear program

$$(\mathcal{P}) \qquad \min \quad f(x) \quad \text{s.t.} \quad g(x) \geq 0,$$

where it is assumed that the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are at least continuously differentiable. More specifically, we are interested in the following question: given an $(x, \lambda) \in \mathbb{R}^{n+m}$ belonging to a sufficiently small neighborhood of a Karush–Kuhn–Tucker (KKT) point $(\bar{x}, \bar{\lambda})$ of problem $(\mathcal{P})$, is it possible to estimate correctly, on the basis of the problem data in x, the set of indices

$$I_0 := \{i |\, g_i(\bar{x}) = 0\}$$

of the active constraints? The correct identification of active constraints is important from both a theoretical and a practical point of view. Such an identification, by removing the difficult combinatorial aspect of the problem, locally reduces the inequality constrained minimization problem to an equality constrained problem which is much easier to deal with. In particular, the study of the local convergence rate of most algorithms for problem $(\mathcal{P})$ implicitly or explicitly depends on the fact that I_0 is eventually identified.

*Received by the editors June 28, 1996; accepted for publication (in revised form) September 17, 1997; published electronically October 30, 1998. This work was partially supported by NATO grant CRG 960137.

http://www.siam.org/journals/siopt/9-1/30588.html

[†] Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza," Via Buonarroti 12, I-00185 Roma, Italy (soler@dis.uniroma1.it).

[‡] University of Dortmund, Department of Mathematics, D-44221 Dortmund, Germany (fischer@mathematik.uni-dortmund.de).

[§] Institute of Applied Mathematics, University of Hamburg, Bundesstrasse 55, D-20146 Hamburg, Germany (kanzow@math.uni-hamburg.de).

Theoretically, the identification of the active constraints is not difficult if strict complementarity holds at the solution; see the discussion in the next section. However, as far as we are aware, to date no technique can successfully identify all active constraints if the strict complementary slackness assumption is violated, except in the case of linear (complementarity) problems (see [7, 8, 18]). In this paper we present a new technique which, under mild assumptions, correctly identifies active constraints in a neighborhood of a KKT point. This technique appears to improve on existing techniques. In particular, it enjoys the following properties:

(i) It is simple and independent of the algorithm used to generate the point (x, λ).
(ii) It does not require strict complementary slackness.
(iii) It does not require uniqueness of the multipliers.
(iv) It is able to handle problems with nonlinear constraints.
(v) It does not rely on any convexity assumption.
(vi) In the case of unique multipliers it also permits the correct identification of strongly active constraints.
(vii) The identification technique can also be applied to the Karush–Kuhn–Tucker system arising from variational inequalities.

Strategies for identifying active constraints are part of the optimization folklore [1, 12, 14]; however, they almost invariably lack some of the good characteristics listed above. In the last 10 years special attention has been devoted to this problem in the field of interior point methods for linear (complementarity) problems [7, 8, 18], where satisfactory results have been reached. Recent works on the nonlinear case include [6, 9, 25], where the case of box constraints is considered, and [10, 37, 38], where the general nonlinear case is studied. Related material can also be found in [2, 3, 4], which deal with the problem of establishing whether a sequence $\{x^k\}$ converging to a solution $\bar{x}$, in some way, eventually identifies the set I_0.

We remark that, in order to identify the active set, we suppose that we are given a pair (x, λ) of primal and dual variables. If we think of algorithmic applications of the results in this paper, we stress that most algorithms will produce a sequence of primal and dual variables. Even in the rare cases in which this does not occur, it is usually possible, under reasonable assumptions, to generate a continuous dual estimate by using a *multiplier function*; see, e.g., [10] and the references therein.

This paper is organized as follows. In the next section we introduce the identification technique and prove its main properties. The identification technique critically depends on the definition of what we call an *identification function*. Therefore, the more technical section 3 is devoted to the definition of identification functions under different sets of assumptions. In section 4 we give some numerical examples and in section 5 we make some final comments.

We conclude this section by providing a list of the notation employed. Throughout the paper, $\|\cdot\|$ indicates the Euclidean vector norm. The symbol B_ϵ denotes the open Euclidean ball with radius $\epsilon > 0$ and center at the origin; the dimension of the space will be clear from the context. The Euclidean distance of a point y from a nonempty set S is abbreviated by $\mathrm{dist}[y, S]$. We write x_+ for the vector $\max\{0, x\}$, where the maximum is taken componentwise. We set $I := \{1, \dots, m\}$ and make use of the notation x_J for $J \subseteq I$ in order to represent the $|J|$-dimensional vector with components $x_i, i \in J$. Finally, the transposed Jacobian of the vector-valued mapping g at a point x will be denoted by $\nabla g(x)$, i.e., the ith column of this matrix is the gradient $\nabla g_i(x)$.

2. Identifying active constraints.

Following the usual terminology in constrained optimization, we call a vector $\bar{x} \in \mathbb{R}^n$ a *stationary point* of $(\mathcal{P})$ if there exists a vector $\bar{\lambda} \in \mathbb{R}^m$ such that $(\bar{x}, \bar{\lambda})$ solves the *Karush–Kuhn–Tucker system*

$$\begin{array}{rcl} \nabla f(x) - \nabla g(x)\lambda & = & 0, \\ \lambda & \geq & 0, \\ g(x) & \geq & 0, \\ \lambda^T g(x) & = & 0. \end{array} \tag{2.1}$$

The pair $(\bar{x}, \bar{\lambda})$ is called a *KKT point* of problem $(\mathcal{P})$. In the following, $\bar{x}$ will always denote a fixed, isolated stationary point, so that there is a neighborhood of $\bar{x}$ which does not contain any further stationary point of $(\mathcal{P})$. Moreover, we shall indicate by Λ the set of all Lagrange multipliers $\bar{\lambda}$ associated with $\bar{x}$ and indicate by $\mathcal{K}$ the set of all KKT points associated with $\bar{x}$, that is,

$$\Lambda := \{\bar{\lambda} \,|\, (\bar{x}, \bar{\lambda}) \text{ solves (2.1)}\}, \qquad \mathcal{K} := \{(\bar{x}, \bar{\lambda}) \,|\, \bar{\lambda} \in \Lambda\}.$$

The set Λ is closed and convex and therefore, so is the set $\mathcal{K}$. Gauvin [13] showed that Λ is bounded (and hence compact) if and only if the *Mangasarian–Fromovitz constraint qualification* (MFCQ) is satisfied, i.e., if and only if

$$\sum_{i \in I_0} u_i \nabla g_i(\bar{x}) = 0, \quad u_i \geq 0 \ \forall i \in I_0 \qquad \Longrightarrow \qquad u_i = 0 \ \forall i \in I_0.$$

On the other hand, Kyparisis [24] showed that Λ reduces to a singleton if and only if the *strict Mangasarian–Fromovitz constraint qualification* (SMFCQ) holds, i.e., if and only if

$$\sum_{i \in I_0} u_i \nabla g_i(\bar{x}) = 0, \quad u_i \geq 0 \ \forall i \in I_0 \setminus I_+ \qquad \Longrightarrow \qquad u_i = 0 \ \forall i \in I_0,$$

where I_+ denotes the index set

$$I_+ := \{i \in I_0 \,|\, \exists \bar{\lambda} \in \Lambda : \bar{\lambda}_i > 0\}.$$

In particular, the *linear independence constraint qualification* (LICQ), i.e., the linear independence of the gradients of the active constraints, implies that Λ is a singleton.

Our basic aim is to construct a rule which is able to assign to every point (x, λ) an estimate $A(x, \lambda) \subseteq I$ so that $A(x, \lambda) = I_0$ holds if (x, λ) lies in a suitably small neighborhood of a point $(\bar{x}, \bar{\lambda}) \in \mathcal{K}$.

Usually, estimates of this kind are obtained by comparing the value of $g_i(x)$ to the value of λ_i. For example, it can easily be shown that the set

$$I_\oplus(x, \lambda) := \{i \in I \,|\, g_i(x) \leq \lambda_i\}$$

coincides with the set I_0 for all (x, λ) in a sufficiently small neighborhood of a KKT point $(\bar{x}, \bar{\lambda})$ which satisfies the strict complementarity condition. If this condition is violated, then only the inclusion

$$I_\oplus(x, \lambda) \subseteq I_0 \tag{2.2}$$

holds. Furthermore, if Λ is a singleton, then we also have, in a sufficiently small neighborhood of $(\bar{x}, \bar{\lambda})$,

$$I_+ \subseteq I_\oplus(x, \lambda) \subseteq I_0. \tag{2.3}$$

This relation was exploited to construct locally superlinearly convergent QP-free optimization algorithms when the unique multiplier $\bar{\lambda}$ does not satisfy the strict complementarity condition; see, e.g., [10, 22, 36].

We refer the reader to [1, 10] and the references therein for a more complete discussion of these kinds of results. An analysis of results established in the literature shows that this conclusion holds in general: if strict complementarity is satisfied, it is usually possible to identify correctly the active constraint set, otherwise a relation such as (2.3) is the best result that has been established in the general nonlinear case.

To overcome this situation we propose to compare $g_i(x)$ to a quantity which goes to 0 at a known rate if (x, λ) converges to a point in the KKT set $\mathcal{K}$. To this end, we introduce the notion of an *identification function.*

DEFINITION 2.1. *A function $\rho : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_+$ is called an* identification function *for $\mathcal{K}$ if*

(a) *ρ is continuous on $\mathcal{K}$,*

(b) *$(\bar{x}, \bar{\lambda}) \in \mathcal{K}$ implies $\rho(\bar{x}, \bar{\lambda}) = 0$,*

(c) *if $(\bar{x}, \bar{\lambda})$ belongs to $\mathcal{K}$, then*

$$\lim_{\substack{(x,\lambda)\to(\bar{x},\bar{\lambda}) \\ (x,\lambda)\notin\mathcal{K}}} \frac{\rho(x, \lambda)}{\operatorname{dist}[(x, \lambda), \mathcal{K}]} = +\infty. \tag{2.4}$$

In the next section we shall give examples of how to build, under appropriate assumptions, identification functions. Basically, Definition 2.1 says that a function is an identification function if it goes to 0 when approaching the set $\mathcal{K}$ at a "slower" rate than the distance from the set $\mathcal{K}$. We note that $\operatorname{dist}[(x, \lambda), \mathcal{K}] > 0$ whenever $(x, \lambda) \notin \mathcal{K}$ since $\mathcal{K}$ is a closed set; hence the denominator in (2.4) is always nonzero.

Using Definition 2.1 it is easy to prove that the index set

$$A(x, \lambda) := \{i \in I \mid g_i(x) \leq \rho(x, \lambda)\} \tag{2.5}$$

correctly identifies all active constraints if (x, λ) is sufficiently close to the KKT set $\mathcal{K}$.

THEOREM 2.2. *Let ρ be an identification function for $\mathcal{K}$. Then, for any $\bar{\lambda} \in \Lambda$, an $\epsilon = \epsilon(\bar{\lambda}) > 0$ exists such that*

$$A(x, \lambda) = I_0 \qquad \forall (x, \lambda) \in \{(\bar{x}, \bar{\lambda})\} + B_\epsilon. \tag{2.6}$$

Proof. Since g is continuously differentiable, g is locally Lipschitz continuous. Hence there exists a constant $c > 0$ such that, for all x sufficiently close to $\bar{x}$,

$$g_i(x) \leq g_i(\bar{x}) + c\|x - \bar{x}\| \qquad (i \in I). \tag{2.7}$$

Suppose now that $g_i(\bar{x}) = 0$. Then, using (2.4) and (2.7), we obviously have, for $(x, \lambda) \notin \mathcal{K}$ in a sufficiently small neighborhood of $(\bar{x}, \bar{\lambda})$,

$$g_i(x) \leq c\|x - \bar{x}\| \leq c \operatorname{dist}[(x, \lambda), \mathcal{K}] \leq \rho(x, \lambda),$$

so that, by (2.5), $i \in A(x, \lambda)$.

If, instead, $(x, \lambda) \in \mathcal{K}$, then we have $x = \bar{x}$ by the local uniqueness of $\bar{x}$. From the definition of an identification function, we also have $\rho(x, \lambda) = 0$, so that

$$g_i(x) = g_i(\bar{x}) = 0 \leq \rho(x, \lambda),$$

and, also in this case, $i \in A(x, \lambda)$.

On the other hand, if $g_i(\bar{x}) > 0$, it follows, by continuity, that $i \notin A(x, \lambda)$ if (x, λ) is sufficiently close to $(\bar{x}, \bar{\lambda})$. Therefore, for any $\bar{\lambda} \in \Lambda$, we can find $\epsilon = \epsilon(\bar{\lambda}) > 0$ such that (2.6) is satisfied. □

From the previous theorem it is obvious that there exists an open set containing $\mathcal{K}$, where the identification of the active constraints is correct. Using the MFCQ condition we can obtain a somewhat stronger result.

THEOREM 2.3. *Let ρ be an identification function for $\mathcal{K}$. If the MFCQ condition holds, then there is an $\epsilon > 0$ such that*

$$A(x, \lambda) = I_0 \qquad \forall (x, \lambda) \in \mathcal{K} + B_\epsilon.$$

Proof. By the previous theorem, for every $(\bar{x}, \bar{\lambda}) \in \mathcal{K}$, there exists a neighborhood $\Omega(\epsilon(\bar{\lambda})) = \{(\bar{x}, \bar{\lambda})\} + B_{\epsilon(\bar{\lambda})}$ such that $A(x, \lambda) = I_0$ for every $(x, \lambda) \in \Omega(\epsilon(\bar{\lambda}))$. The collection of open sets $\Omega(\epsilon(\bar{\lambda}))$, $\bar{\lambda} \in \Lambda$ obviously forms an open cover of $\mathcal{K}$. Since the set $\mathcal{K}$ is compact in view of the MFCQ condition, we can extract from the infinite cover $\Omega(\epsilon(\bar{\lambda}))$, with $\bar{\lambda}$ such that $(\bar{x}, \bar{\lambda}) \in \mathcal{K}$, a finite subcover $\Omega(\epsilon(\bar{\lambda}_j))$, with $j = 1, \ldots, s$. Then it is easy to see that the theorem holds with $\epsilon := \min_{j=1,\ldots,s}\{\epsilon(\bar{\lambda}_j)\}$. □

If the SMFCQ holds, it is even possible to identify the set of *strongly active constraints* at $\bar{x}$, i.e., the set of constraints whose multipliers are positive. To this end, let $A_+(x, \lambda)$ be defined by

$$A_+(x, \lambda) := \{i \in A(x, \lambda) \,|\, \lambda_i \geq \rho(x, \lambda)\}.$$

The following theorem holds.

THEOREM 2.4. *Let ρ be an identification function for $\mathcal{K}$. If the SMFCQ holds at $\bar{x}$, then there is an $\epsilon > 0$ such that*

$$A_+(x, \lambda) = I_+ \qquad \forall (x, \lambda) \in \mathcal{K} + B_\epsilon.$$

Proof. We first recall that the SMFCQ implies that Λ reduces to a singleton, i.e., $\Lambda = \{\bar{\lambda}\}$. Theorem 2.2 shows that $A_+(x, \lambda) \subseteq I_0$ for all (x, λ) in a certain neighborhood of $(\bar{x}, \bar{\lambda})$. Now, consider an index $i \in I_+$. By continuity, this implies $i \in A_+(x, \lambda)$ in a sufficiently small neighborhood of $(\bar{x}, \bar{\lambda})$. On the other hand, let $i \in I \setminus I_+$. Then, $\bar{\lambda}_i = 0$ and, for all (x, λ) in a sufficiently small neighborhood of $(\bar{x}, \bar{\lambda})$, we have

$$\lambda_i \leq |\lambda_i - \bar{\lambda}_i| \leq \|(x, \lambda) - (\bar{x}, \bar{\lambda})\| = \operatorname{dist}[(x, \lambda), \mathcal{K}] \leq \rho(x, \lambda)/2 < \rho(x, \lambda).$$

This means $i \notin A_+(x, \lambda)$. Thus, $A_+(x, \lambda) = I_+$ for all (x, λ) sufficiently close to $\mathcal{K} = (\bar{x}, \bar{\lambda})$. □

Until now we have made reference to the Karush–Kuhn–Tucker system (2.1) which expresses first order necessary optimality conditions for the minimization problem $(\mathcal{P})$. We showed how the active constraints associated with an isolated stationary point $\bar{x}$ can be identified. However, the fact that the Karush–Kuhn–Tucker system (2.1) derives from an optimization problem plays no role in the theory developed. What we actually proved is the following: given a solution $(\bar{x}, \bar{\lambda})$ of a system with the structure of system (2.1) and with an isolated x-part, we can identify, in a suitable neighborhood of this solution, those inequalities which hold as equalities at the solution $(\bar{x}, \bar{\lambda})$. Therefore, if we consider the KKT system

$$\begin{array}{rcl} F(x) - \nabla g(x)\lambda & = & 0, \\ \lambda & \geq & 0, \\ g(x) & \geq & 0, \\ \lambda^T g(x) & = & 0, \end{array} \tag{2.8}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is any continuous function, the theory of this section goes through without any change. This is an important observation, since it allows us to extend the theory developed so far to the identification of active constraints for the *variational inequality problem*:

$$\text{find } \bar{x} \in X \quad \text{such that} \quad F(\bar{x})^T(x - \bar{x}) \geq 0 \quad \forall x \in X,$$

where $X := \{x \in \mathbb{R}^n | \, g(x) \geq 0\}$, $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuous and $g : \mathbb{R}^n \to \mathbb{R}^m$ is continuously differentiable. It is well known that, under a standard regularity assumption [16], a necessary condition for $\bar{x} \in \mathbb{R}^n$ to be a solution of the variational inequality problem is that $\bar{\lambda} \in \mathbb{R}^m$ exists such that $(\bar{x}, \bar{\lambda})$ solves system (2.8). Therefore, if we have a sequence $\{(x^k, \lambda^k)\}$ converging to a solution of system (2.8) which has an isolated primal part, we can apply the techniques described in this section in order to identify which of the constraints $g_i(x) \geq 0$ will be active at $\bar{x}$.

3. Defining identification functions.

From the previous section we see that the crucial point in the identification of active constraints is the definition of an identification function. In this section we show how it is possible to define such a function for problem $(\mathcal{P})$.

We consider three cases. In the first, we assume that the functions f and g are analytic. In the second, we require that the functions be LC^1 and that the MFCQ, as well as a second order sufficiency condition for optimality, be satisfied. Finally, in the third case, the functions are required to be C^2 and the KKT point is assumed to satisfy a regularity condition related to (but weaker than) Robinson's strong regularity [34] and which we call quasi-regularity.

Extensions of these results to the KKT system (2.8) are possible. We shall point out the relevant changes in corresponding remarks.

The cases considered here do not cover all the situations in which an identification function can be defined and computed, but they certainly show that the definition and computation of an identification function is possible in most of the cases of interest.

3.1. The analytic case.

Let f and each g_i $(i \in I)$ be *analytic* around a point x. We recall that this means that f and each g_i $(i \in I)$ possess derivatives of all orders and that they agree with their Taylor expansions around x. We say that f and each g_i $(i \in I)$ are analytic on an open set $X \subseteq \mathbb{R}^n$ if they are analytic around each $x \in X$. We shall make use of the following result from Lojasiewicz [27] and Luo and Pang [28].

THEOREM 3.1. *Let S denote the set of points in $\mathbb{R}^r$ satisfying*

$$s(y) \leq 0, \qquad h(y) = 0,$$

where $s : \mathbb{R}^r \to \mathbb{R}^p$ and $h : \mathbb{R}^r \to \mathbb{R}^t$ are analytic functions defined on an open set $X \subseteq \mathbb{R}^r$. Suppose that $S \neq \emptyset$. Then, for each compact subset $\Omega \subset X$, there exist constants $\tau > 0$ and $\gamma > 0$ such that

$$\text{dist}\,[y, S] \leq \tau(\|[s(y)]_+\| + \|h(y)\|)^\gamma \qquad \forall y \in \Omega. \tag{3.1}$$

Using this result, it is possible to define an identification function for problem $(\mathcal{P})$.

THEOREM 3.2. *Suppose that f and g are analytic in a neighborhood of a stationary point $\bar{x}$. Then, the function $\rho_1 : \mathbb{R}^n \times \mathbb{R}^m \to [0, \infty)$, defined by*

$$\rho_1(x, \lambda) = \begin{cases} 0 & \text{if } r(x, \lambda) = 0, \\ \frac{-1}{\log(r(x,\lambda))} & \text{if } r(x, \lambda) \in (0, 0.9), \\ \frac{-1}{\log(0.9)} & \text{if } r(x, \lambda) \geq 0.9, \end{cases}$$

where

$$r(x,\lambda) = \|\nabla f(x) - \nabla g(x)\lambda\| + |\lambda^T g(x)| + \|[-\lambda]_+\| + \|[-g(x)]_+\|, \tag{3.2}$$

is an identification function for $\mathcal{K}$.

Proof. It is obvious, by definition, that ρ_1 is a nonnegative function such that $\rho_1(\bar{x},\bar{\lambda}) = 0$ for every $(\bar{x},\bar{\lambda}) \in \mathcal{K}$. Furthermore,

$$\lim_{(x,\lambda)\to(\bar{x},\bar{\lambda})} \rho_1(x,\lambda) = 0 = \rho_1(\bar{x},\bar{\lambda}) \qquad \forall(\bar{x},\bar{\lambda}) \in \mathcal{K},$$

so that ρ_1 is also continuous on $\mathcal{K}$. Hence we only have to check the limit

$$\lim_{\substack{(x,\lambda)\to(\bar{x},\bar{\lambda})\\(x,\lambda)\notin\mathcal{K}}} \frac{\rho_1(x,\lambda)}{\operatorname{dist}[(x,\lambda),\mathcal{K}]} = +\infty. \tag{3.3}$$

To this end we recall that, for arbitrary $\tau > 0$ and $\gamma > 0$, the limit

$$\lim_{t\downarrow 0} \frac{-1}{\tau t^\gamma \log t} = +\infty \tag{3.4}$$

holds; see, e.g., [29, p. 328]. We can now apply Theorem 3.1 by considering the system (2.1) which defines KKT points. It is then easy to see that (3.1) yields, for every given compact set $\Omega \subset \mathbb{R}^{n+m}$ containing $(\bar{x},\bar{\lambda})$ in its interior,

$$\operatorname{dist}[(x,\lambda),\mathcal{K}] \le \tau r(x,\lambda)^\gamma \qquad \forall(x,\lambda) \in \Omega, \tag{3.5}$$

where τ and γ are fixed positive constants. Therefore, we can write

$$\lim_{\substack{(x,\lambda)\to(\bar{x},\bar{\lambda})\\(x,\lambda)\notin\mathcal{K}}} \frac{\rho_1(x,\lambda)}{\operatorname{dist}[(x,\lambda),\mathcal{K}]} \ge \lim_{\substack{(x,\lambda)\to(\bar{x},\bar{\lambda})\\(x,\lambda)\notin\mathcal{K}}} \frac{\rho_1(x,\lambda)}{\tau r(x,\lambda)^\gamma},$$

from which (3.3) follows taking into account (3.4), recalling the definition of ρ_1, and noting that $r(x,\lambda)$ is a continuous function that goes to 0 from the right as (x,λ) tends to $(\bar{x},\bar{\lambda})$. □

We stress that Theorem 3.2 holds under the mere assumption that f and g are analytic. In particular, the set of Lagrange multipliers Λ might be unbounded.

Remark 1. If we want to define an identification function for the solutions of the KKT system (2.8), we only have to substitute the definition of the residual (3.2) by the following:

$$r(x,\lambda) = \|F(x) - \nabla g(x)\lambda\| + |\lambda^T g(x)| + \|(-\lambda)_+\| + \|(-g(x))_+\|.$$

Obviously, also in this case, we have to assume that F and each g_i $(i \in I)$ are analytic in a neighborhood of the KKT point under consideration.

3.2. The second order condition case. In this subsection we assume that f and g are LC^1, i.e., that they are differentiable with Lipschitz continuous derivatives. We denote the *Lagrangian* of problem $(\mathcal{P})$ by

$$L(x,\lambda) := f(x) - \lambda^T g(x)$$

and write $\nabla_x L(x,\lambda)$ for the gradient of L with respect to the x variables. Furthermore, we will make use of the MFCQ and of the following second order sufficient condition for optimality.

Assumption 1. *There is $\gamma > 0$ such that, for all $\bar\lambda \in \Lambda$,*

$$h^T H h \geq \gamma \|h\|^2 \qquad \forall h \in W(\bar\lambda),\ \forall H \in \partial_x \nabla_x L(\bar x, \bar\lambda).$$

Here, $W(\bar\lambda)$ denotes the cone

$$\{h \in \mathbb{R}^n \,|\, h^T \nabla g_i(\bar x) \geq 0 \ \ (i \in I_0 : \bar\lambda_i = 0),\ h^T \nabla g_i(\bar x) = 0 \ \ (i \in I_0 : \bar\lambda_i > 0)\},$$

and $\partial_x \nabla_x L(\bar x, \bar\lambda)$ denotes Clarke's [5] generalized Jacobian *with respect to x of the gradient $\nabla_x L$, calculated at $(\bar x, \bar\lambda)$.*

We remark that, if the functions f and g are twice continuously differentiable and only one multiplier exists, then the previous definition reduces to the classical KKT second order sufficient condition for optimality. Moreover, note that requiring MFCQ implies that the KKT set $\mathcal{K}$ is compact.

Using the MFCQ, together with Assumption 1, we will show that the function $\rho_2 : \mathbb{R}^{n+m} \to [0, \infty)$ defined by

$$\rho_2(x, \lambda) := \sqrt{\|\Phi(x, \lambda)\|}$$

is an identification function for $\mathcal{K}$, where the operator $\Phi : \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$ is given by

$$\Phi(x, \lambda) := \begin{pmatrix} \nabla_x L(x, \lambda) \\ \min\{g(x), \lambda\} \end{pmatrix}. \tag{3.6}$$

Note that Φ is continuous on $\mathbb{R}^{n+m}$ and that $(x, \lambda) \in \mathcal{K}$ is equivalent to the nonlinear system

$$\Phi(x, \lambda) = 0.$$

To prove that ρ_2 is actually an identification function, let us first consider the perturbed nonlinear program

$$(\mathcal{P}(t)) \qquad \min \quad f(x, t) := f(x) + x^T t_f \quad \text{s.t.} \quad g(x, t) := g(x) + t_g \geq 0,$$

where $t = (t_f, t_g) \in \mathbb{R}^n \times \mathbb{R}^m$ denotes the perturbation parameter. In what follows we will assign to any vector $(y, \mu) \in \mathbb{R}^n \times \mathbb{R}^m$ a particular perturbation vector

$$\tau(y, \mu) = (\tau_f(y, \mu), \tau_g(y, \mu)) \in \mathbb{R}^n \times \mathbb{R}^m.$$

For this purpose we first define the function $\mu^\oplus : \mathbb{R}^{n+m} \to \mathbb{R}^m_+$ componentwise by

$$\mu_i^\oplus(y, \mu) := \begin{cases} \max\{0, \mu_i\} & \text{if } i \in I_\oplus(y, \mu), \\ 0 & \text{if } i \in I \setminus I_\oplus(y, \mu), \end{cases}$$

where, we recall, $I_\oplus(y, \mu) = \{i \in I \,|\, g_i(y) \leq \mu_i\}$. We can now introduce the function $\tau : \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$ by

$$\begin{aligned} \tau_f(y, \mu) \quad &:= \quad -\nabla_x L(y, \mu^\oplus(y, \mu)), \\ \tau_g(y, \mu)_i \quad &:= \quad \begin{cases} -g_i(y) & \text{if } i \in I_\oplus(y, \mu), \\ -\min\{0, g_i(y)\} & \text{if } i \in I \setminus I_\oplus(y, \mu), \end{cases} \qquad (i \in I). \end{aligned}$$

Using the particular perturbation vector $t = \tau(y, \mu)$, we can prove the following result.

LEMMA 3.3. *Let $(y,\mu) \in \mathbb{R}^n \times \mathbb{R}^m$ be arbitrarily chosen. Then, $(y, \mu^{\oplus}(y,\mu))$ is a KKT point for problem $(\mathcal{P}(t))$, where $t = \tau(y,\mu)$.*

Proof. The KKT system for the perturbed program $(\mathcal{P}(t))$ reads as follows:

$$\nabla_x L(x,\lambda) + t_f = 0, \tag{3.7}$$

$$\lambda \geq 0, \tag{3.8}$$

$$g(x) + t_g \geq 0, \tag{3.9}$$

$$\lambda^T (g(x) + t_g) = 0. \tag{3.10}$$

Let (y,μ) be arbitrary but fixed. Obviously, since $t = \tau(y,\mu)$, we find that $(x,\lambda) := (y, \mu^{\oplus}(y,\mu))$ solves (3.7) and (3.8). Now, we will show that $(y, \mu^{\oplus}(y,\mu))$ also satisfies (3.9) and (3.10). For $i \in I_{\oplus}(y,\mu)$, the definition of $\tau_g(y,\mu)$ yields $(g(y)+t_g)_i = 0$ so that both (3.9) and (3.10) are fulfilled. If, instead, $i \in I \setminus I_{\oplus}(y,\mu)$, it follows from the definition of $\mu^{\oplus}(y,\mu)$ that $\mu_i^{\oplus}(y,\mu) = 0$ and (3.10) is satisfied. Moreover, the definition of $\tau_g(y,\mu)$ implies

$$(g(y)+t_g)_i = g_i(y) - \min\{0, g_i(y)\} = \max\{0, g_i(y)\} \geq 0 \qquad \forall i \in I \setminus I_{\oplus}(y,\mu).$$

Thus, (3.9) is also valid for $i \in I \setminus I_{\oplus}(y,\mu)$. We therefore conclude that $(y, \mu^{\oplus}(y,\mu))$ is a KKT point of $(\mathcal{P}(t))$ when $t = \tau(y,\mu)$. □

The next lemma is a technical result which will be used in the proof of Theorem 3.6 which, in turn, is the basic ingredient used to establish the main result of this subsection, Theorem 3.7 below.

LEMMA 3.4. (a) *It holds that*

$$\|\mu - \mu^{\oplus}(y,\mu)\| \leq \|\min\{g(y),\mu\}\| \leq \|\Phi(y,\mu)\| \qquad \forall (y,\mu) \in \mathbb{R}^n \times \mathbb{R}^m.$$

(b) *If the MFCQ is satisfied, then $\kappa > 0$ exists such that*

$$\|\tau(y,\mu)\| \leq \kappa \|\Phi(y,\mu)\| \qquad \forall (y,\mu) \in \mathcal{K} + B_1.$$

Proof. Let us consider any $(y,\mu) \in \mathbb{R}^n \times \mathbb{R}^m$. We easily see that

$$\min\{g_i(y),\mu_i\} = \begin{cases} g_i(y) \quad \leq \quad \mu_i & \text{if } i \in I_{\oplus}(y,\mu), \\ \mu_i \quad \leq \quad g_i(y) & \text{if } i \in I \setminus I_{\oplus}(y,\mu). \end{cases}$$

Moreover, this and the definitions of the functions $\mu^{\oplus}$ and τ_g yield, for $i \in I_{\oplus}$,

$$\begin{array}{lclcl} |\mu_i - \mu_i^{\oplus}(y,\mu)| & = & |\min\{0,\mu_i\}| & \leq & |\min\{g_i(y),\mu_i\}|, \\ |\tau_g(y,\mu)_i| & = & |g_i(y)| & = & |\min\{g_i(y),\mu_i\}|. \end{array}$$

Similarly, for $i \in I \setminus I_{\oplus}(y,\mu)$, we get

$$\begin{array}{lclcl} |\mu_i - \mu_i^{\oplus}(y,\mu)| & = & |\mu_i| & = & |\min\{g_i(y),\mu_i\}|, \\ |\tau_g(y,\mu)_i| & = & |\min\{0,g_i(y)\}| & \leq & |\min\{g_i(y),\mu_i\}|. \end{array}$$

Thus, property (a) and

$$\|\tau_g(y,\mu)\| \leq \|\min\{g(y),\mu\}\| \leq \|\Phi(y,\mu)\| \qquad \forall (y,\mu) \in \mathbb{R}^n \times \mathbb{R}^m \tag{3.11}$$

follow. To prove (b) let $(y,\mu) \in \mathcal{K} + B_1$. Since, due to the MFCQ, $\mathcal{K} + B_1$ is bounded, the LC^1 property of f and g implies that the function $\|\nabla_x L\|$ is globally Lipschitz continuous on $\mathcal{K} + B_1$ with some modulus $\kappa_0 > 0$. Using property (a) and (3.11), we therefore obtain

$$\begin{array}{rcl} \|\tau(y,\mu)\| & \leq & \|\tau_f(y,\mu)\| + \|\tau_g(y,\mu)\| \\ & \leq & \|\nabla_x L(y,\mu)\| + \kappa_0\|\mu - \mu^{\oplus}(y,\mu)\| + \|\tau_g(y,\mu)\| \\ & \leq & \kappa\|\Phi(y,\mu)\| \end{array}$$

with $\kappa := \kappa_0 + 2$. □

The next result can easily be derived from Theorem 4.5b and formula 3.2f in Klatte [20]. If the functions f and g of the program $(\mathcal{P})$ are twice continuously differentiable it can also be obtained from a corresponding result in Robinson [35, Corollary 4.3]. We further note that Assumption 1 can be weakened by using generalized directional derivatives; see [20] for more details and references.

THEOREM 3.5. *Let the MFCQ and Assumption* 1 *be satisfied. Then*, *there are* $\delta > 0$, $\eta > 0$, *and* $c > 0$ *such that*

$$\mathit{dist}\,[(\bar{x}(t), \bar{\lambda}(t)), \mathcal{K}] \leq c\|t\|$$

for every $t \in B_\delta$ *and for every KKT point* $(\bar{x}(t), \bar{\lambda}(t))$ *of problem* $(\mathcal{P}(t))$ *for which* $\bar{x}(t) \in \{\bar{x}\} + B_\eta$.

Putting together the last three results, we can prove the following theorem.

THEOREM 3.6. *Let the MFCQ and Assumption* 1 *be satisfied. Then*, *there are* $\epsilon > 0$, $\kappa_1 > 0$, *and* $\kappa_2 > 0$ *such that*

$$\kappa_1 \mathrm{dist}\,[(y,\mu),\mathcal{K}] \leq \|\Phi(y,\mu)\| \leq \kappa_2 \mathrm{dist}\,[(y,\mu),\mathcal{K}] \qquad \forall (y,\mu) \in \mathcal{K} + B_\epsilon.$$

Proof. Let us consider any $(y,\mu) \in \mathbb{R}^n \times \mathbb{R}^m$ and let $z_1 \in \mathcal{K}$ and $z_2 \in \mathcal{K}$ be the projections of (y,μ) and $(y,\mu^{\oplus}(y,\mu))$, respectively, on the closed convex set $\mathcal{K}$. Then, using the triangle inequality, we get

$$\begin{array}{rcl} \mathrm{dist}[(y,\mu),\mathcal{K}] & = & \|z_1 - (y,\mu)\| \\ & \leq & \|z_2 - (y,\mu)\| \\ & \leq & \|z_2 - (y,\mu^{\oplus}(y,\mu))\| + \|(y,\mu) - (y,\mu^{\oplus}(y,\mu))\| \\ & = & \mathrm{dist}[(y,\mu^{\oplus}(y,\mu)),\mathcal{K}] + \|\mu - \mu^{\oplus}(y,\mu)\|. \end{array} \tag{3.12}$$

Now we will provide an upper bound for $\mathrm{dist}[(y,\mu^{\oplus}(y,\mu)),\mathcal{K}]$. Taking into account Lemma 3.4(b) and that $\|\Phi\|$ is a continuous function with $\Phi(y,\mu) = 0$ for all $(y,\mu) \in \mathcal{K}$, we have that, for δ from Theorem 3.5, we can find $\bar{\epsilon} > 0$ such that, if $(y,\mu) \in \mathcal{K}+B_{\bar{\epsilon}}$, then $\|\tau(y,\mu)\| \leq \kappa\|\Phi(y,\mu)\| \leq \delta$. Therefore, since $\epsilon \leq \eta$ (with η from Theorem 3.5) can be assumed without loss of generality, Theorem 3.5, together with Lemma 3.3, yields

$$\mathrm{dist}[(y,\mu^{\oplus}(y,\mu)),\mathcal{K}] \leq c\|\tau(y,\mu)\| \qquad \forall (y,\mu) \in \mathcal{K} + B_\epsilon.$$

Using this, (3.12), and Lemma 3.4, we obtain

$$\mathrm{dist}[(y,\mu),\mathcal{K}] \leq c\|\tau(y,\mu)\| + \|\mu - \mu^{\oplus}(y,\mu)\| \leq (c\kappa + 1)\|\Phi(y,\mu)\| \quad \forall (y,\mu) \in \mathcal{K} + B_\epsilon;$$

i.e., the left inequality in the theorem is satisfied with $\kappa_1 := 1/(c\kappa + 1)$. The right inequality can easily be obtained by taking into account that $\mathcal{K}$ is compact and convex and that $\|\Phi\|$ is locally Lipschitz continuous. Therefore, $\kappa_2 > 0$ exists such that

$$\|\Phi(y,\mu)\| = \|\Phi(y,\mu) - \Phi(z_1)\| \leq \kappa_2\|(y,\mu) - z_1\| = \kappa_2 \mathrm{dist}[(y,\mu),\mathcal{K}] \quad \forall (y,\mu) \in \mathcal{K} + B_\epsilon,$$

where (as above) z_1 denotes the projection of (y,μ) onto $\mathcal{K}$. □

THEOREM 3.7. *Let the MFCQ and Assumption 1 be satisfied. Then ρ_2 is an identification function for $\mathcal{K}$.*

Proof. Taking the properties of the operator Φ into account we easily see that ρ_2 is nonnegative and continuous on $\mathbb{R}^{n+m}$ and that $\rho_2(x,\lambda)=0$ for all $(x,\lambda)\in\mathcal{K}$, so that properties (a) and (b) of Definition 2.1 are satisfied. Finally, property (c) immediately follows from Theorem 3.6. □

If, instead of the upper Lipschitz continuity as stated in Theorem 3.5, the multifunction $t\mapsto\mathcal{K}(t)$ is upper Hölder continuous at $t=0$ with a known rate $\nu\in(0,1]$, that is, if for some $\delta>0$, $\eta>0$, and $c>0$,

$$\operatorname{dist}[(\bar{x}(t),\bar{\lambda}(t)),\mathcal{K}]\le c\|t\|^{\nu}$$

for every $t\in B_\delta$ and for every KKT point $(\bar{x}(t),\bar{\lambda}(t))$ of problem $(\mathcal{P}(t))$ for which $\bar{x}(t)\in\{\bar{x}\}+B_\eta$, then the technique presented in this subsection can easily be extended if we define $\rho_2:\mathbb{R}^n\times\mathbb{R}^m\to[0,\infty)$ as

$$\rho_2(x,\lambda):=\|\Phi(x,\lambda)\|^{\nu/2}.$$

In particular, Theorem 3.7 remains valid for this ρ_2 if Assumption 1 is replaced by the upper Hölder continuity.

An interesting case in which it is possible to prove, under an assumption weaker than Assumption 1, the upper Hölder continuity at $t=0$ of the multifunction $t\mapsto\mathcal{K}(t)$, is the case of convex problems. Assume that f is convex and each g_i $(i\in I)$ is concave, that the MFCQ holds, and that the following *growth condition* holds (in place of Assumption 1): positive $\bar{\eta}$ and $\bar{c}$ exist such that

$$f(x)\ge f(\bar{x})+\bar{c}\|x-\bar{x}\|^2\qquad\forall\text{ feasible }x\text{ in }\{\bar{x}\}+B_{\bar{\eta}}.$$

Under these assumptions and using the results in [21], it is possible to show (we omit the details) that $\delta>0$, $\eta>0$, and $c>0$ exist such that

$$\operatorname{dist}[(\bar{x}(t),\bar{\lambda}(t)),\mathcal{K}]\le c\sqrt{\|t\|}$$

for every $t\in B_\delta$ and for every KKT point $(\bar{x}(t),\bar{\lambda}(t))$ of problem $(\mathcal{P}(t))$ for which $\bar{x}(t)\in\{\bar{x}\}+B_\eta$. It may be interesting to note that the growth condition holds, in particular, if Assumption 1 is fulfilled.

Remark 2. The extension of the results of this section to general KKT systems is not straightforward, since the sensitivity analysis of perturbed KKT systems requires, to date, stronger assumptions. The key point is to establish a result analogous to Theorem 3.5. Once this has been done, we can easily prove theorems analogous to Theorem 3.7 by substituting F for ∇f in every relevant formula. As an example of the kind of results that can be obtained, we cite the following. Suppose that F is C^1 and g is C^2. Assume also that the SMFCQ holds at $\bar{x}$ along with Assumption 1. Then, according to [15, Corollary 8(c)], Theorem 3.5 holds and therefore, ρ_2 is a regular identification function for the KKT system (2.8).

3.3. The quasi-regular case. In this subsection we assume that the functions f and g are C^2. We shall introduce a condition which we call *quasi-regularity*. As will be clear later, this quasi-regularity is related to, but weaker than, Robinson's *strong regularity* [34]. In order to motivate the definition of a quasi-regular KKT point we will first recall a condition which is equivalent to the notion of a strongly regular KKT point. To this end we shall use the index set $I_{00}:=I_0\setminus I_+$ of all those indices for which the strict complementarity condition does not hold at the KKT point $(\bar{x}, \bar{\lambda})$. For any $J \subseteq I_{00}$ (empty set included), introduce the matrix

$$M(J) := \begin{pmatrix} \nabla^2_{xx} L & \nabla g_+ & \nabla g_J \\ -\nabla g_+^T & 0 & 0 \\ -\nabla g_J^T & 0 & 0 \end{pmatrix},$$

where $\nabla^2_{xx} L$, ∇g_+, and ∇g_J are abbreviations for the matrices $\nabla^2_{xx} L(\bar{x}, \bar{\lambda})$, $\nabla g_{I_+}(\bar{x})$, and $\nabla g_J(\bar{x})$, respectively. The following result is from Kojima [23].

THEOREM 3.8. *The following statements are equivalent*:

(a) $(\bar{x}, \bar{\lambda})$ *is a strongly regular KKT point.*

(b) *For any* $J \subseteq I_{00}$ (*empty set included*), *the determinants of the matrices* $M(J)$ *all have the same nonzero sign.*

Motivated by point (b) in Theorem 3.8, we introduce the following definition.

DEFINITION 3.9. *The KKT point* $(\bar{x}, \bar{\lambda})$ *is a* quasi-regular *point if the matrices* $M(J)$ *are nonsingular for every* $J \subseteq I_{00}$ (*empty set included*).

Note that, in view of Theorem 3.8, quasi-regularity is implied by Robinson's strong regularity condition, but the converse is not true. In fact, consider the following example:

$$\begin{aligned} &\min x_1^2 + x_2^2 + 4x_1x_2 \\ &\text{s.t. } x_1 \geq 0, \\ &\qquad x_2 \geq 0. \end{aligned}$$

It is easy to check that $\bar{x} = (0, 0)$ is a global minimizer and that the Lagrange multipliers of the two constraints are both 0, so that $I_0 = I_{00} = \{1, 2\}$, while $I_+ = \emptyset$. Furthermore, $detM(\emptyset) < 0$, while, for $J \in \{\{1\}, \{2\}, \{1, 2\}\}$, $detM(J) > 0$. Therefore, $(\bar{x}, 0, 0)$ is a quasi-regular KKT point but not a strongly regular one. Note that in this example the KKT point is an isolated KKT point. This is not a coincidence. In fact, we shall show in this section that quasi-regularity of a KKT point implies its local uniqueness. It is also worth pointing out that quasi-regularity implies the linear independence of the active constraints. This easily follows from the fact that $M(I_{00})$ is nonsingular.

As in subsection 3.2 we make use of the operator $\Phi : \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$ defined in (3.6) which, due to the differentiability assumptions, is locally Lipschitz continuous. Hence by Rademacher's theorem, Φ is differentiable almost everywhere. Denote by D_Φ the set of points where Φ is differentiable. Then we can define the *B-subdifferential* (see, e.g., [32]) of Φ at (x, λ) as

$$\begin{aligned} \partial_B \Phi(x, \lambda) := \{H \in \mathbb{R}^{(n+m)\times(n+m)} \,|\, \exists\{(x^k, \lambda^k)\} \subset D_\Phi : \\ (x^k, \lambda^k) \to (x, \lambda),\ \nabla\Phi(x^k, \lambda^k)^T \to H\}. \end{aligned}$$

Note that the B-subdifferential is a subset of Clarke's generalized Jacobian [5, 32]. The next lemma illustrates the structure of the B-subdifferential of Φ. Before stating this lemma, however, we introduce three index sets:

$$\begin{aligned} \alpha(x, \lambda) &:= \{i \in I \,|\, g_i(x) < \lambda_i\}, \\ \beta(x, \lambda) &:= \{i \in I \,|\, g_i(x) = \lambda_i\}, \\ \gamma(x, \lambda) &:= \{i \in I \,|\, g_i(x) > \lambda_i\}. \end{aligned}$$

LEMMA 3.10. *Let $(x,\lambda) \in \mathbb{R}^{n+m}$ be arbitrary. Then*

$$\partial_B \Phi(x,\lambda)^T \subseteq \begin{pmatrix} \nabla^2_{xx} L(x,\lambda) & \nabla g(x) D_a(x,\lambda) \\ -\nabla g(x)^T & D_b(x,\lambda) \end{pmatrix},$$

where

$$D_a(x,\lambda) := diag\,(a_1(x,\lambda),\ldots,a_m(x,\lambda)),$$
$$D_b(x,\lambda) := diag\,(b_1(x,\lambda),\ldots,b_m(x,\lambda))$$

are diagonal matrices with

$$a_i(x,\lambda) = \begin{cases} 1 & \text{if } i \in \alpha(x,\lambda), \\ 0 \text{ or } 1 & \text{if } i \in \beta(x,\lambda), \\ 0 & \text{if } i \in \gamma(x,\lambda), \end{cases}$$

and $D_b(x,\lambda) = I - D_a(x,\lambda)$.

Proof. This follows immediately from the definition of the operator Φ. □

We are now in the position to prove the following result.

LEMMA 3.11. *Let $(\bar{x},\bar{\lambda}) \in \mathbb{R}^{n+m}$ be a quasi-regular KKT point. Then all matrices $H \in \partial_B \Phi(\bar{x},\bar{\lambda})$ are nonsingular.*

Proof. Let $H \in \partial_B \Phi(\bar{x},\bar{\lambda})^T$. In view of Lemma 3.10, there exists an index set $J \subseteq \beta(\bar{x},\bar{\lambda})$ such that

$$H = \begin{pmatrix} \nabla^2_{xx} L & \nabla g_\alpha & \nabla g_J & 0 & 0 \\ -\nabla g_\alpha^T & 0 & 0 & 0 & 0 \\ -\nabla g_J^T & 0 & 0 & 0 & 0 \\ -\nabla g_{\bar{J}}^T & 0 & 0 & I_{\bar{J}} & 0 \\ -\nabla g_\gamma^T & 0 & 0 & 0 & I_\gamma \end{pmatrix},$$

where $\bar{J} = \beta(\bar{x},\bar{\lambda}) \setminus J$ denotes the complement of J in the set $\beta(\bar{x},\bar{\lambda})$. Obviously, this matrix is nonsingular if and only if the matrix

$$\begin{pmatrix} \nabla^2_{xx} L & \nabla g_\alpha & \nabla g_J \\ -\nabla g_\alpha^T & 0 & 0 \\ -\nabla g_J^T & 0 & 0 \end{pmatrix}$$

is nonsingular. In turn, this matrix is nonsingular if and only if the matrix $M(J)$ is nonsingular. Hence, the thesis follows immediately from Definition 3.9. □

We are now able to prove the main result of this subsection. For this purpose recall that $\rho_2(x,\lambda) = \sqrt{\|\Phi(x,\lambda)\|}$ (see subsection 3.2).

THEOREM 3.12. *Let $(\bar{x},\bar{\lambda}) \in \mathbb{R}^{n+m}$ be a quasi-regular KKT point of problem $(\mathcal{P})$. Then*

(a) *$(\bar{x},\bar{\lambda})$ is an isolated KKT point,*

(b) *the function ρ_2 is an identification function for $\mathcal{K} = \{(\bar{x},\bar{\lambda})\}$.*

Proof. As already shown in the proof of Theorem 3.7, the function ρ_2 has the properties (a) and (b) of Definition 2.1. Furthermore, since f and g have locally Lipschitz continuous gradients and the min operator is semismooth (see [30, 33] for the definition of semismoothness and [30] for the proof that the min operator is semismooth) it follows that Φ, which is the composite of semismooth functions, is also semismooth [30, 33]. Hence it follows from Lemma 3.11 and [31, Proposition 3] that there exists a constant $c > 0$ such that

$$\|\Phi(x,\lambda)\| \geq c\|(x,\lambda) - (\bar{x},\bar{\lambda})\| = c\,\mathrm{dist}[(x,\lambda),\mathcal{K}] \tag{3.13}$$

for all (x, λ) in a neighborhood of $(\bar{x}, \bar{\lambda})$. Therefore, one can easily see that ρ_2 also has property (c) of Definition 2.1, i.e., ρ_2 is an identification function for $\mathcal{K}$. Finally, since $\Phi(x, \lambda) = 0$ if and only if (x, λ) is a KKT point, part (a) of the theorem follows from (3.13). □

Remark 3. In the case of the KKT system (2.8) everything goes through. It is sufficient to assume that F is continuously differentiable and to substitute everywhere the gradient $\nabla_x L(x, \lambda)$ by the function $F(x) - \nabla g(x)\lambda$. Also in this case the definition of quasi-regularity is related to and weaker than that of a strongly regular KKT point since Theorem 3.8 carries over to the KKT system (2.8); see Liu [26, Lemma 3.4]. Actually, the case of KKT systems of variational inequalities is probably the main case in which quasi-regularity can be applied. In fact, it is not difficult to see that, if strict complementarity holds and $\bar{x}$ is a local minimum point of problem $(\mathcal{P})$, quasi-regularity implies the conditions of the previous subsection. However, these conditions and quasi-regularity are fairly distinct if one considers variational inequalities. For example, it can easily be checked that, given the variational inequality defined by the function $F(x) = (x_1 + x_2^2, -x_2)^T$ and the set $X = \{x \in \mathbb{R}^2 |\, x_2 \geq 0\}$, the point $(0, 0)^T$ is a quasi-regular solution but does not satisfy the conditions stated in Remark 2 of the previous subsection.

4. Numerical examples. In this section we illustrate the identification technique on three nonlinear optimization problems. Our aim here is merely to give the reader a feel for the potentialities of the new technique. A detailed study of its numerical behavior is beyond the scope of this paper.

We consider three test problems from the Hock and Schittkowski collection [17]. The first is problem 113, and at the solution both the linear independence constraint qualification and the strict complementarity condition are satisfied. The second problem is a modification of problem 46 and, while the linear independence constraint qualification is satisfied at the solution, the multipliers are all 0. Finally, we consider a modification of problem 43 whose multiplier set Λ is not a singleton.

For these test problems we applied the identification technique for both identifications functions ρ_1 and ρ_2 introduced in section 3. To this end, random points (x, λ) at different fixed distances from the set $\mathcal{K}$ were generated. More precisely, for each $\varepsilon \in \{10, 1, 10^{-1}, 10^{-2}, 10^{-3}\}$, we generated 100 random vectors (x, λ) on the boundary of the set

$$\mathcal{K} + B_\varepsilon^\infty = \{(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m |\, \exists \bar{\lambda} \in \Lambda \,:\, \|(x, \lambda) - (\bar{x}, \bar{\lambda})\|_\infty < \varepsilon\}.$$

For each of these random vectors we compared our approximate active sets $A(x, \lambda)$ with the exact active set I_0. For each of the constraints, for the different values of ε and for both identification functions ρ_1 and ρ_2, we report the sum of the correctly identified constraints over all 100 randomly generated vectors (x, λ); see the tables below. The last column of each table contains the total number of correctly identified constraints over all constraints.

Example 1. This is problem 113 from [17]. It is a convex optimization problem with $n = 10$ variables and $m = 8$ inequality constraints, five of them being nonlinear. The solution is given by

$$\bar{x} \approx (2.17, 2.36, 8.77, 5.10, 0.99, 1.43, 1.32, 9.83, 8.28, 8.38)^T,$$

and the corresponding optimal Lagrange multiplier is unique and given by

$$\bar{\lambda} \approx (1.72, 0.48, 1.38, 0.02, 0.31, 0, 0.29, 0)^T.$$

TABLE 4.1
Numerical results for Example 1.

ε	ρ	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_1 - g_8
$\varepsilon = 10$	ρ_1	54	54	57	89	90	2	78	16	440
	ρ_2	65	60	71	90	93	0	84	12	475
$\varepsilon = 1$	ρ_1	90	76	94	68	75	22	83	100	608
	ρ_2	81	68	86	64	67	36	75	100	577
$\varepsilon = 0.1$	ρ_1	100	100	100	100	100	0	100	100	700
	ρ_2	100	94	100	76	90	100	99	100	759
$\varepsilon = 0.01$	ρ_1	100	100	100	100	100	82	100	100	782
	ρ_2	100	100	100	100	100	100	100	100	800
$\varepsilon = 0.001$	ρ_1	100	100	100	100	100	100	100	100	800
	ρ_2	100	100	100	100	100	100	100	100	800

The solution satisfies the strict complementarity condition; however, since the fourth constraint is active and $\bar{\lambda}_4 \approx 0.02$, the solution is relatively close to being degenerate. Our results are summarized in Table 4.1.

Example 2. This example is a modification of problem 46 from [17]. Problem 46 has two equality constraints which have 0 multipliers at the solution. We converted the equalities to inequalities and added the constraint $x_2 \leq 1$ in order to maintain the uniqueness of the solution considered. Thus, we have $n = 5$ variables and $m = 3$ inequality constraints. The objective function is given by

$$f(x) := (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6,$$

and the constraints are

$$\begin{aligned} g_1(x) &:= x_1^2 x_4 + \sin(x_4 - x_5) - 1 \geq 0, \\ g_2(x) &:= x_2 + x_3^4 x_4^2 - 2 \geq 0, \\ g_3(x) &:= 1 - x_2 \geq 0. \end{aligned}$$

The solution is

$$\bar{x} := (1, 1, 1, 1, 1)^T$$

and the corresponding multiplier is

$$\bar{\lambda} := (0, 0, 0)^T.$$

Since all inequality constraints are active at the solution $\bar{x}$, $(\bar{x}, \bar{\lambda})$ is totally degenerate. We report our results in Table 4.2.

Example 3. This example is a modification of problem 43 from [17]. It has $n = 4$ variables and $m = 4$ inequality constraints. Its objective function is

$$f(x) := x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4,$$

and its constraints are

$$\begin{aligned} g_1(x) &:= -x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8 \geq 0, \\ g_2(x) &:= -x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10 \geq 0, \\ g_3(x) &:= -2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5 \geq 0, \\ g_4(x) &:= x_2^3 + 2x_1^2 + x_4^2 + x_1 - 3x_2 - x_3 + 4x_4 + 7 \geq 0; \end{aligned}$$

Table 4.2
Numerical results for Example 2.

ε	ρ	g_1	g_2	g_3	g_1 - g_3
$\varepsilon = 10$	ρ_1	52	8	92	152
	ρ_2	85	18	100	203
$\varepsilon = 1$	ρ_1	100	82	100	282
	ρ_2	88	73	100	261
$\varepsilon = 0.1$	ρ_1	100	99	100	299
	ρ_2	100	97	100	297
$\varepsilon = 0.01$	ρ_1	100	100	100	300
	ρ_2	100	100	100	300
$\varepsilon = 0.001$	ρ_1	100	100	100	300
	ρ_2	100	100	100	300

Table 4.3
Numerical results for Example 3.

ε	ρ	g_1	g_2	g_3	g_4	g_1 - g_4
$\varepsilon = 10$	ρ_1	100	0	100	26	226
	ρ_2	100	0	100	27	227
$\varepsilon = 1$	ρ_1	100	0	100	100	300
	ρ_2	89	18	96	65	268
$\varepsilon = 0.1$	ρ_1	100	0	100	100	300
	ρ_2	100	36	100	100	336
$\varepsilon = 0.01$	ρ_1	100	97	100	100	397
	ρ_2	100	100	100	100	400
$\varepsilon = 0.001$	ρ_1	100	100	100	100	400
	ρ_2	100	100	100	100	400

i.e., we added the fourth constraint to problem 43 from [17]. The solution of this problem is

$$\bar{x} = (0, 1, 2, -1)^T.$$

The constraints g_1, g_3, and g_4 are active at the solution, and

$$\nabla g_4(\bar{x}) = \nabla g_1(\bar{x}) - \nabla g_3(\bar{x})$$

so that the linear independence constraint qualification is violated. However, the corresponding set of Lagrange multipliers, given by

$$\Lambda := \{\bar{\lambda}(r) := (3 - r, 0, r, r - 2)^T \mid r \in [2, 3]\},$$

is bounded, so that the Mangasarian–Fromovitz constraint qualification is satisfied. Furthermore, if $r \in \{2, 3\}$, then strict complementarity is violated.

To test this problem, the random points (x, λ) on the boundary of $\mathcal{K} + B_\varepsilon^\infty$ were generated as follows. First, the x-part was randomly generated such that $\|x - \bar{x}\|_\infty = \varepsilon$. To obtain the λ-part we took a random number $r \in [2, 3]$ and then generated the vector λ randomly such that $\|\lambda - \bar{\lambda}(r)\|_\infty = \varepsilon$. It is obvious that every point (x, λ) generated in this way lies on the boundary of $\mathcal{K} + B_\varepsilon^\infty$. In Table 4.3 we summarize the results obtained for this example.

We think these three examples suggest that the identification technique is viable in practice even if we are well aware that no firm conclusion can be drawn on the basis of these few tests.

It is also important to point out that if ρ is an identification function, then any positive multiple of ρ is an identification function; in practice an appropriate scaling of the identification functions might be crucial for a good performance of the identification technique. Finally we note that if one wants to employ the identification technique in combination with a specific solution algorithm, one should take into account that sequences generated by specific algorithms may have additional properties which should be exploited to enhance the identification process.

5. Final remarks. In this paper we introduced a technique to accurately identify active constraints in inequality constrained optimization and variational inequality problems. The most remarkable features of the new identification technique are, on the one hand, that it identifies all active constraints even if strict complementarity does not hold and, on the other hand, that, as far as we are aware, it is the first identification technique applicable to nonlinear variational inequalities. Furthermore, as discussed in the introduction, it also enjoys several other favorable characteristics. In particular, the identification technique can be used in combination with any algorithm for the solution of inequality constrained optimization or variational inequality problems.

We believe that the techniques introduced in this paper can be useful in many cases, especially in the theoretical analysis and design of optimization methods.

From a practical point of view, the following questions may be of interest:

(a) How large is the region where exact identification occurs?

(b) Can we build identification functions which are scale invariant?

(c) Can we relax the assumption that $\bar{x}$ is an isolated stationary point and still obtain useful results?

It is difficult to answer these questions at the level of generality adopted in this paper. We think that answers can come from practical experiments and from an analysis of structured classes of problems, e.g., linear or quadratic problems, box or linearly constrained problems, etc.

From a more theoretical point of view, we would like to mention that the identification technique introduced in this paper turned out to be an essential tool in the development of the first algorithm for nonlinearly inequality constrained problems for which convergence to points satisfying the second order necessary condition for optimality can be established; see [11]. Moreover, the identification technique is one basic ingredient for the algorithm suggested in [19] which is the first QP-free method for the solution of variational inequality problems which has a global and superlinear convergence and which generates (in some sense) only feasible iterates. Finally, let us mention that the new identification technique has been advocated in [39] to accommodate a theoretical assumption needed to establish the superlinear convergence of an SQP-type method even when the linear independence of the active constraints is not satisfied at a solution.

Acknowledgment. We would like to thank Professor D. Klatte for helpful discussions on the stability of KKT systems.

REFERENCES

[1] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods,* Academic Press, New York, 1982.

[2] J.V. Burke, *On the identification of active constraints* II: *The nonconvex case,* SIAM J. Numer. Anal., 27 (1990), pp. 1081–1102.

[3] J.V. Burke and J.J. Moré, *On the identification of active constraints,* SIAM J. Numer. Anal., 25 (1988), pp. 1197–1211.

[4] J.V. Burke and J.J. Moré, *Exposing constraints,* SIAM J. Optim., 4 (1994), pp. 573–595.

[5] F.H. Clarke, *Optimization and Nonsmooth Analysis,* John Wiley, New York, 1983 (reprinted by SIAM, Philadelphia, PA, 1990).

[6] A.R. Conn, N.I.M. Gould, and P.L. Toint, *Global convergence for a class of trust region algorithms for optimization problems with simple bounds,* SIAM J. Numer. Anal., 25 (1988), pp. 433–460.

[7] A.S. El-Bakry, R.A. Tapia, and Y. Zhang, *A study of indicators for identifying zero variables in interior-point methods,* SIAM Review, 36 (1994), pp. 45–72.

[8] A.S. El-Bakry, R.A. Tapia, and Y. Zhang, *On the convergence rate of Newton interior-point methods in the absence of strict complementarity,* Comput. Optim. Appl., 6 (1996), pp. 157–167.

[9] F. Facchinei and S. Lucidi, *A Class of Methods for Optimization Problems with Simple Bounds,* Technical Report, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza," Rome, Italy, 1992 (revised 1995).

[10] F. Facchinei and S. Lucidi, *Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems,* J. Optim. Theory Appl., 85 (1995), pp. 265–289.

[11] F. Facchinei and S. Lucidi, *Convergence to second order stationary points in inequality constrained optimization,* DIS Working Paper 32-96, Università di Roma "La Sapienza," Roma, Italy, 1996; Math. Oper. Res., to appear.

[12] R. Fletcher, *Practical Methods of Optimization,* John Wiley, New York, 1987.

[13] J. Gauvin, *A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming,* Math. Programming, 12 (1977), pp. 136–138.

[14] P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization,* Academic Press, London, 1981.

[15] M.S. Gowda and J.-S. Pang, *Stability analysis of variational inequalities and nonlinear complementarity problems, via the mixed linear complementarity problem and degree theory,* Math. Oper. Res., 19 (1994), pp. 831–879.

[16] P.T. Harker and J.-S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications,* Math. Programming, 48 (1990), pp. 161–220.

[17] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes,* Lecture Notes in Economics and Math. Systems 187, Springer-Verlag, Berlin, 1981.

[18] J. Ji and F.A. Potra, *Tapia indicators and finite termination of infeasible-interior-point methods for degenerate LCP,* in The Mathematics of Numerical Analysis, Lectures in Appl. Math. 32, Amer. Math. Soc., Providence, RI, 1996, pp. 443–454.

[19] C. Kanzow and H.-D. Qi, *A QP-free constrained Newton-type method for variational inequality problems.* Math. Programming, to appear.

[20] D. Klatte, *Nonlinear optimization problems under data perturbations,* in W. Krabs and J. Zowe, eds.; Modern Methods of Optimization, Springer-Verlag, Berlin, 1992, pp. 204–235.

[21] D. Klatte, *On quantitative stability for $C^{1,1}$ programs,* in R. Durier and C. Michelot, eds., Recent Developments in Optimization, Springer-Verlag, Berlin, 1995, pp. 215–230.

[22] H. Kleinmichel, C. Richter, and K. Schönefeld, *On a class of hybrid methods for smooth constrained optimization,* J. Optim. Theory Appl., 73 (1992), pp. 465–499.

[23] M. Kojima, *Strongly stable stationary solutions in nonlinear programs,* in S.M. Robinson, ed., Analysis and Computation of Fixed Points, Academic Press, New York, 1979, pp. 93–138.

[24] J. Kyparisis, *On uniqueness of Kuhn Tucker multipliers in nonlinear programming,* Math. Programming, 32 (1985), pp. 242–246.

[25] M. Lescrenier, *Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold,* SIAM J. Numer. Anal., 28 (1991), pp. 476–495.

[26] J. Liu, *Strong stability in variational inequalities,* SIAM J. Control Optim., 33 (1995), pp. 725–749.

[27] M.S. Lojasiewicz, *Sur le problème de la division,* Stud. Math., 18 (1959), pp. 87–136.

[28] Z.-Q. Luo and J.-S. Pang, *Error bounds for analytic systems and their applications,* Math. Programming, 67 (1994), pp. 1–28.

[29] J. Marsden and A. Weinstein, *Calculus* I, Springer-Verlag, New York, 1985.

[30] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization,* SIAM J. Control Optim., 15 (1977), pp. 957–972.

[31] J.-S. Pang and L. Qi, *Nonsmooth equations: Motivation and algorithms,* SIAM J. Optim., 3 (1993), pp. 443–465.

[32] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations,* Math. Oper. Res., 18 (1993), pp. 227–244.

[33] L. Qi and J. Sun, *A nonsmooth version of Newton's method,* Math. Programming, 58 (1993), pp. 353–368.

[34] S.M. Robinson, *Strongly regular generalized equations,* Math. Oper. Res., 5 (1980), pp. 43–62.

[35] S.M. Robinson, *Generalized equations and their solution, part* II: *Applications to nonlinear programming,* Math. Programming Study, 19 (1982), pp. 200–221.

[36] K. Schönefeld, *Hybrid optimization methods without strict complementary slackness conditions,* in Proceedings of the International Conference on Mathematical Optimization—Theory and Applications, Eisenach, Germany, Technische Hochschule Ilmenau, Ilmenau, Germany, 1986, pp. 137–140.

[37] S.J. Wright, *Convergence of SQP like methods for constrained optimization,* SIAM J. Control Optim., 27 (1989), pp. 13–26.

[38] S.J. Wright, *Identifiable surfaces in constrained optimization,* SIAM J. Control Optim., 31 (1993), pp. 1063–1079.

[39] S.J. Wright, *Superlinear convergence of a stabilized SQP method to a degenerate solution,* Comput. Optim. Appl., to appear.

SIAM J. OPTIM.
Vol. 9, No. 1, pp. 33–52

ROBUST SOLUTIONS TO UNCERTAIN SEMIDEFINITE PROGRAMS[*]

LAURENT EL GHAOUI[†], FRANCOIS OUSTRY[†], AND HERVÉ LEBRET[†]

Abstract. In this paper we consider semidefinite programs (SDPs) whose data depend on some unknown but bounded perturbation parameters. We seek "robust" solutions to such programs, that is, solutions which minimize the (worst-case) objective while satisfying the constraints for every possible value of parameters within the given bounds. Assuming the data matrices are rational functions of the perturbation parameters, we show how to formulate sufficient conditions for a robust solution to exist as SDPs. When the perturbation is "full," our conditions are necessary and sufficient. In this case, we provide sufficient conditions which guarantee that the robust solution is unique and continuous (Hölder-stable) with respect to the unperturbed problem's data. The approach can thus be used to regularize ill-conditioned SDPs. We illustrate our results with examples taken from linear programming, maximum norm minimization, polynomial interpolation, and integer programming.

Key words. convex optimization, semidefinite programming, uncertainty, robustness, regularization

AMS subject classifications. 93B35, 49M45, 90C31, 93B60

PII. S1052623496305717

Notation. For a matrix X, $\|X\|$ denotes the largest singular value. If X is square, $X \succeq 0$ (resp., $X \succ 0$) means X is symmetric and positive semidefinite (resp., definite). For $X \succeq 0$, $X^{1/2}$ denotes the symmetric square root of X. The notation I_p denotes the $p \times p$ identity matrix; the subscript is omitted when it can be inferred from context.

1. Introduction.

A semidefinite program (SDP) consists of minimizing a linear objective under a linear matrix inequality (LMI) constraint; precisely,

$$(1) \qquad \mathcal{P}_0: \qquad \text{minimize } c^T x \text{ subject to } F(x) = F_0 + \sum_{i=1}^{m} x_i F_i \succeq 0,$$

where $c \in \mathbf{R}^m - \{0\}$ and the symmetric matrices $F_i = F_i^T \in \mathbf{R}^{n \times n}, i = 0, \ldots, m$, are given. SDPs are convex optimization problems and can be solved in polynomial time with, e.g., primal-dual interior-point methods [24, 35, 26, 19, 2]. SDPs include linear programs and convex quadratically constrained quadratic programs, and arise in a wide range of engineering applications; see, e.g., [12, 1, 35, 22].

In the SDP (1), the "data" consist of the objective vector c and the matrices $F_0, \ldots, F_m$. In practice, these data are subject to uncertainty. An extensive body of work has concentrated on the sensitivity issue, in which the perturbations are assumed to be infinitesimal, and regularity of optimal values and solution(s), as functions of the data matrices, is studied. Recent references on sensitivity analysis include [30, 31, 10] for general nonlinear programs, [33] for semi-infinite programs, and [32] for semidefinite programs.

When the perturbation affecting the data of the problem is not necessarily small, a sensitivity analysis is not sufficient. For general optimization problems, a whole field

[*] Received by the editors June 21, 1996; accepted for publication (in revised form) September 22, 1997; published electronically October 30, 1998.

http://www.siam.org/journals/siopt/9-1/30571.html

[†] Ecole Nationale Supérieure de Techniques Avancées, 32, Bd. Victor, 75739 Paris, France (elghaoui@ensta.fr, oustry@ensta.fr, lebret@ensta.fr).

of study (stochastic programming) concentrates on the case where the perturbation is stochastic with known statistics. One object of this field is to study the impact of, say, a random objective on the distribution of optimal values (this problem is called the "distribution problem"). References relevant to this approach to the perturbation problem include [15, 9, 29]. We are not aware of special references for general SDPs with randomly perturbed data except for the last section of [30], some exercises in the course notes of [13], and section 2.6 in [23].

The main objective of this paper is to quantify the effect of unknown but bounded deterministic perturbation of problem data on solutions. In our framework, the perturbation is not necessarily small, and we seek a solution that is "robust," that is, remains feasible despite the allowable, not necessarily small, perturbation. Our aim is to obtain (approximate) robust solutions via SDP. Links between regularity of solutions and robustness are, of course, expected. One of our side objectives is to clarify these links to some extent. This paper extends results given in [16] for the least-squares problem.

The approach developed here can be viewed as a special case of stochastic programming in which the distribution of the perturbation is uniform.

The ideas developed in this paper draw mainly from two sources: control theory, in which we have found the tools for robustness analysis [36, 17, 12] and some recent work on sensitivity analysis of optimization problems by Shapiro [31] and Bonnans, Cominetti, and Shapiro [10].

Shortly after completion of our manuscript, we became aware of the ongoing work of Ben-Tal and Nemirovski on the same subject. In [7], they apply similar ideas to a truss topology design problem and derive very efficient algorithms for solving the corresponding robustness problem. In [8], the general problem of *tractability* of obtaining a robust solution is studied, and "tractable counterparts" of a large class of uncertain SDPs are given. The case of robust linear programming, under quite general assumptions on the perturbation bounds, is studied in detail in [6]. Our paper can be seen as a complement of [8], giving ways to cope with (not necessarily) tractable robust SDPs by means of upper bounds. (In particular, our paper handles the case of nonlinear dependence of the data on the uncertainties.) A unified treatment, and more results, will appear in [4].

The paper is divided as follows. Our problem is defined in section 2. In section 3, we show how to compute upper bounds on our problem via SDP. We give special attention to the so-called full perturbations case, for which our results are nonconservative. In section 4, we examine sensitivity of the robust solutions in the full perturbations case. We provide conditions which guarantee that the robust solution is unique and a regular function of the data matrices. We then consider several interesting examples in section 5, such as robust linear programming, robust norm minimization, and error-in-variables SDPs.

2. Problem definition.

2.1. Robust SDPs. Let $\mathbf{F}(x, \Delta)$ be a symmetric matrix-valued function of two variables $x \in \mathbf{R}^m$, $\Delta \in \mathbf{R}^{p \times q}$. In the following, we consider x to be the decision variable, and we think of Δ as a perturbation. We assume that Δ is unknown but bounded. Precisely, we assume that Δ is known to belong to a given linear subspace $\mathcal{D}$ of $\mathbf{R}^{p \times q}$, and in addition, $\|\Delta\| \leq \rho$, where $\rho \geq 0$ is given.

In section 2.2, we will be more precise about the dependence of $\mathbf{F}$ on Δ.

We define the *robust feasible set* by

$$\mathcal{X}_\rho = \left\{ x \in \mathbf{R}^m \;\middle|\; \begin{array}{c} \text{for every } \Delta \in \mathcal{D},\ \|\Delta\| \le \rho, \\ \mathbf{F}(x,\Delta) \text{ is well defined and } \mathbf{F}(x,\Delta) \succeq 0 \end{array} \right\}. \tag{2}$$

Now let $\mathbf{c}(\Delta)$ be a vector-valued rational function of the perturbation Δ, such that $\mathbf{c}(0) = c$. We consider the following min-max problem:

$$\text{minimize } \max_{\Delta \in \mathcal{D},\ \|\Delta\| \le \rho} \mathbf{c}(\Delta)^T x \text{ subject to } x \in \mathcal{X}_\rho. \tag{3}$$

From now on, we assume that the function $\mathbf{c}(\Delta)$ is independent of Δ (in other words, the objective vector c is not subject to perturbation). This is done with no loss of generality: introduce a slack variable λ and define

$$\tilde{x} = \begin{bmatrix} x \\ \lambda \end{bmatrix}, \quad \tilde{c} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \tilde{\mathbf{F}}(\tilde{x},\Delta) = \mathbf{diag}(\mathbf{F}(x,\Delta), \lambda - \mathbf{c}(\Delta)^T x).$$

Problem (3) can be formulated as

$$\text{minimize } \tilde{c}^T \tilde{x} \text{ subject to } \tilde{x} \in \tilde{\mathcal{X}}_\rho,$$

where $\tilde{\mathcal{X}}_\rho$ is the robust feasible set corresponding to the function $\tilde{\mathbf{F}}$.

In the following, we thus consider a problem of the form

$$\mathcal{P}_\rho : \quad \text{minimize } c^T x \text{ subject to } x \in \mathcal{X}_\rho \tag{4}$$

and refer to it as a *robust semidefinite problem* (RSDP). In general, although $\mathcal{X}_\rho$ is convex, $\mathcal{P}_\rho$ is not a tractable problem—in particular, it is not an SDP. Our aim is to find a convex, inner approximation of $\mathcal{X}_\rho$ that is described by a linear matrix inequality constraint. This inner approximation is then used to find an upper bound on the optimal value of $\mathcal{P}_\rho$ by solving an SDP. In some cases, we can prove our results are nonconservative, that is, as in the so-called "full perturbation" case.

We refer to the set $\mathcal{X}_0$ (resp., problem $\mathcal{P}_0$, i.e., (1)) as the *nominal* feasible set (resp., nominal SDP). We shall assume that the nominal SDP is feasible, that is, $\mathcal{X}_0 \neq \emptyset$. Of course, the robust feasible set $\mathcal{X}_\rho$ may become empty for some $\rho > 0$; we return to this question later.

2.2. Linear-fractional representation. In this paper, we restrict our attention to functions $\mathbf{F}$ that are given by a "linear-fractional representation" (LFR):

$$\mathbf{F}(x,\Delta) = F(x) + L\Delta(I - D\Delta)^{-1}R(x) + R(x)^T(I - \Delta^T D^T)^{-1}\Delta^T L^T, \tag{5}$$

where $F(x)$ is defined in (1), $R(\cdot)$ is an affine mapping taking values in $\mathbf{R}^{q \times n}$, and $L \in \mathbf{R}^{n \times p}$ and $D \in \mathbf{R}^{q \times p}$ are given matrices. Together, the mappings $F(\cdot)$, $R(\cdot)$, the matrices L, D, the subspace $\mathcal{D}$, and the scalar ρ constitute our *perturbation model* for the nominal SDP (1).

The above class of models seems quite specialized. In fact, these models can be used in a wide variety of situations, for example, in the case where the (matrix) coefficients F_i in $\mathcal{P}_0$ are rational functions of the perturbation. The representation lemma, given below, and the examples of section 5 illustrate this point.

A constructive proof of the following result can be found in [37].

LEMMA 2.1. *For any rational matrix function $\mathbf{M} : \mathbf{R}^k \to \mathbf{R}^{n \times c}$, with no singularities at the origin, there exist nonnegative integers $r_1, \ldots, r_k$, and matrices $M \in \mathbf{R}^{n \times c}$,*

$L \in \mathbf{R}^{n\times N}$, $R \in \mathbf{R}^{N\times c}$, $D \in \mathbf{R}^{N\times N}$, *with* $N = r_1 + \cdots + r_k$, *such that* $\mathbf{M}$ *has the following linear-fractional representation* (*LFR*): *For all* δ *where* $\mathbf{M}$ *is defined*,

$$\mathbf{M}(\delta) = M + L\Delta\left(I - D\Delta\right)^{-1}R, \text{ where } \Delta = \mathbf{diag}\left(\delta_1 I_{r_1}, \ldots, \delta_k I_{r_k}\right). \tag{6}$$

Using the LFR lemma, we may devise LFR models for SDPs, where a perturbation vector $\delta \in \mathbf{R}^k$ enters rationally in the coefficient matrices. The resulting set $\mathcal{D}$ of perturbation matrices Δ is then a set of diagonal matrices of repeated elements, as in (6). Componentwise bounds on the vector δ, such as $|\delta|_i \le \rho$, $i = 1, \ldots, k$, translate into a norm-bound $\|\Delta\| \le \rho$ on the corresponding matrix Δ.

2.3. A special case. We distinguish a special case for which exact (nonconservative) results can be obtained via SDP. This is when $\mathbf{F}(x, \Delta)$ is block diagonal, each block being independently perturbed—precisely, when

$$\mathbf{F}(x, \Delta) = \mathbf{diag}\big(\mathbf{F}_1(x, \Delta_1), \ldots, \mathbf{F}_L(x, \Delta_L)\big), \tag{7}$$

where each $\mathbf{F}_i(x, \Delta_i)$ assumes the form shown in section 2.2 for appropriate L_i, R_i, D_i, with $\Delta_i \in \mathbf{R}^{p_i\times p_i}$, $i = 1, \ldots, L$, and $\mathcal{D}$ consists of block-diagonal matrices of the form

$$\mathcal{D} = \left\{\Delta = \mathbf{diag}(\Delta_1, \ldots, \Delta_L), \ \middle| \ \Delta_i \in \mathbf{R}^{p_i\times q_i}\right\}.$$

We refer to this situation as the *block-full perturbation* case. When $L = 1$, we speak of the *full perturbation* case. As will be seen later, all results given for $L = 1$ can be generalized to the case $L > 1$.

3. Robust solutions for SDPs.

Unless otherwise specified, we fix $\rho > 0$.

3.1. Full perturbations case. In this section, we consider the full perturbations case, that is, $\mathcal{D} = \mathbf{R}^{p\times q}$. We assume $\|D\| < \rho^{-1}$, which is a necessary and sufficient condition for $\mathbf{F}(x, \Delta)$ to be well defined for every $x \in \mathbf{R}^m$ and $\Delta \in \mathbf{R}^{p\times q}$, $\|\Delta\| \le \rho$.

The following lemma is a simple corollary of a classic result on quadratic inequalities, referred to as the $\mathcal{S}$-procedure [12]. Its proof is detailed in [16].

LEMMA 3.1. *Let* $F = F^T$, L, R, D *be real matrices of appropriate size. We have* $\det(I - D\Delta) \ne 0$ *and*

$$F + L\Delta(I - D\Delta)^{-1}R + R^T(I - D\Delta)^{-T}\Delta^T L^T \succeq 0 \tag{8}$$

for every Δ, $\|\Delta\| \le 1$, *if and only if* $\|D\| < 1$ *and there exists a scalar* τ *such that*

$$\begin{bmatrix} F - \tau LL^T & R^T - \tau LD^T \\ R - \tau DL^T & \tau(I - DD^T) \end{bmatrix} \succeq 0. \tag{9}$$

A direct application of the above lemma shows that, in the full perturbations case, the RSDP (4) is an SDP.

THEOREM 3.1. *When* $\mathcal{D} = \mathbf{R}^{p\times q}$, *the RSDP* (4) *and a corresponding solution* x *can be computed by solving the SDP in variables* x, τ:

$$\text{minimize } c^T x \text{ subject to } \begin{bmatrix} F(x) - \tau LL^T & R(x)^T - \tau LD^T \\ R(x) - \tau DL^T & \tau(\rho^{-2}I - DD^T) \end{bmatrix} \succeq 0. \tag{10}$$

Special barrier functions adapted to a conic formulation of the problem can be devised and yield an interior-point algorithm that has the same complexity as the nominal problem; see [24].

We may define the *maximum allowable perturbation level*, which is the largest number $\rho_{\max}$ such that $\mathcal{X}_\rho \neq \emptyset$ for every ρ, $0 \leq \rho \leq \rho_{\max}$ (note $\rho_{\max} > 0$ since $\mathcal{X}_0 \neq \emptyset$). Computing $\rho_{\max}$ is a (quasi-convex) generalized eigenvalue minimization problem [24, 11]:

$$\text{minimize } \lambda \text{ subject to } \begin{bmatrix} F(x) - \tau LL^T & R(x)^T - \tau LD^T \\ R(x) - \tau DL^T & \tau(\lambda I - DD^T) \end{bmatrix} \succeq 0.$$

Remark. The above exact results are readily generalized to the block-full perturbation case ($L > 1$) as defined in section 2.2.

3.2. Structured case. We now turn to the general case ($\mathcal{D}$ is now an arbitrary linear subspace). In this section, we associate with $\mathcal{D}$ the following linear subspace:

$$\begin{array}{c} \mathcal{B} \stackrel{\Delta}{=} \big\{(S,T,G) \in \mathbf{R}^{p\times p} \times \mathbf{R}^{q\times q} \times \mathbf{R}^{q\times p} \mid \\ S\Delta = \Delta T, \;\; G\Delta = -\Delta^T G^T \text{ for every } \Delta \in \mathcal{D}\big\}. \end{array} \tag{11}$$

As shown in [16], a general instance of problem (4) is NP-hard. Therefore, we look for upper bounds on its optimal value. The following lemma is a generalization of Lemma 3.1 that traces back to [17]. Its proof is detailed in [16].

LEMMA 3.2. *Let $F = F^T$, L, R, D be real matrices of appropriate size. Let $\mathcal{D}$ be a subspace of $\mathbf{R}^{p\times q}$, and denote by $\mathcal{B}$ the set of matrices associated with $\mathcal{D}$ as in* (11). *We have* $\det(I - D\Delta) \neq 0$ *and*

$$F + L\Delta(I - D\Delta)^{-1}R + R^T(I - D\Delta)^{-T}\Delta^T L^T \succ 0 \tag{12}$$

for every $\Delta \in \mathcal{D}$, $\|\Delta\| \leq 1$ if there exist a triple $(S,T,G) \in \mathcal{B}$ such that $S \succ 0$, $T \succ 0$, and

$$\begin{bmatrix} F - LSL^T & R^T - LSD^T + LG \\ R - DSL - GL^T & T - GD^T + DG - DSD^T \end{bmatrix} \succ 0. \tag{13}$$

Using Lemma 3.2, we obtain the following result.

THEOREM 3.2. *An upper bound on the RSDP* (4) *and a corresponding solution x can be computed by solving the SDP in variables x, S, T, G :*

$$\begin{array}{c} \textit{inf } c^T x \textit{ subject to } (S,T,G) \in \mathcal{B}, \;\; S \succ 0, \;\; T \succ 0, \\ \begin{bmatrix} F(x) - LSL^T & R(x)^T - LSD^T + LG \\ R(x) - DSL - GL^T & \rho^{-2}T - GD^T + DG - DSD^T \end{bmatrix} \succ 0. \end{array}$$

Note that when the perturbation is full, the variable G is zero and S, T are of the form τI_p, τI_q, resp., for some $\tau \geq 0$. We then recover the exact results of section 3.1.

As before, we may define the maximum allowable perturbation level, which is the largest number $\rho_{\max}$ such that $\mathcal{X}_\rho \neq \emptyset$ for every ρ, $0 \leq \rho \leq \rho_{\max}$. Computing a *lower bound* on this number is a (quasi-convex) generalized eigenvalue minimization problem:

$$\begin{array}{c} \inf \lambda \text{ subject to } (S,T,G) \in \mathcal{B}, \;\; S \succ 0, \;\; T \succ 0, \\ \begin{bmatrix} F(x) - LSL^T & R(x)^T - LSD^T + LG \\ R(x) - DSL - GL^T & \lambda T - GD^T + DG - DSD^T \end{bmatrix} \succ 0. \end{array} \tag{14}$$

4. Uniqueness and regularity of robust solutions. In this section, we derive uniqueness and regularity results for the RSDP in the case of full perturbations. As before, we first take $L = 1$ (one block), that is, $\mathcal{D} = \mathbf{R}^{p\times q}$. The results of this section remain valid in the general case $L > 1$ (several blocks).

We fix ρ, $0 < \rho < \rho_{\max}$. For simplicity of notation (and without loss of generality) we take $\rho = 1$ (and thus $\rho_{\max} > 1$). For well-posedness reasons, we must assume $\|D\| < 1$. We make the further assumption that $D = 0$ (in other words, $\mathbf{F}(x, \Delta)$ is affine in Δ). In section 4.5, we show how the case $D \neq 0$ can be treated.

For full perturbations and $D = 0$, the RSDP is the SDP

$$\text{minimize } c^T x \text{ subject to } \begin{bmatrix} F(x) - \tau LL^T & R(x)^T \\ R(x) & \tau I \end{bmatrix} \succeq 0. \tag{15}$$

4.1. Hypotheses. We assume that the SDP (15) (with $D = 0$) satisfies the following hypotheses:

H1. The *Slater* condition holds, that is, the problem is strictly feasible.

H2. The problem is *inf-compact,* meaning that any unbounded sequence (x_k) of feasible points (if any) produces an unbounded sequence of objectives. An equivalent condition is that the Slater condition holds for the dual problem [28, p. 317, Thm. 30.4].

H3. (a) The nullspace of the matrix

$$\lambda R_0 + \sum_{i=1}^{m} x_i R_i$$

is independent of $(\lambda, x) \neq (0, 0)$ and not equal to the whole space.

(b) For every x,

$$\begin{bmatrix} L^T \\ R(x) \end{bmatrix} \text{ is full column-rank.}$$

Hypotheses H1 and H2 ensure, in particular, the existence of optimal points for problem (15) and its dual. Hypotheses H3(a) and (b) are difficult to check in general, but sometimes can be easily tested in practical examples, as seen in section 5. We note that H3(a) implies that $R(x) \neq 0$ for every x.

Hypothesis H1 is equivalent to Robinson's condition [27], which can be expressed in terms of

$$\mathcal{F}(x, \tau) = \begin{bmatrix} F(x) - \tau LL^T & R(x)^T \\ R(x) & \tau I \end{bmatrix}.$$

Robinson's condition is stated in [27] as the existence of $x_0 \in \mathbf{R}^m$, $\tau_0 \in \mathbf{R}$ such that

$$0 \in \text{int } \left(\mathcal{F}(x_0, \tau_0) + d\mathcal{F}(x_0, \tau_0)\mathbf{R}^{m+1} - \mathcal{S}^+_{n+q} \right),$$

where $d\mathcal{F}$ is the differential of $\mathcal{F}$, and $\mathcal{S}^+_{n+q}$ is the set of positive semidefinite matrices of order $n + q$. The equivalence between H1 and Robinson's assumption is not true, in general. Here, this equivalence stems from the fact that the problem is convex and that the cone $\mathcal{S}^+_{n+q}$ has a nonempty interior.

Remark. Hypothesis H1 holds if and only if it holds for the nominal problem (1) (recall our assumption $\rho_{\max} > 1$). Also, hypothesis H2 implies $L \neq 0$ (otherwise, we can let $\tau \to \infty$ without affecting the objective value). If H2 holds for the nominal problem and $L \neq 0$, then H2 holds for the RSDP (15).

4.2. An equivalent nonlinear program. Let $x_{\text{opt}}, \tau_{\text{opt}}$ be optimal for (15). Hypothesis H3(a) ensures that any τ that is feasible for (15) is nonzero (otherwise, $R(x)$ would be zero for some x). We thus have $\tau_{\text{opt}} > 0$.

We introduce some notation. For $x \in \mathbf{R}^m$, $Z \in \mathbf{R}^{n \times n}$, $\tau > 0$ and $\mu \in \mathbf{R}$, define

$$d = \begin{bmatrix} c \\ 0 \end{bmatrix}, \quad y = \begin{bmatrix} x \\ \tau \end{bmatrix}, \quad Y = \mathbf{diag}(Z, \mu),$$

$$G(y) = F(x) - \tau L L^T - \frac{1}{\tau} R(x)^T R(x), \quad \mathcal{G}(y) = \mathbf{diag}(G(y), \tau - .99\tau_{\text{opt}}),$$
$$\mathcal{L}(y, Y) = d^T y - \mathbf{Tr} Y G(y).$$

Using Schur complements and $\tau_{\text{opt}} > 0$, we obtain that problem (15) can be rewritten as

$$\text{minimize } d^T y \text{ subject to } G(y) \succeq 0 \tag{16}$$

and that $y_{\text{opt}} = [x_{\text{opt}}^T \ \tau_{\text{opt}}]^T$ is optimal for (16). Our aim is first to prove that the so-called quadratic growth condition [10] holds at y_{opt} for problem (16). Then, we will apply the results of [10] to obtain uniqueness and regularity theorems.

4.3. Checking the quadratic growth condition. Following [10], we say that the *quadratic growth condition* (QGC) holds at y_{opt} if there exists a scalar $\alpha > 0$ such that, for every feasible y,

$$d^T y \geq d^T y_{\text{opt}} + \alpha \|y - y_{\text{opt}}\|^2 + o(\|y - y_{\text{opt}}\|^2).$$

Roughly speaking, this condition guarantees that y_{opt} is not on a facet on the boundary of the feasible set.

Define the set of dual variables associated with y_{opt} by

$$\mathcal{Y}(y_{\text{opt}}) = \left\{ Y = \mathbf{diag}(Z, \mu) \;\middle|\; Y \succeq 0, \ \ \mathbf{Tr} Y \frac{\partial G}{\partial y_i}(y_{\text{opt}}) = d_i, \ \ i = 1, \ldots, m+1 \right\}.$$

The following result is a direct consequence of a general result by Bonnans, Cominetti, and Shapiro [10]. Roughly speaking, this result states that, if an optimization problem satisfies Robinson's condition and has an optimal point, and if a certain "curvature" condition is satisfied, then the QGC holds at that point.

THEOREM 4.1. *With the notation above, if* H1 and H2 *hold, and if*

$$\exists\, Y \in \mathcal{Y}(y_{\text{opt}}) \text{ such that } \nabla^2_{yy} \mathcal{L}(y_{\text{opt}}, Y) > 0, \tag{17}$$

then problem (16) *satisfies the QGC.*

The following theorem is proven in appendix A.

THEOREM 4.2. *If* H1–H3 *hold, problem* (15) *satisfies the quadratic growth condition at every optimal point* y_{opt}. *Consequently, there exists a unique solution to the SDP* (15).

Remark. Note that the QGC is satisfied independent of the objective vector. This means that the boundary of the feasible set is strictly convex (it contains no facets).

4.4. Regularity results. In problem (15), the data consist of the matrices L, and F_i, R_i, $i = 0, \ldots, m$. We seek to examine the sensitivity of the problem with respect to small variations in F_i, L_i, and R_i.

In this section, we consider matrices $L(u)$, $R_i(u)$, and $F_i(u)$, $i = 0, \ldots, m$ that are functions of class $\mathcal{C}^1$ of a (small) parameter vector u. Define

$$F(x,u) = F_0(u) + \sum_{i=1}^m x_i F_i(u), \quad R(x,u) = R_0(u) + \sum_{i=1}^m x_i R_i(u).$$

We denote by $\mathcal{P}(u)$ the corresponding problem (15), where $F(\cdot)$, $R(\cdot)$, and L are replaced by $F(\cdot, u)$, $R(\cdot, u)$, and $L(u)$. We assume that $F(\cdot, 0) = F(\cdot)$, $R(\cdot, 0) = R(\cdot)$, and $L(0) = L$, so that $\mathcal{P}(0)$ is (15).

We first note that, in the vicinity of $u = 0$, problem $\mathcal{P}(u)$ satisfies the hypotheses H1 and H2 if $\mathcal{P}(0)$ does. In this case, for every $\epsilon > 0$ we may define the set $\mathcal{S}_\epsilon(u)$ of ϵ-suboptimal points of $\mathcal{P}(u)$:

$$\mathcal{S}_\epsilon(u) = \left\{ x \;\middle|\; x \text{ is feasible for } \mathcal{P}(u) \text{ and } c^T x \le v(u) + \epsilon \right\},$$

where $v(u)$ is the optimal value of $\mathcal{P}(u)$.

Recall that, if $\mathcal{P}_0$ satisfies hypotheses H1 and H2, the optimal value $v(u)$ is continuous, and even directionally differentiable, at $u = 0$ [32, Thm. 5.1]. With the QGC in force, and using [31, Thm. 4.1], we can give quite complete regularity results for the robust *solutions*.

THEOREM 4.3. *If hypotheses* H1–H3 *hold for* $\mathcal{P}(0)$, *then for every* $\epsilon = O(u)$, *there exists a* $\gamma > 0$ *and a neighborhood* V *of* $u = 0$ *such that for every* $u \in V$ *and* $x \in \mathcal{S}_\epsilon(u)$, *we have*

$$\|x - x(0)\| \le \gamma \|u\|^{1/2}. \tag{18}$$

When H1–H3 hold for $\mathcal{P}(0)$, the above theorem states that every (sufficiently) suboptimal solution to $\mathcal{P}(0)$ is Hölder-stable (with coefficient 1/2). This is true, in particular, for any optimal solution of $\mathcal{P}(u)$ (that is, for $\epsilon = 0$). The fact that the theorem remains true for $\epsilon > 0$ guarantees regularity of *numerical* solutions to the RSDP. The main consequence is that even if the nominal SDP is ill conditioned (with respect to variations in the F_i's), the RSDP becomes well conditioned for every $\rho > 0$.

Now assume $\rho \neq 1$. We seek to examine the behavior of problem (10) (with $D = 0$) when the uncertainty level ρ for $0 < \rho < \rho_{\max}$ varies. This is a special case of the problem examined above, with $u = \rho$, $F(\cdot, u) = F(\cdot)$, $R(\cdot, u) = R(\cdot)$, $L(u) = \rho L$.

COROLLARY 4.1. *For every* ρ, $0 < \rho < \rho_{\max}$, *the solution to* (10) (*with* $D = 0$) *is unique and satisfies the regularity results* (*written with* $u = \rho$) *of Theorem* 4.3.

Remark. The results of this section are all valid in the block-full perturbation case ($L > 1$), as defined in section 2.2. Of course, the conditions given in H3 should be understood blockwise.

4.5. Case $D \neq 0$.

When $D \neq 0$, we can get back to the case $D = 0$ as follows. Recall that we have $\|D\| < 1$ in order to ensure that $\mathbf{F}(x, \Delta)$ is defined everywhere on $\mathcal{D}$. With this assumption, we can define, for $x \in \mathbf{R}^m$ and $\tau > 0$,

$$\begin{aligned}
\tilde{L} &= L(I - D^T D)^{-1/2}, \\
\tilde{R}(x) &= (I - DD^T)^{-1/2} R(x), \\
\tilde{F}(x) &= F(x) - LD^T (I - DD^T)^{-1} R(x) - R(x)^T (I - DD^T)^{-1} L^T, \\
\tilde{G}(y) &= \tilde{F}(x) - \tau \tilde{L}\tilde{L}^T - \frac{1}{\tau} \tilde{R}(x)^T \tilde{R}(x).
\end{aligned}$$

Using Schur complements, we have, for every x and $\tau > 0$,

$$\tilde{G}(y) \succeq 0 \text{ if and only if } \begin{bmatrix} F(x) - \tau LL^T & R(x)^T - \tau LD^T \\ R(x) - \tau DL^T & \tau(I - DD^T) \end{bmatrix} \succeq 0.$$

Hypothesis H3 holds for $\tilde{L}, \tilde{R}(\cdot)$ if and only if it holds for $L, R(\cdot)$. We can then follow the steps detailed previously.

Corollary 4.2. *If the SDP* (10) (*with* $\rho = 1$) *satisfies* H1–H3 *and if* $\|D\| < 1$, *then the results of Theorem* 4.3 *hold.*

5. Examples.

5.1. Unstructured perturbations. Assume

$$\mathbf{F}(x, \Delta) = F(x) + \Delta_0 + \Delta_0^T + \sum_{i=1}^{m} x_i(\Delta_i + \Delta_i^T),$$

where $\Delta = [\Delta_0 \ldots \Delta_m]$. This case corresponds to the representation in section 5, with

$$L = I, \quad R(x) = \begin{bmatrix} 1 \\ x \end{bmatrix} \otimes I, \quad D = 0, \quad \mathcal{D} = \mathbf{R}^{n \times nm}. \tag{19}$$

Using Lemma 3.2, we obtain that problem (4) is equivalent to the SDP

$$\text{minimize } c^T x \text{ subject to } \begin{bmatrix} F(x) - \tau I & \begin{bmatrix} 1 & x^T \end{bmatrix} \otimes \rho I \\ \begin{bmatrix} 1 & x^T \end{bmatrix}^T \otimes \rho I & \tau I \end{bmatrix} \succeq 0. \tag{20}$$

It turns out that we may get rid of the variable τ and get back to a convex problem of the same size as that of the unperturbed problem (1). To see this, first note that every feasible variable τ in problem (20) is strictly positive. Use Schur complements to rewrite the matrix inequality in (20) as

$$F(x) \succeq \left(\tau + \rho^2 \frac{1 + \|x\|^2}{\tau} \right) I, \quad \tau > 0.$$

Minimizing (over variable τ) the scalar in the left-hand side of the above inequality shows that the RSDP (1) is equivalent to

$$\text{minimize } c^T x \text{ subject to } F(x) \succeq 2\rho\sqrt{\|x\|^2 + 1} \cdot I. \tag{21}$$

Formulation (21) is more advantageous than (20), since (21) involves a (convex) matrix inequality constraint of the same size as the original problem. As noted before, special barrier functions can be devised for this problem and yield an interior-point algorithm that has the same complexity as the original problem; see [24].

We note that, with the above choice for L, R, hypothesis H3 holds, which yields the following result.

Theorem 5.1. *The optimal value of the RSDP* (20) *can be computed by solving the convex problem* (21). *If* (21) *satisfies hypotheses* H1 and H2, *then for every* $\rho > 0$, *the solution is unique and satisfies the regularity conditions of Theorem* 4.3.

Remark. A sufficient condition for hypotheses H1 and H2 to hold for (21) is that they hold for the nominal problem. A more restrictive sufficient condition is that the nominal feasible set $\mathcal{X}_0$ is nonempty and bounded, and $\rho < \rho_{\max}$.

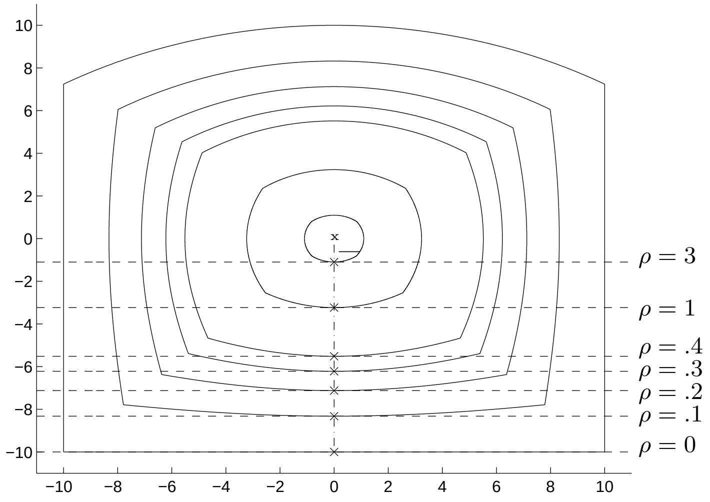


FIG. 1. *Nominal and robust solutions of an SDP, with a* 5×5 *matrix* $F(x)$*. Here* $\rho_{\max} = 5$.

5.2. Robust center of a linear matrix inequality. In this section, for $\rho > 0$, we consider the SDP (21) and corresponding feasible (convex) set $\mathcal{X}_\rho$. We assume that $\mathcal{X}_0$ is nonempty and bounded, and that $\mathcal{P}_0$ is strictly feasible. Then, for every ρ, $0 < \rho < \rho_{\max}$, $\mathcal{X}_\rho$ is nonempty and bounded, and we can define a (unique) solution $x(\rho)$ to the strictly convex problem (21).

In view of Corollary 4.1, $x(\rho)$ is a continuous function of ρ in $]0\,\rho_{\max}[$. Since $(\mathcal{X}_\rho)$ is a decreasing family of bounded sets, we may define

$$x^* = \lim_{\rho \to \rho_{\max}} x(\rho). \tag{22}$$

Note that x^* is independent on the objective vector c.

Thus, to the matrix inequality $F(x) \succeq 0$, we may associate the *robust center,* defined by (22). The robust center has the property of being the most tolerant (with respect to unstructured perturbation) among the feasible points.

An example is depicted in Fig. 1. The nominal feasible set $\mathcal{X}_0$ is described by a linear matrix inequality $F(x) \succeq 0$, where F is a 5×5 matrix. For various values of ρ, we seek to minimize x_2. The dashed lines correspond to the optimal objectives. As ρ increases, we observe that the robust feasible sets shrink. A crucial property of these robust sets is that they do not possess any straight faces, as observed in the figure. For $\rho = \rho_{\max} \simeq 5$, the robust feasible set is a singleton (in this example, $x^\star = 0$). When $\rho = 0$, the optimal solution is not unique and not continuous with respect to changes in the coefficient matrices F_i, $i = 0, 1, 2$ (although the optimal value is continuous). Since the sets $\mathcal{X}_\rho$ become strictly convex as soon as $\rho > 0$, the resulting robust solutions are continuous.

5.3. Robust linear programs. An interesting special case arises with linear programming (LP). Consider the LP

$$\text{minimize } c^T x \text{ subject to } a_i^T x \geq b_i,\ i = 1, \ldots, L.$$

Assume that the a_i's and b_i's are subject to unstructured perturbations. The perturbed value of $[a_i^T \ b_i]^T$ is $[a_i^T \ b_i]^T + \delta_i$, where $\|\delta_i\|_2 \leq \rho$, $i = 1, \ldots, L$. We seek a robust solution to our problem, which is a special case of the block-full perturbation case referred to in section 2.2, with $\mathbf{F}$ given by (7), and

$$\mathbf{F}_i(x, \Delta_i) = a_i^T x - b_i + 2[x^T \ \ -1]\Delta_i, \ \ i = 1, \ldots, L,$$

where $\Delta_i = \delta_i/2$, and $\mathcal{D}$ is the set of diagonal, $L \times L$ matrices. The robust LP is

$$\text{minimize } c^T x \text{ subject to } a_i^T x - \rho\sqrt{\|x\|_2^2 + 1} \geq b_i, \ i = 1, \ldots, L. \tag{23}$$

The above program is readily written as an SDP by introducing slack variables. In fact, the robust LP is a second-order cone program (SOCP) for which efficient special-purpose interior-point methods are available [24, 20, 23].

We note that hypothesis H3 holds blockwise. This yields the following result.

Theorem 5.2. *The optimal value of the robust LP can be computed by solving the convex problem* (23)*. If the latter satisfies hypotheses* H1 and H2, *then for every* ρ, $0 < \rho < \rho_{\max}$, *the solution is unique and satisfies the regularity conditions of Theorem* 4.3.

In [6], robust linear programming is studied in detail. For a wide class of perturbation models, where the data of every linear constraint vary in an ellipsoid, explicit robust solutions are constructed using convex SOCPs. Reference [23] also provides examples of robust linear programs solved via SOCP.

5.4. Robust eigenvalue minimization. Consider the case where the nominal problem consists of minimizing the largest eigenvalue of a matrix-valued function $F(x)$:

$$\text{minimize } \lambda_{\max}(F(x)). \tag{24}$$

When $F(\cdot)$ is subject to unstructured perturbations (as defined in section 5.1), the robust version of the problem is

$$\text{minimize } \lambda + 2\rho\sqrt{\|x\|^2 + 1} \text{ subject to } \lambda I \succeq F(x),$$

or equivalently

$$\text{minimize } \lambda_{\max}(F(x)) + 2\rho\sqrt{\|x\|^2 + 1}. \tag{25}$$

Let $\rho > 0$. When written in an SDP form, the above problem satisfies the hypotheses H1–H3. From Theorem 4.3 we obtain that the solution is unique. If we consider that the data of the above problem consist of the matrices F_i, $i = 0, \ldots, m$, then we know that the corresponding solution is Hölder-stable (with coefficient 1/2). Since the problem is unconstrained, we can use a result of Shapiro [31, Thm. 3.1], by which we conclude that the solution is actually Lipschitz stable (inequality (18) holds with the exponent 1/2 replaced by 1). Finally, using the results from Attouch [3], we can show that computing the solution for $\rho \to 0$ amounts to selecting the minimum norm solution among the solutions of the nominal problem.

Theorem 5.3. *The optimal value of the min-max problem* (24) *can be computed by solving the convex problem* (25)*. For every* $\rho > 0$, *the solution is unique and is Lipschitz stable with respect to perturbations in* F_i, $i = 0, \ldots, m$. *When* $\rho \to 0$, *the solution converges to the minimum norm solution of the nominal problem* (24).

Remark. In this case, the RSDP is a regularized version of the nominal SDP, which belongs to the class of Tikhonov regularizations [34]. The regularization parameter is 2ρ and is chosen according to some a priori information on uncertainty associated with the nominal problem's data. Taking ρ close to zero can be used as a *selection* procedure for choosing a particular (minimum norm, regular) solution among the (not necessarily unique and/or regular) solutions of the nominal problem.

Problem (25) is particularly suitable to the recent so-called $\mathcal{U}$-Newton algorithms for solving problem (24). These methods, described in [21, 25], require that the Hessian of the "smooth part" (the so-called $\mathcal{U}$-Hessian) of the objective of (24) be positive definite. For general mappings $F(\cdot)$, this property is not guaranteed. However, when looking at the robust problem (25), we see that the modified $\mathcal{U}$-Hessian is guaranteed to be positive definite for every x and $\rho > 0$. This indicates that the RSDP approach may be used to devise *robust algorithms* for solving SDPs.

5.5. Robust SOCPs. An SOCP is a problem of the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \|C_i x + d_i\| \leq e_i^T x + f_i, \quad i = 1, \dots, L, \end{array} \tag{26}$$

where $C_i \in \mathbf{R}^{n_i \times m}$, $d_i \in \mathbf{R}^{n_i}$, $e_i \in \mathbf{R}^m$, $f_i \in \mathbf{R}$, $i = 1, \dots, L$. SOCPs can be formulated as SDPs, but special-purpose, more efficient algorithms can be devised for them; see [24, 5, 23].

Assuming that C_i, d_i, e_i, f_i are subject to linear—or even rational—uncertainty, we may formulate the corresponding RSDP as an SDP. This SDP can be written as an SOCP if the uncertainty is unstructured and affects each constraint independently.

The subject of robust SOCPs is explored in [5] in detail. Explicit SDPs that yield robust counterparts to SOCPs *nonconservatively* are given for a wide class of uncertainty structures. In some cases, albeit not all, the robust counterpart is itself an SOCP. In [16, 14], the special case of least-squares problems with uncertainty in the data is studied at length.

5.6. Robust maximum norm minimization. Several engineering problems take the form

$$\text{minimize } \|H(x)\|, \tag{27}$$

where

$$H(x) = H_0 + \sum_{i=1}^{m} x_i H_i,$$

and H_i, $i = 1, \dots, m$ are given $p \times q$ matrices. A well-known instance of this problem is the linear least-squares problem, with $H(x) = Ax - b$. Another example is a minimal norm extension problem for a Hankel operator studied in [18], in which H_0 is a given (arbitrary) $n \times n$ Hankel matrix and H_i, $i = 1, \dots, m$ is the $n \times n$ Hankel matrix associated with the polynomial $1/z^i$. In practice, the matrices H_i, $i = 0, \dots, m$ are subject to perturbation, which motivates a study of the robust version of problem (27). Note that the least-squares case is extensively studied in [16].

Consider the *full perturbation case*, which occurs when each H_i is perturbed independently in a linear manner. Precisely, consider the matrix-valued function

$$\mathbf{H}(x, \Delta) = H_0 + \Delta_0 + \sum_{i=1}^{m} x_i (H_i + \Delta_i),$$

where $\Delta = [\Delta_0 \ldots \Delta_m]$. For a given $\rho > 0$, we address the min-max problem

$$\min_x \max_{\|\Delta\| \leq \rho} \|\mathbf{H}(x, \Delta)\|. \tag{28}$$

This problem is an RSDP for which we can get exact results using SDP. Indeed, for every $x \in \mathbf{R}^m$ and $\lambda \geq 0$, the property

$$\max_{\|\Delta\| \leq \rho} \|\mathbf{H}(x, \Delta)\| \leq \lambda$$

is equivalent to $\mathbf{F}(x, \lambda, \Delta) \succeq 0$ for every Δ, $\|\Delta\| \leq \rho$, where

$$\mathbf{F}(x, \lambda, \Delta) = F(x, \lambda) + L\Delta R(x) + R(x)^T \Delta^T L^T,$$

where

$$F(x, \lambda) = \begin{bmatrix} \lambda I & H(x) \\ H(x)^T & \lambda I \end{bmatrix}, \quad L = \begin{bmatrix} I \\ 0 \end{bmatrix}, \quad R(x) = \begin{bmatrix} 0 & \begin{bmatrix} 1 \\ x \end{bmatrix} \otimes I \end{bmatrix}.$$

We thus write problem (28) as (4), where the perturbation set $\mathcal{D}$ is $\mathbf{R}^{p \times q}$.

Applying Theorem 3.2, we obtain that the RSDP above is equivalent to the SDP (15) (with $D = 0$). As in section 5.1, we may get rid of the variable τ and obtain the equivalent formulation

$$\text{minimize } \|H(x)\| + \rho\sqrt{\|x\|^2 + 1}. \tag{29}$$

This RSDP satisfies hypotheses H1–H3, so we conclude that the results of Theorem 4.3 hold. As in robust eigenvalue minimization, we can get improved results using [31, section 3, Thm. 3.1].

THEOREM 5.4. *The optimal value of the min-max problem* (28) *can be computed by solving the convex problem* (29). *For every $\rho > 0$, the solution is unique and Lipschitz stable with respect to perturbations in H_i, $i = 0, \ldots, m$. When $\rho \to 0$, the solution converges to the minimum norm solution of the nominal problem* (27).

Remark. As for the RSDP arising in robust eigenvalue minimization, the robust minimum norm minimization problem is a regularized version of the nominal problem, which belongs to the class of Tikhonov regularizations.

We now consider the general case where each matrix H_i in (27) is perturbed in a structured manner. To be specific, we concentrate on the minimal norm extension problem mentioned above.

In practice, the matrix H_0 is obtained from measurement and is thus subject to error. We may assume that this matrix is constructed from an $n \times 1$ vector $h_0(\delta) = h_0 + \delta$, where δ is unknown but bounded. The perturbed matrix H_0 is of the form

$$H_0(\Delta) = H_0 + L\Delta R,$$

where L, R are given matrices (the exact form of which we do not detail), and

$$\Delta \in \mathcal{D} = \{\mathbf{diag}(\delta_1 I_1, \ldots, \delta_n I_n) \mid \delta_i \in \mathbf{R}, \ \ i = 1, \ldots, n\}.$$

(In the above, each δ_i corresponds to the uncertainty associated with the ith antidiagonal of H_0.) We address the min-max problem

$$\text{minimize } \max_{\Delta \in \mathcal{D}, \ \|\Delta\| \leq \rho} \|H(x) + L\Delta R\|, \tag{30}$$

where $\rho \geq 0$ is given.

This problem is amenable to the robustness analysis technique. Defining

$$\mathcal{S} \overset{\Delta}{=} \left\{ \mathbf{diag}(S_1, \ldots, S_n) \;\middle|\; S_i \in \mathbf{R}^{i \times i},\;\; i = 1, \ldots, n \right\},$$

we obtain the following result.

THEOREM 5.5. *An upper bound on the objective value of the min-max problem* (30) *can be computed by solving the SDP in variables* x, S, G*:*

$$\inf \lambda \text{ subject to } \quad S = S^T,\; G = -G^T \in \mathcal{S}, \quad \begin{bmatrix} \lambda I - LSL^T & H(x) & LG \\ H(x)^T & \lambda I & \rho R \\ G^T L^T & \rho R^T & S \end{bmatrix} \succ 0.$$

5.7. Polynomial interpolation. This example, taken from [16], can be formulated as an RSDP with rational dependence. For given integers $n \geq 1$, k, we seek a polynomial of degree $n-1$ $p(t) = x_1 + \cdots + x_n t^{n-1}$ that interpolates given points (a_i, b_i), $i = 1, \ldots, k$, that is,

$$p(a_i) = b_i, \;\; i = 1, \ldots, k.$$

If we assume that (a_i, b_i) are known exactly, we obtain a linear equation in the unknown x, with a Vandermonde structure:

$$\begin{bmatrix} 1 & a_1 & \ldots & a_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & a_k & \ldots & a_k^{n-1} \end{bmatrix}, \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix},$$

which can be solved via standard least-squares techniques.

Now assume that the interpolation points are not known exactly. For instance, we may assume that the b_i's are known, while the a_i's are parameter dependent:

$$a_i(\delta) = a_i + \delta_i, \;\; i = 1, \ldots, k,$$

where the δ_i's are unknown but bounded: $|\delta_i| \leq \rho$, $i = 1, \ldots, k$, where $\rho \geq 0$ is given. We seek a robust interpolant, that is, a solution x that minimizes

$$\max_{\|\delta\|_\infty \leq \rho} \|\mathbf{A}(\delta)x - b\|, \tag{31}$$

where

$$\mathbf{A}(\delta) = \begin{bmatrix} 1 & a_1(\delta) & \ldots & a_1(\delta)^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & a_k(\delta) & \ldots & a_k(\delta)^{n-1} \end{bmatrix}.$$

The above problem is an RSDP. Indeed, it can be shown that

$$\begin{bmatrix} \mathbf{A}(\delta) & b \end{bmatrix} = \begin{bmatrix} \mathbf{A}(0) & b \end{bmatrix} + L\Delta(I - D\Delta)^{-1}R,$$

where

$$L = \bigoplus_{i=1}^{k} \begin{bmatrix} 1 & a_i & \ldots & a_i^{n-2} \end{bmatrix}, R = \begin{bmatrix} R_1 \\ \vdots \\ R_k \end{bmatrix}, \;\; D = \bigoplus_{i=1}^{k} D_i, \;\; \Delta = \bigoplus_{i=1}^{k} \delta_i I_{n-1},$$

and, for each i, $i = 1, \ldots, k$,

$$R_i = \begin{bmatrix} 0 & 1 & a_i & \ldots & a_i^{n-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & a_i \\ 0 & \ldots & \ldots & 0 & 1 \end{bmatrix} \in \mathbf{R}^{(n-1)\times n},$$

$$D_i = \begin{bmatrix} 0 & 1 & a_i & \ldots & a_i^{n-3} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & a_i \\ 0 & \ldots & \ldots & 0 & 1 \end{bmatrix} \in \mathbf{R}^{(n-1)\times(n-1)}.$$

(Note that $\det(I - D\Delta) \neq 0$, since D is strictly upper triangular.) With the above notation, if we define $\mathbf{F}(x, \Delta)$ as in section 5, then problem (31) can be formulated as the RSDP (4).

With the approach described in this paper, one can compute an upper bound for the minimizing value of (31), and a corresponding suboptimal x. We do not know if the problem can be solved exactly in polynomial time, e.g., using SDP. We conjecture (as the reviewers of this paper did) that the answer is no. To motivate this claim, note that the solution to the problem of computing (31) for arbitrary *affine* functions $\mathbf{A}$ is already NP-hard [16].

5.8. Error-in-variables RSDPs. In many SDPs that arise in engineering, the variable x represents physical parameters that can be implemented with finite absolute precision only. A typical example is integer programming, where integer solutions to (linear) programs are sought. These problems (which are equivalent to integer programming) are NP-hard. We now show that we may find upper bounds on these problems using robustness analysis.

Consider, for instance, the problem of finding a solution x to the feasibility SDP

$$\text{find an integer vector } x \text{ such that } F(x) \succeq 0. \tag{32}$$

Now, consider the robust SDP

$$\begin{gathered} \text{maximize } \lambda \text{ subject to} \\ \lambda I \leq F_0 + \sum_{i=1}^{m} (x_i + \Delta x_i) F_i \text{ for every } \Delta x,\ \|\Delta x\|_\infty \leq 1/2. \end{gathered} \tag{33}$$

Assume there exists a feasible pair $(x_{\text{feas}}, \lambda)$ to the above problem, with $\lambda \geq 0$. By construction, x_{feas} satisfies $F(x_{\text{feas}}) \succeq 0$. Furthermore, any vector x chosen such that $\|x - x_{\text{feas}}\|_\infty \leq 1/2$ is guaranteed to satisfy $F(x) \succeq 0$. This is true, in particular, for x_{int}, the integer closest to x_{feas}. Thus, if we know a positive lower bound λ, and corresponding feasible point for problem (33), then we can compute an integer solution to our original problem.

Finding a lower bound for (33) and an associated feasible point can be done as follows. For i, $1 \leq i \leq m$, define $F_i = 2L_iR_i$, where $L_i, R_i^T \in \mathbf{R}^{n \times r_i}$, $r_i = \mathbf{Rank}\, F_i$. Let

$$L = \begin{bmatrix} L_1 & \ldots & L_m \end{bmatrix}, \quad R = \begin{bmatrix} R_1 \\ \vdots \\ R_m \end{bmatrix}, \text{ and } \mathcal{D} = \left\{ \Delta = \bigoplus_{i=1}^{m} \Delta x_i I_{r_i},\ \ \Delta x_i \in \mathbf{R} \right\}.$$

Problem (33) can be formulated as

$$\text{maximize } \lambda \text{ subject to } \quad \begin{array}{l} \lambda I \leq F(x) + L\Delta R + R^T \Delta^T L^T \\ \text{for every } \Delta \in \mathcal{D},\ \|\Delta\| \leq 1/2. \end{array} \tag{34}$$

The above is a special instance of the structured problem examined in section 3.2. Define

$$\mathcal{S} \overset{\Delta}{=} \left\{ \mathbf{diag}(S_1, \dots, S_m) \;\middle|\; S_i \in \mathbf{R}^{r_i \times r_i},\ \ i = 1, \dots, n \right\}.$$

THEOREM 5.6. *A sufficient condition for an integer solution to the feasibility SDP* (32) *is that the constraints*

$$\lambda \geq 0,\ \ S = S^T \in \mathcal{S},\ \ G = -G^T \in \mathcal{S},\quad \begin{bmatrix} F(x) - \lambda I - LSL^T & (1/2)R^T + LG \\ (1/2)R - GL^T & S \end{bmatrix} \succ 0$$

are feasible. If x_{feas} is feasible for the above constraints, then any integer vector closest to x_{feas} (*in the maximum norm sense*) *is feasible for* (32).

6. Conclusions. In this paper, we considered semidefinite programs subject to uncertainty. Assuming the latter is unknown but bounded, we have provided sufficient conditions that guarantee "robust" solutions to exist via SDPs. Under some conditions (detailed in section 4), the robust solution is unique, and not surprisingly, stable. The method can then be used to regularize possibly ill-conditioned problems. For some perturbation structures (as for unstructured perturbations), the conditions are also necessary. That is, there is no conservatism induced by the method.

The paper raises several open questions.

In our description, we have considered the problem of making the primal SDP robust, thereby obtaining upper bounds on an SDP subject to uncertainty. The dual point of view should be very interesting. One might be interested in applying the approach to the dual problem instead. Does this lead to lower bounds on the perturbed problem? Also, in some cases, the RSDP approach leads to a unique (and stable) primal solution. May we obtain a unique solution to the dual problem by making the latter robust? (This would lead to analyticity of the primal solution; see [32].)

As seen in section 5.2 the notion of robust center has, certainly, connections with the well-known analytic center; is the latter related to some robustness characterization?

It seems that the RSDP method could be useful for deriving fast and robust (stable) algorithms for solving SDPs (see section 5.4), especially in connection with maximum eigenvalue minimization.

Finally, as said in section 2.2 (Lemma 2.1), an SDP with coefficient matrices depending rationally on a perturbation vector can always be represented by an LFR model. Now, this LFR model is not unique. However, the results given here (for example, Theorem 3.2) hinge on a particular linear-fractional representation for a perturbed SDP. Hence we have the question: are our results independent of the chosen representation? We partially answer this difficult question in Appendix B.

Appendix A. Proof of Theorem 4.2. We take the notation of section 4. Let $Y = \mathbf{diag}(Z, \mu)$ be dual variables associated with $(x_{\mathrm{opt}}, \tau_{\mathrm{opt}})$ that are optimal (their existence is guaranteed by H1 and H2). Then, $Y \in \mathcal{Y}(y_{\mathrm{opt}})$. Let us show that condition (17) holds for this choice of Y.

Since the problem satisfies H1 and H2, the complementarity conditions hold; therefore, the (optimal) dual variable μ associated with the constraint $\tau = \tau_{\rm opt}$ is zero. Consequently the variable Z is nonzero (recall $c \neq 0$). Using

$$\mathbf{Tr} Y \frac{\partial G}{\partial y_{m+1}}(y_{\rm opt}) = d_{m+1} = 0,$$

we obtain

$$\tau_{\rm opt}^2 \mathbf{Tr} LL^T Z = \mathbf{Tr} R(x_{\rm opt})^T R(x_{\rm opt}) Z.$$

From $\tau_{\rm opt} \neq 0$ (implied by H3(a)), and using hypothesis H3(b), we can show that

$$\mathbf{Tr} LL^T Z = 0 \text{ and } \mathbf{Tr} R(x_{\rm opt})^T R(x_{\rm opt}) Z = 0$$

are impossible for $Z \succeq 0$, $Z \neq 0$. This yields $\mathbf{Tr} R(x_{\rm opt})^T R(x_{\rm opt}) Z > 0$.

Now let $\xi \in \mathbf{R}^m$ and $\lambda \in \mathbf{R}$, and define

$$\begin{aligned} \Phi(\xi, \lambda) &= d^2 \mathcal{G}(x_{\rm opt}, \tau_{\rm opt})[(\xi, \lambda), (\xi, \lambda)], \\ \phi(\xi, \lambda) &= \left[\begin{array}{c} \xi \\ \lambda \end{array} \right]^T \nabla^2_{yy} \mathcal{L}(y_{\rm opt}, Y) \left[\begin{array}{c} \xi \\ \lambda \end{array} \right] = -\mathbf{Tr} Z \Phi(\xi, \lambda). \end{aligned}$$

We have, for every i, j, $1 \leq i, j \leq m$,

$$\begin{aligned} &\frac{\partial G}{\partial x_i} = F_i - \frac{1}{\tau}(R(x)^T R_i + R_i^T R(x)), \quad \frac{\partial G}{\partial \tau} = -LL^T + \frac{1}{\tau^2} R(x)^T R(x), \\ &\frac{\partial^2 G}{\partial x_i \partial x_j} = -\frac{1}{\tau}(R_j^T R_i + R_i^T R_j), \quad \frac{\partial^2 G}{\partial x_i \partial \tau} = \frac{1}{\tau^2}(R(x)^T R_i + R_i^T R(x)), \\ &\frac{\partial^2 G}{\partial \tau^2} = -\frac{2}{\tau^3} R(x)^T R(x). \end{aligned}$$

By summation, we have

$$\begin{aligned} -\Phi(\xi, \lambda) &= \frac{2}{\tau_{\rm opt}}(R(\xi) - R(0))^T (R(\xi) - R(0)) + 2\frac{\lambda^2}{\tau_{\rm opt}^3} R(x_{\rm opt})^T R(x_{\rm opt}) \\ &\quad - \frac{\lambda}{\tau_{\rm opt}^2} \left(R(x_{\rm opt})^T (R(\xi) - R(0)) + (R(\xi) - R(0))^T R(x_{\rm opt}) \right) \\ &= \frac{\lambda^2}{\tau_{\rm opt}^3} R(x_{\rm opt})^T R(x_{\rm opt}) + \frac{1}{\tau_{\rm opt}} \mathcal{R}^T \mathcal{R} + \frac{1}{\tau_{\rm opt}} (R(\xi) - R(0))^T (R(\xi) - R(0)), \end{aligned}$$

where $\mathcal{R} = R(\xi) - R(0) - \dfrac{\lambda}{\tau_{\rm opt}} R(x_{\rm opt})$. We obtain finally,

$$\begin{aligned} \phi(\xi, \lambda) &= \frac{1}{\tau_{\rm opt}} \mathbf{Tr} Z \mathcal{R}^T \mathcal{R} + \frac{\lambda^2}{\tau_{\rm opt}^3} \mathbf{Tr} Z R(x_{\rm opt})^T R(x_{\rm opt}) \\ &\quad + \frac{1}{\tau_{\rm opt}} \mathbf{Tr} Z (R(\xi) - R(0))^T (R(\xi) - R(0)). \end{aligned}$$

If $\phi(\xi, \lambda) = 0$, then $\lambda = 0$ (from $\mathbf{Tr} R(x_{\rm opt})^T R(x_{\rm opt}) Z > 0$), and thus $\mathbf{Tr} Z \mathcal{R}^T \mathcal{R} = 0$ with $\mathcal{R} = R(\xi) - R(0)$. Since $Z \succeq 0$, this means that every column of $Z^{1/2}$ belongs to the nullspace of $R(\xi) - R(0)$. Now assume $\xi \neq 0$. By hypothesis H3(a), we obtain that every column of $Z^{1/2}$ also belongs to the nullspace of $R(x_{\rm opt})$, which contradicts

$\mathbf{Tr}R(x_{\rm opt})^TR(x_{\rm opt})Z > 0$. We conclude that $\nabla^2_{yy}\mathcal{L}$ is positive definite at $(y_{\rm opt}, Y)$. Thus, problem (15) satisfies the QCG.

Appendix B. Invariance with respect to the LFR model. In this section, we show that the sufficient conditions obtained in this paper are, in some sense, independent of the LFR model used to describe the perturbation structure.

Consider a function $\mathbf{F}$ taking values in the set of symmetric matrices having an LFR such as that in section 5. This function can be written in a more symmetric form,

$$\mathbf{F}(\Delta) = F + \tilde{L}\tilde{\Delta}(I - D\tilde{\Delta})^{-1}\tilde{L}^T, \tag{35}$$

where we have dropped the dependence on x for convenience, and

$$\tilde{L} = \left[\begin{array}{cc} L & R^T \end{array}\right], \quad \tilde{D} = \left[\begin{array}{cc} 0 & D^T \\ D & 0 \end{array}\right], \quad \tilde{\Delta} = \left[\begin{array}{cc} 0 & \Delta \\ \Delta^T & 0 \end{array}\right].$$

It is easy to check that, if an invertible matrix Z satisfies the relation $Z\tilde{\Delta}Z^T = \tilde{\Delta}$ for every $\Delta \in \mathcal{D}$, then

$$\mathbf{F}(\Delta) = F + (\tilde{L}Z)\tilde{\Delta}(I - (Z^T\tilde{D}Z)\tilde{\Delta})^{-1}(\tilde{L}Z)^T.$$

In other words, the "scaled" triple $(F, (\tilde{L}Z), (Z^T\tilde{D}Z))$ can be used to represent $\mathbf{F}$ instead of $F, \tilde{L}, \tilde{D}$ in (35). By spanning valid scaling matrices Z, we span a subset of all LFR models that describe $\mathbf{F}$.

A valid scaling matrix Z can be constructed as follows. Let $(S, T, G) \in \mathcal{B}$, and define

$$Z = \left[\begin{array}{cc} T^{-1/2} & 0 \\ 0 & S^{1/2} \end{array}\right]\left[\begin{array}{cc} I & G \\ 0 & I \end{array}\right].$$

It turns out that such a Z satisfies the relation $Z\tilde{\Delta}Z^T = \tilde{\Delta}$ for every $\Delta \in \mathcal{D}$.

Using the above facts, we can show that if condition (13) is true for the original LFR model F, L, R, D with appropriate S, T, G, then it is also true for the scaled LFR obtained using any scaling matrix Z such as that above, for appropriate matrices $\tilde{S}$, $\tilde{G}$, $\tilde{T}$. That is, the condition is independent of the scaling Z.

In this sense, the conditions we obtained are independent of the LFR used to represent the perturbation structure.

Acknowledgments. This paper has benefitted from many stimulating discussions with several colleagues, including Aharon Ben-Tal, Stephen Boyd, Arkadii Nemirovski, Michael Overton, and particularly, Lieven Vandenberghe (who pointed out a mistake just before the final version was sent). Last but not least, the authors would like to thank the editor and reviewers for their very helpful comments and revisions.

REFERENCES

[1] F. Alizadeh, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.

[2] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results*, SIAM J. Optim., 8 (1998), pp. 746–768.

[3] H. Attouch, *Viscosity Solutions of Optimization Problems*, Tech. Report 07, Dépt. des Sciences Mathématiques, Université Montpellier 2, France, 1994.

[4] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski, *Robust semidefinite programming*, in Semidefinite Programming and Applications, to appear.

[5] A. Ben-Tal and A. Nemirovski, *Robust Convex Programming*, Tech. Report 1/95, Optimization Laboratory, Faculty of Industrial Engineering and Management, Technion, Israel Institute of Technology, Technion City, Haifa 32000, Israel, 1995; Math. Oper. Res., to appear.

[6] A. Ben-Tal and A. Nemirovski, *Robust Solutions to Uncertain Linear Programs*, Tech. Report 6/95, Optimization Laboratory, Faculty of Industrial Engineering and Management, Technion, Israel Institute of Technology, Technion City, Haifa 32000, Israel, 1995; Oper. Res. Lett., to appear.

[7] A. Ben-Tal and A. Nemirovski, *Robust truss topology design via semidefinite programming*, SIAM J. Optim., 7 (1997), pp. 991–1016.

[8] A. Ben-Tal and A. Nemirovski, *Robust convex programming*, IMA J. Numer. Anal., 1998, to appear.

[9] B. Bereanu, *Some Numerical Methods in Stochastic Programming Under Risk and Uncertainty*, Academic Press, New York, 1980, Ch. 11, pp. 169–205.

[10] J. F. Bonnans, R. Cominetti, and A. Shapiro, *Sensitivity Analysis of Optimization Problems under Second Order Regular Constraints*, Tech. Report 2989, INRIA, 1996; Math. Oper. Res., to appear.

[11] S. Boyd and L. El Ghaoui, *Method of centers for minimizing generalized eigenvalues*, Linear Algebra and Appl., special issue on Numerical Linear Algebra Methods in Control, Signals and Systems, 188 (1993), pp. 63–111.

[12] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, Studies in Applied Mathematics series, SIAM, Philadelphia, PA, 1994.

[13] S. P. Boyd and L. Vandenberghe, *Introduction to convex optimization with engineering applications*, lecture notes for ee392x, Stanford University, Stanford, CA, 1996. Available via anonymous ftp at `isl.stanford.edu/pub/boyd`.

[14] S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed, *An efficient algorithm for a bounded errors-in-variables model*, SIAM J. Matrix Anal. Appl., to appear.

[15] M. Dempster, *Stochastic Programming*, Academic Press, New York, 1980.

[16] L. El Ghaoui and H. Lebret, *Robust solutions to least-squares problems with uncertain data*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1035–1064.

[17] M. K. H. Fan, A. L. Tits, and J. C. Doyle, *Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics*, IEEE Trans. Automat. Control, 36 (1991), pp. 25–38.

[18] J. W. Helton and H. J. Woerdeman, *Symmetric Hankel operators: Minimal norm extensions and eigenstructures*, Linear Algebra Appl., 185 (1993), pp. 1–19.

[19] M. Kojima, M. Shida, and S. Shindoh, *Global and Local Convergence of Predictor-Corrector-Interior-Point Algorithm for Semidefinite Programming*, Tech. Report B-308, Dept. of Information Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguru-ku, Tokyo 152, Japan, 1995.

[20] H. Lebret, *Synthèse de diagrammes de réseaux d'antennes par optimisation convexe*, Ph.D. thesis, Université de Rennes I, Nov. 1994.

[21] C. Lemaréchal, F. Oustry, and C. Sagastizábal, *The $\mathcal{U}$-Lagrangian of a convex function*, Trans. Amer. Math. Soc., to appear.

[22] A. S. Lewis and M. L. Overton, *Eigenvalue optimization*, Acta Numerica, 5 (1996), pp. 149–190.

[23] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, *Second-order cone programming: Interior-point methods and engineering applications*, Linear Algebra Appl., submitted.

[24] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, Studies in Applied Mathematics series, SIAM, Philadelphia, PA, 1994.

[25] F. Oustry, *The $\mathcal{U}$-Lagrangian of the maximum eigenvalue function*, SIAM J. Optim., to appear.

[26] F. A. Potra and R. Sheng, *On homogeneous interior-point algorithms for semidefinite programming*, Optim. Methods Softw., 9 (1998), pp. 161–184.

[27] S. Robinson, *Stability theorems for systems of inequalities, part* II: *Differentiable nonlinear systems*, SIAM J. Numer. Anal., 13 (1976), pp. 497–513.

[28] R. T. Rockafellar, *Convex Analysis*, 2nd ed., Princeton Univ. Press, Princeton, NJ, 1970.

[29] R. Rubinstein and A. Shapiro, *Discrete Event Systems*, John Wiley, New York, 1993.

[30] A. Shapiro, *Perturbation theory of nonlinear programs when the set of optimal solutions is not a singleton*, Appl. Math. Optim., 18 (1988), pp. 215–229.

[31] A. Shapiro, *Perturbation analysis of optimization problems in Banach spaces*, Numer. Funct. Anal. Optim., 13 (1992), pp. 97–116.

[32] A. Shapiro, *First and second order analysis of nonlinear semidefinite programs. Semidefinite programming*, Math. Programming Ser. B, 77 (1997), pp. 301–320.

[33] A. Shapiro and M. K. H. Fan, *On eigenvalue optimization*, SIAM J. Optim., 5 (1995), pp. 552–568.

[34] A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems*, John Wiley, New York, 1977.

[35] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

[36] V. A. Yakubovich, *The solution of certain matrix inequalities in automatic control theory*, Soviet Math. Dokl., 3 (1962), pp. 620–623.

[37] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice–Hall, Upper Saddle River, NJ, 1995.

SIAM J. OPTIM.
Vol. 9, No. 1, pp. 53–61

STOCHASTIC SIMULATION ON INTEGER CONSTRAINT SETS[*]

I. H. DINWOODIE[†]

Abstract. Bounds are given on the number of steps sufficient for convergence of simulation algorithms on domains of nonnegative integer constraint sets.

Key words. Markov chains, eigenvalues, annealing, integer optimization

AMS subject classifications. 60J20, 65K05, 90C10

PII. S1052623496313842

1. Introduction.

This article is concerned with convergence of Markov chains on nonnegative integer constraint sets and applications to simulated annealing algorithms for optimization.

Despite the lack of applicable results on its performance, the annealing algorithm is used for optimization of nonlinear functions on discrete domains. One application of the algorithm is finding modes of probability distributions on finite sets, a problem which arises in Bayesian statistics and image analysis (see [6] and [14]). It is used for other problems in combinatorial optimization as well, some of which are described in [13]. Here we are interested in domains of nonnegative integer lattice points on hyperplanes, which arise in integer optimization problems, image analysis, and statistics. Symmetric Markov chains on these domains were constructed in [4] using algebraic techniques.

Whereas optimal cooling schedules have been widely studied [1], [7], [8], [9], we are interested in establishing clear bounds on the time required for a given accuracy $\delta > 0$ and reliability $\varepsilon > 0$. The main result is Theorem 3.1, which gives a sufficient number of steps in the algorithm in a computable form. We expect that the results can be improved as new technology in Markov chains becomes available. They are based on geometrical techniques from [3] and [9], which also are used in the more general and abstract study [10].

Let us establish some notation. Let $\mu > 0$ be a probability distribution function on the set $S = \{\mathbf{x} \in Z_+^d, A(\mathbf{x}) = \mathbf{b} \in Z_+^r\}$, where A is a linear map or matrix with nonnegative integer entries such that S is finite. We show first in section 2 how to simulate from μ using techniques from [4], then we get convergence rates from eigenvalue estimates and apply these results in section 3 to the case where μ is chosen to put most of its mass where f is small. Our Markov chain is homogeneous, which means that the parameter β corresponding to the reciprocal of temperature is held fixed over time at a level which gives the desired stationary distribution.

2. The algorithm.

We define a symmetric Markov chain on S as follows. Let $Q(\xi_1, \ldots, \xi_d)$ be the ring of polynomials in variables $\xi_1, \ldots, \xi_d$ with coefficients in the rational numbers Q. If $\mathbf{x}$ is a vector of nonnegative integers, define $X^{\mathbf{x}} = \xi_1^{x_1} \cdots \xi_d^{x_d}$.

Let $M = \{g_1, \ldots, g_m\}$, where $g_i \in Z^d$ and $\{X^{g_i^+} - X^{g_i^-} : i = 1, \ldots, m\}$ is a Gröbner basis for the ideal I_A in $Q(\xi_1, \ldots, \xi_d)$ given by $I_A = \langle X^{\mathbf{v}} - X^{\mathbf{w}} : A(\mathbf{v}) =$

[*] Received by the editors December 18, 1996; accepted for publication (in revised form) January 5, 1998; published electronically October 30, 1998.

http://www.siam.org/journals/siopt/9-1/31384.html

[†] Department of Mathematics, Tulane University, New Orleans, LA 70118 (ihd@math.tulane.edu).

$A(\mathbf{w})\rangle$. Recall that $g = g^+ - g^-$, where $g^+ = \max\{g, 0\}$, $g^- = \max\{-g, 0\}$. Our ordering on monomials is purely lexicographic, based on the indeterminate ordering $\xi_1 > \xi_2 > \cdots > \xi_d$.

Now define a Markov chain on S as follows. Let $\mathbf{x} = (x_1, \ldots, x_d)$ denote an element of S, and let $N_\mathbf{x}$ be the set of its neighbors in S, so $N_\mathbf{x} = \{\mathbf{x} \pm g_i : \mathbf{x} \pm g_i \geq 0, g_i \in M\}$. Let $K(\mathbf{x}, \cdot)$ be the probability vector

$$K(\mathbf{x}, \mathbf{y}) = \frac{1}{2m} \qquad\qquad \text{for } \ \mathbf{y} \in N_\mathbf{x}.$$

Also, $K(\mathbf{x}, \mathbf{x}) = 1 - \sum_{\mathbf{y} \in N_\mathbf{x}} K(\mathbf{x}, \mathbf{y})$ makes the vector sum to 1 and will be positive precisely when $|N_\mathbf{x}| < 2m$. The transition can be realized by uniformly choosing an element $\pm g_i$ from among the $2m$ choices $\{\pm g_1, \ldots, \pm g_m\}$ and adding it to $\mathbf{x}$ if the result is nonnegative. Then K is symmetric, irreducible, and aperiodic.

Recall that $\mu > 0$ is an arbitrary distribution on S. Let K_μ be the transition matrix given by

$$K_\mu(\mathbf{x}, \mathbf{y}) = K(\mathbf{x}, \mathbf{y}) \min\{\mu(\mathbf{y})/\mu(\mathbf{x}), 1\}, \qquad\qquad \mathbf{x} \neq \mathbf{y},$$

and the holding probability makes the matrix stochastic. Let the spectrum of K_μ be $1 = \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_{|S|} > -1$, and let $\gamma = 1 - \lambda_2$. In the next section, μ will depend on a parameter $\beta > 0$ interpreted as the reciprocal of temperature.

To estimate γ, observe that

$$\begin{aligned} \gamma &= \inf_\phi \frac{\langle \phi, (I - K_\mu)\phi \rangle_\mu}{\|\phi - \langle \phi, \mathbf{1} \rangle_\mu \mathbf{1}\|_\mu^2} \\ &= \inf_\phi \frac{\sum\limits_{(x,y)} (\phi(x) - \phi(y))^2 \mu(x) K_\mu(x, y)}{\sum\limits_{(x,y)} (\phi(x) - \phi(y))^2 \mu(x)\mu(y)} \\ &= \inf_\phi \frac{\sum\limits_{e \in G} \phi(e)^2 Q(e)}{\sum\limits_{f \in G_c} \phi(f)^2 Q_c(f)}, \end{aligned}$$

where ϕ ranges through nonconstant functions on the state space S, G is the graph with edge $e = \{x, y\}$ if and only if $Q(e) = \mu(x) K_\mu(x, y)$ $(= \mu(y) K_\mu(y, x)) > 0$, and G_c is the complete graph with edges $f = \{x, y\}$ connecting all ordered pairs, $Q_c(f) = \mu(x)\mu(y)$, and $\phi(f)^2 = (\phi(y) - \phi(x))^2$. Now this representation can be used as in [3] and [9] to bound $1/\gamma$ from above in the form of a Poincaré inequality, which we use in Lemma 2.1 below.

Define the following quantities. Let $\mu_{\max} = \max\{\mu(\mathbf{x})\}$, $\mu_{\min} = \min\{\mu(\mathbf{x})\}$, $\rho = \mu_{\max}/\mu_{\min}$. The following eigenvalue estimate uses a result of [3], which is implicit in [9]. Recall that m is the number of moves in the set M.

LEMMA 2.1. *Let* $\gamma = 1 - \lambda_2$. *Then* $1/\gamma \leq m\rho\mu_{\max}|S|^3$.

Proof. If $\mathbf{x} > \mathbf{y}$ are two points in S ordered lexicographically, form a path from $\mathbf{x}$ to $\mathbf{y}$ by dividing the monomial difference $X^\mathbf{x} - X^\mathbf{y}$ by the Gröbner basis $\{X^{g_i^+} - X^{g_i^-} : 1 \leq i \leq m\}$, ordered in some arbitrary but fixed way. The multidegrees of the lead terms in the division give a path $p_{\mathbf{xy}}$ which joins the endpoints $\mathbf{x}$ and $\mathbf{y}$ with decreasing (in lexicographic order) path segments to a common point in S. The number of edges in this path $p_{\mathbf{xy}}$ is no greater than $\#\mathbf{x} - 1$, if $\#\mathbf{x}$ is the rank of $\mathbf{x}$ in the set S using lexicographic order (the smallest element of S in lexicographic order has rank 1, the largest has rank $|S|$). Define the measure of length $|p_{\mathbf{xy}}|_Q = \sum_{e \in p_{\mathbf{xy}}} Q(e)^{-1}$.

Proposition 1 of [3] shows that $1/\gamma \le \max_e \sum_{p_{\mathbf{xy}} \ni e} |p_{\mathbf{xy}}|_Q \, \mu(\mathbf{x})\mu(\mathbf{y})$, where $p_{\mathbf{xy}}$ is the path from $\mathbf{x}$ to $\mathbf{y}$ constructed above and e is an edge in the graph on S, say $\{\mathbf{a}, \mathbf{a} \pm g_\alpha\}$, with $Q(e) = \mu(\mathbf{a})K_\mu(\mathbf{a}, \mathbf{a} \pm g_\alpha) = \mu(\mathbf{a})K(\mathbf{a}, \mathbf{a} \pm g_\alpha) \min\{\mu(\mathbf{a} \pm g_\alpha)/\mu(\mathbf{a}), 1\}$. Then $Q(e)^{-1} \le \mu_{\min}^{-1}(2m)$.

For $\mathbf{x} \in S$, the maximum number of edges in a path joining $\mathbf{x}$ to $\mathbf{y} < \mathbf{x}$ is $\#\mathbf{x} - 1$, so $|p_{\mathbf{xy}}|_Q \le (\#\mathbf{x} - 1)2m/\mu_{\min}$. With $e = \{\mathbf{a}, \mathbf{a} - g_\alpha\}$, the paths through e can be partitioned into collections starting at the different maximal values $\mathbf{x} \ge \mathbf{a}$. There can be at most $\#\mathbf{x} - 1$ paths starting from $\mathbf{x} \ge \mathbf{a}$ and going to some point less than $\mathbf{x}$, and the length of each is at most $\#\mathbf{x} - 1$. Thus

$$\begin{aligned}
\sum_{p_{\mathbf{xy}} \ni e} |p_{\mathbf{xy}}|_Q \, \mu(\mathbf{x})\mu(\mathbf{y}) &\le 2m/\mu_{\min} \sum_{\mathbf{x} \ge \mathbf{a}} (\#\mathbf{x} - 1)^2 {\mu^2}_{\max} \\
&\le 2m \left({\mu^2}_{\max}/\mu_{\min}\right) \sum_{\mathbf{x}} (\#\mathbf{x} - 1)^2 \\
&= 2m \left({\mu^2}_{\max}/\mu_{\min}\right) (|S| - 1)|S|(2|S| - 1)/6 \\
&\le m\rho\mu_{\max}|S|^3. \qquad \square
\end{aligned}$$

The main contribution of the bound in Lemma 2.1 is the estimate for γ when μ is uniform. Since the paths in the estimate joining two points do not change with μ, they cannot yield the optimal result for a particular μ (see [9]). On the other hand, the technique gives a practical estimate of the path length $|p_{\mathbf{xy}}|_Q$. In some examples, one can find a shorter path by going through intermediate points and using various orderings on the variables $\xi_1, \dots, \xi_d$ for different path segments. This applies in particular to Example 3.3.

Finally let T_μ be the Markov chain obtained by running K_μ a Poisson(1) number of steps each time to avoid complications from negative eigenvalues. If we let $\Delta = K_\mu - I$ be the generator for K_μ, we can define

$$T_\mu = e^{\Delta}, \tag{2.1}$$

which is reversible with stationary distribution μ. The spectrum of e^{Δ} is $\{\exp(\lambda_i - 1), i = 1, \dots, |S|\}$.

Recall that any reversible and irreducible kernel T with stationary distribution $\mu > 0$ implies that multiplying the matrix T with a column vector on the right gives a self-adjoint operator on $L^2(\mu)$, which leaves invariant $\mathbf{1}^\perp$, and multiplying T with a row vector on the left gives a self-adjoint operator on $L^2(1/\mu)$, which leaves invariant $\mu^\perp$. If $\langle \cdot, \cdot \rangle_{1/\mu}$ denotes the inner product in $L^2(1/\mu)$,

$$\begin{aligned}
|T^n(\mathbf{x}, A) - \mu(A)| &= \langle (\delta_{\mathbf{x}} - \mu)T^n, (I_A - \mu(A))\mu \rangle_{1/\mu} \\
&\le \|\delta_{\mathbf{x}} - \mu\|_{1/\mu} \|T^n\|_{1/\mu} \|(I_A - \mu(A))\mu\|_{1/\mu}
\end{aligned}$$

and $\|\delta_{\mathbf{x}} - \mu\|_{1/\mu} \le 1/\sqrt{\mu(\mathbf{x})}$, $\|(I_A - \mu(A))\mu\|_{1/\mu} \le 1/2$, $\|T^n\|_{1/_\mu}$ is its largest eigenvalue as an operator restricted to $\mu^\perp$, which in our situation is $\exp(-\gamma)$.

For two distributions μ and ν on S, define the total variation distance $\|\nu - \mu\| = \sup_{A \subset S} |\nu(A) - \mu(A)|$.

PROPOSITION 2.1. *Let* $\varepsilon > 0$, *and let* T_μ *be defined in* (2.1). *Then* $\|T_\mu^n(\mathbf{x}, \cdot) - \mu\| \le \varepsilon$ *for all* $n \ge -m\rho\mu_{\max}|S|^3 \log(\varepsilon\sqrt{\mu(\mathbf{x})})$.

Proof. Since $\|T_\mu^n(\mathbf{x}, \cdot) - \mu\| \le 1/\sqrt{\mu(\mathbf{x})} \exp(-n\gamma)$ (see [3]), it is sufficient that $n \ge -\log(\varepsilon\sqrt{\mu(\mathbf{x})})/\gamma$, which follows if $n \ge -\log(\varepsilon\sqrt{\mu(\mathbf{x})})(m\rho\mu_{\max}|S|^3)$. $\square$

3. Application to simulated annealing. Let $f : S \to \mathbf{R}$, and let $f_{\min} = \min\{f(\mathbf{x}) : \mathbf{x} \in S\} = f(\mathbf{x}_*)$, $f_{\max} = \max\{f(\mathbf{x}) : \mathbf{x} \in S\}$, and set $h = f_{\max} - f_{\min}$. Our optimization problem is to minimize f. Let μ_β be the Gibbs measure on S given by $\mu_\beta(\mathbf{x}) = e^{-\beta f(\mathbf{x})}|S|^{-1}/\phi(-\beta)$, where ϕ is the moment generating function for f with respect to the uniform distribution on S:

$$\phi(t) = \sum_{\mathbf{x}} \frac{e^{tf(\mathbf{x})}}{|S|}. \tag{3.1}$$

Also, let $\phi_\beta(t) = E_\beta e^{tf} = \phi(t-\beta)/\phi(-\beta)$. Let $(\log\phi)^*$ denote the convex conjugate of the function $\log(\phi)$, given by $(\log\phi)^*(a) = \sup_{t\in\mathbf{R}}\{ta - \log\phi(t)\}$. Recall that μ_β converges, as β gets large, to the uniform distribution on the points where f attains its minimum, which we make precise in Lemma 3.1 below.

LEMMA 3.1. *Let $\delta > 0$, let $\varepsilon > 0$, and set $a = f_{\min} + \delta$. Let $\beta > 0$ be sufficiently large that both $E_\beta(f) < a$ and $a(-\beta) - \log\phi(-\beta) \le \log(\varepsilon) + (\log\phi)^*(a)$. Then $\mu_\beta(\{\mathbf{x} : f(\mathbf{x}) > a\}) \le \varepsilon$.*

Proof. Clearly $\mu_\beta(\{\mathbf{x} : f(\mathbf{x}) > a\}) \le e^{-ta}E_\beta e^{tf} = e^{-ta}\phi_\beta(t)$ for all $t \ge 0$, so

$$\begin{aligned}\log\mu_\beta(\{\mathbf{x} : f(\mathbf{x}) > a\}) &\le -\sup_{t\ge 0}\{ta - \log\phi_\beta(t)\}\\ &= -\sup_{t\in\mathbf{R}}\{ta - \log\phi_\beta(t)\} \qquad \text{(since } a > E_\beta(f))\\ &= -\sup_{t}\{(t-\beta)a - \log\phi(t-\beta)\} - a\beta - \log\phi(-\beta)\\ &= -(\log\phi)^*(a) - \beta a - \log\phi(-\beta).\end{aligned}$$

Let $\beta > 0$ be sufficiently large that $E_\beta(f) < a$ and $a(-\beta) - \log\phi(-\beta) \le \log(\varepsilon) + (\log\phi)^*(a)$. Then the result follows. □

Let $\mathbf{x} = X_0, X_1, \ldots$ be the Markov chain defined in (2.1) with stationary distribution μ_β, transition matrix $T_\beta = e^{\Delta}$ with $\Delta = I - K_\beta$, and probability measure $P_{\mathbf{x}}^\beta$ on the sample space. Note that $\max\{\mu_\beta(\mathbf{x})\} = \mu_\beta(\mathbf{x}_*) = \mu_{\beta,\max}$.

THEOREM 3.1. *Let $a = f_{\min} + \delta$ for $\delta > 0$, and let $\beta > 0$ be sufficiently large that $E_\beta(f) < a$ and $a(-\beta) - \log\phi(-\beta) \le \log(\varepsilon) + (\log\phi)^*(a)$. Then $P_{\mathbf{x}}^\beta\{f(X_n) > a\} \le 2\varepsilon$ for all $n \ge -m\mu_{\beta,\max}e^{\beta h}|S|^3\log(\varepsilon\sqrt{\mu_\beta(\mathbf{x})})$.*

Remark. For the uniform distribution, the result simplifies to

$$n \ge -m|S|^2\log(\varepsilon/\sqrt{|S|}).$$

Proof. First,

$$\begin{aligned}P_{\mathbf{x}}^\beta\{f(X_n) > a\} &= P_{\mathbf{x}}^\beta\{X_n \in f^{-1}(a,\infty)\} - \mu_\beta(f^{-1}(a,\infty)) + \mu_\beta(f^{-1}(a,\infty))\\ &\le \|T_\beta^n(\mathbf{x},\cdot) - \mu_\beta\| + \mu_\beta(f^{-1}(a,\infty)).\end{aligned}$$

By Proposition 2.1, $\|T_\beta^n(\mathbf{x},\cdot) - \mu_\beta\| \le \varepsilon$ if $n \ge -m\mu_{\beta,\max}e^{\beta h}|S|^3\log(\varepsilon\sqrt{\mu_\beta(\mathbf{x})})$, since the parameter $\rho = \max\{\mu_\beta\}/\min\{\mu_\beta\} = \exp(\beta(f_{\max} - f_{\min})) = \exp(\beta h)$. In addition, $\mu_\beta(f^{-1}(a,\infty)) \le \varepsilon$ if β is sufficiently large (depending on δ and ε) by Lemma 3.1. □

Theorem 3.1 suggests the following. If $\varepsilon = 1/2$ and the Markov chain starts at a point $\mathbf{x}$ not far from $\mathbf{x}_*$ so that $\mu_\beta(\mathbf{x}) \approx \mu_\beta(\mathbf{x}_*) \approx 1$, then a sufficient number of iterations is roughly $me^{\beta h}|S|^3$. This does not give the exact exponential rate of growth in β of [9], but the bound leaves no unspecified constants and does not rely on detailed knowledge of the function f.

It may be simpler and faster to get the same reliability by running several independent chains with moderate values of β and observing the minimum over all these chains.

EXAMPLE 3.1. *Let $S = \{\mathbf{x} \in Z_+^2 : x_1 + x_2 = k - 1\}$ for an integer $k > 0$. Then $|S| = k$. Let $\beta = 0$, which corresponds to infinite temperature. Then μ_β is the uniform distribution on $S = \{(0, k-1), \dots, (k-1, 0)\}$. A Gröbner basis consists of the difference $x_1 - x_2$ which corresponds to the vector $g_1 = (1, -1)$, and the Markov chain K_0 is essentially a reflecting random walk with second largest eigenvalue* $\cos(\pi/k)$ [5, p. 389]. *Then the chain $T_0 = e^{\Delta}$ has the second largest eigenvalue* $\exp(\cos(\pi/k) - 1) \approx \exp(-(\pi^2/2)/k^2)$. *On the other hand, the bound of Lemma* 2.1 *gives $\gamma \geq |S|^{-2} = k^{-2}$, so the second eigenvalue* $\exp(-\gamma) \leq \exp(-1/k^2)$, *which is not an unreasonable bound.*

For $\beta > 0$ and the objective function f, Theorem 3.1 says that a sufficient number of steps is $e^{\beta h} k^2 (k\mu_{\beta,\max}) \log(\varepsilon\sqrt{\mu_\beta(\mathbf{x})})$. The dependence of β on the desired $\delta > 0$ and $\varepsilon > 0$ is explained in Lemma 3.1 and requires some estimates of the moment generating function ϕ.

EXAMPLE 3.2. *Consider the knapsack problem* [11, p. 14] *with general increasing utility functions $f_i : \mathbf{R}_+ \to \mathbf{R}_+, f_i(0) = 0$. The problem is to maximize $f_1(q_1) + \cdots + f_d(q_d)$, where q_i represents a nonnegative integer quantity of object i, subject to the overall weight constraint $\langle \mathbf{w}, \mathbf{q} \rangle \leq W$, where $\mathbf{w} = (w_1, \dots, w_d)$ is a vector of positive integer weights and W is a positive integer.*

Add a slack variable x_{d+1} so that its corresponding indeterminate ξ_{d+1} is less than ξ_d. This ordering is important to get a simple Gröbner basis. For $\mathbf{x} = (x_1, \dots, x_d, x_{d+1}) \in Z_+^{d+1}$, let $f(\mathbf{x}) = -(f_1(x_1) + \cdots + f_d(x_d))$. Then the problem becomes one to minimize $f(\mathbf{x})$ over S, where $S \subset Z_+^{d+1}$ is defined by $w_1 x_1 + \cdots + w_d x_d + x_{d+1} = W$.

To apply Theorem 3.1, observe that $h = f_{\max} - f_{\min} = 0 + \max\{f_1(q_1) + \cdots + f_d(q_d)\} \leq \sum_{i=1}^d f_i(W/w_i)$. Also, $|S|$ is the coefficient on z^W in the generating function

$$g(z) = \frac{1}{1-z} \prod_{i=1}^{d} \frac{1}{1 - z^{w_i}},$$

which is easily computed and satisfies $|S| \leq (W + d)^d / d!$. By Schur's theorem [17, p. 90], $|S|$ is well approximated by $W^d / (d! w_1 \dots w_d)$ when W is large compared to d.

That a set of d moves on S from a Gröbner basis is, in general, easy to describe. Each corresponds to incrementing a coordinate 1 through d by ±1 and adjusting the slack variable accordingly to maintain the equation $w_1 x_1 + \cdots + w_d x_d + x_{d+1} = W$ (cf. Proposition 3.1).

Theorem 3.1 says that a sufficient number of steps for the uniform distribution ($\beta = 0$) is roughly $d|S|^2 \log(\varepsilon/\sqrt{|S|})$, or essentially $dW^{2d}/(d!^2 w_1^2 \dots w_d^2)$, if we ignore the logarithm. For $\beta > 0$, the quantity is roughly $de^{\beta h} W^{3d} \log(\varepsilon)/(d!^3 w_1^3 \cdots w_d^3)$. This quantity is polynomial in W for fixed d but is not polynomial in both W and d. The Ibarra–Kim theorem [15, p. 262] indicates that when the objective function is linear, there exists a dynamic programming "approximation" algorithm that is polynomial in both d and W, which would in theory be superior. The annealing method has the advantage in terms of generality, since it can easily be applied with any objective function. Neither the annealing algorithm nor the exact dynamic programming approach is fully polynomial in all the parameters.

In general, inequality constraints such as in the knapsack problem can be treated quite simply. Consider the situation where a state space S_0 is the intersection of a finite number of half-spaces, say $S_0 = \{x \in Z_+^d : A\mathbf{x} \leq \mathbf{b}\}$, where $\mathbf{b} \in Z_+^r$ and the nonnegative integer matrix A of rank r is such that $|S_0| < \infty$. This includes the knapsack problem of Example 3.3. By adding r slack variables, S_0 is equivalent to a finite state space S in Z_+^{d+r} with equality constraints. For the algebra, we need r new indeterminates ψ_i, which we order $\xi_1 > \cdots > \xi_d > \psi_1 > \cdots > \psi_r$. A Gröbner basis for the appropriate ideal is the generating set $\{\xi_i - \Psi^{A(\mathbf{e}_i)}, 1 \leq i \leq d\}$, with $\mathbf{e}_i$ the basis element for $\mathbf{R}^d$, since the S-polynomials leave remainder 0 when divided by this set. These polynomials correspond to d moves given by incrementing or decrementing each of the original coordinates, chosen uniformly from the d choices, and adjusting the slack variables accordingly. This is then a special case of the Markov chain described in [2] for uniform generation within an arbitrary convex set of lattice points in Z^d.

A path between points $\mathbf{x}$ and $\mathbf{y}$ can be constructed by moving within S_0 along the edges of a d-dimensional rectangle as follows. Let $\mathbf{z} = \min\{\mathbf{x}, \mathbf{y}\}$, which belongs to S_0. Join $\mathbf{x}$ to $\mathbf{z}$ by decrementing the first coordinate, then the second, etc. Then join $\mathbf{y}$ to $\mathbf{z}$ in the same manner. The two segments form a path from $\mathbf{x}$ to $\mathbf{y}$ within S_0. This path can be somewhat shorter than the one that arises from the division algorithm. The number of steps in the path $p_{\mathbf{xy}}$ is $\Sigma_i |x_i - y_i| \leq 2\max\{\Sigma_i u_i : \mathbf{u} \in S_0\}$. Let $a_i - 1$ bound the ith coordinate of elements in S_0, so $\max\{x_i : (x_1, \ldots, x_d) \in S_0\} \leq a_i - 1, i = 1, \ldots, d$.

Proposition 3.1. *Let S_0 be as above with $D = \max\{\Sigma_i x_i : \mathbf{x} \in S_0\}$, and let $A = \prod_{i=1}^d a_i$. Let T_β be the Markov transition operator on S_0 (or equivalently S) defined in* (2.1). *Then $\|T_\beta^n(\mathbf{x}, \cdot) - \mu_\beta\| \leq \varepsilon$ for all $n \geq -d2^{d+1}e^{\beta h}\mu_{\beta,\max} AD^2 \log(\varepsilon\sqrt{\mu_\beta(\mathbf{x})})$.*

Remark. When $\beta = 0, \mu_{\beta,\max} = |S|^{-1}$, so the bound is roughly

$$-d2^{d+1}D^2(A/|S|)\log(\varepsilon/\sqrt{|S|})$$

for the uniform distribution. For a rectangular region A, $A/|S| = 1$, and this specializes to a quantity on the order of $d2^{d+1}D^2$. D measures the diameter of S_0 in the L^1 sense, so this is consistent with the results of [2], which suggest that for most convex regions in dimension d, the number of steps required is on the order of the squared diameter. The bounds of Theorem 3.1 and Proposition 3.1 are comparable when $|S|$ is comparable to D.

Proof. First we estimate the gap $\gamma = 1 - \lambda_2$ for K_β. With Proposition 1′ of [3], we see that $1/\gamma \leq \max_e Q(e)^{-1}\Sigma_{p_{\mathbf{xy}} \ni e}|p_{\mathbf{xy}}|\mu_\beta(\mathbf{x})\mu_\beta(\mathbf{y})$, where $|p_{\mathbf{xy}}| \leq 2D$ denotes the number of edges in the path joining points $\mathbf{x}$ and $\mathbf{y}$. Now $Q(e)^{-1} \leq 2d/\mu_{\beta,\min}$. To bound $\Sigma_{p_{\mathbf{xy}} \ni e}1$, consider the edge e joining vertices $\mathbf{u} = (u_1, u_2, \ldots, u_i, \ldots, u_d)$ and $\mathbf{u} - \mathbf{e}_i$ (here $\mathbf{e}_i$ is the ith basis element in $\mathbf{R}^d$). We need to count first all the ordered pairs $(\mathbf{x}, \mathbf{y}) \in S_0 \times S_0$ such that their connecting path can traverse e. Then half this number will bound the number of unordered pairs.

The edge e is either on the path from $\mathbf{x}$ down to $\mathbf{z} = \min\{\mathbf{x}, \mathbf{y}\}$ or from $\mathbf{y}$ down to $\mathbf{z}$. Since the ith coordinate is changed along the edge e, the coordinates $1, \ldots, i-1$ remain fixed throughout the remaining part of the path. Thus $u_j = \min\{x_j, y_j\}, j = 1, \ldots, i-1$. Also, since coordinates $i+1, \ldots, d$ have not yet been visited, $u_j = x_j$ for $j = i+1, \ldots, d$, or $u_j = y_j$ for $j = i+1, \ldots, d$, depending on whether the edge is in the segment from $\mathbf{x}$ to $\mathbf{z}$ or from $\mathbf{y}$ to $\mathbf{z}$.

Partition the pairs $(\mathbf{x}, \mathbf{y})$, whose connecting path goes through e into 2^{i-1} groups, where a group is identified with a sequence of 0's and 1's of length $i-1$, and a 0 in the jth place indicates $x_j \geq y_j$, whereas a 1 indicates $x_j < y_j, 1 \leq j \leq i-1$. Now the size of each of these groups is at most $2a_1a_2\cdots a_{i-1}a_iDa_{i+1}\cdots a_d = 2AD$ as follows. For a particular sequence of 0's and 1's, the possible values of $(\mathbf{x}, \mathbf{y})$ are constrained so that $x_j = u_j$ for indices $j \leq i-1$ where there is a 1 (indicating $x_j < y_j$, and thus $x_j = \min(x_j, y_j) = u_j$), and similarly $y_j = u_j$ for indices $j \leq i-1$ where there is a 0 (indicating $y_j \leq x_j$, and thus $y_j = \min(x_j, y_j)$). For indices $i < j \leq d$, either $(x_{i+1}, \ldots, x_d) = (u_{i+1}, \ldots, u_d)$ and the y coordinates are unclear, or $(y_{i+1}, \ldots, y_d) = (u_{i+1}, \ldots, u_d)$ and the x coordinates are unclear. Finally, the ith coordinates in both $\mathbf{x}$ and $\mathbf{y}$ can take at most a_i and also at most $\max\{x_i : \mathbf{x} \in S_0\} + 1 \leq D + 1$ values, so the number of such pairs is at most $2(a_1a_2\cdots a_{i-1} \times a_iD \times a_{i+1}\cdots a_d)$.

Then summing over all 2^{i-1} groups and dividing by 2 to get unordered pairs $\{\mathbf{x}, \mathbf{y}\}$ we get $\Sigma_{p_{\mathbf{xy}} \ni e} 1 \leq 2^{i-1}(2AD/2) \leq 2^{d-1}AD$.

Therefore

$$1/\gamma \leq (2d/\mu_{\beta,\min})\mu_{\beta,\max}^2 2D(2^{d-1}AD) = d\rho\mu_{\beta,\max}2^{d+1}AD^2 = d2^{d+1}e^{\beta h}\mu_{\beta,\max}AD^2.$$

Again using the basic bound $\|T_\beta^n(\mathbf{x}, \cdot) - \mu_\beta\| \leq 1/\sqrt{\mu_\beta(\mathbf{x})}\exp(-n\gamma)$, we see that $\|T_\beta^n(\mathbf{x}, \cdot) - \mu_\beta\| \leq \varepsilon$ for $n \geq -\log(\varepsilon\sqrt{\mu_\beta(\mathbf{x})})/\gamma$, which occurs if

$$n \geq -d2^{d+1}e^{\beta h}\mu_{\beta,\max}AD^2 \log(\varepsilon\sqrt{\mu_\beta(\mathbf{x})}). \qquad \square$$

If we apply Proposition 3.1 to the knapsack problem of Example 3.2, we can let $a_i = (W/w_i) + 1$. Then A is approximately the product $W^d/(w_1\cdots w_d)$, and the quantity $|S| \approx W^d/(d!w_1 \ldots w_d) \approx A/d!$ for W large and fixed d. Since $D \leq W/\min_i w_i$, we get the estimate $d2^{d+1}W^2/(\min_i w_i)^2(A/|S|)$, or roughly the quantity $d2^{d+1}d!W^2/(\min_i w_i)^2$. For fixed d and weights w_i, this grows more slowly in W than the estimate $dW^{2d}/(d!^2w_1^2 \ldots w_d^2)$ from Example 3.2.

We mention finally that significant improvements are possible in situations where a certain block structure is present in the constraint set. Suppose that the constraint set $S \subset Z_+^d$ can be described by g independent constraints on disjoint sets of d_i variables, $1 \leq i \leq g$, as follows. Let $A_i : \mathbf{R}^{d_i} \to \mathbf{R}^{r_i}$ be a matrix with nonnegative integer entries which define a constraint set $S_i = \{\mathbf{x} \in Z_+^{d_i} : A_i(\mathbf{x}) = \mathbf{b}_i\}, i = 1, \ldots, g$. Then we assume that S has the product form $S = S_1 \times \cdots \times S_g$.

Let b_i index the start of the variables for constraint i, so $b_i = d_1 + \cdots + d_{i-1} + 1, i = 1, \ldots, g$. Let M_i of size m_i be the set of moves corresponding to a Gröbner basis for the symmetric Markov chain on each of these sets S_i, $i = 1, \ldots, g$, considered separately. Let K_i be the symmetric kernel on S_i given by

$$(3.2) \qquad K_i(\mathbf{x}, \mathbf{y}) = \frac{1}{2m_i} \qquad \text{if } \mathbf{y} \geq 0 \text{ and } \mathbf{y} = \mathbf{x} \pm f \text{ for some } f \text{ in } M_i,$$

and K_i vanishes otherwise. This means that one updates the d_i coordinates corresponding to the constraints A_i with moves from M_i. Take arbitrary positive weights $w_1, \ldots, w_g$ with $w_1 + \cdots + w_g = 1$, and form the irreducible kernel K on S by

$$(3.3) \qquad K(h_1 \otimes h_2 \cdots \otimes h_g)(\mathbf{x}) = \sum_{i=1}^{g} w_i h_1 \otimes \cdots \otimes K_i(h_i) \otimes \cdots \otimes h_g(\mathbf{x}),$$

which means that one chooses a block of coordinates i with probability w_i, and then one runs the chain K_i in S_i to update those d_i coordinates while leaving the others unchanged. The eigenvectors of K are the weighted averages of tensor products of those for the kernels $\{K_i, i = 1, \dots, g\}$, weighted by the family $\{w_i\}$, with eigenvalues being the corresponding weighted averages. It is then a simple fact that the gap γ for K, the difference between unity and the second largest eigenvalue, is $\gamma = \min\{w_i\gamma_i : i = 1, \dots, g\}$, where γ_i is one minus the second largest eigenvalue for the kernel K_i on the set S_i. Below we consider a special case with uniform distribution on one-dimensional blocks.

Example 3.3 (Reflecting random walk in a box [2]). *Consider the state space* $S_0 = \{\mathbf{x} \in Z_+^d : 0 \leq x_i \leq a-1\}$ *for an integer* $a > 1$. *The reflecting random walk makes a transition from a lattice point by uniformly choosing one of the* $2d$ *neighbors and moving to the one selected if the candidate is within the box. The boundary points have some positive holding probability. Another description is that one uniformly chooses one of the* d *dimensions or coordinates to update, then one runs the one-dimensional reflecting random walk one step on that coordinate in the space* $\{0, 1, \dots, a-1\}$.

The state space S_0 is equivalent for our purposes to $S = \{\mathbf{x} \in Z_+^{2d} : x_{i,1} + x_{i,2} = a - 1, i = 1, \dots, d\}$ by adding d slack variables. Applying Lemma 2.1 at this point to S yields a spectral estimate that is not accurate for $d \geq 2$. Recall that the exact gap $\gamma = (1 - \cos(\pi/a))/d \approx \pi^2/2a^2d$, which is on the order of the square of the euclidean diameter of S [2].

To get something comparable with our method, write $S = S_1 \times \cdots \times S_d$, where $S_i = \{\mathbf{x} \in Z_+^2 : x_{i,1} + x_{i,2} = a - 1\}$. The Markov chain on S_i is run with moves $M_i = \{(1, -1)\}$. With uniform weights $w_i = 1/d$, the kernel (3.3) is exactly the reflecting walk in the box, and Lemma 2.1 gives $\gamma_i \geq |S_i|^{-2} = a^{-2}$. Then K has gap at least $1/a^2d$ as the average of the kernels K_i, and this result is of the right size in the parameters a and d.

We conclude from this example that in situations where a product structure exists on the state space (and also on the objective function, which we have taken to be constant in this example) the bound of Lemma 2.1 can be significantly improved.

4. Conclusions. Nonnegative integer constraint sets are difficult and interesting domains for optimization and simulation problems. The results in this paper give general bounds on the time required for a given accuracy in some problems of simulation on such domains without prior enumeration of the state space. They are formulated in terms of computable quantities. In some interesting examples the bounds are quite accurate, in particular, when the state space S is low dimensional. An example where they would not be accurate would be simulation on the set of multigraphs with given vertex degrees, which can be interpreted as simulation on symmetric nonnegative integer matrices with certain row and column sums. Here the dimension is on the order of the number of vertices, and we would not expect the results to be useful in this case. The basic technique for estimating eigenvalues of Lemma 2.1 is generally not powerful in high dimensions (see [2]) but does not require detailed properties of the objective function f which appear in theoretical results on annealing. Furthermore, the techniques for estimating the spectral gap γ and the path length between two elements of the state space may be adapted in particular situations to yield substantial improvements. Examples of this are shown for inequality constraints and when a product structure exists on the constraint set.

Improving the bounds of this paper and extending them to the nonreversible annealing algorithms treated in somewhat abstract terms in [12] and [16] would serve as interesting problems for further related research.

REFERENCES

[1] O. Catoni, *Applications of sharp large deviations estimates to optimal cooling schedules*, Ann. Inst. H. Poincaré Probab. Statist., 27 (1991), pp. 463–518.

[2] P. Diaconis and L. Saloff-Coste, *Nash inequalities for finite Markov chains*, J. Theoret. Probab., 9 (1996), pp. 459–510.

[3] P. Diaconis and D. Stroock, *Geometric bounds for eigenvalues of Markov chains*, Ann. Appl. Probab., 1 (1991), pp. 36–61.

[4] P. Diaconis and B. Sturmfels, *Algebraic Algorithms for Sampling from Conditional Distributions*, Technical Report 6, Department of Statistics, Stanford University, Stanford, CA, 1993.

[5] W. Feller, *An Introduction to Probability Theory and Its Applications, Vol.* 1, 2nd ed., John Wiley, New York, 1957.

[6] S. Geman and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Trans. Pattern Analysis and Machine Intelligence, 6 (1984), pp. 721–741.

[7] B. Gidas, *Nonstationary Markov chains and convergence of the annealing algorithm*, J. Stat. Phys., 39 (1985), pp. 73–131.

[8] B. Hajek, *Cooling schedules for optimal annealing*, Math. Oper. Res., 13 (1988), pp. 311–329.

[9] R. Holley and D. Stroock, *Simulated annealing via Sobolev inequalities*, Comm. Math. Phys., 115 (1988), pp. 553–569.

[10] S. Ingrassia, *On the rate of convergence of the Metropolis algorithm and Gibbs sampler by geometric bounds*, Ann. Appl. Probab., 4 (1994), pp. 347–389.

[11] E. L. Johnson, *Integer Programming: Facets, Subadditivity, and Duality for Group and Semigroup Problems*, CBMS-NSF Regional Conf. Ser. in Appl. Math., SIAM, Philadelphia, 1980.

[12] L. Miclo, *Remarques sur l'hypercontractivité et l'évolution de l'entropie pour des chaînes de Markov finies*, in Seminaire de Probabilités XXXI, J. Azéma, ed., Springer, New York, 1997, pp. 136–167.

[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C,* 2nd ed., Cambridge University Press, New York, 1992.

[14] B. Ripley, *Stochastic Simulation*, John Wiley, New York, 1987.

[15] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley, New York, 1986.

[16] A. Trouvé, *Rough large deviation estimates for the optimal convergence speed exponent of generalized simulated annealing algorithms*, Ann. Inst. H. Poincaré Probab. Statist., 32 (1996), pp. 299–348.

[17] H. Wilf, *Generating Functionology*, Academic Press, San Diego, 1988.

SIAM J. Optim.
Vol. 9, No. 1, pp. 62–83

A FINITE CONTINUATION ALGORITHM FOR BOUND CONSTRAINED QUADRATIC PROGRAMMING[*]

KAJ MADSEN[†], HANS BRUUN NIELSEN[†], and MUSTAFA ÇELEBI PıNAR[‡]

Abstract. The dual of the strictly convex quadratic programming problem with unit bounds is posed as a linear ℓ_1 minimization problem with quadratic terms. A smooth approximation to the linear ℓ_1 function is used to obtain a parametric family of piecewise-quadratic approximation problems. The unique path generated by the minimizers of these problems yields the solution to the original problem for finite values of the approximation parameter. Thus, a finite continuation algorithm is designed. Results of extensive computational experiments are reported.

Key words. bound constrained quadratic programming, Lagrangian duality, linear ℓ_1 estimation, Huber's M-estimator, robust regression

AMS subject classifications. 90C20, 65K05, 65U05, 65F20

PII. S1052623495297820

1. Introduction.

We consider the strictly convex quadratic programming problem (QP) with unit bounds:

[BCQP]

$$\begin{array}{rl} \min\limits_{y} & H(y) = -d^T y + \frac{1}{2} y^T Q y \\ \text{subject to} & -\mathbf{1} \le y \le \mathbf{1}, \end{array}$$

where Q is an $m \times m$ symmetric, positive definite matrix, and y and d are m-vectors.

In this paper we study a dual continuation algorithm for the solution of [BCQP]. We first show that the dual of [BCQP] is an unconstrained minimization problem, where the function is composed of a linear ℓ_1 term and strictly convex quadratic terms. This nondifferentiable function is approximated by a smooth piecewise linear-quadratic Huber function. The resulting smooth problems yield a unique path that converges to the primal-dual optimal solutions. We follow the path using a continuation algorithm based on Newton's method. This algorithm is inspired by our earlier work on linear programming with unit bounds [11]. In this reference, the dual of a linear program is formulated as an ℓ_1 minimization problem. We solve the dual problem using a continuation algorithm based on the piecewise-linear paths generated by a smooth approximation problem. The smooth problem comes from robust statistics, where it was used by Huber as an alternative to the least squares estimation [7]. The most important property of the smooth problems is that they yield primal-dual optimal solutions for sufficiently small values of a continuation parameter. This allows a new finite, numerically stable continuation algorithm for linear programming.

We apply a similar philosophy here to the dual of [BCQP]. We approximate the ℓ_1 term by a Huber function term. This yields a family of problems parameterized by a smoothing parameter γ. This parameter is alternatively referred to as a continuation

[*] Received by the editors December 3, 1995; accepted for publication (in revised form) February 9, 1998; published electronically October 30, 1998. This research was supported in part by Danish Natural Sciences Research Council grant 11-0505.

http://www.siam.org/journals/siopt/9-1/29782.html

[†] Institute of Mathematical Modelling, Technical University of Denmark, 2800, Lyngby, Denmark (km@imm.dtu.dk, hbn@imm.dtu.dk).

[‡] Department of Industrial Engineering, Bilkent University, 06533 Bilkent, Ankara, Turkey (mustafap@bilkent.edu.tr).

parameter as in the linear programming case. However, unlike the linear programming case, the path generated by the minimizers of the smooth problem is unique and is no longer piecewise linear. This requires a fresh look at the properties of the path and its behavior for sufficiently small values of the continuation parameter, that is, the analysis of [11] does not apply here. However, we are able to establish that primal-dual optimal solutions are obtained from the path for positive, sufficiently small values of the parameter.

The following properties of the approximation are emphasized as the main contributions of this paper:

P0. The primal-dual minimizers of the smooth problem define a unique path as a function of the smoothing parameter γ.

P1. The primal-dual optimal solutions to [BCQP] are obtained for sufficiently small $\gamma > 0$ using information from the path, that is, γ does not have to be decreased to zero in order to obtain an exact solution to the QP problem (Theorems 2.2 and 2.3).

P2. Although the unique path leading to the primal-dual solutions is nonlinear, a powerful extrapolation result allows computation of primal-dual candidates for optimality (Theorem 2.2).

Furthermore, our main results are obtained without any nondegeneracy assumptions on the problem. In particular, Theorem 2.2 (the description of the extrapolation) and Theorem 2.3 (the behavior of the path for small values of the continuation parameter) are established in the absence of any restrictive assumptions.

These properties suggest an algorithm to trace the path to arrive at a solution of [BCQP]. We refer to the path as the "solution path" throughout the rest of the paper. Our algorithm is best interpreted as a continuation algorithm since it possesses the following main features of continuation algorithms.

1. The solution of a parametrized family of subproblems as a parameter varies over an interval; in our case, the smooth "Huber" problem as a function of the smoothing parameter γ.
2. The use of a local iterative method to solve the subproblems. We use a finite Newton method [10] to solve the smooth Huber problem.
3. The use of an extrapolation technique to guess an optimal primal-dual pair from a point on the path.

As a result of P1 and P2 above, the continuation algorithm is a finite procedure provided that γ is decreased by at least a certain factor after each unconstrained minimization. We make these ideas precise in the forthcoming sections.

In this algorithm, Newton's method is used to locate the path for some value of the smoothing parameter. Unless optimality is reached, Newton's method is invoked for a reduced value of the parameter from a point no longer on the path, and the cycle is repeated. We summarize the algorithmic scheme as follows:

```
Compute initial $\gamma$
repeat
    compute a solution of the approximation problem
    decrease $\gamma$
until optimality.
```

This scheme closely relates our algorithm to penalty and barrier methods and in general to path-following methods. To the best of our knowledge, from this perspective, both the theoretical analysis of section 2 and the algorithm stand as novel contributions to the quadratic programming literature.

We develop a numerically stable implementation of the new algorithm for dense problems. We also compare the performance of the algorithm to LSSOL, a software system for quadratic programming from Stanford University's Systems Optimization Laboratory, and to an interior point algorithm of Han, Pardalos, and Ye [6].

For a review of the literature on quadratic programming we refer the reader to the paper by Moré and Toraldo [14]. It seems that currently the fastest algorithms for [BCQP] are the active set methods [14]. For problem [BCQP], active set methods can efficiently add or delete many constraints from the active set at one iteration. Primal-dual interior point algorithms have also been recently developed for [BCQP] [6]. Other related ideas have been proposed in more recent papers by Coleman and Hulbert [1] and Li and Swetits [8, 9]. In [1] Coleman and Hulbert reformulate [BCQP] as an unconstrained minimization problem involving an ℓ_1 term. This reformulation is obtained by manipulating the Karush–Kuhn–Tucker conditions of [BCQP]. They apply a superlinearly convergent modified Newton method to this reformulation. In this regard our point of departure is identical to that of [1]. Li and Swetits [8, 9] reformulate the convex quadratic programming problem as an unconstrained minimization of a convex quadratic spline function.

In the rest of the paper we proceed as follows. In section 2 we present a simple derivation of the dual problem, and we explore the relation of the nondifferentiable dual to the approximation problem. We give the details and analysis of Newton's method applied to the approximation problem in section 3. In section 4 we discuss some implementation details and generation of test problems, and we report the results of extensive computational experiments with the new algorithm. Comparisons to competing algorithms are also made. Concluding remarks are offered in section 5.

2. A nondifferentiable dual problem and its approximation. We begin our study of [BCQP] by deriving a dual problem. Since Q is symmetric positive definite, there exists a full rank matrix $A \in \Re^{m\times m}$ such that $Q = A^TA$. Then the quadratic program can be rewritten

$$\begin{array}{rl} \displaystyle\min_{y} & -(A^Tb)^Ty + \frac{1}{2}y^TA^TAy \\ \text{subject to} & -\mathbf{1} \le y \le \mathbf{1} \end{array}$$

for some $b \in \Re^m$ such that $d = A^Tb$. Let $u = Ay$ and rewrite the problem as

$$\begin{array}{rc} \displaystyle\min_{y,u} & -b^Tu + \frac{1}{2}u^Tu \\ \text{subject to} & Ay = u \\ & -\mathbf{1} \le y \le \mathbf{1}. \end{array}$$

Associating dual multipliers $x \in \Re^m$ with the equality constraints, we form the following Lagrangian max-min problem:

$$\max_{x} \min_{u,-\mathbf{1}\le y\le \mathbf{1}} \left\{ \frac{1}{2}u^Tu - b^Tu + x^T(Ay-u) \right\},$$

which is equivalent to

$$\max_{x} \left\{ \min_{u} \left\{ \frac{1}{2}u^Tu - b^Tu - u^Tx \right\} + \left\{ \min_{-\mathbf{1}\le y\le \mathbf{1}} x^TAy \right\} \right\}.$$

It is easy to see that the first minimization yields the identity

$$Ay = x + b. \tag{2.1}$$

Hence, we get the term

$$-\frac{1}{2}x^T x - b^T x - \frac{1}{2}b^T b.$$

The second minimization over y is also straightforward and yields

$$\min_{-\mathbf{1} \le y_i \le \mathbf{1}} x_i (Ay)_i = \begin{cases} (A^T x)_i & \text{if } (A^T x)_i \le 0, \\ -(A^T x)_i & \text{if } (A^T x)_i \ge 0. \end{cases}$$

However, this is simply the negative of the ℓ_1-norm of $A^T x$. Therefore, our dual problem is

$$\text{minimize} F(x) \;\equiv\; \|A^T x\|_1 + \frac{1}{2}x^T x + b^T x + \frac{1}{2}b^T b. \tag{2.2}$$

As a result of strict convexity, the primal and dual optimal solutions are unique.

Let

$$r(x) = A^T x. \tag{2.3}$$

From the derivation, the conditions for (y_0, x_0) to be optimal can be expressed as

$$\begin{array}{rcl} & Ay_0 = b + x_0 \;, & \\ r_i(x_0) > 0 & \Longrightarrow & y_{0i} = -1, \\ r_i(x_0) < 0 & \Longrightarrow & y_{0i} = 1, \\ -1 < y_{0i} < 1 & \Longrightarrow & r_i(x_0) = 0, \end{array}$$

for all $i = 1, \ldots, m$. From this point on, we use (y_0, x_0) to denote a primal-dual optimal pair.

Let us define a set $\hat{S}$ of "sign vectors" such that $\hat{S} = \{s \in \Re^m \mid s_i \in \{-1, 0, 1\}\}$. Now, define the sign vector $s_0(x)$ such that

$$s_{0i}(x) = \begin{cases} -1 & \text{if } r_i(x) < 0, \\ 0 & \text{if } r_i(x) = 0, \\ 1 & \text{if } r_i(x) > 0, \end{cases} \tag{2.4}$$

and define

$$W_0 = \text{diag}(w_1, \ldots, w_m) \quad \text{with} \quad w_i = 1 - s_{0i}^2. \tag{2.5}$$

Let $s_0 = s_0(x_0)$ and let W_0 be derived from s_0 using (2.5). Now, we can compactly express the optimality conditions as

$$AW_0 y_0 - As_0 = b + x_0. \tag{2.6}$$

Since A has full rank, this implies that the following linear system is consistent:

$$(AW_0 A^T)h = As_0 + b + x_0. \tag{2.7}$$

Since the null space $\mathcal{N}(AW_0A^T)$ coincides with the null space $\mathcal{N}(W_0A^T)$, $W_0A^T h$ is constant no matter which solution h to (2.7) is picked.

2.1. The smooth Huber approximation. Consider the function $\phi : \Re \mapsto \Re$:

$$\phi_\gamma(t) = \begin{cases} \frac{1}{2\gamma}t^2 & \text{if } |t| \le \gamma, \\ |t| - \frac{1}{2}\gamma & \text{if } |t| > \gamma, \end{cases} \tag{2.8}$$

for some scalar parameter $\gamma > 0$. This function is known as Huber's M-estimator function in robust statistics. Now, we replace (2.2) by the following differentiable problem:

$$\min_x \Phi_\gamma(x) + \frac{1}{2}x^T x + b^T x + \frac{1}{2}b^T b, \tag{2.9}$$

where

$$\Phi_\gamma(x) = \sum_{i=1}^{m} \phi_\gamma(r_i(x)). \tag{2.10}$$

We discuss some well-known properties of this function in section 3.1. To view this problem in quadratic programming format, we define a new sign vector s_γ:

$$s_\gamma(x) = [s_{\gamma 1}(x), \ldots, s_{\gamma m}(x)] \quad \text{with} \quad s_{\gamma i}(x) = \begin{cases} -1 & \text{if } r_i(x) < -\gamma, \\ 0 & \text{if } |r_i(x)| \le \gamma, \\ 1 & \text{if } r_i(x) > \gamma, \end{cases} \tag{2.11}$$

and define

$$W_s = \operatorname{diag}(w_1, \ldots, w_m) \quad \text{with} \quad w_i = 1 - s_{\gamma i}^2. \tag{2.12}$$

Therefore, we have the following minimization problem:

$$\text{minimize } F_\gamma(x) \equiv \frac{1}{2\gamma} r^T W_s r + s_\gamma^T \left[r - \frac{1}{2}\gamma s_\gamma \right] + \frac{1}{2}x^T x + b^T x + \frac{1}{2}b^T b, \tag{2.13}$$

where the argument x of r and $\mathbf{s}_\gamma$ is dropped for notational convenience. We refer to the above problem as the "Huber problem" for ease of expression. Clearly, this problem has a unique minimizer as a result of strict convexity. In the following, we use the notations x_γ for the minimizer of F_γ and $W_\gamma = W_s$, where $s = s_\gamma(x_\gamma)$. For notational convenience, we use W_γ and W_s interchangeably in our analysis when the meaning is clear from the context.

It can be shown using Lagrangian duality that the dual problem to (2.13) is given by
[PBCQP]

$$\begin{array}{rl} \displaystyle\min_y & H(y) = -d^T y + \frac{1}{2} y^T (Q + \gamma I) y \\ \text{subject to} & -\mathbf{1} \le y \le \mathbf{1}. \end{array}$$

We notice that the above problem is simply a quadratically perturbed version of [BCQP]. This relates our analysis to previous studies by Mangasarian [12] and Mangasarian and Meyer [13], where quadratic and nonlinear perturbations of linear programs were addressed.

2.2. The relation between $\boldsymbol{F}, \boldsymbol{F_\gamma}$, and [BCQP]. In this section we establish some important properties of the Huber approximation. These properties characterize the proposed algorithm and are used to verify finite convergence.

We begin with some simple results. We can immediately observe the following elementary fact:

$$\lim_{\gamma \to 0} \phi_\gamma(t) = |t| \ , \tag{2.14}$$

for any $t \in \Re$. Now, we have the following simple result.

LEMMA 2.1. *Let x_γ denote the minimizer of the function F_γ. Then,*

$$0 \le F(x_0) - F_\gamma(x_\gamma) \le m\frac{\gamma}{2}. \tag{2.15}$$

Proof. From the definitions of F and F_γ, we have for any $x \in \Re^m$

$$0 \le F(x) - F_\gamma(x) \le m\frac{\gamma}{2}.$$

Since x_0 and x_γ are minimizers of F and F_γ, we therefore obtain

$$F_\gamma(x_\gamma) \le F_\gamma(x_0) \le F(x_0)$$

and

$$F(x_0) - m\frac{\gamma}{2} \le F_\gamma(x_\gamma) - m\frac{\gamma}{2} \le F_\gamma(x_\gamma).$$

This proves (2.15). □

THEOREM 2.1. *Let x_γ denote the minimizer of the function F_γ. Then,*

$$\lim_{\gamma \to 0} x_\gamma = x_0. \tag{2.16}$$

Proof. Since the functions are continuous and strictly convex, i.e., the minimizers are unique, the result follows using (2.14) and (2.15). □

Let $s = s_\gamma(x_\gamma)$. The minimizer x_γ of F_γ satisfies the following necessary condition:

$$A\left[\frac{1}{\gamma}W_s r(x_\gamma) + s\right] + b + x_\gamma = \mathbf{0}, \tag{2.17}$$

which may be written in the form

$$\big(AW_sA^T + \gamma I\big)x_\gamma = -\gamma\big(As + b\big), \tag{2.18}$$

or as

$$Ay_\gamma = b + x_\gamma, \tag{2.19}$$

where we have defined

$$y_\gamma = -\left(\frac{1}{\gamma}W_\gamma r(x_\gamma) + s\right). \tag{2.20}$$

Using (2.17) we see that y_γ is feasible in [BCQP] and optimal in [PBCQP]. Clearly, using (2.1), (2.16), and (2.19) we have

$$\lim_{\gamma \to 0} y_\gamma = y_0. \tag{2.21}$$

In the remainder of this section, we study the behavior of the solution paths $\{x_\gamma\}$ and $\{y_\gamma\}$ as $\gamma \searrow 0$. For fixed s (and therefore W_s) we introduce the singular value decomposition (SVD) of the matrix W_sA^T:

$$W_sA^T = U\Sigma V^T. \tag{2.22}$$

Here, the matrices U and V with columns $\{u_j\}_{j=1}^m$ and $\{v_j\}_{j=1}^m$ are orthogonal, and the singular values are given in Σ :

$$\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_m) \quad \text{with} \quad \sigma_1 \geq \cdots \geq \sigma_q > 0, \quad \sigma_{q+1} = \cdots = \sigma_m = 0. \tag{2.23}$$

The number q is the rank of the matrix W_sA^T, the vectors $\{u_j\}_{j=1}^q$ and $\{v_j\}_{j=1}^q$ form an orthonormal basis of the range of W_sA^T and AW_sA^T, respectively, and $\{v_j\}_{j=1}^m$ is an orthonormal basis of $\Re^m$. This means that we can write

$$As + b = \sum_{j=1}^m \alpha_j v_j = V\alpha, \tag{2.24}$$

and by inserting (2.22) into (2.18) we get

$$\big(V\Sigma^2V^T + \gamma I\big)x_\gamma = -\gamma V\alpha,$$

from which we find

$$x_\gamma \;=\; -\gamma \sum_{j=1}^m \frac{\alpha_j}{\sigma_j^2 + \gamma}\, v_j \;=\; -\gamma \sum_{j=1}^q \frac{\alpha_j}{\sigma_j^2 + \gamma}\, v_j \;-\; \sum_{j=q+1}^m \alpha_j v_j. \tag{2.25}$$

Furthermore, from (2.20) and (2.22) we get

$$y_\gamma = \sum_{j=1}^q \frac{\sigma_j \alpha_j}{\sigma_j^2 + \gamma}\, u_j \;-s. \tag{2.26}$$

As we shall see in Theorem 2.3, $s_\gamma(x_\gamma)$ and therefore, W_s are constant for γ small enough. When the SVD factorization (2.22) corresponds to this W_s, it follows that

$$x_0 = \lim_{\gamma\to 0} x_\gamma = -\sum_{j=q+1}^m \alpha_j v_j \quad \text{and} \quad y_0 = \lim_{\gamma \to 0} y_\gamma = \sum_{j=1}^q \frac{\alpha_j}{\sigma_j}\, u_j \;-s. \tag{2.27}$$

In the algorithm of section 3 we do not compute the SVD, but the following theorem provides us with an extrapolation formula that is used in our algorithm to test for optimality. To the best of our knowledge, this is a new result in the path-following literature.

THEOREM 2.2. *Let x_δ be the minimizer of F_δ for $0 < \delta \leq \gamma$ with $s = s_\delta(x_\delta)$ and $W = W_s$. Assume that $s_\delta(x_\delta) = s$ for $0 < \delta \leq \gamma$. Then,*

$$x_0 = x_\delta + \delta d_\delta^{(\delta)} \quad \textit{and} \quad y_0 = WA^T d_\delta^{(0)} - s\ , \tag{2.28}$$

where $d_\delta^{(\delta)}$ and $d_\delta^{(0)}$ are the minimum-norm solutions to the linear systems

$$(AWA^T)d = As + b + x_\delta \quad \textit{and} \quad (AWA^T)d = As + b + x_0\ , \tag{2.29}$$

respectively.

Proof. From (2.24)–(2.25) we get

$$As + b + x_\delta \;=\; \sum_{j=1}^{q}\Bigl(1 - \frac{\delta}{\sigma_j^2+\delta}\Bigr)\alpha_j v_j \;=\; \sum_{j=1}^{q}\frac{\sigma_j^2}{\sigma_j^2+\delta}\,\alpha_j v_j \tag{2.30}$$

(the contributions for $j = q{+}1, \dots, n$ cancel). Thus, the first of the rank-deficient systems in (2.29) is consistent, and the minimum-norm solution is

$$d_\delta^{(\delta)} = \sum_{j=1}^{q}\frac{\alpha_j}{\sigma_j^2+\delta}\,v_j. \tag{2.31}$$

By adding $\delta d_\delta^{(\delta)}$ to x_δ (given by (2.25)) we get x_0, as expressed in (2.27). For the other system, we find

$$As + b + x_0 = \sum_{j=1}^{q}\alpha_j v_j. \tag{2.32}$$

Thus, the second system in (2.29) is also consistent. The minimum-norm solution is

$$d_\delta^{(0)} = \sum_{j=1}^{q}\frac{\alpha_j}{\sigma_j^2}\,v_j, \tag{2.33}$$

and by inserting this into (2.28) we get y_0 as expressed in (2.27). □

In general, let $(\hat{x}_0, \hat{y}_0)$ denote the quantities computed by (2.28). They provide practical termination criteria for the algorithm defined in section 3.

In Theorem 2.3 we show that $s_\gamma(x_\gamma)$ is constant when γ is small enough. For some of the components of s_γ this is almost trivial. The components which cause difficulty are those for which $r_i(x_0) = 0$ and $|y_{0i}| = 1$. This set is denoted by $\mathcal{D}$, and the set of sign vectors for which the "easy" components equal those of s_0 is denoted by $\mathcal{S}$. More precisely, $\mathcal{D}$ and $\mathcal{S}$ are defined as follows. Let $s \in \hat{S}$, $\kappa_s^+ = \{i : s_i = 1\}$, and $\kappa_s^- = \{i : s_i = -1\}$ with $\kappa_s = \kappa_s^+ \cup \kappa_s^-$ and $\kappa_s^0 = \{i : s_i = 0\}$. Let $\mathcal{D} = \{i : |y_{0i}| = 1\} \cap \kappa_s^0$ and $\mathcal{S} = \{s \in \hat{S} \mid s_i = s_{0i} \text{ for } i \notin \mathcal{D}\}$.

THEOREM 2.3. *Let $s_0 = s_0(x_0)$. There exists γ^* such that $s_\gamma(x_\gamma)$ is constant, with $\kappa_{s_0}^+ \subseteq \kappa_{s_\gamma}^+$, $\kappa_{s_0}^- \subseteq \kappa_{s_\gamma}^-$ for $0 < \gamma \le \gamma^*$.*

Proof. Since the number of different sign vectors is finite, there must exist a sequence of positive numbers $\gamma_1, \gamma_2, \dots$, with $\gamma_k \searrow 0$ for $k \to \infty$ such that $s_\gamma(x_\gamma)$ is constant for $\gamma = \gamma_k$, $k = 1, 2, \dots$. Denote this constant sign vector by s.

According to (2.3) and (2.11), the elements of s are defined by the values of $r_i(x_\gamma) = a_i^T x_\gamma$. Since $x_\gamma \to x_0$, we have $|a_i^T x_\gamma| > \gamma$ for $i \in \kappa_s^0$ and γ small enough. Furthermore, since $y_\gamma \to y_0$, we have from (2.20) that $|a_i^T x_\gamma|/\gamma < 1$ for $i \in \kappa_{s_0}^0 \setminus \mathcal{D}$, and γ small enough. Therefore, since $\gamma_k \searrow 0$, it must be the case that $s \in \mathcal{S}$.

Now, let $W = W_s$ and let (2.22) be the SVD factorization of WA^T. Furthermore, let d_γ be the solution to

$$(AWA^T + \gamma I)d_\gamma = As + b + x_0.$$

By inserting (2.32), we see that

$$d_\gamma = \sum_{j=1}^{q}\frac{\alpha_j}{\sigma_j^2+\gamma}\,v_j. \tag{2.34}$$

We introduce

$$\psi_i(\gamma) \equiv a_i^T d_\gamma = \sum_{j=1}^{q} \frac{\alpha_j}{\sigma_j^2 + \gamma}\, a_i^T v_j$$

for $i = 1, 2, \ldots, m$. Since ψ_i is a rational function for $\gamma{>}0$, it can only have a finite number of oscillations as $\gamma{\rightarrow}0$, and hence there exists $\gamma_1^* > 0$ such that for each i

$$\begin{array}{lll} \text{either} & |\psi_i(\gamma)| > 1 & \text{for } 0 < \gamma \leq \gamma_1^* \\ \text{or} & |\psi_i(\gamma)| \leq 1 & \text{for } 0 < \gamma \leq \gamma_1^*. \end{array}$$

If $i \notin \kappa_{s_0}$, then $r_i(x_0) = 0$ and

$$r_i(x_0 - \gamma d_\gamma) = -\gamma \psi_i(\gamma).$$

Hence, the ith component of $s_\gamma(x_0 - \gamma d_\gamma)$ is constant for $0 < \gamma \leq \gamma_1^*$. Since d_γ is bounded (see (2.34)) the other components of $s_\gamma(x_0 - \gamma d_\gamma)$ must also be constant in some interval $0 < \gamma \leq \gamma_2^*$. Therefore, $s_\gamma(x_0 - \gamma d_\gamma)$ is constant for $0 < \gamma \leq \gamma_3^* \equiv \min\{\gamma_1^*, \gamma_2^*\}$.

Finally, let $\gamma = \gamma_k$, $\gamma_k \leq \gamma_3^*$ denote a value for which $s_\gamma(x_\gamma) = s$. It follows from (2.25), (2.27), and (2.34) that the unique minimizer x_γ is equal to $x_0 - \gamma d_\gamma$. □

Notice that $\mathcal{S}$ may be a singleton, in which case it is possible to establish a stronger result. This depends on a certain nondegeneracy assumption stated below.

THEOREM 2.4. *Let x_0 be the minimizer of F with $s = s_0(x_0)$ and $W = W_s$. Assume there exists $\gamma_1 > 0$ such that the solution d_γ to the system*

$$(AWA^T + \gamma I)d = As + b + x_0 \tag{2.35}$$

has the property

$$\|WA^T d_\gamma\|_\infty \leq 1 \quad \textit{for } \gamma \in (0, \gamma_1]\ . \tag{2.36}$$

Then, there exists $\gamma^* > 0$ such that $s_\gamma(x_\gamma)$ is constant for $\gamma \in (0, \gamma^*]$. Furthermore, $s_\gamma(x_\gamma) = s$ for $\gamma \in (0, \gamma^*]$.

Proof. Let $\delta = \min\{|r_i(x_0)| : r_i(x_0) \neq 0\}$. Choose $\gamma_2 < \delta$ such that, for $0 < \gamma \leq \gamma_2$,

$$r_i(x_0) - \gamma a_i^T d_\gamma > \gamma_2 \text{ for } i \in \kappa_s^+, \tag{2.37}$$

$$r_i(x_0) - \gamma a_i^T d_\gamma < -\gamma_2 \text{ for } i \in \kappa_s^-. \tag{2.38}$$

Using (2.36), $s_\gamma(x_0 - \gamma d_\gamma) = s(x_0)$. Now, from (2.35) and using the fact that $WA^T x_0 = 0$, we get

$$\begin{aligned} (AWA^T + \gamma I)(-\gamma d_\gamma) &= -\gamma(As + b + x_0), \\ (AWA^T + \gamma I)(-\gamma d_\gamma) &= -AWA^T x_0 - \gamma(As + b + x_0), \\ (AWA^T + \gamma I)(x_0 - \gamma d_\gamma) &= -\gamma(As + b). \end{aligned}$$

Hence, $x_0 - \gamma d_\gamma$ is the minimizer of F_γ, and the theorem is proved with $\gamma^* = \min\{\gamma_1, \gamma_2\}$. □

DEFINITION 2.1. *A primal-dual optimal pair (y, x) is nondegenerate if the following condition holds for each zero component $r_i(x)$ of $r(x)$:*

$$r_i(x) = 0 \quad \textit{and} \quad -1 < y_i < 1\ . \tag{2.39}$$

Corollary 2.1. *Let (y_0, x_0) be a nondegenerate primal-dual optimal pair for $[BCQP]$ with $s = s_0(x_0)$ and $W = W_0(x_0)$. Then, there exists $\gamma^* > 0$ with $\gamma^* < \min\{|r_i(x_0)| : i \in \sigma_s\}$ such that $s_\gamma(x_\gamma) = s$ for $\gamma \in (0, \gamma^*]$.*

Proof. Since A has full rank, under the nondegeneracy assumption on (y_0, x_0) any solution d to the optimality system (2.7)

$$(AWA^T)d = As + b + x_0$$

satisfies

$$\|WA^T d\|_\infty < 1.$$

Now, using the fact that $\lim_{\gamma \to 0} d_\gamma = d^*$, where d^* denotes the minimum-norm solution to (2.7) and the continuity of the norm in its argument, there exists $\gamma_1^* > 0$ such that for $\gamma \in (0, \gamma_1^*]$ the unique solution d_γ of (2.35) satisfies

$$\|WA^T d_\gamma\|_\infty < 1.$$

The rest of the proof follows from Theorem 2.4. □

Hence, under a nondegeneracy assumption, the Huber problem is guaranteed to generate a sign vector identical to the sign vector corresponding to the dual optimal point x_0 for a sufficiently small value γ^* of γ. The magnitude of γ^* is related to the smallest nonzero component of $r(x_0)$ as stated in Corollary 2.1.

3. The algorithm.

The new algorithm is based on minimizing the function F_γ for a set of decreasing values of γ. It can be described as follows. Starting from a point x, we find a minimizer of F_γ for some $\gamma > 0$, i.e., we locate the solution path for some value of γ. Utilizing Theorem 2.2 we compute $(\hat{y}_0, \hat{x}_0)$, estimates of primal-dual solutions. If optimality is not reached at $(\hat{y}_0, \hat{x}_0)$, we reduce the value of γ. Starting from a new point corresponding to the reduced value of γ, we compute the exact minimizer of F_γ using a Newton-type algorithm. Hence, we follow the solution path closely without having to stay on it. Based on Theorem 2.2, this process terminates when the duality gap is closed and primal feasibility is obtained.

The algorithm has two main components: (1) the solution of the smooth problem, i.e., minimization of F_γ for a given value of γ; (2) the check for optimality and the reduction of γ with the computation of an initial point for the solution of the subsequent Huber problem. We now consider these two components in detail.

3.1. Solving the Huber problem.

3.1.1. Properties of F_γ.

In this section we describe some essential properties of F_γ.

Clearly, F_γ is composed of a finite number of quadratic functions. In each domain $D \subseteq \Re^m$, where $s_\gamma(x)$ is constant, F_γ is equal to a specific quadratic function. These domains are separated by the following union of hyperplanes:

$$B_\gamma = \{x \in \Re^m \mid \exists i \ : \ |r_i(x)| = \gamma\}.$$

A sign vector s is γ-*feasible* at x if

$$\text{for all } \varepsilon > 0 \ \exists z \in \Re^m \setminus B_\gamma \ : \ \|x - z\| < \varepsilon \ \wedge \ s = s_\gamma(z).$$

If s is a γ-feasible sign vector at some point x, then let Q_s be the quadratic function which equals F_γ on the subset

$$\mathcal{C}_s^\gamma = \text{cl}\{z \in \Re^m \mid s_\gamma(z) = s\}. \tag{3.1}$$

$\mathcal{C}_s^\gamma$ is called a *Q-subset* of $\Re^m$. Notice that any $x \in \Re^m \setminus B_\gamma$ has exactly one corresponding Q-subset ($s = s_\gamma(x)$), whereas a point $x \in B_\gamma$ belongs to two or more Q-subsets. Therefore, in general we must give a sign vector s in addition to x in order to specify which quadratic function we are currently considering as representative of F_γ. However, the gradient of F_γ is independent of the choice of s.

Q_s can be defined as follows:

$$Q_s(z) = \frac{1}{2\gamma}(z-x)^T(AW_sA^T + I)(z-x) + F_\gamma^{'T}(x)(z-x) + F_\gamma(x). \tag{3.2}$$

The gradient of the function F_γ is given by

$$F'_\gamma(x) = A\left[\frac{1}{\gamma}W_s r(x) + s\right] + b + x, \tag{3.3}$$

where s is a γ-feasible sign vector at x. For $x \in \Re^m \setminus B_\gamma$, the Hessian of F_γ exists and is given by

$$F''_\gamma(x) = \frac{1}{\gamma}AW_sA^T + I. \tag{3.4}$$

The set of indices corresponding to "small" residuals

$$\mathcal{A}_\gamma(z) = \{i \mid 1 \le i \le m \ \wedge \ s_{\gamma i}(z) = 0\} \tag{3.5}$$

is called the γ-*active* set at z.

3.1.2. Computing a minimizer of F_γ. The algorithm for computing a minimizer x^* of F_γ is based on a modified Newton algorithm given in [10]. This algorithm becomes simpler in our case as a result of strict convexity of the objective function. The algorithm consists of applying Newton's method to the function F_γ followed by a piecewise linear one-dimensional search. The idea is to locate the Q-subset of $\Re^m$ which contains its own minimizer using Newton's method. A search direction h is computed by minimizing the quadratic Q_s, where $s = s_\gamma(x)$ and x is the current iterate. More precisely, we consider the equation

$$Q''_s h = -Q'_s(x),$$

where Q''_s and Q'_s denote the Hessian and gradient of Q_s, respectively. From (3.2)–(3.4) we obtain

$$(AW_sA^T + \gamma I)h = -AW_s r - \gamma(As + b + x). \tag{3.6}$$

The next iterate is found by a line search aiming for a zero of the directional derivative [10]. More precisely, the next iterate is the point $x + \alpha h$, $\alpha > 0$, for which the function

$$\rho(\alpha) = F_\gamma(x + \alpha h)$$

is minimized. Since ρ is a convex univariate function, the problem is to find a zero of the increasing piecewise-linear smooth function ρ'. The solution α to this problem is positive since $\rho'(0) < 0$ by the definition of h.

Let $\{\alpha_k\}$, $k = 1, \dots, n$ be the set of positive breakpoints where ρ' has kinks, i.e., the set of points where an $s_{\gamma i}(x + \alpha h)$ changes value:

$$\mathcal{K} = \{\alpha > 0 \mid \exists i \in E : \ |(A^T(x + \alpha h))_i| = \gamma\},$$

where $E = \{i \mid 1 \le i \le m \ \wedge \ (A^T h)_i \ne 0\}$. Assume that the points α_k, $k = 1, \dots, n$ are given in ascending order. Then the line search procedure is as follows:

```
$j := 0$
$\alpha_0 = 0$
**repeat**
    $j \leftarrow j + 1$
    find $\rho'(\alpha_j)$
**until** $\rho'(\alpha_j) \ge 0$
find the zero $\alpha$ of the linear function $\rho'$ in the interval $[\alpha_{j-1}, \alpha_j]$.
```

This procedure is computationally cheap as a result of the piecewise-linear nature of F'_γ. First, the elements of the set $\mathcal{K}$ need not be sorted in practice. It suffices to pick the smallest element among the elements that remain in the set as the search proceeds. Furthermore, the quantity $\rho'(\alpha_j)$ is easily obtained from $\rho'(\alpha_{j-1})$, since the move from α_{j-1} to α_j only affects one term in the defining equation of ρ'. A more detailed description of this procedure is given in [10].

We summarize below the modified Newton algorithm:

```
**repeat**
    $s = s_\gamma(x)$
    find $h$ from (3.6)
    if $x + h \in \mathcal{C}_s^\gamma$ then
        $x \leftarrow x + h$
        stop = true
    else
        $x \leftarrow x + \alpha h$ (line search)
    endif
**until** stop.
```

The algorithm stops when we have $x + h \in \mathcal{C}^\gamma_{s(x)}$, i.e., we have found the local quadratic which contains its own minimum. Therefore, $x + h$ is a minimizer of F_γ as a result of (3.1), (3.2), and the convexity of F_γ. Now, we show that this occurs in a finite number of iterations. First, we notice that the line searches made in the algorithm are well defined. This follows from two observations. First, since A has full rank, there exists an index j for which $(A^T h)_j \ne 0$. Hence, the set E of breakpoints is always nonempty. Furthermore, $\rho(\alpha)$ is a strictly convex quadratic function of α, which implies that the line search must terminate at a minimum along the half-line.

THEOREM 3.1. *The Newton algorithm stops at a minimizer of F_γ after a finite number of iterations.*

Proof. The set of iterates is bounded since the method is descent. Suppose that the iteration is infinite. Then, the set of iterates must have an accumulation point, z^*, say. We consider two cases:

(i) $F'_\gamma(z^*) \ne \mathbf{0}$: Since F'_γ is continuous and since F_γ is composed of a finite number of quadratics, all directions are found via a finite set of positive definite matrices $AW_sA^T + \gamma I$. Hence, there exists $\epsilon > 0$ and $\delta > 0$ such that $\|z^* - x\| < \epsilon$ implies $F_\gamma(x) - F_\gamma(x_{next}) > \delta$, where x_{next} is the successor of x in the iteration. Since this happens infinitely often, the function values must tend to $-\infty$, which contradicts the boundedness of F_γ from below.

(ii) $F'_\gamma(z^*) = \mathbf{0}$: In this case, z^* is the minimizer of F_γ because of convexity. Let x be an iterate with $z^* \in \mathcal{C}^\gamma_{s(x)}$. Since z^* minimizes the quadratic Q_s and h is found by (3.6), $x + h = z^*$, and the algorithm stops. □

3.2. Checking optimality and reducing γ. Let x_γ be a minimizer of F_γ computed using the Newton algorithm of the previous section. Then, either the continuation algorithm terminates or the Newton algorithm is restarted using a reduced value of γ.

The stopping test is based on Theorem 2.2. It consists of checking the duality gap $H(\hat{y}_0) - F(\hat{x}_0)$ and the feasibility of $\hat{y}_0$, where $(\hat{y}_0, \hat{x}_0)$ are as given in Theorem 2.2. If the duality gap is zero (within the roundoff tolerance), then the algorithm is stopped provided the components of $\hat{y}_0$ satisfy

$$-1 \leq y_i \leq 1.$$

Otherwise, γ is decreased as

$$\gamma^{new} = \beta \cdot \gamma^{old},$$

where $\beta \in (0, 1)$. The precise description of this procedure is as follows:

```
$s = s_\gamma(x_\gamma)$
compute the minimum norm solution $d_\gamma^{(\gamma)}$ to $(AWA^T)d = As + b + x_\gamma$
compute $\hat{x}_0 = x_\gamma + \gamma d_\gamma^{(\gamma)}$
compute the minimum norm solution $d_\gamma^{(0)}$ to $(AWA^T)d = As + b + \hat{x}_0$
compute $\hat{y}_0 = WA^T d_\gamma^{(0)} - s$
if $H(\hat{y}_0) - F(\hat{x}_0) = 0$ and $\hat{y}_0$ is feasible then
    stop = true
else
    $\gamma \leftarrow \beta \cdot \gamma$
endif
```

To compute an advantageous starting point for the subsequent Newton iteration with γ^{new}, we use the following linear system derived from necessary conditions (2.17):

$$(AWA^T + \gamma^{new} I)x = -\gamma^{new}(As + b), \tag{3.7}$$

where $s = s_\gamma(x_\gamma)$ and $W = W_\gamma(x_\gamma)$. The solution x^{new} of (3.7) is used as the starting point for the Newton iteration.

We note that this procedure guarantees that, unless the duality gap is closed, γ is decreased by a nonzero factor after each unconstrained minimization. Hence, we have the following theorem.

THEOREM 3.2. *The continuation algorithm described in sections* 3.1.2 *and* 3.2 *stops at a primal-dual optimal pair* (y_0, x_0) *after a finite number of iterations.*

Proof. As a result of the above observation, γ is reduced by a certain factor after each unconstrained minimization phase unless optimality is reached. Hence, using Theorem 2.3, γ can only be decreased a finite number of times. Since the Newton algorithm of section 3.1.2 is finite (Theorem 3.1), the result follows. □

4. Implementation and testing. The major effort in the dual algorithm of section 3.1.2 is spent in solving systems (3.6) and (2.29). We use the AAFAC package of [15] to perform this. The solution is obtained via an LDL^T factorization of the matrix $C_k = AW_sA^T + \gamma I$ (where γ is zero in the case of (2.29)), so D and L are computed directly from the γ-active columns of A, i.e., without squaring the condition number as would be the case if C_k was first computed. The efficiency of the Newton algorithm depends critically on the fact that the difference between the γ-active set $\mathcal{A}_\gamma(x_k)$ and $\mathcal{A}_\gamma(x_{k-1})$ is caused by a few elements. This implies that the factorization of C_k can be obtained by relatively few up- and downdates of the factorization of C_{k-1}. Therefore, the computational cost of a typical iteration step is $O(m^2)$. Occasionally, a refactorization is performed. This consists of the successive updating of $LDL^T \leftarrow LDL^T + a_j a_j^T$ for all j in the γ-active set (starting with $L = I, D = \gamma I$). It is considered only when some columns of A leave the active set, i.e., when downdating is involved. If many columns leave, we may refactorize because it is cheaper. This part of the algorithm combines ideas from [3, 4]. For details see section 2 in [15]. The refactorization is an $O(m^3)$ process.

When a minimizer x_γ is at hand, a refactorization is needed to compute the minimum-norm solutions in system (2.29).

The stopping criteria in the Newton algorithm are implemented as follows. The iterate $x + h$ is considered to be in $\mathcal{C}_s^\gamma$ if

$$\begin{aligned}&[\text{for all } i \in \mathcal{A}_\gamma(x): \; |r_i + (A^T h)_i| \le \gamma + \tau\,] \;\text{ and}\\ &[\text{for all } i \notin \mathcal{A}_\gamma(x): \; s_{\gamma i} \cdot (r_i + (A^T h)_i) > \gamma - \tau\,].\end{aligned}$$

Here, $\tau \approx O(\varepsilon_M \|A\|_\infty \|x\|_\infty)$ is used to take into account effects of rounding errors; ε_M denotes unit roundoff of the computer. We refer to the subroutine that implements the algorithm as QPASL1. With the exception of some internal tolerance parameters (e.g., tolerances used for numerical checks for zero) QPASL1 does not allow any control over the execution of the algorithm. Hence, all the results reported in this study were obtained under identical algorithmic choices. Further implementation details are given in [16].

4.1. Test problems. We generate test problems using ideas described in [1, 6, 14].

A symmetric positive definite matrix Q is generated as $Q = M^T M$, where $M = D^{1/2} Y$ and $Y = I - (2/\|y\|_2) y y^T$ for some vector $y \in \Re^m$ randomly generated in the interval $(-1, 1)$. The matrix D is diagonal with components d_i:

$$\log d_i = \frac{(i-1)}{(n-1)} ncond \quad \text{for } i = 1, \dots, m.$$

It is easy to verify that *ncond* specifies the condition number of the matrix Q. The matrix A is obtained as the Cholesky factor of Q. This implies that A is triangular, and it is easy to recover the dual optimal solution from the generated "residual" vector r using (2.3).

The components of vectors y and r are generated simultaneously in accordance with a randomly generated sign vector s as follows.

```
for $i = 1 : m$ do
    Generate $\mu$ uniformly in $(-1, 1)$
    if $|m \cdot \mu| < nb$ then
        $s_i = (-1)^{i-1}$
        Generate $\nu$ uniformly in $(0, 1)$
        $r_i = s_i 10^{-\nu \cdot ndeg}$
    else
        $y_i = \mu$
        $r_i = 0$
        $s_i = 0$
    endif
end
```

To introduce near-degeneracy, we use the following identity to define r_i if $s_i = 1$ or -1:

$$r_i = s_i 10^{-\nu \cdot ndeg}.$$

Near-degeneracy is turned off by choosing $ndeg = 1$. Furthermore, the parameter nb in the above procedure is chosen as a fraction of m. Knowing r, x is computed from definition (2.3) by solving the linear system

$$A^T x = r.$$

Finally, using the necessary condition for a minimizer (2.17) of F_γ we obtain b from the identity:

$$b = Ay - x.$$

4.2. Competing algorithms. The main competitors of the proposed algorithm are active set methods and interior point methods.

Active set methods choose a subset of the set of variables to be fixed at their lower and upper bounds. The resulting quadratic problem is solved over the free variables. The algorithm generates a descent direction keeping the variables in the active set fixed at their bounds, and performs a line search restricted by the largest step that can be used before one of the free variables reaches a bound. This scheme is repeated until a unit step length is found. At the end of this phase the Karush–Kuhn–Tucker optimality conditions are checked at the candidate point. If there is a variable which fails to satisfy the optimality conditions, it is removed from the active set. The algorithm repeats by solving a new quadratic problem over the updated set of free variables. The software system LSSOL contains a numerically stable and efficient implementation of the active set algorithm [5].

In [14], Moré and Toraldo propose a modification of the active set algorithm. The modification consists of taking projected gradient steps starting from a point obtained from solving the quadratic problem over the free variables as described above. This way, the proposed algorithm is able to make bigger changes to the active set than the original active set algorithm which makes a single change at a time. Unfortunately, an implementation of this algorithm was not available for comparison.

Our algorithm makes significant changes to the active set at each iteration and also when γ is reduced. In this regard, it is closer to the Moré–Toraldo algorithm than the pure active set strategy.

In [6], Han, Pardalos, and Ye develop a primal-dual potential reduction algorithm for bound constrained quadratic programming problems. The main computational effort in their algorithm is the solution of a linear system of the form

$$(I + R^T D^{-1} R)p = g,$$

where R is an $m \times n$ matrix, D is a diagonal $n \times n$ positive definite matrix, and p and g are m-vectors. As this algorithm was simple to program, we developed an efficient implementation making extensive use of BLAS routines for comparison to QPASL1. We refer to this code as HPY.

In [1], Coleman and Hulbert propose a superlinearly convergent Newton algorithm for bound constrained quadratic programs with unit bounds. The main effort in this algorithm is also the solution of a linear system

$$(|Y| + R^{1/2} H R^{1/2})v = g,$$

where Y is a diagonal matrix with nonzero entries, R is a nonsingular matrix, and H is the matrix of the quadratic term in [BCQP]. Clearly, both linear systems have a structure similar to (3.6). The algorithm by Coleman and Hulbert also uses a one-dimensional search which is similar to that described in section 3.1.2. However, in the algorithms of [6] and [1] a numerical refactorization needs to be performed at each iteration, whereas we only perform a refactorization when it is cheaper or numerically advisable to so. Hence, our average iteration is cheaper than any iteration of these algorithms. An implementation of the Coleman–Hulbert algorithm is not available for comparison. However, a close inspection of the results of [1] reveals that our algorithm uses consistently much smaller numbers of iterations to solve test problems with similar characteristics. To give an example, the Coleman–Hulbert algorithm requires between 10.8 and 17.0 iterations (varying *lcnd* and *ndeg*) on the average for m = 100, whereas our algorithm only requires between 3.8 and 9.6 for the same size for a similar degree of accuracy.

4.3. Initialization. We tested both QPASL1 and LSSOL with different starting points based on the recommendation of an anonymous referee. For LSSOL, we use the following starting points: (1) we choose a starting point y^0 as $y^0_j = 0$ for all $j = 1, \dots, m$; (2) we compute $\bar{y} = Q^{-1}d$ and select the initial point as

$$y_i = \begin{cases} -1 & \text{if } \bar{y}_i \le -1, \\ 1 & \text{if } \bar{y}_i \ge +1, \\ \bar{y}_i & \text{otherwise.} \end{cases}$$

For QPASL1 we also use two different starting points. The first starting point is computed as follows. We fix a value of γ and use the following procedure, based on treating the objective function as

$$\frac{1}{2\gamma} r^T(x) r(x) + b^T x + \frac{1}{2} x^T x + \frac{1}{2} b^T b.$$

The necessary condition for a minimizer is

$$(AA^T + \gamma I)x = -\gamma b.$$

We compute a solution x to the above linear system and use $x^0 = x$. This is referred to as the *least squares starting point.* The second starting point is inspired by the second starting point used for LSSOL. We fix a value of γ and compute $\bar{y} = Q^{-1}d$. Then we set

$$s_i = \begin{cases} -1 & \text{if } \bar{y}_i \le -1, \\ 1 & \text{if } \bar{y}_i \ge +1, \\ 0 & \text{otherwise.} \end{cases}$$

We compute x_0 as the solution to the system

$$(AWA^T + \gamma I)x = -\gamma(As + b),$$

where W is the diagonal matrix associated with s.

For HPY we use the initial point suggested in [6].

4.4. Numerical results. In this section we report our numerical experience with a Fortran 77 implementation of the new algorithm, which does not exploit sparsity. We have three goals when we perform numerical experiments. The first is to examine the growth in solution time and iteration count of the new algorithm as the problem size is increased. The second is to test the numerical accuracy of the algorithm. The third is to estimate the relative standing of the algorithm vis-á-vis other software systems. We compare our results to a library routine, E04NCF, from the NAG subroutine library. E04NCF is based on LSSOL from the Stanford Systems Optimization Library. We also offer comparisons with our own implementation of the interior point algorithm of Han, Pardalos, and Ye [6].

Below we report the results of the following experiments:

1. The effect of near-degeneracy.
2. The effect of the condition number.
3. The effect of the number of variables at their bounds at the optimal solution.
4. The impact of the problem size.

We solve 10 problems of each size. The parameter nb is kept at the value $m/2$ unless otherwise indicated. The tests were performed on a SPARC 4 Workstation running Solaris with the `-O` switch of the F77 compiler. In all tables below, each line reports the average over 10 problems of the following QPASL1 statistics: number of iterations, run time in CPU seconds, number of refactorizations, and number of γ reductions. The column “it” refers to the total number of iterations of the Newton method and the total number of optimality checks during the execution of the algorithm. The column “rf” refers to the total number of refactorizations in connection with the computations of the factors L and D. The column “rd” refers to the total number of times the optimality check was performed and/or γ was reduced. The heading QPASL1(2) refers to the second starting point for QPASL1, whereas QPASL1(1) refers to the least squares starting point. Similarly, LSSOL(2) indicates the second initial point, while LSSOL(1) refers to the use of the origin as the initial point. The columns under the heading LSSOL contain the run time statistics of LSSOL averaged over 10 problems for each line. All runs with QPASL1, LSSOL, and HPY were performed using default parameters, i.e., no fine tuning of the codes was done for any test problem.

QPASL1 is stopped when the relative duality gap

$$(H(\hat{y}_0) - F(\hat{x}_0))/(1 + F(\hat{x}_0))$$

is less than or equal to 10^{-8} and the primal feasibility measure $\|\hat{y}_0\|_\infty$ is less than or equal to $1 + \epsilon_y$ with $\epsilon_y = 10^{-5}$. The final accuracy obtained in QPASL1 is measured using the accuracy in the objective function and the primal solution with respect to the known optimal value and optimal solution vector. The accuracy in the optimal value is checked using

$$q_1 = \frac{H(y_0) - H(\hat{y}_0)}{H(y_0)},$$

where $H(y_0)$ is the known optimal value, and the accuracy in the solution is checked using

$$q_2 = \|y_0 - \hat{y}_0\|_2 / \|y_0\|_2,$$

where y_0 and $\hat{y}_0$ denote the known and computed optimal values, respectively. In all test problems solved in this study, we have

$$10^{-16} \leq q_1 \leq 10^{-12}.$$

Depending on the conditioning of the problem, we also obtain

$$10^{-12} \leq q_2 \leq 10^{-9}.$$

This indicates that we achieve high accuracy in the computed optimal solution. Regarding other parameters, we use $\gamma^0 = 10^{-3}$ as the starting value of γ, and $\beta = 1/100$.

LSSOL yields objective function values accurate to machine precision in all cases. For HPY, the quantities q_1 and q_2 vary as follows:

$$10^{-9} \leq q_1 \leq 10^{-8},$$

$$10^{-8} \leq q_2 \leq 10^{-5}.$$

4.4.1. Experiment 1: The effect of near-degeneracy. In Table 4.1 we give computational results obtained when the near-degeneracy parameter *ndeg* is increased.

We make the following observations.

- QPASL1 is competitive with LSSOL for small values (1,3) of the parameter *ndeg*, whereas for larger values it loses its advantage. It is also substantially faster than HPY.
- The iteration number of QPASL1 remains very small and almost constant with the increasing problem size for small values of *ndeg*.
- The parameter *ndeg* has almost no effect on the performance of LSSOL.
- The two starting points for QPASL1 tend to perform similarly when near-degeneracy is increased.

The reason for the deterioration in performance of QPASL1 for larger values of *ndeg* is precisely related to Corollary 2.1. It is shown in this corollary that the value of

Table 4.1

Solution statistics of QPASL1 and LSSOL when near-degeneracy is increased.

$m, lcnd, ndeg$	QPASL1(2)				QPASL1(1)			
	it	rf	rd	CPU	it	rf	rd	CPU
100, 1, 1	3.8	2	1	0.4	4.1	3	1	0.6
100, 1, 3	5.2	2.1	1.1	0.5	5.9	3.1	1.1	0.7
100, 1, 6	9.6	3.1	2.1	1.1	10.3	3.4	2.1	1.3
200, 1, 1	4.2	2	1	2.3	5.1	3	1	4.0
200, 1, 3	5.1	2.1	1.1	3.0	6	3.1	1.1	4.8
200, 1, 6	9.5	3.1	2.1	6.9	10.2	3.3	2.1	8.6
300, 1, 1	4	2	1	6.8	3.8	3	1	13.1
300, 1, 3	4.8	2.2	1.2	8.7	5.6	3.2	1.2	15.2
300, 1, 6	9.3	3.3	2.3	22.1	10.9	3.8	2.3	27.5

$m, lcnd, ndeg$	LSSOL(2)		LSSOL(1)		HPY	
	it	CPU	it	CPU	it	CPU
100, 1, 1	14.5	0.5	50	0.9	18	2.6
100, 1, 3	21.6	0.6	50	0.9	16.9	2.3
100, 1, 6	23.5	0.6	45.7	0.8	14.9	2.0
200, 1, 1	27.8	3.8	100.6	6.0	16	15.9
200, 1, 3	39.9	4.1	100.6	6.0	16.2	15.6
200, 1, 6	46.4	4.4	91.7	5.6	17.5	16.7
300, 1, 1	16	10.1	152.4	19.7	16.8	51.5
300, 1, 3	34.6	10.9	152.4	19.5	18.6	56.9
300, 1, 6	44.5	11.7	140.2	18.7	18.2	55.5

Table 4.2

Solution statistics of QPASL1 and LSSOL when the condition number is increased.

$m, lcnd, ndeg$	QPASL1(2)				QPASL1(1)			
	it	rf	rd	CPU	it	rf	rd	CPU
100, 4, 1	3.8	2	1	0.4	3.9	3.1	1	0.6
100, 8, 1	3.8	2	1	0.4	4.1	3.1	1	0.6
200, 4, 1	4	2	1	2.2	5.2	3	1	4.0
200, 8, 1	8.5	2.2	1	2.9	5.5	3	1	4.0
300, 4, 1	3.9	2	1	6.8	3.8	3	1	12.3
300, 8, 1	3.9	2	1	6.9	4.1	3	1	12.4

$m, lcnd, ndeg$	LSSOL(2)		LSSOL(1)		HPY	
	it	CPU	it	CPU	it	CPU
100, 4, 1	12.8	0.5	50	0.9	14.6	1.9
100, 8, 1	13.9	0.5	49.7	0.8	17.2	2.3
200, 4, 1	32.2	3.9	100.6	5.9	14.9	14.2
200, 8, 1	29.2	3.8	100.2	5.8	17.3	16.6
300, 4, 1	18	10.2	152.6	19.1	17.3	52.9
300, 8, 1	32.4	11.4	152.4	19.1	17	51.9

γ^* is affected by the magnitude of nonzero residuals $r(x_0)$ at the optimal solution x_0. The smaller the residuals, the more γ should be reduced in order to reach the optimal solution. This increases the number of reduction steps and the total number of iterations, thereby causing a degradation in performance.

4.4.2. Experiment 2: The effect of the condition number. In Table 4.2 we summarize the average performance of the three codes when the conditioning parameter *lcond* is increased.

It is observed that all three codes handle problems with increasing condition number equally well.

4.4.3. Experiment 3: The effect of the number of variables at bounds. The number of variables at a bound at an optimal solution can be controlled by varying the parameter *nb*. We do so in this experiment and report the results in Table 4.3.

We notice that the performance of LSSOL improves significantly when *nb* becomes smaller than $m/2$ and worsens when it exceeds that value. This improvement is more marked when the zero starting point is used. A similar improvement occurs with HPY, whereas the opposite is true of QPASL1.

4.4.4. Experiment 4: The effect of the problem size. To illustrate the effect of increasing problem size on the performance of the three codes, we provide some results in Table 4.4.

We notice that LSSOL consumes about 1.5 times more CPU time than QPASL1 as we increase the problem size, while HPY uses approximately 10 times more CPU compared to QPASL1.

5. Summary and concluding remarks. In this paper, we presented a dual approach to strictly convex quadratic programming with unit bounds.

Our dual approach consisted of posing the problem [BCQP] as an unconstrained ℓ_1 minimization problem and approximating this nondifferentiable problem by a smooth Huber problem. The minimizers of the smooth problem define a unique path that converges to the primal-dual optimal solutions as a function of a scalar parameter

Table 4.3
*Solution statistics of QPASL*1 *and LSSOL when nb is varied.*

$m, lcnd, ndeg, nb$	QPASL1(2)				QPASL1(1)			
	it	rf	rd	CPU	it	rf	rd	CPU
$100, 1, 1, m/2$	3.8	2	1	0.4	4.1	3	1	0.6
$100, 1, 1, m/10$	3.7	2	1	0.5	4.2	2	1	0.5
$100, 1, 1, 3m/4$	5	3.1	1	0.4	3.5	3	1	0.5
$200, 1, 1, m/2$	4.2	2	1	2.3	5.1	3	1	4.0
$200, 1, 1, m/10$	4.1	2	1	3.6	6	4.6	2	4.0
$200, 1, 1, 3m/4$	8.1	3	1	2.8	4.5	3	1	3.1
$300, 1, 1, m/2$	4	2	1	6.8	3.8	3	1	13.1
$300, 1, 1, m/10$	3.9	2	1	11.4	4.2	2	1	12.7
$300, 1, 1, 3m/4$	9.3	3	1	8.8	3.9	3	1	10.0

$m, lcnd, ndeg, nb$	LSSOL(2)		LSSOL(1)		HPY	
	it	CPU	it	CPU	it	CPU
$100, 1, 1, m/2$	14.5	0.5	50	0.9	18	2.6
$100, 1, 1, m/10$	13	0.3	10.6	0.21	13.6	1.9
$100, 1, 1, 3m/4$	14.4	0.4	76.1	1.1	15.6	2.1
$200, 1, 1, m/2$	27.8	3.8	100.6	6.0	16	15.9
$200, 1, 1, m/10$	17.2	1.9	19.4	1.4	14.4	14.0
$200, 1, 1, 3m/4$	28.5	3.0	150.4	7.5	16	15.3
$300, 1, 1, m/2$	16	10.1	152.4	19.7	16.8	51.5
$300, 1, 1, m/10$	26.1	6.19	30	4.5	15.5	47.7
$300, 1, 1, 3m/4$	25.3	9.1	223.3	24.0	16	49.9

Table 4.4
*Solution statistics of QPASL*1 *and LSSOL when the problem size is increased.*

$m, lcnd, ndeg$	QPASL1(2)				QPASL1(1)			
	it	rf	rd	CPU	it	rf	rd	CPU
100, 1, 1	3.8	2	1	0.4	4.1	3	1	0.6
200, 1, 1	4.2	2	1.0	2.3	5.1	3	1	4
300, 1, 1	4	2	1	6.8	3.8	3	1	13.1
400, 1, 1	4.2	2	1	16.0	4.9	3.1	1	29.7
500, 1, 1	4.3	2	1	30.4	5.3	3.1	1	58.7

$m, lcnd, ndeg$	LSSOL(2)		LSSOL(1)		HPY	
	it	CPU	it	CPU	it	CPU
100, 1, 1	14.5	0.5	50	0.9	18	2.6
200, 1, 1	27.8	3.8	100.6	6.0	16	15.9
300, 1, 1	16	10.1	152.4	19.7	16.8	51.5
400, 1, 1	30.8	25.4	203.7	45.1	18.8	139.4
500, 1, 1	40.8	48.1	253	85.9	20.2	288.4

γ. This suggested a continuation algorithm, where we follow this path to arrive at primal-dual optimal solutions.

On the theoretical front, we established an extrapolation property of the solution path and a constant sign property (for sufficiently small γ), which formed the pillar of finite convergence of the continuation algorithm. We also gave a finite Newton algorithm to solve the Huber problems.

On the practical front, we developed a stable and efficient implementation of the algorithm for dense problems. We compared our results to an established software system for quadratic programming, LSSOL, and to more recent algorithms for [BCQP]. The following picture emerged from our experiments. The new algorithm is competitive with a state-of-the-art implementation of active set methods for problems with low degree of near-degeneracy. It also handles problems with increasing condition number very well. It is also substantially faster than an interior point algorithm proposed for [BCQP].

Finally we remark that the duality framework of section 2 can be easily extended to problems where bounds are different from unity and/or where one of the bounds is missing; see [2]. Nonunit bounds simply change the slope of the nondifferentiable function arising in the dual problem. By way of illustration, consider the following case:

$$\begin{array}{rl} \min\limits_{y} & H(y) = -d^T y + \frac{1}{2} y^T Q y \\ \text{subject to} & l \leq y. \end{array}$$

The nondifferentiable dual problem corresponding to the program above is

$$\text{minimize } F(x) \;\equiv\; \sum_{i=1}^{m} \rho_i(r_i(x)) + \frac{1}{2} x^T x + b^T x + \frac{1}{2} b^T b,$$

where

$$\rho_i(r_i) = \begin{cases} l_i r_i & \text{if } r_i \geq 0 \\ \infty & \text{otherwise,} \end{cases}$$

and the vectors r and b are defined as in section 2. The nondifferentiable function ρ can be approximated by the following smooth Huber function

$$\psi_\gamma(r_i) = \begin{cases} l_i r_i - \frac{1}{2}\gamma & \text{if } r_i \geq \gamma, \\ \frac{1}{2\gamma} r_i^2 & \text{if } r_i < \gamma, \end{cases}$$

for some scalar parameter $\gamma > 0$. The properties and the algorithm derived in this paper apply to the above approximation as well.

Acknowledgments. The helpful comments and suggestions of two anonymous referees are gratefully acknowledged.

REFERENCES

[1] T. Coleman and L. Hulbert, *A globally and superlinearly convergent algorithm for quadratic programming with simple bounds*, SIAM J. Optim., 3 (1993), pp. 298–321.

[2] O. Edlund, *Private communication*, Luleå University of Technology, Luleå, Sweden, 1997.

[3] R. Fletcher and M. J. D. Powell, *On the modification of LDL^T factorizations*, Math. Comp., 28 (1974), pp. 1067–1087.

[4] W. M. Gentleman, *Least squares computation by Givens transformations without square roots*, J. Inst. Math. Appl., 12 (1973), pp. 329–336.

[5] P. E. Gill, S. J. Hammarling, W. Murray, M. A. Saunders, and M. H. Wright, *User's Guide for LSSOL* (*Version* 1.0): *A Fortran Package for Constrained Linear Least Squares and Convex Quadratic Programming*, Technical Report SOL 86-1, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA, 1986.

[6] C.-G. Han, P. Pardalos, and Y. Ye, *Computational aspects of an interior point algorithm for quadratic programming problems with box constraints*, in Large-Scale Numerical Optimization, T. F. Coleman and Y. Li, eds., SIAM, Philadelphia, PA, 1990, pp. 92–112.

[7] P. Huber, *Robust Statistics*, John Wiley, New York, 1981.

[8] W. Li and J. Swetits, *A Newton method for convex regression, data smoothing, and quadratic programming with bounded constraints*, SIAM J. Optim., 3 (1993), pp. 466–488.

[9] W. Li and J. Swetits, *A new algorithm for solving strictly convex quadratic programs,* SIAM J. Optim., 7 (1997), pp. 595–619.

[10] K. Madsen and H. B. Nielsen, *Finite algorithms for robust linear regression*, BIT, 30 (1990), pp. 682–699.

[11] K. Madsen, H. B. Nielsen, and M.Ç. Pinar, *A new finite continuation algorithm for linear programming*, SIAM J. Optim., 6 (1996), pp. 600–616.

[12] O. L. Mangasarian, *Normal solution of linear programs*, Math. Programming Stud., 22 (1984), pp. 206–216.

[13] O. L. Mangasarian and R. R. Meyer, *Nonlinear perturbation of linear programs*, SIAM J. Control Optim., 17 (1979), pp. 745–752.

[14] J. Moré and G. Toraldo, *Algorithms for bound constrained quadratic programming problems*, Numer. Math., 55 (1989), pp. 377–400.

[15] H. B. Nielsen, *AAFAC: A Package of Fortran* 77 *Subprograms for Solving $A^TAx = c$*, Report NI-90-11, Institute of Mathematical Modelling, Numerical Analysis Group, Technical University of Denmark, DK-2800 Lyngby, Denmark, 1990.

[16] H. B. Nielsen, *Implementation of a Finite Algorithm for Linear ℓ_1 Estimation*, Report NI-91-01, Institute of Mathematical Modelling, Numerical Analysis Group, Technical University of Denmark, DK-2800 Lyngby, Denmark, 1991.

SIAM J. OPTIM.
Vol. 9, No. 1, pp. 84–111

ILL-CONDITIONING AND COMPUTATIONAL ERROR IN INTERIOR METHODS FOR NONLINEAR PROGRAMMING[*]

MARGARET H. WRIGHT[†]

Abstract. Ill-conditioning has long been regarded as a plague on interior methods, but its damaging effects have rarely been documented. In fact, implementors of interior methods who ignore warnings about the dire consequences of ill-conditioning usually manage to compute accurate solutions. We offer some insight into this seeming contradiction by analyzing ill-conditioning within a primal-dual method in which the full, usually well-conditioned primal-dual matrix is transformed to a "condensed," inherently ill-conditioned matrix M_{pd}. We show that ill-conditioning in the exact condensed matrix closely resembles that known for the primal barrier Hessian, and then examine the influence of cancellation in the computed constraints.

Using the structure of M_{pd}, various bounds are obtained on the absolute accuracy of the computed primal-dual steps. Without cancellation, the portion of the computed x step in the small space of M_{pd} (a subspace close to the null space of the Jacobian of the active constraints) has an absolute error bound comparable to machine precision, and its large-space component has a much smaller error bound. With cancellation (the usual case), the absolute error bounds for both the small- and large-space components of the computed x step are comparable to machine precision. In either case, the absolute error bound for the computed multiplier steps associated with active constraints is comparable to machine precision; the computed multiplier steps for inactive constraints, although converging to zero, retain (approximately) full relative precision.

Because of errors in forming the right-hand side, the absolute error in the computed solution of the full, well-conditioned primal-dual system is shown to be comparable to machine precision. Thus, under quite general conditions, ill-conditioning in M_{pd} does not noticeably impair the accuracy of the computed primal-dual steps. (A similar analysis applies to search directions obtained by direct solution of the primal Newton equations.)

Key words. interior method, primal-dual method, barrier method, constrained optimization

AMS subject classifications. 65K05, 90C30

PII. S1052623497322279

1. Introduction.

The perplexing issue of extreme but mystifyingly harmless ill-conditioning has haunted interior methods for more than 25 years. It is widely known that the Hessian matrices in the original interior method—the logarithmic barrier method developed, analyzed, and popularized by Fiacco and McCormick in the late 1960s [10]—suffer from structural ill-conditioning that worsens as the solution is approached; see, for example, [23, 25, 34]. Several authors have described how to compute accurate search directions via alternative formulations that finesse the ill-conditioning; see, for example, [20, 34]. Yet practitioners who apply standard linear-equation solvers directly to the ill-conditioned system have consistently experienced very few harmful effects. In fact, given the ill-conditioning of the matrices, the computed solutions in interior methods are almost always much more accurate than they should be (see, for example, [33, 27]).

Previous publications about this seeming anomaly include work by several authors showing that fully accurate solutions can be obtained when specific direct methods, such as the Cholesky or symmetric indefinite factorizations, are applied to certain matrices from interior methods. In particular, ill-conditioned systems from linear programming and linear complementarity problems are analyzed in detail in

[*] Received by the editors May 1, 1997; accepted for publication (in revised form) April 14, 1998; published electronically October 30, 1998.

http://www.siam.org/journals/siopt/9-1/32227.html

[†] Bell Laboratories, Murray Hill, NJ 07974 (mhw@research.bell-labs.com).

[37, 38, 39], including the significance of rounding errors; and symmetrized full primal-dual systems for nonlinear problems are considered in [14, 12]. In contrast with earlier research, this paper looks at the generic process of forming the ill-conditioned "condensed" primal-dual system and solving it using *any* backward-stable numerical method. An element in our analysis is the effect of cancellation error in the active constraints. (The same approach applies in an obvious way to the primal barrier Hessian.) Selected subsequent related work is briefly summarized in section 8.

1.1. The role of ill-conditioning. When solving $Mx = b$ with a nonsingular but ill-conditioned matrix M, the relative sensitivity of the solution is bounded by—in the worst case, equal to—the condition number of M multiplied by the relative perturbations in b or M. In certain situations, however, special properties of M or b ensure much more favorable bounds. Definitions of alternative condition numbers have been discussed in the literature for many years; see [30, 31]. In [5], "effectively well conditioned" linear systems are defined in which properties of the right-hand side are central; also see [7, 1].

Let $M = U\Sigma V^T$ be the singular value decomposition (SVD) of an $n \times n$ nonsingular matrix M, where U and V are orthogonal and Σ is a nonnegative diagonal matrix. Our interest is in the case when the singular values of M fall into exactly two groups with the property that the two submatrices of Σ are separately much better conditioned than M itself. This special structure is not contrived: it is precisely that of the primal barrier Hessian (see [34]) and the condensed matrix defined in section 3. The consequences of this structure are explored in sections 2.2–2.4.

1.2. Notation. Unless otherwise indicated, all norms are two-norms; the singular values $\{\sigma_i\}$ of a matrix are ordered so that $\sigma_i \ge \sigma_{i+1}$, and similarly for the eigenvalues. We use the following notation; see [26].

DEFINITION 1.1. (Order notation.) *Let ϕ be a scalar, vector, or matrix function of a positive variable h, let p be fixed, and let κ_u and κ_l denote constants.*

- *If there exists $\kappa_u > 0$ such that $\|\phi\| \le \kappa_u h^p$ for all sufficiently small h, we write $\phi = O(h^p)$.*
- *If there exists $\kappa_l > 0$ such that $\|\phi\| \ge \kappa_l h^p$ for all sufficiently small h, we write $\phi = \Omega(h^p)$.*
- *If there exist $\kappa_l > 0$ and $\kappa_u > 0$ such that $\kappa_l h^p \le \|\phi\| \le \kappa_u h^p$ for all sufficiently small h, we write $\phi = \Theta(h^p)$.*

2. Solving an ill-conditioned linear system.

2.1. Background. The effects of changes in b and M on the exact solution of $Mx = b$ are well known (see, for example, [18, 22]). Let $\tilde{x}$ denote the exact solution of $M\tilde{x} = b + \Delta b$, and let $\Delta x = \tilde{x} - x$. Then

$$\|\Delta x\| \le \|M^{-1}\|\,\|\Delta b\| \ \text{ and } \ \frac{\|\Delta x\|}{\|x\|} \le \operatorname{cond}(M)\,\frac{\|\Delta b\|}{\|b\|}, \tag{2.1}$$

where $\operatorname{cond}(M) = \|M\|\,\|M^{-1}\|$. Equality is achieved in the first inequality of (2.1) when $\|M^{-1}\Delta b\| = \|M^{-1}\|\,\|\Delta b\|$; equality holds in the second when Δb satisfies this same condition and $\|M^{-1}b\| = \|b\|/\|M\|$.

When the matrix changes by ΔM, the exact solution $\tilde{x}$ of the perturbed system satisfies

$$(M + \Delta M)\tilde{x} = Mx = b, \quad \text{or} \quad \tilde{x} - x = -(M + \Delta M)^{-1}\Delta M\tilde{x}. \tag{2.2}$$

If we ignore second-order terms (which is acceptable as long as $\mathrm{cond}(M) \approx \mathrm{cond}(M + \Delta M)$), an approximation to (2.2) is satisfied by $\Delta x \approx \tilde{x} - x$:

$$M\,\Delta x = -\Delta M\,x \quad \text{or} \quad \Delta x = -M^{-1}\Delta M\,x, \tag{2.3}$$

which gives the bounds

$$\|\Delta x\| \le \|M^{-1}\|\,\|\Delta M\|\,\|x\| \quad \text{and} \quad \frac{\|\Delta x\|}{\|x\|} \le \mathrm{cond}(M)\,\frac{\|\Delta M\|}{\|M\|}. \tag{2.4}$$

Equality can hold in these relations for any vector b in (2.2) (see [32] and [22, p. 133]).

Since the normwise bounds (2.1) and (2.4) can be achieved, when M is ill-conditioned we tend to expect substantial *relative* inaccuracy in the computed solution, i.e., a "large" value of $\|\Delta x\|/\|x\|$. In section 2.2, however, we show that more favorable bounds apply to the accuracy of certain parts of the solution when M has a particular structure.

In some circumstances it is convenient to work with perturbations only in the matrix rather than in the right-hand side. The following lemma, based on Theorem III.2.16 of [31], characterizes the effect of folding a perturbation from the right-hand side into the matrix.

LEMMA 2.1. *Let x be the solution of $Mx = b$ and $\tilde{x}$ be the solution of $M\tilde{x} = \tilde{b}$, where b and $\tilde{b}$ are nonzero and M is nonsingular. Then $\tilde{x}$ is also the solution of $(M + E)\tilde{x} = b$, where E is the rank-one matrix $(b - \tilde{b})\tilde{x}^T/\|\tilde{x}\|^2$. For this choice of E,*

$$\frac{\|E\|}{\|M\|} \le \frac{\|\tilde{b} - b\|}{\|\tilde{b}\|}. \tag{2.5}$$

Beginning with section 3, our interest will be entirely in symmetric systems. The following theorem, a simplified version of Theorem 3 of [2], shows that, when M is symmetric, we are allowed to consider only symmetric perturbations.

THEOREM 2.1. *Suppose that M is symmetric and nonsingular and that the nonzero vector z satisfies $(M+E)z = b$ for some matrix E. Then there is a* symmetric *perturbation F satisfying $\|F\| = \|E\|$ such that $(M + F)z = b$.*

With the most widely used numerical methods, the computed solution of a linear system is typically the exact solution of a nearby problem; see, e.g., [18, 22]. In particular, when solving the symmetric system $Mx = b$ in finite precision with any backward-stable method, the computed solution $\tilde{x}$ is the exact solution of a nearby system involving a perturbed symmetric matrix $\widetilde{M}$:

$$\widetilde{M}\tilde{x} = b, \quad \text{where} \quad \widetilde{M} = M + \Delta_{\mathrm{s}}M \quad \text{and} \quad \Delta_{\mathrm{s}}M = (\Delta_{\mathrm{s}}M)^T. \tag{2.6}$$

The subscript "s" indicates that the perturbation arises entirely from numerical solution; symmetry can be assumed because of Theorem 2.1. For the most common backward-stable methods performed on a machine with unit roundoff $\mathbf{u}$ (see (4.1)), the perturbation $\Delta_{\mathrm{s}}M$ satisfies

$$\|\Delta_{\mathrm{s}}M\| \;\le\; \mathbf{u}\gamma_n\|M\|, \tag{2.7}$$

where γ_n is a function involving a low-order polynomial in n and characteristics of M (such as the growth factor).

Characterizations of γ_n are known under various conditions for (i) the Cholesky factorization when M is sufficiently positive definite (see, e.g., [21]); (ii) the symmetric indefinite factorization with partial pivoting (see [22]); (iii) Gaussian elimination with partial pivoting (see, e.g., [22]); (iv) the partial Cholesky factorization of [13]; and (v) the modified Cholesky factorizations of [16] and [28] (see [6]). In the absence of pathologies such as extreme growth, γ_n is of "reasonable" size for all these methods in the sense that $\mathbf{u}\gamma_n \ll 1$.

2.2. Structured ill-conditioning. We now consider the effects of perturbations when M is nonsingular and ill-conditioned, but its singular values split into two well-behaved subgroups, as follows:

$$M = U\Sigma V^T = \begin{pmatrix} U_{\scriptscriptstyle L} & U_{\scriptscriptstyle S} \end{pmatrix} \begin{pmatrix} \Sigma_{\scriptscriptstyle L} & 0 \\ 0 & \Sigma_{\scriptscriptstyle S} \end{pmatrix} \begin{pmatrix} V_{\scriptscriptstyle L}^T \\ V_{\scriptscriptstyle S}^T \end{pmatrix}, \tag{2.8}$$

where U and V are orthogonal and Σ is a positive diagonal matrix with diagonal elements in decreasing order. (The subscripts "$\scriptstyle L$" and "$\scriptstyle S$" should be interpreted as signifying "large" and "small".)

Let $\hat{m}$ denote the dimension of $\Sigma_{\scriptscriptstyle L}$; we assume that $n > 1$, $0 < \hat{m} < n$, and $\sigma_{\hat{m}} > \sigma_{\hat{m}+1}$. By definition of the two-norm,

$$\|M\| = \|\Sigma_{\scriptscriptstyle L}\| \quad \text{and} \quad \|M^{-1}\| = \|\Sigma_{\scriptscriptstyle S}^{-1}\|, \quad \text{so that} \quad \operatorname{cond}(M) = \|\Sigma_{\scriptscriptstyle L}\|\,\|\Sigma_{\scriptscriptstyle S}^{-1}\|.$$

The results to be derived are of interest when $\Sigma_{\scriptscriptstyle L}$ and $\Sigma_{\scriptscriptstyle S}$ are individually much better conditioned than M itself, i.e.,

$$\frac{\sigma_1}{\sigma_{\hat{m}}} \ll \frac{\sigma_1}{\sigma_n} \quad \text{and} \quad \frac{\sigma_{\hat{m}+1}}{\sigma_n} \ll \frac{\sigma_1}{\sigma_n}. \tag{2.9}$$

Note that this property does not imply a large gap between $\sigma_{\hat{m}}$ and $\sigma_{\hat{m}+1}$, nor that the diagonal elements are comparable within each of $\Sigma_{\scriptscriptstyle L}$ and $\Sigma_{\scriptscriptstyle S}$. For example, (2.9) is satisfied for $n = 4$ by the singular values 1, 10^{-4}, 10^{-5}, and 10^{-10} with $\hat{m} = 1$, 2, or 3.

To begin our analysis, we express b in terms of $U_{\scriptscriptstyle L}$ and $U_{\scriptscriptstyle S}$, and x in terms of $V_{\scriptscriptstyle L}$ and $V_{\scriptscriptstyle S}$:

$$b = b_{\scriptscriptstyle L} + b_{\scriptscriptstyle S} = U_{\scriptscriptstyle L}\beta_{\scriptscriptstyle L} + U_{\scriptscriptstyle S}\beta_{\scriptscriptstyle S} \quad \text{and} \quad x = x_{\scriptscriptstyle L} + x_{\scriptscriptstyle S} = V_{\scriptscriptstyle L}\xi_{\scriptscriptstyle L} + V_{\scriptscriptstyle S}\xi_{\scriptscriptstyle S}. \tag{2.10}$$

At a slight risk of ambiguity, the same subscripts denote the representations of b and x in terms of different sets of vectors; however, the right-hand side is always associated with U and the solution with V. The *large space* of M means the subspace of vectors spanned by the columns of $V_{\scriptscriptstyle L}$ or $U_{\scriptscriptstyle L}$; the choice will always be clear from context. Similarly, the *small space* of M refers to the subspace of vectors spanned by the columns of $V_{\scriptscriptstyle S}$ or $U_{\scriptscriptstyle S}$. Either $b_{\scriptscriptstyle L}$ or $\beta_{\scriptscriptstyle L}$ may be called the "large-space part" of b, and either $b_{\scriptscriptstyle S}$ or $\beta_{\scriptscriptstyle S}$ is the "small-space part" of b, with a similar terminology for x.

Because U and V are orthogonal, we know that

$$\|b\|^2 = \|b_{\scriptscriptstyle L}\|^2 + \|b_{\scriptscriptstyle S}\|^2, \quad \|b_{\scriptscriptstyle L}\|^2 = \|\beta_{\scriptscriptstyle L}\|^2, \quad \text{and} \quad \|b_{\scriptscriptstyle S}\|^2 = \|\beta_{\scriptscriptstyle S}\|^2,$$

with analogous relations involving x and ξ. Substituting the partitioned SVD of M (2.8) and the representations (2.10) into the equation $Mx = b$, we have

$$\Sigma\xi = \begin{pmatrix} \Sigma_{\scriptscriptstyle L}\xi_{\scriptscriptstyle L} \\ \Sigma_{\scriptscriptstyle S}\xi_{\scriptscriptstyle S} \end{pmatrix} = \begin{pmatrix} \beta_{\scriptscriptstyle L} \\ \beta_{\scriptscriptstyle S} \end{pmatrix} = \beta,$$

giving the bounds

$$\|b_L\| \le \|\Sigma_L\|\,\|x_L\| \quad\text{and}\quad \|b_S\| \le \|\Sigma_S\|\,\|x_S\| \tag{2.11}$$

as well as the relations $\xi_L = \Sigma_L^{-1}\beta_L$ and $\xi_S = \Sigma_S^{-1}\beta_S$, so that

$$\|x_L\| \le \|\Sigma_L^{-1}\|\,\|b_L\| \text{ and } \|x_S\| \le \|\Sigma_S^{-1}\|\,\|b_S\|. \tag{2.12}$$

2.3. Perturbations in the right-hand side. First we examine the effects of perturbing the right-hand side by Δb, and consider Δx satisfying $M(x+\Delta x) = b + \Delta b$. Expressing Δb and Δx in terms of the partitioned U and V as in (2.10) gives

$$\begin{pmatrix} \Delta\xi_L \\ \Delta\xi_S \end{pmatrix} = \begin{pmatrix} \Sigma_L^{-1}\Delta\beta_L \\ \Sigma_S^{-1}\Delta\beta_S \end{pmatrix},$$

so that

$$\|\Delta x_L\| \le \|\Sigma_L^{-1}\|\,\|\Delta b_L\| \quad\text{and}\quad \|\Delta x_S\| \le \|\Sigma_S^{-1}\|\,\|\Delta b_S\|. \tag{2.13}$$

Assuming that $b_L \ne 0$ and $b_S \ne 0$, these relations can be combined with the bounds on $\|b_L\|$ and $\|b_S\|$ from (2.11) to produce two distinct bounds for the relative perturbations in x_L and x_S, both involving factors substantially less than $\text{cond}(M)$:

$$\frac{\|\Delta x_L\|}{\|x_L\|} \le \|\Sigma_L\|\,\|\Sigma_L^{-1}\|\,\frac{\|\Delta b_L\|}{\|b_L\|} \quad\text{and}\quad \frac{\|\Delta x_S\|}{\|x_S\|} \le \|\Sigma_S\|\,\|\Sigma_S^{-1}\|\,\frac{\|\Delta b_S\|}{\|b_S\|}.$$

These relations are of interest because they show, in effect, that the relative perturbations in the solution within the column spaces of V_L and V_S separate according to the portions of b and Δb in the column spaces of U_L and U_S: the maximum relative change in x_L depends only on $\text{cond}(\Sigma_L)$ and the relative change in b_L, and similarly for x_S.

In general, however, we may not know the size of the relative perturbations in b_L and b_S separately; the only available bounds may be on the overall $\|\Delta b\|$ and $\|b\|$. In this case, there is a dramatic difference in the bounds for perturbations in x_L and x_S. Beginning with the first bound in (2.13) and applying the two inequalities $\|\Delta b_L\| \le \|\Delta b\|$ and $\|b\| \le \|\Sigma_L\|\,\|x\|$, we obtain

$$\frac{\|\Delta x_L\|}{\|x\|} \le \|\Sigma_L^{-1}\|\,\|\Sigma_L\|\,\frac{\|\Delta b\|}{\|b\|}. \tag{2.14}$$

Thus, when Σ_L is much better conditioned than M, the change in x_L compared to $\|x\|$ is guaranteed to be *much smaller* than $\text{cond}(M)$ times the relative perturbation in b. In contrast, since the bound in (2.13) on $\|\Delta x_S\|$ includes $\|\Sigma_S^{-1}\|$, the worst-case bound on the relative change in x_S compared to $\|x\|$ includes the condition of M rather than of Σ_S.

2.4. Perturbations in the matrix. When the matrix M changes, we use the first-order approximation (2.3), $\Delta x = -M^{-1}\Delta M\,x$. To analyze this relation, we consider each column of ΔM as a linear combination of columns of U and define a matrix B such that $\Delta M = UBV^T$, where U and V are the orthogonal matrices from the SVD of M. Thus, $\|\Delta M\| = \|B\|$ and

$$B = U^T\Delta M V, \quad\text{with}\quad B = \begin{pmatrix} B_L \\ B_S \end{pmatrix} = \begin{pmatrix} B_{L1} & B_{L2} \\ B_{S1} & B_{S2} \end{pmatrix}. \tag{2.15}$$

Expressing all quantities in partitioned form and writing $\Delta x = V\Delta\xi$, we have

$$\begin{pmatrix} \Delta\xi_L \\ \Delta\xi_S \end{pmatrix} = -\begin{pmatrix} \Sigma_L^{-1} B_L \xi \\ \Sigma_S^{-1} B_S \xi \end{pmatrix}.$$

It follows that

$$\|\Delta x_L\| \;\le\; \|\Sigma_L^{-1}\|\,\|B_L\|\,\|x\| \;\le\; \|\Sigma_L^{-1}\|\,\|\Delta M\|\,\|x\| \quad \text{and} \tag{2.16}$$

$$\|\Delta x_S\| \;\le\; \|\Sigma_S^{-1}\|\,\|B_S\|\,\|x\| \;\le\; \|\Sigma_S^{-1}\|\,\|\Delta M\|\,\|x\|. \tag{2.17}$$

Relative perturbations in x_L and x_S cannot be developed without further assumptions about the structure of B. However, since $\|\Sigma_L\| = \|M\|$, (2.16) implies that

$$\frac{\|\Delta x_L\|}{\|x\|} \;\le\; \|\Sigma_L^{-1}\|\,\|\Sigma_L\|\,\frac{\|\Delta M\|}{\|M\|},$$

so that the change in x_L relative to x can be blown up compared to the relative perturbation in M only by $\operatorname{cond}(\Sigma_L)$ rather than $\operatorname{cond}(M)$. In contrast, the perturbation in x_S relative to x can in general be blown up by $\operatorname{cond}(M)$.

3. Linear systems in primal-dual methods.

In this section we consider properties of the condensed matrices arising in primal-dual methods for constrained optimization, and develop connections with the structure just described.

3.1. Inequality-constrained optimization.

Consider an optimization problem with only inequality constraints:

$$\underset{x\in\mathcal{R}^n}{\text{minimize}}\; f(x) \quad \text{subject to} \quad c_j(x) \ge 0,\;\; j = 1,\dots,m, \tag{3.1}$$

where f and $\{c_j\}$ are smooth. We use the following notation: $g(x)$ and $H(x)$ are the gradient and Hessian matrix of $f(x)$; $a_j(x)$ and $H_j(x)$ are the gradient and Hessian of $c_j(x)$; and the $m \times n$ matrix $A(x)$ is the Jacobian of $c(x)$, so that its jth row is $a_j(x)^T$. Exact second derivatives of f and $\{c_i\}$ are assumed to be available; note that all Hessian matrices are symmetric.

Let x^* denote a point where the following conditions hold:

Feasibility. $c(x^*) \ge 0$.

Constraint qualification. The gradients of the constraints active (equal to zero) at x^* are linearly independent.

First-order Karush–Kuhn–Tucker (KKT) condition. $g(x^*) = A^T(x^*)\lambda^*$ for a Lagrange multiplier λ^*, with $\lambda^* \ge 0$ and $\lambda_j^* c_j(x^*) = 0$ for $j = 1, \dots, m$.

Strict complementarity. $\lambda_j^* > 0$ if $c_j(x^*) = 0$.

Second-order KKT condition. The matrix $Z^{*T} W^* Z^*$ is positive definite, where Z^* is a basis for the null space of the Jacobian of the active constraints at x^* and $W^* = H(x^*) - \sum_{j=1}^m \lambda_j^* H_j(x^*)$, so that W^* is the Hessian of the Lagrangian function $f(x) - \lambda^T c(x)$ evaluated at (x^*, λ^*). (See (3.7).)

Under the above conditions, it is well known (see, for example, [10] and [11]) that x^* is an isolated local constrained minimizer of problem (3.1) and that λ^* is unique.

The *logarithmic barrier function* associated with the inequality-constrained problem (3.1) is

$$B(x,\mu) = f(x) - \mu \sum_{j=1}^{m} \ln c_j(x), \tag{3.2}$$

where μ is a positive scalar called the *barrier parameter.* Interior methods for constrained optimization, many based on this barrier function, have been the subject of intense research since their revival in 1984; see, for example, [19, 33] for a selection of references.

Given a sequence of monotonically decreasing and sufficiently small values of μ, our assumptions about x^* imply that there is a sequence of isolated local unconstrained minimizers x_μ of the barrier function (3.2) such that $\lim_{\mu\to 0} x_\mu = x^*$ and $\lim_{\mu\to 0} \mu/c_j(x_\mu) = \lambda_j^*$; the points $\{x_\mu\}$ define the *barrier trajectory.* For proofs and details, see, for example, [10] and [33].

3.2. The primal and primal-dual equations. The viewpoint taken in this paper is that the barrier parameter μ is specified at each iteration of algorithms based on the logarithmic barrier function, and that μ can be interpreted as characterizing an (unknown) target point on the barrier trajectory toward which we wish the current iterate to move. Algorithm-dependent rules govern selection of the initial μ and the decision about when to decrease μ. The strategy for obtaining a new μ is also algorithm-dependent; for example, the old μ may be multiplied by a constant factor, or modified using rules intended to achieve superlinear convergence (see, e.g., [9]).

To move from x to x_μ, an obvious strategy is to seek a zero of the barrier gradient $\nabla B(x,\mu)$ by applying Newton's method to a local quadratic model of the barrier function. The gradient and Hessian of $B(x,\mu)$ are given by

$$\nabla B(x,\mu) = g(x) - \mu A^T(x) C^{-1}(x)\mathbf{1}; \tag{3.3}$$

$$\nabla^2 B(x,\mu) = H(x) - \sum_{j=1}^{m} \frac{\mu}{c_j(x)}\, H_j(x) + \mu A^T(x) C^{-2}(x) A(x). \tag{3.4}$$

(When v is a vector, V denotes diag(v), and $\mathbf{1}$ denotes the vector of appropriate dimension whose components are all equal to one.) Omitting arguments, the resulting $n \times n$ *primal Newton barrier equations* are

$$M_{\mathrm{p}} p = -g + \mu A^T C^{-1}\mathbf{1} \equiv b_{\mathrm{p}}, \quad \text{where} \quad M_{\mathrm{p}} \equiv \nabla^2 B \tag{3.5}$$

and "primal" refers to the original problem variables x.

The primal barrier Hessian M_{p} has many well known properties. Of particular interest here is its ill-conditioning at points lying on the barrier trajectory as $\mu \to 0$, which was observed in the late 1960s (see [23, 25]) and was one of the reasons for the decline in use of barrier methods. More recently, a detailed analysis was given in [34] of the structure of the primal barrier Hessian (3.4) in an entire neighborhood of the solution. In addition to ill-conditioning of the Hessian matrix (which can be overcome by various means), primal barrier methods suffer from other, more serious drawbacks; see, for example, [35].

An alternative, increasingly popular, approach for locating x^* via x_μ is to use a *primal-dual method* based on the properties of x_μ; see, for example, [9, 4, 12, 8, 15]. In a primal-dual method, x and λ (the Lagrange multipliers, or dual variables) are treated as independent. Once we define $n+m$ nonlinear equations that hold along the barrier trajectory, Newton's method is invoked to solve for steps in x and λ.

Primal-dual equations are usually derived by interpreting x_μ as a point where two conditions hold: (i) the objective gradient g is a linear combination of the constraint gradients $\{a_j\}$, and (ii) the coefficients in the linear combination are given by $\lambda_\mu(x) = \mu C(x)^{-1}\mathbf{1}$, and so have a special relationship to the constraint values and barrier parameter. Thus the following $n+m$ nonlinear equations are satisfied at $(x_\mu, \lambda_\mu(x_\mu))$:

$$g = A^T\lambda \quad \text{and} \quad c_i\lambda_i = \mu, \quad i = 1, \ldots, m. \tag{3.6}$$

The first equation in (3.6) is a universal ingredient in primal-dual methods. Applying Newton's method, we obtain n equations satisfied by the primal-dual steps p (in x) and ℓ (in λ):

$$W(\lambda)p - A^T\ell = -g + A^T\lambda, \quad \text{with} \quad W(\lambda) \equiv H - \sum_{j=1}^{m} \lambda_j H_j, \tag{3.7}$$

where vector and matrix functions are evaluated at the current x and λ. The second relation in (3.6), called "approximate complementarity," provides m additional equations to complete a primal-dual method. Four mathematically equivalent forms have been suggested (see, e.g., [19]):

(i) $c_i\lambda_i - \mu = 0$; (ii) $c_i - \mu/\lambda_i = 0$; (iii) $\lambda_i - \mu/c_i = 0$; (iv) $1/\mu - 1/(c_i\lambda_i) = 0$.

For many reasons, (i) leads to the most effective primal-dual method—in particular, it is the only formulation among (i)–(iv) in which the condition of the primal-dual matrix asymptotically reflects the condition of the problem; see, for example, [24, 33, 12]. The (full) $n+m$ *primal-dual equations* associated with (i) are

$$\begin{pmatrix} W(\lambda) & -A^T \\ \Lambda A & C \end{pmatrix} \begin{pmatrix} p \\ \ell \end{pmatrix} = \begin{pmatrix} -g + A^T\lambda \\ \mu\mathbf{1} - C\lambda \end{pmatrix}. \tag{3.8}$$

One option for calculating the primal-dual steps (p, ℓ) is to solve (3.8) explicitly at every iteration. An advantage of this approach is that, as already mentioned, the matrix in (3.8) has a bounded condition number as x and λ converge to x^* and λ^*. However, because the dimension of the linear system is $n+m$, the associated linear algebraic work may be excessive. A more substantive difficulty is that, for nonconvex problems, an effective primal-dual method needs to be able to move away from nonminimizing stationary points and to determine whether the reduced Hessian of the Lagrangian function is positive definite. No straightforward way is known to achieve these results automatically while factorizing the unsymmetric matrix (3.8).

In [14], the effect of symmetrizing (3.8) is analyzed. Although this creates an ill-conditioned matrix without reducing the dimensionality, the ill-conditioning is shown to be benign when the symmetrized system is solved using a suitable symmetric indefinite factorization, which can also detect indefiniteness in the reduced Hessian of the Lagrangian; the primal-dual method of [12] is based on this approach.

An alternative—used in [15], for example—is to eliminate the $(1,2)$ block of (3.8), leading to an $n \times n$ symmetric linear system in p alone:

$$M_{\text{pd}}p = -g + \mu A^T C^{-1}\mathbf{1} \equiv b_{\text{p}}, \quad \text{where} \quad M_{\text{pd}} \equiv W(\lambda) + A^T C^{-1}\Lambda A. \tag{3.9}$$

We shall call M_{pd} the *condensed* primal-dual matrix. Observe that the right-hand side of (3.9) is the negative barrier gradient b_{p}, as in (3.5), and that, when λ is taken as $\lambda_\mu(x) = \mu C^{-1}\mathbf{1}$, the condensed primal-dual "Hessian" M_{pd} and the primal barrier Hessian M_{p} are the same matrix. (This does not imply, however, an algorithmic equivalence between primal and primal-dual methods; see [9].)

3.3. Properties of the condensed primal-dual matrix. We wish to analyze M_{pd} when it is evaluated at strictly feasible (x, λ) satisfying

$$\|x - x^*\| = \delta \quad \text{and} \quad \|\lambda - \lambda^*\| = O(\delta), \tag{3.10}$$

where $\delta \ll 1$. (The value of δ will of course be *unknown* in a practical setting.)

Given the target value μ, a primal-dual step p will be useful only if it moves from x toward x_μ and closer to x^*. We thus further limit our analysis to points such that x is *farther* from x^* than x_μ is, so that x and x_μ satisfy

$$\|x - x^*\| \geq \|x_\mu - x^*\|. \tag{3.11}$$

(Although this condition cannot be guaranteed in practice, it should hold at later iterates in primal-dual methods with sensible rules for adjusting μ.) Under the conditions of section 3.1, $\|x_\mu - x^*\| = \kappa_\mu \mu$, where $\kappa_\mu = \Theta(1)$ (see, for example, [10] and [33]). Combined with (3.11) and the definition (3.10) of δ, this property of κ_μ implies that $\kappa_\mu \mu \leq \delta$, i.e.,

$$\mu = O(\delta). \tag{3.12}$$

It should be noted that, although the current barrier parameter μ does not affect M_{pd}, it influences the right-hand side of the condensed primal-dual equations (3.9).

Let $\hat{m}$ denote the number of constraints active at x^*. For any point x, $\hat{A}(x)$ denotes the Jacobian of the active constraints (where "active" means active at x^*), and $\bar{A}(x)$ is the Jacobian of the inactive constraints. For any m-vector v, $\hat{v}$ denotes the $\hat{m}$-subvector of components corresponding to active constraints, and similarly for $\bar{v}$ and the inactive constraints. The analysis to be given is interesting only if $0 < \hat{m} < n$, which we henceforth assume. (See [34] for a discussion of $\hat{m} = 0$ and $\hat{m} = n$ in the case of the primal barrier Hessian.)

Because of smoothness, (3.10) implies boundedness of g, c, A, H, and $\{H_j\}$, as well as $O(\delta)$ closeness to their values at x^*. Although $\hat{c}(x) = O(\delta)$, components of $\hat{c}(x)$ can be arbitrarily small. As we shall see, the smallest constraint value affects M_{pd}, and hence we define $c_{\min}$ as

$$c_{\min}(x) \equiv \min \hat{c}_i(x).$$

The next two results from [31] and [34] are quoted for completeness. Lemma 3.1 is Corollary IV.4.10 of [31], and Theorem 3.1 combines Theorems 2.3 and 2.4 of [34].

LEMMA 3.1. (Closeness of eigenvalues.) *Let M and $\widetilde{M}$ be real symmetric matrices with eigenvalues $\{\eta_i\}$ and $\{\tilde{\eta}_i\}$, respectively. Then* $\max\{|\tilde{\eta}_i - \eta_i|\} \leq \|\widetilde{M} - M\|$.

THEOREM 3.1. *Let M denote a real symmetric matrix, and define the perturbed matrix $\widetilde{M}$ as $M + E$, where E is symmetric. Consider an orthogonal matrix $(X_1 \;\; X_2)$ where X_1 has ℓ columns, such that* $\text{range}(X_1)$ *is a simple invariant subspace of M, with*

$$\begin{pmatrix} X_1^T \\ X_2^T \end{pmatrix} M (X_1 \;\; X_2) = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix} \quad \textit{and} \quad \begin{pmatrix} X_1^T \\ X_2^T \end{pmatrix} E (X_1 \;\; X_2) = \begin{pmatrix} E_{11} & E_{12} \\ E_{12}^T & E_{22} \end{pmatrix}.$$

Let $d_1 = \text{sep}(L_1, L_2) - \|E_{11}\| - \|E_{22}\|$ and $\upsilon = \|E_{12}\|/d_1$, where $\text{sep}(L_1, L_2) = \min_{i,j} |eig_i(L_1) - eig_j(L_2)|$, *with $eig_k(\cdot)$ denoting the kth eigenvalue of its argument. If $d_1 > 0$ and $\upsilon < \frac{1}{2}$, then*

(i) *there are orthonormal bases $\widetilde{X}_1$ and $\widetilde{X}_2$ for simple invariant subspaces of the perturbed matrix $\widetilde{M}$ satisfying $\|X_1 - \widetilde{X}_1\| \le 2\upsilon$ and $\|X_2 - \widetilde{X}_2\| \le 2\upsilon$;*

(ii) *for $i = 1, \ldots, \ell$, there is an eigenvalue $\tilde{\omega}$ of $\widetilde{M}$ satisfying $|\tilde{\omega} - \breve{\omega}_i| \le 3\|E_{12}\|\,\upsilon$, where $\{\breve{\omega}_i\}$ are the eigenvalues of $X_1^T \widetilde{M} X_1$.*

Using analysis very similar to that in [34], we now establish several mathematical properties of $M_{\rm pd}$ when δ of (3.10) is sufficiently small.

THEOREM 3.2. (Properties of $M_{\rm pd}$.) *Suppose that the condensed primal-dual matrix $M_{\rm pd}$ of* (3.9) *is evaluated at* (x, λ) *satisfying* (3.10) *for sufficiently small δ. Let $\{\phi_k\}$ denote the n eigenvalues of $M_{\rm pd}$, ordered so that $|\phi_1| \ge \cdots \ge |\phi_n|$, and let $(Y\ \ Z)$ be an orthogonal matrix, where the columns of Z span the null space of $\hat{A}(x)$. Then*

(i) *the $\hat{m}$ largest-magnitude eigenvalues of $M_{\rm pd}$ are positive, with $\phi_1 = \Theta(1/c_{\rm min})$ and $\phi_{\hat{m}} = \Omega(1/\delta)$;*

(ii) *the $n - \hat{m}$ smallest-magnitude eigenvalues of $M_{\rm pd}$ are $\Theta(1)$;*

(iii) *if $0 < \hat{m} < n$,* $\mathrm{cond}(M_{\rm pd}) = \Theta(1/c_{\rm min})$;

(iv) *there are matrices $\widetilde{Y}$ and $\widetilde{Z}$ whose columns form orthonormal bases for simple invariant subspaces of $M_{\rm pd}$, such that $Y - \widetilde{Y} = O(\delta)$ and $Z - \widetilde{Z} = O(\delta)$.*

Proof. The form of $M_{\rm pd}$ is

$$M_{\rm pd} = W(\lambda) + A^T C^{-1} \Lambda A = W(\lambda) + \bar{A}^T \bar{C}^{-1} \bar{\Lambda} \bar{A} + \hat{A}^T \hat{C}^{-1} \hat{\Lambda} \hat{A}, \tag{3.13}$$

and we examine its three elements in turn.

The matrix W is $O(1)$; this follows from smoothness of H and H_j, full rank of $\hat{A}(x^*)$, strict complementarity, and closeness of λ to λ^*. It is straightforward that

$$\bar{A}^T \bar{C}^{-1} \bar{\Lambda} \bar{A} = O(\delta), \tag{3.14}$$

since $\bar{A} = O(1)$, each $\bar{c}_i$ is bounded away from zero, and $\bar{\lambda}$ is $O(\delta)$. Thus

$$M_{\rm pd} = \hat{A}^T \hat{C}^{-1} \hat{\Lambda} \hat{A} + O(1). \tag{3.15}$$

Since $\hat{C} > 0$ and $\hat{\Lambda} > 0$, the eigenvalues $\{\eta_i\}$ of $\hat{A}^T \hat{C}^{-1} \hat{\Lambda} \hat{A}$ satisfy $\eta_i > 0$ for $i = 1, \ldots, \hat{m}$, and $\eta_i = 0$ for $i = \hat{m} + 1, \ldots, n$. Next we consider the size of the elements of $\hat{C}^{-1} \hat{\Lambda}$. Because $\|x - x^*\| = \delta$ and $\hat{\lambda} - \hat{\lambda}^* = O(\delta)$, we know that $\hat{c}(x) = O(\delta)$ and every element of $\hat{\lambda}$ is $\Theta(1)$. Hence the smallest diagonal element in $\hat{C}^{-1} \hat{\Lambda}$ is $\Omega(1/\delta)$ and the largest element of $\hat{C}^{-1} \hat{\Lambda}$ is $\Theta(1/c_{\rm min})$. For sufficiently small δ, smoothness and the constraint qualification imply that $\hat{A}$ is $\Theta(1)$ and has rank $\hat{m}$. It follows from, for example, Theorem I.4.5 in [31] that

$$\eta_1 = \Theta(1/c_{\rm min}) \quad \text{and} \quad \eta_{\hat{m}} = \Omega(1/\delta).$$

Result (i) follows by applying Lemma 3.1 in conjunction with these bounds and (3.15).

We now apply Theorem 3.1 with $\widetilde{M} = M_{\rm pd}$, $M = \hat{A}^T \hat{C}^{-1} \hat{\Lambda} \hat{A}$, $X_1 = Z$, and $X_2 = Y$. As shown in [34], X_1 and X_2 form orthonormal bases for simple invariant subspaces of $\hat{A}^T \hat{C}^{-1} \hat{\Lambda} \hat{A}$, and we have $L_1 = 0$ and $L_2 = Y^T \hat{A}^T \hat{C}^{-1} \hat{\Lambda} \hat{A} Y$. The smallest eigenvalue of L_2 is $\Omega(1/\delta)$ and the matrix E is $O(1)$, from which it follows that the conditions on d_1 and υ of Theorem 3.1 are satisfied.

Part (ii) of Theorem 3.1 implies that $n - \hat{m}$ eigenvalues of $M_{\rm pd}$ differ by $O(\delta)$ from those of $Z^T M_{\rm pd} Z$. Combining (3.13) and (3.14), we see that $Z^T M_{\rm pd} Z - Z^T W Z = O(\delta)$. Since $Z^{*T} W^* Z^*$ is positive definite, smoothness implies that $Z^T W Z$ is positive definite for small enough δ, and hence that all of its eigenvalues are $\Theta(1)$. Result (ii) follows by invoking Lemma 3.1. When $0 < \hat{m} < n$, results (i) and (ii) imply (iii).

Result (iv) is obtained by applying part (i) of Theorem 3.1. □

Analogues of Theorem 3.2 hold for the condensed $n \times n$ matrices in the primal-dual methods derived from forms (ii)–(iv) of the second relation in (3.6). For each matrix, there are slightly different assumptions and bounds on the eigenvalues and the closeness of the invariant subspaces, depending on δ, $c_{\min}$, and μ.

3.4. Connections with section 2.

3.4.1. Special structure of M_{pd}. Parts (i) and (ii) of Theorem 3.2 show that M_{pd} has $\hat{m}$ large positive eigenvalues that are well separated from the $n - \hat{m}$ small eigenvalues, thereby implying that the largest $\hat{m}$ singular values of M_{pd} are equal to its $\hat{m}$ large eigenvalues. In the notation of section 2,

$$M_{\mathrm{pd}} = U\Sigma V^T = \begin{pmatrix} U_L & U_S \end{pmatrix} \begin{pmatrix} \Sigma_L & 0 \\ 0 & \Sigma_S \end{pmatrix} \begin{pmatrix} V_L^T \\ V_S^T \end{pmatrix}, \tag{3.16}$$

where Σ_L contains the $\hat{m}$ large eigenvalues of M_{pd}, Σ_S contains the absolute values of the $n - \hat{m}$ small eigenvalues of M_{pd}, the columns of U are the eigenvectors of M_{pd}, $V_L = U_L$, and V_S is equal to U_S, possibly with the signs of its columns changed. The ordering of the elements of Σ_S and the associated changes of sign in V_S do not affect the results here, which depend only on the condition of Σ_S and the subspace spanned by V_S; hence we assume (subject to this proviso) that $V_S = U_S$.

We know from part (iv) of Theorem 3.2 that, for suitably chosen Y and Z,

$$U_L - Y = O(\delta) \quad \text{and} \quad U_S - Z = O(\delta). \tag{3.17}$$

Although it is therefore natural (and correct) to associate the large space of M_{pd} with the range space of $\hat{A}^T$, and the small space of M_{pd} with the null space of $\hat{A}$, we stress that these subspaces must, strictly speaking, always be distinguished. In particular, a nonsingular M_{pd} has a nontrivial small space of dimension $n - \hat{m}$ that is close to the null space of $\hat{A}$, but only a trivial (zero-dimensional) null space.

Much of our later analysis relies on the closeness of Y to U_L and Z to U_S. In particular, relation (3.17) implies that

$$U_L = Y + \Upsilon_Y O(\delta) \quad \text{and} \quad U_S = Z + \Upsilon_Z O(\delta), \tag{3.18}$$

where Υ_Y and Υ_Z are matrices of unit norm. Consequently,

$$Y^T U_L = I + O(\delta) \quad \text{and} \quad Y^T U_S = O(\delta); \tag{3.19}$$

$$Z^T U_S = I + O(\delta) \quad \text{and} \quad Z^T U_L = O(\delta). \tag{3.20}$$

As indicated in the proof of Theorem 3.2, the condition of Σ_L depends on $\mathrm{cond}(\hat{A})$, $\mathrm{cond}(\hat{\Lambda})$, and the ratio $\hat{c}_{\max}/\hat{c}_{\min}$, which is $O(\delta/c_{\min})$. The condition of Σ_S ultimately reflects the condition of $Z^{*T}W^*Z^*$.

In summary, when (3.10) holds for small enough δ, we have

$$\|M_{\mathrm{pd}}\| = \|\Sigma_L\| = \Theta(1/c_{\min}), \quad \Sigma_L^{-1} = O(\delta), \tag{3.21}$$

$$\Sigma_S = \Theta(1), \quad \text{and} \quad \|M_{\mathrm{pd}}^{-1}\| = \|\Sigma_S^{-1}\| = \Theta(1). \tag{3.22}$$

3.4.2. Properties of perturbed matrices. The properties discussed in section 3.3 apply to the exact matrix $M_{\rm pd}$. Assuming that δ of (3.10) is sufficiently small, we examine circumstances in which these properties continue to hold for $\widetilde{M}_{\rm pd}$, a symmetric perturbation of $M_{\rm pd}$:

$$\widetilde{M}_{\rm pd} = M_{\rm pd} + \varDelta, \quad \text{where } \varDelta \text{ is symmetric.} \tag{3.23}$$

The main point of interest is how large a perturbation can be tolerated while retaining the special structure of $M_{\rm pd}$.

The proof of Theorem 3.2 shows that $\eta_{\hat{m}}$, the smallest positive eigenvalue of $\hat{A}^T\hat{C}^{-1}\hat{\Lambda}\hat{A}$, is $\Omega(1/\delta)$. If the perturbation $\varDelta$ in (3.23) satisfies

$$\|\varDelta\| \ll \eta_{\hat{m}}, \tag{3.24}$$

then the combination of Lemma 3.1 and our analysis of the eigenvalues of $M_{\rm pd}$ shows that the perturbed matrix $\widetilde{M}_{\rm pd} = M_{\rm pd} + \varDelta$ continues to have $\hat{m}$ large positive eigenvalues that are $\Omega(1/\delta)$ and $n - \hat{m}$ small eigenvalues that are $O(\max(1, \|\varDelta\|))$ in magnitude. It follows from result (iv) of Theorem 3.1 that, if $\|\varDelta\|$ is sufficiently small compared to $\eta_{\hat{m}}$, the perturbed matrix $\widetilde{M}_{\rm pd}$ has invariant subspaces close to the range space of $\hat{A}^T$ and the null space of $\hat{A}$.

The perturbations to $M_{\rm pd}$ that we derive later are bounded by multiples of $\|M_{\rm pd}\|$ rather than $\eta_{\hat{m}}$, which means that we cannot use (3.24) as stated. However, result (i) of Theorem 3.2 shows that the ratio of $\|M_{\rm pd}\|$ to $\eta_{\hat{m}}$ is $O(\delta/c_{\min})$, so that the following guideline may be applied.

GUIDELINE 3.1. *If $\delta/c_{\min}$ is not too large, then any positive θ satisfying $\theta \ll \|M_{\rm pd}\|$ also satisfies $\theta \ll \eta_{\hat{m}}$, where $\eta_{\hat{m}}$ is the $\hat{m}$-th eigenvalue of $\hat{A}^T\hat{C}^{-1}\hat{\Lambda}\hat{A}$.*

4. Forming the matrix and right-hand side.

In addition to errors associated with ill-conditioning, numerical errors incurred while *forming* the needed quantities in finite precision can exert a major influence on the accuracy of various interior methods, including the primal-dual methods considered here. In particular, [37, 38, 39] discuss the important role of rounding errors in primal-dual methods for linear programming and linear complementarity problems.

4.1. Cancellation in calculating the active constraints.

Let $\mathbf{u}$ denote unit roundoff as defined in, for example, [22, pages 42–44], and let $fl(\cdot)$ denote the *rounded* version of its argument—which may be a scalar, vector, or matrix—in a floating-point number system. For any real number x in the range of a floating-point number system and any two representable numbers y and z in that system, $\mathbf{u}$ is the smallest positive number such that

$$fl(x) = x(1+\tau), \quad |\tau| < \mathbf{u}, \quad \text{and} \quad fl(y \text{ op } z) = (y \text{ op } z)(1+\tau), \quad |\tau| \le \mathbf{u}, \tag{4.1}$$

where "op" denotes one of $\{+, -, *, /\}$. Thus $\mathbf{u}$ is a bound on the relative error occurring in representing a single number or performing one floating-point operation on two representable numbers. With binary IEEE arithmetic, $\mathbf{u} \approx 6{\times}10^{-8}$ in single precision and $\mathbf{u} \approx 1.1{\times}10^{-16}$ in double precision.

A common rule of thumb in computation is that floating-point quantities of interest involve a relative error bounded by an order-unity multiple of $\mathbf{u}$. In the primal-dual method considered here, however, relative errors substantially exceeding unit roundoff often occur because *inaccuracy in the computed active constraints* propagates through their (large) reciprocals into the condensed primal-dual matrix and the associated right-hand side. By contrast, errors in the active constraints do not have an inordinately large effect on calculation of the search direction in other optimization algorithms, such as sequential quadratic programming (SQP) methods (see, for example, [11]).

By definition, the active constraints are converging to zero and will be small during the final iterations of an interior method. Calculation of small quantities often involves *cancellation* (subtraction of close numbers that have previously been rounded), and it is well known that cancellation may produce a large *relative* error in a computed quantity; see, for example, the detailed discussions in [17, pages 40–42] and [22, pages 10–11]. Although some small quantities, such as $fl(x^2)$ for x near zero, can be computed with high relative accuracy, in general this cannot be guaranteed.

Suppose that the active constraint c_i is evaluated in finite precision at a strictly feasible point x at which $\mathbf{u} < c_i(x) \ll 1$. If calculation of $c_i(x)$ is subject to cancellation, the computed value will satisfy $fl(c_i(x)) = c_i(x) + O(\mathbf{u})$, giving an absolute error that is $O(\mathbf{u})$ and a relative error that is $O(\mathbf{u}/c_i)$. Note that the *relative* accuracy bound worsens as c_i becomes smaller, i.e., as the iterates converge to x^*. The computed value of $1/c_i$ is obtained by performing one additional floating-point operation, so that the relative error in the computed value of $1/c_i$ thus remains the (unfavorable) value $O(\mathbf{u}/c_i)$. The relative error arising from multiplication by C^{-1} is $O(\mathbf{u}/c_{\min})$, where $c_{\min}$ is the smallest constraint value.

In typical optimization problems, the constraints are calculated by the user in a standard way, which means that their values will almost certainly be subject to cancellation. In some instances, however, the active constraints may be calculated to higher precision—for example, the constraints may be nonnegativity bounds on the variables (so that their values are exactly those of the variables), or may represent a delicate physical process computed from a numerical simulation whose accuracy is controllable by the user [29]. We therefore consider two cases: either the active constraints are calculated to full precision or they are subject to cancellation. Analysis of the no-cancellation case allows us to separate the effects of the constraints' accuracy from those of ill-conditioning in the condensed matrix.

The quantity ζ will be used to indicate an upper bound on the relative accuracy of the computed constraints:

$$\begin{aligned} &\zeta = \mathbf{u} && \text{when all constraints have full relative precision;} \\ &\zeta = \mathbf{u}/c_{\min} && \text{when cancellation occurs in the active constraints.} \end{aligned} \tag{4.2}$$

4.2. Accuracy of the matrix and right-hand side. The relative accuracy of the active constraints affects the accuracy of the computed versions of $M_{\rm pd}$ and $b_{\rm p}$, both of which involve division by the constraint values. The actual errors associated with any particular computation are unknown in advance. All of our results involve only *bounds* on the error.

4.2.1. The condensed primal-dual matrix. At points of interest, $M_{\rm pd}$ is dominated by $\hat{A}^T\hat{C}^{-1}\hat{\Lambda}\hat{A}$, where the (ultimately unbounded) active constraint reciprocals appear in the diagonal matrix $\hat{C}^{-1}$. To obtain an expression for the computed version of $M_{\rm pd}$, denoted by *computed*$(M_{\rm pd})$, we work through the associated floating point operations (see, for example, [22, pages 77–78]). The notation "*computed*$(\cdot)$" is meant to indicate that a sequence of floating-point operations is performed. The result is

$$\mathit{computed}(M_{\rm pd}) = (A + \Delta A)^T(D + \Delta D)(A + \Delta A) + W + E, \tag{4.3}$$

where D denotes the (positive diagonal) matrix $C^{-1}\Lambda$, $|\Delta d_i| \le \zeta d_i$ (see (4.2)), $\|\Delta A\| = \mathbf{u}\,O(\|A\|)$, and E is symmetric, with $\|E\| = O(\mathbf{u}/c_{\min})$. Thus the computed M_{pd} is the *exact* sum of $\widetilde{A}^T\widetilde{D}\widetilde{A}$, the exact matrix W, and an error matrix E, where $\widetilde{A}$ is a perturbation of the exact A and $\widetilde{D}$ is a diagonal perturbation of the exact D. The major effects of cancellation error are represented by the appearance of ζ in the elements of ΔD.

Separating active and inactive constraints and grouping the elements of (4.3) by size, we have

$$\hat{A}^T\hat{D}\hat{A} = \Theta(1/c_{\min}), \quad \hat{A}^T\Delta\hat{D}\hat{A} = O(\zeta/c_{\min}), \quad \text{and} \quad \hat{D}\Delta\hat{A} = O(\mathbf{u}/c_{\min}).$$

All other terms are $O(\mathbf{u}/c_{\min})$ or smaller, and may be subsumed in E. The result is

$$\begin{aligned} \mathit{computed}(M_{\mathrm{pd}}) &= \hat{A}^T\hat{D}\hat{A} + \bar{A}^T\bar{D}\bar{A} + W + \hat{A}^T\Delta\hat{D}\hat{A} + O(\mathbf{u}/c_{\min}) \\ &= M_{\mathrm{pd}} + \hat{A}^T\Delta\hat{D}\hat{A} + O(\mathbf{u}/c_{\min}). \end{aligned}$$

We now analyze the size and structure of the perturbation ΔM_{pd}, defined as

$$\begin{aligned} \Delta M_{\mathrm{pd}} &\equiv \mathit{computed}(M_{\mathrm{pd}}) - M_{\mathrm{pd}} \;=\; \Delta M + R, \quad \text{where} \\ \Delta M &\equiv \hat{A}^T\Delta\hat{D}\hat{A} = O(\zeta/c_{\min}) \quad \text{and} \quad R = O(\mathbf{u}/c_{\min}). \end{aligned} \tag{4.4}$$

Note that we have split the full perturbation ΔM_{pd} into two pieces: a matrix ΔM lying entirely in the range of $\hat{A}^T$ and a remainder matrix R.

If the active constraints retain full accuracy, both ΔM and R are $O(\mathbf{u}/c_{\min})$, so that

$$\Delta M_{\mathrm{pd}} = O(\mathbf{u}/c_{\min}), \tag{4.5}$$

and no general conclusions can be drawn about its structure. In this case, the perturbation ΔM_{pd} created by forming M_{pd} is typically *much smaller* than in the case of cancellation; see the numerical examples in section 4.3.

When cancellation occurs in calculating the active constraints, so that $\zeta = \mathbf{u}/c_{\min}$, ΔM_{pd} satisfies

$$\Delta M_{\mathrm{pd}} = \hat{A}^T\Delta\hat{D}\hat{A} + O(\mathbf{u}/c_{\min}) = \Delta M + O(\mathbf{u}/c_{\min}) = O(\mathbf{u}/c_{\min}^2). \tag{4.6}$$

Relation (4.6) reveals that cancellation has two major and related effects: ΔM_{pd} has a much larger error bound than for the non-cancellation case (see (4.5)), and furthermore is likely to be dominated by a matrix that lies *entirely in the range space of* $\hat{A}^T$. As we shall see in section 6, we thus have the comforting property that, although there is a potentially large error in the computed M_{pd}, the nature of the perturbation ensures that the search direction does not blow up as it would following a general perturbation to M_{pd} of the same size. (See section 6.2.2 and (6.25).)

4.2.2. The right-hand side. The significance of rounding errors for the right-hand side depends primarily on errors in the constraints, but also on the size of $c_{\min}$ and closeness of x to x_μ. Exactly as with the computed matrix M_{pd}, both the size and the nature of the error in $\mathit{computed}(b_{\mathrm{p}})$ are affected when cancellation occurs in the constraints. (We assume that errors of representation in g and A are $O(\mathbf{u})$.) Let F denote μC^{-1}, and recall our assumption (3.12) that $\mu = O(\delta)$. The vector $\mathit{computed}(A^T F\mathbf{1})$ may be expressed as

$$\begin{aligned} \mathit{computed}(A^T F\mathbf{1}) &= (A + \Delta A)^T(F + \Delta F)\mathbf{1} + \tilde{e}, \quad \text{where} \\ \Delta A &= O(\mathbf{u}), \; \Delta F = O(\delta\zeta/c_{\min}) \;\text{ and }\; \tilde{e} = O(\mathbf{u}). \end{aligned}$$

Forming *computed*(b_{p}) and simplifying, we see that

$$\begin{aligned} \mathit{computed}(b_{\mathrm{p}}) - b_{\mathrm{p}} &\equiv \mathit{\Delta} b_{\mathrm{p}} = \mathit{\Delta} b + e, \quad \text{where} \\ \mathit{\Delta} b &\equiv \hat{A}^T \mathit{\Delta}\hat{F}\,\mathbf{1} = O(\mu\zeta/c_{\min}) \quad \text{and} \quad e = O(\delta/\mathbf{u}c_{\min}). \end{aligned} \tag{4.7}$$

When the constraints are calculated with full precision, so that $\zeta = \mathbf{u}$, the error in *computed*(b_{p}) has no special structure and satisfies

$$\mathit{\Delta} b_{\mathrm{p}} = O(\delta\mathbf{u}/c_{\min}). \tag{4.8}$$

In contrast, when c is subject to cancellation, so that $\zeta = \mathbf{u}/c_{\min}$, (4.7) shows that, when μ is not too small, the error in *computed*(b_{p}) tends to be dominated by $\mathit{\Delta} b$, which lies entirely in the range of $\hat{A}^T$ and is $O(\delta\mathbf{u}/c_{\min}^2)$:

$$\mathit{\Delta} b_{\mathrm{p}} = \mathit{\Delta} b + e, \quad \text{where} \quad \mathit{\Delta} b = O(\delta\mathbf{u}/c_{\min}^2). \tag{4.9}$$

With or without cancellation, the relative error in *computed*(b_{p}) may be large when x is very close to x_μ because of cancellation error in subtracting g and $\mu A^T C^{-1}\mathbf{1}$.

4.3. Numerical examples. To illustrate the effects of forming M_{pd} and b_{p}, we examine a specific problem:

$$\begin{array}{lrl} \text{minimize} & 5x_1x_2x_3 - \frac{1}{2}x_1^2 + 10(x_1-1)^2 - 2x_2x_3 - x_3 - \frac{3}{2}x_2^2 - x_3^2 & \\ \text{subject to} & -x_1^2 - x_3^2 - x_1 - 2x_2 - x_3 + 2 & \geq 0 \\ & x_1 + \frac{3}{4} & \geq 0 \\ & (x_1 - x_3)^2 + x_2^3 - 0.1x_1 + 0.05x_1^2 + 1.05 & \geq 0. \end{array} \tag{4.10}$$

An optimal solution of (4.10) is $x^* = (1, -1, 1)^T$, where the first and third constraints are active, with $\lambda^* = (2, 0, \frac{10}{3})^T$. All calculations were performed on a Silicon Graphics 4D/440VGX using binary IEEE arithmetic. Values labeled as *computed*$(\cdot)$ or *sing*$(\cdot)$ were computed in single precision ($\mathbf{u} \approx 6 \times 10^{-8}$); other values (designated as "exact") were obtained by rounding the final results of calculations performed in double precision ($\mathbf{u} \approx 1.1{\times}10^{-16}$). The numbers displayed are correctly rounded to the number of digits shown.

For $\mu = 10^{-4}$, we consider the points

$$\hat{x} = \begin{pmatrix} 1 + 2^{-12} \\ -1 + 2^{-12} \\ 1 - 2^{-11} \end{pmatrix} \approx \begin{pmatrix} 1.00024 \\ -0.99976 \\ 0.99951 \end{pmatrix} \quad \text{and} \quad x_\mu = \begin{pmatrix} 1.00002311749 \\ -0.99999000148 \\ 0.99995354848 \end{pmatrix}, \tag{4.11}$$

where $\hat{x}$ is representable exactly in IEEE single precision and satisfies $\|\hat{x} - x^*\| = 5.98{\times}10^{-4}$ and $\|\hat{x} - x_\mu\| = 5.47{\times}10^{-4}$. We chose an exactly representable $\hat{x}$ to focus attention on errors in computing the constraints.

The form of the first constraint c_1 in (4.10) is obviously subject to cancellation if implemented in a standard way. The single-precision and "exact" (double-precision, rounded to eight digits) versions of $c_1(\hat{x})$ are

$$\mathit{sing}(c_1(\hat{x})) = 2.4390221{\times}10^{-4} \quad \text{and} \quad c_1(\hat{x}) = 2.4384260{\times}10^{-4}.$$

These values differ in the fourth decimal place, revealing a relative error much larger than $\mathbf{u}$. The relative error $|\mathit{sing}(c_1) - c_1|/c_1$ is $2.4{\times}10^{-4}$, which is well estimated by the bound $\mathbf{u}/c_1 = 4.1{\times}10^{-4}$. As expected, the single-precision version of $1/c_1(\hat{x})$ displays the same relative error, $2.4{\times}10^{-4}$, as $sing(c_1)$.

We now consider the matrix M_{pd} at $\hat{x}$ of (4.11) with

$$\lambda = (2 + 3/4096,\ 1/1024,\ 10/3 + 3/4096)^T, \tag{4.12}$$

for which $\|\lambda - \lambda^*\| = 1.42{\times}10^{-3}$. The constraints were first computed in double precision and then rounded to single, thereby avoiding cancellation error in the constraints at the single-precision level; all other computations were performed in single precision. The results, denoted by the subscript "2," are

$$\|\varDelta_2 M_{\text{pd}}\| = 8.24{\times}10^{-3}, \quad \|Y^T\varDelta_2 M_{\text{pd}}\| = 7.15{\times}10^{-3}, \quad \text{and} \quad \|Z^T\varDelta_2 M_{\text{pd}}\| = 6.51{\times}10^{-3},$$

where Y and Z are orthonormal bases for the range space of $\hat{A}^T$ and the null space of $\hat{A}$. When the constraints are subject to cancellation, the prediction in (4.6) of a much larger perturbation $\varDelta M_{\text{pd}}$ lying almost entirely in the range space of $\hat{A}^T$ is confirmed by the values

$$\|\varDelta M_{\text{pd}}\| = 43.006, \quad \|Y^T\varDelta M_{\text{pd}}\| = 43.006, \quad \text{and} \quad \|Z^T\varDelta M_{\text{pd}}\| = 6.8416{\times}10^{-3}. \tag{4.13}$$

Comparing the exact and computed right-hand sides at $\hat{x}$, we have

$$\|computed(b_{\text{p}}) - b_{\text{p}}\| = 5.027{\times}10^{-4} \quad \text{and} \quad \|Z^T(computed(b_{\text{p}}) - b_{\text{p}})\| = 8.437{\times}10^{-8},$$

showing, as predicted by (4.9), that $computed(b_{\text{p}}) - b_{\text{p}}$ lies almost entirely in the range space of $\hat{A}^T$.

5. Numerical solution of the condensed primal-dual equations.

In this section we consider the effects of applying a generic backward-stable numerical method to solve the $n \times n$ condensed primal-dual system $M_{\text{pd}}\, p = b_{\text{p}}$. To simplify the error bounds in the remainder of the paper, we henceforth assume that, at all points x of interest,

$$c_{\min}(x) = \Omega(\delta). \tag{5.1}$$

This assumption can be interpreted as excluding points x that are "too close to the boundary"; see Guideline 3.1. Even with this restriction, there are too many unknown factors and mathematically imprecise rules of thumb to permit a rigorous theorem. Nonetheless, our analysis justifies the following guideline.

GUIDELINE 5.1. *Assume that the matrix M_{pd} and vector b_{p} of* (3.9) *are evaluated at (x, λ) satisfying* (3.10) *for sufficiently small δ, that the given barrier parameter μ satisfies $\mu = O(\delta)$, and that, in addition,*

(a) *$c_{\min}(x) = \Omega(\delta)$, i.e., x is not too close to the boundary of the feasible region;*

(b) *$c_{\min}$ is large enough relative to $\mathbf{u}$ to ensure that the chosen factorization runs successfully to completion;*

(c) *the* computed *primal-dual step $\tilde{\mathbf{p}}$ is obtained by applying a backward-stable method to the computed version of the linear system $M_{\text{pd}}p = b_{\text{p}}$, and the corresponding value γ_n (see* (2.7)*) is bounded by a reasonable constant.*

Then we can expect several results:

(i) **Bounds on the computed matrix and right-hand side.** *The computed matrix M_{pd} and right-hand side b_{p} satisfy the following relations:*

$$computed(M_{\text{pd}}) - M_{\text{pd}} = \varDelta M_{\text{pd}} = O(\zeta)\,\|M_{\text{pd}}\| = O(\zeta/\delta) \quad \textit{and} \tag{5.2}$$

$$computed(b_{\text{p}}) - b_{\text{p}} = \varDelta b_{\text{p}} = O(\zeta), \tag{5.3}$$

where ζ is defined by (4.2).

(ii) **Special structure in the computed matrix and right-hand side.** *The perturbations* ΔM_{pd} *of* (5.2) *and* Δb_{p} *of* (5.3) *may be written as*

$$\Delta M_{\mathrm{pd}} = \Delta M + O(\mathbf{u}/\delta), \quad \textit{where} \quad \Delta M \equiv \hat{A}^T \Delta\hat{D}\hat{A} = O(\zeta/\delta), \quad \textit{and} \tag{5.4}$$

$$\Delta b_{\mathrm{p}} = \Delta b + O(\mathbf{u}), \quad \textit{where} \quad \Delta b \equiv \hat{A}^T \Delta\hat{F}\,\mathbf{1} = O(\zeta). \tag{5.5}$$

When cancellation occurs, these relations imply that ΔM_{pd} *is likely to be dominated by a matrix whose columns lie in the range space of* $\hat{A}^T$*, and* Δb_{p} *is likely to be dominated by a vector in the range space of* $\hat{A}^T$*.*

(iii) **Backward error bounds for the primal-dual step in** x**.** *The computed step* $\widetilde{\boldsymbol{p}}$ *is the exact solution of the symmetric system*

$$\widetilde{M}_{\mathrm{pd}}\widetilde{\boldsymbol{p}} = (M_{\mathrm{pd}} + \varDelta)\widetilde{\boldsymbol{p}} = b_{\mathrm{p}} + \Delta b, \quad \textit{where} \tag{5.6}$$

$$\varDelta = \hat{A}^T \Delta\hat{D}\hat{A} + O(\mathbf{u}/\delta) = O(\zeta/\delta) \quad \textit{and} \quad \Delta b = \hat{A}^T\Delta\hat{F}\,\mathbf{1} = O(\zeta). \tag{5.7}$$

The assumptions of this guideline are included for various reasons. The analysis in section 3 of M_{pd} applies only if δ is sufficiently small. Assumption (a) implies that *computed*(M_{pd}) has properties similar to those of M_{pd} (see Guideline 3.1), and ensures that the condition number of Σ_L (see (3.16)) is $\Theta(1)$. Condition (b) implicitly bounds $\mathrm{cond}(M_{\mathrm{pd}})$ with respect to machine precision, thereby ensuring that most backward-stable direct methods applied to M_{pd} will run to completion. Because of Lemma 2.1 and condition (c), we can fold $\Delta b_{\mathrm{p}} - \Delta b$ into the matrix and reflect the effects of numerical solution in an additional $O(\mathbf{u}/\delta)$ perturbation to the matrix. Theorem 2.1 can then be invoked to create the symmetric perturbation $\varDelta$ of (5.7).

6. Properties of the computed solution.

The computed solution $\widetilde{\boldsymbol{p}}$ of the condensed primal-dual equations satisfies the perturbed equation (5.6). This section develops bounds on the error in $\widetilde{\boldsymbol{p}}$ using the special structure of M_{pd}, $\varDelta$, and Δb derived in sections 3, 4, and 5, subject to the assumptions of Guideline 5.1. In particular, note that the error bounds have been simplified by assuming that

$$c_{\min}(x) = \Omega(\delta) \quad \text{and} \quad \mu = O(\delta).$$

Let U_L and U_S denote the orthonormal matrices of singular vectors associated with the large and small singular values of the *exact* matrix M_{pd} (see (3.16)), and let Y and Z denote orthonormal bases for the range space of $\hat{A}^T$ and the null space of $\hat{A}$. The exact solution p of the condensed primal-dual equations can be expressed as

$$p = U_L p_L + U_S p_S = Y p_Y + Z p_Z, \tag{6.1}$$

with analogous forms for $\widetilde{\boldsymbol{p}}$ and other n-vectors of interest.

6.1. Perturbing the right-hand side.

Let $\bar{p}$ denote the exact solution of the intermediate system $M_{\mathrm{pd}}\bar{p} = b_{\mathrm{pd}} + \Delta b$. Since $M_{\mathrm{pd}}^{-1} = \Theta(1)$ (see (3.22)) and $\Delta b = O(\zeta)$ (see (5.7)), the general bound (2.1) would imply that

$$\bar{p} - p = O(\zeta). \tag{6.2}$$

However, we shall see that a more favorable bound can be obtained because Δb lies entirely in the range space of $\hat{A}^T$.

By definition of $\bar{p}$,

$$\bar{p} - p \;=\; M_{\mathrm{pd}}^{-1}\Delta b \;=\; U_L \Sigma_L^{-1}\Delta b_L + U_S\Sigma_S^{-1}\Delta b_S,$$

so that

$$\|\bar{p}_L - p_L\| \;\le\; \|\Sigma_L^{-1}\|\,\|\Delta b_L\|, \qquad \|\bar{p}_S - p_S\| \;\le\; \|\Sigma_S^{-1}\|\,\|\Delta b_S\|, \quad \text{and} \tag{6.3}$$

$$\|\bar{p} - p\| \;\le\; \|\Sigma_L^{-1}\|\,\|\Delta\beta_L\| + \|\Sigma_S^{-1}\|\,\|\Delta\beta_S\|. \tag{6.4}$$

Because Δb lies in the range space of $\hat{A}^T$, $\Delta b_Z = 0$ and $\|\Delta b_Y\| = \|\Delta b\|$. It follows from (3.19) and the fact that $\Delta b = O(\zeta)$ (see (5.5)) that

$$\|U_L^T \Delta b\| = \|\Delta b_L\| = O(\zeta) \quad \text{and} \quad \|U_S^T \Delta b\| = \|\Delta b_S\| = O(\delta\zeta), \tag{6.5}$$

where the smaller (by a factor of δ) bound on $\|\Delta b_S\|$ compared to $\|\Delta b_L\|$ arises because $\Delta b_Z = 0$.

We know from (3.21) and (3.22) that the matrices Σ_L and Σ_S associated with $M_{\rm pd}$ satisfy $\Sigma_L^{-1} = O(\delta)$ and $\|\Sigma_S^{-1}\| = \|M_{\rm pd}^{-1}\| = \Theta(1)$. Combining these estimates with (6.3), (6.4), and (6.5), we see that

$$\bar{p}_L - p_L = O(\delta\zeta), \quad \bar{p}_S - p_S = O(\delta\zeta), \quad \text{and} \quad \bar{p} - p = O(\delta\zeta), \tag{6.6}$$

with the same form of bound in all cases. Note that these bounds are smaller than the standard bound (6.2) by a factor of δ, and reflect both the size and structure of Δb. We conclude from (6.6) that $\|\bar{p}\| \approx \|p\|$ as long as p is not too small relative to δ.

6.2. Perturbing the matrix. We now turn to the relationship between $\tilde{\boldsymbol{p}}$ and $\bar{p}$, which satisfy an equation analogous to (2.2),

$$\widetilde{M}_{\rm pd}\tilde{\boldsymbol{p}} = (M_{\rm pd} + \Delta)\tilde{\boldsymbol{p}} = M_{\rm pd}\bar{p} = b + \Delta b,$$

in which the matrix is perturbed by Δ of (5.7). Since $M_{\rm pd}^{-1} = \Theta(1)$ and $\Delta = O(\zeta/\delta)$, application of the general bound (2.4) would give

$$\|\tilde{\boldsymbol{p}} - \bar{p}\| \le \|M_{\rm pd}^{-1}\|\,\|\Delta\|\,\|\bar{p}\| = \|\bar{p}\|\,O(\zeta/\delta). \tag{6.7}$$

By exploiting the special structure of Δ and the ill-conditioning of $M_{\rm pd}$, we obtain an improved bound for the range-space part of $\tilde{\boldsymbol{p}}$ in the no-cancellation case (section 6.2.1), and for all of $\tilde{\boldsymbol{p}}$ when cancellation occurs (section 6.2.2).

To begin, we express Δ in the form of (2.15):

$$U^T \Delta U = \begin{pmatrix} B_L \\ B_S \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{pmatrix}, \tag{6.8}$$

where $B_{11} = U_L^T \Delta U_L$, $B_{12} = U_L^T \Delta U_S$, and $B_{22} = U_S^T \Delta U_S$. The analysis of $\tilde{\boldsymbol{p}}$ varies with the presence or absence of cancellation, so that we consider the two cases separately.

6.2.1. Without cancellation. When cancellation does not occur, Δ is $O(\mathbf{u}/\delta)$ and has no special structure, so that both B_L and B_S in (6.8) are $O(\mathbf{u}/\delta)$. Applying the second inequalities of (2.16) and (2.17), we have

$$\|\tilde{\boldsymbol{p}}_L - \bar{p}_L\| \;\le\; \|\Sigma_L^{-1}\|\,\|\Delta\|\,\|\bar{p}\| \;=\; \|\bar{p}\|\,O(\mathbf{u}) \quad \text{and} \tag{6.9}$$

$$\|\tilde{\boldsymbol{p}}_S - \bar{p}_S\| \;\le\; \|\Sigma_S^{-1}\|\,\|\Delta\|\,\|\bar{p}\| \;=\; \|\bar{p}\|\,O(\mathbf{u}/\delta). \tag{6.10}$$

A key feature of these relations is that the bound on $\|\tilde{\boldsymbol{p}}_L - \bar{p}_L\|$ is smaller by a factor of δ than the bound on $\|\tilde{\boldsymbol{p}}_S - \bar{p}_S\|$, reflecting the (expected) result that only the small-space part of $\tilde{\boldsymbol{p}}$ will be blown up by ill-conditioning in $\widetilde{M}_{\rm pd}$; see the numerical example (6.23).

Putting together (6.6), (6.9), (6.10), and standard norm inequalities, we obtain the following bounds involving the computed $\tilde{\boldsymbol{p}}$ and the exact p when cancellation does not occur:

$$\|\tilde{\boldsymbol{p}}_L - p_L\| \;\le\; \|\tilde{\boldsymbol{p}}_L - \bar{p}_L\| + \|\bar{p}_L - p_L\| \;\le\; \|\bar{p}\|\, O(\mathbf{u}) + O(\delta\mathbf{u}); \tag{6.11}$$

$$\|\tilde{\boldsymbol{p}}_S - p_S\| \;\le\; \|\tilde{\boldsymbol{p}}_S - \bar{p}_S\| + \|\bar{p}_S - p_S\| \;\le\; \|\bar{p}\|\, O(\mathbf{u}/\delta) + O(\delta\mathbf{u}); \tag{6.12}$$

$$\|\tilde{\boldsymbol{p}} - p\| \;\le\; \|\tilde{\boldsymbol{p}} - \bar{p}\| + \|\bar{p} - p\| \;\le\; \|\bar{p}\|\, O(\mathbf{u}/\delta) + O(\delta\mathbf{u}). \tag{6.13}$$

The first term in the second bound on $\|\tilde{\boldsymbol{p}}_L - p_L\|$ is smaller than the corresponding bound for the small space by a factor of δ because perturbations in the large space are multiplied by at most $\|\Sigma_L^{-1}\| = O(\delta)$. The first term in the second bound on each of $\|\tilde{\boldsymbol{p}}_S - p_S\|$ and $\|\tilde{\boldsymbol{p}} - p\|$ reveals the worst-case effects of $\operatorname{cond}(M_{\mathrm{pd}})$ when the matrix undergoes a relative perturbation of $O(\mathbf{u})$.

Relation (6.13) shows that

$$\|\tilde{\boldsymbol{p}}\| \approx \|p\| \quad \text{as long as} \quad p = \Omega(\delta). \tag{6.14}$$

Although (6.14) indicates that $\tilde{\boldsymbol{p}}$ and p are similar in norm, it does not imply that they display a small relative error when both are small.

6.2.2. With cancellation. When the constraints are subject to cancellation ($\zeta = \mathbf{u}/\delta$), the perturbation $\mathit{\Delta}$ is $O(\mathbf{u}/\delta^2)$, and the standard bound (6.7) would imply a (horrific) relative error bound for $\tilde{\boldsymbol{p}}$ of $O(\mathbf{u}/\delta^2)$; see (6.25). Fortunately, when we lose because $\|\mathit{\Delta}\|$ is large, there is a countervailing win because the perturbation lies almost entirely in the range of $\hat{A}^T$, the subspace that is not blown up by M_{pd}^{-1}.

Using (3.19), the matrices in (6.8) satisfy $B_{11} = O(\mathbf{u}/\delta^2)$, $B_{12} = O(\mathbf{u}/\delta)$, and $B_{22} = O(\mathbf{u}/\delta)$, so that

$$B_L = O(\mathbf{u}/\delta^2) \quad \text{and} \quad B_S = O(\mathbf{u}/\delta), \tag{6.15}$$

implying that $\|B_S\|$ has a *much smaller* upper bound than $\|B_L\|$, in contrast to the parity of these bounds in the noncancellation case. Hence it is meaningful to apply the first inequalities in (2.16) and (2.17), yielding the following connections between $\tilde{\boldsymbol{p}}$ and $\bar{p}$:

$$\|\tilde{\boldsymbol{p}}_L - \bar{p}_L\| \;\le\; \|\Sigma_L^{-1}\|\,\|B_L\|\,\|\bar{p}\| \;=\; \|\bar{p}\|\, O(\mathbf{u}/\delta); \tag{6.16}$$

$$\|\tilde{\boldsymbol{p}}_S - \bar{p}_S\| \;\le\; \|\Sigma_S^{-1}\|\,\|B_S\|\,\|\bar{p}\| \;=\; \|\bar{p}\|\, O(\mathbf{u}/\delta). \tag{6.17}$$

The bound on $\|\tilde{\boldsymbol{p}}_L - \bar{p}_L\|$ is larger by a factor of $1/\delta$ than the analogous bound (6.9) in the noncancellation case, reflecting the $O(\mathbf{u}/\delta^2)$ large-space perturbation to M_{pd} created by cancellation; see (5.4). However, the bound on $\|\tilde{\boldsymbol{p}}_S - \bar{p}_S\|$ is the same as in the noncancellation case because the relative perturbation to M_{pd} *in the small space* is $O(\mathbf{u})$ with or without cancellation.

Relations (6.6), (6.16), (6.17), and standard norm inequalities imply that, with cancellation, the following bounds apply to the computed $\tilde{\boldsymbol{p}}$ and the exact p:

$$\|\tilde{\boldsymbol{p}}_L - p_L\| \;\le\; \|\bar{p}\|\, O(\mathbf{u}/\delta) + O(\mathbf{u}), \tag{6.18}$$

$$\|\tilde{\boldsymbol{p}}_S - p_S\| \;\le\; \|\bar{p}\|\, O(\mathbf{u}/\delta) + O(\mathbf{u}), \quad \text{and} \tag{6.19}$$

$$\|\tilde{\boldsymbol{p}} - p\| \;\le\; \|\bar{p}\|\, O(\mathbf{u}/\delta) + O(\mathbf{u}). \tag{6.20}$$

Together with (6.6), (6.20) shows that

$$\|\tilde{\boldsymbol{p}}\| \approx \|p\| \quad \text{as long as} \quad p = \Omega(\delta), \tag{6.21}$$

so that $\tilde{\boldsymbol{p}}$ and p are similar in norm, but it does not imply that they display a small relative error when both are small.

6.3. Numerical examples. To illustrate the results just described, we return to example (4.10) with $\mu = 10^{-4}$, $\hat{x}$ from (4.11), and λ from (4.12); the condition number of M_{pd} is $3.2{\times}10^4$.

When we evaluate the constraints in double precision (without cancellation at the single-precision level), performing all other calculations in single precision, the exact p and the computed $\tilde{\boldsymbol{p}}_2$ are

$$p = \begin{pmatrix} -2.212210{\times}10^{-4} \\ -2.339970{\times}10^{-4} \\ 4.419855{\times}10^{-4} \end{pmatrix} \quad \text{and} \quad \tilde{\boldsymbol{p}}_2 = \begin{pmatrix} -2.216191{\times}10^{-4} \\ -2.339966{\times}10^{-4} \\ 4.423836{\times}10^{-4} \end{pmatrix}, \tag{6.22}$$

which gives $\|\tilde{\boldsymbol{p}}_2 - p\| = 5.63{\times}10^{-7}$. Although the overall error in $\tilde{\boldsymbol{p}}_2$ is comparable to machine precision, almost all the error lies in the small space, as predicted by (6.11) and (6.12):

$$\|U_{\scriptscriptstyle L}^T(\tilde{\boldsymbol{p}}_2 - p)\| = 8.46{\times}10^{-11}, \quad \text{whereas} \quad \|U_{\scriptscriptstyle S}^T(\tilde{\boldsymbol{p}}_2 - p)\| = 5.63{\times}10^{-7}. \tag{6.23}$$

When cancellation occurs,

$$\tilde{\boldsymbol{p}} = \begin{pmatrix} -2.216789{\times}10^{-4} \\ -2.339558{\times}10^{-4} \\ 4.424361{\times}10^{-4} \end{pmatrix},$$

which gives $\|\tilde{\boldsymbol{p}} - p\| = 6.43{\times}10^{-7}$. As expected, $\tilde{\boldsymbol{p}}$ has overall accuracy comparable to that of $\tilde{\boldsymbol{p}}_2$; compare (6.13) and (6.20). Note, however, that the large- and small-space parts of $\tilde{\boldsymbol{p}}$ are of similar absolute accuracy, as suggested by the bounds (6.18) and (6.19):

$$\|\tilde{\boldsymbol{p}}_{\scriptscriptstyle L} - p_{\scriptscriptstyle L}\| = 4.10{\times}10^{-8}, \quad \text{and} \quad \|\tilde{\boldsymbol{p}}_{\scriptscriptstyle S} - p_{\scriptscriptstyle S}\| = 6.42{\times}10^{-7}. \tag{6.24}$$

It was observed in section 4.2.1 that, when cancellation causes a large error in the computed M_{pd}, the resulting perturbation tends to lie almost entirely in the range space of $\hat{A}^T$ (see (5.4)). The saving grace of this property can be seen if we perturb M_{pd} by a *random* 3×3 symmetric matrix $\mathit{\Delta} M$ of approximately the same size as $\vartriangle\! M_{\text{pd}}$ of (4.13) and then solve in double precision for the solution $\hat{p}$ of the perturbed system:

$$(M_{\text{pd}} + \mathit{\Delta} M)\hat{p} = b_{\text{p}}, \ \text{ with } \ \|\mathit{\Delta} M\| = 40, \ \text{ so that } \ \frac{\|\mathit{\Delta} M\|}{\|M_{\text{pd}}\|} = 2.3{\times}10^{-4}.$$

The ill-conditioning is obvious from the striking difference between $\hat{p}$ and the exact p:

$$\hat{p} = \begin{pmatrix} 3.899913{\times}10^{-4} \\ -2.344138{\times}10^{-4} \\ -1.693300{\times}10^{-4} \end{pmatrix}, \quad \text{with} \quad \frac{\|\hat{p} - p\|}{\|p\|} = 1.58. \tag{6.25}$$

Even so, the large-space parts of $\hat{p}$ and p remain close. The ill-conditioning magnifies only the part of $\hat{p}$ in the small space:

$$\frac{\|U_L^T(\hat{p}-p)\|}{\|p\|} = 4.3\times10^{-4} \quad \text{and} \quad \frac{\|U_S^T(\hat{p}-p)\|}{\|p\|} = 1.58.$$

7. Calculating the multiplier estimate.

This section shows why, under the conditions of Guideline 5.1, inaccuracies in $\tilde{\boldsymbol{p}}$ resulting from ill-conditioning of M_{pd} have very little effect on the accuracy of the computed multiplier steps, even in the presence of cancellation. From (3.8), the m-vector ℓ, the change in the multiplier estimate, satisfies

$$\Lambda A p + C\ell = \mu\mathbf{1} - C\lambda, \quad \text{so that} \quad \ell = C^{-1}(\mu\mathbf{1} - \Lambda A p) - \lambda. \tag{7.1}$$

Since $\delta = \|x^* - x\|$, a "reasonable" primal-dual step p will satisfy $\|p\| \approx \delta$. Furthermore, we know that $Z^T b_{\mathrm{p}} = O(\delta)$, which means, using (3.20), that $U_S^T b_{\mathrm{p}}$ is $O(\delta)$. It follows from (2.12) that p cannot be too much larger than δ. We thus assume in this section that

$$\|p\| \approx \delta, \quad \text{which implies that} \quad \|\tilde{\boldsymbol{p}}\| \approx \|\bar{p}\| \approx \delta, \tag{7.2}$$

using (6.6), (6.14), and (6.21).

7.1. Inactive constraints.

A component-wise version of (7.1) is

$$\ell_i = \frac{\mu - \lambda_i a_i^T p}{c_i} - \lambda_i. \tag{7.3}$$

Let $\tilde{\boldsymbol{\ell}}_i$ denote the computed change in the ith multiplier estimate, obtained by performing the calculations shown in (7.3) in floating point using $\tilde{\boldsymbol{p}}$ instead of p, and with the computed value of c_i. To analyze $\tilde{\boldsymbol{\ell}}_i - \ell_i$, we apply the rules of floating-point computation (4.1) and the relation $fl(c_i) = c_i(1 + O(\zeta))$ (see section 4.1) to (7.3):

$$\tilde{\boldsymbol{\ell}}_i - \ell_i = \frac{-\lambda_i a_i^T(\tilde{\boldsymbol{p}} - p)}{c_i} + O(\zeta)\,\frac{\mu - \lambda_i a_i^T \tilde{\boldsymbol{p}}}{c_i} + O(\mathbf{u})\,(\lambda_i + \mu + \|\tilde{\boldsymbol{p}}\|). \tag{7.4}$$

In all cases, $a_i = O(1)$ and $\tilde{\boldsymbol{p}} = O(\delta)$.

When constraint i is inactive, $\lambda_i = O(\delta)$ and $c_i = \Theta(1)$, and we have assumed that $\mu = O(\delta)$. Relations (6.13) and (6.20) applied to (7.4) imply that, with or without cancellation,

$$\tilde{\boldsymbol{\ell}}_i - \ell_i = O(\delta\mathbf{u}). \tag{7.5}$$

This bound shows that the multiplier estimates for inactive constraints retain (approximately) full relative precision as they converge to zero.

7.2. Active constraints.

To estimate the accuracy of the multiplier steps for active constraints, we exploit the backward-error formulation of $\tilde{\boldsymbol{p}}$ from section 5. (With cancellation, direct application of the bounds from section 6.2.2 gives an overly pessimistic result.)

Limiting ourselves to active constraints, observe that (7.1) may be written as

$$\hat{\lambda}' \;\equiv\; \hat{\lambda} + \hat{\ell} \;=\; \hat{C}^{-1}(\mu\mathbf{1} - \hat{\Lambda}\hat{A}p), \tag{7.6}$$

where $\hat{\lambda}'$ can be interpreted as the "new" multiplier estimate (if a step of unity is taken along $\hat{\ell}$). We next show that the expression on the right of (7.6) is closely related to the condensed primal-dual equations. By definition, the *exact* solution of $M_{\mathrm{pd}}p = b_{\mathrm{p}}$ satisfies

$$(7.7)\qquad \bigl(W + \hat{A}^T\hat{C}^{-1}\hat{\Lambda}\hat{A} + \bar{A}^T\bar{C}^{-1}\bar{\Lambda}\bar{A}\bigr)p \;=\; -g + \mu\hat{A}^T\hat{C}^{-1}\mathbf{1} + \mu\bar{A}^T\bar{C}^{-1}\mathbf{1},$$

which becomes, after rearrangement,

$$(7.8)\qquad \hat{A}^T(\mu\hat{C}^{-1}\mathbf{1} - \hat{C}^{-1}\hat{\Lambda}\hat{A}p) \;=\; g - \mu\bar{A}^T\bar{C}^{-1}\mathbf{1} + Wp + \bar{A}\bar{C}^{-1}\bar{\Lambda}\bar{A}p.$$

Since range$(\hat{A}^T)$ and null$(\hat{A})$ are orthogonal complements, equality must hold between the range-space parts of both sides of (7.8). To separate the elements of this equation into range$(\hat{A}^T)$ and null$(\hat{A})$, we define

$$(7.9)\qquad \begin{aligned} g = \hat{A}^T g_{\scriptscriptstyle A} + Zg_{\scriptscriptstyle Z}, &\quad \mu\bar{A}^T\bar{C}^{-1}\mathbf{1} \;=\; \hat{A}^T r_{\scriptscriptstyle A} + Zr_{\scriptscriptstyle Z}, \\ Wp = \hat{A}^T w_{\scriptscriptstyle A} + Zw_{\scriptscriptstyle Z}, &\quad \text{and} \quad \bar{A}\bar{C}^{-1}\bar{\Lambda}\bar{A}p = \hat{A}^T a_{\scriptscriptstyle A} + Za_{\scriptscriptstyle Z}. \end{aligned}$$

Substituting from (7.9), rearranging, and using (7.6), we have

$$(7.10)\qquad \hat{\lambda}' \;=\; \hat{C}^{-1}\bigl(\mu\mathbf{1} - \hat{\Lambda}\hat{A}p\bigr) \;=\; g_{\scriptscriptstyle A} - r_{\scriptscriptstyle A} + w_{\scriptscriptstyle A} + a_{\scriptscriptstyle A}.$$

To analyze the computed version of $\hat{\lambda}'$, we revisit the formulation of the perturbed system satisfied by $\tilde{\boldsymbol{p}}$. Let $\hat{\mathbf{c}}_i$ denote $\mathit{fl}(\hat{c}_i)$, an *exact* number satisfying $\hat{\mathbf{c}}_i \equiv \mathit{fl}(\hat{c}_i) = \hat{c}_i + \xi$, where $\xi = O(\mathbf{u})$. Let $\hat{\mathbf{C}}$ denote diag$(\hat{\mathbf{c}}_i)$. The computed version of $\hat{\lambda}'$, denoted by $\hat{\boldsymbol{\lambda}}'$, may then be written as

$$(7.11)\qquad \hat{\boldsymbol{\lambda}}' = \mathit{computed}\Bigl(\hat{\mathbf{C}}^{-1}\bigl(\mu\mathbf{1} - \hat{\Lambda}\hat{A}\tilde{\boldsymbol{p}}\bigr)\Bigr).$$

Because $\hat{\mathbf{c}}_i$ is simply a floating-point number, it satisfies

$$(7.12)\qquad \mathit{fl}\Bigl(\frac{1}{\hat{\mathbf{c}}_i}\Bigr) = \mathit{computed}\Bigl(\frac{1}{\mathit{fl}(\hat{c}_i)}\Bigr) = \Bigl(\frac{1}{\hat{\mathbf{c}}_i}\Bigr)(1 + O(\mathbf{u})),$$

with only an $O(\mathbf{u})$ relative difference between $\mathit{computed}(1/\mathit{fl}(\hat{c}_i))$ and $1/\hat{\mathbf{c}}_i$.

The definition (7.12), the assumption that $c_{\min} = \Omega(\delta)$, and standard floating-point rules (4.1) lead to the following relationships:

$$(7.13)\qquad \begin{aligned} \mathit{computed}(\hat{A}^T\hat{C}^{-1}\hat{\Lambda}\hat{A}) &= \hat{A}^T\hat{\mathbf{C}}^{-1}\hat{\Lambda}\hat{A} + O(\mathbf{u}/\delta) \quad \text{and} \\ \mathit{computed}(\mu\hat{A}^T\hat{C}^{-1}\mathbf{1}) &= \mu\hat{A}^T\hat{\mathbf{C}}^{-1}\mathbf{1} + O(\mathbf{u}). \end{aligned}$$

The key here is that $\hat{\mathbf{C}}^{-1}$ appears in the computed versions of *both* the matrix and the right-hand side of (7.7); i.e., the same (albeit possibly inaccurate) quantities serve as the active constraint values throughout the computation.

Using (7.13) and reasoning like that leading to (5.6), we can show that the computed solution $\tilde{\boldsymbol{p}}$ satisfies a relation very similar to (7.7), except that the matrix multiplying $\tilde{\boldsymbol{p}}$ involves $\hat{\mathbf{C}}$ and contains a perturbation $\boldsymbol{\Delta}$:

$$(7.14)\quad \bigl(W + \hat{A}^T\hat{\mathbf{C}}^{-1}\hat{\Lambda}\hat{A} + \bar{A}^T\bar{C}^{-1}\bar{\Lambda}\bar{A} + \boldsymbol{\Delta}\bigr)\tilde{\boldsymbol{p}} \;=\; -g + \mu\hat{A}^T\hat{\mathbf{C}}^{-1}\mathbf{1} + \mu\bar{A}^T\bar{C}^{-1}\mathbf{1},$$

where $\boldsymbol{\Delta} = O(\mathbf{u}/\delta)$. Exactly as with (7.9), we separate out the range-space parts of this equation and obtain an analogue of (7.10):

$$\hat{\mathbf{C}}^{-1}\big(\mu\mathbf{1} - \hat{\Lambda}\hat{A}\tilde{\boldsymbol{p}}\big) \;=\; g_A - r_A + \tilde{w}_A + a_A + d_A, \tag{7.15}$$

$$\text{where} \quad W\tilde{\boldsymbol{p}} \;=\; \hat{A}^T\tilde{w}_A + Z\tilde{w}_Z \quad \text{and} \quad \boldsymbol{\Delta}\tilde{\boldsymbol{p}} = \hat{A}^T d_A + Z d_Z. \tag{7.16}$$

The vector d_A is $O(\mathbf{u})$; this follows from the relations $\boldsymbol{\Delta} = O(\mathbf{u}/\delta)$ and $\|\tilde{\boldsymbol{p}}\| \approx \delta$.

Two points emerge from this analysis. First, by combining (7.10) and (7.15) we have

$$\hat{\mathbf{C}}^{-1}\big(\mu\mathbf{1} - \hat{\Lambda}\hat{A}\tilde{\boldsymbol{p}}\big) \;=\; \hat{\lambda}' + \tilde{w}_A - w_A + d_A. \tag{7.17}$$

Since $W = O(1)$ and $\tilde{\boldsymbol{p}} - p = O(\mathbf{u})$ (see (6.13) and (6.20)), the vector $\tilde{w}_A - w_A$ is $O(\mathbf{u})$. Thus

$$\hat{\mathbf{C}}^{-1}\big(\mu\mathbf{1} - \hat{\Lambda}\hat{A}\tilde{\boldsymbol{p}}\big) \;=\; \hat{\lambda}' + O(\mathbf{u}). \tag{7.18}$$

Second, the computed version of the expression on the left-hand side of (7.18) is precisely $\hat{\boldsymbol{\lambda}}'$ (see (7.11)). Using the floating-point rules (4.1) and relation (7.12), we obtain

$$\hat{\boldsymbol{\lambda}}' \;=\; \mathit{computed}\Big(\hat{\mathbf{C}}^{-1}\big(\mu\mathbf{1} - \hat{\Lambda}\hat{A}\tilde{\boldsymbol{p}}\big)\Big) \;=\; \hat{\mathbf{C}}^{-1}\big(\mu\mathbf{1} - \hat{\Lambda}\hat{A}\tilde{\boldsymbol{p}}\big) + O(\mathbf{u}). \tag{7.19}$$

Combining (7.18) and (7.19) gives

$$\hat{\boldsymbol{\lambda}}' = \hat{\lambda}' + O(\mathbf{u}), \quad \text{so that} \quad \hat{\boldsymbol{\ell}} = \hat{\ell} + O(\mathbf{u}), \tag{7.20}$$

since subtracting $\hat{\lambda}$ from $\hat{\boldsymbol{\lambda}}'$ to produce $\hat{\boldsymbol{\ell}}$ introduces only one further error of $O(\mathbf{u})$.

In many ways this result is quite remarkable: despite the possibility of substantial relative error in both the matrix and right-hand side, the vector $\hat{\boldsymbol{\ell}}$ calculated using $\tilde{\boldsymbol{p}}$ differs from the exact $\hat{\ell}$ by (approximately) machine precision. Thus, since $\hat{\lambda}' = \Theta(1)$, we are able to obtain *close to full precision* in the new multiplier estimate, even with cancellation.

7.3. Numerical examples. We return once more to example (4.10), with $\hat{x}$ from (4.11), λ from (4.12), and $\mu = 10^{-4}$. Let λ' denote $\lambda + \ell$, and $\tilde{\boldsymbol{\lambda}}'$ its computed version, so that the components of $\tilde{\boldsymbol{\lambda}}'$ corresponding to active constraints are given by $\hat{\boldsymbol{\lambda}}'$. The exact λ' and computed $\tilde{\boldsymbol{\lambda}}'$ are

$$\lambda' = \begin{pmatrix} 1.9999053 \\ 5.7258318{\times}10^{-5} \\ 3.3333196 \end{pmatrix} \quad \text{and} \quad \tilde{\boldsymbol{\lambda}}' = \begin{pmatrix} 1.9999059 \\ 5.7258585{\times}10^{-5} \\ 3.3333199 \end{pmatrix},$$

and the exact ℓ and computed $\tilde{\boldsymbol{\ell}}$ are

$$\ell = \begin{pmatrix} -8.271445{\times}10^{-4} \\ -9.193042{\times}10^{-4} \\ -7.461919{\times}10^{-4} \end{pmatrix} \quad \text{and} \quad \tilde{\boldsymbol{\ell}} = \begin{pmatrix} -8.264780{\times}10^{-4} \\ -9.193039{\times}10^{-4} \\ -7.457733{\times}10^{-4} \end{pmatrix}. \tag{7.21}$$

Thus the error is

$$\tilde{\boldsymbol{\ell}} - \ell = \begin{pmatrix} 6.6646{\times}10^{-7} \\ 2.6671{\times}10^{-10} \\ 4.1858{\times}10^{-7} \end{pmatrix}, \tag{7.22}$$

which corresponds well to the bounds of (7.5) and (7.20). Note the better absolute accuracy of the component corresponding to the single inactive constraint, as predicted by (7.5).

8. Alternative derivations of the error bounds. After publication of [36] (the initial version of the present paper), R. H. Byrd [3] pointed out to the author that the bound (6.20) could be derived by relating $\tilde{\boldsymbol{p}}$ to the exact solution of a perturbed version of the primal-dual system (3.8) in which both the matrix and right-hand side include perturbations attributable to cancellation. In a referee's report on the present paper, S. J. Wright sketched a derivation of the error bounds based on a similar observation. We now summarize this derivation.

Along the lines of the discussion in section 7.2, let $\mathbf{c}$ denote $fl(c)$. When the constraints are subject to cancellation,

$$\mathbf{c} = c + \boldsymbol{\xi}, \quad \text{with} \quad \boldsymbol{\xi} = O(\mathbf{u}). \tag{8.1}$$

Consider the (exact) system that contains the computed constraints,

$$\begin{pmatrix} W(\lambda) & -A^T \\ \varLambda A & \mathbf{C} \end{pmatrix} \begin{pmatrix} p' \\ \ell' \end{pmatrix} = \begin{pmatrix} -g + A^T\lambda \\ \mu\mathbf{1} - \mathbf{C}\lambda \end{pmatrix}, \tag{8.2}$$

and observe that the matrix and right-hand side of the computed condensed system (3.9) can be interpreted as the result of performing block elimination on (8.2). The numerical solution of the computed condensed system can be analyzed by bounding first the errors associated with carrying out this block elimination, and then the errors arising from solution of the resulting system with a backward-stable method. It follows from this analysis that $\tilde{\boldsymbol{p}}$ satisfies

$$\begin{aligned} (W(\lambda) + A^T\mathbf{C}^{-1}\varLambda A + \varDelta_W)\tilde{\boldsymbol{p}} &= -g + \mu A^T\mathbf{C}^{-1}\mathbf{1} + \vartriangle_b, \quad \text{where} \\ \varDelta_W &= O(\mathbf{u}/c_{\min}) \quad \text{and} \quad \vartriangle_b = O(\mu\mathbf{u}/c_{\min}). \end{aligned} \tag{8.3}$$

It can then be shown that $\tilde{\boldsymbol{p}}$ and an approximate multiplier step satisfy a perturbed version of the exact primal-dual system (3.8), where the perturbations are $\varDelta_W$, $\vartriangle_b$, and $\boldsymbol{\xi}$ of (8.1). Since the full primal-dual matrix is well conditioned by assumption, an expression involving these perturbations can be derived that relates the computed $\tilde{\boldsymbol{p}}$ and the exact p, leading to the bound (6.20). By partitioning the constraints into active and inactive, the bounds of section 7 on the computed multiplier estimates can be derived. When cancellation does not occur in the constraints, S. J. Wright [41] has subsequently shown how to derive the sharper bound (6.13) for the error in the large-space component $\tilde{\boldsymbol{p}}_L$ using the approach described above and the relation (3.17).

A complete presentation of the alternative approach is given in [40], which discusses the effects of finite precision in interior-point methods when the Mangasarian–Fromowitz constraint qualification holds (rather than the stronger assumption made here of linear independence of the active constraint normals).

9. Solving the full primal-dual system. We have examined in detail the properties of the vectors $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{\ell}}$ obtained by solving the condensed (necessarily ill-conditioned) primal-dual equations. To complete our analysis, we consider the accuracy of the vectors resulting from solving the full primal-dual system $Pz = d$, with

$$P \equiv \begin{pmatrix} W(\lambda) & -A^T \\ \varLambda A & C \end{pmatrix}, \quad z \equiv \begin{pmatrix} p \\ \ell \end{pmatrix}, \quad \text{and} \quad d \equiv \begin{pmatrix} -g + A^T\lambda \\ \mu\mathbf{1} - C\lambda \end{pmatrix}. \tag{9.1}$$

We stress that the condition of P corresponds to the condition of the original constrained problem; see, for example, [14].

Suppose that we wish to solve $Pz = d$ at a point (x, λ) satisfying the conditions listed in Guideline 5.1, which means that $P = \Theta(1)$. Let us make the further assumption that $P^{-1} = \Theta(1)$, so that $\text{cond}(P) \approx 1$ and the full system is perfectly conditioned. Then the accuracy that can be expected in $\tilde{\boldsymbol{z}}$, the computed z, depends directly on the size of the perturbations associated with representing P and d in finite precision.

Since the elements of P are $O(1)$, the absolute error in *computed*(P) is $O(\mathbf{u})$, as is the relative error $\|\mathit{computed}(P) - P\|/\|P\|$. We now show that the *absolute* error in *computed*(d) is $O(\mathbf{u})$. Since g is $\Theta(1)$, its computed version will in general have an error that is $O(\mathbf{u})$; therefore *computed*$(-g + A^T\lambda)$, the first block of the computed d in (9.1), has an absolute error that is $O(\mathbf{u})$ and independent of whether or not cancellation occurs. When the constraints are subject to cancellation, each component of c has an associated absolute error of $O(\mathbf{u})$. Since the multiplier estimate for an active constraint is $\Theta(1)$, the computed version of $\lambda_i c_i$ will also contain an absolute error of $O(\mathbf{u})$. Consequently, with cancellation the second block of the computed d in (9.1) contains an $O(\mathbf{u})$ error in each component corresponding to an active constraint. Although the absolute error in *computed*(d) is $O(\mathbf{u})$, d itself is *not* $O(1)$. In fact, we expect each component of d to be $O(\delta)$.

If only the matrix P were subject to computational errors, relation (2.4) would imply a relative error of $O(\mathbf{u})$ in the computed solution. But since the *absolute* difference between *computed*(d) and d is $O(\mathbf{u})$, the first inequality in (2.1) implies that

$$\|\tilde{\boldsymbol{z}} - z\| \le \|P^{-1}\| \, \|\mathit{computed}(d) - d\| = O(\mathbf{u}). \tag{9.2}$$

Thus, because of absolute errors that are $O(\mathbf{u})$ in the computed d, the numerical solution of the well-conditioned system (9.1) produces steps in x and λ that may contain *absolute* errors comparable to machine precision.

The import of this result can be seen by recalling that, when $\delta/c_{\min} = O(1)$, the primal-dual steps $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{\ell}}$ for the active constraints computed from the condensed matrix M_{pd} have essentially the same bound (an $O(1)$ multiple of machine precision) on their deviations from the exact p and ℓ (see (6.13), (6.20), and (7.20)). Rather surprisingly, we are only marginally better off with the full primal-dual matrix P than with its ill-conditioned cousin M_{pd}! (In effect, the favorable condition of P cannot overcome the inherent error in the right-hand side.)

This phenomenon can be seen in our familiar example (4.10), with $\hat{x}$ from (4.11), λ from (4.12), and $\mu = 10^{-4}$. If we form P and d and solve $Pz = d$ in single precision, the computed d and $\tilde{\boldsymbol{z}}$ satisfy

$$\mathit{computed}(d) - d = \begin{pmatrix} -2.166\times10^{-7} \\ -5.962\times10^{-7} \\ 1.788\times10^{-7} \\ -1.193\times10^{-7} \\ -5.342\times10^{-11} \\ 4.075\times10^{-7} \end{pmatrix} \quad\text{and}\quad \tilde{\boldsymbol{z}} = \begin{pmatrix} -2.212746\times10^{-4} \\ -2.339562\times10^{-4} \\ 4.420320\times10^{-4} \\ -8.270238\times10^{-4} \\ -9.193041\times10^{-4} \\ -7.457245\times10^{-4} \end{pmatrix}. \tag{9.3}$$

Observe that the absolute error in each component of *computed*(d) is similar in size to $\mathbf{u}$, except for the component in the second block corresponding to the inactive constraint. Splitting $\tilde{\boldsymbol{z}}$ into its p and ℓ subvectors, we have

$$(9.4)\qquad (\tilde{\boldsymbol{z}})_p - p = \begin{pmatrix} -5.368{\times}10^{-8} \\ 4.074{\times}10^{-8} \\ 4.646{\times}10^{-8} \end{pmatrix} \text{ and } (\tilde{\boldsymbol{z}})_\ell - \ell = \begin{pmatrix} 1.206{\times}10^{-7} \\ 3.389{\times}10^{-11} \\ 4.674{\times}10^{-7} \end{pmatrix},$$

which gives $\|\tilde{\boldsymbol{z}} - z\| = 4.896{\times}10^{-7}$. (The exact p is given in (6.22) and the exact ℓ in (7.21).) All components are shown to confirm that neither the step in x nor the step in λ is significantly more accurate than $\tilde{\boldsymbol{p}}$ or $\tilde{\boldsymbol{\ell}}$; compare (9.4) with (6.24) and (7.22). The more accurate component of $(\tilde{\boldsymbol{z}})_\ell$ corresponding to the single inactive constraint, like the same component in $\tilde{\boldsymbol{\ell}} - \ell$ of (7.22), can be explained via an analysis along the lines of section 7.1.

10. Summary and conclusions. This paper contains several related results. First, under conditions that usually hold in the final stages of a primal-dual method, the exact condensed primal-dual matrix M_{pd} (3.9) is structurally ill-conditioned. Like the primal barrier Hessian, M_{pd} has two widely separated sets of eigenvalues, where the invariant subspace corresponding to the large eigenvalues (the large space) is close to the range of $\hat{A}^T$ (the transposed Jacobian of the active constraints), and the complementary small space is close to the null space of $\hat{A}$.

Active constraints computed in a standard way are likely to be subject to cancellation, thereby degrading their relative accuracy. With cancellation, we have shown that both the condensed matrix and the right-hand side are likely to experience relative perturbations much larger than machine precision. However, these lie almost entirely in the range space of $\hat{A}^T$ and hence are not magnified by the ill-conditioning in M_{pd}.

If the condensed system is solved using any backward-stable method, then $\tilde{\boldsymbol{p}}$, the computed step in x, can be characterized as the exact solution of a perturbed system. If x and λ are close enough to optimal (as measured by (3.10) for sufficiently small δ), and if $c_{\min} = \Omega(\delta)$ and $\mu = O(\delta)$, this backward-error form leads to two conclusions:

(i) without cancellation, the large-space part of $\tilde{\boldsymbol{p}}$ is much more accurate than the small-space part, whose absolute accuracy is bounded by an $O(1)$ multiple of machine precision;

(ii) with cancellation, both large- and small-space parts of $\tilde{\boldsymbol{p}}$ have the same bound—an $O(1)$ multiple of machine precision—on their absolute accuracy.

Based on these properties of $\tilde{\boldsymbol{p}}$, we have also demonstrated that, with or without cancellation, the step $\tilde{\boldsymbol{\ell}}$ in λ for the active constraints calculated using $\tilde{\boldsymbol{p}}$ has an absolute error bounded by an $O(1)$ multiple of machine precision. For the inactive constraints, the multiplier step retains (approximately) full relative precision.

Finally, we have noted that the computed right-hand side in the full primal-dual system will almost always (because of finite precision) be subject to absolute errors that are $O(\mathbf{u})$. Hence, even though the matrix is well-conditioned, the steps in x and λ calculated from the full system can be expected to contain absolute errors of order machine precision—i.e., errors not much smaller than those associated with steps computed from the ill-conditioned condensed system.

Because of the intermingled effects of structure, cancellation, and asymptotic properties of the primal-dual iterates, we conclude that in most cases ill-conditioning in the condensed matrix impairs only marginally the accuracy of the computed results. Although solving ill-conditioned systems should emphatically be avoided in general, we now understand why, in this very special case, the negative consequences of ill-conditioning are likely to be imperceptible.

Obvious generalizations of the same approach apply to the effects of computing the search direction from the ill-conditioned Hessian in a primal barrier method. We believe that the results in this paper may explain the lack of documented difficulties in primal barrier methods that are explicitly attributable to ill-conditioning rather than to poor properties of the Newton barrier direction.

Acknowledgments. I thank the referees very much for their careful reading of the manuscript, their detection of several errors, and their interesting perspectives on ill-conditioning in primal-dual methods.

REFERENCES

[1] J. M. Banoczi, N.-C. Chiu, G. E. Cho, and I. C. F. Ipsen, *The Influence of the Right-Hand Side on the Accuracy of Linear System Solution*, Technical Report, North Carolina State University, Raleigh, NC, 1996.

[2] J. R. Bunch, J. W. Demmel, and C. F. Van Loan, *The strong stability of algorithms for solving symmetric linear systems*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 494–499.

[3] R. H. Byrd, *Private communication*, 1997.

[4] R. H. Byrd, J. C. Gilbert, and J. Nocedal, *A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming*, Technical Report OTC 96-02, Northwestern University, Evanston, IL, 1996.

[5] T. F. Chan and D. E. Foulser, *Effectively well-conditioned linear systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 963–969.

[6] S. H. Cheng and N. J. Higham, *A modified Cholesky algorithm based on a symmetric indefinite factorization*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 1097–1110.

[7] S. Christiansen and P. C. Hansen, *The effective condition number applied to error analysis of certain boundary collocation methods*, J. Comput. Appl. Math., 54 (1994), pp. 15–36.

[8] A. R. Conn, N. I. M. Gould, and P. L. Toint, *A Primal-Dual Algorithm for Minimizing a Non-convex Function Subject to Bound and Linear Equality Constraints*, Report RC 20639, IBM T. J. Watson Research Center, Yorktown Heights, NY, 1996.

[9] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang, *On the formulation and theory of the Newton interior-point method for nonlinear programming*, J. Optim. Theory Appl., 89 (1996), pp. 507–541.

[10] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York, 1968; republished by the Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[11] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley and Sons, Chichester, 1987.

[12] A. Forsgren and P. E. Gill, *Primal-dual interior methods for nonconvex nonlinear programming*, SIAM J. Optim., 8 (1998), pp. 1132–1152.

[13] A. Forsgren, P. E. Gill, and W. Murray, *Computing modified Newton directions using a partial Cholesky factorization*, SIAM J. Sci. Comput., 16 (1995), pp. 139 –150.

[14] A. Forsgren, P. E. Gill, and J. R. Shinnerl, *Stability of symmetric ill-conditioned systems arising in interior methods for constrained optimization*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 187–211.

[15] D. M. Gay, M. L. Overton, and M. H. Wright, *A primal-dual interior method for nonconvex nonlinearly constrained optimization*, in Advances in Nonlinear Programming, Y. Yuan, ed., Kluwer, Dordrecht, 1998, pp. 31–56.

[16] P. E. Gill, W. Murray and M. H. Wright, *Practical Optimization*, Academic Press, London, New York, 1981.

[17] P. E. Gill, W. Murray, and M. H. Wright, *Numerical Linear Algebra and Optimization*, Vol. 1, Addison-Wesley, Redwood City, CA, 1991.

[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[19] C. C. Gonzaga, *Path-following methods for linear programming*, SIAM Rev., 34 (1992), pp. 167–224.

[20] N. I. M. Gould, *On the accurate determination of search directions for simple differentiable penalty functions*, IMA J. Numer. Anal., 6 (1986), pp. 357–372.

[21] N. J. Higham, *Analysis of the Cholesky decomposition of a semi-definite matrix*, in Reliable Numerical Computation, M. G. Cox and S. Hammarling, eds., Clarendon Press, Oxford, 1990, pp. 161–185.

[22] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1996.

[23] F. A. Lootsma, *Hessian Matrices of Penalty Functions for Solving Constrained Optimization Problems*, Philips Res. Repts. 24, Eindhoven, The Netherlands, 1969, pp. 322–331.

[24] G. P. McCormick, *The Superlinear Convergence of a Nonlinear Primal-Dual Algorithm*, Report T-550/91, Department of Operations Research, George Washington University, Washington, DC, 1991.

[25] W. Murray, *Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions*, J. Optim. Theory Appl., 7 (1971), pp. 189–196.

[26] C. R. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[27] M. A. Saunders, *Major Cholesky would feel proud*, ORSA J. Comput., 6 (1994), pp. 23–27.

[28] R. B. Schnabel and E. Eskow, *A new modified Cholesky factorization*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 1136–1158.

[29] R. K. Smith, *Private communication*, 1997.

[30] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[31] G. W. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

[32] A. van der Sluis, *Stability of solutions of linear algebraic systems*, Numer. Math., 14 (1970), pp. 246–251.

[33] M. H. Wright, *Interior methods for constrained optimization*, in Acta Numerica 1992, A. Iserles, ed., Cambridge University Press, NY, 1992, pp. 341–407.

[34] M. H. Wright, *Some properties of the Hessian of the logarithmic barrier function*, Math. Programming, 67 (1994), pp. 265–295.

[35] M. H. Wright, *Why a pure primal Newton barrier step may be infeasible*, SIAM J. Optim., 5 (1995), pp. 1–12.

[36] M. H. Wright, *Ill-Conditioning and Computational Error in Interior Methods for Nonlinear Programming*, Technical Report 97-4-04, Computing Sciences Research Center, Bell Laboratories, Murray Hill, NJ, 1997.

[37] S. J. Wright, *Stability of linear equation solvers in interior-point methods*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1287–1307.

[38] S. J. Wright, *Modified Cholesky Factorizations in Interior-Point Algorithms for Linear Programming*, Technical Report ANL/MCS-P600-0596, Argonne National Laboratory, Argonne, IL, 1996.

[39] S. J. Wright, *Stability of augmented system factorizations in interior-point methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 191–222.

[40] S. J. Wright, *Finite-Precision Effects on the Local Convergence of Interior-Point Algorithms for Nonlinear Programming*, Preprint ANL/MCS P705-0198, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1998.

[41] S. J. Wright, *Private communication*, 1998.

SIAM J. OPTIM.
Vol. 9, No. 1, pp. 112–147

CONVERGENCE PROPERTIES OF THE NELDER–MEAD SIMPLEX METHOD IN LOW DIMENSIONS[*]

JEFFREY C. LAGARIAS[†], JAMES A. REEDS[‡], MARGARET H. WRIGHT[§], AND PAUL E. WRIGHT[¶]

Abstract. The Nelder–Mead simplex algorithm, first published in 1965, is an enormously popular direct search method for multidimensional unconstrained minimization. Despite its widespread use, essentially no theoretical results have been proved explicitly for the Nelder–Mead algorithm. This paper presents convergence properties of the Nelder–Mead algorithm applied to strictly convex functions in dimensions 1 and 2. We prove convergence to a minimizer for dimension 1, and various limited convergence results for dimension 2. A counterexample of McKinnon gives a family of strictly convex functions in two dimensions and a set of initial conditions for which the Nelder–Mead algorithm converges to a nonminimizer. It is not yet known whether the Nelder–Mead method can be proved to converge to a minimizer for a more specialized class of convex functions in two dimensions.

Key words. direct search methods, Nelder–Mead simplex methods, nonderivative optimization

AMS subject classifications. 49D30, 65K05

PII. S1052623496303470

1. Introduction.

Since its publication in 1965, the Nelder–Mead "simplex" algorithm [6] has become one of the most widely used methods for nonlinear unconstrained optimization. The Nelder–Mead algorithm should not be confused with the (probably) more famous simplex algorithm of Dantzig for linear programming; both algorithms employ a sequence of simplices but are otherwise completely different and unrelated—in particular, the Nelder–Mead method is intended for unconstrained optimization.

The Nelder–Mead algorithm is especially popular in the fields of chemistry, chemical engineering, and medicine. The recent book [16], which contains a bibliography with thousands of references, is devoted entirely to the Nelder–Mead method and variations. Two measures of the ubiquity of the Nelder–Mead method are that it appears in the best-selling handbook *Numerical Recipes* [7], where it is called the "amoeba algorithm," and in MATLAB [4].

The Nelder–Mead method attempts to minimize a scalar-valued nonlinear function of n real variables using only function values, without any derivative information (explicit or implicit). The Nelder–Mead method thus falls in the general class of *direct search methods*; for a discussion of these methods, see, for example, [13, 18]. A large subclass of direct search methods, including the Nelder–Mead method, maintain at each step a nondegenerate *simplex*, a geometric figure in n dimensions of nonzero volume that is the convex hull of $n + 1$ vertices.

Each iteration of a simplex-based direct search method begins with a simplex, specified by its $n + 1$ vertices and the associated function values. One or more test points are computed, along with their function values, and the iteration terminates

[*] Received by the editors May 13, 1996; accepted for publication (in revised form) November 24, 1997; published electronically December 2, 1998.
http://www.siam.org/journals/siopt/9-1/30347.html

[†] AT&T Labs–Research, Florham Park, NJ 07932 (jcl@research.att.com).
[‡] AT&T Labs–Research, Florham Park, NJ 07932 (reeds@research.att.com).
[§] Bell Laboratories, Murray Hill, NJ 07974 (mhw@research.bell-labs.com).
[¶] AT&T Labs–Research, Florham Park, NJ 07932 (pew@research.att.com).

with a new (different) simplex such that the function values at its vertices satisfy some form of descent condition compared to the previous simplex. Among such algorithms, the Nelder–Mead algorithm is particularly parsimonious in function evaluations per iteration, since in practice it typically requires only one or two function evaluations to construct a new simplex. (Several popular direct search methods use n or more function evaluations to obtain a new simplex.) There is a wide array of folklore about the Nelder–Mead method, mostly along the lines that it works well in "small" dimensions and breaks down in "large" dimensions, but very few careful numerical results have been published to support these perceptions. Apart from the discussion in [12], little attention has been paid to a systematic analysis of why the Nelder–Mead algorithm fails or breaks down numerically, as it often does.

Remarkably, there has been no published theoretical analysis explicitly treating the *original* Nelder–Mead algorithm in the more than 30 years since its publication. Essentially no convergence results have been proved, although in 1985 Woods [17] studied a modified[1] Nelder–Mead algorithm applied to a strictly convex function. The few known facts about the original Nelder–Mead algorithm consist mainly of negative results. Woods [17] displayed a nonconvex example in two dimensions for which the Nelder–Mead algorithm converges to a nonminimizing point. Very recently, McKinnon [5] gave a family of strictly convex functions and a starting configuration in two dimensions for which all vertices in the Nelder–Mead method converge to a nonminimizing point.

The theoretical picture for other direct search methods is much clearer. Torczon [13] proved that "pattern search" algorithms converge to a stationary point when applied to a general smooth function in n dimensions. Pattern search methods, including multidirectional search methods [12, 1], maintain uniform linear independence of the simplex edges (i.e., the dihedral angles are uniformly bounded away from zero and π) and require only simple decrease in the best function value at each iteration. Rykov [8, 9, 10] introduced several direct search methods that converge to a minimizer for strictly convex functions. In the methods proposed by Tseng [15], a "fortified descent" condition—stronger than simple descent—is required, along with uniform linear independence of the simplex edges. Depending on a user-specified parameter, Tseng's methods may involve only a small number of function evaluations at any given iteration and are shown to converge to a stationary point for general smooth functions in n dimensions.

Published convergence analyses of simplex-based direct search methods impose one or both of the following requirements: (i) the edges of the simplex remain uniformly linearly independent at every iteration; (ii) a descent condition stronger than simple decrease is satisfied at every iteration. In general, the Nelder–Mead algorithm fails to have either of these properties; the resulting difficulties in analysis may explain the long-standing lack of convergence results.

Because the Nelder–Mead method is so widely used by practitioners to solve important optimization problems, we believe that its theoretical properties should be understood as fully as possible. This paper presents convergence results in one and two dimensions for the original Nelder–Mead algorithm applied to strictly convex functions with bounded level sets. Our approach is to consider the Nelder–Mead algorithm

[1] The modifications in [17] include a contraction acceptance test different from the one given in the Nelder–Mead paper and a "relative decrease" condition (stronger than simple decrease) for accepting a reflection step. Woods did not give any conditions under which the iterates converge to the minimizer.

as a discrete dynamical system whose iterations are "driven" by the function values. Combined with strict convexity of the function, this interpretation implies restrictions on the allowed sequences of Nelder–Mead moves, from which convergence results can be derived. Our main results are as follows:

1. In dimension 1, the Nelder–Mead method converges to a minimizer (Theorem 4.1), and convergence is eventually M-step linear[2] when the reflection parameter $\rho = 1$ (Theorem 4.2).
2. In dimension 2, the function values at all simplex vertices in the standard Nelder–Mead algorithm converge to the same value (Theorem 5.1).
3. In dimension 2, the simplices in the standard Nelder–Mead algorithm have diameters converging to zero (Theorem 5.2).

Note that Result 3 does *not* assert that the simplices converge to a single point $\mathbf{x}_*$. No example is known in which the iterates fail to converge to a single point, but the issue is not settled.

For the case of dimension 1, Torczon [14] has recently informed us that some convergence results for the original Nelder–Mead algorithm can be deduced from the results in [13]; see section 4.4. For dimension 2, our results may appear weak, but the McKinnon example [5] shows that convergence to a minimizer is not guaranteed for general strictly convex functions in dimension 2. Because the smoothest McKinnon example has a point of discontinuity in the fourth derivatives, a logical question is whether or not the Nelder–Mead method converges to a minimizer in two dimensions for a more specialized class of strictly convex functions—in particular, for smooth functions. This remains a challenging open problem. At present there is no function in any dimension greater than 1 for which the original Nelder–Mead algorithm has been proved to converge to a minimizer.

Given all the known inefficiencies and failures of the Nelder–Mead algorithm (see, for example, [12]), one might wonder why it is used *at all*, let alone why it is so extraordinarily popular. We offer three answers. First, in many applications, for example in industrial process control, one simply wants to find parameter values that improve some performance measure; the Nelder–Mead algorithm typically produces significant improvement for the first few iterations. Second, there are important applications where a function evaluation is enormously expensive or time-consuming, but derivatives cannot be calculated. In such problems, a method that requires at least n function evaluations at every iteration (which would be the case if using finite-difference gradient approximations or one of the more popular pattern search methods) is too expensive or too slow. When it succeeds, the Nelder–Mead method tends to require substantially fewer function evaluations than these alternatives, and its relative "best-case efficiency" often outweighs the lack of convergence theory. Third, the Nelder–Mead method is appealing because its steps are easy to explain and simple to program.

In light of weaknesses exposed by the McKinnon counterexample and the analysis here, future work involves developing methods that retain the good features of the Nelder–Mead method but are more reliable and efficient in theory and practice; see, for example, [2].

The contents of this paper are as follows. Section 2 describes the Nelder–Mead algorithm, and section 3 gives its general properties. For a strictly convex function

[2] By M-step linear convergence we mean that there is an integer M, independent of the function being minimized, such that the simplex diameter is reduced by a factor no less than 1/2 after M iterations.

with bounded level sets, section 4 analyzes the Nelder–Mead method in one dimension, and section 5 presents limited convergence results for the standard Nelder–Mead algorithm in two dimensions. Finally, section 6 discusses open problems.

2. The Nelder–Mead algorithm.

The Nelder–Mead algorithm [6] was proposed as a method for minimizing a real-valued function $f(\mathbf{x})$ for $\mathbf{x} \in \mathcal{R}^n$. Four scalar parameters must be specified to define a complete Nelder–Mead method: coefficients of *reflection* (ρ), *expansion* (χ), *contraction* (γ), and *shrinkage* (σ). According to the original Nelder–Mead paper, these parameters should satisfy

$$\rho > 0, \quad \chi > 1, \quad \chi > \rho, \quad 0 < \gamma < 1, \quad \text{and} \quad 0 < \sigma < 1. \tag{2.1}$$

(The relation $\chi > \rho$, while not stated explicitly in the original paper, is implicit in the algorithm description and terminology.) The nearly universal choices used in the *standard* Nelder–Mead algorithm are

$$\rho = 1, \quad \chi = 2, \quad \gamma = \tfrac{1}{2}, \quad \text{and} \quad \sigma = \tfrac{1}{2}. \tag{2.2}$$

We assume the general conditions (2.1) for the one-dimensional case but restrict ourselves to the standard case (2.2) in the two-dimensional analysis.

2.1. Statement of the algorithm.

At the beginning of the kth iteration, $k \geq 0$, a nondegenerate simplex Δ_k is given, along with its $n+1$ vertices, each of which is a point in $\mathcal{R}^n$. It is always assumed that iteration k begins by ordering and labeling these vertices as $\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_{n+1}^{(k)}$, such that

$$f_1^{(k)} \leq f_2^{(k)} \leq \cdots \leq f_{n+1}^{(k)}, \tag{2.3}$$

where $f_i^{(k)}$ denotes $f(\mathbf{x}_i^{(k)})$. The kth iteration generates a set of $n+1$ vertices that define a different simplex for the next iteration, so that $\Delta_{k+1} \neq \Delta_k$. Because we seek to minimize f, we refer to $\mathbf{x}_1^{(k)}$ as the *best* point or vertex, to $\mathbf{x}_{n+1}^{(k)}$ as the *worst* point, and to $\mathbf{x}_n^{(k)}$ as the *next-worst* point. Similarly, we refer to $f_{n+1}^{(k)}$ as the worst function value, and so on.

The 1965 paper [6] contains several ambiguities about strictness of inequalities and tie-breaking that have led to differences in interpretation of the Nelder–Mead algorithm. What we shall call "the" Nelder–Mead algorithm (Algorithm NM) includes well-defined tie-breaking rules, given below, and accepts the better of the reflected and expanded points in step 3 (see the discussion in section 3.1 about property 4 of the Nelder–Mead method).

A single generic iteration is specified, omitting the superscript k to avoid clutter. The result of each iteration is either (1) a single new vertex—the *accepted point*—which replaces $\mathbf{x}_{n+1}$ in the set of vertices for the next iteration, or (2) if a shrink is performed, a set of n new points that, together with $\mathbf{x}_1$, form the simplex at the next iteration.

One iteration of Algorithm NM (the Nelder–Mead algorithm).

1. **Order.** Order the $n+1$ vertices to satisfy $f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \cdots \leq f(\mathbf{x}_{n+1})$, using the tie-breaking rules given below.

2. **Reflect.** Compute the *reflection point* $\mathbf{x}_r$ from

$$\mathbf{x}_r = \bar{\mathbf{x}} + \rho(\bar{\mathbf{x}} - \mathbf{x}_{n+1}) = (1+\rho)\bar{\mathbf{x}} - \rho\mathbf{x}_{n+1}, \tag{2.4}$$

where $\bar{\mathbf{x}} = \sum_{i=1}^{n} \mathbf{x}_i/n$ is the centroid of the n best points (all vertices except for $\mathbf{x}_{n+1}$). Evaluate $f_r = f(\mathbf{x}_r)$.

If $f_1 \le f_r < f_n$, accept the reflected point $\mathbf{x}_r$ and terminate the iteration.

3. **Expand.** If $f_r < f_1$, calculate the *expansion point* $\mathbf{x}_e$,

$$\mathbf{x}_e = \bar{\mathbf{x}} + \chi(\mathbf{x}_r - \bar{\mathbf{x}}) = \bar{\mathbf{x}} + \rho\chi(\bar{\mathbf{x}} - \mathbf{x}_{n+1}) = (1 + \rho\chi)\bar{\mathbf{x}} - \rho\chi\mathbf{x}_{n+1}, \tag{2.5}$$

and evaluate $f_e = f(\mathbf{x}_e)$. If $f_e < f_r$, accept $\mathbf{x}_e$ and terminate the iteration; otherwise (if $f_e \ge f_r$), accept $\mathbf{x}_r$ and terminate the iteration.

4. **Contract.** If $f_r \ge f_n$,

perform a *contraction* between $\bar{\mathbf{x}}$ and the better of $\mathbf{x}_{n+1}$ and $\mathbf{x}_r$.

a. Outside. If $f_n \le f_r < f_{n+1}$ (i.e., $\mathbf{x}_r$ is strictly better than $\mathbf{x}_{n+1}$), perform an *outside contraction*: calculate

$$\mathbf{x}_c = \bar{\mathbf{x}} + \gamma(\mathbf{x}_r - \bar{\mathbf{x}}) = \bar{\mathbf{x}} + \gamma\rho(\bar{\mathbf{x}} - \mathbf{x}_{n+1}) = (1 + \rho\gamma)\bar{\mathbf{x}} - \rho\gamma\mathbf{x}_{n+1}, \tag{2.6}$$

and evaluate $f_c = f(\mathbf{x}_c)$. If $f_c \le f_r$, accept $\mathbf{x}_c$ and terminate the iteration; otherwise, go to step 5 (perform a shrink).

b. Inside. If $f_r \ge f_{n+1}$, perform an *inside contraction*: calculate

$$\mathbf{x}_{cc} = \bar{\mathbf{x}} - \gamma(\bar{\mathbf{x}} - \mathbf{x}_{n+1}) = (1 - \gamma)\bar{\mathbf{x}} + \gamma\mathbf{x}_{n+1}, \tag{2.7}$$

and evaluate $f_{cc} = f(\mathbf{x}_{cc})$. If $f_{cc} < f_{n+1}$, accept $\mathbf{x}_{cc}$ and terminate the iteration; otherwise, go to step 5 (perform a shrink).

5. **Perform a shrink step.** Evaluate f at the n points $\mathbf{v}_i = \mathbf{x}_1 + \sigma(\mathbf{x}_i - \mathbf{x}_1)$, $i = 2, \ldots, n+1$. The (unordered) vertices of the simplex at the next iteration consist of $\mathbf{x}_1$, $\mathbf{v}_2$, $\ldots$, $\mathbf{v}_{n+1}$.

Figures 1 and 2 show the effects of reflection, expansion, contraction, and shrinkage for a simplex in two dimensions (a triangle), using the standard coefficients $\rho = 1$, $\chi = 2$, $\gamma = \frac{1}{2}$, and $\sigma = \frac{1}{2}$. Observe that, except in a shrink, the one new vertex always lies on the (extended) line joining $\bar{x}$ and x_{n+1}. Furthermore, it is visually evident that the simplex shape undergoes a noticeable change during an expansion or contraction with the standard coefficients.

The Nelder–Mead paper [6] did not describe how to order points in the case of equal function values. We adopt the following tie-breaking rules, which assign to the new vertex the highest possible index consistent with the relation $f(\mathbf{x}_1^{(k+1)}) \le f(\mathbf{x}_2^{(k+1)}) \le \cdots \le f(\mathbf{x}_{n+1}^{(k+1)})$.

Nonshrink ordering rule. When a nonshrink step occurs, the worst vertex $\mathbf{x}_{n+1}^{(k)}$ is discarded. The accepted point created during iteration k, denoted by $\mathbf{v}^{(k)}$, becomes a new vertex and takes position $j+1$ in the vertices of Δ_{k+1}, where

$$j = \max_{0 \le \ell \le n} \{\, \ell \mid f(\mathbf{v}^{(k)}) < f(\mathbf{x}_{\ell+1}^{(k)}) \,\};$$

all other vertices retain their relative ordering from iteration k.

Shrink ordering rule. If a shrink step occurs, the only vertex carried over from Δ_k to Δ_{k+1} is $\mathbf{x}_1^{(k)}$. Only one tie-breaking rule is specified, for the case in which $\mathbf{x}_1^{(k)}$ and one or more of the new points are tied as the best point: if

$$\min\{f(\mathbf{v}_2^{(k)}), \ldots, f(\mathbf{v}_{n+1}^{(k)})\} = f(\mathbf{x}_1^{(k)}),$$

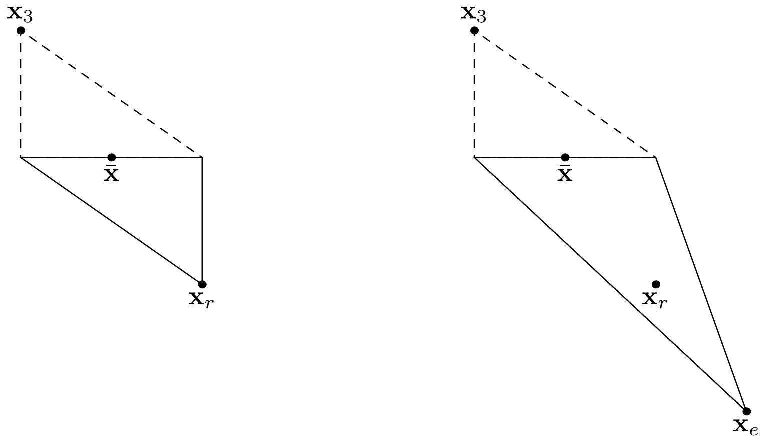


FIG. 1. *Nelder–Mead simplices after a reflection and an expansion step. The original simplex is shown with a dashed line.*

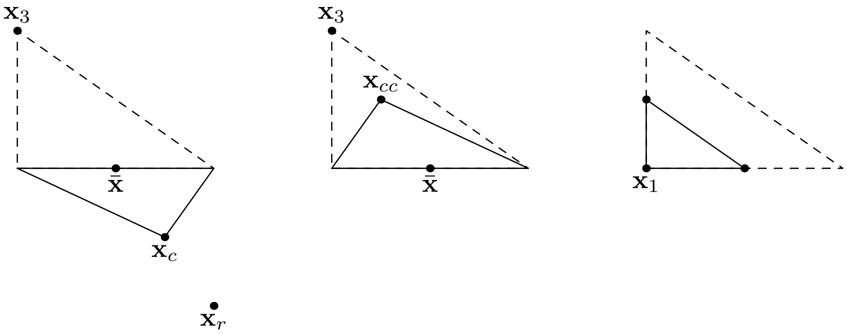


FIG. 2. *Nelder–Mead simplices after an outside contraction, an inside contraction, and a shrink. The original simplex is shown with a dashed line.*

then $\mathbf{x}_1^{(k+1)} = \mathbf{x}_1^{(k)}$. Beyond this, whatever rule is used to define the original ordering may be applied after a shrink.

We define the *change index* k^* of iteration k as the smallest index of a vertex that differs between iterations k and $k+1$:

$$k^* = \min\{\ i \ \mid\ \mathbf{x}_i^{(k)} \neq \mathbf{x}_i^{(k+1)}\ \}. \tag{2.8}$$

(Tie-breaking rules are needed to define a unique value of k^*.) When Algorithm NM terminates in step 2, $1 < k^* \le n$; with termination in step 3, $k^* = 1$; with termination in step 4, $1 \le k^* \le n+1$; and with termination in step 5, $k^* = 1$ or 2. A statement that "$\mathbf{x}_j$ changes" means that j is the change index at the relevant iteration.

The rules and definitions given so far imply that, for a nonshrink iteration,

$$
\begin{aligned}
&f_j^{(k+1)} = f_j^{(k)} \quad \text{and} \quad \mathbf{x}_j^{(k+1)} = \mathbf{x}_j^{(k)},\ j < k^*;\\
&f_{k^*}^{(k+1)} < f_{k^*}^{(k)} \quad \text{and} \quad \mathbf{x}_{k^*}^{(k+1)} \neq \mathbf{x}_{k^*}^{(k)};\\
&f_j^{(k+1)} = f_{j-1}^{(k)} \quad \text{and} \quad \mathbf{x}_j^{(k+1)} = \mathbf{x}_{j-1}^{(k)},\ j > k^*.
\end{aligned}
\tag{2.9}
$$

Thus the vector $(f_1^{(k)}, \ldots, f_{n+1}^{(k)})$ strictly lexicographically decreases at each nonshrink iteration.

For illustration, suppose that $n = 4$ and the vertex function values at a nonshrink iteration k are $(1, 2, 2, 3, 3)$. If $f(\mathbf{v}^{(k)}) = 2$, the function values at iteration $k + 1$ are $(1, 2, 2, 2, 3)$, $\mathbf{x}_4^{(k+1)} = \mathbf{v}^{(k)}$, and $k^* = 4$. This example shows that, following a single nonshrink iteration, the worst function value need not strictly decrease; however, the worst function value must strictly decrease after at most $n + 1$ consecutive nonshrink iterations.

2.2. Matrix notation. It is convenient to use matrix notation to describe Nelder–Mead iterations. The simplex Δ_k can be represented as an $n \times (n + 1)$ matrix whose columns are the vertices

$$\Delta_k = \begin{pmatrix} \mathbf{x}_1^{(k)} & \cdots & \mathbf{x}_{n+1}^{(k)} \end{pmatrix} = \begin{pmatrix} B_k & \mathbf{x}_{n+1}^{(k)} \end{pmatrix}, \quad \text{where} \quad B_k = \begin{pmatrix} \mathbf{x}_1^{(k)} & \cdots & \mathbf{x}_n^{(k)} \end{pmatrix}.$$

For any simplex Δ_k in $\mathcal{R}^n$, we define M_k as the $n \times n$ matrix whose jth column represents the "edge" of Δ_k between $\mathbf{x}_j^{(k)}$ and $\mathbf{x}_{n+1}^{(k)}$:

$$M_k \equiv \begin{pmatrix} \mathbf{x}_1^{(k)} - \mathbf{x}_{n+1}^{(k)} & \mathbf{x}_2^{(k)} - \mathbf{x}_{n+1}^{(k)} & \cdots & \mathbf{x}_n^{(k)} - \mathbf{x}_{n+1}^{(k)} \end{pmatrix} = B_k - \mathbf{x}_{n+1}^{(k)} \mathbf{e}^T, \tag{2.10}$$

where $\mathbf{e} = (1, 1, \ldots, 1)^T$. The n-dimensional volume of Δ_k is given by

$$\operatorname{vol}(\Delta_k) = \frac{|\det(M_k)|}{n!}. \tag{2.11}$$

A simplex Δ_k is *nondegenerate* if M_k is nonsingular or, equivalently, if $\operatorname{vol}(\Delta_k) > 0$. The volume of the simplex obviously depends only on the coordinates of the vertices, not on their ordering. For future reference, we define the diameter of Δ_k as

$$\operatorname{diam}(\Delta_k) = \max_{i \neq j} \|\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\|,$$

where $\|\cdot\|$ denotes the two-norm.

During a nonshrink iteration, the function is evaluated only at *trial points* of the form

$$\mathbf{z}^{(k)}(\tau) := \bar{\mathbf{x}}^{(k)} + \tau(\bar{\mathbf{x}}^{(k)} - \mathbf{x}_{n+1}^{(k)}) = (1 + \tau)\bar{\mathbf{x}}^{(k)} - \tau \mathbf{x}_{n+1}^{(k)}, \tag{2.12}$$

where the coefficient τ has one of four possible values:

$$
\begin{aligned}
&\tau = \rho \quad \text{(reflection)}; && \tau = \rho\chi \quad \text{(expansion)};\\
&\tau = \rho\gamma \quad \text{(outside contraction)}; && \tau = -\gamma \quad \text{(inside contraction)}.
\end{aligned}
\tag{2.13}
$$

In a nonshrink step, the single accepted point is one of the trial points, and we let τ_k denote the coefficient associated with the accepted point at iteration k. Thus the new vertex $\mathbf{v}^{(k)}$ produced during iteration k, which will replace $\mathbf{x}_{n+1}^{(k)}$, is given by $\mathbf{v}^{(k)} = \mathbf{z}^{(k)}(\tau_k)$. We sometimes call τ_k the *type* of move for a nonshrink iteration k.

During the kth Nelder–Mead iteration, (2.12) shows that each trial point (reflection, expansion, contraction) may be written as

$$\mathbf{z}^{(k)}(\tau) = \Delta_k \mathbf{t}(\tau), \quad \text{where} \quad \mathbf{t}(\tau) = \left(\frac{1+\tau}{n}, \ \dots, \ \frac{1+\tau}{n}, -\tau \right)^T. \tag{2.14}$$

Following the kth Nelder–Mead iteration, the (unordered) vertices of the next simplex are the columns of $\Delta_k S_k$, where S_k is an $(n+1)\times(n+1)$ matrix given by

$$\begin{pmatrix} I_n & \dfrac{(1+\tau_k)}{n}\mathbf{e} \\ \mathbf{0}^T & -\tau_k \end{pmatrix} \text{ for a step of type } \tau \text{ and by } \begin{pmatrix} 1 & (1-\sigma)\mathbf{e}^T \\ \mathbf{0} & \sigma I_n \end{pmatrix}$$

for a shrink step, with $\mathbf{0}$ being an n-dimensional zero column and I_n being the n-dimensional identity matrix. After being ordered at the start of iteration $k+1$, the vertices of Δ_{k+1} satisfy

$$\Delta_{k+1} = \Delta_k T_k, \quad \text{with} \quad T_k = S_k P_k, \tag{2.15}$$

where P_k is a permutation matrix chosen to enforce the ordering and tie-breaking rules (so that P_k depends on the function values at the vertices).

The updated simplex Δ_{k+1} has a disjoint interior from Δ_k for a reflection, an expansion, or an outside contraction, while $\Delta_{k+1} \subseteq \Delta_k$ for an inside contraction or a shrink.

By the *shape* of a nondegenerate simplex, we mean its equivalence class under similarity, i.e., Δ and $\lambda\Delta$ have the same shape when $\lambda > 0$. The shape of a simplex is determined by its angles, or equivalently by the singular values of the associated matrix M (2.10) after scaling so that Δ has unit volume. The Nelder–Mead method was deliberately designed with the idea that the simplex shapes would "adapt to the features of the local landscape" [6]. The Nelder–Mead moves apparently permit any simplex shape to be approximated—in particular, arbitrarily flat or needle-shaped simplices (as in the McKinnon examples [5]) are possible.

3. Properties of the Nelder–Mead algorithm.

This section establishes various basic properties of the Nelder–Mead method. Although there is a substantial level of folklore about the Nelder–Mead method, almost no proofs have appeared in print, so we include details here.

3.1. General results.

The following properties follow immediately from the definition of Algorithm NM.

1. A Nelder–Mead iteration requires one function evaluation when the iteration terminates in step 2, two function evaluations when termination occurs in step 3 or step 4, and $n+2$ function evaluations if a shrink step occurs.

2. The "reflect" step is so named because the reflection point $\mathbf{x}_r$ (2.4) is a (scaled) reflection of the worst point $\mathbf{x}_{n+1}$ around the point $\bar{\mathbf{x}}$ on the line through $\mathbf{x}_{n+1}$ and $\bar{\mathbf{x}}$. It is a genuine reflection on this line when $\rho = 1$, which is the standard choice for the reflection coefficient.

3. For general functions, a shrink step can conceivably lead to an increase in every vertex function value except f_1, i.e., it is possible that $f_i^{(k+1)} > f_i^{(k)}$ for $2 \le i \le n+1$. In addition, observe that with an outside contraction (case 4a), the algorithm takes a shrink step if $f(\mathbf{x}_c) > f(\mathbf{x}_r)$, even though a new point $\mathbf{x}_r$ has already been found that strictly improves over the worst vertex, since $f(\mathbf{x}_r) < f(\mathbf{x}_{n+1})$.

4. In the expand step, the method in the original Nelder–Mead paper accepts $\mathbf{x}_e$ if $f(\mathbf{x}_e) < f_1$ and accepts $\mathbf{x}_r$ otherwise. Standard practice today (which we follow) accepts the better of $\mathbf{x}_r$ and $\mathbf{x}_e$ if both give an improvement over $\mathbf{x}_1$. The proofs of Lemmas 4.6 and 5.2 depend on the rule that the expansion point is accepted only if it is strictly better than the reflection point.

It is commonly (and correctly) assumed that nondegeneracy of the initial simplex Δ_0 implies nondegeneracy of all subsequent Nelder–Mead simplices. We first give an informal indication of why this property holds. By construction, each trial point (2.12) in the Nelder–Mead method lies strictly outside the face defined by the n best vertices, along the line joining the worst vertex to the centroid of that face. If a nonshrink iteration occurs, the worst vertex is replaced by one of the trial points. If a shrink iteration occurs, each current vertex except the best is replaced by a point that lies a fraction of the step to the current best vertex. In either case it is clear from the geometry that the new simplex must be nondegenerate. For completeness, we present a proof of nondegeneracy based on a useful result about the volumes of successive simplices.

Lemma 3.1. (Volume and nondegeneracy of Nelder–Mead simplices.)

(1) *If the initial simplex Δ_0 is nondegenerate, so are all subsequent Nelder–Mead simplices.*

(2) *Following a nonshrink step of type τ,* $\operatorname{vol}(\Delta_{k+1}) = |\tau| \operatorname{vol}(\Delta_k)$.

(3) *Following a shrink step at iteration k,* $\operatorname{vol}(\Delta_{k+1}) = \sigma^n \operatorname{vol}(\Delta_k)$.

Proof. A simplex Δ is nondegenerate if it has nonzero volume. Result (1) will follow immediately from (2) and (3) because $\tau \neq 0$ (see (2.13)) and $\sigma \neq 0$.

When iteration k is a nonshrink, we assume without loss of generality that the worst point is the origin. In this case, it follows from (2.14) that the new vertex is

$$\mathbf{v}^{(k)} = M_k \mathbf{w}, \quad \text{where} \quad \mathbf{w} = \left(\frac{1+\tau}{n}, \cdots, \frac{1+\tau}{n} \right)^T, \tag{3.1}$$

so that the vertices of Δ_{k+1} consist of the vector $M_k\mathbf{w}$ and the columns of M_k. Since the volume of the new simplex does not depend on the ordering of the vertices, we may assume without affecting the volume that the new vertex is the worst. Applying the form of M in (2.10), we have

$$|\det(M_{k+1})| = |\det(M_k - M_k\mathbf{w}\mathbf{e}^T)| = |\det(M_k)|\,|\det(I - \mathbf{w}\mathbf{e}^T)|.$$

The matrix $I - \mathbf{w}\mathbf{e}^T$ has $n-1$ eigenvalues of unity and one eigenvalue equal to $1 - \mathbf{w}^T\mathbf{e} = -\tau$, so that $\det(I - \mathbf{w}\mathbf{e}^T) = -\tau$. Application of (2.11) gives result (2).

If iteration k is a shrink step, each edge of the simplex is multiplied by σ. Thus M_{k+1} is a permutation of σM_k and result (3) for a shrink follows from a standard property of determinants for $n \times n$ matrices. □

Lemma 3.1 shows that, in any dimension, a reflection step with $\rho = 1$ preserves volume. The choice $\rho = 1$ is natural geometrically, since a reflection step is then a genuine reflection. A reflected simplex with $\rho = 1$ is necessarily congruent to the original simplex for $n = 1$ and $n = 2$, but this is no longer true for $n \geq 3$.

Note that, although the Nelder–Mead simplices are nondegenerate in exact arithmetic, there is in general no upper bound on $\operatorname{cond}(M_k)$. In fact, the algorithm permits $\operatorname{cond}(M_k)$ to become arbitrarily large, as it does in the McKinnon example [5].

Our next result involves affine-invariance of the Nelder–Mead method when both the simplex and function are transformed appropriately.

LEMMA 3.2. (Affine-invariance.) *The Nelder–Mead method is invariant under affine motions of $\mathcal{R}^n$, i.e., under a change of variables $\phi(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ in which A is invertible, in the following sense: when minimizing $f(\mathbf{x})$ starting with simplex Δ_0, the complete sequence of Nelder–Mead steps and function values is the same as when minimizing the function $\tilde{f}(\mathbf{z}) = f(\phi(\mathbf{z}))$ with initial simplex $\tilde{\Delta}_0$ defined by*

$$\tilde{\Delta}_0 = \phi^{-1}(\Delta_0) = A^{-1}(\Delta_0) - A^{-1}\mathbf{b}.$$

Proof. At the vertices of $\tilde{\Delta}_0$, $\tilde{f}(\tilde{\mathbf{x}}_i^{(0)}) = f(\mathbf{x}_i^{(0)})$. We proceed by induction, assuming for simplicity that $\mathbf{b} = 0$. If $\tilde{\Delta}_k = A^{-1}\Delta_k$ and $\tilde{f}(\tilde{\mathbf{x}}_i^{(k)}) = f(\mathbf{x}_i^{(k)})$ for $1 \le i \le n+1$, then relation (2.14) shows that the trial points generated from $\tilde{\Delta}_k$ satisfy $\tilde{\mathbf{z}}(\tau) = A^{-1}\mathbf{z}(\tau)$, which means that $\tilde{f}(\tilde{\mathbf{z}}(\tau)) = f(\mathbf{z}(\tau))$. The matrix T_k of (2.15) will therefore be the same for both Δ_k and $\tilde{\Delta}_k$, so that $\tilde{\Delta}_{k+1} = A^{-1}\Delta_{k+1}$. It follows that $\tilde{f}(\tilde{\mathbf{x}}_i^{(k+1)}) = f(\mathbf{x}_i^{(k+1)})$ for $1 \le i \le n+1$, which completes the induction. A similar argument applies when $\mathbf{b} \ne 0$. □

Using Lemma 3.2, we can reduce the study of the Nelder–Mead algorithm for a general strictly convex quadratic function on $\mathcal{R}^n$ to the study of $f(\mathbf{x}) = \|\mathbf{x}\|^2 = x_1^2 + \cdots + x_n^2$.

The next lemma summarizes several straightforward results.

LEMMA 3.3. *Let f be a function that is bounded below on $\mathcal{R}^n$. When the Nelder–Mead algorithm is applied to minimize f, starting with a nondegenerate simplex Δ_0, then*

(1) *the sequence $\{f_1^{(k)}\}$ always converges;*

(2) *at every nonshrink iteration k, $f_i^{(k+1)} \le f_i^{(k)}$ for $1 \le i \le n+1$, with strict inequality for at least one value of i;*

(3) *if there are only a finite number of shrink iterations, then*

 (i) *each sequence $\{f_i^{(k)}\}$ converges as $k \to \infty$ for $1 \le i \le n+1$,*

 (ii) *$f_i^* \le f_i^{(k)}$ for $1 \le i \le n+1$ and all k, where $f_i^* = \lim_{k\to\infty} f_i^{(k)}$,*

 (iii) *$f_1^* \le f_2^* \le \cdots \le f_{n+1}^*$;*

(4) *if there are only a finite number of nonshrink iterations, then all simplex vertices converge to a single point.* □

We now analyze the Nelder–Mead algorithm in the case when *only nonshrink steps occur.* Torczon [12] observes that shrink steps essentially never happen in practice (she reports only 33 shrink steps in 2.9 million Nelder–Mead iterations on a set of general test problems), and the rarity of shrink steps is confirmed by our own numerical experiments. We show in Lemma 3.5 that no shrink steps are taken when the Nelder–Mead method is applied to a strictly convex function. All of our results that assume no shrink steps can obviously be applied to cases when only a finite number of shrink steps occur.

Assuming that there are no shrink steps, the next lemma gives an important property of the $n+1$ limiting vertex function values whose existence is verified in part (3) of Lemma 3.3.

LEMMA 3.4. (Broken convergence.) *Suppose that the function f is bounded below on $\mathcal{R}^n$, that the Nelder–Mead algorithm is applied to f beginning with a nondegenerate initial simplex Δ_0, and that no shrink steps occur. If there is an integer j, $1 \le j \le n$, for which*

$$f_j^* < f_{j+1}^*, \quad \textit{where} \quad f_j^* = \lim_{k\to\infty} f_j^{(k)}, \tag{3.2}$$

then there is an iteration index K such that for all $k \geq K$, the change index satisfies

$$k^* > j, \tag{3.3}$$

i.e., the first j vertices of all simplices remain fixed *after iteration K. (We refer to property* (3.2) *as* broken convergence *for vertex j.)*

Proof. The lemma is proved by contradiction. By hypothesis (3.2), $f_j^* + \delta = f_{j+1}^*$ for some $\delta > 0$. Pick $\epsilon > 0$ such that $\delta - \epsilon > 0$. Since $f_j^* = \lim_{k\to\infty} f_j^{(k)}$, there exists K such that for all $k \geq K$, $f_j^{(k)} - \epsilon \leq f_j^*$. Then, for all $k \geq K$,

$$f_j^{(k)} < f_j^{(k)} - \epsilon + \delta \leq f_j^* + \delta = f_{j+1}^*.$$

But, from Lemma 3.3, part (3), for any index ℓ, $f_{j+1}^* \leq f_{j+1}^{(\ell)}$. Therefore, for all $k \geq K$ and any ℓ,

$$f_j^{(k)} < f_{j+1}^{(\ell)}. \tag{3.4}$$

But if $k^* \leq j$ for any $k \geq K$, then, using the third relation in (2.9), it must be true that $f_{j+1}^{(k+1)} = f_j^{(k)}$, which contradicts (3.4). Thus $k^* > j$ for all $k \geq K$. □

The following corollary is an immediate consequence of Lemma 3.4.

COROLLARY 3.1. *Assume that f is bounded below on $\mathcal{R}^n$, the Nelder–Mead algorithm is applied beginning with a nondegenerate initial simplex Δ_0, and no shrink steps occur. If the change index is* 1 *infinitely often, i.e., the best point changes infinitely many times, then $f_1^* = \cdots = f_{n+1}^*$.* □

3.2. Results for strictly convex functions.

Without further assumptions, very little more can be said about the Nelder–Mead algorithm, and we henceforth assume that f is strictly convex.

DEFINITION 3.1. (Strict convexity.) *The function f is* strictly convex *on $\mathcal{R}^n$ if, for every pair of points $\mathbf{y}$, $\mathbf{z}$ with $\mathbf{y} \neq \mathbf{z}$ and every λ satisfying $0 < \lambda < 1$,*

$$f(\lambda\mathbf{y} + (1-\lambda)\mathbf{z}) \ < \ \lambda f(\mathbf{y}) + (1-\lambda)f(\mathbf{z}). \tag{3.5}$$

When f is strictly convex on $\mathcal{R}^n$ and

$$\mathbf{c} = \sum_{i=1}^{\ell} \lambda_i \mathbf{z}_i, \qquad \text{with} \quad 0 < \lambda_i < 1 \ \text{ and } \ \sum_{i=1}^{\ell} \lambda_i = 1,$$

$$\text{then } \ f(\mathbf{c}) < \sum_{i=1}^{\ell} \lambda_i f(\mathbf{z}_i) \qquad \text{and hence } \ f(\mathbf{c}) < \max\{f(\mathbf{z}_1), \ldots, f(\mathbf{z}_\ell)\}. \tag{3.6}$$

We now use this property to show that, when the Nelder–Mead method is applied to a strictly convex function, shrink steps cannot occur. (This result is mentioned without proof in [12].)

LEMMA 3.5. *Assume that f is strictly convex on $\mathcal{R}^n$ and that the Nelder–Mead algorithm is applied to f beginning with a nondegenerate initial simplex Δ_0. Then no shrink steps will be taken.*

Proof. Shrink steps can occur only if the algorithm reaches step 4 of Algorithm NM and fails to accept the relevant contraction point. When $n = 1$, $f(\bar{\mathbf{x}}) = f_n$. When $n > 1$, application of (3.6) to $\mathbf{x}_1, \ldots, \mathbf{x}_n$ shows that $f(\bar{\mathbf{x}}) < f_n$.

Consider an outside contraction, which is tried if $f_n \leq f_r < f_{n+1}$. Since the contraction coefficient γ satisfies $0 < \gamma < 1$, $\mathbf{x}_c$ as defined by (2.6) is a convex combination of $\bar{\mathbf{x}}$ and the reflection point $\mathbf{x}_r$. Thus, by (3.6),

$$f(\mathbf{x}_c) < \max\{f(\bar{\mathbf{x}}), f_r\}.$$

We know that $f(\bar{\mathbf{x}}) \leq f_n$ and $f_n \leq f_r$, so that $\max\{f(\bar{\mathbf{x}}), f_r\} = f_r$. Hence $f(\mathbf{x}_c) < f_r$, $\mathbf{x}_c$ will be accepted, and a shrink step will not be taken.

A similar argument applies for an inside contraction, since $f_{n+1} \leq f_r$ and $\mathbf{x}_{cc}$ is a convex combination of $\bar{\mathbf{x}}$ and $\mathbf{x}_{n+1}$. □

Note that simple convexity of f (for example, f constant) is not sufficient for this result, which depends in the case of an inside contraction on the fact that $f(\mathbf{x}_{cc})$ is *strictly* less than $f(\mathbf{x}_{n+1})$.

By combining the definition of a Nelder–Mead iteration, Lemma 3.4, and a mild further restriction on the reflection and contraction coefficients, we next prove that the limiting worst and next-worst function values are the same. (For $n = 1$, the result holds without the additional restriction; see Lemma 4.4).

LEMMA 3.6. *Assume that f is strictly convex on $\mathcal{R}^n$ and bounded below. If, in addition to the properties $\rho > 0$ and $0 < \gamma < 1$, the reflection coefficient ρ and the contraction coefficient γ satisfy $\rho\gamma < 1$, then*

(1) *$f_n^* = f_{n+1}^*$; and*

(2) *there are infinitely many iterations for which $\mathbf{x}_n^{(k+1)} \neq \mathbf{x}_n^{(k)}$.*

Proof. The proof is by contradiction. Assume that $f_n^* < f_{n+1}^*$. From Lemma 3.4, this means that there exists an iteration index K such that the change index $k^* = n+1$ for $k \geq K$. Without loss of generality, we may take $K = 0$. Since $k^* = n + 1$ for all $k \geq 0$, the best n vertices, which must be distinct, remain constant for all iterations; thus the centroid $\bar{\mathbf{x}}^{(k)} = \bar{\mathbf{x}}$, a constant vector, and $f(\mathbf{x}_n)$ is equal to its limiting value f_n^*. Because f is strictly convex, $f(\bar{\mathbf{x}}) \leq f(\mathbf{x}_n) = f_n^*$. (This inequality is strict if $n > 1$.)

The change index will be $n + 1$ at every iteration only if a contraction point is accepted and becomes the new worst point. Therefore, the vertex $\mathbf{x}_{n+1}^{(k+1)}$ satisfies one of the recurrences

$$\mathbf{x}_{n+1}^{(k+1)} = (1 + \rho\gamma)\bar{\mathbf{x}} - \rho\gamma\mathbf{x}_{n+1}^{(k)} \quad \text{or} \quad \mathbf{x}_{n+1}^{(k+1)} = (1 - \gamma)\bar{\mathbf{x}} + \gamma\mathbf{x}_{n+1}^{(k)}. \tag{3.7}$$

The homogeneous forms of these equations are

$$\mathbf{y}_{n+1}^{(k+1)} = -\rho\gamma\mathbf{y}_{n+1}^{(k)} \quad \text{or} \quad \mathbf{y}_{n+1}^{(k+1)} = \gamma\mathbf{y}_{n+1}^{(k)}. \tag{3.8}$$

Since $0 < \gamma < 1$ and $0 < \rho\gamma < 1$, we have $\lim_{k\to\infty} \mathbf{y}_{n+1}^{(k)} = \mathbf{0}$, so that the solutions of both equations in (3.8) are zero as $k \to \infty$.

Now we need only to find a particular solution to the inhomogeneous forms of (3.7). Both are satisfied by the constant vector $\bar{\mathbf{x}}$, so that their general solutions are given by $\mathbf{x}_{n+1}^{(k)} = \mathbf{y}_{n+1}^{(k)} + \bar{\mathbf{x}}$, where $\mathbf{y}_{n+1}^{(k)}$ satisfies one of the relations (3.8). Since $\lim_{k\to\infty} \mathbf{y}_{n+1}^{(k)} = \mathbf{0}$, it follows that

$$\lim_{k\to\infty} \mathbf{x}_{n+1}^{(k)} = \mathbf{x}_{n+1}^* = \bar{\mathbf{x}}, \qquad \text{with } f_{n+1}^* = f(\bar{\mathbf{x}}).$$

But we know from the beginning of the proof that $f(\bar{\mathbf{x}}) \leq f_n^*$, which means that $f_{n+1}^* \leq f_n^*$. Lemma 3.3, part (3), shows that this can be true only if $f_n^* = f_{n+1}^*$, which gives part (1).

The result of part (2) is immediate because we have already shown a contradiction if there exists K such that $\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_n^{(k)}$ remain constant for $k \geq K$. □

In analyzing convergence, we know from Lemma 3.4 that, if broken convergence occurs, there exists an index j such that all vertices $\{\mathbf{x}_i^{(k)}\}$, for $1 \leq i \leq j$, remain constant from some point on. If this happens, the best point $\mathbf{x}_1^{(k)}$ will not be changed, and hence expansion steps cannot occur. (Nor can reflection steps in which a strict improvement is found over f_1.) For this reason, it is interesting to consider a *restricted* Nelder–Mead algorithm in which expansion steps are not taken; the analysis of the restricted algorithm is simpler because both $\mathrm{vol}(\Delta_k)$ and $\mathrm{diam}(\Delta_k)$ are nonincreasing if $\rho \leq 1$. We do not discuss the restricted algorithm further in this paper, but see [3].

In the remainder of this paper we consider strictly convex functions f with bounded level sets. The *level set* $\Gamma_\mu(f)$ is defined as

$$\Gamma_\mu(f) = \{\, \mathbf{x} : f(\mathbf{x}) \leq \mu \,\}. \tag{3.9}$$

A function f has *bounded level sets* if $\Gamma_\mu(f)$ is bounded for every μ; this restriction excludes strictly convex functions like e^{-x}. The point of this restriction is that a strictly convex function with bounded level sets has a unique minimizer $\mathbf{x}_{\min}$.

4. Nelder–Mead in dimension 1 for strictly convex functions.

We analyze the Nelder–Mead algorithm in dimension 1 on strictly convex functions with bounded level sets. The behavior of the Nelder–Mead algorithm in dimension 1 depends nontrivially on the values of the reflection coefficient ρ, the expansion coefficient χ, and the contraction coefficient γ. (The shrink coefficient σ is irrelevant because shrink steps cannot occur for a strictly convex function; see Lemma 3.5.) We show that convergence to $x_{\min}$ always occurs as long as $\rho\chi \geq 1$ (Theorem 4.1) and that convergence is M-step linear when $\rho = 1$ (Theorem 4.2). The algorithm does not always converge to the minimizer $x_{\min}$ if $\rho\chi < 1$. An interesting feature of the analysis is that M-step linear convergence can be guaranteed even though infinitely many expansion steps may occur.

4.1. Special properties in one dimension.

In one dimension, the "next-worst" and the "best" vertices are the same point, which means that the centroid $\bar{x}^{(k)}$ is equal to $x_1^{(k)}$ at every iteration. A Nelder–Mead simplex is a line segment, so that, given iteration k of type τ_k,

$$\mathrm{diam}(\Delta_{k+1}) = |\tau_k| \, \mathrm{diam}(\Delta_k). \tag{4.1}$$

Thus, in the special case of the standard parameters $\rho = 1$ and $\chi = 2$, a reflection step retains the same diameter and an expansion step doubles the diameter of the simplex. To deal with different orderings of the endpoints, we use the notation $\mathrm{int}(y, z)$ to denote the open interval with endpoints y and z (even if $y > z$), with analogous notation for closed or semiopen intervals.

The following lemma summarizes three important properties, to be used repeatedly, of strictly convex functions in $\mathcal{R}^1$ with bounded level sets.

LEMMA 4.1. *Let f be a strictly convex function on $\mathcal{R}^1$ with a unique minimizer $x_{\min}$.*

(1) *Let y_1, y_2, and y_3 be three distinct points such that $y_2 \in \mathrm{int}(y_1, y_3)$. Then*

$$f(y_1) \geq f(y_2) \quad \textit{and} \quad f(y_2) \leq f(y_3) \implies x_{\min} \in \mathrm{int}(y_1, y_3).$$

(2) *If* $x_{\min} \in \operatorname{int}[y_1, y_2]$*, then* $f(y_2 + \xi_2(y_1 - y_2)) > f(y_2 + \xi_1(y_1 - y_2))$ *if* $\xi_2 > \xi_1 \geq 1$*.*

(3) f *is continuous.* □

A special property of the one-dimensional case is that a Nelder–Mead iteration can never terminate in step 2 of Algorithm NM (see section 2): either a contraction will be taken (step 4), or an expansion step will be tried (step 3). Using the rule in step 3 that we must accept the better of the reflection and expansion points, a reflection step will be taken only if $f_r < f_1$ and $f_e \geq f_r$.

4.2. Convergence to the minimizer. We first consider general Nelder–Mead parameters satisfying (2.1) and show that the condition $\rho\chi \geq 1$ is *necessary* for the global convergence of the algorithm to $x_{\min}$. If $\rho\chi < 1$, the so-called "expand" step actually *reduces* the simplex diameter, and the endpoints of the Nelder–Mead interval can move a distance of at most $\operatorname{diam}(\Delta_0)/(1 - \rho\chi)$ from the initial vertex $x_1^{(0)}$. Thus convergence to $x_{\min}$ will not occur whenever

$$\rho\chi < 1 \quad \text{and} \quad |x_{\min} - x_1^{(0)}| > \operatorname{diam}(\Delta_0)/(1 - \rho\chi).$$

We next show the general result that the condition $\rho\chi \geq 1$, combined with the requirements (2.1), is sufficient for global convergence to $x_{\min}$ of the Nelder–Mead algorithm in one dimension.

THEOREM 4.1. (Convergence of one-dimensional Nelder–Mead method.) *Let* f *be a strictly convex function on* $\mathcal{R}^1$ *with bounded level sets. Assume that the Nelder–Mead algorithm is applied to* f *with parameters satisfying* $\rho > 0$*,* $\chi > 1$*,* $\chi > \rho$*,* $\rho\chi \geq 1$*, and* $0 < \gamma < 1$*, beginning with a nondegenerate initial simplex* Δ_0*. Then both endpoints of the Nelder–Mead interval converge to* $x_{\min}$*.*

The proof of this theorem depends on several intermediate lemmas. First we show that the Nelder–Mead algorithm finds, within a finite number of iterations, an "interval of uncertainty" in which the minimizer must lie.

LEMMA 4.2. (Bracketing of $x_{\min}$.) *Let* f *be a strictly convex function on* $\mathcal{R}^1$ *with bounded level sets. Assume that the Nelder–Mead algorithm is applied to* f *beginning with a nondegenerate initial simplex* Δ_0 *and that the reflection and expansion coefficients satisfy* $\rho > 0$*,* $\chi > 1$*,* $\chi > \rho$*, and* $\rho\chi \geq 1$*. Then there is a smallest integer* K *satisfying*

$$K \leq \frac{|x_{\min} - x_1^{(0)}|}{\operatorname{diam}(\Delta_0)}, \quad \textit{such that} \quad f_2^{(K)} \geq f_1^{(K)} \textit{ and } f_1^{(K)} \leq f_e^{(K)}. \tag{4.2}$$

In this case, $x_{\min} \in \operatorname{int}(x_2^{(K)}, x_e^{(K)})$ *and we say that* $x_{\min}$ *is* bracketed *by* $x_2^{(K)}$ *and* $x_e^{(K)}$*.*

Proof. To reduce clutter, we drop the superscript k and use a prime to denote quantities associated with iteration $k + 1$. By definition, $f_2 \geq f_1$, so that the first inequality in the "up–down–up" relation involving f in (4.2) holds automatically for every Nelder–Mead interval. There are two possibilities.

(i) If $f_1 \leq f_e$, the "up–down–up" pattern of f from (4.2) holds at the current iteration.

(ii) If $f_1 > f_e$, we know from strict convexity that $f_r < f_1$, and the expansion point is accepted. At the next iteration, $x'_2 = x_1$ and $x'_1 = x_e$. There are two cases to consider.

First, suppose that $x_{\min}$ lies in $\operatorname{int}(x_2', x_1'] = \operatorname{int}(x_1, x_e]$. Using result (2) of Lemma 4.1, both $f(x_r')$ and $f(x_e')$ must be strictly larger than $f(x_1')$. Hence the "up–down–up" pattern of (4.2) holds at the next iteration.

Alternatively, suppose that $x_{\min}$ lies "beyond" x_e, i.e., beyond x_1'. Then

$$|x_{\min} - x_1'| = |x_{\min} - x_1| - \operatorname{diam}(\Delta').$$

It follows from (4.1) and the inequality $\rho\chi \geq 1$ that $\operatorname{diam}(\Delta') = \rho\chi \operatorname{diam}(\Delta) \geq \operatorname{diam}(\Delta)$. Thus the distance from $x_{\min}$ to the current best point is reduced by an amount bounded below by Δ_0, the diameter of the initial interval. This gives the upper bound on K of (4.2). □

The next result shows that, once $x_{\min}$ lies in a specified interval defined by the current Nelder–Mead interval and a number depending only on the reflection, expansion, and contraction coefficients, it lies in an analogous interval at all subsequent iterations.

LEMMA 4.3. *Let f be a strictly convex function on $\mathcal{R}^1$ with bounded level sets. Assume that the Nelder–Mead algorithm with parameters satisfying $\rho > 0$, $\chi > 1$, $\chi > \rho$, $\rho\chi \geq 1$, and $0 < \gamma < 1$, is applied to f beginning with a nondegenerate initial simplex. We define N_{NM} as*

$$N_{NM} = \max\Big(\frac{1}{\rho\gamma},\ \frac{\rho}{\gamma},\ \rho\chi,\ \chi - 1\Big), \tag{4.3}$$

and we say that the proximity property *holds at iteration k if*

$$x_{\min} \in \operatorname{int}\Big(x_2^{(k)},\ \ x_1^{(k)} + N_{NM}(x_1^{(k)} - x_2^{(k)})\Big]. \tag{4.4}$$

Then, if the proximity property holds at iteration k, it holds at iteration $k+1$.

Proof. To reduce clutter, we omit the index k and use a prime to denote quantities associated with iteration $k+1$. The proof considers all possible cases for location of $x_{\min}$ in the interval defined by (4.4). We have either $x_2 < x_1 < x_r < x_e$ or $x_e < x_r < x_1 < x_2$.

Case 1. $x_{\min} \in \operatorname{int}(x_2, x_1]$.

Lemma 4.1, part (2), implies that $f_r > f_1$, which means that a contraction step will be taken.

1a. If $f_r \geq f_2$, an inside contraction will occur, with $x_{cc} = x_1 - \gamma(x_1 - x_2)$. Strict convexity implies that $f_{cc} < f_2$.

(i) If $f_{cc} \geq f_1$, $x_{\min}$ lies in $\operatorname{int}(x_{cc}, x_1]$. The next Nelder–Mead interval is given by $x_2' = x_{cc}$ and $x_1' = x_1$, which means that $x_{\min} \in \operatorname{int}(x_2', x_1']$, and the proximity property holds at the next iteration.

(ii) If $f_{cc} < f_1$, the next Nelder–Mead interval is $x_2' = x_1$ and $x_1' = x_{cc}$. We also know that $x_{\min} \neq x_1$, so that $x_{\min} \in \operatorname{int}(x_2, x_1) = \operatorname{int}(x_2, x_2')$. To check whether (4.4) holds, we express x_2 in terms of the new Nelder–Mead interval as $x_2 = x_1' + \xi(x_1' - x_2')$. Using the definition of x_{cc} gives

$$x_2 = x_{cc} + \xi(x_{cc} - x_1) = x_1 + \gamma(x_2 - x_1) + \xi\gamma(x_2 - x_1), \quad \text{so that} \quad \xi = 1/\gamma - 1.$$

For $\rho > 1$, we have $1/\gamma - 1 < \rho/\gamma \leq N_{NM}$, while for $0 < \rho \leq 1$ we have $1/\gamma - 1 < 1/(\rho\gamma) \leq N_{NM}$, so that the proximity property (4.4) holds at the next iteration.

1b. If $f_r < f_2$, an outside contraction will occur, with $x_c = x_1 + \rho\gamma(x_1 - x_2)$. Since $x_{\min} \in \operatorname{int}(x_2, x_1]$, part (2) of Lemma 4.1 implies that $f_c > f_1$. The new Nelder–Mead interval is given by $x_2' = x_c$ and $x_1' = x_1$, and the interval of uncertainty remains $\operatorname{int}(x_2, x_1']$. Expressing x_2 as $x_1' + \xi(x_1' - x_2')$ gives

$$x_2 = x_1 + \xi(x_1 - x_c) = x_1 - \xi\rho\gamma(x_1 - x_2), \quad \text{so that} \quad \xi = 1/\rho\gamma \le N_{\scriptscriptstyle NM},$$

and (4.4) holds at the next iteration.

Case 2. $x_{\min} \in \operatorname{int}(x_1, x_r]$.

2a. If $f_r < f_1$, we try the expansion step x_e. Part (2) of Lemma 4.1 implies that $f_e > f_r$, which means that the reflection step is accepted, and the new Nelder–Mead interval is $x_2' = x_1$ and $x_1' = x_r$. Then $x_{\min} \in \operatorname{int}(x_2', x_1']$, and (4.4) holds at the next iteration.

2b. If $f_r \ge f_2$, an inside contraction will be taken, $x_{cc} = x_1 - \gamma(x_1 - x_2)$. We also know that $x_{\min} \ne x_r$, so that $x_{\min} \in \operatorname{int}(x_1, x_r)$. Part (2) of Lemma 4.1 implies that $f_{cc} > f_1$, and the next Nelder–Mead interval is $x_2' = x_{cc}$ and $x_1' = x_1$, with $x_{\min} \in \operatorname{int}(x_1', x_r)$. We express x_r as $x_1' + \xi(x_1' - x_2')$, which gives

$$x_r = x_1 + \rho(x_1 - x_2) = x_1 + \xi(x_1 - x_{cc}) = x_1 + \xi\gamma(x_1 - x_2), \quad \text{so that} \quad \xi = \rho/\gamma \le N_{\scriptscriptstyle NM},$$

and (4.4) holds at the next iteration.

2c. If $f_r \ge f_1$ and $f_r < f_2$, an outside contraction will be taken, $x_c = x_1 + \rho\gamma(x_1 - x_2)$. We also know that $x_{\min} \ne x_r$, so that $x_{\min} \in \operatorname{int}(x_1, x_r)$.

(i) If $f_c > f_1$, the new Nelder–Mead interval is $x_2' = x_c$ and $x_1' = x_1$. Because $f_c > f_1$, $x_{\min} \in \operatorname{int}(x_1, x_c) = \operatorname{int}(x_2', x_1')$, and (4.4) holds at the next iteration.

(ii) If $f_c < f_1$, the new Nelder–Mead interval is $x_2' = x_1$ and $x_1' = x_c$, and $x_{\min} \ne x_1$. The interval of uncertainty remains $\operatorname{int}(x_1, x_r) = \operatorname{int}(x_2', x_r)$. We thus write x_r as $x_1' + \xi(x_1' - x_2')$:

$$x_r = x_c + \xi(x_c - x_1) = x_1 + \rho\gamma(x_1 - x_2) + \xi\rho\gamma(x_1 - x_2), \quad \text{so that} \quad \xi = 1/\gamma - 1 < N_{\scriptscriptstyle NM},$$

and (4.4) holds at the next iteration.

Case 3. $x_{\min} \in \operatorname{int}(x_r, x_e]$.

3a. If $f_e \ge f_r$, the new Nelder–Mead interval is $x_2' = x_1$ and $x_1' = x_r$; furthermore, $x_{\min} \ne x_e$ and $x_{\min} \in \operatorname{int}(x_1', x_e)$. Expressing x_e as $x_1' + \xi(x_1' - x_2')$ gives

$$x_e = x_1 + \rho\chi(x_1 - x_2) = x_1 + \rho(x_1 - x_2) + \xi\rho(x_1 - x_2), \quad \text{so that} \quad \xi = \chi - 1.$$

Since $\xi \le N_{\scriptscriptstyle NM}$, (4.4) holds at the next iteration.

3b. If $f_e < f_r$, we accept x_e. The new Nelder–Mead interval is $x_2' = x_1$ and $x_1' = x_e$. Since x_r lies between x_1 and x_e, $x_{\min} \in \operatorname{int}(x_2', x_1')$ and (4.4) holds at the next iteration.

Case 4. $x_{\min} \in \operatorname{int}(x_e,\ x_1 + N_{\scriptscriptstyle NM}(x_1 - x_2)]$.

Case 4 can happen only if $N_{\scriptscriptstyle NM} > \rho\chi$, since $x_e = x_1 + \rho\chi(x_1 - x_2)$. Thus it must be true that $f_1 > f_r > f_e$, and the expansion point will be accepted. The new Nelder–Mead interval is defined by $x_2' = x_1$ and $x_1' = x_e$. Writing $x_1 + N_{\scriptscriptstyle NM}(x_1 - x_2)$ as $x_e + \xi(x_e - x_1)$ gives

$$x_1 + N_{\scriptscriptstyle NM}(x_1 - x_2) = x_1 + \rho\chi(x_1 - x_2) + \xi\rho\chi(x_1 - x_2), \text{ so that } \xi = (N_{\scriptscriptstyle NM} - \rho\chi)/\rho\chi.$$

Since $\rho\chi \ge 1$, $\xi < N_{\scriptscriptstyle NM}$ and the proximity property holds at the next iteration.

Cases 1–4 are exhaustive, and the lemma is proved. □

We prove that the Nelder–Mead simplex diameter converges to zero by first showing that the result of Lemma 3.6 holds, i.e., the function values at the interval endpoints converge to the same value, even when $\rho\gamma \ge 1$.

LEMMA 4.4. *Let f be a strictly convex function on $\mathcal{R}^1$ with bounded level sets. Assume that the Nelder–Mead algorithm with parameters satisfying $\rho > 0$ and $0 < \gamma < 1$ is applied to f beginning with a nondegenerate initial simplex. Then $f_1^* = f_2^*$.*

Proof. If $\rho\gamma < 1$, the result follows from Lemma 3.6. Hence we assume that $\rho\gamma \geq 1$, which means that $\rho > 1$. The proof is by contradiction, beginning as in the proof of Lemma 3.6. If $f_1^* < f_2^*$, there is an iteration index K such that, for $k \geq K$, every iteration k is a contraction and x_1 does not change. (Without loss of generality, we may take $K = 0$.)

If iteration k is an inside contraction, $\operatorname{diam}(\Delta_{k+1}) = \gamma \operatorname{diam}(\Delta_k) < \operatorname{diam}(\Delta_k)$. If iteration k is an outside contraction, $\operatorname{diam}(\Delta_{k+1}) = \rho\gamma \operatorname{diam}(\Delta_k) \geq \operatorname{diam}(\Delta_k)$. Thus $\lim_{k\to\infty} \operatorname{diam}(\Delta_k) \to 0$ if there are a finite number of outside contractions, and so we need to consider only the case of an infinite number of outside contractions.

Suppose that iteration k is an outside contraction. Then $f_r^{(k)} \geq f_1^{(k)}$, $f_r^{(k)} < f_2^{(k)}$, and the contraction point is $x_c^{(k)} = x_1^{(k)} + \rho\gamma(x_1^{(k)} - x_2^{(k)})$. Since the best point does not change, $f_c^{(k)} \geq f_1^{(k)}$ and $x_2^{(k+1)} = x_c^{(k)}$. By strict convexity, $f_c^{(k)} < f_r^{(k)}$.

Define $z(\xi)$ as

$$z(\xi) \equiv x_1^{(k)} + \xi\big(x_1^{(k)} - x_2^{(k)}\big),$$

so that $x_2^{(k)} = z(-1)$ and $x_r^{(k)} = z(\rho)$. Expressing $f_2^{(k)}$, $f_1^{(k)}$, and $f_c^{(k)}$ in this form, we have

$$f\big(z(-1)\big) \;>\; f\big(z(0)\big) \;\leq\; f\big(z(\rho\gamma)\big) = f_2^{(k+1)}, \tag{4.5}$$

so that $x_{\min} \in \operatorname{int}\big(\, z(-1),\, z(\rho\gamma)\,\big)$. The relation $f(z(-1)) = f_2^{(k)} > f_2^{(k+1)}$ and result (2) of Lemma 4.1 then imply that

$$f(z(\xi)) > f_2^{(k+1)} \quad \text{if} \quad \xi \leq -1. \tag{4.6}$$

The next reflection point $x_r^{(k+1)}$ is given by

$$x_r^{(k+1)} = x_1^{(k)} + \rho(x_1^{(k)} - x_2^{(k+1)}) = x_1^{(k)} - \rho^2\gamma(x_1^{(k)} - x_2^{(k)}) = z(-\rho^2\gamma).$$

Since $\rho\gamma \geq 1$ and $\rho > 1$, we have $\rho^2\gamma > 1$, and we conclude from (4.6) that $f_r^{(k+1)}$ strictly exceeds $f_2^{(k+1)}$. Iteration $k+1$ must therefore be an *inside* contraction, with

$$x_{cc}^{(k+1)} = x_1^{(k+1)} + \gamma(x_1^{(k+1)} - x_2^{(k+1)}) = x_1^{(k)} + \rho\gamma^2(x_1^{(k)} - x_2^{(k)}) = z(\rho\gamma^2).$$

Because x_1 does not change, $x_2^{(k+2)} = x_{cc}^{(k+1)}$ and the reflection point at iteration $k+2$ is given by

$$x_r^{(k+2)} = x_1^{(k)} + \rho(x_1^{(k)} - x_2^{(k+2)}) = x_1^{(k)} - \rho^2\gamma^2(x_1^{(k)} - x_2^{(k)}) = z(-\rho^2\gamma^2).$$

Since $\rho^2\gamma^2 \geq 1$, (4.6) again implies that the value of f at $x_r^{(k+2)}$ exceeds $f_2^{(k+2)}$, and iteration $k+2$ must be an inside contraction. Continuing, if iteration k is an outside contraction followed by j inside contractions, the (rejected) reflection point at iteration $k+j$ is $z(-\rho^2\gamma^j)$ and the (accepted) contraction point is $z(\rho\gamma^{j+1})$.

Because of (4.6), iteration $k+j$ must be an inside contraction as long as $\rho^2\gamma^j \geq 1$. Let c^* denote the smallest integer such that $\rho^2\gamma^{c^*} < 1$; note that $c^* > 2$. It follows that the sequence of contractions divides into blocks, where the jth block consists of a single outside contraction followed by some number c_j of inside contractions, with $c_j \geq c^*$ in each case. Letting k_j denote the iteration index at the start of the jth such block, we have

$$\operatorname{diam}(\Delta_{k_j}) = \rho\gamma^{c_j} \operatorname{diam}(\Delta_{k_{j-1}}) \leq \theta \operatorname{diam}(\Delta_{k_{j-1}}), \quad \text{with} \quad \theta = \rho\gamma^{c^*} < 1.$$

The simplex of largest diameter within each block occurs after the outside contraction, and has diameter $\rho\gamma \operatorname{diam}(\Delta_{k_j})$. Thus we have

$$\lim_{k\to\infty} \operatorname{diam}(\Delta_k) \to 0, \quad \lim_{k\to\infty} x_2^{(k)} = x_1^{(k)}, \quad \text{and} \quad f_2^* = f_1^*,$$

contradicting our assumption that $f_1^* < f_2^*$ and giving the desired result. □

We next show that in all cases the simplex diameter converges to zero, i.e., the simplex shrinks to a point.

LEMMA 4.5. *Let f be a strictly convex function on $\mathcal{R}^1$ with bounded level sets. Assume that the Nelder–Mead algorithm with parameters satisfying $\rho > 0$ and $0 < \gamma < 1$ is applied to f beginning with a nondegenerate initial simplex Δ_0. Then* $\lim_{k\to\infty} \operatorname{diam}(\Delta_k) = 0$.

Proof. Lemma 4.4 shows that $f_1^* = f_2^*$. If $f_1^* = f_{\min}$, this function value is assumed at exactly one point, $x_{\min}$, and the desired result is immediate. If $f_1^* > f_{\min}$, we know from strict convexity that f takes the value f_1^* at exactly two distinct points, denoted by x_1^* and x_2^*, with $x_1^* < x_{\min} < x_2^*$. The vertex function values converge from above to their limits and f is continuous. Thus for any $\epsilon > 0$ there is an iteration index $\widetilde{K}$ such that, for $k \geq \widetilde{K}$, $x_1^{(k)}$ and $x_2^{(k)}$ are confined to $\mathcal{I}_1^\epsilon \cup \mathcal{I}_2^\epsilon$, where

$$\mathcal{I}_1^\epsilon = [x_1^* - \epsilon,\, x_1^*] \quad \text{and} \quad \mathcal{I}_2^\epsilon = [x_2^*,\, x_2^* + \epsilon]. \tag{4.7}$$

There are two cases to consider.

Case 1. Both endpoints $x_1^{(k)}$ and $x_2^{(k)}$ lie in the same interval for infinitely many iterations, i.e., for one of $j = 1, 2$, the relation

$$x_1^{(k)} \in \mathcal{I}_j^\epsilon \ \text{ and } \ x_2^{(k)} \in \mathcal{I}_j^\epsilon \tag{4.8}$$

holds for infinitely many k.

In this case we assert that both endpoints remain in one of these intervals for *all* sufficiently large k. This result is proved by contradiction: assume that for any $\epsilon > 0$ and iteration K_1 where (4.8) holds, there is a later iteration K_2 at which $x_1^{(K_2)}$ and $x_2^{(K_2)}$ are in different intervals. Then, since $\operatorname{diam}(\Delta_{K_1}) \leq \epsilon$ and $\operatorname{diam}(\Delta_{K_2}) \geq x_2^* - x_1^*$, we may pick ϵ so small that $\operatorname{diam}(\Delta_{K_2}) > \max(1, \rho\chi) \operatorname{diam}(\Delta_{K_1})$. The simplex diameter can be increased only by reflection, expansion, or outside contraction, and the maximum factor by which the diameter can increase in a single iteration is $\rho\chi$. If $x_1^{(K_1)}$ and $x_2^{(K_1)}$ are both in $\mathcal{I}_1^\epsilon$, then strict convexity implies that any reflection, expansion, or outside contraction must move toward $\mathcal{I}_2^\epsilon$ (and vice versa if the two vertices lie in $\mathcal{I}_2^\epsilon$). But if ϵ is small enough so that $\epsilon\rho\chi < x_2^* - x_1^*$, then some trial point between iterations K_1 and K_2 must lie in the open interval $(x_1^*,\, x_2^*)$, and by strict convexity its associated function value is less than f_1^*, a contradiction. We conclude that, since the Nelder–Mead endpoints $x_1^{(k)}$ and $x_2^{(k)}$ are in $\mathcal{I}_j^\epsilon$ for all sufficiently large k, and since $f_2^{(k)} \to f_1^{(k)} \to f_1^*$, both endpoints must converge to the point x_j^*, and $\operatorname{diam}(\Delta_k) \to 0$.

Case 2. Both endpoints $x_1^{(k)}$ and $x_2^{(k)}$ are in separate intervals $\mathcal{I}_1^\epsilon$ and $\mathcal{I}_2^\epsilon$ for all $k \geq K_1$.

We show by contradiction that this cannot happen because an inside contraction eventually occurs that generates a point inside (x_1^*, x_2^*). Let x_r^* denote the reflection point for the Nelder–Mead interval $[x_1^*,\, x_2^*]$, where either point may be taken as the "best" point; we know from strict convexity that $f(x_r^*) > f_1^*$, with $f_r^* = f_1^* + \delta_r$ for some $\delta_r > 0$. Because f is continuous and $x_r^{(k)}$ is a continuous function of $x_1^{(k)}$ and $x_2^{(k)}$, it follows that, given any $\delta > 0$, eventually $f_1^{(k)}$, $f_2^{(k)}$, and $f_r^{(k)}$ are within δ of their limiting values. Thus, for sufficiently large k, $f_r^{(k)} > f_2^{(k)} \geq f_1^{(k)}$ and an inside contraction will be taken.

Since $x_1^{(k)}$ and $x_2^{(k)}$ are in different intervals, the inside contraction point $x_{cc}^{(k)}$ satisfies

$$x_1^* - \epsilon + \gamma\big(x_2^* - (x_1^* - \epsilon)\big) \;\leq\; x_{cc}^{(k)} \;\leq\; x_2^* + \epsilon + \gamma\big(x_1^* - (x_2^* + \epsilon)\big).$$

If ϵ is small enough, namely, $\epsilon < \gamma(x_2^* - x_1^*)/(1-\gamma)$, then

$$x_1^* \;<\; x_1^* + \gamma(x_2^* - x_1^*) - (1-\gamma)\epsilon \;\leq\; x_{cc}^{(k)} \;\leq\; x_2^* - \gamma(x_2^* - x_1^*) + (1-\gamma)\epsilon < x_2^*,$$

i.e., $x_{cc}^{(k)}$ lies in the open interval $(x_1^*,\, x_2^*)$ and $f(x_{cc}^{(k)}) < f_1^*$, a contradiction. □

We now combine these lemmas to prove Theorem 4.1.

Proof of Theorem 4.1. (Convergence of Nelder–Mead in one dimension.) Lemma 4.2 shows that $x_{\min}$ is eventually bracketed by the worst vertex and the expansion point, i.e., for some iteration K,

$$x_{\min} \in \operatorname{int}\big(x_2^{(K)}, x_1^{(K)} + \rho\chi(x_1^{(K)} - x_2^{(K)})\big).$$

Since the constant N_{NM} of (4.3) satisfies $N_{NM} \geq \rho\chi$, Lemma 4.3 shows that, for all $k \geq K$, $x_{\min}$ satisfies the proximity property (4.4),

$$x_{\min} \in \operatorname{int}\big(x_2^{(k)}, x_1^{(k)} + N_{NM}(x_1^{(k)} - x_2^{(k)})\big),$$

which implies that

$$|x_{\min} - x_1^{(k)}| \leq N_{NM}\,\mathrm{diam}(\Delta_k). \tag{4.9}$$

Lemma 4.5 shows that $\mathrm{diam}(\Delta_k) \to 0$. Combined with (4.9), this gives the desired result. □

4.3. Linear convergence with $\rho = 1$.

When the reflection coefficient is the standard choice $\rho = 1$, the Nelder–Mead method not only converges to the minimizer, but its convergence rate is eventually M-step linear, i.e., the distance from the best vertex to the optimal point decreases every M steps by at least a fixed multiplicative constant less than one. This result follows from analyzing the special structure of permitted Nelder–Mead move sequences.

THEOREM 4.2. (Linear convergence of Nelder–Mead in one dimension with $\rho = 1$.) *Let f be a strictly convex function on $\mathcal{R}^1$ with bounded level sets. Assume that the Nelder–Mead algorithm with reflection coefficient $\rho = 1$, and expansion and contraction coefficients satisfying $\chi > 1$ and $0 < \gamma < 1$, is applied to f beginning with a nondegenerate initial simplex Δ_0. Then there is an integer M depending only on χ and γ such that*

$$\mathrm{diam}(\Delta_{k+M}) \leq \tfrac{1}{2}\,\mathrm{diam}(\Delta_k) \quad \textit{for all} \quad k \geq K,$$

where K is the iteration index defined in Lemma 4.2.

As the first step in proving this theorem, we obtain two results unique to dimension 1 about sequences of Nelder–Mead iterations.

LEMMA 4.6. *Let f be a strictly convex function on $\mathcal{R}^1$ with bounded level sets, and assume that the Nelder–Mead method with parameters $\rho = 1$, $\chi > 1$, and $0 < \gamma < 1$, is applied to f beginning with a nondegenerate initial simplex. Then*

(1) *the number of consecutive reflections is bounded by $r^* = \lceil \chi - 1 \rceil$;*

(2) *the iteration immediately following a reflection cannot be an expansion.*

Proof. For any iteration k, define $z^{(k)}(\xi)$ as

$$z^{(k)}(\xi) \equiv x_1^{(k)} + \xi\big(x_1^{(k)} - x_2^{(k)}\big), \tag{4.10}$$

so that $x_2^{(k)} = z^{(k)}(-1)$, $x_r^{(k)} = z^{(k)}(1)$, and $x_e^{(k)} = z^{(k)}(\chi)$.

If iteration k is a reflection,

$$f_r^{(k)} < f_1^{(k)}, \quad f_e^{(k)} \geq f_r^{(k)}, \quad x_1^{(k+1)} = x_r^{(k)}, \quad \text{and} \quad x_2^{(k+1)} = x_1^{(k)}. \tag{4.11}$$

Applying Lemma 4.1 to the first two relations in (4.11), we can see that $x_{\min} \in \operatorname{int}(x_1^{(k)}, x_e^{(k)})$ and

$$f\big(z^{(k)}(\xi)\big) \geq f_1^{(k+1)} \quad \text{if} \quad \xi \geq \chi. \tag{4.12}$$

Starting with iteration k, the (potential) ℓth consecutive reflection point is given by

$$x_r^{(k+\ell-1)} = x_1^{(k)} + \ell(x_1^{(k)} - x_2^{(k)}) = z^{(k)}(\ell), \tag{4.13}$$

which can be accepted only if its function value is strictly less than $f(x_1^{(k+\ell-1)})$. Strict convexity and (4.12) show that any point $z^{(k)}(\xi)$ with $\xi \geq \chi$ cannot be an accepted reflection point. Thus the number of consecutive reflections is bounded by the integer r^* satisfying

$$r^* < \chi \;\text{ and }\; r^* + 1 \geq \chi, \quad \text{i.e.,} \quad r^* = \lceil \chi - 1 \rceil.$$

This completes the proof of (1).

If iteration k is a reflection, the expansion point at iteration $k+1$ is given by

$$x_e^{(k+1)} = x_1^{(k+1)} + \chi(x_1^{(k+1)} - x_2^{(k+1)}) = x_1^{(k)} + (1+\chi)(x_1^{(k)} - x_2^{(k)}) = z^{(k)}(1+\chi).$$

Relation (4.12) implies that the function value at $x_e^{(k+1)}$ exceeds $f_1^{(k+1)}$, so that $x_e^{(k+1)}$ will not be accepted. This proves result (2) and shows that the iteration immediately following a successful reflection must be either a reflection or a contraction. □

Note that $r^* = 1$ whenever the expansion coefficient $\chi \leq 2$; thus there cannot be two consecutive reflections with the standard Nelder–Mead coefficients (2.2) for $n = 1$.

As a corollary, we show that a contraction must occur no later than iteration $K + r^*$, where K is the first iteration at which the minimizer is bracketed by x_2 and the expansion point (Lemma 4.2).

COROLLARY 4.1. *Let f be a strictly convex function on $\mathcal{R}^1$ with bounded level sets. Assume that the Nelder–Mead algorithm with $\rho = 1$ is applied to f beginning with a nondegenerate initial simplex Δ_0, and let K denote the iteration index defined by Lemma 4.2 at which, for the first time, $f_1^{(K)} \leq f_e^{(K)}$. Then a contraction must occur no later than iteration $K + r^*$.*

Proof. There are two cases. If $f_r^{(K)} \geq f_1^{(K)}$, iteration K is a contraction, and the result is immediate. Otherwise, if $f_r^{(K)} < f_1^{(K)}$, iteration K is a reflection. Lemma 4.6 shows that there cannot be more than r^* consecutive reflections, and any sequence of consecutive reflections ends with a contraction. Hence a contraction must occur no later than iteration $K + r^*$. □

The next lemma derives a bound on the number of consecutive expansions immediately following a contraction.

LEMMA 4.7. (Bounded consecutive expansions.) *Let f be a strictly convex function on $\mathcal{R}^1$ with bounded level sets. Assume that the Nelder–Mead algorithm with $\rho = 1$, $\chi > 1$, and $0 < \gamma < 1$ is applied to f beginning with a nondegenerate initial simplex Δ_0. Let $N_{NM} = \max(\chi, 1/\gamma)$, which is equivalent to its general definition* (4.3) *when $\rho = 1$. If iteration k is a contraction, then for all subsequent iterations there can be no more than j^* consecutive expansion steps, where j^* is defined as follows:*

(a) *if $\chi = N_{NM}$, $j^* = 0$;*

(b) *if $\chi < N_{NM}$, j^* is the largest integer satisfying $\chi + \chi^2 + \cdots + \chi^{j^*} < N_{NM}$.*

Proof. Since iteration k is a contraction, $x_{\min} \in \text{int}(x_2^{(k)},\ x_r^{(k)})$. Thus the proximity property (4.4) is satisfied at iteration k and, by Lemma 4.3, for all subsequent iterations. The first expansion in a sequence of consecutive expansions must immediately follow a contraction (see result (2) of Lemma 4.6), and strict convexity imposes a bound on the number of subsequent consecutive expansions.

Using the notation of (4.10), we consider inequalities that apply to the best function value $f_1^{(k+1)}$ at the next iteration, which is (possibly) the first expansion step in a sequence of consecutive expansions.

Case 1. If $f_r^{(k)} < f_2^{(k)}$, iteration k is an outside contraction with $x_c^{(k)} = x_1^{(k)} + \gamma(x_1^{(k)} - x_2^{(k)})$.

(i) If $f_c^{(k)} \geq f_1^{(k)}$, the next Nelder–Mead interval is defined by $x_2^{(k+1)} = x_c^{(k)}$ and $x_1^{(k+1)} = x_1^{(k)}$, and $x_{\min} \in \text{int}(x_2^{(k)}, x_2^{(k+1)})$. (The tie-breaking rule in section 2 is invoked if $f_c^{(k)} = f_1^{(k)}$.) If an expansion occurs, the interval will expand toward $x_2^{(k)}$, which satisfies

$$(4.14)\quad x_2^{(k)} = x_1^{(k+1)} + \left(x_1^{k+1} - x_2^{(k+1)}\right)/\gamma = z^{(k+1)}(1/\gamma), \ \text{ with } \ f_2^{(k)} > f_1^{(k+1)}.$$

(ii) If $f_c^{(k)} < f_1^{(k)}$, the next Nelder–Mead interval is defined by $x_2^{(k+1)} = x_1^{(k)}$ and $x_1^{(k+1)} = x_c^{(k)}$, and $x_{\min} \in \text{int}(x_2^{(k+1)}, x_r^{(k)})$. Any expansion will be toward $x_r^{(k)}$, which satisfies

$$(4.15)\qquad x_r^{(k)} = x_1^{(k+1)} + (1/\gamma - 1)\left(x_1^{(k+1)} - x_2^{(k+1)}\right) = z^{(k+1)}(1/\gamma - 1),$$

with $f_r^{(k)} > f_1^{(k+1)}$.

Case 2. If $f_r^{(k)} \geq f_2^{(k)}$, iteration k is an inside contraction with $x_{cc}^{(k)} = x_1^{(k)} - \gamma(x_1^{(k)} - x_2^{(k)})$.

(i) If $f_{cc}^{(k)} \geq f_1^{(k)}$, the next Nelder–Mead interval is defined by $x_2^{(k+1)} = x_{cc}^{(k)}$ and $x_1^{(k+1)} = x_1^{(k)}$, and $x_{\min} \in \text{int}(x_2^{(k+1)}, x_r^{(k)})$. (The tie-breaking rule in section 2 is invoked if $f_{cc}^{(k)} = f_1^{(k)}$.) If an expansion occurs, the interval will expand toward $x_r^{(k)}$, which satisfies

$$(4.16)\qquad x_r^{(k)} = x_1^{(k+1)} + \left(x_1^{(k+1)} - x_2^{(k+1)}\right)/\gamma = z^{(k+1)}(1/\gamma),$$

with $f_r^{(k)} > f_1^{(k+1)}$.

(ii) If $f_{cc}^{(k)} < f_1^{(k)}$, the next Nelder–Mead interval is defined by $x_2^{(k+1)} = x_1^{(k)}$ and $x_1^{(k+1)} = x_{cc}^{(k)}$, and $x_{\min} \in \operatorname{int}(x_2^{(k)}, x_2^{(k+1)})$. Any expansion will be toward $x_2^{(k)}$, which satisfies

$$x_2^{(k)} = x_1^{(k+1)} + (1/\gamma - 1)\left(x_1^{k+1} - x_2^{(k+1)}\right) = z^{(k+1)}(1/\gamma - 1), \tag{4.17}$$

with $f_2^{(k)} > f_1^{(k+1)}$.

For each of the four cases 1(i)–2(ii), the value of f at $z^{(k+1)}(\xi)$ exceeds $f_1^{(k+1)}$ for some ξ that is equal to or bounded above by N_{NM}. Applying result (2) of Lemma 4.1 to the interval in which $x_{\min}$ lies and the corresponding expression from (4.14)–(4.17), we conclude that, if a sequence of consecutive expansions *begins* at iteration $k+1$, then

$$f(z^{(k+1)}(\xi)) > f(x_1^{(k+1)}) \quad \text{whenever} \quad \xi \geq N_{NM}. \tag{4.18}$$

The remainder of the proof is similar to that of Lemma 4.6. The expansion point at iteration $k+1$ is $x_e^{(k+1)} = z^{(k+1)}(\chi)$. If $\chi = N_{NM}$, it follows from (4.18) that this point will not be accepted, and consequently iteration $k+1$ cannot be an expansion; this corresponds to the case $j^* = 0$. If $\chi < N_{NM}$, then, starting with iteration $k+1$, the (potential) jth consecutive expansion point for $j \geq 1$ is given by

$$x_e^{(k+j)} = z^{(k+1)}\left(\chi + \chi^2 + \cdots + \chi^j\right). \tag{4.19}$$

This point can be accepted only if its function value is strictly less than $f(x_1^{(k+j)})$, which strictly decreases after each accepted expansion. Relations (4.18) and (4.19) together show that, for $j \geq 1$, $x_e^{(k+j)}$ might be accepted only if

$$\chi + \chi^2 + \cdots + \chi^j \quad < \quad N_{NM}.$$

Applying the definition of j^*, it follows that the value of j must be bounded above by j^*. □

For the standard expansion coefficient $\chi = 2$, the value of N_{NM} is $\max(2, 1/\gamma)$ and the values of j^* for several ranges of γ are

$$j^* = 0 \;\text{ when }\; \tfrac{1}{2} \leq \gamma < 1; \quad j^* = 1 \;\text{ when }\; \tfrac{1}{6} \leq \gamma < \tfrac{1}{2}; \quad j^* = 2 \;\text{ when }\; \tfrac{1}{14} \leq \gamma < \tfrac{1}{6}.$$

In the "standard" Nelder–Mead algorithm with contraction coefficient $\gamma = \frac{1}{2}$, the zero value of j^* means that no expansion steps can occur once the minimizer is bracketed by the worst point and the reflection point at any iteration.

We now examine the effects of valid Nelder–Mead move sequences on the simplex diameter.

LEMMA 4.8. *Let f be a strictly convex function on $\mathcal{R}^1$ with bounded level sets. Assume that the Nelder–Mead algorithm with $\rho = 1$, $\chi > 1$, and $0 < \gamma < 1$ is applied to f beginning with a nondegenerate initial simplex Δ_0. Let Δ denote the simplex immediately following any contraction, and Δ' the simplex immediately following the next contraction. Then there exists a value φ depending only on χ and γ such that* $\operatorname{diam}(\Delta') \leq \varphi \operatorname{diam}(\Delta)$, *where $\varphi < 1$.*

Proof. Lemma 4.7 shows that the number of consecutive expansions between any two contractions cannot exceed j^*. Since $N_{NM} = \max(1/\gamma, \chi)$ and reflection does not change the diameter, the worst-case growth occurs when j^* expansions are followed by a contraction, which corresponds to $\varphi = \chi^{j^*}\gamma$. If $j^* = 0$, $\varphi = \gamma$ and is consequently less than 1. If $N_{NM} = \chi$, j^* must be zero. In the remaining case when $N_{NM} = 1/\gamma$ and $j^* > 0$, the condition defining j^* (part (b) of Lemma 4.7) may be written as

$$\gamma(\chi + \cdots + \chi^{j^*}) < 1, \ \text{ which implies that } \ \varphi = \gamma\chi^{j^*} < 1, \text{ the desired result.} \qquad \square$$

Combining all these results, we prove M-step linear convergence of Nelder–Mead in dimension 1 when $\rho = 1$.

Proof of Theorem 4.2. In proving M-step linear convergence, we use a directed graph to depict the structure of valid Nelder–Mead move sequences. We have shown thus far that the minimizer is bracketed at iteration K (Lemma 4.2) and that a contraction must occur no later than iteration $K+r^*$ (Lemmas 4.6 and Corollary 4.1). Thereafter, no more than j^* consecutive expansions can occur (Lemma 4.7), and any sequence of consecutive expansions must end with either a contraction alone or a sequence of at most r^* consecutive reflections followed by a contraction (see Lemma 4.6).

The structure of legal iteration sequences following a contraction can thus be represented by a directed graph with four states (nodes): expansion, reflection, and the two forms of contraction. Each state is labeled by the absolute value of its move type, so that an inside contraction is labeled "γ", an outside contraction is labeled "$\rho\gamma$", a reflection is labeled "ρ", and an expansion is labeled "$\rho\chi$". For example, Figure 3 shows the graph corresponding to $\rho = 1$, $\chi = 2$, and any contraction coefficient satisfying $\frac{1}{14} \le \gamma < \frac{1}{6}$. For these coefficients, at most two consecutive expansion steps can occur ($j^* = 2$), and at most one consecutive reflection ($r^* = 1$). (Because $\rho = 1$, we have not distinguished between inside and outside contractions.)

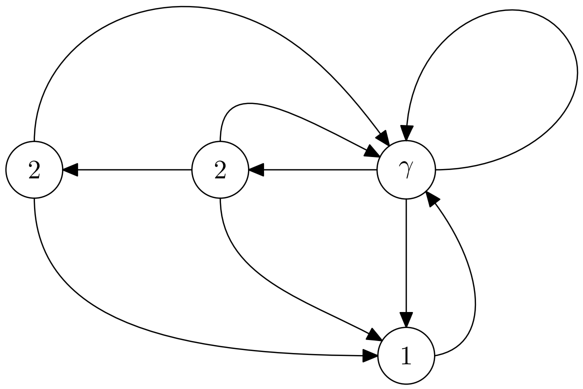


FIG. 3. *Directed graph depicting legal Nelder–Mead moves for $\rho = 1$, $\chi = 2$, and $\frac{1}{14} \le \gamma < \frac{1}{6}$.*

According to (4.1), the simplex diameter is multiplied by ρ for a reflection, $\rho\chi$ for an expansion, $\rho\gamma$ for an outside contraction, and γ for an inside contraction. Starting in the contraction state with initial diameter 1, the diameter of the Nelder–Mead interval after any sequence of moves is thus the product of the state labels encountered. The first contraction in the Nelder–Mead method can occur no later than iteration $K + r^*$. Thereafter, Lemmas 4.6 and 4.7 show that any minimal cycle in the graph of valid Nelder–Mead moves (i.e., a cycle that does not pass through any node twice) has length at most $j^* + r^* + 1$; Lemma 4.8 shows that the product of state labels over any cycle in the Nelder–Mead graph cannot exceed φ. For any integer m, a path of length $m(j^* + r^* + 1)$ must contain at least m minimal cycles. Given any such path, we can remove minimal cycles until at most $j^* + r^*$ edges are left over. Consequently, the simplex diameter at the end of the associated sequence of Nelder–Mead iterations must be multiplied by a factor no larger than $\chi^{j^*+r^*}\varphi^m$. If we choose m as the smallest value such that

$$\chi^{j^*+r^*}\varphi^m \le \tfrac{1}{2}, \quad \text{then } M = m(j^* + r^* + 1) \text{ satisfies } \operatorname{diam}(\Delta_{k+M}) \le \tfrac{1}{2}\operatorname{diam}(\Delta_k),$$

which gives the desired result. □

M-step linear convergence can also be proved for certain ranges of parameter values with $\rho \neq 1$ by imposing restrictions that guarantee, for example, that $j^* = 0$ and $r^* = 1$.

4.4. A pattern search method interpretation of Nelder–Mead for $n = 1$. Pattern search methods [13] are direct search methods that presuppose a lattice grid pattern for search points. Torczon [14] has recently informed us that the analysis [13] for pattern search methods can be adapted in dimension 1 to the Nelder–Mead method when

$$\rho = 1 \qquad \text{and } \chi \text{ and } \gamma \text{ are rational.} \tag{4.20}$$

(These restrictions are satisfied for the standard coefficients $\rho = 1$, $\chi = 2$, and $\gamma = \frac{1}{2}$.) The condition $\rho = 1$ is needed to guarantee that, following an outside contraction at iteration k, the reflection point at iteration $k+1$ is identical to the inside contraction point at iteration k (and vice versa). Rationality of χ and γ is needed to retain the lattice structure that underlies pattern search methods. When (4.20) holds and f is once-continuously differentiable, the Nelder–Mead method generates the same sequence of points as a (related) pattern search method with relabeled iterations. Consequently, the results in [13] imply that $\liminf |\nabla f(x_k)| \to 0$, where x_k denotes the best point in Δ_k.

5. Standard Nelder–Mead in dimension 2 for strictly convex functions.

In this section we consider the *standard* Nelder–Mead algorithm, with coefficients $\rho = 1$, $\chi = 2$, and $\gamma = \frac{1}{2}$, applied to a strictly convex function $f(\mathbf{x})$ on $\mathcal{R}^2$ with bounded level sets. The assumption that $\rho = 1$ is essential in our analysis.

We denote the (necessarily unique) minimizer of f by $\mathbf{x}_{\min}$, and let $f_{\min} = f(\mathbf{x}_{\min})$. Note that the level set $\{\mathbf{x} \mid f(\mathbf{x}) \le \mu\}$ is empty if $\mu < f_{\min}$, the single point $\mathbf{x}_{\min}$ if $\mu = f_{\min}$, and a closed convex set if $\mu > f_{\min}$.

5.1. Convergence of vertex function values. Our first result shows that, for the standard Nelder–Mead algorithm, the limiting function values at the vertices are equal.

THEOREM 5.1. (Convergence of vertex function values for $n = 2$.) *Let f be a strictly convex function on $\mathcal{R}^2$ with bounded level sets. Assume that the Nelder–Mead algorithm with reflection coefficient $\rho = 1$ and contraction coefficient $\gamma = \frac{1}{2}$ is applied to f beginning with a nondegenerate initial simplex Δ_0. Then the three limiting vertex function values are the same, i.e.,*

$$f_1^* = f_2^* = f_3^*.$$

Proof. Corollary 3.1, which applies in any dimension, gives the result immediately if the best vertex $\mathbf{x}_1^{(k)}$ changes infinitely often. The following lemma treats the only remaining case, in which $\mathbf{x}_1^{(k)}$ eventually becomes constant.

LEMMA 5.1. *Let f be a strictly convex function on $\mathcal{R}^2$ with bounded level sets. Assume that the Nelder–Mead algorithm with $\rho = 1$ and $\gamma = \frac{1}{2}$ is applied to f beginning with a nondegenerate initial simplex Δ_0. If the best vertex $\mathbf{x}_1^{(k)}$ is constant for all k, then the simplices Δ_k converge to the point $\mathbf{x}_1^{(0)}$ as $k \to \infty$.*

Proof. Without loss of generality, the (constant) best vertex $\mathbf{x}_1$ may be taken as the origin. The proof that $\mathbf{x}_2$ and $\mathbf{x}_3$ converge to the origin has four elements: (i) a matrix recursion that defines the Nelder–Mead vertices at the infinite subsequence of iterations when $\mathbf{x}_2$ changes; (ii) a special norm that measures progress toward the origin; (iii) bounds on this norm obtained from the singular values of a matrix constrained to a subspace; and (iv) the illegality of certain patterns of Nelder–Mead move types in the iteration subsequence.

(i) *The matrix recursion.* We know from Lemma 3.6 that the next-worst vertex $\mathbf{x}_2^{(k)}$ must change infinitely often. There is thus a subsequence of iterations $\{k_\ell\}$, $\ell = 0, 1, \ldots,$ with $k_0 = 0$, where $\mathbf{x}_2$ changes, i.e.,

$$\mathbf{x}_2^{(k_{\ell+1})} \neq \mathbf{x}_2^{(k_\ell)} \quad \text{and} \quad \mathbf{x}_2^{(i)} = \mathbf{x}_2^{(i-1)}, \quad i = k_\ell + 1, \ldots, k_{\ell+1} - 1.$$

We then define new sequences $\tilde{\mathbf{x}}_2$ and $\tilde{\mathbf{x}}_3$ from

$$\tilde{\mathbf{x}}_2^{(\ell)} = \mathbf{x}_2^{(k_\ell)} \quad \text{and} \quad \tilde{\mathbf{x}}_3^{(\ell)} = \mathbf{x}_3^{(k_\ell)}. \tag{5.1}$$

Because $\mathbf{x}_1$ is constant and $\mathbf{x}_2$ changes at iteration k_ℓ, $\mathbf{x}_3$ thereupon becomes the "old" $\mathbf{x}_2$, i.e.,

$$\tilde{\mathbf{x}}_3^{(\ell)} = \tilde{\mathbf{x}}_2^{(\ell-1)}. \tag{5.2}$$

For each iteration strictly between k_ℓ and $k_{\ell+1}$, only $\mathbf{x}_3$ changes, so that

$$\mathbf{x}_3^{(i)} = \tfrac{1}{2}\mathbf{x}_2^{(i-1)} + \tau_{i-1}\big(\tfrac{1}{2}\mathbf{x}_2^{(i-1)} - \mathbf{x}_3^{(i-1)}\big) \quad \text{for} \quad i = k_\ell + 1, \ldots, k_{\ell+1} - 1, \tag{5.3}$$

where τ_i is the type of iteration i. Note that any iteration in which only $\mathbf{x}_3$ changes must be a contraction, so that τ_i is necessarily $\pm\frac{1}{2}$ when $k_\ell < i < k_{\ell+1}$; the value of τ_{k_ℓ}, however, can be 1 or $\pm\frac{1}{2}$. Since only $\mathbf{x}_3$ is changing between iterations k_ℓ and $k_{\ell+1}$, relation (5.3) implies that

$$\mathbf{x}_3^{(k_\ell+j)} = \tfrac{1}{2}\mathbf{x}_2^{(k_\ell)} + (-1)^{j-1} \prod_{i=0}^{j-1} \tau_{k_\ell+i} \left(\tfrac{1}{2}\mathbf{x}_2^{(k_\ell)} - \mathbf{x}_3^{(k_\ell)}\right) \tag{5.4}$$

for $j = 1, \ldots, k_{\ell+1} - k_\ell - 1$.

Using (5.1), (5.2) and (5.4), we obtain an expression representing $\tilde{\mathbf{x}}_2^{(\ell+1)}$ entirely in terms of $\tilde{\mathbf{x}}_2^{(\ell)}$ and $\tilde{\mathbf{x}}_2^{(\ell-1)}$:

$$\tilde{\mathbf{x}}_2^{(\ell+1)} = \tfrac{1}{2}\tilde{\mathbf{x}}_2^{(\ell)} + \tilde{\tau}_\ell\big(\tfrac{1}{2}\tilde{\mathbf{x}}_2^{(\ell)} - \tilde{\mathbf{x}}_2^{(\ell-1)}\big), \tag{5.5}$$

where

$$\tilde{\tau}_\ell = (-1)^{\tilde{\ell}} \prod_{i=0}^{\tilde{\ell}} \tau_{k_\ell+i}, \quad \text{with} \quad \tilde{\ell} = k_{\ell+1} - k_\ell - 1.$$

Because reflections cannot occur between iterations k_ℓ and $k_{\ell+1}$, we know that $|\tilde{\tau}_\ell| \le \frac{1}{2}$ or $\tilde{\tau}_\ell = 1$. (The latter happens only when iterations k_ℓ and $k_{\ell+1}$ are consecutive).

Using matrix notation, we have

$$\tilde{\mathbf{x}}_2^{(\ell)} = \begin{pmatrix} \tilde{x}_{21}^{(\ell)} \\ \tilde{x}_{22}^{(\ell)} \end{pmatrix} = \begin{pmatrix} u_\ell \\ v_\ell \end{pmatrix}; \quad \text{(5.1) then gives } \tilde{\mathbf{x}}_3^{(\ell)} = \tilde{\mathbf{x}}_2^{(\ell-1)} = \begin{pmatrix} u_{\ell-1} \\ v_{\ell-1} \end{pmatrix}. \tag{5.6}$$

The Nelder–Mead update embodied in (5.5) can be written as a matrix recursion in u and v:

$$\begin{pmatrix} u_{\ell+1} & v_{\ell+1} \\ u_\ell & v_\ell \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(1+\tilde{\tau}_\ell) & -\tilde{\tau}_\ell \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u_\ell & v_\ell \\ u_{\ell-1} & v_{\ell-1} \end{pmatrix}. \tag{5.7}$$

Define $\mathbf{u}_\ell$ and $\mathbf{v}_\ell$ by

$$\mathbf{u}_\ell = \begin{pmatrix} u_\ell \\ u_{\ell-1} \end{pmatrix} \quad \text{and} \quad \mathbf{v}_\ell = \begin{pmatrix} v_\ell \\ v_{\ell-1} \end{pmatrix},$$

so that $\mathbf{u}_\ell$ contains the x-coordinates of the current second-worst and worst vertices, $\tilde{\mathbf{x}}_2^{(\ell)}$ and $\tilde{\mathbf{x}}_3^{(\ell)}$, and $\mathbf{v}_\ell$ contains their y coordinates. The desired conclusion of Lemma 5.1 follows if we can show that

$$\lim_{\ell\to\infty} \mathbf{u}_\ell = 0 \quad \text{and} \quad \lim_{\ell\to\infty} \mathbf{v}_\ell = 0. \tag{5.8}$$

We shall prove only the first relation in (5.8); the proof of the second is similar.

(ii) *Measuring progress toward the origin.* To prove convergence of $\mathbf{u}_\ell$ to the origin, it might appear that we could simply apply norm inequalities to the matrix equation (5.7). Unfortunately, the two-norm of the matrix in (5.7) exceeds one for all valid $\tilde{\tau}_\ell$, which means that $\|\mathbf{u}_{\ell+1}\|$ can be larger than $\|\mathbf{u}_\ell\|$. Hence we need to find a suitable nonincreasing size measure associated with each Nelder–Mead iteration (5.7).

Such a size measure is given by a positive definite quadratic function Q of two scalar arguments (or, equivalently, of a 2-vector):

$$Q(a,b) = 2(a^2 - ab + b^2) = a^2 + b^2 + (a-b)^2. \tag{5.9}$$

Evaluating $Q(\mathbf{u}_{\ell+1})$ using (5.7) gives

$$Q(\mathbf{u}_{\ell+1}) = (\tfrac{3}{2} + \tfrac{1}{2}\tilde{\tau}_\ell^2)u_\ell^2 - 2\tilde{\tau}_\ell^2 u_\ell u_{\ell-1} + 2\tilde{\tau}_\ell^2 u_{\ell-1}^2.$$

After substitution and manipulation, we obtain

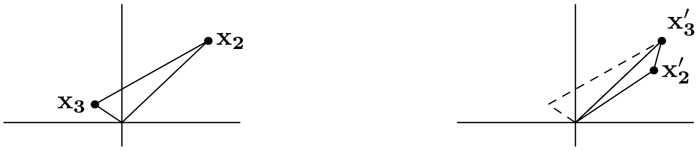
$$Q(\mathbf{u}_\ell) - Q(\mathbf{u}_{\ell+1}) \;=\; 2(1-\tilde{\tau}_\ell^2)(\tfrac{1}{2}u_\ell - u_{\ell-1})^2, \tag{5.10}$$

which shows that

$$Q(\mathbf{u}_{\ell+1}) \;\le\; Q(\mathbf{u}_\ell) \quad \text{when} \quad -1 \le \tilde{\tau}_\ell \le 1. \tag{5.11}$$

It follows that Q is, as desired, a size measure that is nonincreasing for all valid values of $\tilde{\tau}_\ell$. Furthermore, because Q is positive definite, we can prove that $\mathbf{u}_\ell \to 0$ by showing that $Q(\mathbf{u}_\ell) \to 0$.

An obvious and appealing geometric interpretation of Q in terms of the Nelder–Mead simplices is that the quantity $Q(\mathbf{u}_\ell) + Q(\mathbf{v}_\ell)$ is the sum of the squared side lengths of the Nelder–Mead triangle at iteration k_ℓ, with vertices at the origin, $\tilde{\mathbf{x}}_2^{(\ell)}$, and $\tilde{\mathbf{x}}_3^{(\ell)}$. Relation (5.11) indicates that, after a reflection or contraction in which $\mathbf{x}_2$ changes, the sum of the squared side lengths of the new Nelder–Mead triangle cannot increase, even though $\|\mathbf{u}_{\ell+1}\|$ may be larger. Figure 4 depicts an example in which, after an outside contraction, both $\|\mathbf{u}_{\ell+1}\|$ and $\|\mathbf{v}_{\ell+1}\|$ increase. Nonetheless, the sum of the squared triangle side lengths is reduced.



sum of squared sides = 3.895

$$x_{21}^2 + x_{31}^2 = 0.9925$$

$$x_{22}^2 + x_{32}^2 = 0.85$$

sum of squared sides = 2.9003

$$(x'_{21})^2 + (x'_{31})^2 = 1.646$$

$$(x'_{22})^2 + (x'_{32})^2 = 1.1406$$

FIG. 4. *A triangle and its outside contraction.*

(iii) *Singular values in a subspace.* To obtain worst-case bounds on the size of Q, it is convenient to interpret Q as the two-norm of a specially structured 3-vector derived from $\mathbf{u}_\ell$. Within the context of a Nelder–Mead iteration (5.6), we use the notation

$$\boldsymbol{\xi}_\ell = \begin{pmatrix} u_\ell \\ u_{\ell-1} \\ u_\ell - u_{\ell-1} \end{pmatrix}, \quad \text{so that} \quad Q(\mathbf{u}_\ell) = \|\boldsymbol{\xi}_\ell\|^2. \tag{5.12}$$

The structure of $\boldsymbol{\xi}$ (5.12) can be formalized by observing that it lies in the two-dimensional null space of the vector $(1, -1, -1)$. Let Z denote the following 3×2 matrix whose columns form a (nonunique) orthonormal basis for this null space:

$$Z = \begin{pmatrix} z_1 & z_2 \end{pmatrix}, \quad \text{where} \quad z_1 = \frac{1}{\sqrt{6}} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \quad \text{and} \quad z_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}.$$

Let $\mathbf{q}_\ell$ denote the unique 2-vector satisfying

$$\boldsymbol{\xi}_\ell = Z\mathbf{q}_\ell = \begin{pmatrix} u_\ell \\ u_{\ell-1} \\ u_\ell - u_{\ell-1} \end{pmatrix}. \tag{5.13}$$

Since $Z^T Z = I$, we have

$$\|\boldsymbol{\xi}_\ell\| = \|\mathbf{q}_\ell\| \quad \text{and} \quad Q(\mathbf{u}_\ell) = \|\boldsymbol{\xi}_\ell\|^2 = \|\mathbf{q}_\ell\|^2, \tag{5.14}$$

so that we may use $\|\mathbf{q}_\ell\|$ to measure Q.

The Nelder–Mead move (5.7) can be written in terms of a 3×3 matrix M_ℓ applied to $\boldsymbol{\xi}_\ell$:

$$\boldsymbol{\xi}_{\ell+1} = M_\ell \boldsymbol{\xi}_\ell, \quad \text{where} \quad M_\ell = \begin{pmatrix} \frac{1}{2}(1+\tilde{\tau}_\ell) & -\tilde{\tau}_\ell & 0 \\ 1 & 0 & 0 \\ -\frac{1}{2} & -\frac{1}{2}\tilde{\tau}_\ell & \frac{1}{2}\tilde{\tau}_\ell \end{pmatrix}. \tag{5.15}$$

As we have already shown, the special structure of the vector $\boldsymbol{\xi}_\ell$ constrains the effects of the transformation M_ℓ to a subspace. To analyze these effects, note that, by construction of M_ℓ, its application to any vector in the column space of Z produces a vector in the same column space, i.e.,

$$M_\ell Z = ZW_\ell, \quad \text{where} \quad W_\ell = Z^T M_\ell Z. \tag{5.16}$$

A single Nelder–Mead move (5.7) is thus given by

$$\boldsymbol{\xi}_{\ell+1} = M_\ell \boldsymbol{\xi}_\ell = M_\ell Z \mathbf{q}_\ell = ZW_\ell \mathbf{q}_\ell,$$

so that, using (5.14),

$$Q(\mathbf{u}_{\ell+1}) = \|\boldsymbol{\xi}_{\ell+1}\|^2 = \|W_\ell \mathbf{q}_\ell\|^2,$$

and we may deduce information about the behavior of Q from the structure of the 2×2 matrix W.

Direct calculation shows that, for any $\tilde{\tau}_\ell$, W_ℓ is the product of an orthonormal matrix $\tilde{Z}$ and a diagonal matrix:

$$W_\ell = \tilde{Z}\varSigma_\ell, \quad \text{where} \quad \tilde{Z} = \begin{pmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{pmatrix} \quad \text{and} \quad \varSigma_\ell = \begin{pmatrix} 1 & 0 \\ 0 & -\tilde{\tau}_\ell \end{pmatrix}, \tag{5.17}$$

with $\tilde{Z}$ representing a rotation through 60 degrees. The form (5.17), analogous to the singular value decomposition apart from the possibly negative diagonal element of $\varSigma_\ell$, reveals that the extreme values of $\|W_\ell \mathbf{q}_\ell\|$ are

$$\begin{aligned} \max_{\|\mathbf{q}\|=1} \|W_\ell \mathbf{q}\| &= 1 \qquad \text{when} \quad \mathbf{q} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \\ \min_{\|\mathbf{q}\|=1} \|W_\ell \mathbf{q}\| &= |\tilde{\tau}_\ell| \quad \text{when} \quad \mathbf{q} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \end{aligned} \tag{5.18}$$

For a reflection ($\tilde{\tau}_\ell = 1$), the value of Q is unchanged for all $\mathbf{q}$ and hence for all $\mathbf{u}$. When $|\tilde{\tau}_\ell| = \frac{1}{2}$, relationship (5.13) indicates how the extremes of (5.18) map into $\mathbf{u}$-space. The value of Q remains constant, i.e., $Q(\mathbf{u}_{\ell+1}) = Q(\mathbf{u}_\ell)$, only when $\mathbf{u}_\ell$ has the form $(2\alpha, \alpha)$ for some nonzero α; this can also be seen directly in (5.10). The maximum reduction in Q, by a factor of $\tilde{\tau}_\ell^2$, occurs only when $\mathbf{u}_\ell$ has the form $(0, \alpha)$ for some nonzero α.

A geometric interpretation of reflection and contraction moves is depicted in Figure 5. The plane in each case represents $\mathbf{u}$-space. The first figure shows an elliptical level curve of points $(u_\ell, u_{\ell-1})$ for which $Q = 2$; three particular points on the level curve are labeled as $\mathbf{u}_i$. The second figure shows the image of this level curve following the reflection move (5.7) with $\tilde{\tau} = 1$. Points on the level curve are transformed by a reflection to rotated points on the same level curve; the image points of $\mathbf{u}_i$ are labeled as $\mathbf{u}_i'$. The third figure shows the image of the level curve in the first figure after a Nelder–Mead contraction move (5.7) with $\tilde{\tau} = \frac{1}{2}$. The transformed points are not only rotated, but their Q-values are (except for two points) reduced. The points $\mathbf{u}_2 = (2/\sqrt{3},\, 1/\sqrt{3})$ and $\mathbf{u}_3 = (0,\, 1)$ represent the extreme effects of contraction, since $Q(\mathbf{u}_2') = Q(\mathbf{u}_2)$, and $Q(\mathbf{u}_3') = \frac{1}{4}Q(\mathbf{u}_3)$.

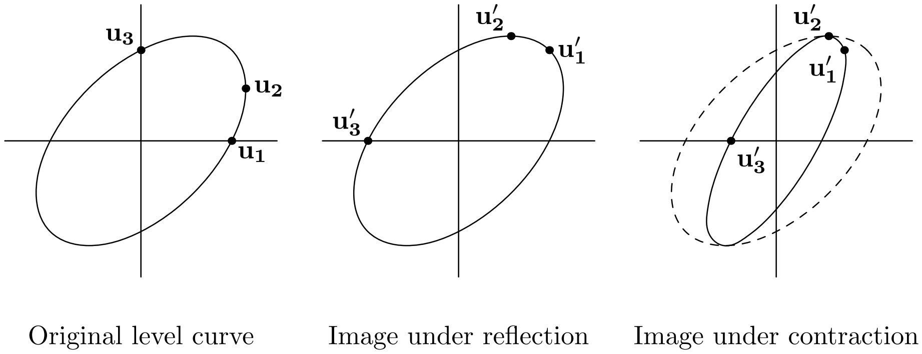


FIG. 5. *The effects of reflection and contraction moves in* $\mathbf{u}$*-space on a level curve of constant* Q.

Our next step is to analyze what can happen to the value of Q following a *sequence* of Nelder–Mead iterations and to show that even in the worst case Q must eventually be driven to zero. Relation (5.17) implies that, for any vector $\mathbf{q}$,

$$\|W_j\mathbf{q}\| \le \|W_k\mathbf{q}\| \quad \text{if} \quad |\tilde{\tau}_j| \le |\tilde{\tau}_k|.$$

In determining upper bounds on Q, we therefore need to consider only the two values $\tilde{\tau}_\ell = 1$ and $\tilde{\tau}_\ell = \frac{1}{2}$ (the latter corresponding to the largest possible value of $|\tilde{\tau}|$ when $\tilde{\tau} \ne 1$).

Using (5.16) repeatedly to move Z to the left, we express a sequence of N Nelder–Mead moves (5.7) starting at iteration ℓ as

$$\boldsymbol{\xi}_{\ell+N} = M_{\ell+N-1}\cdots M_\ell Z\mathbf{q}_\ell = ZW_{\ell+N-1}\cdots W_\ell\mathbf{q}_\ell.$$

Substituting for each W from (5.17), the Euclidean length of $\mathbf{q}_{\ell+N}$ is bounded by

$$\|\mathbf{q}_{\ell+N}\| \;\le\; \|\tilde{Z}\Sigma_{\ell+N-1}\cdots\tilde{Z}\Sigma_\ell\|\,\|\mathbf{q}_\ell\|. \tag{5.19}$$

A relatively straightforward calculation shows that $\|\mathbf{q}_{\ell+N}\|$ is strictly smaller than $\|\mathbf{q}_\ell\|$ after any of the move sequences:

$$\begin{array}{ll} (c,c)\ \text{for } N=2, & (c,1,c)\ \text{for } N=3, \\ (c,1,1,1,c)\ \text{for } N=5, & (c,1,1,1,1,c)\ \text{for } N=6, \end{array} \tag{5.20}$$

where "c" denotes $\tilde{\tau} = \frac{1}{2}$ and "1" denotes $\tilde{\tau} = 1$. For these sequences,

$$\|\mathbf{q}_{\ell+N}\| \;\le\; \beta_{cc}\,\|\mathbf{q}_\ell\|, \quad \text{where } \beta_{cc} \approx 0.7215.$$

(The quantity β_{cc} is the larger root of the quadratic $\lambda^2 + \frac{41}{64}\lambda + \frac{1}{16}$.) Following any of the Nelder–Mead type patterns (5.20), the size measure Q must be decreased by a factor of at least $\beta_{cc}^2 \approx 0.5206$.

(iv) *Illegal patterns of Nelder–Mead move types.* At this point we add the final element of the proof: certain patterns of Nelder–Mead move types cannot occur in the subsequence (5.1). Recall that a new point can be accepted only when its function value is strictly less than the current worst function value. Now consider five consecutive Nelder–Mead iterations (5.7) of types $(1,1,\tilde\tau_3,1,1)$ in which $\mathbf{x}_2$ changes. After such a pattern, the newly accepted vertex is defined by

$$\begin{pmatrix} u_{\ell+5} & v_{\ell+5} \\ u_{\ell+4} & v_{\ell+4} \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ 1 & 0 \end{pmatrix}^2 \begin{pmatrix} \frac{1}{2}(1+\tilde\tau_3) & -\tilde\tau_3 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 0 \end{pmatrix}^2 \begin{pmatrix} u_\ell & v_\ell \\ u_{\ell-1} & v_{\ell-1} \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 \\ -\tilde\tau_3 & \frac{1}{2}(1+\tilde\tau_3) \end{pmatrix} \begin{pmatrix} u_\ell & v_\ell \\ u_{\ell-1} & v_{\ell-1} \end{pmatrix}. \tag{5.21}$$

The first row of this relation gives

$$(u_{\ell+5}, v_{\ell+5}) = (u_{\ell-1}, v_{\ell-1}), \quad \text{so that} \quad \tilde{\mathbf{x}}_2^{(\ell+5)} = \tilde{\mathbf{x}}_3^{(\ell)},$$

which implies the impossible result that the newly accepted vertex is the same as the worst vertex in a previous simplex. Hence the type sequence $(1,1,\tilde\tau_3,1,1)$ cannot occur.

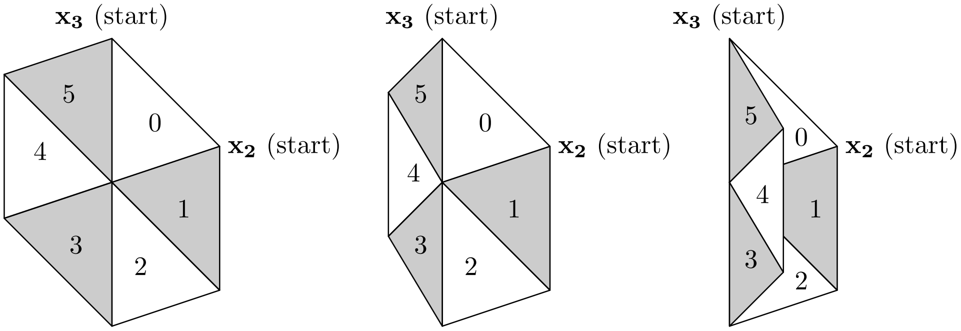


FIG. 6. *Returning to the original worst point with Nelder–Mead type patterns* $(1,1,1,1,1)$, $(1,1,\frac{1}{2},1,1)$, *and* $(1,1,-\frac{1}{2},1,1)$.

Figure 6 depicts these unacceptable move sequences geometrically. From left to right, we see five consecutive reflections; two reflections, an outside contraction, and two further reflections; and two reflections, an inside contraction, and two more reflections.

If we eliminate both the norm-reducing patterns (5.20) and the illegal pattern $(1,1,*,1,1)$, only three valid 6-move sequences remain during which Q might stay unchanged:

$$(1,1,1,1,c,1), \quad (1,c,1,1,1,1), \quad \text{and} \quad (1,c,1,1,c,1).$$

Examination of these three cases shows immediately that *no legal sequence* of 7 steps exists for which Q can remain constant, since the next move creates either a norm-reducing or illegal pattern. In particular, for all legal sequences of 7 steps it holds that

$$\|\mathbf{q}_{\ell+7}\| \ \le \ \beta_{cc}\|\mathbf{q}_\ell\| \ < \ 0.7216\|\mathbf{q}_\ell\|.$$

We conclude that $\|\mathbf{q}_\ell\| \to 0$ and hence, using (5.14), that $Q(\mathbf{u}_\ell) \to 0$, as desired. This completes the proof of Lemma 5.1. $\square$

To finish the proof of Theorem 5.1, we note that, in the case when $\mathbf{x}_1^{(k)}$ eventually becomes constant, the just-completed proof of Lemma 5.1 implies convergence of $\mathbf{x}_2$ and $\mathbf{x}_3$ to $\mathbf{x}_1$, which gives $f_1^* = f_2^* = f_3^*$, as desired. $\square$

5.2. Convergence of simplex diameters to zero. Knowing that the vertex function values converge to a common value does not imply that the vertices themselves converge. We next analyze the evolution of the shapes of the triangles Δ_k produced by the Nelder–Mead algorithm on a strictly convex function in $\mathcal{R}^2$. First, we show that they "collapse" to zero volume, i.e., to either a point or a line segment.

LEMMA 5.2. (Convergence of simplex volumes to zero.) *Assume that f is a strictly convex function on $\mathcal{R}^2$ with bounded level sets and that the Nelder–Mead algorithm with reflection coefficient $\rho = 1$, expansion coefficient $\chi = 2$, and contraction coefficient $\gamma = \frac{1}{2}$ is applied to f beginning with a nondegenerate initial simplex Δ_0. Then the simplices $\{\Delta_k\}$ generated by the algorithm satisfy*

$$\lim_{k\to\infty} \operatorname{vol}(\Delta_k) = 0. \tag{5.22}$$

Proof. We know from Theorem 5.1 that the limiting function values at the vertices are equal, say to f^*. If $f^* = f_{\min}$, then by strict convexity this value is assumed at a unique point, in which case the desired result (5.22) follows immediately and the proof is complete. Furthermore, Lemma 5.1 shows that, if the best vertex $\mathbf{x}_1$ eventually becomes constant, then the remaining two vertices converge to $\mathbf{x}_1$, and (5.22) holds in this case also.

In the rest of the proof we assume that $f^* > f_{\min}$ and that $\mathbf{x}_1$ changes infinitely often. Corresponding to f^*, we define the level set $\mathcal{L}_*$ and its boundary $\mathit{\Gamma}_*$:

$$\mathcal{L}_* = \{\mathbf{x} \mid f(\mathbf{x}) \le f^*\} \quad \text{and} \quad \mathit{\Gamma}_* = \{\mathbf{x} \mid f(\mathbf{x}) = f^*\}. \tag{5.23}$$

It follows from our assumptions about f that $\mathcal{L}_*$ is nonempty, closed, and strictly convex.

The proof is obtained by contradiction. Suppose that (5.22) does not hold, so that

$$\limsup_{k\to\infty} \operatorname{vol}(\Delta_k) > 0. \tag{5.24}$$

We know that all Nelder–Mead simplices Δ_k lie inside the compact level set $\{\mathbf{x} \mid f(\mathbf{x}) \le f(\mathbf{x}_3^{(0)})\}$, and that all vertex function values converge to f^*. Hence we can extract at least one subsequence $\{k_j\}$ of iterations such that the simplices Δ_{k_j} satisfy

$$\lim_{j\to\infty} \Delta_{k_j} = T, \tag{5.25}$$

where T is a triangle of *nonzero volume* whose vertices all lie on $\mathit{\Gamma}_*$.

Next we consider properties of the set $\mathcal{T}_*$ of all triangles T satisfying (5.25) for some subsequence k_j. Since shrink steps cannot occur, a Nelder–Mead iteration on a given triangle is specified by two values: a distinguished (worst) vertex and a move type τ, where τ is one of $(1, 2, \frac{1}{2}, -\frac{1}{2})$. For each sequence k_j satisfying (5.25) with limit triangle T, there is a sequence of pairs of distinguished vertices and move types associated with moving from Δ_{k_j} to the next simplex Δ_{k_j+1}. For any such pair that occurs infinitely often in the sequence of iterations $\{k_j\}$, the vertices of Δ_{k_j+1}, the successor simplices, are a continuous function of the vertices of Δ_{k_j}. Since all limit vertex function values are equal to f^*, so that all limit vertices lie on $\mathit{\Gamma}_*$, there is a subsequence $\{k_{j_i}\}$ of $\{k_j\}$ such that

$$\lim_{i\to\infty} \Delta_{k_{j_i}+1} \to \widetilde{T} \in \mathcal{T}_*.$$

We conclude that for every triangle T in $\mathcal{T}_*$, there is a Nelder–Mead move which, applied to T, yields a new triangle $\widetilde{T}$ in $\mathcal{T}_*$. A similar argument shows that every triangle in $\mathcal{T}_*$ is the result of applying a Nelder–Mead move to another triangle in $\mathcal{T}_*$.

Next we consider sequences of possible Nelder–Mead moves among elements of $\mathcal{T}_*$. Observe first that no move of type $-\frac{1}{2}$ (inside contraction) can occur, since the new vertex would lie inside the convex hull of the original three vertices, contradicting the fact that the original three vertices and the new vertex must lie on $\mathit{\Gamma}_*$.

The volumes of triangles in $\mathcal{T}_*$ are bounded above because all vertices of such triangles lie on the boundary of $\mathcal{L}_*$. Define

$$V = \sup\{\, \mathrm{vol}(T) \mid T \in \mathcal{T}_* \,\}, \tag{5.26}$$

where $V > 0$ because of assumption (5.24), and choose a triangle T' in $\mathcal{T}_*$ for which

$$\mathrm{vol}(T') > \tfrac{1}{2}V. \tag{5.27}$$

Let V_* be the volume of the level set $\mathcal{L}_*$, and define the integer h_* as

$$h_* = 1 + \left\lceil \frac{V_*}{V} \right\rceil. \tag{5.28}$$

Now consider all sequences of h_* *consecutive* simplices produced by the Nelder–Mead algorithm applied to the initial simplex Δ_0,

$$\Delta_{r+1}, \Delta_{r+2}, \ldots, \Delta_{r+h_*}, \tag{5.29}$$

and define a sequence $\{T_i\}$ of h_* triangles in $\mathcal{T}_*$, ending with the triangle T' of (5.27), by extracting a subsequence $\{m_j\}$ for which

$$\lim \Delta_{m_j+i} = T_i \quad \text{for } i = 1, \ldots, h_*, \quad \text{with} \quad T_i \in \mathcal{T}_* \quad \text{and} \quad T_{h_*} = T'. \tag{5.30}$$

During any sequence of consecutive Nelder–Mead moves of type 1 (reflections), volume is preserved (see Lemma 3.1) and all triangles are disjoint; no triangle can be repeated because of the strict decrease requirement on the vertex function values. Suppose that there is a sequence of consecutive reflections in the set of iterations $m_j + 1$, …, $m_j + h_*$; then the associated limiting triangles have disjoint interiors, cannot repeat, and lie inside the curve $\mathit{\Gamma}_*$. Since the volume enclosed by $\mathit{\Gamma}_*$ is V_*, there can be at most $h_* - 1$ consecutive reflections (see (5.28)), and it follows that, for some i, the move from T_i to T_{i+1} is not a reflection.

Consider the first predecessor T_i of T_{h_*} in the sequence (5.30) for which $\mathrm{vol}(T_i) \neq \mathrm{vol}(T_{h_*})$. The Nelder–Mead move associated with moving from T_i to T_{i+1} cannot be a contraction; if it were, then

$$\mathrm{vol}(T_i) = 2\,\mathrm{vol}(T') > V,$$

which is impossible by definition of V (5.26) and T' (5.27). Thus, in order to satisfy (5.30), the move from T_i to T_{i+1} must be an expansion step, i.e., a move of type 2.

We now show that this is impossible because of the strict convexity of $\mathcal{L}_*$ and the logic of the Nelder–Mead algorithm.

For the sequence $\{m_j\}$ of (5.30), the function value at the (accepted) expansion point must satisfy $f(\mathbf{x}_e^{(m_j+i)}) \geq f^*$, since the function values at all vertices converge from above to f^*. The reflection point $\mathbf{r}_i$ for T_i is outside T_i and lies strictly inside the triangle T_{i+1}, all of whose vertices lie on the curve Γ_*. (See Figure 7.) Since the level set $\mathcal{L}_*$ is strictly convex, $f(\mathbf{r}_i)$ must be strictly less than f^*, the value of f on Γ_*, and there must be a small open ball around $\mathbf{r}_i$ within which the values of f are strictly less than f^*.

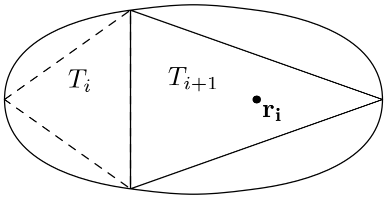


FIG. 7. *Position of the reflection point* $\mathbf{r}_i$ *when the vertices of* T_i *and the expansion point (the new vertex of* T_{i+1}*) lie on the boundary of a bounded strictly convex set.*

The test reflection points $\mathbf{x}_r^{(m_j+i)}$ converge to $\mathbf{r}_i$, and hence eventually $f(\mathbf{x}_r^{(m_j+i)})$ must be strictly less than f^*. It follows that the Nelder–Mead algorithm at step m_j+i could have chosen a new point (the reflection point) with a lower function value than at the expansion point, but failed to do so; this is impossible, since the Nelder–Mead method accepts the better of the reflection and expansion points. (Note that this conclusion would *not* follow for the Nelder–Mead algorithm in the original paper [6], where the expansion point could be chosen as the new vertex even if the value of f at the reflection point were smaller.) Thus we have shown that the assumption $\limsup_{k\to\infty} \operatorname{vol}(\Delta_k) > 0$ leads to a contradiction. This gives the desired result that the Nelder–Mead simplex volumes converge to zero. □

Having shown that the simplex volumes converge to zero, we now prove that the *diameters* converge to zero, so that the Nelder–Mead simplices collapse to a point.

THEOREM 5.2. (Convergence of simplex diameters to zero.) *Let f be a strictly convex function on $\mathcal{R}^2$ with bounded level sets. Assume that the Nelder–Mead algorithm with reflection coefficient $\rho = 1$, expansion coefficient $\chi = 2$, and contraction coefficient $\gamma = \frac{1}{2}$ is applied to f beginning with a nondegenerate initial simplex Δ_0. Then the simplices $\{\Delta_k\}$ generated by the algorithm satisfy*

$$\lim_{k\to\infty} \operatorname{diam}(\Delta_k) = 0. \tag{5.31}$$

Proof. The proof is by contradiction. Lemma 5.2 shows that $\operatorname{vol}(\Delta_k) \to 0$. Since reflection preserves volume, infinitely many nonreflection steps must occur.

Suppose that the conclusion of the theorem is not true, i.e., that $\operatorname{diam}(\Delta_k)$ does not converge to zero. Then we can find a infinite subsequence $\{k_j\}$ for which the associated simplices Δ_{k_j} have diameters bounded away from zero, so that

$$\operatorname{diam}(\Delta_{k_j}) \geq \alpha > 0. \tag{5.32}$$

For each k_j in this subsequence, consider the sequence of iterations k_j, $k_j + 1$, $\ldots$, and let k'_j denote the first iteration in this sequence that immediately precedes a nonreflection step. Then the simplex $\Delta_{k'_j}$ is congruent to Δ_{k_j}, so that $\operatorname{diam}(\Delta_{k'_j}) \geq \alpha$, and a nonreflection step occurs when moving from $\Delta_{k'_j}$ to $\Delta_{k'_j+1}$.

Now we define a subsequence k''_j of k'_j with the following properties:

1. $\Delta_{k''_j}$ converges to a fixed line segment $[\mathbf{v}_0, \mathbf{v}_1]$, with $\mathbf{v}_0 \neq \mathbf{v}_1$ and $\|\mathbf{v}_1 - \mathbf{v}_0\|_2 \geq \alpha$;
2. each Nelder–Mead step from $\Delta_{k''_j}$ to $\Delta_{k''_j+1}$ has the same combination of distinguished (worst) vertex and move type among the nine possible pairs of three vertices and three nonreflection moves.

Note that the vertices of $\Delta_{k''_j+1}$ are continuous functions of the vertices of $\Delta_{k''_j}$ and that the values of f at all vertices of $\Delta_{k''_j+1}$ must converge monotonically from above to f^*.

The points $\mathbf{v}_0$ and $\mathbf{v}_1$ must lie on the boundary of the strictly convex level set $\mathcal{L}_*$ (5.23). If the vertices of $\Delta_{k''_j}$ converge to three distinct points on the line segment $[\mathbf{v}_0, \mathbf{v}_1]$, strict convexity would imply that the function value at the interior point is strictly less than f^*, which is impossible. Thus two of the three vertices must converge to one of $\mathbf{v}_0$ and $\mathbf{v}_1$, which means that two of the vertices of $\Delta_{k''_j}$ will eventually lie close to one of $\mathbf{v}_0$ or $\mathbf{v}_1$. Without loss of generality we assume that two of the vertices are near $\mathbf{v}_0$ and the remaining vertex is near $\mathbf{v}_1$.

To obtain a contradiction, we show that all nonreflection steps are unacceptable.

(i) An inside contraction applied to $\Delta_{k''_j}$ with distinguished vertex near $\mathbf{v}_0$ produces a (limit) vertex for $\Delta_{k''_j+1}$ at $\frac{3}{4}\mathbf{v}_0 + \frac{1}{4}\mathbf{v}_1$; an inside contraction with distinguished vertex near $\mathbf{v}_1$ produces a limit vertex at $\frac{1}{2}\mathbf{v}_0 + \frac{1}{2}\mathbf{v}_1$. In either case, the limit vertex for $\Delta_{k''_j+1}$ lies strictly between $\mathbf{v}_0$ and $\mathbf{v}_1$, giving a function value smaller than f^*, a contradiction.

(ii) An outside contraction applied to $\Delta_{k''_j}$ with distinguished vertex near $\mathbf{v}_0$ produces a limit vertex for $\Delta_{k''_j+1}$ at $\frac{1}{4}\mathbf{v}_0 + \frac{3}{4}\mathbf{v}_1$, giving a contradiction as in (i). With distinguished vertex near $\mathbf{v}_1$, an outside contraction produces a limit vertex at $-\frac{1}{2}\mathbf{v}_1 + \frac{3}{2}\mathbf{v}_0$. Since $\mathbf{v}_0$ and $\mathbf{v}_1$ lie on the boundary of the strictly convex set $\mathcal{L}_*$, this limit vertex point lies outside the level set and hence has function value greater than f^*. This contradicts the fact that the associated vertex function values in $\Delta_{k''_j+1}$ must converge to f^*.

(iii) An expansion step with distinguished vertex near $\mathbf{v}_0$ produces a limit vertex for $\Delta_{k''_j+1}$ at $3\mathbf{v}_1 - 2\mathbf{v}_0$, and an expansion step with distinguished vertex near $\mathbf{v}_1$ produces a limit vertex at $3\mathbf{v}_0 - 2\mathbf{v}_1$. In both cases, the limit vertex lies outside $\mathcal{L}_*$. This means that its function value exceeds f^*, giving a contradiction.

Since a contradiction arises from applying every possible non-reflection move to the simplex $\Delta_{k''_j}$, the sequence k_j of (5.32) cannot exist. Thus we have shown that $\lim \operatorname{diam}(\Delta_k) \to 0$, namely that each Nelder–Mead simplex eventually collapses to a point. □

Note that this theorem does *not* imply that the sequence of simplices $\{\Delta_k\}$ converges to a limit point $\mathbf{x}_*$. We do know, however, that all vertices converge to $\mathbf{x}_1$ if this vertex remains constant (see Lemma 5.1); this situation occurs in the McKinnon examples [5].

6. Conclusions and open questions. In dimension 1, the generic Nelder–Mead method converges to the minimizer of a strictly convex function with bounded level sets if and only if the expansion step is a genuine expansion (i.e., if $\rho\chi \geq 1$).

It is interesting that, apart from this further requirement, the conditions (2.1) given in the original Nelder–Mead paper suffice to ensure convergence in one dimension. The behavior of the algorithm in dimension 1 can nonetheless be very complicated; for example, there can be an infinite number of expansions even when convergence is M-step linear (Theorem 4.2).

In two dimensions, the behavior of even the standard Nelder–Mead method (with $\rho = 1$, $\chi = 2$, and $\gamma = \frac{1}{2}$) is more difficult to analyze for two reasons:

1. The space of simplex shapes is not compact, where the *shape* of a simplex is its similarity class; see the discussion at the end of section 2. It appears that the Nelder–Mead moves are dense in this space, i.e., any simplex can be transformed by some sequence of Nelder–Mead moves to be arbitrarily close to any other simplex shape; this property reflects the intent expressed by Nelder and Mead [6] that the simplex shape should "adapt itself to the local landscape." This contrasts strongly with the nature of many pattern search methods [13], in which the simplex shapes remain constant.

2. The presence of the expansion step means that vol(Δ) is not a Lyapunov function[3] for the iteration.

The two-dimensional results proved in section 5 seem very weak but conceivably represent the limits of what can be proved for arbitrary strictly convex functions. In particular, Theorem 5.2 leaves open the possibility that the ever-smaller simplices endlessly "circle" the contour line $f(x) = f^*$. Since no examples of this behavior are known, it may be possible to prove the stronger result that the simplices always converge to a *single* point $\mathbf{x}_*$.

An obvious question concerns *how* the Nelder–Mead method can fail to converge to a minimizer in the two-dimensional case. Further analysis suggests that, for suitable strictly convex functions (C^1 seems to suffice), failure can occur only if the simplices elongate indefinitely and their shape goes to "infinity" in the space of simplex shapes (as in the McKinnon counterexample).

An interesting open problem concerns whether there exists *any* function $f(\mathbf{x})$ in $\mathcal{R}^2$ for which the Nelder–Mead algorithm always converges to a minimizer. The natural candidate is $f(x, y) = x^2 + y^2$, which by affine-invariance is equivalent to all strictly convex quadratic functions in two dimensions. A complete analysis of Nelder–Mead for $x^2 + y^2$ remains an open problem.

Our general conclusion about the Nelder–Mead algorithm is that the main mystery to be solved is not whether it ultimately converges to a minimizer—for general (nonconvex) functions, it does not—but rather why it tends to work so well in practice by producing a rapid initial decrease in function values.

Acknowledgments. The authors greatly appreciate the referee's detailed and constructive comments, which helped us to improve the content and presentation of the paper. We also thank Virginia Torczon for making us aware of the connections described in section 4.4. Margaret Wright is extremely grateful to Steve Fortune for many interesting discussions and his consistently enlightening geometric insights.

REFERENCES

[1] J. E. Dennis and V. Torczon, *Direct search methods on parallel machines*, SIAM J. Optim., 1 (1991), 448–474.

[3] See the discussion of Lyapunov functions in, for example, [11, pp. 23–27] in the context of stability of nonlinear fixed points.

[2] C. T. Kelley, *Detection and Remediation of Stagnation in the Nelder-Mead Algorithm Using a Sufficient Decrease Condition*, Technical report, Department of Mathematics, North Carolina State University, Raleigh, NC, 1997.

[3] J. C. Lagarias, B. Poonen, and M. H. Wright, *Convergence of the restricted Nelder-Mead algorithm in two dimensions*, in preparation, 1998.

[4] Math Works, Matlab, The Math Works, Natick, MA, 1994.

[5] K. I. M. McKinnon, *Convergence of the Nelder-Mead simplex method to a nonstationary point*, SIAM J. Optim., 9 (1998), 148–158.

[6] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Computer Journal 7 (1965), 308–313.

[7] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vettering, *Numerical Recipes in C*, Cambridge University Press, Cambridge, UK, 1988.

[8] A. Rykov, *Simplex direct search algorithms*, Automation and Robot Control, 41 (1980), 784–793.

[9] A. Rykov, *Simplex methods of direct search*, Engineering Cybernetics, 18 (1980), 12–18.

[10] A. Rykov, *Simplex algorithms for unconstrained optimization*, Problems of Control and Information Theory, 12 (1983), 195–208.

[11] A. M. Stuart and A. R. Humphries, *Dynamical Systems and Numerical Analysis*, Cambridge University Press, New York, 1996.

[12] V. Torczon, *Multi-directional Search: A Direct Search Algorithm for Parallel Machines*, Ph.D. thesis, Rice University, Houston, TX, 1989.

[13] V. Torczon, *On the convergence of pattern search algorithms*, SIAM J. Optim., 7 (1997), 1–25.

[14] V. Torczon, *Private communication*, 1997.

[15] P. Tseng, *Fortified-Descent Simplicial Search Method: A General Approach*, Technical report, Department of Mathematics, University of Washington, Seattle, WA, 1995; SIAM J. Optim., submitted.

[16] F. H. Walters, L. R. Parker, S. L. Morgan, and S. N. Deming, *Sequential Simplex Optimization*, CRC Press, Boca Raton, FL, 1991.

[17] D. J. Woods, *An Interactive Approach for Solving Multi-objective Optimization Problems*, Ph.D. thesis, Rice University, Houston, TX, 1985.

[18] M. H. Wright, *Direct search methods: Once scorned, now respectable*, in Numerical Analysis 1995: Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis, D. F. Griffiths and G. A. Watson, eds., Addison Wesley Longman, Harlow, UK, 1996, 191–208.

SIAM J. OPTIM.
Vol. 9, No. 1, pp. 148–158

CONVERGENCE OF THE NELDER–MEAD SIMPLEX METHOD TO A NONSTATIONARY POINT*

K. I. M. MCKINNON†

Abstract. This paper analyzes the behavior of the Nelder–Mead simplex method for a family of examples which cause the method to converge to a nonstationary point. All the examples use continuous functions of two variables. The family of functions contains strictly convex functions with up to three continuous derivatives. In all the examples the method repeatedly applies the inside contraction step with the best vertex remaining fixed. The simplices tend to a straight line which is orthogonal to the steepest descent direction. It is shown that this behavior cannot occur for functions with more than three continuous derivatives. The stability of the examples is analyzed.

Key words. Nelder–Mead method, direct search, simplex, unconstrained optimization

AMS subject classification. 65K05

PII. S1052623496303482

1. Introduction.

Direct search methods are very widely used in chemical engineering, chemistry, and medicine. They are a class of optimization methods which are easy to program, do not require derivatives, and are often claimed to be robust for problems with discontinuities or where the function values are noisy. In [12, 13] Torczon produced convergence results for a class of methods called pattern search methods. This class includes several well-known direct search methods such as the two-dimensional case of the Spendley, Hext, and Himsworth simplex method [8] but does not include the most widely used method, the Nelder–Mead simplex method [4]. In the Nelder–Mead method the simplex can vary in shape from iteration to iteration. Nelder and Mead introduced this feature to allow the simplex to adapt its shape to the local contours of the function, and for many problems this is effective. However, it is this change of shape which excludes the Nelder–Mead method from the class of methods covered by the convergence results of Torczon [13], which rely on the vertices of the simplices lying on a lattice of points.

The Nelder–Mead method uses a small number of function evaluations per iteration, and for many functions of low dimension its rules for adapting the simplex shape lead to low iteration counts. In [11, 1], however, Torczon and Dennis report results from tests in which the Nelder–Mead method frequently failed to converge to a local minimum of smooth functions of low dimension: it was observed even for functions with as few as eight variables. In the cases where failure occurred, the search line defined by the method became orthogonal to the gradient direction; however, the reasons for this behavior were not fully understood. Some theoretical results about the convergence of a modified version of the Nelder–Mead method are given by Woods [15]. In a recent paper, Lagarias et al. [3] derive a range of convergence results which apply to the original Nelder–Mead method. Among these results is a proof that the method converges to a minimizer for strictly convex functions of one variable and also a proof that for strictly convex functions of two variables the simplex diameters converge to zero. However, it is not yet known even for the function $x^2 + y^2$, the sim-

*Received by the editors May 13, 1996; accepted for publication (in revised form) November 12, 1997; published electronically December 2, 1998.
http://www.siam.org/journals/siopt/9-1/30348.html

†Department of Mathematics and Statistics, The University of Edinburgh, King's Buildings, Edinburgh EH9 3JZ, UK (ken@maths.ed.ac.uk).

plest strictly convex quadratic functions of two variables, whether the method always converges to the minimizer, or indeed whether it always converges to a single point.

The current paper presents a family of examples of functions of two variables, where convergence occurs to a nonstationary point for a range of starting simplices. Some examples have a discontinuous first derivative and others are strictly convex with between one and three continuous derivatives. The simplices converge to a line which is orthogonal to the steepest descent direction and have interior angles which tend to zero.

We assume that the problem to be solved is

$$\min_{v \in \mathbb{R}^2} f(v).$$

For functions defined over $\mathbb{R}^2$ (i.e., functions of two variables) the Nelder–Mead method operates with a simplex in $\mathbb{R}^2$, which is specified by its three vertices. The Nelder–Mead method is described below for the two-variable case and without the termination test. The settings for the parameter ρ in $L(\rho)$ are the most commonly used values. A fuller description of the method can be found in the papers by Lagarias et al. [3] and Nelder and Mead [4].

The Nelder–Mead method.

```
ORDER: Label the three vertices of the current simplex $b$, $s$, and $w$ so that their
    corresponding function values $f_b$, $f_s$, and $f_w$ satisfy $f_b \leq f_s \leq f_w$.
    $m := (b+s)/2$, {the midpoint of the best and second worst points}.
    Let $L(\rho)$ denote the function $L(\rho) = m + \rho(m - w)$, {$L$ is the search line}.
    $r := L(1)$; $f_r := f(r)$.
    If $f_r < f_b$ then
        $e := L(2)$; $f_e := f(e)$.
        If $f_e < f_b$ then accept $e$ {Expand} else accept $r$ {Reflect}.
    else {$f_b \leq f_r$} if $f_r < f_s$ then
        Accept $r$ {Reflect}.
    else {$f_s \leq f_r$} if $f_r < f_w$ then
        $c := L(0.5)$; $f_c := f(c)$.
        If $f_c \leq f_r$ then accept $c$ {Outside Contract} else → SHRINK.
    else {$f_w \leq f_r$}
        $c := L(-0.5)$; $f_c := f(c)$.
        If $f_c < f_w$ then accept $c$ {Inside Contract} else → SHRINK.
    Replace $w$ by the accepted point; → ORDER.
SHRINK: Replace $s$ by $(s+b)/2$ and $w$ by $(w+b)/2$; → ORDER.
```

The examples in this paper cause the Nelder–Mead method to apply the inside contraction step repeatedly with the best vertex remaining fixed. This behavior will be referred to as repeated focused inside contraction (RFIC). No other type of step occurs for these examples, and this greatly simplifies their analysis. The examples are very simple and highlight a serious deficiency in the method: the simplices collapse along the steepest descent direction, a direction along which we would like them to enlarge.

It should be noted that it is now common to use a variant of the original Nelder–Mead algorithm in which the expand step is accepted if $f_e < f_r$, which is a more restrictive condition. Since the examples in this paper are constructed so that $f_r > f_b$, i.e., the reflected point is never an improvement on the best point, the expand step is never considered. Hence this common variant of the Nelder–Mead method behaves in an identical manner to the original algorithm for the examples in this paper.

Other examples are known where the Nelder–Mead method or its variants fail. In [2], Dennis and Woods give a strictly convex example, where a variant of the Nelder–Mead method performs an unbroken sequence of shrink steps toward a single point which is at a discontinuity of the gradient and at which there is no zero subgradient. In their variant the condition for accepting a contraction step is that $f_c < f_s$, which is more stringent than the original Nelder–Mead method, so more shrink steps are performed. This behavior cannot occur for the original version of the Nelder–Mead method as this method never performs shrink steps on strictly convex functions (see Lagarias et al. [3]). In [15] Woods also gives a sketch of a differentiable nonconvex function for which the Nelder–Mead method converges to a nonminimizing point by a sequence of repeated shrinks. However, it can be shown that for this behavior to occur with the original form of the Nelder–Mead method, the point to which the simplex shrinks must be a stationary point. It is also possible to construct examples of nonconvex differentiable functions for which the original form of the Nelder–Mead method in exact arithmetic converges by repeated contractions to a degenerate simplex of finite length, none of whose vertices are stationary points [9, 10]. An example of this case is the function $f(x, y) = x^2 - y(y-2)$ with initial simplex (1,0), (0,-3), (0,3), which tends in the limit to (0,0), (0,-3), (0,3). The examples given in this paper are, however, the first examples known where the Nelder–Mead method fails to converge to a minimizer of a strictly convex differentiable function.

A wide variety of simplex methods which allow the simplex to vary in shape in a similar manner to the Nelder–Mead method has been proposed and analyzed by, among others, Rykov [5, 6, 7] and more recently by Tseng [14]. These methods accept certain trial steps only if there is a sufficient decrease in an objective function. In this they differ from the Nelder–Mead method and the methods of Torczon [12] which require only strict decrease and whose behavior depends only on the order of the function values at the trial points, not on the actual values. Convergence results for the methods of Rykov and Tseng rely on this sufficient decrease. One of the variants of Tseng's method is the same as the Nelder–Mead method except for the sufficient decrease condition and a condition which bounds the simplex interior angles away from zero. Because of this, when Tseng's variant is applied to the examples in this paper, it eventually performs shrink steps instead of the inside contraction steps performed by the original Nelder–Mead method. This allows it to escape from the nonstationary point which is the focus of the RFIC in the original Nelder–Mead method.

The structure of this paper is as follows. In section 2 the sequence of simplices is derived corresponding to RFIC. In section 3 a family of functions are given which produce this behavior and result in the method converging to a nonstationary point. In section 4 the range of functions which can give the RFIC behavior is derived. Section 5 contains an analysis of how perturbations of the initial simplex affect the RFIC behavior of the examples in section 3.

2. Analysis of the repeated inside contraction behavior. Consider a simplex in two dimensions with vertices at 0 (i.e., the origin), $v^{(n+1)}$, and $v^{(n)}$. Assume that

$$f(0) < f(v^{(n+1)}) < f(v^{(n)}). \tag{2.1}$$

After the ORDER step of the algorithm, $b = 0$, $s = v^{(n+1)}$, and $w = v^{(n)}$. The Nelder–Mead method calculates $m^{(n)} = v^{(n+1)}/2$, the midpoint of the line joining the best and second worst points, and then reflects the worst point, $v^{(n)}$, in $m^{(n)}$ with a reflection factor of $\rho = 1$ to give the point

$$r^{(n)} = m^{(n)} + \rho(m^{(n)} - v^{(n)}) = v^{(n+1)} - v^{(n)}. \tag{2.2}$$

Assume that

$$f(v^{(n)}) < f(r^{(n)}). \tag{2.3}$$

In this case the point $r^{(n)}$ is rejected and the point $v^{(n+2)}$ is calculated using a reflection factor $\rho = -0.5$ in

$$v^{(n+2)} = m^{(n)} + \rho(m^{(n)} - v^{(n)}) = \frac{1}{4}v^{(n+1)} + \frac{1}{2}v^{(n)}.$$

$v^{(n+2)}$ is the midpoint of the line joining $m^{(n)}$ and $v^{(n)}$. Provided $f(v^{(n+2)}) < f(v^{(n+1)})$, i.e., (2.1) holds with n replaced by $n+1$, the Nelder–Mead method does the inside contraction step rather than a shrink step. The inside contraction step replaces $v^{(n)}$ with the point $v^{(n+2)}$, so that the new simplex consists of $v^{(n+1)}$, $v^{(n+2)}$, and the origin. Provided this pattern repeats, the successive simplex vertices will satisfy the linear recurrence relation

$$4v^{(n+2)} - v^{(n+1)} - 2v^{(n)} = 0.$$

This has the general solution

$$v^{(n)} = A_1\lambda_1^n + A_2\lambda_2^n, \tag{2.4}$$

where $A_i \in \mathbb{R}^2$ and

$$\lambda_1 = \frac{1+\sqrt{33}}{8}, \qquad \lambda_2 = \frac{1-\sqrt{33}}{8}. \tag{2.5}$$

Hence $\lambda_1 \cong 0.84307$ and $\lambda_2 \cong -0.59307$. It follows from (2.2) and (2.4) that

$$r^{(n)} = -A_1\lambda_1^n(1-\lambda_1) - A_2\lambda_2^n(1-\lambda_2). \tag{2.6}$$

It is this repeated inside contraction toward the same fixed vertex which is being referred to as repeated focused inside contraction (RFIC). In [3] Lagarias et al. formally prove that no step of the Nelder–Mead method can transform a nondegenerate simplex to a degenerate simplex. In the two-dimensional case this corresponds to the fact that the area of the simplex either increases by a factor of 2, stays the same, or decreases by a factor of 2 or 4. Hence, provided the Nelder–Mead method is started from a nondegenerate initial simplex, then no later simplex can be degenerate and if RFIC occurs, then the initial simplex for RFIC is nondegenerate. This implies that A_1 and A_2 in (2.4) are linearly independent.

Consider now the initial simplex with vertices $v^{(0)} = (1,1), v^{(1)} = (\lambda_1, \lambda_2)$, and $(0,0)$. Substituting into (2.4) yields $A_1 = (1,0)$ and $A_2 = (0,1)$. It follows that the particular solution for these initial conditions is $v^{(n)} = (\lambda_1^n, \lambda_2^n)$. This solution is

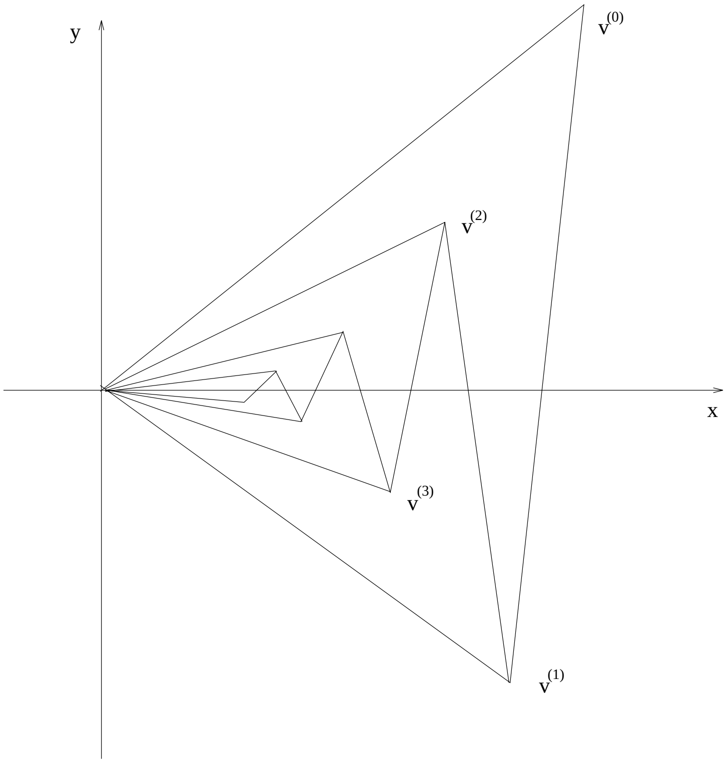


FIG. 2.1. *Successive simplices with* RFICs.

shown in Figure 2.1. The general form of the three points needed at one step of the Nelder–Mead method is therefore

$$v^{(n)} = (\lambda_1^n, \lambda_2^n), \tag{2.7}$$

$$v^{(n+1)} = (\lambda_1^{n+1}, \lambda_2^{n+1}), \tag{2.8}$$

$$r^{(n)} = (-\lambda_1^n(1-\lambda_1), -\lambda_2^n(1-\lambda_2)). \tag{2.9}$$

Provided (2.1) and (2.3) hold at these points, the simplex method will take the inside contraction step assumed above.

Note that the x coordinates of $v^{(n)}$ and $v^{(n+1)}$ are positive and the x coordinate of $r^{(n)}$ is negative.

3. Functions which cause RFIC. Consider the function $f(x, y)$ given by

$$\begin{aligned} f(x,y) &= \theta\phi|x|^\tau + y + y^2, \quad x \le 0, \\ &= \quad \theta x^\tau + y + y^2, \quad x \ge 0, \end{aligned} \tag{3.1}$$

where θ and ϕ are positive constants. Note that (0,-1) is a descent direction from the origin (0,0) and that f is strictly convex provided $\tau > 1$. f has continuous first derivatives if $\tau > 1$, continuous second derivatives if $\tau > 2$, and continuous third derivatives if $\tau > 3$. Figure 2.2 shows the contour plot of this function and the first two steps of the Nelder–Mead method for the case $\tau = 2$, $\theta = 6$, and $\phi = 60$. Both steps are inside contractions.

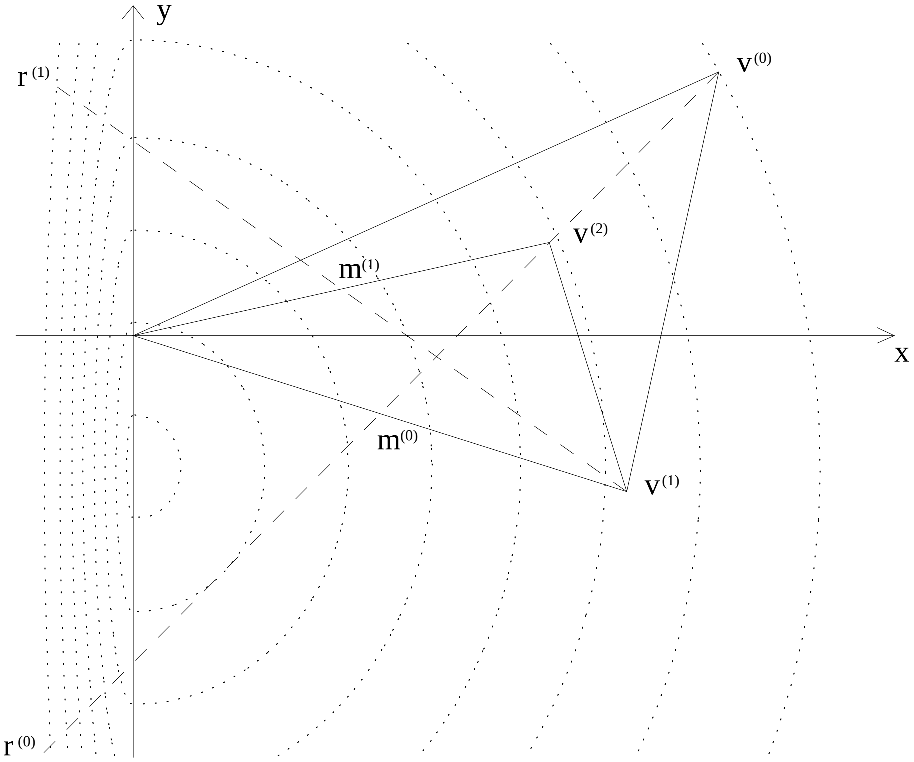


FIG. 2.2. $f(x,y) = 360x^2 + y + y^2$ *if* $x \leq 0$ *and* $f(x,y) = 6x^2 + y + y^2$ *if* $x \geq 0$, *i.e., function* (3.1) *for case* $\tau = 2,\ \theta = 6,\ \phi = 60$.

Define $\hat{\tau}$ to be such that

$$\lambda_1^{\hat{\tau}} = |\lambda_2|, \tag{3.2}$$

so $\hat{\tau}$ is given by

$$\hat{\tau} = \frac{\ln|\lambda_2|}{\ln\lambda_1} \cong 3.0605. \tag{3.3}$$

In what follows assume that τ satisfies

$$0 < \tau < \hat{\tau}. \tag{3.4}$$

Since $0 < \lambda_1 < 1$, it therefore follows that

$$\lambda_1^{\tau} > \lambda_1^{\hat{\tau}} = |\lambda_2|. \tag{3.5}$$

Using (2.7) and (2.9) it follows that

$$\begin{aligned} f(v^{(n)}) &= \theta\lambda_1^{\tau n} + \lambda_2^n + \lambda_2^{2n} \\ \text{and } f(r^{(n)}) &= \phi\theta(\lambda_1^{\tau n}(1-\lambda_1)^{\tau}) - \lambda_2^n(1-\lambda_2) + \lambda_2^{2n}(1-\lambda_2)^2. \end{aligned}$$

Hence $f(v^{(n)}) > f(v^{(n+1)})$ iff

$$\theta\lambda_1^{\tau n}(1-\lambda_1^{\tau}) > \lambda_2^n(\lambda_2 - 1) + \lambda_2^{2n}(\lambda_2^2 - 1).$$

Since $\lambda_1^\tau > |\lambda_2|$ and $\lambda_2^2 - 1 < 0$, this is true for all $n \geq 0$ if θ is such that

$$\theta(1 - \lambda_1^\tau) > |\lambda_2 - 1|. \tag{3.6}$$

Also $f(v^{(n+1)}) > f(0)$ iff

$$\theta\lambda_1^{\tau(n+1)} + \lambda_2^{n+1} + \lambda_2^{2(n+1)} > 0.$$

Since $\lambda_1^\tau > |\lambda_2|$, this is true for all $n \geq 0$ if

$$\theta > 1. \tag{3.7}$$

Also $f(r^{(n)}) > f(v^{(n)})$ iff

$$\begin{aligned} &\phi\theta(\lambda_1^{\tau n}(1 - \lambda_1)^\tau) - \lambda_2^n(1 - \lambda_2) + \lambda_2^{2n}(1 - \lambda_2)^2 > \theta\lambda_1^{\tau n} + \lambda_2^n + \lambda_2^{2n}, \\ \iff\ &\theta\lambda_1^{\tau n}(\phi(1 - \lambda_1)^\tau - 1) > \lambda_2^n(2 - \lambda_2) - \lambda_2^{2n}((1 - \lambda_2)^2 - 1). \end{aligned}$$

Since $\lambda_2 < 0$ and $\lambda_1^\tau > |\lambda_2|$, this is true for all $n \geq 0$ if θ and ϕ are such that

$$\theta(\phi(1 - \lambda_1)^\tau - 1) > (2 - \lambda_2). \tag{3.8}$$

For any τ in the range given by (3.4), θ can be chosen so that (3.6) and (3.7) hold and then ϕ can be chosen so that (3.8) holds. It then follows that (2.1) and (2.3) will hold, so the inside contraction step will be taken at every iteration and the simplices will be as derived in section 2. The method will therefore converge to the origin, which is not a stationary point. Examples of values of θ and ϕ which make (3.6), (3.7), and (3.8) hold are as follows: for $\tau = 1$, $\theta = 15$ and ϕ=10; for $\tau = 2$, $\theta = 6$ and $\phi = 60$; for $\tau = 3$, $\theta = 6$ and $\phi = 400$.

4. Necessary conditions for RFIC to occur.

In this section we will derive necessary conditions for RFIC to occur. For notational convenience the results are given for RFIC with the origin as focus, but by change of origin they can be applied to any point.

It follows from the description of the algorithm that a necessary condition for RFIC to occur is

$$f_0 = f(0) \leq f(v^{(n+1)}) \leq f(v^{(n)}) \leq f(r^{(n)}). \tag{4.1}$$

(The examples in section 3 satisfy the strict form of the (4.1) relations as given in (2.1) and (2.3).)

If f is s times differentiable at the origin, then f can be written in the form $f(v) = p_s(v) + o(\|v\|^s)$, where p_s is a polynomial of degree at most s, and $D^i f(0) = D^i p_s(0)$ for $i = 0, ..., s$, i.e., the derivatives of f and p_s agree. Making a change of variable to z-space using $v = A_1 z_1 + A_2 z_2$, f and p_s can be viewed as functions of $(z_1, z_2) \in \mathbb{R}^2$. When the necessary derivatives exist, define

$$f_0 = f(0), \quad g_i = \frac{\partial f}{\partial z_i}(0), \quad h = \frac{1}{2}\frac{\partial^2 f}{\partial z_1^2}(0), \text{ and } k = \frac{1}{6}\frac{\partial^3 f}{\partial z_1^3}(0).$$

Then (g_1, g_2) is the gradient of f in z-space, and g_i, h, and k are the z_i, z_1^2, and z_1^3 coefficients in the Taylor expansion of f in z-space. Since $|\lambda_2| < \lambda_1$ and (2.4) holds, $\|v^{(n)}\| = O(\lambda_1^n)$, so

$$f(v^{(n)}) = p_s(v^{(n)}) + o(\lambda_1^{sn}). \tag{4.2}$$

THEOREM 4.1. *If the origin is the focus of repeated inside contraction starting from a simplex with limiting direction A_1, then*

(a) *if f is differentiable at the origin, then $g_1 = 0$;*

(b) *if f is 2 times differentiable at the origin, then $h = 0$;*

(c) *if f is 3 times differentiable at the origin, then $k = 0$.*

Proof. (a) From (4.1) it follows that a necessary condition for RFIC to occur is that $f_0 \le f(v^{(n)})$ and $f_0 \le f(r^{(n)})$. This is true iff

$$\begin{aligned} & f_0 \le f_0 + g_1\lambda_1^n + g_2\lambda_2^n + o(\lambda_1^n), \\ \text{and} \quad & f_0 \le f_0 - g_1\lambda_1^n(1-\lambda_1) - g_2\lambda_2^n(1-\lambda_2) + o(\lambda_1^n). \end{aligned}$$

Since $|\lambda_2| < \lambda_1 < 1$, this cannot occur for all n unless $g_1 = 0$.

(b) Since f is 2 times differentiable at the origin, part (a) holds, so $g_1 = 0$. Hence $p_2(v^{(n)}) - (f_0 + g_2\lambda_2^n + h\lambda_1^{2n}) = O(|\lambda_1\lambda_2|^n) = o(\lambda_1^{4n})$, since $|\lambda_2| < \lambda_1^3$. From this and (4.2) it follows that

$$f(v^{(n)}) = f_0 + g_2\lambda_2^n + h\lambda_1^{2n} + o(\lambda_1^{2n}).$$

From (4.1) it follows that a necessary condition for RFIC to occur is that $f_0 \le f(v^{(n)})$ and $f(v^{(n)}) \le f(r^{(n)})$. This is true iff

$$\begin{aligned} & f_0 \le f_0 + g_2\lambda_2^n + h\lambda_1^{2n} + o(\lambda_1^{2n}) \\ \text{and} \quad & 0 \le -g_2\lambda_2^n(2-\lambda_2) - h\lambda_1^{2n+1}(2-\lambda_1) + o(\lambda_1^{2n}). \end{aligned}$$

Since $|\lambda_2| < \lambda_1^2 < 1$, this cannot occur for all n unless $h = 0$.

(c) Since f is 3 times differentiable at the origin, parts (a) and (b) hold, so $g_1 = 0$ and $h = 0$. Hence $p_3(v^{(n)}) - (f_0 + g_2\lambda_2^n + k\lambda_1^{3n}) = O(|\lambda_1\lambda_2|^n) = o(\lambda_1^{4n})$. From this and (4.2) it follows that

$$f(v^{(n)}) = f_0 + g_2\lambda_2^n + k\lambda_1^{3n} + o(\lambda_1^{3n}).$$

From (4.1) it follows that a necessary condition for RFIC to occur is that $f_0 \le f(v^{(n)})$ and $f_0 \le f(r^{(n)})$. This is true iff

$$\begin{aligned} & f_0 \le f_0 + g_2\lambda_2^n + k\lambda_1^{3n} + o(\lambda_1^{3n}) \\ \text{and} \quad & f_0 \le f_0 - g_2\lambda_2^n(1-\lambda_2) - k\lambda_1^{3n}(1-\lambda_1)^3 + o(\lambda_1^{3n}). \end{aligned}$$

Since $\lambda_1^3 > |\lambda_2|$, this cannot occur for all n unless $k = 0$. □

THEOREM 4.2. *If f has a nonzero gradient at the origin and in a neighborhood of the origin can be expressed in the form*

$$f(v) = p_4(v) + o(\|v\|^{\hat{\tau}}), \tag{4.3}$$

where p_4 is at least 4 times differentiable at the origin, and if the initial simplex is not degenerate, then the origin cannot be the focus of repeated inside contractions.

Proof. Assume that the origin is the focus of repeated contractions.

The first three derivatives of f and p_4 at the origin are the same. Theorem 4.1 shows that $g_1 = h = k = 0$. Hence $p_4(v^{(n)}) - (f_0 + g_2\lambda_2^n) = O(|\lambda_1\lambda_2|^n) = o(\lambda^{4n})$. Since $\hat{\tau} < 4$ and $o(\|v^{(n)}\|^{\hat{\tau}}) = o(\lambda_1^{\hat{\tau}n})$ and $\lambda_1^{\hat{\tau}} = |\lambda_2|$ (by the definition of $\hat{\tau}$), it follows that

$$f(v^{(n)}) = f_0 + g_2\lambda_2^n + o(|\lambda_2|^n).$$

From (4.1) it follows that a necessary condition for RFIC to occur is that $f_0 \leq f(v^{(n)})$ and $f_0 \leq f(v^{(n+1)})$. Since $\lambda_2 < 0$, this cannot occur for all n unless $g_2 = 0$. However, since a condition of the theorem is that the gradient is nonzero at the origin and since $g_1 = 0$, it is not possible that $g_2 = 0$. This contradicts the original assumption and so proves that the origin cannot be the focus of repeated contractions. □

Theorem 4.2 shows that RFIC cannot occur for sufficiently smooth functions, the limit being slightly more than 3 times differentiable. The examples in section 3 show that if the conditions of Theorem 4.2 do not hold, then RFIC is possible.

5. Perturbations of the initial simplex. In this section the behavior of the examples is analyzed for perturbations of the starting simplex. The perturbed position for the vertex at the origin must be on the y axis; otherwise the contracting simplex will eventually lie within a region where all derivatives of the function exist, and Theorems 4.1 and 4.2 show that a nonstationary point cannot be the focus of RFIC in such a region. Also if $\tau > 1$, the gradient exists where $x = 0$ and its direction is parallel to the y axis. It follows from Theorem 4.1 that the only initial simplices which can yield RFIC are those with the dominant eigenvector A_1 perpendicular to the y axis. We therefore consider only perturbations where the vertex at the origin is perturbed to $(0, y_0)$ giving the general form

$$v^{(n)} = \begin{bmatrix} 0 \\ y_0 \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \lambda_1^n + \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \lambda_2^n, \tag{5.1}$$

and when $\tau > 1$ we take $y_1 = 0$. The reflected point is then given by

$$r^{(n)} = \begin{bmatrix} 0 \\ y_0 \end{bmatrix} - \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \lambda_1^n (1 - \lambda_1) - \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \lambda_2^n (1 - \lambda_2). \tag{5.2}$$

We are considering y_0, $x_1 - 1$, y_1, x_2, and $y_2 - 1$ to be close to zero.

Repeating the analysis of section 3 gives $f(v^{(n)}) > f(v^{(n+1)})$ iff

$$\begin{aligned} &\theta \lambda_1^{\tau n} x_1^\tau \left(\left(1 + \frac{x_2}{x_1} \left(\frac{\lambda_2}{\lambda_1} \right)^n \right)^\tau - \left(1 + \frac{x_2}{x_1} \left(\frac{\lambda_2}{\lambda_1} \right)^{n+1} \right)^\tau \lambda_1^\tau \right) \\ &+ \lambda_1^n (1 - \lambda_1) y_1 (1 + 2y_0 + \lambda_1^n (1 + \lambda_1) y_1 + \lambda_2^n (1 + \lambda_2) y_2) \\ &> \lambda_2^n (1 - \lambda_2) y_2 (1 + 2y_0 + \lambda_1^n (1 + \lambda_1) y_1) + \lambda_2^{2n} (\lambda_2^2 - 1) y_2^2. \end{aligned}$$

Also $f(v^{(n)}) > f(0, y_0)$ iff

$$\begin{aligned} &\theta \lambda_1^{\tau(n+1)} x_1^\tau \left(1 + \frac{x_2}{x_1} \left(\frac{\lambda_2}{\lambda_1} \right)^{n+1} \right)^\tau + y_1 \lambda_1^{n+1} (1 + 2y_0 + y_1 \lambda_1^{n+1} + y_2 \lambda_2^{n+1}) \\ &+ y_2 \lambda_2^{n+1} (1 + 2y_0 + y_1 \lambda_1^{n+1}) + y_2^2 \lambda_2^{n+1} > 0. \end{aligned} \tag{5.3}$$

Note that for $x_1 - 1$ and x_2 sufficiently close to zero, the x coordinate of $r^{(n)}$ is negative, so the negative x case for the form of f holds. Hence $f(r^{(n)}) > f(v^{(n)})$ iff

$$\begin{aligned} &\theta \lambda_1^{\tau n} x_1^\tau \left(\phi \left(1 - \lambda_1 - \frac{x_2}{x_1} \left(\frac{\lambda_2}{\lambda_1} \right)^n (1 - \lambda_2) \right)^\tau - \left(1 + \frac{x_2}{x_1} \left(\frac{\lambda_2}{\lambda_1} \right)^n \right)^\tau \right) \\ &- y_1 \lambda_1^n (2 - \lambda_1)(1 + 2y_0 + y_1 \lambda_1^{n+1} + y_2 \lambda_2^{n+1}) \\ &> y_2 \lambda_2^n (2 - \lambda_2)(1 + 2y_0 + y_1 \lambda_1^{n+1}) + y_2^2 \lambda_2^n (2 - \lambda_2). \end{aligned}$$

Since the corresponding inequalities are strict in section 3 and all the functions are continuous, it follows that there exists a symmetric neighborhood of $y_0 = 0$, $x_1 = 1$, $y_1 = 0$, $x_2 = 0$, and $y_2 = 1$ in which the above three relations hold for $n = 0$. Since $|\lambda_1| < 1$ and $|\lambda_2| < 1$, it follows that if $\tau \leq 1$, the inequalities still hold for all $n \geq 0$. If $\tau > 1$, then the RFIC behavior will not change in the neighborhood provided $y_1 = 0$. The set of possible perturbations which maintain the RFIC behavior is therefore of dimension 4 for $\tau > 1$ and of dimension 5 for $\tau \leq 1$.

Because of this we would expect the behavior of the examples to be stable against small numerical perturbations caused by rounding error when $\tau \leq 1$ and not to be stable when $\tau > 1$. This behavior is confirmed by numerical tests. Rounding error introduces a component of the larger eigenvector in the y direction and this is enough to prevent the algorithm converging to the origin when $\tau > 1$, but is not enough to disturb the convergence to the origin when $\tau \leq 1$. Note, however, that in the $\tau > 1$ case the behavior is very sensitive to the representation of the problem and to the details of the implementation of the Nelder–Mead method and of the function. For example, a translation or rotation of the axes can affect whether or not the method converges to the minimizer. The example with $\tau = 1$ is not strictly convex; however, a strictly convex example which is numerically stable can be constructed by taking the average of examples with $\tau = 1$ and with $\tau = 2$.

6. Conclusions. A family of functions of two variables has been presented which cause the Nelder–Mead method to converge to a nonstationary point. Members of the family are strictly convex with up to three continuous derivatives. The examples cause the Nelder–Mead method to perform the inside contraction step repeatedly with the best vertex remaining fixed. It has been shown that this behavior cannot occur for smoother functions. These examples are the best behaved functions currently known which cause the Nelder–Mead method to converge to a nonstationary point. They provide a limit to what can be proved about the convergence of the Nelder–Mead method.

There are six values necessary to specify the initial simplex for functions of two variables. It has been shown that for examples in the family which have a discontinuous first derivative, there is a neighborhood of the initial simplex of dimension 5 in which all the simplices exhibit the same behavior. These examples appear to be numerically stable. For those examples in the family where the gradient exists, the dimension of the neighborhood is only 4. These examples are often numerically unstable and so are less likely to occur in practice due to rounding errors, even for starting simplices within the neighborhood. However, even in cases where numerical errors eventually perturb the simplex enough to escape from the nonstationary focus point, the method can spend a very large number of steps close to this point before escaping. These results highlight the need for variants of the original Nelder–Mead method which have guaranteed convergence properties.

Acknowledgments. The author would like to thank Margaret Wright and Tony Gilbert for help in clarifying the problem and for valuable suggestions, and a referee for very detailed and helpful comments.

REFERENCES

[1] J. E. Dennis, jr. and V. Torczon, *Direct search methods on parallel machines*, SIAM J. Optim., 1 (1991), pp. 448–474.

[2] J. E. Dennis, jr. and D. J. Woods, *Optimization on microcomputers: The Nelder-Mead simplex algorithm*, in New Computing Environments: Microcomputers in Large-Scale Computing, A. Wouk, ed., SIAM, Philadelphia, PA, 1987, pp. 116–122.

[3] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, *Convergence properties of the Nelder–Mead simplex algorithm in low dimensions*, SIAM J. Optim., 9 (1998), pp. 112–147.

[4] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Comput. J., 7 (1965), pp. 308–313.

[5] A. S. Rykov, *Simplex direct search algorithms*, Automat. Remote Control, 41 (1980), pp. 784–793.

[6] A. S. Rykov, *Simplex methods of direct search*, Engrg. Cyber., 18 (1980), pp. 12–18.

[7] A. S. Rykov, *Simplex algorithms for unconstrained optimization*, Prob. Control Inform. Theory, 12 (1983), pp. 195–208.

[8] W. Spendley, G. R. Hext, and F. R. Himsworth, *Sequential applications of simplex designs in optimization and evolutionary operation*, Technometrics, 4 (1962), pp. 441–461.

[9] M. Strasser, *Übertrangung des Optimierungsverfahrens von Nelder und Mead auf restringierte Probleme*, Diploma thesis, Numerical Mathematics Group, Technical University of Darmstadt, Germany, 1994.

[10] P. D. Surry, *Convergence Results for the Nelder–Mead Method*, private communication, Department of Mathematics and Statistics, University of Edinburgh, Edinburgh, UK, 1995.

[11] V. J. Torczon, *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*, Ph.D. thesis, Tech. Report TR90-7, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1990.

[12] V. Torczon, *On the convergence of the multidirectional search algorithm*, SIAM J. Optim., 1 (1991), pp. 123–145.

[13] V. Torczon, *On the convergence of pattern search algorithms*, SIAM J. Optim., 7 (1997), pp. 1–25.

[14] P. Tseng, *Fortified-descent simplicial search method: A general approach*, SIAM J. Optim., to appear.

[15] D. J. Woods, *An Interactive Approach for Solving Multi-Objective Optimization Problems*, Ph.D. thesis, Tech. Report TR85-5, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1985.

SIAM J. OPTIM.
Vol. 9, No. 1, pp. 159–178

THREADING HOMOTOPIES AND DC OPERATING POINTS OF NONLINEAR CIRCUITS[*]

ROSS GEOGHEGAN[†], JEFFREY C. LAGARIAS[‡], AND ROBERT C. MELVILLE[§]

Abstract. This paper studies continuation methods for finding isolated zeros of nonlinear functions. Given a nonlinear function $F : \mathbb{R}^n \to \mathbb{R}^n$, a *threading homotopy* is a function $H(\mathbf{x}, \lambda) : \mathbb{R}^{n+1} \to \mathbb{R}^n$ with $H(\mathbf{x}, 0) \equiv F(\mathbf{x})$, such that the zero set of H is a single connected curve containing all zeros of $F(\mathbf{x})$. For a C^1 function F, a necessary condition for the existence of a nondegenerate C^1 threading homotopy is that the topological degree of $F(\mathbf{x})$ be 1, 0, or -1. For C^2 mappings in all dimensions, except possibly $n = 2$, this condition is also a sufficient condition for existence of a C^2 threading homotopy which is weakly proper over 0. A homotopy H is *weakly proper over* 0 if, for every interval $[a, b]$, the set $H^{-1}(\mathbf{0}) \cap (\mathbb{R}^n \times [a, b])$ is compact. This condition rules out any part of the zero set escaping to infinity at a finite value of the homotopy parameter.

Threading homotopies are potentially applicable in continuation methods for finding all dc operating points of nonlinear circuits. We show that most transistor circuits have dc operating point equations $F(\mathbf{x}) = \mathbf{0}$ with $\deg(F) = \pm 1$, so that threading homotopies exist in principle for such operating point equations. The explicit construction of such threading homotopies remains an open problem.

Key words. homotopy methods, nonlinear circuits, topological degree

AMS subject classifications. 65H20, 94C05

PII. S1052623496251586

1. Introduction.

This paper studies continuation methods for finding all zeros of a nonlinear function $F : \mathbb{R}^n \to \mathbb{R}^n$, which has a finite number of isolated zeros. The continuation approach for finding zeros is to find a function $H(\mathbf{x}, \lambda) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ such that $H(\mathbf{x}, 0) \equiv F(\mathbf{x})$, while $H(\mathbf{x}, 1) \equiv G(\mathbf{x})$ is a function with known zeros, whose zero set

$$\Gamma(H) = \{(\mathbf{x}, \lambda) : H(\mathbf{x}, \lambda) = \mathbf{0}\} \tag{1.1}$$

is a union of curves (one-dimensional components), and these curves can be individually traced from the known zero set

$$\Gamma_1 = \{\mathbf{x} : H(\mathbf{x}, 1) = \mathbf{0}\}$$

to find all solutions Γ_0 of $F(\mathbf{x})$. The function $H(\mathbf{x}, \lambda)$ is called a *homotopy,* and a *homotopy path* is a path $(\mathbf{x}(t),\ \lambda(t))$ for $t \in [0, 1]$ on which $H(\mathbf{x}, \lambda) = 0$. One method for finding all the zeros is to choose a homotopy $H(\mathbf{x}, \lambda)$ such that each zero of $F(\mathbf{x})$ is on a separate connected component of the zero set of $H(\mathbf{x}, \lambda)$, and separate homotopy paths are followed to find each zero of $F(\mathbf{x})$; see, for example, Allgower and Georg [1, section 6], [2], Chow, Mallet-Paret, and Yorke [8], Drexler [17], and Garcia and Zangwill [20], [21]. This has the advantage of permitting the use of parallel computation to find

[*] Received by the editors March 22, 1996; accepted for publication (in revised form) September 3, 1997; published electronically December 2, 1998. Supported in part by a grant from the National Science Foundation.

http://www.siam.org/journals/siopt/9-1/25158.html

[†] Department of Mathematical Sciences, SUNY at Binghamton, Binghamton, NY 13902 (ross@math.binghamton.edu).

[‡] AT&T Labs, Florham Park, NJ 07932-0971 (jcl@research.att.com).

[§] Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 (rcm@research.bell-labs.com).

different zeros. This approach has been proposed in particular to find complex zeros of univariate polynomials $F(z)$; see Kojima, Nishino, and Arima [33].

In this paper we study the opposite extreme, which are homotopies $H(\mathbf{x}, \lambda) : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ with $H(\mathbf{x}, \mathbf{0}) \equiv F(\mathbf{x})$, such that the zero set of $H(\mathbf{x}, \lambda)$ is a single connected curve. We call a homotopy with this property a *threading homotopy* for F, because the zeros of $F(\mathbf{x})$ are threaded along a single curve in the zero set of $H(\mathbf{x}, \lambda)$, which passes back and forth through the hyperplane $\lambda = 0$. More generally we consider *semithreading homotopies*, which are homotopies H in which all zeros of $F(\mathbf{x})$ are on a single connected component of the zero set $\Gamma(H)$ of H, although $\Gamma(H)$ may contain other connected components. Using a semithreading homotopy, all zeros of $F(\mathbf{x})$ can be located by tracing a single curve.

This study of threading homotopies is motivated by the problem of numerically computing all dc operating points of nonlinear resistive circuits, e.g., circuits with transistors. A *dc operating point*[1] for a nonlinear resistive circuit is any solution of a given system of network equations $F(\mathbf{x}) = \mathbf{0}$ for the circuit. The detection of multiple operating points is of considerable practical concern in circuit simulation, because some solutions of the network equations may represent unintended pathological modes of operation, so that the circuit may fail in the field. To avoid this, one would like to detect all possible operating points during the circuit-design phase, or at least alert the designer to the presence of more than one operating point, before a decision is made to fabricate an integrated circuit. Existing circuit simulators do not guarantee finding all operating points, and there is now considerable interest in developing methods that will find all operating points; cf. Mathis and Wettlaufer [34], Trajković, Melville, and Fang [43], and Melville et al. [36]. The use of continuation methods to find individual operating points has a long history; see Chao and Saeks [6]. However, the problem of developing continuation methods guaranteed to find all operating points has received relatively little study. The idea of finding several zeros of $F(\mathbf{x})$ along one curve was made in the early 1970s in Branin [5] and Chua and Ushida [12]. In some of their examples, there are zeros of $F(\mathbf{x})$ on several connected components of $\Gamma(H)$. It is natural in pursuing this approach to try to get all zeros on a single component, which is the threading homotopy problem.

We call a homotopy $H(x, \lambda)$ *weakly proper over* 0 if, for every closed interval $[a, b]$, the restriction H to $\mathbb{R}^n \times [a, b]$ is *proper over* 0; i.e., $H^{-1}(\mathbf{0}) \cap (\mathbb{R}^n \times [a, b])$ is compact. For such a homotopy, no part of the zero set of $H(\mathbf{x}, \lambda)$ can escape to infinity at a finite value of λ. We consider the following problem.

Weakly proper threading homotopy problem. *Let $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a C^r function $(1 \leq r \leq \infty)$ having a finite set of isolated zeros. Construct, if possible, a C^r homotopy*

$$H(\mathbf{x}, \lambda) : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$$

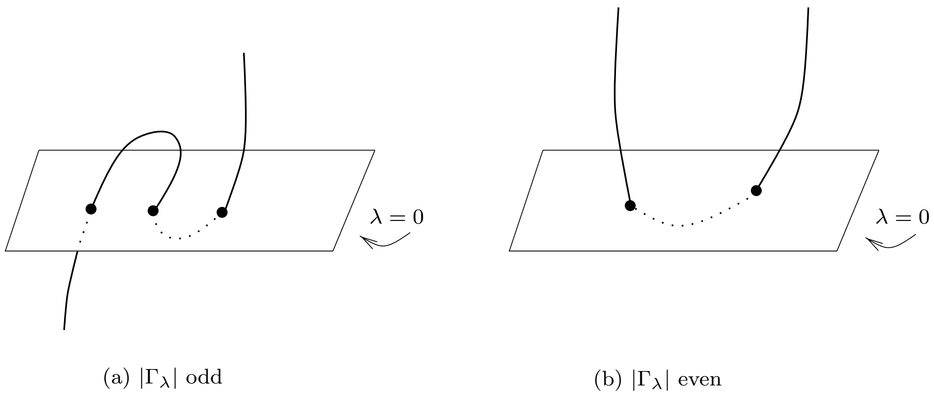
with the following properties.

(i) $H(\mathbf{x}, 0) \equiv F(\mathbf{x})$.

(ii) *Nondegeneracy condition. The Jacobian $DH(\mathbf{x}, \lambda)$ has rank n whenever $H(\mathbf{x}, \lambda) = \mathbf{0}$.*

(iii) *Connectedness condition. The zero set $\Gamma(H) = \{(\mathbf{x}, \lambda) : H(\mathbf{x}, \lambda) = \mathbf{0}\}$ is connected.*

[1] Some authors use the term *dc equilibrium point* for a solution to the network equations $F(\mathbf{x}) = \mathbf{0}$ and reserve the term dc operating point for a linearly stable equilibrium point. We call the latter a *stable dc operating point*, as in Green [24] and Green and Willson [26].



(a) $|\Gamma_\lambda|$ odd

(b) $|\Gamma_\lambda|$ even

Fig. 1.1. *Threading paths.*

(iv) *Weakly proper over* 0 *condition. For every closed interval* $[a, b] \subseteq \mathbb{R}$*, the restriction*

$$H| : \mathbb{R}^n \times [a, b] \rightarrow \mathbb{R}^n \textit{ is proper over } 0.$$

The condition (ii) implies that the zero curves $\{(\mathbf{x}, \lambda) : H(\mathbf{x}, \lambda) = 0\}$ have no bifurcations, and with condition (iii) this implies that the set $H(\mathbf{x}, \lambda) = \mathbf{0}$ is a single curve containing all the zeros of $F(\mathbf{x})$. As mentioned above, condition (iv) prevents the zero set from escaping to infinity at any finite value of λ. The conditions (ii)–(iv) lead to two cases as pictured in Figure 1.1.

Case (a). $F(\mathbf{x}) = \mathbf{0}$ has an odd number of solutions. Then the sets

$$\Gamma_\lambda = \{\mathbf{x} \in \mathbb{R}^n : H(\mathbf{x}, \lambda) = \mathbf{0}\}$$

are nonempty for all $\lambda \in \mathbb{R}$, and $|\Gamma_\lambda| = 1$ for all $|\lambda|$ sufficiently large. (Here, $|\Gamma_\lambda|$ denotes the number of elements in the set Γ_λ.)

Case (b). $F(\mathbf{x}) = \mathbf{0}$ has an even number of solutions. Then $|\Gamma_\lambda| = 0$ for all large λ of one sign, and for large λ of the other sign, $|\Gamma_\lambda| = 0$ or 2 according to whether the zero set $\Gamma(H)$ is bounded or unbounded.

We are particularly interested in Case (a). There, $H(\mathbf{x}, \lambda) = \mathbf{0}$, for large fixed λ_0, has a single zero $\mathbf{x}_{\lambda_0}$, which one can use as the starting point for a homotopy method to find all zeros.

In section 2, we present necessary conditions and sufficient conditions for existence of threading homotopies. It is clear that, given a finite set of isolated points in $\mathbb{R}^{n+1}$, one can always construct a smooth curve in $\mathbb{R}^{n+1}$ passing through these points. However, it is sometimes impossible to extend a map $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ to a weakly proper threading homotopy $H : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$. For a C^1 mapping F, a necessary condition for the existence of a nondegenerate C^1 semithreading homotopy is that the topological degree of F be 0 or ±1 (Theorem 2.1). An immediate consequence is that there exists no C^1 semithreading homotopy for finding all zeros of a complex polynomial $p : \mathbb{C} \rightarrow \mathbb{C}$, where $\mathbb{C}$ is identified with $\mathbb{R}^2$, whenever $p(z)$ is nonlinear (Corollary 2.1). We show, for mappings F that are $C^r (2 \leq r \leq \infty)$ with a finite set of isolated nondegenerate zeros, that the condition $\deg(F) = 0$ or ± 1 is necessary and sufficient for C^r weakly proper threading homotopies to exist in all dimensions, except possibly dimension $n = 2$ (Theorem 2.2). We then show, for mappings F that are C^r $(1 \leq r \leq \infty)$ with a finite set of isolated nondegenerate zeros, that the condition $\deg(F) = 0$ or ± 1 is necessary and sufficient for the existence of a weakly proper C^r semithreading homotopy in all dimensions $n \geq 1$ (Theorem 2.3).

To design threading homotopies, it is clearly useful to have criteria which verify that the threading property holds. Diener [16] gives a (somewhat restrictive) set of global conditions on a C^2 function $H : \mathbb{R}^{n+1} \to \mathbb{R}^n$ which guarantee that it has the threading property. Diener's condition is that there exists some positive K such that

$$\sup\{||(DH(\mathbf{x})DH(\mathbf{x})^T)^{-1}|| : \mathbf{x} \in \mathbb{R}^{n+1}\} \leq K < \infty, \tag{1.2}$$

where the Frobenius norm $||M||$ for the matrix M is $||M||^2 = \sum_{i,j} M_{ij}^2$. He proves that, when (1.2) holds, the Newton method flow gives a retraction of $\mathbb{R}^{n+1}$ onto the set $\Gamma(H)$, thus establishing that $\Gamma(H)$ is a connected set.

In section 3, we return to our motivating problem, which concerns the possible existence of threading homotopies for finding dc operating points of nonlinear circuits. We present theoretical results which indicate that threading homotopies exist for a large class of nonlinear circuits without exhibiting such homotopies explicitly. More precisely, we show that a large class of circuits can be modeled so as to have operating point equations $F(\mathbf{x}) = \mathbf{0}$ with $\deg(F) = \pm 1$. Results showing that $\deg(F) = 1$ for some classes of circuits were already obtained in the 1970s by Wu [52] and Chua and Wang [13], and we describe one such result (Theorem 3.1). This result already applies to a large class of circuits of practical interest. Our main new result of section 3 is a result implying that $\deg(F) = \pm 1$ for operating point equations of circuits in the Sandberg–Willson form with nonlinear elements satisfying a suitable passivity condition (Theorem 3.2). This condition is quite general. It applies to circuits using bipolar junction transistors and may well hold for all other transistor types. In any case, it appears that most if not all transistor models can be easily modified outside the "physically relevant" parameter range to satisfy this passivity condition. The resulting operating point equations then detect all the "physically relevant" operating points. Thus we can construct operating point equations for which threading homotopies exist in principle; the explicit construction of such homotopies remains an open problem. At the end of section 3 we briefly sketch a class of "circuit deformation" homotopies, some of which have been used in circuit simulators (see [35], [43]). These homotopies satisfy a "no-gain" condition which insures properness of the homotopy, as observed in [36], [42]. It may well be that a subclass of these homotopies have the threading property.

The problem of explicitly constructing threading homotopies to find dc operating points seems to warrant further investigation in view of the lack of reasonable alternative methods for finding multiple dc operating points for nonlinear circuits. We are not aware of any existing method that can specify in advance the number of operating points of a given circuit, and this seems to rule out approaches that follow distinct paths to find each zero separately. Other zero-finding methods that proceed by a grid search to find zeros would be prohibitively slow due to the very large dimensionality of the search space for any reasonably sized circuit. Various algorithms have been given to find all operating points for piecewise linear models of circuits; see Chua and Ying [14], Pastore and Premoli [39], and Yamamura [54]. Here the enormous dimensionality of the search space presents difficulties. In contrast, methods that trace a single connected component can be immediately implemented in any software that uses continuation methods. Indeed they are already in use but at present come with no guarantee of finding all dc operating points (see [35], [36], [43]). Finally we note a recent approach using multiparameter homotopies proposed by Wolf and Sanders [51].

This paper presents rigorous results for functions $F(\mathbf{x})$ and homotopies that are continuously differentiable. Similar questions can be raised for piecewise linear functions $F(\mathbf{x})$ using piecewise linear homotopies. Piecewise linear functions and homotopies have been considered in modeling nonlinear circuits; see, for example, Huang and Liu [30], Ohtsuki, Fujisawa, and Kumagai [38], and Vandenberghe, de Moor, and Vandewalle [45].

2. Existence of threading homotopies. We derive necessary conditions and sufficient conditions for the existence of threading homotopies. The basic invariant used is the topological degree of a mapping. Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be continuous and suppose that F is proper over 0, i.e., that the zero set $\Gamma(F)$ is compact. If $\Gamma(F) \subseteq B(\mathbf{0}, T) = \{\mathbf{x} : ||\mathbf{x}|| < T\}$ and $S^{n-1} = \{\mathbf{x} : ||\mathbf{x}|| = 1\}$, then for $R > T$ the map F induces a mapping $\phi_{F,R} : S^{n-1} \to S^{n-1}$ given by

$$\phi_{F,R}(\mathbf{x}) = \frac{F(R\mathbf{x})}{||F(R\mathbf{x})||} \quad \text{for} \quad \mathbf{x} \in S^{n-1}.$$

The homotopy class of $\phi_{F,R}(\mathbf{x})$ in the homotopy group $\pi_{n-1}(S^{n-1}) \cong \mathbf{Z}$ is independent of $R > T$ and is called the *degree of* F, denoted $\deg(F)$. We identify $\pi_{n-1}(S^{n-1})$ with $\mathbf{Z}$, using the isomorphism in which the degree of the identity map is 1, and henceforth view $\deg(F)$ as an integer.

Now suppose that the zeros of F are isolated and finite in number. The *index* $\mathrm{ind}_{\mathbf{x}_0}(F)$ is defined, for any isolated zero $\mathbf{x}_0$ of a continuous function $F(\mathbf{x})$, as the degree of the mapping $F_\epsilon : S^{n-1} \to S^{n-1}$ given by

$$F_\epsilon(\mathbf{x}) := \frac{F(\mathbf{x}_0 + \epsilon\mathbf{x})}{||F(\mathbf{x}_0 + \epsilon\mathbf{x})||}\ , \quad ||\mathbf{x}|| = 1,$$

for small enough positive ϵ (see Cronin [9, p. 53]); any integer can occur as a value of $\mathrm{ind}_{\mathbf{x}_0}(F)$. The *degree of* F is given in terms of the indexes of the zeros of F by

$$\deg(F) = \sum_{F(\mathbf{x}_0)=\mathbf{0}} \mathrm{ind}_{\mathbf{x}_0}(F). \tag{2.1}$$

More generally, for an open set U in $\mathbb{R}^n$ whose closure $\bar{U}$ is compact and with $F(\mathbf{x}) \neq \mathbf{0}$ everywhere on its boundary ∂U, we set

$$\deg(F; U) := \sum_{\substack{F(\mathbf{x}_0)=\mathbf{0} \\ \mathbf{x}_0 \in U}} \mathrm{ind}_{\mathbf{x}_0}(F).$$

Now suppose that F is C^1. A zero $\mathbf{x}_0$ of $F(\mathbf{x})$ is *nondegenerate* if $\det(DF(\mathbf{x}_0)) \neq 0$. (Nondegenerate zeros are always isolated.) The *index* of a nondegenerate zero $\mathbf{x}_0$ then satisfies

$$\mathrm{ind}_{\mathbf{x}_0}(F) = \mathrm{sign}\ (\det(DF(\mathbf{x}_0))) = \pm 1.$$

The degree is an invariant of homotopies which are weakly proper over 0 in the following sense. Suppose the C^r function $H(\mathbf{x}, \lambda) : \mathbb{R}^n \times [0, 1] \to \mathbb{R}^n$ is proper over 0, where $r \geq 1$, and set $F_\lambda = H(\mathbf{x}, \lambda)$. Assume that $\mathbf{0}$ is a regular value for H, for F_0, and for F_1; i.e., all three Jacobians DH, DF_0, and DF_1 have rank n at all points of the zero set. Then $H^{-1}(\mathbf{0})$ is a one-dimensional "neat" C^r-submanifold of $\mathbb{R}^n \times [0, 1]$ (Hirsch [28, Theorem 1.4.1]). This 1-manifold is compact because H is proper over 0, so the zero set does not "escape to infinity." Then one has $\deg(F_0) = \deg(F_1)$ by an easy adaptation of the proof of Corollary 5.1.3 of Hirsch [28]. Similarly, if U is as above and $H^{-1}(\mathbf{0})$ is disjoint from $\partial U \times [0, 1]$, then $\deg(F_0; U) = \deg(F_1; U)$. The necessity for the assumption "proper over 0" in such a homotopy is shown (for $n = 1$) by

$$H(\mathbf{x}, \lambda) = \frac{2}{\pi} \ \arctan(x) - \lambda,$$

where "escape to infinity" occurs, and $\deg(F_0) \neq \deg(F_1)$.

We give a necessary condition for the existence of a semithreading homotopy. Call a C^1 homotopy H *nondegenerate* if its Jacobian $DH(x, \lambda)$ has full rank n at every zero of $H(\mathbf{x}, \lambda)$.

Theorem 2.1. *Suppose that the zero set of a C^1 mapping $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ consists of a finite number of isolated nondegenerate zeros. If the C^1 function $H(\mathbf{x}, \lambda) : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ is a nondegenerate semithreading homotopy extending $F(\mathbf{x})$, then the degree of F is* 1, 0, *or* -1.

The simple proof of this result is based on the following well-known fact, which concerns the index of successive zeros encountered in following a continuation method path having no bifurcations. It is essentially Corollary 5.1.1 in Hirsch [28], who, however, assumes all functions are C^∞; see also [3, Corollary 11.5.6]. We include a proof for the reader's convenience.

Lemma 2.1. *Suppose that the C^1 mapping $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ has a finite zero set with all zeros nondegenerate. If $H(\mathbf{x}, \lambda) : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ is a C^1 function with $H(\mathbf{x}, 0) = F(\mathbf{x})$ and the Jacobian $DH(\mathbf{x}, \lambda)$ has full rank n at every zero of H, then any two consecutive zeros $\mathbf{x}'$, $\mathbf{x}''$ of $F(\mathbf{x})$ found by traversing a solution curve $(\mathbf{x}(t),\ \lambda(t))$ of $H(\mathbf{x}, \lambda) = 0$ have opposite index; i.e.,*

$$\det(DF(\mathbf{x}')) \det(DF(\mathbf{x}'')) < 0. \tag{2.2}$$

Proof. By the implicit function theorem, $(\mathbf{x}(t),\ \lambda(t))$ is locally defined and C^1 in a neighborhood of every zero $(\mathbf{x}_0,\ \lambda_0)$ of $H(\mathbf{x},\ \lambda)$. When traversing the curve $(\mathbf{x}(t),\ \lambda(t))$, in the zero set $\Gamma(H)$ from $\mathbf{x}'$ to $\mathbf{x}''$, the augmented gradient of $H = (H_1, \ldots, H_n)^t$ is

$$\mathbf{J} := \begin{bmatrix} D\tilde{H} & \dfrac{\partial H}{\partial \lambda} \\ & \\ \dfrac{d\mathbf{x}}{dt} & \dfrac{d\lambda}{dt} \end{bmatrix},$$

in which $D\tilde{H} = \left[\frac{\partial H_i}{\partial \mathbf{x}_j}\right]$ and $\frac{d\mathbf{x}}{dt} = \left(\frac{d\mathbf{x}_1(t)}{dt}, \ldots, \frac{d\mathbf{x}_n(t)}{dt}\right)$. The augmented Jacobian $\det(\mathbf{J})$ is always nonzero, because the tangent vector $\mathbf{v} = \left(\frac{d\mathbf{x}}{dt}\ ,\ \frac{d\lambda(t)}{dt}\right)$ to the curve is perpendicular to the row space of $DH(\mathbf{x}(t),\ \lambda(t))$. Hence $\det(\mathbf{J})$ has constant sign; call this sign $\hat{\epsilon}$. In addition this perpendicularity gives

$$\begin{bmatrix} D\tilde{H} & \dfrac{\partial H}{\partial \lambda} \\ & \\ \dfrac{d\mathbf{x}}{dt} & \dfrac{d\lambda}{dt} \end{bmatrix} \begin{bmatrix} I & \dot{\mathbf{x}}^T \\ & \\ \mathbf{0} & \dfrac{d\lambda}{\partial t} \end{bmatrix} = \begin{bmatrix} D\tilde{H} & \mathbf{0} \\ & \\ \dfrac{d\mathbf{x}}{dt} & 1 \end{bmatrix},$$

because $\left(\frac{d\lambda}{dt}\right)^2 + \sum_{i=1}^n \left(\frac{dx_i}{dt}\right)^2 = 1$, using the arclength parametrization. Taking determinants, we obtain

$$\det(\mathbf{J})\frac{d\lambda}{dt} = \det(D\tilde{H}).$$

At a point $(\mathbf{x}(t'),\ \lambda(t')) = (\mathbf{x}',\ 0)$ which gives a zero of F, $D\tilde{H}(\mathbf{x}') = DF(\mathbf{x}')$; hence

$$\text{sign }(\det DF(\mathbf{x}')) = \hat{\epsilon}\text{ sign}\left(\frac{d\lambda}{dt}\right). \tag{2.3}$$

If $t' < t''$ are two consecutive zeros of $\lambda(t)$ along the curve, then the sign of $\lambda(t)$ is constant on the interval (t', t''), while

$$\text{sign }\left(\frac{d\lambda}{dt}(t')\right) = -\text{ sign}\left(\frac{d\lambda}{dt}(t'')\right).$$

Then (2.3) shows that $\det DF(\mathbf{x}')$ and $\det(DF(\mathbf{x}''))$ have opposite signs, and (2.2) follows. □

Proof of Theorem 2.1. Suppose that $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ are the zeros of $F(\mathbf{x})$ in the order they are encountered when traversing, in a fixed direction, the curve $\{(\mathbf{x}(t),\ \lambda(t)) : t \in \mathbb{R}\}$ comprising the connected component of the zero set $\Gamma(H)$ that contains the zeros of $F(\mathbf{x})$. By Lemma 2.1,

$$\text{ind}_{\mathbf{x}_i}(F) + \text{ ind}_{\mathbf{x}_{i+1}}(F) = 0.$$

Applying this in pairs, we have $\deg(F) = 0$ if $F(\mathbf{x})$ has an even number of zeros, and

$$\deg(F) = \text{ ind}_{\mathbf{x}_m}(F) = \pm 1$$

if $F(\mathbf{x})$ has an odd number of zeros. □

This degree constraint of Theorem 2.1 is automatically satisfied in dimension $n = 1$, and in that case the homotopy

$$H(x, \lambda) = F(x) - \lambda \tag{2.4}$$

is always a threading homotopy. However, in dimensions $n \geq 2$ the degree constraint is a nontrivial obstruction.

Corollary 2.1. *Let $p(z) = \sum_{j=0}^d a_j z^j$ be a polynomial of degree $d \geq 2$ with distinct roots. If $p(z)$ is regarded as a mapping $p : \mathbb{C} \to \mathbb{C}$, then there exists no semithreading homotopy $H(z, \lambda) : \mathbb{C} \times \mathbb{R} \to \mathbb{C}$ for p.*

Proof. The index of each simple zero of a polynomial $p(z)$ is 1. To see this, translate the zero to $z = 0$, and, by simplicity of the zero, only linear terms in $p(z)$ contribute to the index, so without loss of generality suppose that $p(z) = \alpha z$, with $\alpha \neq 0$. Write $\alpha = a + bi$ and $z = x + yi$, and viewing $p(z) = (\text{Re}(p(z)), \text{Im}(p(z)))$ in $\mathbb{R}^2$ one finds

$$Dp(\mathbf{0}) = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}.$$

Hence $\det(Dp(0)) = a^2 + b^2 > 0$ since $\alpha \neq 0$. Thus $\deg(p)$ is just the algebraic degree of $p(z)$ and is at least 2 if $p(z)$ is not linear; hence Theorem 2.1 gives the result. □

Corollary 2.1 also holds for polynomials $p(z)$ having multiple zeros, using the general definition of index (cf. Milnor [37, p. 32]), but it does not apply to general multivariate polynomial maps $P : \mathbb{C}^n \to \mathbb{C}^n$. For $n \geq 2$, such a polynomial map can have an isolated zero with index -1. However, one can show that if the map $P(\mathbf{z})$ has real coefficients, then all nondegenerate real zeros of $P(\mathbf{z})$ have index 1; see Cronin [9, Lemma 9.3.2]. In particular, if such a map has at least two zeros, with all zeros real and nondegenerate, then $\deg(P) \geq 2$, so that Theorem 2.1 applies to show that no threading homotopy exists.

We next establish sufficiency of the condition $\deg(F) = 0$, or ± 1 for the existence of a threading homotopy for C^2 mappings in dimensions $n \neq 2$. For this we introduce a condition stronger than "weakly proper over 0." Call a function $H(\mathbf{x}, \lambda)$ $\mathbb{R}$*-proper over* 0 if there is a compact set $B \subseteq \mathbb{R}^n$ such that $H^{-1}(\mathbf{0}) \subseteq B \times \mathbb{R}$. This is "weakly proper over 0" with an additional uniformity condition in the $\mathbb{R}$-direction.

Theorem 2.2. *For any $n \neq 2$, let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a C^r mapping* $(2 \leq r \leq \infty)$ *whose zero set consists of a finite number of isolated nondegenerate zeros. If* $\deg(F)$ *is* $1, 0,$ *or* -1*, then there exists a nondegenerate threading homotopy* $H(\mathbf{x}, \lambda) : \mathbb{R}^{n+1} \to \mathbb{R}^n$ *for F, such that H is a C^r mapping which is* $\mathbb{R}$*-proper over* 0.

We do not know if Theorem 2.2 is true when $n = 2$.

The main part of the proof is the following.

Lemma 2.2. *Suppose that $n \geq 3$ and that $F : \mathbb{R}^n \to \mathbb{R}^n$ is a C^r mapping* $(2 \leq r \leq \infty)$ *which has exactly two zeros* $\mathbf{x}^{\pm} = (\pm 1,\ 0, \ldots, 0)$ *with* $ind_{\mathbf{x}^+}(F) = 1$ *and* $ind_{\mathbf{x}^-}(F) = -1$*. Then there exists a C^r homotopy* $H(\mathbf{x}, \lambda) : \mathbb{R}^n \times [0,1] \to \mathbb{R}^n$ *with* $H(\mathbf{x}, 0) = F(\mathbf{x})$*, which is stationary outside a preassigned neighborhood of the line segment connecting* $\mathbf{x}^+$ *and* $\mathbf{x}^-$*, such that* $H^{-1}(\mathbf{0})$ *is a* C^r*-embedded "neat" arc in* $\mathbb{R}^n \times [0,1]$.

Here, as in Hirsch [28, p. 30], "neat" means that the arc meets the boundary at $(\mathbf{x}^+, 0)$ and at $(\mathbf{x}^-, 0)$ in a C^r-manner.

Proof. A form of Lemma 2.2 is essentially to be found in Whitney [46]; topologists call all of its variants "the Whitney lemma." The condition $n \geq 3$ arises from Whitney's need to approximate a singular disk in $\mathbb{R}^{2n}$ by an embedded disk. It is not clear to us, however, that the proof in [46] avoids introducing extra circle components in H^{-1} $(\mathbf{0})$ $\cap(\mathbb{R}^n \times [0,1])$: compare the difference between semithreading and threading above. However, this problem is avoided in a rather elementary proof of Lemma 2.2 appearing in Jezierski [31, Lemma 2.2]. The proof of Jezierski makes no mention of embedded disks. Rather, it uses advanced calculus and the fact that spheres of dimension ≥ 2 are simply connected. Like Whitney's proof, it is presented for the C^∞-case; however, the proof requires only the hypothesis C^2, hence our restriction $r \geq 2$. Jezierski uses $n \geq 3$ for the property of $(n-1)$-spheres mentioned above. □

Proof of Theorem 2.2. The necessary degree condition was already shown to be sufficient in dimension 1 (see (2.4)), so suppose that $n \geq 3$. Lemma 2.2 shows how to "remove" a pair of zeros of opposite degree. Now suppose $\deg(F) = \pm 1$. Then one can arrange the zeros in an order $x_1,\ x_2, \ldots, x_{2m+1}$ so that consecutive zeros have opposite degree. One can find arcs connecting them in pairs $(x_i,\ x_{i+1})$ so that all tubular neighborhoods of disjoint pairs are disjoint. One can then combine the homotopies above for the pairs (x_1, x_2), $(x_3, x_4), \ldots, (x_{2m-1}, x_{2m})$, with homotopy parameter $1 \geq \lambda \geq 0$ and those for (x_2, x_3), $(x_4, x_5), \ldots, (x_{2m}, x_{2m+1})$, with homotopy parameter $0 \geq \lambda \geq -1$, to obtain a threading homotopy, which is $\mathbb{R}$-proper over 0.

With care, one can ensure that the "combined" homotopy is still C^r; see Jezierski [31] for a discussion of similar matters. A slight and obvious modification handles the case $\deg(F) = 0$. □

Finally, we establish the sufficiency of the condition $\deg(F) = 0$ or ± 1 for the existence of a semithreading homotopy for C^1 mappings in all dimensions $n \geq 1$. We include this result because it is the best we can do when $n = 2$, and, while it obtains a weaker conclusion than Theorem 2.2, it has a more elementary proof.

THEOREM 2.3. *For all $n \geq 1$, let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a C^r mapping $(1 \leq r \leq \infty)$ whose zero set consists of a finite number of isolated nondegenerate zeros. If $\deg(F) = 1,\ 0,$ or -1, then there exists a nondegenerate C^r homotopy $H(\mathbf{x}, \lambda) : \mathbb{R}^{n+1} \to \mathbb{R}^n$ extending F, which is weakly proper over 0, such that the following are true.*

(i) *All zeros of $F(\mathbf{x})$ lie on a single connected component of the zero set of $H(\mathbf{x}, \lambda)$.*

(ii) *All other components of the zero set of $H(\mathbf{x}, \lambda)$ are closed loops on which $0 < |\lambda| < 1$.*

Proof. The necessary degree condition was already shown to be sufficient in dimension 1; see (2.4). For $n \geq 2$, we use an approach which starts from the proof of Lemma 5.2.9 of Hirsch [28]. That lemma essentially shows that, given two zeros of degree 1 and -1, respectively, together with an arc connecting them, and a tubular neighborhood U_2 of the arc, then there is a continuous function G agreeing with F outside U_2 which has no zeros in U_2. That lemma is stated for C^∞ maps, but the cited proof and all other proofs cited below go through without change for C^r maps, with $r \geq 1$.

Now pick a nested collection of tubular neighborhoods $U_2 \subset U_1 \subset U_0 \subset U$, where the closure of each lies in the next; to find these, use the tubular neighborhood theorem, Theorem 4.5.2 of Hirsch [28]. We have already used U_2. Note that $G = F$ outside U_1 so G is C^r outside U_1. We use the relative approximation theorem (Theorem 2.2.5) of Hirsch [28] to obtain a C^r map $\tilde{G}$ agreeing with $G \equiv F$ outside U_1, which is close enough to G inside U_2 that it has no zeros there. (To be specific, use that theorem with Hirsch's U and K being U_0 and with his W being U_1.)

Let $\phi : \mathbb{R} \to \mathbb{R}$ be a C^r function satisfying $0 \leq \phi(\lambda) \leq 1$, with $\phi(\lambda) = 0$ for $\lambda \leq 0$ and $\phi(\lambda) = 1$ for $\lambda \geq 1$, and define $J : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ by

$$J(\mathbf{x}, \lambda) = \phi(1 - \lambda)F(\mathbf{x}) \ + \ \phi(\lambda)\tilde{G}(\mathbf{x}) \ .$$

Now J is C^r. Let $N = \overline{U_0 \times [0, 1]}$. N has "corners" at $\partial U_0 \times \{0, 1\}$, so N is a C^0 manifold but not a differentiable manifold. The set $(J|_{\partial N})^{-1}(\mathbf{0}) = \{\mathbf{x}', \mathbf{x}''\}$, where $\mathbf{x}'$ and $\mathbf{x}''$ are the two zeros of F that we are trying to remove. By "smoothing out the corners" (see Kirby and Siebenmann [32, pp. 8 and 119]), we can find a differentiable manifold M lying in the interior of N with respect to $\mathbb{R}^n \times [0, \infty)$—interior in the topological sense—such that $U_1 \times \{0\} \subseteq M$; see Figure 2.1.

There is a relative transversality theorem, stated as "Corollary" on p. 73 of Guillemin and Pollack [27], which says that $J|_{\partial M}$ can be extended to a map $\tilde{J} : M \to \mathbb{R}^n$ which is transverse to $\{\mathbf{0}\}$. Extend $\tilde{J}$ to $\mathbb{R}^n \times \mathbb{R}$ so as to be C^r and agree with J outside M. Then $(\tilde{J}|_M)^{-1}(\mathbf{0})$ includes an arc of zeros of the type desired. However, $(\tilde{J}|_M)^{-1}(\mathbf{0})$ may also contain extra components, which are closed loops in the interior of M. If $\mathbf{x} \notin U_0$, then $\tilde{J}(\mathbf{x}, \lambda) = F(\mathbf{x}, \lambda)$ for all $\lambda \in \mathbb{R}$. Thus we have shown how to connect and "remove" a pair of zeros of opposite degree.

Now if $\deg(F) = -1,\ 0,$ or 1, one proceeds exactly as in the proof of Theorem 2.2 to thread all the zeros together. □

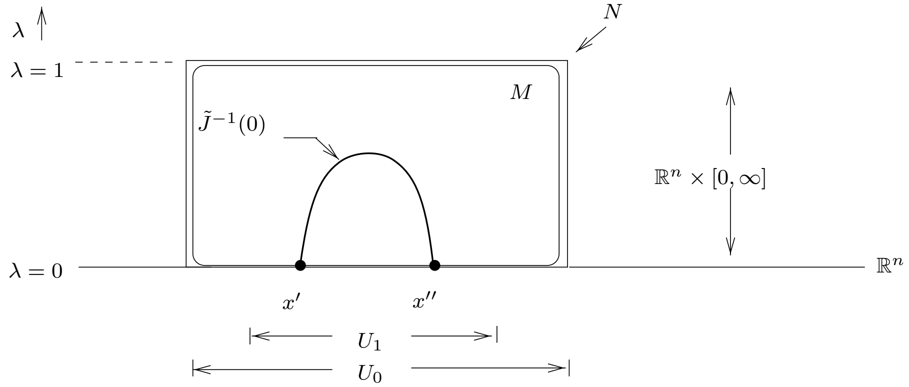


FIG. 2.1. *Smoothing out the corners.*

3. DC operating points of nonlinear resistive circuits. The theory of dc operating points of transistor circuits is surveyed in Trajković and Willson [44] and, for work before 1974, in Willson [48]. In this section we make the theoretical observation that threading homotopy methods potentially apply to the dc operating point problem by showing that most circuits can be modeled with operating point equations $F(\mathbf{x}) = \mathbf{0}$ such that $\deg(F) = \pm 1$. It follows that there is no topological obstruction to the existence of threading homotopies for such equations, and they certainly exist whenever Theorem 2.2 applies, i.e., when F is C^2; see also Theorem 2.3.

There is a long history of results on topological degree applied to nonlinear networks. These methods were developed to prove the existence of dc operating points, for which it suffices to prove that $\deg(F)$ is odd; see Chua and Wang [13, Property 7]. The original method of Wu [52] uses passivity properties of the circuits to prove $\deg(F) = \pm 1$, and we follow this approach here.

We consider nonlinear circuits made up of transistors and nonlinear diodes driven by active sources which are current sources or voltage sources. A nonlinear resistive network is *passive* if

$$P(\mathbf{v}, \mathbf{i}) := \langle \mathbf{v}, \mathbf{i} \rangle := v_1 i_1 + v_2 i_2 + \cdots + v_n i_n \geq 0 \tag{3.1}$$

for any allowed set of voltages $\mathbf{v}$ and currents $\mathbf{i}$. Here $P(\mathbf{v}, \mathbf{i})$ measures the *power* consumed by the network, and the passivity condition[2] asserts that a network never generates power internally, but it may consume power. Circuits composed solely of nonlinear passive resistors and transistors with no voltage or current sources are passive.

For a general circuit we extract a set of n independent variables $\mathbf{x} = (x_1, \ldots, x_n)$ among the $2n$ variables $\{v_1, \ldots, v_n, i_1, \ldots, i_n\}$, one from each pair (v_j, i_j), and solve for the remaining variables $\mathbf{y} = (y_1, \ldots, y_n)$ using Kirchhoff's voltage and current laws, to obtain

$$\mathbf{y} = F(\mathbf{x}).$$

That is, the variables $\mathbf{y}$ are uniquely determined as functions of $\mathbf{x}$, and we call $\mathbf{x}$ the *controlling variables.* The simplest case consists of *voltage-controlled circuits*, in

[2] More general definitions of passivity are discussed in Chua, Desoer, and Kuh [10] and Wyatt et al. [53].

which the controlling variables $\mathbf{v} = (v_1, \ldots, v_n)$ are the node voltages, giving potentials measured from a reference node ("ground") in the network, and the remaining variables $\mathbf{i} = (i_i, \ldots, i_n)$ give the currents at each node. (There are no voltage or current variables for the reference node.) We may force the node voltage at node k to be v_k by attaching a new branch from the reference node to node k which either contains a voltage source with potential v_k or a current source with current i_k, with the branch oriented towards node k. We define the column vector

$$F(\mathbf{v}) := (F_1(\mathbf{v}), \ldots, F_n(\mathbf{v}))^T, \tag{3.2}$$

where $i_k = F_k(\mathbf{v})$ denotes the current at node k entering from the branch containing the voltage source v_k. The operating point equations for a voltage-controlled circuit with offered currents $\mathbf{i} = (i_1, i_2, \ldots, i_n)$ is

$$F(\mathbf{v}) = \mathbf{i}. \tag{3.3}$$

For fixed $\mathbf{i} \in \mathbb{R}^n$ this equation may have zero, one, or many solutions in $\mathbf{v}$. The power $P(\mathbf{v}, \mathbf{i})$ drawn by the circuit from the voltage sources is

$$P(\mathbf{v}, \mathbf{i}) := \langle \mathbf{v}, \mathbf{i} \rangle = \langle \mathbf{v}, F(\mathbf{v}) \rangle, \tag{3.4}$$

and the passivity condition asserts that $P(\mathbf{v}, \mathbf{i}) \geq 0$.

The relevance of a passivity condition to the topological degree of $F(\mathbf{v}) - \mathbf{i}$ is the following well-known fact.

LEMMA 3.1. *If a function $G : \mathbb{R}^n \to \mathbb{R}^n$ satisfies a strict coercivity condition*

$$\langle \mathbf{x}, G(\mathbf{x}) \rangle > 0 \quad \textit{if} \quad ||\mathbf{x}|| \geq R, \tag{3.5}$$

then $\deg(G) = 1$.

Proof. The condition (3.5) shows that all zeros of $G(\mathbf{x})$ lie in the compact set $||\mathbf{x}|| \leq R$. The map $\phi_{G,R}(\mathbf{x}) = \frac{G(R\mathbf{x})}{||G(R\mathbf{x})||}$ is homotopic to the identity map on S^{n-1} using radial projection of the map

$$G_\lambda(\mathbf{x}) = \lambda G(R\mathbf{x}) + (1 - \lambda)\mathbf{x}, \qquad \mathbf{x} \in S^{n-1}, \qquad 0 \leq \lambda \leq 1,$$

onto S^{n-1}, which is well-defined since the strict coercivity condition gives $\langle \mathbf{x}, G_\lambda(\mathbf{x}) \rangle > 0$; hence $G_\lambda(\mathbf{x}) \neq \mathbf{0}$. Now $\deg(G) = 1$ by the invariance of degree for homotopies proper over 0, as explained in section 2. □

If we set

$$F_{\mathbf{i}}(\mathbf{v}) := F(\mathbf{v}) - \mathbf{i}, \tag{3.6}$$

then a sufficient condition for $F_{\mathbf{i}}(\mathbf{v})$ to satisfy a strict coercivity condition (3.5) can be given in terms of the power drawn by the circuit. We say that function $F(\mathbf{x})$ is *eventually strongly passive* if there exist $c > 0$ and $R > 0$, such that

$$\langle \mathbf{x}, F(\mathbf{x}) \rangle \geq c||\mathbf{x}||^2 \quad \text{for } ||\mathbf{x}|| > R. \tag{3.7}$$

A positive linear resistor has this property. Eventual strong passivity of $F(\mathbf{x})$ implies eventual strong passivity of $F_{\mathbf{c}}(\mathbf{x})$ for each $\mathbf{c} \in \mathbb{R}^n$, since

$$\begin{aligned} \langle \mathbf{x}, F_{\mathbf{c}}(\mathbf{x}) \rangle &= \langle \mathbf{x}, F(\mathbf{x}) \rangle - \langle \mathbf{x}, \mathbf{c} \rangle \\ &\geq c||\mathbf{x}||^2 - ||\mathbf{x}|| \ \ ||\mathbf{c}|| \quad \text{if } ||\mathbf{x}|| > R, \\ &\geq \frac{1}{2}c||\mathbf{x}||^2 \qquad \text{if } ||\mathbf{x}|| > R', \end{aligned} \tag{3.8}$$

with $R' = \max(\frac{2}{c}\|\mathbf{c}\|, R)$.

We now present results which show that $\deg(F) = \pm 1$ for two large classes of circuit equations. The first class of transistor circuits has dc operating point equations that are of the form

$$\tilde{F}(\mathbf{x}) + P\mathbf{x} = \mathbf{s}, \tag{3.9}$$

where $\tilde{F}(\mathbf{x})$ is a vector of n functions of $\mathbf{x}$ describing the effect of nonlinear elements of the circuit, assumed to be eventually passive (defined below); the conductance matrix P is assumed to be positive definite but not necessarily symmetric; and $\mathbf{s}$ is a constant describing the active sources in the circuit. Chua and Wang [13, Theorem 2] prove the following result.

THEOREM 3.1. *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a function*

$$F(\mathbf{x}) := \tilde{F}(\mathbf{x}) + P\mathbf{x}, \tag{3.10}$$

in which P is a positive definite matrix and $\tilde{F}(\mathbf{x})$ is eventually passive; i.e.,

$$\langle \mathbf{x}, \tilde{F}(\mathbf{x})\rangle \geq 0 \quad \textit{for all } ||\mathbf{x}|| \geq R. \tag{3.11}$$

Then $F(\mathbf{x})$ is eventually strongly passive; hence if $F_{\mathbf{c}}(\mathbf{x}) = F(\mathbf{x}) - \mathbf{c}$, then $\deg(F_{\mathbf{c}}) = 1$ *for all $\mathbf{c} \in \mathbb{R}^n$.*

Proof. Positive definiteness of P gives

$$\langle \mathbf{x}, P\mathbf{x}\rangle \geq c||\mathbf{x}||^2, \quad \text{for all} \quad \mathbf{x} \in \mathbb{R}^n,$$

for some positive constant c. The eventual passivity condition for $\tilde{F}(\mathbf{x})$ yields

$$\langle \mathbf{x}, F(\mathbf{x})\rangle \geq \langle \mathbf{x}, P\mathbf{x}\rangle \geq c||\mathbf{x}||^2 \quad \text{if } ||\mathbf{x}|| \geq R; \tag{3.12}$$

i.e., $F(\mathbf{x})$ is eventually strongly passive. Now $\deg(F_{\mathbf{c}}) = 1$ follows from (3.8) and Lemma 3.1. □

Theorem 3.1 is readily applicable to a wide class of practical circuits. Consider for the moment voltage-controlled circuits using bipolar junction transistors. These circuits are modeled using variants of the Ebers–Moll transistor model as a two-port in the common base configuration; see Appendix A. The resulting circuit equation has the form (3.9) except that P is positive semidefinite rather than positive definite. Existing circuit simulators, such as SPICE, add small shunt conductances to the Ebers–Moll model; see, for example, [4, pp. 14, 44, and 45], where the variable is denoted GMIN. These conductances are modeled as two resistors with resistances $(GMIN)^{-1}$ between the base and the other two nodes of the transistor. If these resistors are migrated to the linear part of the circuit, this will change the matrix P to $P + \mathrm{diag}(GMIN)$, which is positive definite, and Theorem 3.1 applies. Green and Willson [26] give a detailed description of circuits satisfying Theorem 3.1. This theorem may also apply to some circuit equations represented in other forms, such as those used in Ho, Ruehli, and Brennan [29] and Willson and Wu [50].

We next prove a general result which applies to nonlinear circuits in the Sandberg–Willson form that separates linear and nonlinear parts of the circuit (see [40], [41], and [47]) and that assumes a weaker passivity condition than Theorem 3.1. This result applies to circuits with Ebers–Moll transistors without shunt conductances added. The nonlinear elements are treated as voltage-controlled, with response function

$$F(\mathbf{v}) = -\mathbf{i}\ , \quad \text{with} \quad \mathbf{i} = \begin{bmatrix} i_1 \\ \vdots \\ i_n \end{bmatrix}. \tag{3.13}$$

The linear part of the circuit has response

$$Q\mathbf{i} = P(\mathbf{v} - \mathbf{c}), \tag{3.14}$$

in which (P, Q) are a *passive pair* of $n \times n$ matrices, i.e.,

$$Q\mathbf{i} = P\mathbf{v} \quad \text{implies } \langle \mathbf{v}, \mathbf{i} \rangle = \mathbf{v}^T \mathbf{i} \geq 0, \tag{3.15}$$

and $\mathbf{c}$ is a vector of constants representing independent sources. Any linear circuit consisting of positive linear resistors and independent voltage sources can be put in the form (3.14), as well as many linear circuits containing current sources; see Sandberg and Willson [41, Theorem 1 ff.]. This set of equations is converted to circuit equations in Sandberg–Willson form by eliminating the current variables $\mathbf{i}$ to obtain the nonlinear system of equations

$$QF(\mathbf{v}) + P(\mathbf{v} - \mathbf{c}) = \mathbf{0}. \tag{3.16}$$

We establish the following result.

Theorem 3.2. *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a C^1 mapping and consider the mapping $\tilde{G} : \mathbb{R}^n \to \mathbb{R}^n$ given by*

$$\tilde{G}(\mathbf{x}) := QF(\mathbf{x}) + P(\mathbf{x} - \mathbf{c}), \tag{3.17}$$

in which (P, Q) is a passive pair of $n \times n$ matrices, and $\mathbf{c}$ is given. If there exists $R > 0$ such that $F(\mathbf{x})$ satisfies

$$\langle \mathbf{x} - \mathbf{c},\ F(\mathbf{x}) \rangle > 0 \quad for \quad ||\mathbf{x}|| > R, \tag{3.18}$$

then $\deg(\tilde{G}) = \pm 1$.

Remark. The condition (3.18) is a passivity condition that is less stringent than the eventually strong passivity condition (3.7). Note also that the form of $\tilde{G}(\mathbf{x})$ can apply to operating point equations using any set of controlling variables (hybrid variables) rather than voltages.

Proof. We first study the $2n \times 2n$ system $G = (G_1, G_2)$ given by

$$\begin{aligned} G_1(\mathbf{x}, \mathbf{y}) &:= F(\mathbf{x}) + \mathbf{y}, \\ G_2(\mathbf{x}, \mathbf{y}) &:= Q\mathbf{y} - P(\mathbf{x} - \mathbf{c}). \end{aligned}$$

We consider the homotopy $H : \mathbb{R}^{2n+1} \to \mathbb{R}^{2n}$ given by $H = (H_1, H_2)$ with

$$\begin{aligned} H_1(\mathbf{x}, \mathbf{y}, \lambda) &:= (1 - \lambda)F(\mathbf{x}) + \lambda(\mathbf{x} - \mathbf{c}) + \mathbf{y}, \\ H_2(\mathbf{x}, \mathbf{y}, \lambda) &= Q\mathbf{y} - P(\mathbf{x} - \mathbf{c}). \end{aligned} \tag{3.19}$$

We will show that H is a homotopy proper over $\mathbf{0}$ and that $\deg(H(\mathbf{x}, \mathbf{y}; 1)) = \pm 1$. This will imply that $\deg(G) = \deg(H(\mathbf{x}, \mathbf{y}, 0)) = \pm 1$ by invariance of degree for proper homotopies.

To see that H is a proper homotopy, we show that all zeros of $H(\mathbf{x}, \mathbf{y}, \lambda)$ for $0 \leq \lambda \leq 1$ lie in a compact set. Any such zero satisfies

$$\begin{aligned} &(1 - \lambda)F(\mathbf{x}) + \lambda(\mathbf{x} - \mathbf{c}) + \mathbf{y} = \mathbf{0}, \\ &Q\mathbf{y} = P(\mathbf{x} - \mathbf{c}). \end{aligned} \tag{3.20}$$

Now

$$\langle \mathbf{x}-\mathbf{c}, H_1(\mathbf{x},\mathbf{y},\lambda)\rangle = (1-\lambda)\langle \mathbf{x}-\mathbf{c}, F(\mathbf{x})\rangle + \lambda||\mathbf{x}-\mathbf{c}||^2 + \langle \mathbf{x}-\mathbf{c},\mathbf{y}\rangle.$$

The passive pair condition gives

$$\langle \mathbf{x}-\mathbf{c},\mathbf{y}\rangle \geq 0,$$

which, with (3.18), gives for $0 \leq \lambda \leq 1$ that

$$\langle \mathbf{x}-\mathbf{c}, H_1(\mathbf{x},\mathbf{y},\lambda)\rangle > 0 \quad \text{if} \quad ||\mathbf{x}|| > R',$$

where we define $R' = \max(R, ||\mathbf{c}||)$.

To see that $\deg(H(\mathbf{x},\mathbf{y},1)) = \pm 1$, we observe that $G^*(\mathbf{x},\mathbf{y}) = H(\mathbf{x},\mathbf{y},1)$ has

$$\begin{aligned} G_1^*(\mathbf{x},\mathbf{y}) &= \mathbf{x}-\mathbf{c}+\mathbf{y}, \\ G_2^*(\mathbf{x},\mathbf{y}) &= G_2(\mathbf{x},\mathbf{y}) \ = \ Q\mathbf{y} - P(\mathbf{x}-\mathbf{c}). \end{aligned}$$

Thus any zero of G^* has $\mathbf{y} = -(\mathbf{x}-\mathbf{c})$, and the equation $G_2^*(\mathbf{x},\mathbf{y}) = \mathbf{0}$ becomes

$$Q(-(\mathbf{x}-\mathbf{c})) = P(\mathbf{x}-\mathbf{c}).$$

Since (P,Q) is a passive pair, this gives

$$-(\mathbf{x}-\mathbf{c})^T(\mathbf{x}-\mathbf{c}) = -||\mathbf{x}-\mathbf{c}||^2 \geq 0.$$

This forces $\mathbf{x} = \mathbf{c}$, hence G^* has a unique zero $(\mathbf{c},\mathbf{0})$. Since G^* is an affine map that has a unique zero, it is invertible, hence its Jacobian $\det(DG^*)$ does not vanish. Thus

$$\deg(H(\mathbf{x},\mathbf{y},1)) = \deg(G^*) = sgn\left(\det\begin{vmatrix} I & I \\ -P & Q \end{vmatrix}\right) = \pm 1. \tag{3.21}$$

We have now established that

$$\deg(G) = \deg(H(\mathbf{x},\mathbf{y},0)) = \deg(H(\mathbf{x},\mathbf{y},1)) = \pm 1.$$

To complete the proof, we show that $|\deg(\tilde{G})| = |\deg(G)|$. Let $K : \mathbb{R}^{2n+1} \to \mathbb{R}^n$ be the homotopy $K = (K_1, K_2)$ with

$$\begin{aligned} K_1(\mathbf{x},\mathbf{y},\lambda) &:= G_1(\mathbf{x},\mathbf{y}-\lambda F(\mathbf{x})), \\ K_2(\mathbf{x},\mathbf{y},\lambda) &:= G_2(\mathbf{x},(1-\lambda)\mathbf{y}-\lambda F(\mathbf{x})). \end{aligned} \tag{3.22}$$

Now K is proper over $\mathbf{0}$ by an argument similar to that given for H. Also

$$K(\mathbf{x},\mathbf{y},0) = G(\mathbf{x},\mathbf{y}) \quad \text{and} \quad K(\mathbf{x},\mathbf{y},1) = (\mathbf{y}, -\tilde{G}(\mathbf{x})), \tag{3.23}$$

so we have $\deg(G) = \deg(K(\mathbf{x},\mathbf{y},1))$. Interchanging coordinates in $K(\mathbf{x},\mathbf{y},1)$ and multiplying by -1 does not change the absolute value of the degree, hence

$$|\deg(G)| = |\deg(K(\mathbf{x},\mathbf{y},1))| = |\deg \tilde{K}(\mathbf{x},\mathbf{y})|, \tag{3.24}$$

where

$$\tilde{K}(\mathbf{x},\mathbf{y}) = (\tilde{G}(\mathbf{x}),\mathbf{y}) = (\tilde{G}\times I)(\mathbf{x},\mathbf{y}). \tag{3.25}$$

Now $\deg(\tilde{K}(\mathbf{x}, \mathbf{y})) = \deg(\tilde{G}(\mathbf{x}))$, which proves that $|\deg(\tilde{G})| = |\deg(G)| = 1$. □

Theorem 3.2 applies to nearly all transistor circuits of practical interest. To verify the passivity condition (3.18), it suffices to check it on each nonlinear circuit element separately. For example, it holds for the Ebers–Moll model for bipolar junction transistors for all $\mathbf{c} \in \mathbb{R}^2$ as is shown in Sandberg and Willson [41, Theorem 5]; see Theorem A.1 in Appendix A. If there are nonlinear elements for which (3.18) does not hold, we may modify their responses for large $||\mathbf{x}||$ to force (3.18) to hold. In this way we obtain modified operating point equations that detect all the "physically relevant" dc operating points. We propose such model modifications purely as artificial adjustments to the transistor model, but actual transistors exhibit breakdown behavior which is roughly equivalent to a passivity property like (3.18).

The degree results above show that for most circuits there exist network equations having threading homotopies. Finding explicit threading homotopies for particular classes of network equations remains an open problem

One of the difficulties in using homotopy methods in circuit simulators to find all zeros is forcing properness of the homotopy, to prevent zeros "escaping to infinity." Trajković, Melville, and Fang [42] and Melville et al. [36] noted that this can be achieved for various circuits that have the "no-gain" property defined in Willson [49] and Chua, Lam, and Stromsoe [11]. A circuit has the *no-gain property* if, for any set of attached independent sources (either voltage or current sources), the voltage difference between any two nodes of the circuit does not exceed the absolute values of voltages across all the independent sources, and the current flowing into any node does not exceed the sum of the magnitudes of currents flowing through all the independent sources. In [49] it is shown that all connected networks composed of two-terminal and three-terminal no-gain elements have the no-gain property, and that linear resistors, bipolar junction transistors, and MOSFETS all have the no-gain property. Suppose that one can find a homotopy $H(\mathbf{x}, \lambda)$ for $0 \leq \lambda \leq 1$ with the following two properties.

(i) $H(\mathbf{x}, \lambda) = F_\lambda(\mathbf{x})$, where each $F_\lambda(\mathbf{x})$ is the operating point equation for a circuit C_λ that has the no-gain property, for $0 \leq \lambda \leq 1$.

(ii) $H(\mathbf{x}, 0) = F(\mathbf{x})$, while $H(\mathbf{x}, 1) = F_1(\mathbf{x})$ corresponds to a circuit with a unique operating point.

The no-gain property of all circuits C_λ then implies that the homotopy is proper. In this case it directly follows that $\deg(F) = 1$ from the invariance of degree for proper homotopies, because $\deg F_0(\mathbf{x}) = 1$ by (ii). Such "no-gain" homotopies can often be found by varying the parameters of the circuit elements, as described in [36]. The particular usefulness of such "no-gain" homotopies is to give a priori bounds on a region containing all zeros of such homotopies; see Trajković, Melville, and Fang [42]. These bounds provide a simple error check on correctness of homotopy computations.

There is a natural class of candidate homotopies to consider for use in circuit simulators, which we may call *sandwich homotopies*, that may well include threading homotopies. These homotopies are constructed using circuit deformation homotopies $\{H(\mathbf{x}, \lambda) : 0 \leq \lambda \leq 1\}$, which deform the circuit parameters of a no-gain circuit to obtain a circuit having a unique operating point. A *sandwich homotopy* consists of combining two circuit deformation homotopies which vary the circuit parameters in different ways, with one used on $0 \leq \lambda \leq 1$, and the other on $-1 \leq \lambda \leq 0$, and then we set $H(\mathbf{x}, \lambda) \equiv H(\mathbf{x}, 1)$ for $\lambda \geq 1$, and $H(\mathbf{x}, \lambda) = H(\mathbf{x}, -1)$ for $\lambda \leq -1$. Some care is needed to make such a homotopy C^2 at the boundary values $\lambda = 1$, 0, and -1.

We describe one kind of *circuit-deformation* homotopy, following the approach of Melville et al. [36], for circuits consisting of linear resistors and bipolar junction transistors. First, the coupling elements in the bipolar junction transistors, the forward and reverse gains,[3] are each reduced monotonically to zero. By results of Willson [49], the transistors produced during this process retain the no-gain property throughout. Now one has a network of uncoupled diodes whose $v - i$ curves are eventually monotone; i.e., $f'(x) > 0$ for $|x| > R$. The second part of the homotopy is to deform the voltage-current curves of the diodes to make them all monotone by a C^2 homotopy applied to each voltage-current curve on a bounded region. (The diode $v - i$ curves must satisfy some mild conditions for this to be possible. If $f'(x) > 0$ for $|x| \geq R$ then $f(-R) < 0 < f(R)$ suffices.) The resulting circuit of strictly monotone diodes has a unique operating point by a well-known result of Duffin [18, Theorem 3]. One wants such homotopies $H(\mathbf{x}, \lambda)$ to be bifurcation-free, i.e., for the rank n condition (iii) above to be satisfied. This can be done by allowing a space of small C^2 deformations around the homotopy described above, using the approach of Chow, Mallet-Paret, and Yorke [7]. These homotopies certify that $\deg(F_0) = 1$, because $\deg(F_1) = 1$ by the result of Duffin [18] and the homotopy can be shown to be proper using the no-gain condition.

Sandwich homotopies come with no guarantee of being threading homotopies. However, they have successfully been used to find more than one operating point; see Green and Melville [25]. In particular, Melville et al. [36] describe a variable-gain homotopy which seems to work well in practice and which has been implemented in `Sframe`, a circuit simulation platform; see [35]. Some of these homotopies have been observed empirically to have the threading property. Perhaps a subclass of them can be proved to have the threading property, using Diener's condition (1.2) or an analogous criterion.

Appendix A. Ebers–Moll model for bipolar junction transistors. The Ebers–Moll large signal model [4], [19], [22] for a bipolar junction transistor is pictured in Figure A.1. This is the injection version of the Ebers–Moll model given in Getreu [22, p. 12]. A *node* in a circuit designates a connected set of points in the circuit which are all at the same voltage with respect to a reference point, usually called ground. There are only three nodes in the Ebers–Moll model: *collector*, *base*, and *emitter*. In Figure A.1 the base node has been drawn as two terminals (b_1, b_2) in order to treat the transistor as a two-port; this arrangement is conventionally called the *common base configuration* for the Ebers–Moll model. This circuit element contains two nonlinear diodes with (different) response curves of the form

$$f(v) = m(e^{nv} - 1), \tag{A.1}$$

where m and n are both positive parameters. The *exponential diodes* (A.1) are sometimes called Ebers–Moll diodes. It also contains two current-controlled current sources with *current gains* α_F, α_R that satisfy $0 \leq \alpha_F, \alpha_R < 1$. The current flowing through a current-controlled current source is equal to a fixed current gain α times a controlling current I flowing on a branch somewhere else in the circuit. Thus a current-controlled current source is a linear element that produces coupling between different parts of the circuit. Figure A.1 models specifically an *npn* transistor; the model for a *pnp* transistor is obtained by systematically reversing the current flow throughout this model.

[3] These are the gain parameters α_F and α_R appearing in Appendix A.

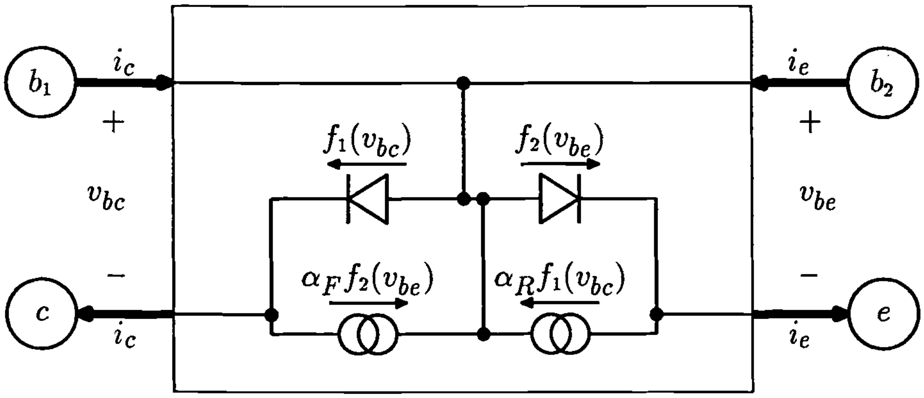


FIG. A.1. *Ebers–Moll model (common base configuration).*

This two-port can be viewed as a voltage-controlled two-port with the current responses

$$\begin{bmatrix} i_c \\ i_e \end{bmatrix} = \begin{bmatrix} 1 & -\alpha_F \\ -\alpha_R & 1 \end{bmatrix} \begin{bmatrix} f_1(v_{bc}) \\ f_2(v_{be}) \end{bmatrix}, \tag{A.2}$$

where v_{be} and v_{bc} are the branch voltages. For example, in Figure A.1 the current i_c flowing out of the two-port into the collector terminal is the sum of two components: a current $f_1(v_{bc})$ flowing in the same direction as i_c and a current $\alpha_F f_2(v_{bc})$ flowing in the opposite direction as i_c, in accordance with the minus sign in (A.2).

In (A.2) the exponential diodes are

$$f_1(v_1) = \tilde{I}_{cs}(e^{n_1 v_1} - 1), \tag{A.3}$$

and

$$f_2(v_2) = \tilde{I}_{es}(e^{n_2 v_2} - 1), \tag{A.4}$$

where $\tilde{I}_{cs}$ is a parameter called the collector-base saturation current, and $\tilde{I}_{es}$ is a parameter called the emitter-base saturation current. The quantities $n_1 = \frac{q}{\kappa T_1}$ and $n_2 = \frac{q}{\kappa T_2}$, in which q is the electron charge, κ is Boltzmann's constant, and T_1 and T_2 are the temperatures at the collector and emitter nodes, respectively. The temperatures are usually equal under normal operating conditions. The power consumed by the transistor is

$$P = i_c f_1(v_{bc}) + i_e f_2(v_{be}). \tag{A.5}$$

Sufficient conditions for such a transistor to be *passive* [23] are that

$$\alpha_F \leq \frac{\tilde{I}_{cs}}{\tilde{I}_{es}} \leq \frac{1}{\alpha_R} \quad \text{and} \quad \alpha_F \leq \frac{n_1}{n_2} \leq \frac{1}{\alpha_R}. \tag{A.6}$$

Sufficient conditions for such a transistor to satisfy the *no-gain condition* [47] are that

$$\alpha_F \leq \frac{\tilde{I}_{cs}}{\tilde{I}_{es}} \leq \frac{1}{\alpha_R} \quad \text{and} \quad n_1 = n_2. \tag{A.7}$$

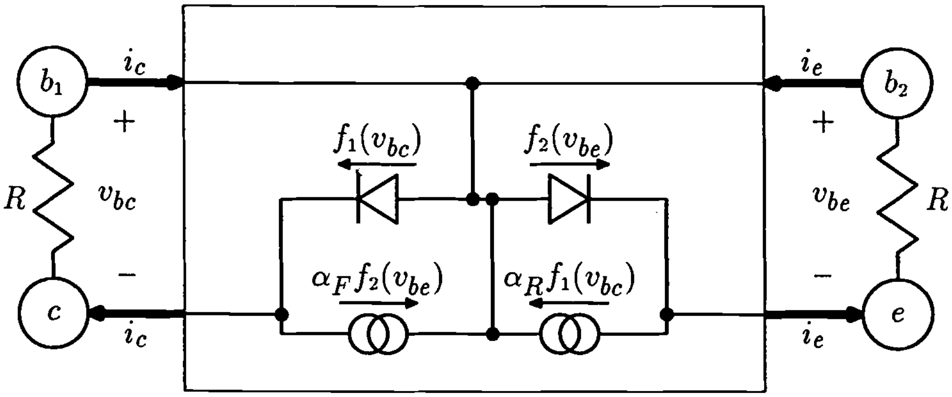


FIG. A.2. *Added shunt conductances (resistors).*

The conditions (A.7) hold under normal operation.

Sandberg and Willson [41, Theorem 5 and footnote 5] establish the following passivity property of Ebers–Moll bipolar junction transistors.

THEOREM A.1 (Sandberg and Willson). *Let* $0 < \alpha_1 < 1$ *and* $0 < \alpha_2 < 1$ *be given. Suppose that*

$$f_k(v_k) = m_k(\exp(n_k v_k) - 1) \quad for \quad 1 = 1, 2, \tag{A.8}$$

with $m_k n_k > 0$ *and with*

$$\alpha_1 \leq \frac{m_1}{m_2} \leq \frac{1}{\alpha_2} \quad and \quad \alpha_1 \leq \frac{n_1}{n_2} \leq \frac{1}{\alpha_2}. \tag{A.9}$$

Then, for any $(c_1, c_2) \in \mathbb{R}^2$*, the quantity*

$$P(v_1, v_2) = [v_1 \ v_2] \begin{bmatrix} 1 & -\alpha_1 \\ -\alpha_2 & 1 \end{bmatrix} \begin{bmatrix} f_1(v_1 + c_1) \\ f_2(v_2 + c_2) \end{bmatrix} \tag{A.10}$$

satisfies

$$\lim_{||\mathbf{v}|| \to \infty} P(v_1, v_2) = +\infty. \tag{A.11}$$

Detailed models for bipolar junction transistors (see [4], [12], [15], [22]) elaborate on the Ebers–Moll large signal model. In SPICE, additional conductances are added for stability in solving the algorithms, which amount to adding linear resistors with large resistances $R = (GMIN)^{-1}$ as pictured in Figure A.2.

Acknowledgments. We are indebted to M. Green for corrections and improvements to an early draft version of the paper, and to L. Trajković for comments and references concerning the no-gain condition. We thank S.-C. Fang, L. T. Watson, and the many referees for helpful comments.

REFERENCES

[1] E. Allgower and K. Georg, *Simplicial and continuation methods for approximating fixed points and solutions to systems of equations,* SIAM Rev., 22 (1980), pp. 28–85.

[2] E. Allgower and K. Georg, *Homotopy methods for approximating several solutions to nonlinear systems of equations*, in Numerical Solution of Highly Nonlinear Problems, W. Forster, ed., North-Holland, Amsterdam, 1980, pp. 253–270.

[3] E. Allgower and K. Georg, *Numerical Continuation Methods: An Introduction,* Springer-Verlag, New York, 1990.

[4] P. Antognetti and G. Massobrio, eds., *Semiconductor Device Modelling with SPICE*, McGraw–Hill, New York, NY, 1988.

[5] F. H. Branin, *Widely convergent method for finding multiple solutions of simultaneous nonlinear equations*, IBM J. Res. Develop., 16 (1972), pp. 504–522.

[6] K. S. Chao and R. Saeks, *Continuation methods in circuit analysis*, Proc. IEEE, 65 (1977), pp. 1187–1194.

[7] S. Chow, J. Mallet-Paret, and J. A. Yorke, *Finding zeros of maps: Homotopy methods that are constructive with probability one*, Math. Comp., 32 (1978), pp. 887–899.

[8] S. Chow, J. Mallet-Paret, and J. A. Yorke, *A homotopy method for locating all zeros of a system of polynomials*, in Functional Differential Equations and Approximation of Fixed Points, Lecture Notes in Math. 730, H. O. Peitgen and H. O. Walter, eds., Springer-Verlag, New York, 1979, pp. 77–88.

[9] J. Cronin, *Fixed Points and Topological Degree in Nonlinear Analysis,* American Math. Society, Providence, RI, 1964.

[10] L. O. Chua, C. S. Desoer, and E. S. Kuh, *Linear and Nonlinear Circuits,* McGraw–Hill, New York, 1987.

[11] L. O. Chua, Y. F. Lam, and K. A. Stromsoe, *Qualitative properties of resistive networks containing multiterminal nonlinear elements: No gain properties*, IEEE Trans. Circuits Systems, 24 (1977), pp. 93–117.

[12] L. O. Chua and A. Ushida, *A parameter-switching algorithm for finding multiple solutions of nonlinear resistive circuits*, Internat. J. Circuit Theory Appl., 4 (1976), pp. 215–239.

[13] L. O. Chua and N. N. Wang, *On the application of degree theory to the analysis of resistive nonlinear networks*, Internat. J. Circuit Theory Appl., 5 (1977), pp. 35–68.

[14] L. O. Chua and R. L. P. Ying, *Finding all solutions and piecewise linear circuits*, Internat. J. Circuit Theory Appl., 10 (1982), pp. 201–229.

[15] H. C. de Graaff and F. M. Klassen, *Compact Transistor Modelling for Circuit Design*, Springer-Verlag, New York, 1990.

[16] I. Diener, *On the global convergence of path-following methods to determine all solutions of a system of nonlinear equations*, Math. Programming, 39 (1987), pp. 181–188.

[17] F. J. Drexler, *A homotopy method for the calculation of all the zeros of zero-dimensional polynomial ideals*, in Continuation Methods, H. Wacker, ed., Academic Press, New York, 1978.

[18] R. L. Duffin, *Nonlinear networks* IIa, Bull. Amer. Math. Soc., 53 (1947), pp. 963–971.

[19] J. J. Ebers and J. L. Moll, *Large-signal behavior of junction transistors*, Proc. of the I.R.E., 42 (1954), pp. 1761–1772.

[20] C. B. Garcia and W. I. Zangwill, *Finding all solutions of polynomial systems and other systems of equations*, Math. Programming, 16 (1979), pp. 159–176.

[21] C. B. Garcia and W. I. Zangwill, *Pathways to Solutions, Fixed Points and Equilibria,* Prentice–Hall, Englewood Cliffs, NJ, 1981.

[22] I. Getreu, *Modelling the Bipolar Transistor*, Tektronix Inc., Beaverton, OR, 1976.

[23] B. Gopinath and D. Mitra, *When is a transistor passive?*, Bell System Tech. J., 50 (1971), pp. 2835–2847.

[24] M. M. Green, *How to identify unstable dc operating points*, IEEE Trans. Circuits Systems I. Fund. Theory Appl., 39 (1992), pp. 820–832.

[25] M. M. Green and R. C. Melville, *Sufficient conditions for finding multiple operating points of dc circuits using continuation methods*, in Proc. ISCAS 1995, Seattle, WA, Vol. I, IEEE Press, Piscataway, NJ, pp. 117–121.

[26] M. M. Green and A. N. Willson, Jr., *(Almost) half of any circuit's operating points are unstable*, IEEE Trans. Circuits Systems I. Fund. Theory Appl., 41 (1994), pp. 286–293.

[27] V. Guillemin and A. Pollack, *Differential Topology,* Prentice–Hall, Englewood Cliffs, NJ, 1974.

[28] M. Hirsch, *Differential Topology,* Springer-Verlag, New York, 1976.

[29] C. W. Ho, A. E. Ruehli, and P. A. Brennan, *The modified nodal approach to network analysis*, IEEE Trans. Circuits Systems, 22 (1975), pp. 678–687.

[30] Q. Huang and R. W. Liu, *A simple algorithm for finding all solutions of piecewise-linear networks*, IEEE Trans. Circuits Systems, 36 (1989), pp. 600–609.

[31] J. Jezierski, *One codimensional Wecken type theorems*, Forum Math., 5 (1993), pp. 421–439.

[32] R. C. Kirby and L. C. Siebenmann, *Foundational Essays on Topological Manifolds, Smoothings and Triangulations,* Ann. of Math. Stud. 88, Princeton University Press, Princeton, NJ, 1977.

[33] M. Kojima, H. Nishino, and N. Arima, *A PL homotopy for finding all the roots of a polynomial*, Math. Programming, 16 (1979), pp. 37–62.

[34] W. Mathis and G. Wettlaufer, *Finding all DC-equilibrium-points of nonlinear circuits*, Proc. 32nd Midwest Sym. on Circuits and Systems, Urbana, IL, Vol. I, IEEE Press, Piscataway, NJ, 1989, pp. 462–465.

[35] R. Melville, S. Moinian, P. Feldmann, and L. Watson, *Sframe: An efficient system for detailed dc simulation of bipolar analog integrated circuits using continuation methods*, Analog Integrated Circuits and Signal Processing, 3 (1993), pp. 163–180.

[36] R. C. Melville, L. Trajković, S.-C. Fang, and L. T. Watson, *Artificial parameter convergent homotopy methods for the dc operating point problem*, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 6 (1993), pp. 861–877.

[37] J. W. Milnor, *Topology from the Differentiable Viewpoint,* The University of Virginia Press, Charlottesville, VA, 1965.

[38] T. Ohtsuki, T. Fujisawa, and S. Kumagai, *Existence theorems and a solution algorithm for piecewise-linear resistor networks*, SIAM J. Math. Anal., 8 (1977), pp. 69–99.

[39] S. Pastore and A. Premoli, *Polyhedral elements: A new algorithm for capturing all the equilibrium points of piecewise-linear circuits*, IEEE Trans. Circuits Systems I Fund. Theory Appl., 40 (1993), pp. 129–132.

[40] I. Sandberg and A. N. Willson, Jr., *Some theorems on properties of dc equations of nonlinear networks*, Bell System Tech. J., 48 (1969), pp. 1–34.

[41] I. Sandberg and A. N. Willson, Jr., *Existence of solution for the equations of transistor-resistor-voltage source networks*, IEEE Trans. Circuit Systems, 18 (1970), pp. 619–625.

[42] L. Trajković, R. C. Melville, and S. C. Fang, *Passivity and no gain properties establish global convergence of a homotopy method for dc operating points*, Proc. IEEE Internat. Sym. Circuits Systems, New Orleans, LA, 1990, pp. 914–917.

[43] L. Trajković, R. C. Melville, and S. C. Fang, *Finding dc operating points of transistor circuits using homotopy methods*, Proc. IEEE Internat. Conf. on Circuits and Systems, Singapore, 1991, pp. 758–761.

[44] L. Trajković and A. N. Willson, Jr., *Theory of dc operating points of transistor networks*, Internat. J. Electron. Comm., 46 (1992), pp. 228–241.

[45] L. Vandenberghe, B. L. de Moor, and J. Vandewalle, *The generalized linear complementarity problem applied to the complete analysis of piecewise linear resistive circuits*, IEEE Trans. Circuits Systems, 36 (1989), pp. 1382–1391.

[46] H. Whitney, *The self-intersection of a smooth n-manifold in 2n-space*, Ann. of Math., 48 (1944), pp. 220–246.

[47] A. N. Willson, Jr., *New theorems on the equations of nonlinear dc transistor networks*, Bell System Tech. J., 49 (1970), pp. 1713–1738.

[48] A. N. Willson, Jr., *Nonlinear Networks: Theory and Analysis*, IEEE Press, New York, 1974.

[49] A. N. Willson, Jr., *The no-gain property for networks containing three-terminal elements*, IEEE Trans. Circuits Systems, 22 (1975), pp. 678–687.

[50] A. N. Willson, Jr. and J. Wu, *Existence criteria for dc solutions of nonlinear networks which involve the independent sources*, IEEE Trans. Circuits Systems, 31 (1984), pp. 952–959.

[51] D. Wolf and S. Sanders, *Multi-parameter homotopy methods for finding dc operating points of nonlinear circuits*, IEEE Trans. Circuits Systems I Fund. Theory Appl., 43 (1996), pp. 824–838.

[52] F. F. Wu, *Existence of an operating point for a nonlinear circuit using the degree of a mapping*, IEEE Trans. Circuits Systems, 21 (1974), pp. 671–677.

[53] J. L. Wyatt, Jr., L. O. Chua, J. W. Gannett, I. C. Göknar, and D. N. Green, *Energy concepts in the state-space theory of nonlinear n-ports: Part* I *– passivity*, IEEE Trans. Circuits Systems, 28 (1981), pp. 48–61.

[54] K. Yamamura, *Finding all solutions of piecewise linear resistive circuits using simple sign tests*, IEEE Trans. Circuits Systems I Fund. Theory Appl., 40 (1993), pp. 546–551.

SIAM J. OPTIM.
Vol. 9, No. 1, pp. 179–189

WEAK SHARP SOLUTIONS OF VARIATIONAL INEQUALITIES*

PATRICE MARCOTTE† AND DAOLI ZHU‡

Abstract. In this work we give sufficient conditions for the finite convergence of descent algorithms for solving variational inequalities involving generalized monotone mappings.

Key words. sharp solution, variational inequality, descent algorithm, generalized monotonicity

AMS subject classifications. 90C33, 49M99

PII. S1052623496309867

1. Introduction.

Recently, Burke and Ferris [5] introduced sufficient conditions for the finite identification, by iterative algorithms, of local minima associated with mathematical programs. To this aim, they introduced the notion of a weak sharp minimum, which extends the notion of a sharp or strongly unique minimum to mathematical programs admitting nonisolated local minima. In our work, we extend their results and those of Al-Khayyal and Kyparisis [1] to generalized monotone variational inequalities and provide a characterization of their solution sets. Our work is also closely related to that of Patriksson [14], who analyzed the finite convergence of approximation algorithms for solving monotone variational inequalities under a sharpness assumption.

The paper is organized as follows. Section 2 introduces the main definitions. In section 3, we reformulate the variational inequality problem (VIP) as a convex program and show that its objective is continuously differentiable at any solution of the VIP, under a regularity assumption. In section 4 we introduce the notion of weak sharpness for the VIP and derive a necessary and sufficient condition for a solution set to be weakly sharp. Finally, section 5 addresses the finite convergence of iterative algorithms for solving variational inequalities whose solution set is weakly sharp.

2. Notation and definitions.

Let X denote a nonempty, closed, and convex subset of R^n and let F be a mapping from X into R^n. We consider the VIP that consists of finding a vector $x^* \in X$ that satisfies the variational inequality:

$$\langle F(x^*), x - x^* \rangle \geq 0 \qquad \forall x \in X, \tag{2.1}$$

where $\langle x, y \rangle$ denotes the Euclidean inner product of two vectors in R^n. Throughout the paper, we will denote by X^* the set of solutions of the variational inequality (2.1). If X is a subset of R^n, its *polar set* X° is defined as

$$X^\circ := \{y \in R^n : \langle y, x \rangle \leq 0 \quad \forall x \in X\}.$$

*Received by the editors September 27, 1996; accepted for publication (in revised form) September 2, 1997; published electronically December 2, 1998. This research was supported by NSERC (Canada) and FCAR (Québec).

http://www.siam.org/journals/siopt/9-1/30986.html

†DIRO, Université de Montréal, C.P. 6128, succursale Centre-Ville, Montréal, Québec, H3C 3J7 Canada (marcotte@iro.umontreal.ca).

‡CRT, Université de Montréal, C.P. 6128, succursale Centre-Ville, Montréal, Québec, H3C 3J7 Canada (daoli@crt.umontreal.ca).

We denote by int(C) the interior of a set C. The *projection* of a point $x \in R^n$ onto the set X is defined as

$$\mathrm{proj}_X(x) := \arg\min_{y \in X} \|x - y\|.$$

If X is a convex set, its *normal cone* at x is

$$N_X(x) := \begin{cases} \{y \in R^n : \langle y, z - x\rangle \le 0 \quad \forall z \in X\} & \text{if } x \in X, \\ \emptyset & \text{otherwise,} \end{cases} \tag{2.2}$$

and its *tangent cone* at x is $T_X(x) := [N_X(x)]^\circ$. Using this notation, a vector x^* is a solution of the VIP if and only if

$$-F(x^*) \in N_X(x^*) \tag{2.3}$$

or, equivalently,

$$\mathrm{proj}_{T_X(x^*)}(-F(x^*)) = 0. \tag{2.4}$$

A mapping F from a convex set X into R^n is *monotone* on X if, $\forall x$, y in X,

$$\langle F(y) - F(x), y - x\rangle \ge 0. \tag{2.5}$$

It is *strongly monotone* on X if there exists a positive number α such that, $\forall x$, y in X,

$$\langle F(y) - F(x), y - x\rangle \ge \alpha\|y - x\|^2. \tag{2.6}$$

It is *pseudomonotone* on X if, $\forall x$, y in X,

$$\langle F(x), y - x\rangle \ge 0 \quad \Rightarrow \quad \langle F(y), y - x\rangle \ge 0. \tag{2.7}$$

It is *strongly pseudomonotone* on X if there exists a positive number β such that, $\forall x$, y in X,

$$\langle F(x), y - x\rangle \ge 0 \quad \Rightarrow \quad \langle F(y), y - x\rangle \ge \alpha\|y - x\|^2. \tag{2.8}$$

It is *monotone*$^+$ on C if F is monotone and for every pair of points x, y in X,

$$\langle F(y) - F(x), y - x\rangle = 0 \quad \Rightarrow \quad F(y) = F(x). \tag{2.9}$$

It is *pseudomonotone*$^+$ on X if F is pseudomonotone and, $\forall x$, y in X,

$$\langle F(x), y - x\rangle \ge 0 \quad \text{and} \quad \langle F(y), y - x\rangle = 0 \quad \Rightarrow \quad F(y) = F(x). \tag{2.10}$$

It is *quasimonotone* on X if, $\forall x$, y in X,

$$\langle F(x), y - x\rangle > 0 \quad \Rightarrow \quad \langle F(y), y - x\rangle \ge 0. \tag{2.11}$$

Several results concerning mappings satisfying the above monotonicity or generalized monotonicity conditions can be found in Schaible [15] and Zhu and Marcotte [11, 17]. Finally, a mapping F from the set X into R^n is *Lispchitz continuous* on X, with Lipschitz constant L, if, $\forall x$, y in X,

$$\|F(x) - F(y)\| \le L\|x - y\|. \tag{2.12}$$

3. The dual gap function for the pseudomonotone VIP. If the mapping F is pseudomonotone, then the solution set of the VIP can be characterized as the intersection of half-spaces, i.e., x^* is a solution of the VIP if and only if it satisfies

$$\langle F(x), x - x^* \rangle \geq 0 \qquad \forall x \in X. \tag{3.1}$$

It follows that the solution set of the VIP is closed and convex. The proof of this result is identical to that given in Auslender [2] for monotone variational inequalities. Note that we cannot substitute quasimonotonicity for pseudomonotonicity in (3.1), as shown by the VIP involving the quasimonotone function $F(x) = x^2$ and the set $X = R$. We define the *dual gap function* $G(x)$ associated with the VIP as

$$\begin{aligned} G(x) &= \max_{z \in X} \langle F(z),\ x - z \rangle \\ &= \langle F(\tilde{y}), x - \tilde{y} \rangle, \end{aligned} \tag{3.2}$$

where $\tilde{y}$ is any point in the set $\Lambda(x) := \arg\max_{z \in X} \langle F(z), x - z \rangle$.

Since the function G is the pointwise supremum of affine functions, it is closed and convex on X. Moreover, G is nonnegative and achieves its minimum value (zero) only at points of X that satisfy the original variational inequality. Thus, any solution of the VIP is a global minimum for the convex optimization program

$$\min_{x \in X} G(x). \tag{3.3}$$

If F is pseudomonotone$^+$, the dual gap function G enjoys the nice properties given in the theorem below.

THEOREM 3.1. *Let F be continuous and pseudomonotone$^+$ on X. Then*

(i) *F is constant over X^*;*

(ii) *for any x^* in X^*, F is constant and equal to $F(x^*)$ over $\Lambda(x^*)$;*

(iii) *$\Lambda(x^*) = X^*$ for any x^* in X^*;*

(iv) *if X is compact, then G is continuously differentiable over X^*, and $\nabla G(x^*) = F(x^*)$ $\forall x^*$ in X^*.*

Proof. (i) Let x^* and x^{**} be any two solutions of the VIP. It follows from (2.1) and (3.1) that

$$\begin{aligned} &\langle F(x^{**}), x^{**} - x^* \rangle \geq 0, \\ &\langle F(x^{**}), x^{**} - x^* \rangle \leq 0, \end{aligned}$$

from which we deduce

$$\langle F(x^{**}), x^{**} - x^* \rangle = 0.$$

Now, the inequality

$$\langle F(x^*), x^{**} - x^* \rangle \geq 0,$$

together with the pseudomonotonicity$^+$ of F, yields $F(x^*) = F(x^{**})$.

(ii) For every $x^* \in X^*$ and y^* in $\Lambda(x^*)$,

$$G(x^*) = \langle F(y^*), x^* - y^* \rangle = 0 \tag{3.4}$$

holds. Now, $\langle F(x^*), y^* - x^* \rangle \geq 0$ (x^* is a solution of the VIP) and $\langle F(y^*), y^* - x^* \rangle = 0$ imply, by the pseudomonotonicity$^+$ of F, that $F(x^*) = F(y^*)$.

(iii) Let $\tilde{x} \in \Lambda(x^*)$. From (ii) we have that $F(x^*) = F(\tilde{x})$ and $\langle F(\tilde{x}), x^* - \tilde{x}\rangle = 0$. This implies that, for any y in X,

$$\langle F(\tilde{x}), \tilde{x} - y\rangle = \langle F(\tilde{x}), \tilde{x} - x^*\rangle + \langle F(\tilde{x}), x^* - y\rangle \tag{3.5}$$

$$= \langle F(x^*), x^* - y\rangle \tag{3.6}$$

$$\leq 0 \tag{3.7}$$

and $\tilde{x}$ is in X^*. Conversely, for any x^* in X^*,

$$\tilde{x} \in X^* \Rightarrow \langle F(\tilde{x}), x^* - \tilde{x}\rangle = 0 \tag{3.8}$$

$$\Rightarrow \tilde{x} \in \Lambda(x^*). \tag{3.9}$$

(iv) From a result of Danskin [4] the derivative of G at x^* in the direction d is given by the expression

$$\begin{aligned} G'(x^*; d) &= \max\{\langle F(y), d\rangle : y \in \Lambda(x^*)\} \\ &= \langle F(x^*), d\rangle, \end{aligned}$$

since, by (ii), F is constant and equal to $F(x^*)$ over $\Lambda(x^*)$. Thus, G is continuously differentiable at every point $x^* \in X^*$, with gradient $\nabla G(x^*) = F(x^*)$. □

4. Sharp solutions of variational inequalities. Recently, in the context of convex smooth optimization, Burke and Ferris [5] have extended the notion of a strongly unique solution to optimization problems whose solution set is not necessarily a singleton. To this aim, they introduced the notion of a weak sharp solution for a convex minimization problem. We recall that the solution set X^* is *weakly sharp* for the program $\min_{x \in X} f(x)$ if there exists a positive number α (*modulus* of sharpness) such that

$$f(x) \geq f(x^*) + \alpha \operatorname{dist}(x, X^*) \qquad \forall x^* \in X^*, \tag{4.1}$$

where $\operatorname{dist}(x, X^*) := \min_{x^* \in X^*} \|x - x^*\|$. These authors proved that if f is a closed, proper, and convex function and if the sets X and X^* are nonempty, closed, and convex, then the solution set of the convex optimization program (4.1) is weakly sharp if and only if the geometric condition

$$-\nabla f(x^*) \in \operatorname{int}\left(\bigcap_{x \in X^*} [T_X(x) \cap N_{X^*}(x)]^\circ\right) \qquad \forall x^* \in X^* \tag{4.2}$$

holds. Since the VIP lacks a "natural" objective function, it is natural to define weak sharpness of the solution set of a variational inequality with reference to (4.2). Precisely, following Patriksson [14], we say that the solution set of the VIP is *weakly sharp* if we have, for any x^* in X^*,

$$-F(x^*) \in \operatorname{int}\left(\bigcap_{x \in X^*} [T_X(x) \cap N_{X^*}(x)]^\circ\right). \tag{4.3}$$

Alternatively, one could have defined weak sharpness with respect to an "artificial" convex programming reformulation of the VIP. If F is pseudomonotone, an obvious choice for such a reformulation is the one based on the dual gap function defined earlier (see (3.2) and (3.3)). This would have led to the definition

$$G(x) \geq \alpha \operatorname{dist}(x, X^*) \tag{4.4}$$

$\forall x$ in X. If this condition is fulfilled, the function G provides an error bound for the distance from a feasible point to the set of solutions to the VIP. The constant α is again called the *modulus* of sharpness for the solution set X^*. Note that the very evaluation of G at a point x requires the solution of a possibly nonconvex mathematical program.

From this point on, we will adopt the geometric condition (4.3) as the definition of weak sharpness and show that both definitions are actually equivalent whenever F is pseudomonotone$^+$.

THEOREM 4.1. *Let F be continuous and pseudomonotone$^+$ over the compact set X. Let the solution set X^* of the VIP be nonempty. Then X^* is weakly sharp if and only if there exists a positive number α such that*

$$G(x) \geq \alpha \operatorname{dist}(x, X^*)$$

$\forall x$ in X.

Proof. Let B denote the unit ball in R^n. We first prove that the inclusion

$$\alpha B \subset F(x^*) + [T_X(x^*) \cap N_{X^*}(x^*)]^\circ \tag{4.5}$$

holds at $x^* \in X^*$ if and only if we have

$$\langle F(x^*), z\rangle \geq \alpha \|z\| \qquad \forall z \in T_X(x^*) \cap N_{X^*}(x^*). \tag{4.6}$$

Indeed, if (4.5) holds, then for every $y \in B$, we have

$$\alpha y - F(x^*) \in [T_X(x^*) \cap N_{X^*}(x^*)]^\circ. \tag{4.7}$$

Thus, for every $z \in [T_X(x^*) \cap N_{X^*}(x^*)]$, we have $\langle \alpha y - F(x^*), z\rangle \leq 0$. Taking $y = z/\|z\|$ in the above inequality, we obtain (4.6).

Now assume that (4.6) holds. Then there exists a positive number α such that, for $x^* \in X^*$, $y \in B$, and $z \in T_X(x^*) \cap N_{X^*}(x^*)$,

$$\begin{aligned}\langle -F(x^*) + \alpha y, z\rangle &= \langle -F(x^*), z\rangle + \alpha\langle y, z\rangle \\ &\leq \langle -F(x^*), z\rangle + \alpha\|y\|\,\|z\| \\ &\leq \langle -F(x^*), z\rangle + \alpha\|z\| \\ &\leq 0 \qquad \text{by (4.6).}\end{aligned}$$

This implies that (4.5) holds as well.

If $-F(x^*) \in \operatorname{int}\left(\bigcap_{x \in X^*}[T_X(x) \cap N_{X^*}(x)]^\circ\right)$ $\forall x^*$ in X^*, then there must exist a positive number α such that (4.5) is satisfied for every $x^* \in X^*$. From the above derivation, we have that $\langle F(x^*), z\rangle \geq \alpha\|z\|$ for every z in $T_X(x^*) \cap N_{X^*}(x^*)$. Now set, for x in X, $\bar{x} = \operatorname{proj}_{X^*}(x)$. Clearly, $x - \bar{x} \in T_X(\bar{x}) \cap N_{X^*}(\bar{x})$ and there follows

$$\langle F(\bar{x}), x - \bar{x}\rangle \geq \alpha\|x - \bar{x}\| = \alpha \operatorname{dist}(x, X^*).$$

Since G is a convex function, differentiable at $\bar{x} \in X^*$, we have

$$\begin{aligned}G(x) &= G(x) - G(\bar{x}) \\ &\geq \langle \nabla G(\bar{x}), x - \bar{x}\rangle \\ &= \langle F(\bar{x}), x - \bar{x}\rangle \\ &\geq \alpha \operatorname{dist}(x, X^*).\end{aligned}$$

Conversely, let X^* satisfy (4.4) for some positive number α and let x^* be a point in X^*. If $T_X(x^*) \cap N_{X^*}(x^*) = \{0\}$, then $[T_X(x^*) \cap N_{X^*}(x^*)]^\circ = R^n$ and $\alpha B \subset F(x^*) + [T_X(x^*) \cap N_{X^*}(x^*)]^\circ$, trivially. Otherwise, let d be a nonzero vector in $T_X(x^*) \cap N_{X^*}(x^*)$. For any $y^* \in X^*$, we have

$$\begin{aligned} \langle d, y^* - x^* \rangle &\geq 0 \quad \text{since } d \in T_X(x^*), \\ \langle d, y^* - x^* \rangle &\leq 0 \quad \text{since } d \in N_{X^*}(x^*). \end{aligned}$$

Those inequalities imply that $\langle d, y^* - x^* \rangle = 0$, and X^* is a subset of a hyperplane H_d orthogonal to d. Let $\{d^k\}$ be a sequence converging to d such that $x^* + t_k d^k \in X$ for some sequence of positive numbers $\{t_k\}$. We can write

$$\begin{aligned} \operatorname{dist}(x^* + t_k d^k, X^*) &\geq \operatorname{dist}(x^* + t_k d^k, H_d) \\ &= \frac{t_k \langle d, d^k \rangle}{\|d\|}. \end{aligned}$$

Since X^* satisfies (4.4) with modulus α, we obtain

$$G(x^* + t_k d^k) \geq \alpha \operatorname{dist}(x^* + t_k d^k, X^*) \geq \alpha t_k \frac{\langle d, d^k \rangle}{\|d\|}$$

and

$$(G(x^* + t_k d^k) - G(x^*))/t_k \geq \alpha \frac{\langle d, d^k \rangle}{\|d\|}.$$

Taking the limit as $t_k \to 0$ and $d^k \to d$ leads to

$$\langle \nabla G(x^*), d \rangle \geq \alpha \|d\|$$

$\forall d$ in $T_X(x^*) \cap N_{X^*}(x^*)$. Therefore, for any w in B,

$$\begin{aligned} \langle \alpha w - F(x^*), d \rangle &= \langle \alpha w, d \rangle - \langle \nabla G(x^*), d \rangle \\ &\leq \alpha \|d\| - \alpha \|d\| \\ &= 0, \end{aligned}$$

and it follows that $\alpha B \subset F(x^*) + [T_X(x^*) \cap N_{X^*}(x^*)]^\circ$. Since F is constant over X^*, we conclude that (4.3) holds. □

We now show, by means of an example, that pseudomonotonicity of F is too weak a condition for the above result to hold. Indeed, consider the variational inequality defined by the two-dimensional mapping $F(x) = (-x_2, 2x_1)$ and the set $X = \{0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1\}$. One can check that the mapping F is pseudomonotone but not pseudomonotone$^+$ on X. Indeed, F is not constant over its solution set $X^* = \{x \in X : x_2 = 0\}$, in contradiction with the first statement of Theorem 3.1. We have

$$\begin{aligned} G(x) &= \max_{y \in X} \langle (-y_2, 2y_1), (x_1 - y_1, x_2 - y_2) \rangle \\ &= \max_{y \in X} -x_1 y_2 - y_1 y_2 + 2x_2 y_1 \\ &= 2x_2 \\ &= 2 \operatorname{dist}(x, X^*), \end{aligned}$$

and X^* satisfies (4.4) with modulus $\alpha = 2$. However, for any x^* in X^*, we have $[T_X(x^*) \cap N_{X^*}(x^*)]^\circ = \{x_2^* \leq 0\}$. Consequently $-F(x^*)$ does not lie inside

$$\bigcap_{x^* \in X^*} [T_X(x^*) \cap N_{X^*}(x^*)]^\circ$$

for any x^* in the solution set X^*, and the solution set X^* is not weakly sharp.

Our second characterization of weak sharpness involves the notion of *minimum principle sufficiency* introduced by Ferris and Mangasarian [6]. Consider the reformulation of the VIP as the (possibly nonconvex and/or nonsmooth) optimization problem $\min_{x \in X} g(x)$, where the *primal gap function* g is defined as

$$g(x) := \max_{y \in X} \langle F(x), x - y \rangle, \tag{4.8}$$

and let

$$\begin{aligned} \Gamma(x) &:= \arg\max_{y \in X} \langle F(x), x - y \rangle \\ &= \arg\min_{y \in X} \langle F(x), y \rangle. \end{aligned}$$

We say that the VIP possesses the *minimum principle sufficiency* (MPS) property if $\Gamma(x^*)$ coincides with the solution set X^*, for every x^* in X^*.

THEOREM 4.2. *Assume that F is continuous on X and that the set*

$$K := \operatorname{int} \left(\bigcap_{x \in X^*} [T_X(x) \cap N_{X^*}(x)]^\circ \right)$$

is nonempty. Then, for each z in K, one has that $\arg\max\{\langle z, y \rangle : y \in X\} \subset X^*$. Moreover, if F is pseudomonotone and $-F(x^*) \in K$ for every $x^* \in X^*$, then the VIP possesses the MPS property.

Proof. Let $x \in X$, $x \notin X^*$, and $\bar{x} = \operatorname{proj}_{X^*}(x)$. We have that $x - \bar{x} \in T_X(\bar{x}) \cap N_{X^*}(\bar{x})$, and, for any given z in K, there exists a positive number δ such that $\langle z + w, x - \bar{x} \rangle < 0$ $\forall w$ in δB. Thus,

$$\langle z, x \rangle < \langle z, \bar{x} \rangle - \delta \|x - \bar{x}\|;$$

i.e., $x \notin \arg\max\{\langle z, y \rangle : y \in X\}$, which brings about the conclusion by contradiction.

Next, let $-F(x^*) \in K$ for $x^* \in X^*$. In the first part of the proof, it has been established that

$$\arg\max\{\langle -F(x^*), y \rangle : y \in X\} \subset X^*.$$

Let $\hat{x}$ be in X^*. We have, as before, $\langle F(x^*), \hat{x} - x^* \rangle = 0$. Now, for any y in X,

$$\begin{aligned} \langle F(x^*), \hat{x} - y \rangle &= \langle F(x^*), \hat{x} - x^* \rangle + \langle F(x^*), x^* - y \rangle \\ &\leq 0. \end{aligned}$$

Therefore, $\hat{x} \in \Gamma(x^*)$ and $X^* \subset \Gamma(x^*)$. By gathering the two preceding inclusions, we conclude that $\arg\max\{\langle -F(x^*), y \rangle : y \in X\} = X^*$, as claimed. □

THEOREM 4.3. *Let F be pseudomonotone$^+$ and continuous on the compact polyhedral set X. Then the VIP possesses the MPS property if and only if it is weakly sharp, i.e., $X^* = \Gamma(x^*) = \Lambda(x^*)$.*

Proof. The "if" part of the statement is a consequence of Theorem 4.2. To prove the converse, first observe that the solution set $\Gamma(x^*) = X^*$ of the linear program

$$\min_{x\in X}\langle F(x^*), x\rangle$$

is weakly sharp (see appendix in Mangasarian and Meyer [10], for instance) with positive modulus α, and that α only depends on the constant vector $F(x^*)$ and X. We develop

$$\begin{array}{rcll} G(x) & = & \max_{y\in X}\langle F(y), x-y\rangle & \\ & \geq & \langle F(x^*), x-x^*\rangle & \forall x^*\in X^* \\ & = & \langle F(x^*), x-\hat{x}\rangle & \forall \hat{x}\in\Gamma(x^*) \\ & \geq & \alpha\|x-\mathrm{proj}_{\Gamma(x^*)}(x)\| & \\ & = & \alpha\|x-\mathrm{proj}_{X^*}(x)\| & \\ & \geq & \alpha\,\mathrm{dist}(x, X^*), & \end{array}$$

and from Theorem 4.1, X^* is weakly sharp. □

5. Finite convergence of algorithms for solving the VIP. In this section we will derive finite convergence results for classes of algorithms under the condition that the solution set of the VIP be weakly sharp. The first such result generalizes a result of Al-Khayyal and Kyparisis [1] to the case where the solution set is not necessarily a singleton.

THEOREM 5.1. *Let F be continuous and pseudomonotone$^+$ over the set X, and let the solution set X^* of the VIP be weakly sharp. Also let $\{x^k\}$ be a sequence in R^n. If either*

(i) *the sequence $\{\mathrm{dist}(x^k, X^*)\}$ converges to zero and the mapping F is uniformly continuous on an open set containing the sequence $\{x^k\}$ and the set X^*, or*

(ii) *the sequence $\{x^k\}$ converges to some $x^*\in X^*$,*

then there exists a positive integer k_0 such that, for any index $k\geq k_0$, any solution of the linear program

$$\min_{x\in X}\langle F(x^k), x\rangle \tag{5.1}$$

is a solution of the VIP.

Proof. First assume that (i) holds. From Theorem 3.1, $-F(x^*)$ is constant over X^* and there must exist a uniform positive constant α such that

$$-F(x^*) + \alpha B \in \bigcap_{x\in X^*}[T_X(x)\cap N_{X^*}(x)]^\circ \tag{5.2}$$

for every x^* in X^*. Since F is uniformly continuous and $\mathrm{dist}(x^k, X^*)\to 0$, there exists an integer k_0 such that

$$\|F(x^k) - F(x^*)\| < \alpha \quad \forall k\geq k_0,$$

i.e., $-F(x^k)\in \mathrm{int}\,(\bigcap_{x\in X^*}[T_X(x)\cap N_{X^*}(x)]^\circ)$. Therefore, by Theorem 4.2,

$$\arg\min_{x\in X}\langle F(x^k), x\rangle \subset X^*.$$

Under condition (ii) the result (5.2) is still valid for every $x^*\in X^*$, and we obtain the result as a consequence of the convergence of the sequence $\{\|F(x^k)-F(x^*)\|\}_k$ to zero. □

If Ω is a nonempty, closed, and convex subset of X, Burke and Ferris [5] have proved the inclusion

$$\Omega + \bigcap_{x\in\Omega} [T_X(x^*) \cap N_\Omega(x)]^\circ \subset \bigcup_{x\in\Omega} [x + N_X(x)]. \tag{5.3}$$

We will now use this result to provide a geometric characterization of sequences that achieve the finite identification of a solution to the VIP.

Theorem 5.2. *Let F be pseudomonotone$^+$ and continuous over the compact set X. Let the solution set X^* of the VIP be weakly sharp. Let $\{x^k\}$ be a subsequence with elements in X such that the real sequence $\{\mathrm{dist}(x^k, X^*)\}$ converges to zero. If F is uniformly continuous on an open set containing $\{x^k\}$ and X^*, then there exists a positive integer k_0 such that, for any index $k \geq k_0$, x^k is a solution of the VIP if and only if*

$$\lim_{k\to\infty} \mathrm{proj}_{T_X(x^k)}(-F(x^k)) = 0. \tag{5.4}$$

Proof. If $x^k \in X^*$, then $-F(x^k) \in N_X(x^k)$ and (5.4) holds trivially.

Otherwise, assume that (5.4) is satisfied. The Moreau decomposition of $-F(x^k)$ along $T_X(x)$ and its polar cone $N_X(x)$ yields

$$-F(x^k) = \mathrm{proj}_{T_X(x)}(-F(x^k)) + \mathrm{proj}_{N_X(x)}(-F(x^k)).$$

By Theorem 3.1, we have that F is constant over X^*. Thus, for any $x^* \in X^*$, the assumptions imply

$$\|F(x^*) + \mathrm{proj}_{N_X(x)}(-F(x^k))\| \to 0,$$

and so

$$\mathrm{dist}(x^k + \mathrm{proj}_{N_X(x^k)}(-F(x^k)), X^* - F(x^*)) \to 0.$$

But, from the weak sharpness property, one has

$$X^* - F(x^*) \subset \mathrm{int}\left(X^* + \bigcap_{x\in X^*} [T_X(x) \cap N_{X^*}(x)]^\circ\right).$$

Now, for x^k close to x^* in X^*, we have, using (5.3),

$$\begin{aligned} x^k + \mathrm{proj}_{N_X(x^k)}(-F(x^k)) &\in \mathrm{int}\left(X^* + \bigcap_{x\in X^*} [T_X(x) \cap N_{X^*}(x)]^\circ\right) \\ &\subset \bigcup_{x\in X^*} [x + N_X(x)]. \end{aligned}$$

Therefore, $\forall k$ sufficiently large,

$$\begin{aligned} x^k &= \mathrm{proj}_X(x^k + \mathrm{proj}_{N_X(x^k)}(-F(x^k))) \\ &\in \mathrm{proj}_X\left(\bigcup_{x\in X^*} [x + N_X(x)]\right) \\ &\subset \bigcup_{x\in X^*} \{x\} \\ &= X^* \quad . \end{aligned}$$

This completes the proof. □

Several authors have proposed general iterative frameworks for solving variational inequalities. For instance, Cohen [3], or Zhu and Marcotte [18] investigated a scheme in which x^{k+1} is a solution of the variational inequality

$$\langle \sigma F(x^k) + H(x^{k+1}) - H(x^k), x - x^{k+1}\rangle \geq 0 \quad \forall x \in X, \tag{5.5}$$

where σ is a positive constant and H is an auxiliary mapping, usually taken to be strongly monotone. Under suitable assumptions on F (strong monotonicity or co-coercivity[1]) and σ, the sequence $\{x^k\}$ is known to converge to a solution of the original variational inequality. From now on, we restrict our attention to those cases where the algorithm returns a convergent sequence $\{x^k\}$ whose limiting point is a solution of the VIP, and provide a sufficient condition for its finite termination.

Lemma 5.1. *Let F and H be uniformly continuous on X and $\{x^k\} \to x^*$. Then the sequence $\{\mathrm{proj}_{T_X(x^{k+1})}(-F(x^{k+1}))\}$ converges to zero.*

Proof. Since the sequence $\{x^k\}$ is convergent, $\|x^{k+1} - x^k\| \to 0$. From (5.5), we have

$$-[\sigma F(x^k) + H(x^{k+1}) - H(x^k)] \in N_X(x^{k+1}).$$

The Moreau decomposition technique yields

$$\begin{aligned}\|\mathrm{proj}_{-T_X(x^{k+1})}(F(x^{k+1}))\| &= \min_{v \in N_X(x^{k+1})} \| - (x^{k+1}) - v\| \\ &= \min_{z \in -F(x^{k+1}) - N_X(x^{k+1})} \|z\| \\ &\leq \left\| [F(x^{k+1}) - F(x^k)] + \frac{1}{\sigma}[H(x^{k+1}) - H(x^k)] \right\|.\end{aligned}$$

From the uniform continuity of F and H, the right-hand side of the above inequality converges to zero, and we obtain that $\{\mathrm{proj}_{T_X(x^{k+1})}(-F(x^{k+1})\}$ converges to zero, as claimed. □

Combining Theorem 5.2 and Lemma 5.1, we obtain the following result.

Theorem 5.3. *Under the assumptions of Theorem* 5.2 *and Lemma* 5.1*, the general iterative algorithm for solving the VIP based on the auxiliary problem* (5.5) *generates a sequence $\{x^k\}$ such that, for all k sufficiently large, x^k is a solution of the VIP.*

Recently, Zhu and Marcotte [18] have proposed a descent framework for the VIP, based on the auxiliary variational inequality (5.5), that includes as particular cases Fukushima's projective method [7] and Taji, Fukushima, and Ibaraki's Newton method [16] (see also Fukushima [8] or Larsson and Patriksson [9] for a survey of descent methods for the VIP). Given a mapping $H(w, x)$ defined on $X \times X$, continuous and strongly monotone with respect to the variable x and such that $H(x, x)$ coincides with the values $F(x)$ of the original mapping, a direction d^k is specified, at iteration k, as $d^k = w^k - x^k$, where w^k is the unique solution of the auxiliary variational inequality

$$\langle H(w^k, x^k) - H(x^k, x^k), x - x^k\rangle \geq 0 \quad \forall x \in X. \tag{5.6}$$

[1] The mapping F is co-coercive on the set X if there exists a positive number β such that $\langle F(x) - F(y), x - y\rangle \geq \beta\|F(x) - F(y)\|^2$ for all x, y in X; i.e., its inverse mapping is strongly monotone.

The iterate x^{k+1} is then obtained by minimizing some merit function (related to the auxiliary mapping H) along the direction d^k. Under a suitable assumption, we have that d^k is a descent direction for the merit function at the point x^k, and it can be shown that d^k converges to zero, while both w^k and x^k converge to a solution of the VIP. If F and H are both uniformly continuous, respectively, on X and $X \times X$, we then obtain that $\mathrm{proj}_{T_X(w^k)}(-F(w^k))$ converges to zero. If, furthermore, the assumptions of Theorem 5.2 are satisfied, the sequence $\{x^k\}$ converges to x^* after a finite number of iterations.

REFERENCES

[1] F. A. Al-Khayyal and J. Kyparisis, *Finite convergence of algorithms for nonlinear programming and variational inequalities*, J. Optim. Theory Appl., 70 (1991), pp. 319–332.

[2] A. Auslender, *Optimisation*, Méthodes numériques, Masson, Paris, 1976.

[3] G. Cohen, *Auxiliary problem principle extended to variational inequalities*, J. Optim. Theory Appl., 59 (1988), pp. 325–333.

[4] J. M. Danskin, *The theory of min-max with applications*, SIAM J. Appl. Math., 14 (1966), pp. 641–664.

[5] J. V. Burke and M. C. Ferris, *Weak sharp minima in mathematical programming*, SIAM J. Control Optim., 31 (1993), pp. 1340–1359.

[6] M. C. Ferris and O. L. Mangasarian, *Minimum principle sufficiency*, Math. Programming, 57 (1992), pp. 1–14.

[7] M. Fukushima, *Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems*, Math. Programming, 53 (1992), pp. 99–110.

[8] M. Fukushima, *Merit functions for variational inequality and complementarity problems*, Nonlinear Optimization and Applications, G. Di Pillo and F. Giannessi, eds., Plenum Press, 1996, pp. 155–179.

[9] T. Larsson and M. Patriksson, *A class of gap functions for variational inequalities*, Math. Programming, 64 (1994), pp. 53–80.

[10] O. L. Mangasarian and R. R. Meyer, *Nonlinear perturbation of linear programming*, SIAM J. Control Optim., 17 (1979), pp. 745–752.

[11] P. Marcotte and D. L. Zhu, *Monotone$^+$ mappings and variational inequalities*, in Fifth International Symposium on Generalized Convexity, Luminy, France, June 1996.

[12] J. J. Moreau, *Décomposition orthogonale dans un espace hilbertien selon deux cônes mutuellement polaires*, Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences (Paris), Série A, 255 (1962), pp. 233–240.

[13] S. Nguyen and C. Dupuis, *An efficient method for computing traffic equilibria in networks with asymmetric transportation costs*, Transportation Sci., 18 (1984), pp. 185–202.

[14] M. Patriksson, *A unified framework of descent algorithms for nonlinear programs and variational inequalities*, Ph.D. thesis, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden, 1993.

[15] S. Schaible, *Generalized monotonicity*, in Proceedings of the 10th International Summer School on Nonsmooth Optimization, Analysis and Applications, Erice, Italy, 1991, F. Giannessi, ed., Gordon and Breach, Amsterdam, The Netherlands, 1992.

[16] K. Taji, M. Fukushima, and T. Ibaraki, *A globally convergent Newton method for solving strongly monotone variational inequalities*, Math. Programming, 58 (1993), pp. 369–383.

[17] D. L. Zhu and P. Marcotte, *New classes of generalized monotononicity*, J. Optim. Theory. Appl., 87 (1995), pp. 457–471.

[18] D. L. Zhu and P. Marcotte, *An extended descent framework for variational inequalities*, J. Optim. Theory. Appl., 80 (1994), pp. 349–360.

[19] D. L. Zhu and P. Marcotte, *Co-coercivity and its role in the convergence of iterative schemes for solving variational inequalities*, SIAM J. Optim., 6 (1996), pp. 714–726.

SIAM J. OPTIM.
Vol. 9, No. 1, pp. 190–206

TOWARDS A PRACTICAL VOLUMETRIC CUTTING PLANE METHOD FOR CONVEX PROGRAMMING[*]

KURT M. ANSTREICHER[†]

Abstract. We consider the volumetric cutting plane method for finding a point in a convex set $\mathcal{C} \subset \Re^n$ that is characterized by a separation oracle. We prove polynomiality of the algorithm with each added cut placed directly through the current point and show that this "central cut" version of the method can be implemented using no more than $25n$ constraints at any time.

Key words. convex programming, cutting plane method, volumetric barrier

AMS subject classification. 90C25

PII. S1052623497318013

1. Introduction.

Let $\mathcal{C} \subset \Re^n$ be a convex set. Given a point $\bar{x} \in \Re^n$, a *separation oracle* for $\mathcal{C}$ either reports that $\bar{x} \in \mathcal{C}$ or returns a *separating hyperplane* $a \in \Re^n$ such that $a^T x > a^T \bar{x}$ for every $x \in \mathcal{C}$. The *convex feasibility problem* is to use such an oracle to find a point in $\mathcal{C}$ or prove that the volume of $\mathcal{C}$ must be less than that of an n-dimensional sphere of radius 2^{-L} for some given $L > 0$.

It is well known [9] that a variety of convex optimization problems can be cast as instances of the convex feasibility problem, and, moreover, the problem plays a fundamental role in the complexity analysis of many combinatorial optimization problems. Algorithms for the convex feasibility problem include the center of gravity method [12], the ellipsoid method [5], [9], the method of simplices [21], and the inscribed ellipsoid method [19]. In [20], Vaidya proposed an algorithm for the convex feasibility problem based on a new barrier for a polyhedral set, the *volumetric barrier*. On each iteration $k \geq 0$, Vaidya's algorithm has a point $x^k \in \Re^n$ and a polyhedral set $\mathcal{P}^k = \{x \mid A^k x \geq b^k\}$, where A^k is an $m_k \times n$ matrix. For each k the set $\mathcal{P}^k$ is bounded, $\mathcal{C} \subset \mathcal{P}^k$, and $x^k \in \mathcal{P}^k$ is an approximation of the *volumetric center* of $\mathcal{P}^k$, the minimizer of the volumetric barrier (see section 2). The algorithm then either deletes one constraint that defines $\mathcal{P}^k$ or calls the separation oracle to see if $x^k \in \mathcal{C}$. If not, the oracle returns a separating hyperplane which is used to add a constraint to $\mathcal{P}^k$. After the addition or deletion of a constraint, the algorithm takes a number of Newton, or Newton-like, steps for the volumetric barrier to obtain a new point x^{k+1} which is an approximation of the volumetric center of the new polyhedron $\mathcal{P}^{k+1}$.

Let T represent the cost, in numerical operations, of a call to the separation oracle. Vaidya's fundamental result is that the complexity of his volumetric cutting plane algorithm for the convex feasibility problem is $O(nLT + n^4 L)$ operations, compared to $O(n^2 LT + n^4 L)$ operations for the ellipsoid algorithm. (In theory, the complexity of Vaidya's method can be further reduced through the use of "fast matrix multiplication," which cannot be applied to the ellipsoid algorithm.) Although Vaidya's result is theoretically significant, the algorithm of [20] does not appear to be very practical. In

[*] Received by the editors March 17, 1997; accepted for publication (in revised form) January 30, 1998; published electronically December 2, 1998. This research was conducted while the author was visiting the Center for Operations Research and Econometrics, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, supported by a CORE fellowship.

http://www.siam.org/journals/siopt/9-1/31801.html

[†] Department of Management Sciences, University of Iowa, Iowa City, IA 52242 (kurt-anstreicher@uiowa.edu).

particular, the analysis of [20] requires that the polyhedral sets $\mathcal{P}^k$ have up to $10^7 n$ constraints, and the algorithm might require thousands of Newton-like steps following the addition or deletion of a constraint.

A strengthened version of Vaidya's volumetric cutting plane algorithm for the convex feasibility problem is described in [3]. The algorithm of [3] reduces the maximum number of constraints to $200n$, while requiring no more than 5 Newton steps following a constraint addition or deletion. Although these figures represent a substantial improvement over [20], the algorithm of [3] is still not fully practical, particularly in light of the following.

(i) For reasonable n, $200n$ constraints is still quite large, given that least-squares systems with this number of rows must be repeatedly solved on each iteration.

(ii) The algorithm of [3] uses true Newton steps, which in practice are expensive to compute compared to the Newton-like steps used in [20].

(iii) As in [20], the algorithm of [3] cannot place a new constraint directly through the current point, but must instead "back off" each separating hyperplane to generate a shallow cut.

Ramaswamy and Mitchell [18] describe a "central cut" version of the volumetric cutting plane algorithm that allows for the placement of each new constraint through the current point, and uses Newton-like steps following constraint additions and deletions. (The algorithm of [18] actually solves the problem of minimizing a linear function over a convex set $\mathcal{C}$ using a separation oracle, but most of the analysis is very similar to that required to solve the convex feasibility problem.) Unfortunately [18], which uses many results from [20], requires that the algorithm maintain up to $10^8 n$ constraints.

The purpose of this paper is to develop a central cut volumetric cutting plane algorithm that also improves on the $200n$ constraints required by the algorithm of [3]. As in [18], the algorithm uses an "affine" step to move off of a cut placed through the current point. The use of such a step in the context of a cutting plane method based on *analytic centers* is well known [7]. (See [4], [8], [11], [13], [15], [17], and [22] for other results on analytic center cutting plane methods.) In fact the affine step we use is based on that of [7] (see also [14]), rather than the step used in [18]. Our analysis uses a number of results from [2] and [3] and an improved second-order expansion of the volumetric barrier to improve upon the analysis of [18]. As in [18], the method described here requires $O(\sqrt{n})$ Newton-like steps following the addition or deletion of a constraint compared to $O(1)$ Newton, or Newton-like, steps in [20] and [3]. Although this is certainly a disadvantage from the standpoint of theoretical complexity, the fact that the $O(\sqrt{n})$ bound arises from a worst-case analysis of descent in the volumetric barrier suggests that, in practice, far fewer steps would likely be required. Our final result is a central cut volumetric cutting plane method that requires no more than $25n$ constraints at any time.

In Table 1.1 we summarize important attributes of four papers (including this paper) on volumetric cutting plane methods. These features are the placement of added cuts (shallow or central), the number of Newton or Newton-like steps required after a constraint addition or deletion, the maximum number of constraints required, and the value of a scalar ΔV, defined as the difference between the minimal increase in the volumetric barrier following a constraint addition, and the maximal decrease following a constraint deletion (see section 3). For all four algorithms the number of

TABLE 1.1
Volumetric cutting plane algorithms.

Reference	Placement of cut	Steps after addition/deletion	Number of constraints	ΔV
Vaidya [20]	Shallow	$O(1)$	$10^7 n$	1.3×10^{-7}
Anstreicher [3]	Shallow	$O(1)$	$200n$	3.7×10^{-4}
Ramaswamy and Mitchell [18]	Central	$O(\sqrt{n})$	$10^8 n$	6.8×10^{-7}
This paper	Central	$O(\sqrt{n})$	$25n$	1.4×10^{-3}

oracle calls is $O(nL)$, with a constant that is inversely proportional to ΔV (see, for example, the proof of [3, Theorem 3.2]).

Notation. We use notation consistent with [1], [2], [3], except that here we use H to denote the Hessian of the logarithmic barrier (as in [20] and [18]) rather than the Hessian of the volumetric barrier. We use $\succeq$ to denote the semidefinite ordering: if A and B are symmetric matrices, then $A \succeq B \iff A - B$ is positive semidefinite. If B is semidefinite, then $B^{1/2}$ denotes the symmetric, semidefinite "square root" of B. For a positive definite matrix A, we use $\|x\|_A = \sqrt{x^T A x}$. We use e to denote a vector of arbitrary dimension with each component equal to one. For a vector $x \in \Re^n$, $\mathrm{diag}(x)$ is the diagonal matrix whose diagonal components are those of x.

2. The volumetric barrier.

In this section we collect a number of properties of the volumetric barrier $V(\cdot)$ which will be used in the subsequent analysis. To start, let $\mathcal{P} = \{x \in \Re^n \mid Ax \geq b\}$, where A is an $m \times n$ matrix with independent columns, and $b \in \Re^m$. Whenever we refer to $\mathcal{P}$, we are implicitly referring to the constraint system $[A, b]$ which defines it. The volumetric barrier for $\mathcal{P}$ is the function

$$V(x) = \frac{1}{2}\,\mathrm{ldet}(A^T S(x)^{-2} A),$$

where $\mathrm{ldet}(\cdot) = \ln(\det(\cdot))$, $s(x) = Ax - b > 0$, and $S(x) = \mathrm{diag}(s(x))$. Let x be a point having $s = s(x) > 0$, and let $\sigma = \sigma(s)$ denote the vector equal to the diagonal of the projection matrix $P = P(s)$, where

$$P(s) = S^{-1} A (A^T S^{-2} A)^{-1} A^T S^{-1}.$$

In other words, $\sigma_i = p_{ii}$, $i = 1, \ldots, m$. It is then easy to show (see, for example, the appendix of [1]) that $0 \leq \sigma \leq e$, $e^T \sigma = n$. The gradient and Hessian of $V(\cdot)$ at x are given by

$$\begin{aligned} g = g(x) = \quad \nabla V(x)^T &= -A^T S^{-1} \sigma, \\ \nabla^2 V(x) &= A^T S^{-1}(3\Sigma - 2P^{(2)}) S^{-1} A, \end{aligned} \tag{2.1}$$

where $\Sigma = \mathrm{diag}(\sigma)$, and $P^{(2)}$ denotes the Schur or Hadamard product of P with itself: $p_{ij}^{(2)} = p_{ij}^2$. Let $Q = Q(x) = A^T S^{-2} \Sigma A$. Then $Q(x)$ is a good approximation of $\nabla^2 V(x)$ in that

$$Q(x) \preceq \nabla^2 V(x) \preceq 3Q(x). \tag{2.2}$$

See, for example, the appendix of [1] for a derivation of (2.1) and a proof of (2.2); these and other properties of $V(\cdot)$ are originally from Vaidya [20]. It is worthwhile to note that an appropriate multiple of $V(\cdot)$ provides an $O(\sqrt{mn})$-self-concordant barrier for

$\mathcal{P}$; see [16, Chapter 5.5] or [2, section 5]. However, most of the analysis required here does *not* follow from this self-concordancy result, the reason being that (as in [20] and [3]) we make extensive use of properties of $V(\cdot)$ that depend explicitly on σ.

In the following discussion we will often be interested in the behavior of $V(\cdot)$ for a step of the form $\bar{x} = x+\xi$, where $s = s(x) > 0$ and $\|S^{-1}A\xi\|_\infty = \delta < 1$. For such an $\bar{x}$ let $\bar{s} = s(\bar{x})$, $\bar{\sigma} = \sigma(\bar{s})$, $\bar{Q} = Q(\bar{x})$. The proof of the following is very straightforward; see, for example, [20, Lemma 5] or [1, Lemma 2.2].

PROPOSITION 2.1. *Let $\bar{x} = x + \xi$, where $s = s(x) > 0$ and $\|S^{-1}A\xi\|_\infty \le \delta < 1$. Then*

$$1-\delta \le \frac{\bar{s}_i}{s_i} \le 1+\delta \qquad \textit{and} \qquad \frac{(1-\delta)^2}{(1+\delta)^2} \le \frac{\bar{\sigma}_i}{\sigma_i} \le \frac{(1+\delta)^2}{(1-\delta)^2}, \qquad i = 1,\dots,m.$$

It follows immediately from Proposition 2.1 that if $\bar{x} = x + \xi$ and $\|S^{-1}A\xi\|_\infty = \delta < 1$, then

$$\frac{(1-\delta)^2}{(1+\delta)^4} Q \preceq \bar{Q} \preceq \frac{(1+\delta)^2}{(1-\delta)^4} Q\ . \tag{2.3}$$

Using a Taylor series expansion, (2.2), and (2.3), it is then easy to show that

$$V(x) + g^T\xi + \frac{(1-\delta)^2}{2(1+\delta)^4}\xi^T Q\xi \le V(\bar{x}) \le V(x) + g^T\xi + \frac{3(1+\delta)^2}{2(1-\delta)^4}\xi^T Q\xi. \tag{2.4}$$

The bounds in (2.4) have been used in [1], [18], and [20]. The following theorem provides a strengthening of (2.4) that will be used throughout the paper.

THEOREM 2.2. *Suppose that $\bar{x} = x + \xi$, where $s = s(x) > 0$ and $\|S^{-1}A\xi\|_\infty \le \delta < 1$. Then*

$$V(x) + g^T\xi + \frac{1}{2(1+\delta)^2}\xi^T Q\xi \le V(\bar{x}) \le V(x) + g^T\xi + \frac{3+\delta^2}{2(1-\delta)^2}\xi^T Q\xi.$$

Proof. We have

$$\begin{aligned} V(\bar{x}) &= V(x) + \int_0^1 \xi^T g(x+\alpha\xi)\,d\alpha \\ &= V(x) + \int_0^1 \xi^T \left(g(x) + \int_0^\alpha \nabla^2 V(x+\beta\xi)\xi\,d\beta\right) d\alpha \\ &= V(x) + g^T\xi + \int_0^1 \int_0^\alpha \xi^T \nabla^2 V(x+\beta\xi)\xi\,d\beta\,d\alpha. \end{aligned} \tag{2.5}$$

To prove the theorem we will obtain lower and upper bounds on the final term in (2.5). We begin with the lower bound. Using (2.2) and (2.3), we have

$$\xi^T\nabla^2 V(x+\beta\xi)\xi \ge \frac{(1-\beta\delta)^2}{(1+\beta\delta)^4}\xi^T Q\xi, \tag{2.6}$$

and therefore

$$V(\bar{x}) \ge V(x) + g^T\xi + \xi^T Q\xi \int_0^1 \int_0^\alpha \frac{(1-\beta\delta)^2}{(1+\beta\delta)^4}\,d\beta\,d\alpha. \tag{2.7}$$

However, it is straightforward to compute that

$$\frac{d}{d\beta}\left(\frac{1-\beta\delta}{1+\beta\delta}\right)^3 = -6\,\delta\frac{(1-\beta\delta)^2}{(1+\beta\delta)^4},$$

and therefore

$$\int_0^\alpha \frac{(1-\beta\delta)^2}{(1+\beta\delta)^4}\,d\beta = \frac{-1}{6\delta}\left(\left(\frac{1-\alpha\delta}{1+\alpha\delta}\right)^3 - 1\right) = \frac{6\alpha\delta+2\alpha^3\delta^3}{6\delta(1+\alpha\delta)^3} \ge \frac{\alpha}{(1+\alpha\delta)^3}\ . \tag{2.8}$$

Substituting (2.8) into (2.7), we obtain

$$V(\bar{x}) \ge V(x) + g^T\xi + \xi^T Q\xi \int_0^1 \frac{\alpha}{(1+\alpha\delta)^3}\,d\alpha. \tag{2.9}$$

An integration by parts shows that

$$\int_0^1 \frac{\alpha}{(1+\alpha\delta)^3}\,d\alpha = \frac{1}{2(1+\delta)^2}, \tag{2.10}$$

and substituting (2.10) into (2.9) produces the lower bound of the theorem. The proof of the upper bound is similar. Again using (2.2) and (2.3), we have

$$\xi^T\nabla^2 V(x+\beta\xi)\xi \le \frac{3(1+\beta\delta)^2}{(1-\beta\delta)^4}\xi^T Q\xi,$$

and therefore

$$V(\bar{x}) \le V(x) + g^T\xi + 3\xi^T Q\xi \int_0^1\int_0^\alpha \frac{(1+\beta\delta)^2}{(1-\beta\delta)^4}\,d\beta\,d\alpha. \tag{2.11}$$

However,

$$\frac{d}{d\beta}\left(\frac{1+\beta\delta}{1-\beta\delta}\right)^3 = 6\delta\,\frac{(1+\beta\delta)^2}{(1-\beta\delta)^4},$$

and therefore

$$\int_0^\alpha \frac{(1+\beta\delta)^2}{(1-\beta\delta)^4}\,d\beta = \frac{1}{6\delta}\left(\left(\frac{1+\alpha\delta}{1-\alpha\delta}\right)^3 - 1\right) = \frac{6\alpha\delta+2\alpha^3\delta^3}{6\delta(1-\alpha\delta)^3} \le \frac{\alpha(1+\delta^2/3)}{(1-\alpha\delta)^3}. \tag{2.12}$$

Substituting (2.12) into (2.11), we obtain

$$V(\bar{x}) \le V(x) + g^T\xi + \xi^T Q\xi(3+\delta^2)\int_0^1 \frac{\alpha}{(1-\alpha\delta)^3}\,d\alpha. \tag{2.13}$$

Another integration by parts shows that

$$\int_0^1 \frac{\alpha}{(1-\alpha\delta)^3}\,d\alpha = \frac{1}{2(1-\delta)^2}, \tag{2.14}$$

and substituting (2.14) into (2.13) produces the upper bound of the theorem. □

Since the bounds in Theorem 2.2 involve both $\|\xi\|_Q = \sqrt{\xi^T Q \xi}$ and $\|S^{-1}A\xi\|_\infty$, it is natural to consider how these two quantities are related. For x with $s = s(x) > 0$, $\sigma = \sigma(s)$, let $\sigma_{\min} = \min_i\{\sigma_i\}$. Note that $\sigma_{\min} > 0$ under the trivial assumption that A contains no zero row. Define

$$\mu = \mu(x) = \left(2\sqrt{\sigma_{\min}} - \sigma_{\min}\right)^{-1/2}. \tag{2.15}$$

In the following theorem we give two bounds for $\|S^{-1}A\xi\|_\infty$ in terms of $\|\xi\|_Q$: one involving μ, and therefore $\sigma_{\min}$, and the other independent of σ.

Theorem 2.3. *Let x have $s = s(x) > 0$. Then for any $\xi \in \Re^n$,*

1. $\|S^{-1}A\xi\|_\infty \le \mu\|\xi\|_Q$,
2. $\|S^{-1}A\xi\|_\infty \le [(1+\sqrt{m})/2]^{1/2}\|\xi\|_Q$.

Proof. See [1, Theorem 3.3] for the proof of 2, and [3, Lemma 2.3] for the proof of 1. □

Motivated by Theorem 2.3, we define

$$\hat{\mu} = \hat{\mu}(x) = \min\{\mu(x), [(1+\sqrt{m})/2]^{1/2}\}. \tag{2.16}$$

It then follows from Theorem 2.3 that

$$\|S^{-1}A\xi\|_\infty \le \hat{\mu}\|\xi\|_Q, \qquad \xi \in \Re^n. \tag{2.17}$$

The fundamental proximity criterion that we employ throughout the paper is $\hat{\mu}\|g\|_{Q^{-1}}$. When this quantity is "large" (that is, $\Omega(1)$), we will take a damped Newton-like step in an effort to reduce $V(\cdot)$, and thus move closer to ω, the volumetric center of $\mathcal{P}$. When $\hat{\mu}\|g\|_{Q^{-1}} \le O(1)$, we will be close enough to ω to adequately control the effect of adding or deleting a constraint, as required. The following theorem and corollary obtain a simple condition on $\hat{\mu}\|g\|_{Q^{-1}}$ that suffices to demonstrate the boundedness of $\mathcal{P}$.

Theorem 2.4. *Let $\mathcal{P} = \{x \mid Ax \ge b\}$, where the columns of A are independent. Let x have $s = s(x) > 0$, and let $d = Q^{-1}g$. Suppose that $\|S^{-1}Ad\|_\infty < 1$. Then $\mathcal{P}$ is bounded.*

Proof. $\mathcal{P}$ is bounded if and only if $\nexists x \ne 0$, $Ax \ge 0$. Since the columns of A are independent,

$$\begin{aligned} \nexists x \ne 0,\ Ax \ge 0 &\iff \nexists x,\ Ax \ne 0,\ Ax \ge 0 \\ &\iff \nexists x, Ax \ge 0, e^T Ax \ge 1 \\ &\iff \exists u \ge 0, v > 0,\ u^T A + v e^T A = 0 \\ &\iff \exists u > 0,\ A^T u = 0, \end{aligned} \tag{2.18}$$

where the third equivalence uses a standard "theorem of the alternative" for systems of linear inequalities. However, $Q = A^T S^{-2}\Sigma A$, so $Qd = g$ is exactly $A^T S^{-2}\Sigma A d = -A^T S^{-1}\sigma$, which can be written as

$$A^T S^{-1}\Sigma(e + S^{-1}Ad) = 0.$$

It follows that if $\|S^{-1}Ad\|_\infty < 1$, then $u = S^{-1}\Sigma(e + S^{-1}Ad)$ satisfies (2.18), and therefore $\mathcal{P}$ is bounded. □

Corollary 2.5. *Let $\mathcal{P} = \{x \mid Ax \ge b\}$, where the columns of A are independent. Let x have $s = s(x) > 0$, and suppose that $\hat{\mu}\|g\|_{Q^{-1}} < 1$. Then $\mathcal{P}$ is bounded.*

Proof. This follows from Theorem 2.4, (2.17), and $\|d\|_Q = \|g\|_{Q^{-1}}$. □

Next we show that if $\hat{\mu}\|g\|_{Q^{-1}}$ is sufficiently small, then we can bound the possible remaining decrease in $V(\cdot)$. The proximity allowed in Theorem 2.6, $\hat{\mu}\|g\|_{Q^{-1}} \le 1/6$, is weaker than in previous, similar results in [20], [2], and [3].

THEOREM 2.6. *Let x have $s = s(x) > 0$, and $\hat{\mu}\|g\|_{Q^{-1}} \le \gamma \le 1/6$. Then $\hat{\mu}\|\omega - x\|_Q \le 1$, where ω is the minimizer of $V(\cdot)$, and*

$$V(\omega) - V(x) \ge \min_{0\le\alpha\le1} \frac{1}{\hat{\mu}^2}\left(-\gamma\alpha + \frac{\alpha^2}{2(1+\alpha)^2}\right).$$

Proof. Assume that $\hat{\mu}\|\omega - x\|_Q > 1$. Then there is a λ, $0 < \lambda < 1$, so that $\bar{x} = x + \lambda(\omega - x)$ has $\hat{\mu}\|\bar{x} - x\|_Q = 1$. Let $\xi = \bar{x} - x$, and $x(\alpha) = x + \alpha\xi$, $0 \le \alpha \le 1$. Since $\|S^{-1}A\xi\|_\infty \le 1$, from (2.17), we have

$$\begin{aligned} \frac{d}{d\alpha}V(x(\alpha)) &= g(x+\alpha\xi)^T\xi \\ &= \xi^T\left(g(x) + \int_0^\alpha \nabla^2 V(x+\beta\xi)\xi\, d\beta\right) \\ &\ge g^T\xi + \xi^T Q\xi \int_0^\alpha \frac{(1-\beta)^2}{(1+\beta)^4}\, d\beta \\ &= g^T\xi + \xi^T Q\xi \frac{6\alpha + 2\alpha^3}{6(1+\alpha)^3}, \end{aligned} \tag{2.19}$$

where the inequality uses (2.6), and the final equality uses (2.8), both with $\delta = 1$. Using the fact that $|g^T\xi| = |g^T Q^{-1/2}Q^{1/2}\xi| \le \|g\|_{Q^{-1}}\|\xi\|_Q$, we then obtain

$$\begin{aligned} \left.\frac{d}{d\alpha}V(x(\alpha))\right|_{\alpha=1^-} &\ge -\|g\|_{Q^{-1}}\|\xi\|_Q + \frac{\|\xi\|_Q^2}{6} \\ &= \frac{1}{\hat{\mu}^2}\left(-(\hat{\mu}\|g\|_{Q^{-1}})(\hat{\mu}\|\xi\|_Q) + \frac{\hat{\mu}^2\|\xi\|_Q^2}{6}\right) \\ &\ge 0, \end{aligned}$$

where the last inequality uses $\hat{\mu}\|\xi\|_Q = 1$, and the assumption that $\hat{\mu}\|g\|_{Q^{-1}} \le 1/6$. Since $V(\cdot)$ is strictly convex, it follows that $V(\omega) > V(\bar{x})$, which is a contradiction. Therefore $\hat{\mu}\|\omega - x\|_Q \le 1$, as claimed.

Since $\hat{\mu}\|x - \omega\|_Q \le 1$, we can write $\omega = x(\alpha) = x + \alpha\xi$ for some ξ with $\hat{\mu}\|\xi\|_Q = 1$, and $0 \le \alpha \le 1$. Then $\|S^{-1}A\xi\|_\infty \le 1$, from (2.17), so Theorem 2.2 implies that

$$\begin{aligned} V(x(\alpha)) &\ge V(x) - \alpha g^T\xi + \alpha^2\frac{\xi^T Q\xi}{2(1+\alpha)^2} \\ &\ge V(x) - \alpha\|g\|_{Q^{-1}}\|\xi\|_Q + \alpha^2\frac{\|\xi\|_Q^2}{2(1+\alpha)^2} \\ &= V(x) + \frac{1}{\hat{\mu}^2}\left(-\alpha(\hat{\mu}\|g\|_{Q^{-1}})(\hat{\mu}\|\xi\|_Q) + \alpha^2\frac{\hat{\mu}^2\|\xi\|_Q^2}{2(1+\alpha)^2}\right) \\ &\ge V(x) + \frac{1}{\hat{\mu}^2}\left(-\alpha\gamma + \frac{\alpha^2}{2(1+\alpha)^2}\right), \end{aligned} \tag{2.20}$$

where the second inequality uses $|g^T\xi| \le \|g\|_{Q^{-1}}\|\xi\|_Q$, and the final inequality uses $\hat{\mu}\|\xi\|_Q = 1$ and the assumption that $\hat{\mu}\|g\|_{Q^{-1}} \le \gamma$. □

In the corollary below we use Theorem 2.6 to establish bounds on $V(x) - V(\omega)$ for two values of the parameter γ which are useful in the the analysis to follow.

COROLLARY 2.7. *Let x have $s = s(x) > 0$. Then $\hat{\mu}\|g\|_{Q^{-1}} \le 4/27$ implies that $V(x) - V(\omega) \le .0232/\hat{\mu}^2$, and $\hat{\mu}\|g\|_{Q^{-1}} \le 1/8$ implies that $V(x) - V(\omega) \le .0113/\hat{\mu}^2$.*

Proof. For $\gamma = 1/8$, it is straightforward to show that the minimum in (2.20), for $0 \le \alpha \le 1$, occurs at $\alpha = \sqrt{5} - 2$. Substituting this value of α into (2.20) and simplifying then implies that

$$V(\omega) - V(x) \ge -\frac{5\sqrt{5} - 11}{16\hat{\mu}^2} > -\frac{.0113}{\hat{\mu}^2} \;.$$

It is also easy to show that, for $\gamma \ge 4/27$, the right-hand side in (2.20) is monotonically decreasing for $0 \le \alpha \le 1$. Substituting $\gamma = 4/27$, $\alpha = 1$ into (2.20) then implies that

$$V(\omega) - V(x) \ge (-4/27 + 1/8)/\hat{\mu}^2 > -.0232/\hat{\mu}^2. \qquad \square$$

The final topic that we consider in this section is that of reducing $V(\cdot)$ when $\hat{\mu}\|g\|_{Q^{-1}} \ge \Omega(1)$. To accomplish this we use a Newton-like step of the form

$$x(\alpha) = x - \alpha \frac{Q^{-1}g}{\hat{\mu}\|g\|_{Q^{-1}}} \tag{2.21}$$

for some $\alpha > 0$. When $\hat{\mu}\|g\|_{Q^{-1}} \ge \gamma$, for any $\gamma = \Omega(1)$, it can be shown that α may be chosen in (2.21) so that an $\Omega(1/\sqrt{m})$ reduction is obtained in $V(\cdot)$. In the following lemma we give a result for a particular value of γ used later in the paper.

LEMMA 2.8. *Suppose that x has $s = s(x) > 0$, and $\hat{\mu}\|g\|_{Q^{-1}} \ge .01$. Let $x(\alpha)$ be as in* (2.21). *Then $\alpha = .0033$ obtains $V(x(\alpha)) \le V(x) - (1.65 \times 10^{-5})/\sqrt{m}$.*

Proof. Let $\xi = -Q^{-1}g/(\hat{\mu}\|g\|_{Q^{-1}})$. By construction we have $\hat{\mu}\|\xi\|_Q = 1$, so $\|S^{-1}A\xi\|_\infty \le 1$ by (2.17). Applying Theorem 2.2, we obtain

$$\begin{aligned} V(x(\alpha)) &\le V(x) + \alpha g^T\xi + \alpha^2 \xi^T Q \xi \frac{3 + \alpha^2}{2(1-\alpha)^2} \\ &= V(x) - \frac{\alpha}{\hat{\mu}}\|g\|_{Q^{-1}} + \frac{\alpha^2}{\hat{\mu}^2}\frac{3 + \alpha^2}{2(1-\alpha)^2} \\ &= V(x) + \frac{1}{\hat{\mu}^2}\left(-\alpha\hat{\mu}\|g\|_{Q^{-1}} + \alpha^2 \frac{3+\alpha^2}{2(1-\alpha)^2}\right). \end{aligned} \tag{2.22}$$

Substituting $\alpha = .0033$ into (2.22), and using $\hat{\mu}\|g\|_{Q^{-1}} \ge .01$, we obtain $V(x(\alpha)) \le V(x) - (1.65 \times 10^{-5})/\hat{\mu}^2$. Finally, from (2.16), $\hat{\mu}^2 \le (1 + \sqrt{m})/2 \le \sqrt{m}$. $\square$

3. The algorithm and its complexity. In this section we describe the central cut volumetric cutting plane method, and establish its complexity using results from the two following sections. At the start of each iteration $k \ge 0$, we have an interior point x^k of a bounded polyhedron $\mathcal{P}^k \supset \mathcal{C}$, where $\mathcal{P}^k = \{x \,|\, A^k x \ge b^k\}$, and A^k is an $m_k \times n$ matrix with independent columns. We assume that $\mathcal{C}$ is contained in the hypercube $\|x\|_\infty \le 1$, and set $\mathcal{P}^0 = \{x \,|\, -e \le x \le e\}$, $x^0 = 0$. (It is straightforward to show that x^0 is the volumetric center of $\mathcal{P}^0$.) The algorithm to be analyzed is as follows.

CENTRAL CUT VOLUMETRIC CUTTING PLANE ALGORITHM.

```
Step 0. Given $x^0$, $\mathcal{P}^0$, $0 < \epsilon < 1$, $0 < \gamma < 1$, $L \ge 1$. Go to Step 1.
Step 1. If $V^k(x^k) \ge V^k_{\max}$, then STOP. Else go to Step 2.
```

Step 2. If $\sigma^k_{\min} \ge \epsilon$, go to Step 3. Else go to Step 4.

Step 3. (*Constraint Addition*) Call the oracle to see if $x^k \in \mathcal{C}$. If so, STOP. Otherwise the oracle returns a vector $a \in \Re^n$ such that $a^Tx > a^Tx^k$ for all $x \in \mathcal{C}$. Let (A^{k+1}, b^{k+1}) be an augmented constraint system having $a^{k+1}_{m_k+1} = a$, $b^{k+1}_{m_k+1} = a^Tx^k$. Let $s^k = A^kx^k - b^k$, $S^k = \operatorname{diag}(s^k)$, and $\bar{x}^0 = x^k + \alpha({A^k}^T(S^k)^{-2}A^k)^{-1}a$, for suitable $\alpha > 0$. Go to Step 5.

Step 4. (*Constraint Deletion*) Suppose that $\sigma^k_j = \sigma^k_{\min} < \epsilon$. Let (A^{k+1}, b^{k+1}) be the reduced constraint system obtained by removing the jth row of (A^k, b^k), and let $\bar{x}^0 = x^k$. Go to Step 5.

Step 5. (*Centering Steps*) Take a sequence of damped Newton-like steps of the form $\bar{x}^{j+1} = \bar{x}^j - \alpha(\bar{Q}^j)^{-1}\bar{g}^j$, $j \ge 0$, until $\hat{\bar{\mu}}^J\|\bar{g}^J\|_{(\bar{Q}^J)^{-1}} \le \gamma$, where $\bar{Q}^j = Q(\bar{x}^j)$, $\bar{g}^j = g(\bar{x}^j)$, $\hat{\bar{\mu}}^J = \hat{\mu}(\bar{x}^J)$. Let $x^{k+1} = \bar{x}^J$, $k = k+1$, and go to Step 1.

In Step 1 of the algorithm, the value of $V^k_{\max}$ is such that $V(x^k) \ge V^k_{\max}$ proves that the volume of $\mathcal{P}^k$, and therefore also $\mathcal{C} \subset \mathcal{P}^k$, is less than that of an n-dimensional sphere of radius 2^{-L}. An explicit value for $V^k_{\max}$ is given in Lemma 3.1 below. A suitable steplength α in Step 3 is given in Theorem 4.6. Note that, by construction, each $\mathcal{P}^k$ is bounded by Corollary 2.5, since $\hat{\mu}^k\|g^k\|_{(Q^k)^{-1}} \le \gamma < 1$ for each k. In addition, the fact that a constraint is only added if $\sigma^k_{\min} \ge \epsilon$, and $e^T\sigma^k = n$ for all k, implies that $m_k \le n/\epsilon + 1$ for every k. For the Newton-like steps in Step 5, we assume that the steplengths α are chosen so that each step produces an $\Omega(1/\sqrt{n})$ decrease in $V^k(\cdot)$. That this is always possible follows from $\hat{\bar{\mu}}^j\|\bar{g}^j\|_{(\bar{Q}_j)^{-1}} > \gamma = \Omega(1)$ (see, for example, Lemma 2.8) and the fact that $m_k \le n/\epsilon + 1 = O(n)$. Finally, if A^k has independent columns and A^{k+1} is obtained by adding a constraint, then trivially the columns of A^{k+1} are independent. If A^{k+1} is obtained by deleting a constraint, independence of the columns of A^{k+1} follows easily from the dropping rule $\sigma^k_{\min} < \epsilon \le 1$; see [3, section 5].

LEMMA 3.1. *Consider the volumetric cutting plane algorithm with $\gamma \le .03$. Assume that $L \ge 1$, and let $V^k_{\max} = .7nL + n\ln(m_k)$. Then termination in Step* 1 *proves that the volume of $\mathcal{C}$ is less than that of an n-dimensional sphere of radius 2^{-L}.*

Proof. See [3, Lemma 3.1]. □

Next we consider the issue of how many iterations might be required for the algorithm to terminate. Assume that each time a constraint is added, the algorithm achieves

$$V^{k+1}(x^{k+1}) \ge V^k(x^k) + \Delta V^+, \tag{3.1}$$

where $\Delta V^+ > 0$, while each time a constraint is deleted, it is ensured that

$$V^{k+1}(x^{k+1}) \ge V^k(x^k) - \Delta V^-, \tag{3.2}$$

where $0 \le \Delta V^- < \Delta V^+$. The following theorem provides a complexity result for the algorithm under simple assumptions regarding $\Delta V \equiv \Delta V^+ - \Delta V^-$ and the number of Newton-like steps taken in Step 5.

THEOREM 3.2. *Assume that the iterates of the volumetric cutting plane algorithm, using $\gamma \le .03$, satisfy* (3.1) *and* (3.2) *on iterations where a constraint is added or deleted, respectively. Assume further that ΔV^+ is $O(1)$, $\Delta V = \Delta V^+ - \Delta V^- > 0$ is $\Omega(1)$, and the number of Newton-like steps in Step* 5 *of the algorithm is $O(\sqrt{n})$. Then, for $L = \Omega(\ln(n))$, the algorithm terminates in $O(nL)$ iterations, using a total of $O(nLT + n^{4.5}L)$ operations, where T is the cost of a call to the separation oracle.*

Proof. See the proof of [3, Theorem 3.2]. □

Compared to the algorithms of [20] and [3], Theorem 3.2 demonstrates that the central cut method of this paper has the same order for the number of oracle calls, $O(nL)$, but performs more non-oracle work, $O(n^{4.5}L)$ versus $O(n^4L)$ operations. The reason for the latter is the larger number of centering steps, $O(\sqrt{n})$ versus $O(1)$, required after a constraint addition or deletion. Using results from the next two sections, we now show that the assumptions of Theorem 3.2 hold for certain choices of the parameters ϵ and γ.

THEOREM 3.3. *Let $\epsilon = .04$, $\gamma = .01$. Then the central cut volumetric cutting plane method satisfies the assumptions of Theorem* 3.2, *with $\Delta V = .0014$.*

Proof. First consider an iteration where a cut is added. In Theorem 4.6, it is shown that for a particular choice of α in Step 3, it is ensured that

$$V^{k+1}(\bar{x}^0) \le V^k(x^k) + .3\|g^k\|_{(Q^k)^{-1}} + 1.78 < V^k(x^k) + 1.79, \tag{3.3}$$

where we are using the fact that $\|g^k\|_{(Q^k)^{-1}} \le \hat{\mu}^k\|g^k\|_{(Q^k)^{-1}} \le .01$. In addition, in Theorem 4.5, it is shown that for $\|g^k\|_{(Q^k)^{-1}} \le .01$ and $\sigma^k_{\min} \ge .04$,

$$V^{k+1}(\omega^{k+1}) \ge V^k(x^k) + .0340, \tag{3.4}$$

where ω^{k+1} is the volumetric center of $\mathcal{P}^{k+1}$. Combining (3.3) and (3.4), we obtain

$$V^{k+1}(\bar{x}^0) - V^{k+1}(\omega^{k+1}) < 1.76. \tag{3.5}$$

Next, in Lemma 2.8, it is shown that if $\hat{\bar{\mu}}^j\|\bar{g}^j\|_{(\bar{Q}^j)^{-1}} \ge \gamma = .01$ in Step 5, then a steplength α may be chosen so that

$$V^{k+1}(\bar{x}^{j+1}) \le V^{k+1}(\bar{x}^j) - \Omega(1/\sqrt{m_{k+1}}). \tag{3.6}$$

However, $m_k \le n/\epsilon + 1 = O(n)$ for all k, so (3.5) and (3.6) together imply that after $J = O(\sqrt{n})$ steps, we must obtain $\bar{x}^J$ having $\hat{\bar{\mu}}^J\|\bar{g}^J\|_{(\bar{Q}^J)^{-1}} \le \gamma = .01$. Finally, $V^{k+1}(x^{k+1}) = V^{k+1}(\bar{x}^J) \ge V^{k+1}(\omega^{k+1})$, so (3.4) implies that

$$V^{k+1}(x^{k+1}) - V^k(x^k) \ge \Delta V^+ = .0340.$$

Next, consider an iteration where a constraint is deleted. In Theorem 5.2 it is shown that if $\hat{\mu}^k\|g^k\|_{(Q^k)^{-1}} \le .01$ and $\sigma^k_i = \sigma^k_{\min} \le .04$, then dropping constraint i to obtain a new polyhedron $\mathcal{P}^{k+1}$ results in $V^{k+1}(x^k) - V^{k+1}(\omega^{k+1}) \le .0121$. Arguing exactly as above, it follows that after $J = O(\sqrt{n})$ steps in Step 5, we must obtain $\bar{x}^J$ with $\hat{\bar{\mu}}^J\|\bar{g}^J\|_{(\bar{Q}^J)^{-1}} \le \gamma = .01$. In addition, in Theorem 5.2 it is shown that $V^{k+1}(\omega^{k+1}) \ge V^k(x^k) - .0326$. Since we must have $V^{k+1}(x^{k+1}) \ge V^{k+1}(\omega^{k+1})$, it follows that

$$V^{k+1}(x^{k+1}) - V^k(x^k) \ge -\Delta V^- = -.0326.$$

The assumptions of Theorem 3.2 thus hold with $\Delta V = .0340 - .0326 = .0014$. □

The value $\Delta V = .0014$ demonstrated in Theorem 3.3 may seem relatively small, but it should be noted that this is the largest value of ΔV to date for a volumetric cutting plane algorithm; see Table 1.1.

4. Adding a central cut. Let x be an interior point of $\mathcal{P}$. In this section we consider augmenting the constraint system defining $\mathcal{P}$ by imposing a central cut through x to obtain a new polyhedron $\tilde{\mathcal{P}} = \{\tilde{x} \mid A\tilde{x} \geq b,\ a^T\tilde{x} \geq a^T x\}$. Let $\tilde{V}(\cdot)$ be the volumetric barrier for $\tilde{\mathcal{P}}$, and $\tilde{\omega}$ be the volumetric center. Note that for any $\bar{x}$ with $\bar{s} = s(\bar{x}) > 0$, $a^T\bar{x} > a^T x$, we have

$$\begin{aligned}\tilde{V}(\bar{x}) &= \frac{1}{2}\,\mathrm{ldet}\left(A^T\bar{S}^{-2}A + \frac{aa^T}{(a^T\bar{x} - a^T x)^2}\right)\\ &= \frac{1}{2}\,\mathrm{ldet}\left(A^T\bar{S}^{-2}A\left(I + \frac{(A^T\bar{S}^{-2}A)^{-1}aa^T}{(a^T\bar{x} - a^T x)^2}\right)\right)\\ &= V(\bar{x}) + \frac{1}{2}\ln\left(1 + \frac{a^T(A^T\bar{S}^{-2}A)^{-1}a}{(a^T\bar{x} - a^T x)^2}\right). \qquad (4.1)\end{aligned}$$

We will first use (4.1) to establish a lower bound on $\tilde{V}(\tilde{\omega}) - V(x)$ when a cut is added through x. We will obtain two versions of this result. The first, using $x = \omega$, produces a relatively simple bound for the fundamental quantity $\tilde{V}(\tilde{\omega}) - V(\omega)$. Although this bound may be of some independent interest, in practice it cannot be used since $x = \omega$ is unattainable. Therefore we will also obtain a second lower bound for x in a certain neighborhood of ω. We begin with a series of lemmas. Throughout we let $s = s(x)$, $\sigma_{\min} = \sigma_{\min}(s)$, $Q = Q(x)$, $g = g(x)$, $\mu = \mu(x)$.

LEMMA 4.1. *Assume that* $\bar{s} = s(\bar{x}) > 0$, $a^T\bar{x} > a^T x$, *and* $\|\bar{S}^{-1}A(\bar{x} - x)\| \leq \rho$. *Then*

$$\frac{a^T(A^T\bar{S}^{-2}A)^{-1}a}{(a^T\bar{x} - a^T x)^2} \geq \frac{1}{\rho^2}.$$

Proof. Let $\bar{H} = A^T\bar{S}^{-2}A$, so $\|\bar{x} - x\|_{\bar{H}} = \|\bar{S}^{-1}A(\bar{x} - x)\| \leq \rho$, and $\|a\|^2_{\bar{H}^{-1}} = a^T(A^T\bar{S}^{-2}A)^{-1}a$. Then $|a^T(\bar{x} - x)| = |a^T\bar{H}^{-1/2}\bar{H}^{1/2}(\bar{x} - x)| \leq \|a\|_{\bar{H}^{-1}}\|\bar{x} - x\|_{\bar{H}}$, implying that

$$\frac{\|a\|^2_{\bar{H}^{-1}}}{[a^T(x - \bar{x})]^2} \geq \frac{1}{\rho^2}. \qquad \square$$

LEMMA 4.2. *Assume that* $\bar{s} = s(\bar{x}) > 0$, $a^T\bar{x} > a^T x$, $\|\bar{x} - x\|_Q \leq r$, *and* $\mu r \leq 1$. *Then*

$$\frac{a^T(A^T\bar{S}^{-2}A)^{-1}a}{(a^T\bar{x} - a^T x)^2} \geq \frac{(1 - \mu r)^2\sigma_{\min}}{r^2}\ .$$

Proof. Since $Q = A^T S^{-2}\Sigma A$, $\|\bar{x} - x\|_Q \leq r$ immediately implies that

$$\|S^{-1}A(\bar{x} - x)\| \leq \frac{r}{\sqrt{\sigma_{\min}}} \qquad (4.2)$$

and also, from Theorem 2.3, that

$$\|S^{-1}A(\bar{x} - x)\|_\infty \leq \mu r. \qquad (4.3)$$

From (4.3) and Proposition 2.1, it follows that

$$\frac{s_i}{\bar{s}_i} \leq \frac{1}{1 - \mu r}, \qquad i = 1, \ldots, m. \qquad (4.4)$$

Then (4.2) and (4.4) together imply that

$$\|\bar{S}^{-1}A(\bar{x}-x)\| \leq \frac{r}{(1-\mu r)\sqrt{\sigma_{\min}}} \ ,$$

and the lemma follows from Lemma 4.1. □

LEMMA 4.3. *Assume that* $\bar{s} = s(\bar{x}) > 0$, $\|\bar{x}-x\|_Q = r$, *and* $\mu r \leq 1$. *Then*

$$V(\bar{x}) \geq V(x) + g^T(\bar{x}-x) + \frac{r^2}{2(1+\mu r)^2} \ .$$

Proof. This follows from (4.3) and the lower bound of Theorem 2.2. □

Now let $x = \omega$, the volumetric center of $\mathcal{P}$. We will use Lemmas 4.2 and 4.3 to establish a lower bound on $\tilde{V}(\tilde{\omega}) - V(\omega)$ when $\tilde{\mathcal{P}}$ is obtained by placing a central cut through ω.

THEOREM 4.4. *Suppose that* ω *is the volumetric center of* $\mathcal{P}$, $s = s(\omega)$, *and* $\sigma = \sigma(s)$. *Let* $\tilde{\mathcal{P}} = \{\tilde{x} \,|\, A\tilde{x} \geq b,\, a^T\tilde{x} \geq a^T\omega\}$. *Let* $\tilde{V}(\cdot)$ *be the volumetric barrier for* $\tilde{\mathcal{P}}$, *and* $\tilde{\omega}$ *be the volumetric center. Then* $\tilde{V}(\tilde{\omega}) - V(\omega) \geq (2\sqrt{\sigma_{\min}} - \sigma_{\min})/10$.

Proof. Let $r = \|\tilde{\omega} - \omega\|_Q$, and assume for the moment that $r = \delta/\mu$ for some $\delta \in [0,1]$, $\mu = \mu(\omega)$. Using (4.1), Lemmas 4.2 and 4.3, and the fact that $g = g(\omega) = 0$, we obtain

$$\tilde{V}(\tilde{\omega}) - V(\omega) \geq \frac{1}{2}\ln\left(1 + \frac{(1-\delta)^2\mu^2\sigma_{\min}}{\delta^2}\right) + \frac{\delta^2}{2(1+\delta)^2\mu^2} \ . \tag{4.5}$$

Next we use the fact that $\ln(1+\lambda) \geq \lambda/(1+\lambda)$, for all $\lambda \geq 0$, to obtain

$$\begin{aligned}\ln\left(1 + \frac{(1-\delta)^2\mu^2\sigma_{\min}}{\delta^2}\right) &\geq \frac{(1-\delta)^2\mu^2\sigma_{\min}}{\delta^2 + (1-\delta)^2\mu^2\sigma_{\min}} \\ &\geq \frac{(1-\delta)^2}{\mu^2[4\delta^2 + (1-\delta)^2]} \\ &\geq \frac{(1-\delta)^2}{\mu^2(1+\delta)^2} \ , \end{aligned} \tag{4.6}$$

where the second inequality uses $\mu \geq 1$, and

$$\mu^4\sigma_{\min} = \frac{\sigma_{\min}}{(2\sqrt{\sigma_{\min}} - \sigma_{\min})^2} = \frac{1}{4 - 4\sqrt{\sigma_{\min}} + \sigma_{\min}} \geq \frac{1}{4},$$

and the final inequality uses $\delta^2 \leq \delta$. Substituting (4.6) into (4.5) then gives

$$\tilde{V}(\tilde{\omega}) - V(\omega) \geq \frac{1}{2\mu^2}\left(\frac{(1-\delta)^2 + \delta^2}{(1+\delta)^2}\right). \tag{4.7}$$

A straightforward differentiation shows that the minimum of the right-hand side of (4.7), for $0 \leq \delta \leq 1$, occurs at $\delta = 2/3$, with value $1/(10\mu^2)$. From (4.7) we then have $V(\tilde{\omega}) - V(\omega) \geq 1/(10\mu^2) = (2\sqrt{\sigma_{\min}} - \sigma_{\min})/10$.

Next assume that $r > 1/\mu$. Then there is an $\alpha \in (0,1)$ so that $\bar{x} = \omega + \alpha(\tilde{\omega} - \omega)$ has $\|\bar{x} - \omega\|_Q = 1/\mu$. From the convexity of $V(\cdot)$ and Lemma 4.3, we obtain

$$V(\tilde{\omega}) \geq V(\bar{x}) \geq V(\omega) + \frac{2\sqrt{\sigma_{\min}} - \sigma_{\min}}{8},$$

and (4.1) certainly implies that $\tilde{V}(\tilde{\omega}) \geq V(\tilde{\omega})$. It follows that $\tilde{V}(\tilde{\omega}) - V(\omega) \geq (2\sqrt{\sigma_{\min}} - \sigma_{\min})/8$. □

It is worthwhile to mention that the analysis in [18, section 4.1] actually shows that $\tilde{V}(\tilde{\omega}) - V(\omega) = \Omega(\sqrt{\sigma_{\min}})$, although the authors of [18] do not note this fact. In practice the added cut $a^T\tilde{x} \geq a^T x$ cannot be passed through $x = \omega$ as in Theorem 4.4 but rather through a point x which is close to ω in some sense. As a result, the lower bound of Theorem 4.4 must be modified to account for the use of $x \neq \omega$. In the next theorem we give a result based on particular parameter choices used throughout the paper. In the proof of the theorem we numerically evaluate some functions of one variable, as opposed to using weaker analytical bounds, in order to obtain the best possible result.

THEOREM 4.5. *Let x have $s = s(x) > 0$, $\|g\|_{Q^{-1}} \leq .01$, and $\sigma_{\min} = \sigma_{\min}(s) \geq .04$. Let $\tilde{\mathcal{P}} = \{\tilde{x} \mid A\tilde{x} \geq b,\ a^T\tilde{x} \geq a^T x\}$, $\tilde{V}(\cdot)$ be the volumetric barrier for $\tilde{\mathcal{P}}$, and $\tilde{\omega}$ be the volumetric center. Then $\tilde{V}(\tilde{\omega}) - V(x) \geq .0340$.*

Proof. Let $r = \|\tilde{\omega} - x\|_Q$, and assume for the moment that $r = \delta/\mu$ for $0 \leq \delta \leq 1$, $\mu = \mu(x)$. Proceeding as in the proof of Theorem 4.4, but including the effect of $g = g(x) \neq 0$, we have

$$\begin{aligned} \tilde{V}(\tilde{\omega}) - V(x) &\geq \frac{1}{2}\ln\left(1 + \frac{(1-\delta)^2\mu^2\sigma_{\min}}{\delta^2}\right) + g^T(\tilde{\omega} - x) + \frac{\delta^2}{2(1+\delta)^2\mu^2} \\ &\geq \frac{1}{2}\ln\left(1 + \frac{(1-\delta)^2\mu^2\sigma_{\min}}{\delta^2}\right) - \frac{\delta\|g\|_{Q^{-1}}}{\mu} + \frac{\delta^2}{2(1+\delta)^2\mu^2}, \end{aligned} \tag{4.8}$$

where the second inequality uses the fact that $|g^T(\tilde{\omega} - x)| \leq \|g\|_{Q^{-1}}\|\tilde{\omega} - x\|_Q$. We distinguish two cases.

Case 1. $\sigma_{\min} \leq .04725$. Note that $1/\mu^2 = 2\sqrt{\sigma_{\min}} - \sigma_{\min}$ is monotonically increasing in $\sigma_{\min}$, so $\sigma_{\min} \geq .04$ implies that $1/\mu^2 \geq 2\sqrt{.04} - .04 = .36$. In addition, $\mu^2\sigma_{\min} = \sigma_{\min}/(2\sqrt{\sigma_{\min}} - \sigma_{\min})$ is monotonically increasing in $\sigma_{\min}$, so $\sigma_{\min} \geq .04$ also implies that $\mu^2\sigma_{\min} \geq .04/.36 = 1/9$. Finally, $\sigma_{\min} \leq .04725$ implies that $\mu \geq (2\sqrt{.04725} - .04725)^{-1/2} > 1.606$. Using these facts in (4.8) and the assumption that $\|g\|_{Q^{-1}} \leq .01$, we obtain

$$\tilde{V}(\tilde{\omega}) - V(x) \geq \frac{1}{2}\ln\left(1 + \frac{(1-\delta)^2}{9\delta^2}\right) - \frac{.01\delta}{1.606} + \frac{.18\delta^2}{(1+\delta)^2}. \tag{4.9}$$

It can be verified numerically that the minimum of the right-hand side in (4.9), for $\delta \in [0, 1]$, occurs at approximately $\delta = .8$, with value greater than .0340. (See Figure 4.1, Case 1, for a plot of the right-hand side of (4.9) for $\delta \in [.78, .83]$.)

Case 2. $\sigma_{\min} \geq .04725$. In this case we have $1/\mu^2 \geq 2\sqrt{.04725} - .04725 > .3874$, $\sigma_{\min}\mu^2 \geq .04725/(2\sqrt{.04725} - .04725) > .1219$. Using these facts in (4.8), with $\mu \geq 1$ and the assumption that $\|g\|_{Q^{-1}} \leq .01$, we obtain

$$\tilde{V}(\tilde{\omega}) - V(x) \geq \frac{1}{2}\ln\left(1 + \frac{.1219(1-\delta)^2}{\delta^2}\right) - .01\delta + \frac{.1937\delta^2}{(1+\delta)^2}. \tag{4.10}$$

It can be verified numerically that the minimum of the right-hand side in (4.10), for $\delta \in [0, 1]$, occurs at approximately $\delta = .81$, with value greater than .0340. (See Figure 4.1, Case 2, for a plot of the right-hand side of (4.9) for $\delta \in [.78, .83]$.)

This completes the proof under the assumption that $r \leq 1/\mu$. However, arguing as at the end of Theorem 4.4, it is easy to show that if $\|\tilde{\omega} - x\|_Q > 1/\mu$, then

$$\tilde{V}(\tilde{\omega}) - V(x) \geq -\frac{\|g\|_{Q^{-1}}}{\mu} + \frac{1}{8\mu^2} \geq -.01 + \frac{.36}{8} = .035. \qquad □$$

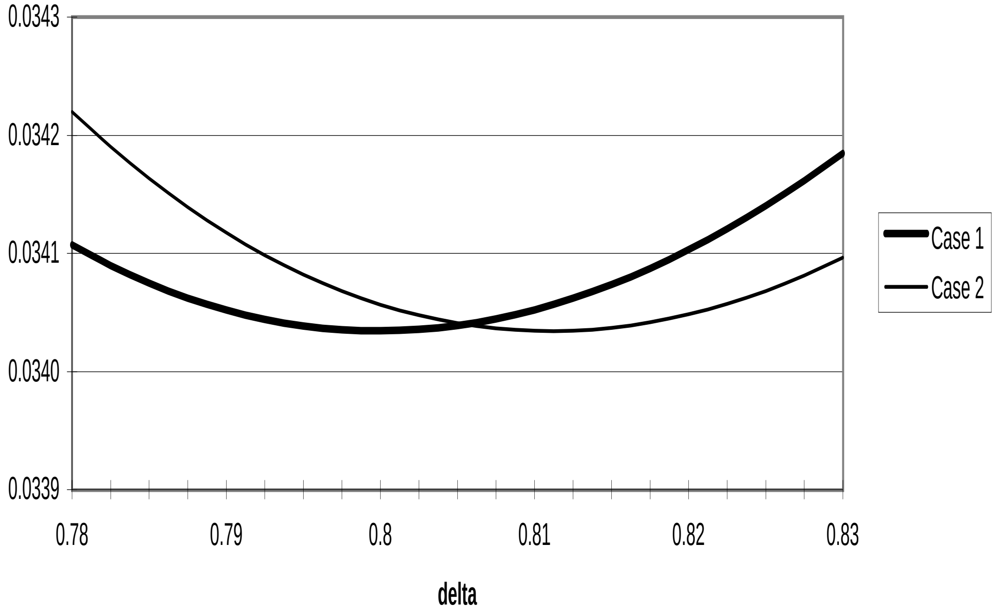


FIG. 4.1. *Lower bound on* $\tilde{V}(\tilde{\omega}) - V(x)$ *versus* δ.

For the final topic of the section, we consider moving off of the cut $a^T\tilde{x} \geq a^Tx$ to a new point $\bar{x}$ having $a^T\bar{x} > a^Tx$. Our goal is to obtain an upper bound for the quantity $\tilde{V}(\bar{x}) - V(x)$. For $0 < \delta \leq 1$, consider a point of the form

$$\bar{x} = x + \delta\,\frac{(A^TS^{-2}A)^{-1}a}{\sqrt{a^T(A^TS^{-2}A)^{-1}a}}\ . \tag{4.11}$$

THEOREM 4.6. *Suppose that* x *has* $s = s(x) > 0$. *Let* $\tilde{\mathcal{P}} = \{\tilde{x}\,|\,A\tilde{x} \geq b,\ a^T\tilde{x} \geq a^Tx\}$, *and let* $\tilde{V}(\cdot)$ *be the volumetric barrier for* $\tilde{\mathcal{P}}$. *Then using* $\delta = .3$ *in* (4.11) *produces* $\bar{x}$ *having* $\tilde{V}(\bar{x}) \leq V(x) + .3\|g\|_{Q^{-1}} + 1.78$.

Proof. Let $H = A^TS^{-2}A$, $\bar{H} = A^T\bar{S}^{-2}A$. By construction we then have $a^T\bar{x} - a^Tx = \delta\|a\|_{H^{-1}}$, and also

$$\|S^{-1}A(\bar{x} - x)\| = \delta. \tag{4.12}$$

It follows that

$$\frac{a^T(A^T\bar{S}^{-2}A)^{-1}a}{(a^T\bar{x} - a^Tx)^2} = \frac{a^T(A^T\bar{S}^{-2}A)^{-1}a}{\delta^2 a^T(A^TS^{-2}A)^{-1}a} \leq \frac{(1+\delta)^2}{\delta^2}, \tag{4.13}$$

where the last inequality uses (4.12), Proposition 2.1, and the fact that $H \preceq (1+\delta)^2\bar{H} \Rightarrow \bar{H}^{-1} \preceq (1+\delta)^2H^{-1}$ (see, for example, [10, Corollary 7.7.4]). Let $\xi = \bar{x} - x$. Then from (4.12) and Theorem 2.2, we have

$$\begin{aligned} V(\bar{x}) &\leq V(x) + g^T\xi + \xi^TQ\xi\frac{3+\delta^2}{2(1-\delta)^2} \\ &\leq V(x) + \|g\|_{Q^{-1}}\|\xi\|_Q + \|\xi\|_Q^2\frac{3+\delta^2}{2(1-\delta)^2} \\ &\leq V(x) + \delta\|g\|_{Q^{-1}} + \frac{\delta^2(3+\delta^2)}{2(1-\delta)^2}, \end{aligned} \tag{4.14}$$

where the last inequality uses the facts that $Q \preceq H$ and $\|\xi\|_H = \delta^2$ from (4.12). Combining (4.1), (4.13), and (4.14), we obtain

$$\tilde{V}(\bar{x}) - V(x) \le \delta\|g\|_{Q^{-1}} + \frac{\delta^2(3+\delta^2)}{2(1-\delta)^2} + \frac{1}{2}\ln\left(1 + \frac{(1+\delta)^2}{\delta^2}\right). \tag{4.15}$$

The proof is completed by substituting $\delta = .3$ into (4.15). □

Note that $\bar{x}$ in (4.11) is based on $H = A^T S^{-2} A$, the Hessian of the logarithmic barrier at x, and not Q, as used in [18, section 4.1.2]. However, we must have $Q \preceq H \preceq (1/\epsilon)Q$ since $\sigma_{\min} \ge \epsilon$ whenever a constraint is added, and therefore a step based on Q can also be analyzed using methods similar to those employed here. The advantage of our approach, using H, is that we obtain a result which is independent of ϵ.

5. Dropping a constraint. In this section we consider the effect of dropping a constraint, as in Step 4 of the algorithm. For simplicity we assume that $\sigma_m = \sigma_{\min}$, and let $\tilde{\mathcal{P}}$ be the new constraint system obtained by deleting the mth constraint in the original system $[A, b]$ defining $\mathcal{P}$. Throughout we use the tilde (˜) notation to denote quantities related to the reduced constraint system $[\tilde{A}, \tilde{b}]$.

THEOREM 5.1. *Suppose that x has $s = s(x) > 0$, $\sigma_m = \sigma_{\min}$, and $\tilde{\mathcal{P}}$ is obtained by deleting the mth constraint defining $\mathcal{P}$. Then*

1. $\tilde{V}(x) = V(x) + (1/2)\ln(1 - \sigma_{\min})$,
2. $\sigma_i \le \tilde{\sigma}_i \le \sigma_i/(1-\sigma_{\min}), \quad i = 1, \ldots, m-1$,
3. $\|\tilde{g}\|_{\tilde{Q}^{-1}} \le \frac{1}{\sqrt{1-\sigma_{\min}}}\left(\|g\|_{Q^{-1}} + \sigma_{\min}\left(1 + \frac{1}{\sqrt{1-\sigma_{\min}}}\right)\right)$.

Proof. See [3, Lemma 5.1, Lemma 5.2, and Theorem 5.3]. □

We will use Theorem 5.1 to bound the change in our fundamental proximity measure $\hat{\mu}\|g\|_{Q^{-1}}$ following the deletion of a constraint. We use $\hat{\tilde{\mu}} = \hat{\tilde{\mu}}(x)$ to denote the value of $\hat{\mu}$ with respect to the reduced constraint system $[\tilde{A}, \tilde{b}]$.

THEOREM 5.2. *Assume that x has $s = s(x) > 0$, $\hat{\mu}\|g\|_{Q^{-1}} \le .01$, and $\sigma_m = \sigma_{\min} \le .04$. Let $\tilde{\mathcal{P}}$ be obtained by deleting the mth constraint defining $\mathcal{P}$. Then $\tilde{\mathcal{P}}$ is bounded, $\tilde{V}(\tilde{\omega}) \ge \tilde{V}(x) - .0121$, and $\tilde{V}(\tilde{\omega}) \ge V(x) - .0326$, where $\tilde{\omega}$ is the volumetric center of $\tilde{\mathcal{P}}$.*

Proof. Note that $\hat{\tilde{\mu}} \le \hat{\mu} \le \mu$, from part 2 of Theorem 5.1, and the fact that $\tilde{m} = m - 1$. Applying part 3 of Theorem 5.1, we obtain

$$\begin{aligned}
\hat{\tilde{\mu}}\|\tilde{g}\|_{\tilde{Q}^{-1}} &\le \frac{1}{\sqrt{1-\sigma_{\min}}}\left(\hat{\mu}\|g\|_{Q^{-1}} + \sigma_{\min}\mu\left(\frac{\hat{\tilde{\mu}}}{\mu}\right)\left(1 + \frac{1}{\sqrt{1-\sigma_{\min}}}\right)\right) \\
&\le \frac{1}{\sqrt{1-\sigma_{\min}}}\left(.01 + \frac{\sigma_{\min}^{3/4}}{(2-\sqrt{\sigma_{\min}})^{1/2}}\left(\frac{\hat{\tilde{\mu}}}{\mu}\right)\left(1 + \frac{1}{\sqrt{1-\sigma_{\min}}}\right)\right), \qquad (5.1)
\end{aligned}$$

where the second inequality uses the assumption that $\hat{\mu}\|g\|_{Q^{-1}} \le .01$. It is clear that the right-hand side of (5.1) is increasing in $\sigma_{\min}$, and substituting $\sigma_{\min} = .04$ into (5.1) results in

$$\hat{\tilde{\mu}}\|\tilde{g}\|_{\tilde{Q}^{-1}} \le .0103 + .1375\frac{\hat{\tilde{\mu}}}{\mu}. \tag{5.2}$$

Assume for the moment that $\hat{\tilde{\mu}} \le .833\mu$. Then (5.2) implies that $\hat{\tilde{\mu}}\|\tilde{g}\|_{\tilde{Q}^{-1}} \le .125$, so

$$\tilde{V}(\tilde{\omega}) - \tilde{V}(x) \ge -\frac{.0113}{\hat{\tilde{\mu}}^2} \ge -.0113$$

from Corollary 2.7. Alternatively, assume that $\hat{\hat{\mu}} \geq .833\mu$. Since in any case $\hat{\hat{\mu}} \leq \mu$, (5.2) implies that $\hat{\hat{\mu}}\|\tilde{g}\|_{\tilde{Q}^{-1}} \leq .1478 < 4/27$. In addition, $\sigma_{\min} \leq .04$ implies $\mu \geq (2\sqrt{.04} - .04)^{-1/2} = 5/3$, and therefore $\hat{\hat{\mu}} \geq .833(5/3) > 1.388$. From Corollary 2.7 we then have

$$\tilde{V}(\tilde{\omega}) - \tilde{V}(x) \geq -\frac{.0232}{1.388^2} \geq -.0121,$$

so in all cases $\tilde{V}(\tilde{\omega}) - \tilde{V}(x) \geq -.0121$, as claimed. In addition, we have

$$\tilde{V}(\tilde{\omega}) = V(x) + [\tilde{V}(x) - V(x)] + [\tilde{V}(\tilde{\omega}) - \tilde{V}(x)], \tag{5.3}$$

and part 1 of Theorem 5.1 gives

$$\tilde{V}(x) - V(x) = \frac{1}{2}\ln(1 - \sigma_{\min}) \geq \frac{1}{2}\ln(.96) \geq -.0205. \tag{5.4}$$

Then $\tilde{V}(\tilde{\omega}) \geq V(x) - .0326$ follows from (5.3), (5.4), and $\tilde{V}(\tilde{\omega}) - \tilde{V}(x) \geq -.0121$. □

6. Conclusion.

From a practical standpoint, this paper gives the best result to date for a cutting plane method for the convex feasibility problem based on the volumetric barrier. From the standpoint of theoretical complexity, the most interesting open problem is how to use central cuts with the volumetric barrier, while requiring only $O(1)$ Newton (or Newton-like) steps following the introduction of a cut, as is possible when shallow cuts are employed [20], [3]. Although the affine step (4.11) is sufficient to obtain an $O(1)$ bound on $\tilde{V}(\bar{x}) - V(x)$, as in Theorem 4.6, this bound is too weak relative to $\bar{\sigma}_{\min}$ to show that $O(1)$ steps suffice to return to a suitable proximity of the new volumetric center $\tilde{\omega}$. As a result, it becomes necessary to use a proximity measure based on $\hat{\mu}$ in place of μ, leading to a worst-case decrease of $\Omega(1/\sqrt{n})$ instead of $\Omega(1)$ in the steps on Step 5 of the algorithm. In practice the algorithm might of course do much better than these worst-case bounds indicate, but serious computational work using the volumetric barrier has not yet been conducted.

For the *analytic center* cutting plane method it is relatively easy to show that $O(1)$ steps suffice to return to a suitable proximity of the new analytic center following the addition of a central cut [7]. The complexity analysis for the analytic center cutting plane method can also be extended to multiple cuts [13], [17], [22], and deep cuts [6], [8]. (It should be noted that most versions of the analytic center cutting plane method are not polynomial-time algorithms, but the analysis in [17] can be applied to the polynomial-time version of Atkinson and Vaidya [4].) Similar results for the volumetric cutting plane method would be desirable. In [18] a result allowing multiple cuts is developed, but in addition to the very small constants required throughout [18], the multiple cut result requires a "selective orthonormalization" procedure that weakens the original cuts in the interest of constructing a feasible affine step.

REFERENCES

[1] K. M. Anstreicher, *Large step volumetric potential reduction algorithms for linear programming*, Ann. Oper. Res., 62 (1996), pp. 521–538.

[2] K. M. Anstreicher, *Volumetric path following algorithms for linear programming*, Math. Programming, 76 (1997), pp. 245–263.

[3] K. M. Anstreicher, *On Vaidya's volumetric cutting plane method for convex programming*, Math. Oper. Res., 22 (1997), pp. 63–89.

[4] D. S. Atkinson and P. M. Vaidya, *A cutting plane algorithm for convex programming that uses analytic centers*, Math. Programming, 69 (1995), pp. 1–43.

[5] R. G. Bland, D. Goldfarb, and M. J. Todd, *The ellipsoid method: A survey*, Oper. Res., 29 (1981), pp. 1039–1091.

[6] J.-L. Goffin, *Using the Primal Dual Infeasible Newton Method in the Analytic Center Method for Problems Defined by Deep Cutting Planes*, Tech. report, Faculty of Management, McGill University, Montreal, Canada, 1994.

[7] J.-L. Goffin, Z.-Q. Luo, and Y. Ye, *Complexity analysis of an interior cutting plane method for convex feasibility problems*, SIAM J. Optim., 6 (1996), pp. 638–652.

[8] J.-L. Goffin and J.-Ph. Vial, *Shallow, Deep, and Very Deep Cuts in the Analytic Center Cutting Plane Method*, Logilab Tech. report 96.1, Dept. of Management Studies, University of Geneva, Switzerland, 1996.

[9] M. Grötschel, L. Lovasz, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.

[10] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[11] K. C. Kiwiel, *Complexity of some cutting plane methods that use analytic centers*, Math. Programming, 74 (1996), pp. 47–54.

[12] A. Y. Levin, *On an algorithm for minimizing a convex function*, Soviet Math. Dokl., 6 (1965), pp. 286–290.

[13] Z.-Q. Luo, *Analysis of a cutting plane method that uses weighted analytic center and multiple cuts*, SIAM J. Optim., 7 (1997), pp. 697–716.

[14] J. E. Mitchell and M. J. Todd, *Solving combinatorial optimization problems using Karmarkar's algorithm*, Math. Programming, 56 (1992), pp. 245–284.

[15] Y. Nesterov, *Complexity estimates of some cutting plane methods based on analytical barrier*, Math. Programming, 69 (1995), pp. 149–176.

[16] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, PA, 1994.

[17] S. Ramaswamy and J. E. Mitchell, *On Updating the Analytic Center after the Addition of Multiple Cuts*, Tech. report 37-94-423, Decision Sciences and Engineering Systems Dept., Rensselaer Polytechnic Institute, Troy, NY, 1994.

[18] S. Ramaswamy and J. E. Mitchell, *A Long Step Cutting Plane Algorithm that Uses the Volumetric Barrier*, Tech. report, Dept. of Math. Sciences, Rensselaer Polytechnic Institute, Troy, NY, 1995.

[19] S. P. Tarasov, L. G. Khachiyan, and I. I. Erlich, *The method of inscribed ellipsoids*, Soviet Math. Dokl., 37 (1988), pp. 226–230.

[20] P. M. Vaidya, *A new algorithm for minimizing convex functions over convex sets*, Math. Programming, 73 (1996), pp. 291–341.

[21] B. Yamnitsky and L. A. Levin, *An old linear programming algorithm runs in polynomial time*, in Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, IEEE, New York, 1982, pp. 327–328.

[22] Y. Ye, *Complexity analysis of the analytic center cutting plane method that uses multiple cuts*, Math. Programming, 78 (1997), pp. 85–104.

SIAM J. Optim.
Vol. 9, No. 1, pp. 207–216

ON THE DIMENSION OF THE SET OF RIM PERTURBATIONS FOR OPTIMAL PARTITION INVARIANCE[*]

HARVEY J. GREENBERG[†], ALLEN G. HOLDER[†], KEES ROOS[‡], AND TAMÁS TERLAKY[‡]

Abstract. Two new dimension results are presented. For linear programs, it is shown that the sum of the dimension of the optimal set and the dimension of the set of objective perturbations for which the optimal partition is invariant equals the number of variables. A decoupling principle shows that the primal and dual results are additive. The main result is then extended to convex quadratic programs, but the dimension relationships are no longer dependent only on problem size. Furthermore, although the decoupling principle does not extend completely, the dimensions are additive, as in the linear case.

Key words. linear programming, optimal partition, polyhedron, polyhedral combinatorics, quadratic programming, computational economics

AMS subject classification. 90C05

PII. S1052623497316798

1. Introduction and background.

Consider the primal-dual linear programs (LPs)

$$\min\{cx : x \geq 0,\ Ax = b\}, \qquad \max\{yb : s \geq 0,\ yA + s = c\},$$

where c is a row vector in $\mathbb{R}^n$, called *objective coefficients*; x is a column vector in $\mathbb{R}^n$, called *levels*; b is a column vector in $\mathbb{R}^m$, called *right-hand sides*; y is a row vector in $\mathbb{R}^m$ called *prices*; and A is an $m \times n$ matrix with rank m.

Let P and D denote the primal and dual polyhedra, respectively, and let P^* and D^* denote their optimality regions, which we assume to be nonempty. Let (x^*, y^*, s^*) be a strictly complementary optimal solution, and let the optimal partition be denoted by $(B|N)$, where

$$B = \sigma(x^*) \equiv \{j : x_j^* > 0\} \qquad \text{and} \qquad N = \sigma(s^*) \equiv \{j : s_j^* > 0\}.$$

(For background, see [6].)

This paper first presents a result concerning the dimension of P^* (D^*) in connection with the set of direction vectors in $\mathbb{R}^n$ (respectively, $\mathbb{R}^m$) for which the optimal partition does not change when the objective coefficients (respectively, right-hand sides) are perturbed in that direction. After establishing fundamental relations for LPs, we consider extensions to convex quadratic programs. The technical terms used throughout this paper are defined in the *Mathematical Programming Glossary* [2].

[*] Received by the editors February 10, 1997; accepted for publication (in revised form) February 9, 1998; published electronically December 2, 1998.

http://www.siam.org/journals/siopt/9-1/31679.html

[†] University of Colorado at Denver, Mathematics Department, P.O. Box 173364, Denver, CO 80217-3364 (hgreenbe@carbon.cudenver.edu, http://www-math.cudenver.edu/~hgreenbe; agholder@tiger.cudenver.edu, http://www-math.cudenver.edu/~agholder).

[‡] Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft, The Netherlands (c.roos@twi.tudelft.nl; t.terlaky@twi.tudelft.nl, http://www.twi.tudelft.nl/People/Staf/T.Terlaky.html).

2. Linear programs. Following Greenberg [3], let $r = (b, c)$ denote the *rim data*, and let H denote the set of rim direction vectors, $h = (\delta b, \delta c)$, for which the optimal partition does not change on the interval $[r, r + \theta\, h]$ for some $\theta > 0$, i.e.,

$$\begin{aligned} H = \{(\delta b, \delta c) : \text{there is } x \geq 0,\ y \geq 0,\ \theta > 0 \quad &\text{such that} \\ Ax = b + \theta\, \delta b,\ x_B > 0,\ x_N = 0&; \\ yA + s = c + \theta\, \delta c,\ s_B = 0,\ s_N > 0&\}. \end{aligned}$$

Here we follow the notation in [1, 6], where a subscript on a vector means it is the subvector restricted to the indexes in the subscript. For example, x_B is the vector of positive levels. This notation extends to matrices: A partitions into $[A_B\, A_N]$.

Let H_c denote the projection of H onto $\mathbb{R}^n$ for changing only c:

$$H_c \equiv \{\delta c : (0, \delta c) \in H\}.$$

Similarly, let H_b denote the projection of H onto $\mathbb{R}^m$ for changing only b:

$$H_b \equiv \{\delta b : (\delta b, 0) \in H\}.$$

Greenberg [3] showed that H is a convex cone that satisfies a *decoupling principle*: $H = H_b \times H_c$.

To help build intuition, notice first that if the dimension of the primal optimality region, $\dim(P^*)$, is zero, this means it is an extreme point. In that case, every vector in $\mathbb{R}^n$ can be used to change c without changing the optimal partition, so $\dim(H_c) = n$. At the other extreme, suppose $\dim(P^*) = n - m$, such as when $c = 0$, so every feasible solution is optimal in the primal LP. In that case, H_c consists of change vectors that maintain equal net effects among the positive variables, so $\dim(H_c) = m$. This latter case can be illustrated with the following.

Example. $\min\{\sum_j 0x_j : \sum_j x_j = 1, x \geq 0\}$.

In this case, $B = \{1, \ldots, n\}$. In order for this partition not to change for the LP: $\min\{\sum_j \delta c_j x_j : \sum_j x_j = 1, x \geq 0\}$, it is necessary and sufficient that $\delta c_j = \delta c_1$ for all j. Thus, $\dim(H_c) = 1$. □

In both cases, we see that $\dim(P^*) + \dim(H_c) = n$. This is what we shall prove in general along with related results.

THEOREM 2.1. *The following equations hold for any LP whose primal and dual sets have nonempty strict interiors.*

1. $\dim(P^*) + \dim(H_c) = n$.
2. $\dim(D^*) + \dim(H_b) = m$.
3. $\dim(P^* \times D^*) + \dim(H) = n + m$.

Proof. From Lemma IV.44 in [6], we have $\dim(P^*) = |B| - \operatorname{rank}(A_B)$. The conditions for $\delta c \in H_c$ are

$$yA_B = c_B + \theta\, \delta c_B \text{ and } yA_N < c_N + \theta\, \delta c_N$$

for some $\theta > 0$. Thus, δc_N can be arbitrary, so

$$\begin{aligned} \dim(H_c) &= |N| + \dim(\{\delta c_B : \exists \delta y \in \mathbb{R}^m \ni \delta y A_B = \delta c_B\}) \\ &= |N| + \operatorname{rank}(A_B). \end{aligned}$$

This implies $\dim(P^*) + \dim(H_c) = |B| + |N| = n$.

The second statement has a similar argument. From Lemma IV.44 in [6], $\dim(D^*) = m - \operatorname{rank}(A_B)$. The conditions for $\delta b \in H_b$ are

$$A_B x_B = b + \theta\ \delta b \text{ and } x_B > 0.$$

Thus, $\dim(H_b) = \operatorname{rank}(A_B)$, so $\dim(D^*) + \dim(H_b) = m$. The last statement follows from the decoupling principle, upon adding the first two equations $H = H_b \times H_c \Rightarrow \dim(H) = \dim(H_b) + \dim(H_c)$. □

We now consider some corollaries whose proofs follow directly from the theorem but whose meanings lend insight into how perturbation relates to the dimensions of the primal and dual optimality regions.

The dimension of a set is sometimes called the *degrees of freedom*. If there are n variables and no constraints on their values, the set has the full degrees of freedom, which is n; i.e., each variable can vary independently. When the set is defined by a system of m independent equations, as in our case, we sometimes refer to m as the *degrees of freedom lost*. Because we assume that there exists a strict interior solution ($x > 0$), there are no implied equalities among the nonnegativity constraints, so $\dim(P) = n - m$. Thus, the feasibility region has m degrees of freedom lost due to the equations that relate the variables.

A meaningful special case is when there is an excess number of columns, say $|B| = m + k$, and there is enough linear independence retained in the columns so that $\operatorname{rank}(A_B) = m$ (recall that we assume $\operatorname{rank}(A) = m$). Then, $\dim(P^*) = k$, so $\dim(H_c) = n - k$. Expressed in words, the degrees of freedom lost in varying objective coefficients equals the number of excess columns over those of a basic optimal solution. Furthermore, $\operatorname{rank}(A_B) = m$ is equivalent to $\dim(D^*) = 0$ (i.e., unique dual solution), so we can say the following.

Corollary 2.2. *The following are equivalent.*

1. *The dual solution is unique.*
2. $\dim(H_c) = n + m - |B|$.
3. $\dim(H_b) = m$.

Another special case arises when the LP is a conversion from the inequality constraints, $A'x \geq b$, where A' is $m \times n'$, and $\operatorname{rank}(A') = m$. In that case, $A = [A' \ -I]$, and $n = n' + m$. Suppose $x^* > 0$, so B includes all of the structural variables and some of the surplus variables, say $|B| = n' + k$. Then, $\dim(P^*) = k$, and Theorem 2.1 implies $\dim(H_c) = n' + m - k$. Since we do not allow the costs of the surplus variables to be nonzero, we can reduce this by m, giving $\dim(H_{c'}) = n' - k$. Expressed in words, this says that the degrees of freedom lost in varying (structural) cost coefficients equals the number of positive surplus variables.

A similar result follows for the primal. The next corollary says, in part, that $\dim(P^*) = 0$ if and only if $\dim(H_c) = n$. Expressed in words, this says that the primal solution is unique if and only if every objective coefficient can be perturbed independently without changing the optimal partition. The last equivalence includes the special case of a nondegenerate basic solution, in which case $|B| = m$, so every right-hand side can be perturbed without changing the optimal partition.

Corollary 2.3. *The following are equivalent.*

1. *The primal solution is unique.*
2. $\dim(H_c) = n$.
3. $\dim(H_b) = |B|$.

These corollaries combine into the following, which is the familiar case of a unique strictly complementary optimum (which is basic).

Corollary 2.4. *The following are equivalent.*

1. *The primal-dual solution is unique.*
2. $\dim(H_c) = n$ *and* $\dim(H_b) = m$.
3. $\dim(H) = m + n$.

The following corollary says that $\dim(H_c) \geq m$, and it follows from the main theorem since the maximum dimension of P^* is $n - m$. (The analogous bound for $\dim(H_b)$ is merely that it is nonnegative since the maximum dimension of D^* is m.)

COROLLARY 2.5. *There are at least m degrees of freedom to vary the objective coefficients without changing the optimal partition.*

In the next section, we extend Theorem 2.1 to convex quadratic programs, and note that care must be taken when specializing it to an LP.

3. Quadratic programs. We now extend Theorem 2.1 to the convex quadratic program

$$\min\{cx + \tfrac{1}{2}x^T Qx : Ax = b,\ x \geq 0\},$$

where Q is symmetric and positive semidefinite. We use the Wolfe dual [2]

$$\max\{yb - \tfrac{1}{2}u^T Qu : yA + s - u^T Q = c,\ s \geq 0\}.$$

Let QP and QD denote primal and dual feasibility regions, respectively. Let us introduce the following notation:

$$\begin{aligned} QP^* &= \{x : x \in QP\ , \text{ and } x \text{ is primal optimal}\},\\ QD^* &= \{(y,s) : (y,s,u) \in QD \text{ and } (y,s,u) \text{ is dual optimal}\},\\ \mathcal{QD}^* &= \{(y,s,u) : (y,s,u) \in Q\mathcal{D} \text{ and } (y,s,u) \text{ is dual optimal}\}.\end{aligned}$$

Here QP^* and QD^* denote their optimality regions, except that we define QD^* exclusive of the u-variables, while $\mathcal{QD}^*$ denotes the full dual optimality region to distinguish it from QD^* . We shall explain this shortly.

Following Jansen [4] and Berkelaar, Roos, and Terlaky [1], an optimal partition is defined by three sets $(B|T|N)$, where

$$\begin{aligned} B &= \{j : x_j > 0 \text{ for some } x \in QP^*\ \},\\ N &= \{j : s_j > 0 \text{ for some } (y,s) \in QD^*\ \}, \text{ and}\\ T &= \{1,\ldots,n\} \setminus (B \cup N).\end{aligned}$$

We assume that the solution obtained is *maximal* [1]:

$$x_j > 0 \Longleftrightarrow j \in B \text{ and } s_j > 0 \Longleftrightarrow j \in N.$$

Güler and Ye [5] show that many interior point algorithms converge to a solution whose support sets comprise the maximal partition: $B = \sigma(x), N = \sigma(s)$, and $T = \{1,\ldots,n\} \setminus (B \cup N)$.

Unlike linear programming, there is no guarantee of a strictly complementary optimal solution, so T need not be empty. For this and other reasons, there are some important differences (see [1, 4] for details) that affect our extension of Theorem 2.1. In particular, the decoupling principle does not apply since a change in c affects both primal and dual optimality conditions.

We begin our extension with the following lemma. In the proof we use the following notation:

$$\begin{aligned} \mathrm{col}(G) &= \text{column space of } G &&= \{u : u = Gx \text{ for some } x \in \mathbb{R}^n\}, \\ \mathcal{N}(G) &= \text{null space of } G &&= \{x : Gx = 0\}. \end{aligned}$$

LEMMA 3.1. *Let F and G be $m \times n$ and $g \times n$ matrices, respectively, and consider the set: $S = \{v : v = Fu \text{ for some } u \ni Gu = 0\}$. Then, $\dim(S) = \mathrm{rank}\binom{F}{G} - \mathrm{rank}(G)$.*

Proof. Without loss in generality assume G has full row rank, and let $\{u_1, \ldots, u_g\}$ be a basis for $\mathrm{col}(G)$. Let $\{v_1, \ldots, v_s\}$ be a basis for S (where $\dim(S) = s$), and consider the following set in col $\binom{F}{G}$:

$$\left\{ \begin{pmatrix} w_1 \\ u_1 \end{pmatrix} \cdots \begin{pmatrix} w_g \\ u_g \end{pmatrix} \begin{pmatrix} v_1 \\ 0 \end{pmatrix} \cdots \begin{pmatrix} v_s \\ 0 \end{pmatrix} \right\},$$

where $w_i \equiv FG^T[GG^T]^{-1}u_i$. Once we prove that this is a basis for col $\binom{F}{G}$, we have that $g + s = \mathrm{rank}\binom{F}{G}$, which implies the desired result.

First, we shall prove that these vectors are linearly independent. Suppose

$$\sum_i \alpha_i \begin{pmatrix} w_i \\ u_i \end{pmatrix} + \sum_j \beta_j \begin{pmatrix} v_j \\ 0 \end{pmatrix} = 0.$$

Since $\{u_1, \ldots, u_g\}$ is a basis, $\alpha = 0$, which then implies $\beta = 0$, because $\{v_1, \ldots, v_s\}$ are also linearly independent.

Second, we shall prove that these vectors span col $\binom{F}{G}$. Let $\binom{v}{u} = \binom{F}{G} x$ for some $x \in \mathbb{R}^n$. Decompose $x = y + z$, where $y \in \mathrm{col}(G^T)$ and $z \in \mathcal{N}(G)$. Then, $Gx = Gy = GG^T\lambda$, where $y = G^T\lambda$, and $Fx = Fy + Fz$. Since $Fz \in S, Fx = FG^T\lambda + \sum_j \beta_j v_j$. We thus have $u = Gx = GG^T\lambda$, but since $u \in \mathrm{col}(G)$, $GG^T\lambda = \sum_i \alpha_i u_i$. This implies $\lambda = \sum_i \alpha_i [GG^T]^{-1} u_i$, so

$$\begin{aligned} Fx &= FG^T \textstyle\sum_i \alpha_i [GG^T]^{-1} u_i + \sum_j \beta_j v_j \\ &= \textstyle\sum_i \alpha_i FG^T [GG^T]^{-1} u_i + \sum_j \beta_j v_j. \end{aligned}$$

By the definition of w, we have derived α, β such that

$$\begin{pmatrix} v \\ u \end{pmatrix} = \textstyle\sum_i \alpha_i \begin{pmatrix} w_i \\ u_i \end{pmatrix} + \sum_j \beta_j \begin{pmatrix} v_j \\ 0 \end{pmatrix}. \qquad \square$$

To prove the main theorem, we use the following dimension results of Berkelaar, Roos, and Terlaky [1]:

$$\dim(QP) = |B| - \mathrm{rank}\begin{pmatrix} A_B \\ Q_{BB} \end{pmatrix}, \tag{3.1}$$

$$\dim(\mathcal{QD}^*) = m - \mathrm{rank}(A_B \; A_T) + n - \mathrm{rank}(Q). \tag{3.2}$$

The last portion, $n - \mathrm{rank}(Q)$, accounts for the u-variables because the dual conditions can use x^TQ in place of u^TQ, leaving u to appear only in the equation $Qu = Qx$. For our purposes it is not necessary (or desirable) to include this, so we define the dual optimality region exclusive of the u-variables:

$$QD^* = \{(y,s) : (y,s,x) \in \mathcal{QD}^* \text{ for some } x \in QP^*\}.$$

Then, (3.2) yields the dimension of the dual optimality region that we shall use:

$$\dim(QD^*) = m - \operatorname{rank}(A_B\ A_T). \tag{3.3}$$

As in the linear case, $s_N > 0$ implies that each component of c_N can vary independently, so $\dim(H)$ is the sum of $|N|$ and the dimension of the set of other possible changes. Keeping $x_{N\cup T} = 0$ and $s_{B\cup T} = 0$, the partition does not change if and only if there exists $(\delta y, \delta u, \delta x_B)$ to satisfy the following primal-dual conditions:

$$\begin{pmatrix} A_B^T & -Q_{B\bullet} & 0 \\ A_T^T & -Q_{T\bullet} & 0 \\ 0 & 0 & A_B \\ 0 & -Q & Q_{\bullet B} \end{pmatrix} \begin{pmatrix} \delta y \\ \delta u \\ \delta x_B \end{pmatrix} = \begin{pmatrix} \delta c_B \\ \delta c_T \\ \delta b \\ 0 \end{pmatrix}. \tag{3.4}$$

Here we follow the notation in [1]:

$$\begin{aligned} Q_{I\bullet} &= \text{rows of } Q \text{ associated with index set } I, \\ Q_{\bullet J} &= \text{columns of } Q \text{ associated with index set } J, \\ Q_{IJ} &= \text{submatrix of } Q \text{ associated with index sets } I \text{ and } J. \end{aligned}$$

The quadratic extensions rely on the fact that the rank of the matrix in (3.4) is related to the rank of the matrices found in statements (3.1) and (3.3). These relations are formalized in the following lemma.

Lemma 3.2. *The following relations hold for Q positive semidefinite:*

$$\operatorname{rank}(A_B\ A_T) + \operatorname{rank}\begin{pmatrix} A_B \\ Q_{BB} \end{pmatrix} = \operatorname{rank}\begin{pmatrix} A_B^T & -Q_{B\bullet} & 0 \\ A_T^T & -Q_{T\bullet} & 0 \\ 0 & 0 & A_B \\ 0 & -Q & Q_{\bullet B} \end{pmatrix} - \operatorname{rank}(Q) \tag{3.5}$$

$$= \operatorname{rank}\begin{pmatrix} A_B & A_T \\ Q_{BB} & Q_{BT} \end{pmatrix} + \operatorname{rank}(A_B). \tag{3.6}$$

Proof. To prove (3.5), performing elementary row and column operations on the large matrix (first on the right) produces the following matrix of the same rank:

$$\begin{pmatrix} A_B^T & 0 & -Q_{BB} \\ A_T^T & 0 & -Q_{TB} \\ 0 & 0 & A_B \\ 0 & Q & 0 \end{pmatrix}.$$

So,

$$\operatorname{rank}\begin{pmatrix} A_B^T & -Q_{B\bullet} & 0 \\ A_T^T & -Q_{T\bullet} & 0 \\ 0 & 0 & A_B \\ 0 & -Q & Q_{\bullet B} \end{pmatrix} = \operatorname{rank}\begin{pmatrix} A_B^T & -Q_{BB} \\ A_T^T & -Q_{TB} \\ 0 & A_B \end{pmatrix} + \operatorname{rank}(Q).$$

The positive semidefiniteness of Q implies that Q_{TB} is linearly dependent on Q_{BB} [1]. Hence,

$$\begin{pmatrix} A_B^T & -Q_{BB} \\ A_T^T & -Q_{TB} \\ 0 & A_B \end{pmatrix} \Rightarrow \begin{pmatrix} \tilde{A}_B^T & * & 0 & 0 \\ 0 & 0 & 0 & \tilde{Q}_{BB} \\ 0 & \tilde{A}_T^T & 0 & 0 \\ 0 & 0 & \tilde{A}_B & * \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} \tilde{A}_B^T & * & 0 & 0 \\ 0 & \tilde{A}_T^T & 0 & 0 \\ 0 & 0 & \tilde{A}_B & * \\ 0 & 0 & 0 & \tilde{Q}_{BB} \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

where $\Rightarrow$ is used to represent a series of row and column operations that preserve rank, and $*$ represents an arbitrary matrix of appropriate size. Hence,

$$\begin{aligned} \operatorname{rank}\begin{pmatrix} A_B^T & -Q_{B\bullet} & 0 \\ A_T^T & -Q_{T\bullet} & 0 \\ 0 & 0 & A_B \\ 0 & -Q & Q_{\bullet B} \end{pmatrix} &= \operatorname{rank}(Q) \\ &\quad + \operatorname{rank}\begin{pmatrix} \tilde{A}_B^T & * \\ 0 & \tilde{A}_T^T \end{pmatrix} + \operatorname{rank}\begin{pmatrix} \tilde{A}_B & * \\ 0 & \tilde{Q}_{BB} \end{pmatrix} \\ &= \operatorname{rank}(Q) \; + \; \operatorname{rank}\begin{pmatrix} A_B^T \\ A_T^T \end{pmatrix} + \operatorname{rank}\begin{pmatrix} A_B \\ Q_{BB} \end{pmatrix}, \end{aligned}$$

which yields the result.

The proof of (3.6) is similar, using the positive semidefiniteness property of Q in reducing the large matrix to row echelon form. □

We now have what we need to prove the following extension of Theorem 2.1.

THEOREM 3.3. *The following equations hold for any convex quadratic program whose primal and dual sets are not empty.*

1. $\dim(QP^*) + \dim(H_c) = n - |T| + \operatorname{rank}(A_B\ A_T) - \operatorname{rank}(A_B)$.
2. $\dim(QD^*) + \dim(H_b) = m - \operatorname{rank}(A_B\ A_T) + \operatorname{rank}(A_B)$.
3. $\dim(QP^* \times QD^*) + \dim(H) = n + m - |T|$.

Proof. To prove 1, we set $\delta b = 0$ in (3.4), and apply Lemmas 3.1 and 3.2 to produce the following:

$$\begin{aligned}\dim(H_c) &= |N| + \operatorname{rank}\begin{pmatrix} A_B^T & -Q_{B\bullet} & 0 \\ A_T^T & -Q_{T\bullet} & 0 \\ 0 & 0 & A_B \\ 0 & -Q & Q_{\bullet B}\end{pmatrix} - \operatorname{rank}\begin{pmatrix} 0 & -Q & Q_{\bullet B} \\ 0 & 0 & A_B\end{pmatrix} \\ &= |N| + \operatorname{rank}(A_B\ A_T) + \operatorname{rank}\begin{pmatrix} A_B \\ Q_{BB}\end{pmatrix} \\ &\quad + \operatorname{rank}(Q) - \operatorname{rank}(Q) - \operatorname{rank}(A_B) \\ &= |N| + \operatorname{rank}(A_B\ A_T) - \operatorname{rank}\begin{pmatrix} A_B \\ Q_{BB}\end{pmatrix} - \operatorname{rank}(A_B).\end{aligned}$$

Adding (3.1) to the last statement and substituting $n - |T| = |B| + |N|$ gives the first result. Similarly, to prove 2, set $\delta c_B = 0$ and $\delta c_T = 0$ in (3.4). Then, Lemma 3.1 implies the equation

$$\dim(H_b) = \operatorname{rank}\begin{pmatrix} A_B^T & -Q_{B\bullet} & 0 \\ A_T^T & -Q_{T\bullet} & 0 \\ 0 & 0 & A_B \\ 0 & -Q & Q_{\bullet B}\end{pmatrix} - \operatorname{rank}\begin{pmatrix} A_B^T & -Q_{B\bullet} & 0 \\ A_T^T & -Q_{T\bullet} & 0 \\ 0 & -Q & Q_{\bullet B}\end{pmatrix}.$$

Using row and column operations on the matrix in the last term together with Lemma 3.2 we obtain the dimension of H_b:

$$\begin{aligned}\dim(H_b) &= \operatorname{rank}\begin{pmatrix} A_B^T \\ A_T^T\end{pmatrix} + \operatorname{rank}\begin{pmatrix} A_B \\ Q_{BB}\end{pmatrix} - \operatorname{rank}\begin{pmatrix} A_B^T & Q_{BB} \\ A_T^T & Q_{TB}\end{pmatrix} \\ &= \operatorname{rank}(A_B),\end{aligned}$$

where the last equation follows from (3.6). Adding this to (3.3) yields the second result.

The third result does not follow from a decoupling principle, as in the linear case (where $H = H_b \times H_c$). Rather, it needs a development similar to the first two parts just obtained. Using Lemmas 3.1 and 3.2 yields the following equations

$$\begin{aligned}\dim(H) &= |N| + \operatorname{rank}\begin{pmatrix} A_B^T & -Q_{B\bullet} & 0 \\ A_T^T & -Q_{T\bullet} & 0 \\ 0 & 0 & A_B \\ 0 & -Q & Q_{\bullet B} \end{pmatrix} - \operatorname{rank}\begin{pmatrix} 0 & -Q & Q_{\bullet B} \end{pmatrix} \\ &= |N| + \operatorname{rank}\begin{pmatrix} A_B^T \\ A_T^T \end{pmatrix} + \operatorname{rank}\begin{pmatrix} A_B \\ Q_{BB} \end{pmatrix}.\end{aligned}$$

The sum of the last statement with (3.1) and (3.3), plus substituting $n - |T| = |B| + |N|$, implies the third result. □

Notice that the statements in Theorem 3.3 reduce to the corresponding statements in Theorem 2.1 when $T = \emptyset$ and $Q = 0$, which is the case for an LP. (This reduction occurs because we eliminated the u-variables.) In fact, the statements in the theorem imply each of the following.

$$\begin{array}{rcll} \dim(QP^*) + \dim(H_c) & \leq & n & \text{with equality if } \; T = \emptyset. \\ \dim(QD^*) + \dim(H_b) & \leq & m & \text{with equality if } \; T = \emptyset. \\ \dim(QP^* \times QD^*) + \dim(H) & \leq & m+n & \text{with equality if } \; T = \emptyset. \end{array}$$

The reduction of $\mathcal{QD}^*$ also enables us to have the following extension of Corollary 2.2. (In fact, u is unique if and only if Q is positive definite because it can be any solution to $Qu = Qx$ for any $x \in QP^*$.)

COROLLARY 3.4. *The following are equivalent.*

1. *The dual solution is unique.*
2. $\dim(H_c) = n + m - |T| - \operatorname{rank}(A_B) + \operatorname{rank}(A_B^T\ Q_{BB})$.
3. $\dim(H_b) = m + \operatorname{rank}(A_B) - \operatorname{rank}(A_B\ A_T)$.

The above cases reduce to the corresponding LP cases in Corollary 2.2, where $Q = 0$ and $T = \emptyset$, as does the following extension of Corollary 2.3.

COROLLARY 3.5. *The following are equivalent.*

1. *The primal solution is unique.*
2. $\dim(H_c) = n - |T| + \operatorname{rank}(A_B\ A_T) - \operatorname{rank}(A_B)$.
3. $\dim(H_b) = |B| - \operatorname{rank}(A_B^T\ Q_{BB}) + \operatorname{rank}(A_B)$.

Combining these, despite the absence of a decoupling principle, the dimensions are additive, so we also obtain the following extension of Corollary 2.4.

COROLLARY 3.6. *The following are equivalent.*

1. *The primal-dual solution is unique.*
2. $\dim(H_c) = n - |T|$ *and* $\dim(H_b) = m$.
3. $\dim(H) = m + n - |T|$.

Unlike the LP case, this shows that we lose $|T|$ degrees of freedom in varying the cost coefficients. For example, if $\delta c_j > 0$ for $j \in T$, the partition immediately changes since $s_j = \delta c_j$ is optimal for the perturbed quadratic program. This loss appears in the last extension, which follows.

COROLLARY 3.7. *There are at least* $m - |T| + \operatorname{rank}(A_B\ A_T) - \operatorname{rank}(A_B)$ *degrees of freedom to vary the rim data without a change in the optimal partition.*

This lower bound on $\dim(H_c)$ follows in the same way as in Corollary 2.5, and it is m when $T = \emptyset$. More generally, we see that the bound is at most m, which reflects the fact that we can lose some degrees of freedom by lacking strict complementarity.

4. Concluding comments. For LPs, the dimension of the cone of rim direction vectors for which the optimal partition does not change has an Eulerian property with the dimension of the optimality region: they sum to the number of variables and equations. This decouples into Eulerian properties for varying the primal and dual right-hand sides separately: cost coefficients change with lost degrees of freedom equal to the dimension of primal space; right-hand sides change with lost degrees of freedom equal to the dimension of dual space. The comparable equation for quadratic programs is not Eulerian in that the sum of dimensions depends on the partition—notably on the number of complementary coordinate pairs that are both zero.

REFERENCES

[1] A. Berkelaar, C. Roos, and T. Terlaky, *The optimal set and optimal partition approach to linear and quadratic programming*, in Advances in Sensitivity Analysis and Parametric Programming, T. Gal and H. Greenberg, eds., Kluwer Academic Publishers, Boston, MA, 1997, Chapter 6.

[2] H. Greenberg, *Mathematical Programming Glossary*, http://www-math.cudenver.edu/~hgreenbe/glossary/glossary.html, 1996.

[3] H. Greenberg, *Rim Sensitivity Analysis from an Interior Solution*, Technical report CCM 86, Center for Computational Mathematics, Mathematics Department, University of Colorado at Denver, Denver, CO, 1996.

[4] B. Jansen, *Interior Point Techniques in Optimization: Complexity, Sensitivity, and Algorithms*, Kluwer Academic Publishers, Boston, MA, 1997.

[5] O. Güler and Y. Ye, *Convergence behavior of interior-point algorithms*, Math. Programming, 60 (1993), pp. 215–228.

[6] C. Roos, T. Terlaky, and J.-P. Vial, *Theory and Algorithms for Linear Optimization: An Interior Point Approach*, John Wiley and Sons, Chichester, UK, 1997.

SIAM J. Optim.
Vol. 9, No. 1, pp. 217–235

AN ANALYTIC CENTER BASED COLUMN GENERATION ALGORITHM FOR CONVEX QUADRATIC FEASIBILITY PROBLEMS[*]

ZHI-QUAN LUO[†] AND JIE SUN[‡]

Abstract. We consider the analytic center based column generation algorithm for the problem of finding a feasible point in a set defined by a finite number of convex quadratic inequalities. At each iteration the algorithm computes an approximate analytic center of the set defined by the intersection of quadratic inequalities generated in the previous iterations. If this approximate analytic center is a solution, then the algorithm terminates; otherwise a quadratic inequality violated at the current center is selected and a new quadratic cut (defined by a convex quadratic inequality) is placed near the approximate center. As the number of cuts increases, the set defined by their intersection shrinks and the algorithm eventually finds a solution to the problem. Previously, similar analytic center based column generation algorithms were studied first for the linear feasibility problem and later for the general convex feasibility problem. Our method differs from these early methods in that we use "quadratic cuts" in the computation instead of linear cuts. Moreover, our method has a polynomial worst case complexity of $O(n \ln \frac{1}{\varepsilon})$ on the total number of cuts to be used, where n is the number of convex quadratic polynomial inequalities in the problem and ε is the radius of the largest ball contained in the feasible set. In contrast, the early column generation methods using linear cuts can only solve the convex quadratic feasibility problem in pseudopolynomial time.

Key words. convex quadratic feasibility problem, analytic center, potential reduction, column generation

AMS subject classifications. 90C25, 90C26, 90C60

PII. S1052623495294943

1. Introduction.

Consider the problem of finding a feasible point in a convex body Γ, where $\Gamma \subset \mathbb{R}^m$ is defined by the intersection of a finite number of convex quadratic inequalities; that is,

$$\Gamma = \left\{ y \in \mathbb{R}^m : c_j - \langle a_j, y \rangle - \frac{1}{2} \langle y, Q_j y \rangle \geq 0,\ j = 1, \ldots, n \right\}, \tag{1.1}$$

where, for each $j = 1, \ldots, n$, Q_j is a symmetric positive semidefinite matrix, a_j is an m-vector, c_j is a scalar, and $\langle \cdot, \cdot \rangle$ denotes the usual Euclidean inner product. The convex quadratic feasibility problem is quite general as it includes as special cases the linear programming problem, the linear feasibility problem, and the convex quadratic programming problem with quadratic constraints. The latter problem has previously been studied using conventional interior point methods (see [9, 11, 12]).

In this paper we consider the column generation algorithm for solving the above convex quadratic feasibility problem. At each iteration, the algorithm computes an approximate analytic center of the set defined by the intersection of quadratic inequalities generated in the previous iterations. If this approximate analytic center is

[*] Received by the editors November 20, 1995; accepted for publication (in revised form) February 9, 1998; published electronically December 2, 1998.
http://www.siam.org/journals/siopt/9-1/29494.html

[†] Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, L8S 4L7, Canada (luozq@mcmail.cis.mcmaster.ca). The research of this author was supported by the Natural Sciences and Engineering Research Council of Canada grant OPG0090391.

[‡] Department of Decision Sciences, National University of Singapore, Singapore 119260 (jsun@nus.edu.sg). The research of this author was supported by National University of Singapore grant RP930033.

a solution, then the algorithm terminates; otherwise a quadratic inequality violated at the current approximate center is selected and a new quadratic cut (defined by a convex quadratic inequality) is placed near the center. As the number of cuts increases, the set defined by their intersection shrinks and the algorithm eventually finds a solution to the problem. Previously, similar analytic center based column generation algorithms were studied first for the linear feasibility problem [16] and later for the general convex feasibility problem [1, 2, 3, 7, 8, 13]. Recently, these methods were further extended to allow multiple cuts and weighted analytic centers [10, 17]. Our method differs from these early methods in that we use "quadratic cuts" in the computation instead of linear cuts. Moreover, our method has a polynomial worst case complexity of $O(n \ln \frac{1}{\varepsilon})$ on the total number of cuts to be used, where n is the number of convex quadratic polynomial inequalities in the problem and ε is the radius of the largest ball contained in the feasible set. In contrast, the early methods using linear cuts can only solve the convex quadratic feasibility problem in pseudopolynomial time.

It is possible to formulate the convex quadratic feasibility problem in terms of a linear minimization problem over the positive semidefinite cone [14] and solve it in $O(\sqrt{n} \ln \frac{1}{\varepsilon})$ Newton iterations. While the column generation algorithm considered in this paper has a less favorable complexity bound, it nevertheless does provide several advantages. Specifically, the column generation algorithm permits a high degree of flexibility, since it does not require the full knowledge of all the constraints in Γ and allows the inequalities to be generated dynamically. In this sense, it is well suited for problems with a large number of constraints. Algorithms similar to the one considered here (e.g., the analytic center based cutting plane algorithm, the Dantzig–Wolfe decomposition method) have been very successful in solving large scale linear multicommodity flow problems and stochastic linear programs [4, 5].

We shall adopt the standard notation used in the interior point method literature. For example, for any generic vector x, x^T will denote the vector transpose, and the corresponding capital letter X will denote the diagonal matrix whose (i, i)th entry is given by the ith component of x. For any twice continuously differentiable function $f : \mathbb{R}^m \to \mathbb{R}$ we use ∇f and $\nabla^2 f$ to represent the gradient vector and the Hessian matrix of f, respectively. Also, we shall use the letter e to stand for the vector of ones (in an appropriate Euclidean space). For any two square matrices A, B we write $A \geq B$ to mean that $A - B$ is a positive semidefinite matrix.

2. Preliminaries. Let Ω be a bounded set in $\mathbb{R}^m$ defined by n $(> m)$ convex quadratic inequalities, i.e.,

$$\Omega = \left\{ y \in \mathbb{R}^m : c_j - \langle a_j, y \rangle - \frac{1}{2} \langle y, Q_j y \rangle \geq 0, \ j = 1, \ldots, n \right\}. \tag{2.1}$$

Suppose that Ω has a nonempty interior which we denote by $\operatorname{int} \Omega$. We define the potential function of Ω as

$$\phi(y) := - \sum_{j=1}^{n} \ln s_j, \tag{2.2}$$

where

$$s_j := c_j - \langle a_j, y \rangle - \frac{1}{2} \langle y, Q_j y \rangle$$

is the residual for the jth inequality. It can be seen that $\phi(\cdot)$ is a smooth strictly convex function defined over the interior of Ω (see [11] for a proof). The gradient and the Hessian of $\phi(\cdot)$ at $y \in \operatorname{int}\Omega$ are given by

$$g(y) := \nabla\phi(y) = \sum_{j=1}^{n} \frac{a_j + Q_j y}{s_j}$$

and

$$H(y) := \nabla^2\phi(y) = \sum_{j=1}^{n} \left[\frac{(a_j + Q_j y)(a_j + Q_j y)^T}{s_j^2} + \frac{Q_j}{s_j} \right].$$

The *potential* of Ω is defined as

$$P(\Omega) := \min_{y \in \operatorname{int}\Omega} \phi(y). \tag{2.3}$$

Since Ω is bounded, $\phi(\cdot)$ is strictly convex and approaches infinity near the boundary of Ω; the minimum value $P(\Omega)$ exists and is attained at a unique point. This minimum point is defined (see [15]) as the *analytic center* of Ω. It should be noted that the analytic center of Ω depends not only on the set Ω but also on its algebraic representation. Different forms of representation of the same geometric set Ω may give rise to different analytic centers. In this regard, analytic centers are not "geometric" quantities.

Let y^a denote the analytic center of Ω. Then

$$g(y^a) = \nabla\phi(y^a) = \sum_{j=1}^{n} \frac{a_j + Q_j y^a}{s_j^a} = 0,$$

where $s_j^a := c_j - \langle a_j, y^a \rangle - \frac{1}{2}\langle y^a, Q_j y^a \rangle$, holds. In this paper we often need to measure the "proximity" between the analytic center y^a and an arbitrary vector y in the interior of Ω. One commonly used proximity measure is the norm of the scaled gradient vector:

$$\delta(y;\Omega) := \left\langle g(y), H^{-1}(y) g(y) \right\rangle^{1/2}. \tag{2.4}$$

Clearly, $\delta(y^a;\Omega) = 0$. Another commonly used proximity measure is the "gap" $\phi(y) - \phi(y^a)$. We summarize some well-known properties of ϕ and of these proximity measures below. These results will be used later in the analysis of the column generation algorithm.

Lemma 2.1. *If $x,\ y \in \operatorname{int}\Omega$ are such that $\langle y - x, H(x)(y - x)\rangle < 1$, then*

$$\left| \phi(y) - \phi(x) - \langle \nabla\phi(x), y - x\rangle - \frac{1}{2}\langle y - x, H(x)(y - x)\rangle \right| \\ \leq \frac{\langle y - x, H(x)(y - x)\rangle^{3/2}}{3\left[1 - \langle y - x, H(x)(y - x)\rangle^{1/2}\right]}.$$

Lemma 2.2. *Let $y \in \operatorname{int}\Omega$ be a vector such that $\delta(y;\Omega) < 1/8$. Then*

$$\phi(y) - \phi(y^a) \leq 4\delta(y;\Omega)^2.$$

Lemma 2.1 first appeared in [11] and was later strengthened in [6]. Lemma 2.2 is taken from [6] (see Lemma 5.5 therein). Next we shall use Lemma 2.1 to establish a relation that is in some sense a reverse of Lemma 2.2.

LEMMA 2.3. *For any $y \in \Omega$ with $\phi(y) - \phi(y^a) \leq 0.04$,*

$$\delta(y; \Omega)^2 \leq 0.14$$

holds.

Proof. Suppose the contrary so that $\delta(y; \Omega)^2 > 0.14$. Consider the vector $y_\tau = y^a + \tau(y - y^a)$, where $\tau \in (0, 1)$ is the smallest positive number such that

$$\delta(y_\tau; \Omega)^2 = \left\langle g(y_\tau), H^{-1}(y_\tau) g(y_\tau) \right\rangle = 0.14.$$

By the convexity of the potential function ϕ,

$$\phi(y_\tau) \leq \tau \phi(y) + (1 - \tau)\phi(y^a) \leq \phi(y)$$

holds. Consider the Newton procedure

$$y_\tau^+ = y_\tau - H^{-1}(y_\tau) g(y_\tau).$$

We have

$$\left\langle y_\tau^+ - y_\tau, H(y_\tau)(y_\tau^+ - y_\tau) \right\rangle = \left\langle g(y_\tau), H^{-1}(y_\tau) g(y_\tau) \right\rangle = 0.14 < 1.$$

Then it follows from Lemma 2.1 that

$$\begin{aligned}
\phi(y_\tau) - \phi(y_\tau^+) &\geq \left\langle g(y_\tau), H^{-1}(y_\tau) g(y_\tau) \right\rangle - \frac{1}{2} \left\langle g(y_\tau), H^{-1}(y_\tau) g(y_\tau) \right\rangle \\
&\quad - \frac{\left\langle g(y_\tau), H^{-1}(y_\tau) g(y_\tau) \right\rangle^{3/2}}{3 \left[1 - \langle g(y_\tau), H^{-1}(y_\tau) g(y_\tau) \rangle^{1/2} \right]} \\
&\geq 0.295 \left\langle g(y_\tau), H^{-1}(y_\tau) g(y_\tau) \right\rangle \\
&> 0.04.
\end{aligned}$$

This shows that

$$\phi(y) - \phi(y^a) \geq \phi(y_\tau) - \phi(y^a) \geq \phi(y_\tau) - \phi(y_\tau^+) > 0.04,$$

which contradicts the assumption on y. □

The next lemma from to Nesterov [13].

LEMMA 2.4. *Suppose $\gamma \in [0, (\sqrt{2} - 1)^2]$ and $y \in \operatorname{int} \Omega$ are such that*

$$\delta(y; \Omega) \leq \gamma.$$

Then

$$\langle y - y^a, H(y)(y - y^a) \rangle^{1/2} \leq \frac{\gamma}{2 - \sqrt{2}}.$$

Lemma 2.4 was established in [13] for the class of so-called *strictly self-concordant* functions (see [14] for the definition); this class of functions is very broad and certainly includes the function $\phi(\cdot)$ considered in this paper.

Next we present an error bound to be used in section 5 to upper bound the potential function.

LEMMA 2.5. *Let $Q \in \mathbb{R}^{m\times m}$ be a symmetric positive semidefinite matrix and let $a \in \mathbb{R}^m$ and $c \in \mathbb{R}$. Assume*

$$\max\{\|a\|, \|Q\|\} = 1.$$

Denote

$$\mathcal{S} := \left\{ y \in \mathbb{R}^m \ : \ c - \langle a, y\rangle - \frac{1}{2}\langle y, Qy\rangle \geq 0 \right\}$$

and suppose that $\mathcal{S}$ contains an ε-ball centered at y^*. Then

$$c - \langle a, y^*\rangle - \frac{1}{2}\langle y^*, Qy^*\rangle \geq \min\left\{\varepsilon, \frac{\varepsilon^2}{2}\right\}.$$

Proof. By a translation and an orthonormal transformation if necessary, we may assume, without loss of generality, that $y^* = 0$ and Q is diagonal. Denote $Q = \text{diag}(\lambda_1, \lambda_2 \ldots, \lambda_m)$ and $a = (a_1, \ldots, a_m)^T$. Since orthonormal transformation preserves the Euclidean norm, it follows that $\lambda_i \leq 1$ and $\|a\|^2 = \sum_i a_i^2 \leq 1$. We consider two cases. In the first case, $\lambda_i = 1$ for some i. Let the vector $y(t)$ be defined by

$$y_j(t) = \begin{cases} 0 & \text{if } j \neq i \\ t & \text{if } j = i. \end{cases}$$

Notice that

$$c - \langle a, y(t)\rangle - \frac{1}{2}\langle y(t), Qy(t)\rangle = c - a_i t - \frac{1}{2}t^2.$$

As a function in t, the above quadratic polynomial has two roots, say t_1 and t_2. It is easily seen that $t_1 t_2 = 2c$. By assumption, $y(t_1)$ and $y(t_2)$ are at least a distance of ε away from the origin. This implies $|t_1| \geq \varepsilon$, $|t_2| \geq \varepsilon$. Therefore $c \geq \varepsilon^2/2$.

In the second case, $\lambda_i < 1$ for all i. Then it follows that $\|a\| = 1$. Consider the vector $\bar{y} = \varepsilon a$ which is a distance of ε away from the origin. By assumption we have

$$c - \langle a, \bar{y}\rangle - \frac{1}{2}\langle \bar{y}, Q\bar{y}\rangle \geq 0.$$

Since $Q \geq 0$, this implies

$$c \geq \langle a, \bar{y}\rangle = \varepsilon\|a\|^2 = \varepsilon.$$

Now we can combine the estimates in both cases to obtain the desired bound. ☐

We close this section by stating a useful result from linear algebra.

LEMMA 2.6. *Let ρ be a positive constant. Then*

$$\left\langle a, (\rho I + aa^T)^{-1} a\right\rangle = \frac{\|a\|^2}{\rho + \|a\|^2}, \qquad \textit{for all } a \in \mathbb{R}^n.$$

Proof. By the Sherman–Morrison formula

$$\left[I + \frac{aa^T}{\rho}\right]^{-1} = I - \frac{aa^T}{\rho + \|a\|^2},$$

we have

$$\begin{aligned}\left\langle a,(\rho I+aa^T)^{-1}a\right\rangle &= \frac{1}{\rho}\left\langle a,\left(I-aa^T/(\rho+\|a\|^2)\right)a\right\rangle\\ &= \frac{1}{\rho}\left[\|a\|^2-\|a\|^4/(\rho+\|a\|^2)\right]\\ &= \frac{\|a\|^2}{\rho+\|a\|^2}. \qquad \square\end{aligned}$$

3. Potential increase. In this section we analyze how the potential $P(\Omega)$ (cf. (2.3)) changes as the set Ω (cf. (2.1)) is modified in some controlled ways. We consider two ways in which the set Ω is modified; the first is by translating an existing quadratic inequality of Ω and the second is by introducing a new convex quadratic inequality to Ω.

For ease of reference, we write below the representation of Ω again:

$$\Omega=\left\{y\in\mathbb{R}^m : c_j-\langle a_j,y\rangle-\frac{1}{2}\langle y,Q_jy\rangle\geq 0,\ j=1,\ldots,n\right\}. \tag{3.1}$$

Let y^a denote its analytic center. Suppose that y^b is an approximate center in the sense that $\delta(y^b;\Omega)\leq 1-\beta\leq(\sqrt{2}-1)^2$. Consider the following set Ω_β obtained by translating the last inequality of Ω:

$$\begin{aligned}\Omega_\beta=\Bigg\{y\ :\quad & c_j-\langle a_j,y\rangle-\frac{1}{2}\langle y,Q_jy\rangle\geq 0,\ j=1,\ldots,n-1,\\ & \beta s_n^b+\left\langle a_n,y^b\right\rangle+\frac{1}{2}\left\langle y^b,Q_ny^b\right\rangle-\langle a_n,y\rangle-\frac{1}{2}\langle y,Q_ny\rangle\geq 0\Bigg\},\end{aligned} \tag{3.2}$$

where $\beta\in[0,1]$ is some constant and $s_n^b:=c_n-\left\langle a_n,y^b\right\rangle-\frac{1}{2}\left\langle y^b,Q_ny^b\right\rangle$. Let $\bar{y}^a$ denote the analytic center of Ω_β. Notice that if $\beta=0$, the last inequality of Ω_β is placed through the approximate analytic center y^b of Ω. With $\beta>0$, the translation of the last inequality of Ω does not go all the way to y^b; the approximate analytic center y^b is kept inside Ω_β. The following lemma estimates the increase of the potential $P(\Omega)$ as the last inequality of Ω is translated.

LEMMA 3.1. *Let Ω and Ω_β be given as above and let $\beta\in[1-(\sqrt{2}-1)^2,1]$. Then*

$$P(\Omega_\beta)\geq P(\Omega)+\frac{1-\beta}{1+3(1-\beta)}.$$

Proof. Since $\delta(y^b;\Omega)\leq 1-\beta\leq(\sqrt{2}-1)^2$, by Nesterov's lemma (Lemma 2.4) we have

$$\left\langle y^a-y^b,H(y^b)(y^a-y^b)\right\rangle^{1/2}\leq\frac{1-\beta}{2-\sqrt{2}}.$$

Note that

$$\frac{(Q_ny^b+a_n)(Q_ny^b+a_n)^T}{(c_n-\langle a_n,y^b\rangle-\frac{1}{2}\langle y^b,Q_ny^b\rangle)^2}+\frac{Q_n}{c_n-\langle a_n,y^b\rangle-\frac{1}{2}\langle y^b,Q_ny^b\rangle}\leq H(y^b).$$

Multiplying left and right, respectively, by y^a-y^b and $(y^a-y^b)^T$ and using the preceding inequality, we obtain

$$\left|\frac{\left\langle Q_ny^b+a_n,y^b-y^a\right\rangle}{c_n-\langle a_n,y^b\rangle-\frac{1}{2}\langle y^b,Q_ny^b\rangle}\right|\leq\frac{1-\beta}{2-\sqrt{2}}$$

and

$$\frac{\left\langle y^b - y^a, Q_n(y^b - y^a)\right\rangle}{c_n - \langle a_n, y^b\rangle - \frac{1}{2}\left\langle y^b, Q_n y^b\right\rangle} \le \left(\frac{1-\beta}{2-\sqrt{2}}\right)^2 \le \frac{1-\beta}{2-\sqrt{2}}.$$

Therefore

$$\begin{aligned}
&\left|\frac{c_n - \langle a_n, y^a\rangle - \frac{1}{2}\left\langle y^a, Q_n y^a\right\rangle}{c_n - \langle a_n, y^b\rangle - \frac{1}{2}\langle y^b, Q_n y^b\rangle} - 1\right| \\
&\quad = \left|\frac{\langle Q_n y^b + a_n, y^a - y^b\rangle + \frac{1}{2}\langle y^a - y^b, Q_n(y^a - y^b)\rangle}{c_n - \langle a_n, y^b\rangle - \frac{1}{2}\left\langle y^b, Q_n y^b\right\rangle}\right| \\
&\quad \le \left|\frac{\left\langle Q_n y^b + a_n, y^b - y^a\right\rangle}{c_n - \langle a_n, y^b\rangle - \frac{1}{2}\left\langle y^b, Q_n y^b\right\rangle}\right| \\
&\qquad + \frac{1}{2}\frac{\left\langle y^a - y^b, Q_n(y^b - y^a)\right\rangle}{c_n - \langle a_n, y^b\rangle - \frac{1}{2}\left\langle y^b, Q_n y^b\right\rangle} \\
&\quad \le \frac{3}{2}\frac{1-\beta}{2-\sqrt{2}} \le 3(1-\beta).
\end{aligned}$$

From this we get

$$\frac{c_n - \left\langle a_n, y^b\right\rangle - \frac{1}{2}\left\langle y^b, Q_n y^b\right\rangle}{c_n - \langle a_n, y^a\rangle - \frac{1}{2}\left\langle y^a, Q_n y^a\right\rangle} \ge \frac{1}{1 + 3(1-\beta)}.$$

Since y^a is the analytic center of Ω,

$$0 = \nabla\phi(y^a) = \sum_{j=1}^{n} \frac{a_j + Q_j y^a}{s_j^a},$$

where

$$s_j^a := c_j - \langle a_j, y^a\rangle - \frac{1}{2}\left\langle y^a, Q_j y^a\right\rangle,$$

holds. Let $x_j^a := (s_j^a)^{-1}$, for $j = 1, \ldots, n$. Also, we denote the analytic center of Ω_β by $\bar{y}^a$ and denote the "slacks" by

$$\begin{aligned}
\bar{s}_j^a &:= c_j - \langle a_j, \bar{y}^a\rangle - \tfrac{1}{2}\left\langle \bar{y}^a, Q_j \bar{y}^a\right\rangle, \quad j = 1, \ldots, n-1, \\
\bar{s}_n^a &:= \beta s_n^b + \left\langle a_n, y^b\right\rangle + \tfrac{1}{2}\left\langle y^b, Q_n y^b\right\rangle - \langle a_n, \bar{y}^a\rangle - \tfrac{1}{2}\left\langle \bar{y}^a, Q_n \bar{y}^a\right\rangle.
\end{aligned}$$

Consider the following:

$$\begin{aligned}
\langle e, X^a \bar{s}^a\rangle &= \sum_{j=1}^{n} x_j^a \bar{s}_j^a \\
&= \sum_{j=1}^{n-1} x_j^a \left(c_j - \langle a_j, \bar{y}^a\rangle - \frac{1}{2}\left\langle \bar{y}^a, Q_j \bar{y}^a\right\rangle\right) \\
&\quad + x_n^a \left(\beta s_n^b + \left\langle a_n, y^b\right\rangle + \frac{1}{2}\left\langle y^b, Q_n y^b\right\rangle - \langle a_n, \bar{y}^a\rangle - \frac{1}{2}\left\langle \bar{y}^a, Q_n \bar{y}^a\right\rangle\right)
\end{aligned}$$

$$\begin{aligned}
&= \sum_{j=1}^{n-1} x_j^a \left(c_j - \langle a_j, \bar{y}^a \rangle - \frac{1}{2} \langle \bar{y}^a, Q_j \bar{y}^a \rangle \right) \\
&\quad -(1-\beta) x_n^a s_n^b + x_n^a \left(c_n - \langle a_n, \bar{y}^a \rangle - \frac{1}{2} \langle \bar{y}^a, Q_n \bar{y}^a \rangle \right) \\
&\le \sum_{j=1}^{n} x_j^a \left(c_j - \langle a_j, \bar{y}^a \rangle - \frac{1}{2} \langle \bar{y}^a, Q_j \bar{y}^a \rangle \right) \\
&\quad - \frac{1-\beta}{1+3(1-\beta)} x_n^a \left(c_n - \langle a_n, y^a \rangle - \frac{1}{2} \langle y^a, Q_n y^a \rangle \right) \\
&= \sum_{j=1}^{n} x_j^a \left(c_j - \langle a_j, \bar{y}^a \rangle - \frac{1}{2} \langle \bar{y}^a, Q_j \bar{y}^a \rangle \right) - \frac{1-\beta}{1+3(1-\beta)},
\end{aligned}$$

where the second step follows from the definition of s_n^a and the last step is due to $x_n^a s_n^a = 1$. Next we use the Taylor expansion of $c_j - \langle a_j, \bar{y}^a \rangle - \frac{1}{2} \langle \bar{y}^a, Q_j \bar{y}^a \rangle$ at y^a to obtain

$$\begin{aligned}
\langle e, X^a \bar{s}^a \rangle &= \sum_{j=1}^{n} x_j^a \left(c_j - \langle a_j, y^a \rangle - \frac{1}{2} \langle y^a, Q_j y^a \rangle \right) \\
&\quad - \sum_{j=1}^{n} x_j^a \langle a_j + Q_j y^a, \bar{y}^a - y^a \rangle - \sum_{j=1}^{n} x_j^a \cdot \frac{1}{2} \langle \bar{y}^a - y^a, Q_j (\bar{y}^a - y^a) \rangle \\
&\quad - \frac{1-\beta}{1+3(1-\beta)} \\
&= n - \sum_{j=1}^{n} x_j^a \cdot \frac{1}{2} \langle \bar{y}^a - y^a, Q_j (\bar{y}^a - y^a) \rangle - \frac{1-\beta}{1+3(1-\beta)} \\
&\le n - \frac{1-\beta}{1+3(1-\beta)}, \qquad (3.3)
\end{aligned}$$

where the second equality follows from

$$x_j^a \left(c_j - \langle a_j, y^a \rangle - \frac{1}{2} \langle y^a, Q_j y^a \rangle \right) = x_j^a s_j^a = 1, \quad j = 1, \ldots, n$$

and

$$0 = \nabla \phi(y^a) = \sum_{j=1}^{n} \frac{a_j + Q_j y^a}{s_j^a} = \sum_{j=1}^{n} x_j^a (a_j + Q_j y^a);$$

the last step is due to the fact that each Q_j is positive semidefinite. Now we can use (3.3) to finish the proof as follows:

$$\begin{aligned}
\frac{\exp\, P(\Omega)}{\exp\, P(\Omega_\beta^+)} &= \prod_{j=1}^{n} \frac{\bar{s}_j^a}{s_j^a} = \prod_{j=1}^{n} x_j^a \bar{s}_j^a \\
&\le \left(\frac{1}{n} \sum_{j=1}^{n} x_j^a \bar{s}_j^a \right)^n \le \left[\frac{1}{n} \left(n - \frac{1-\beta}{1+3(1-\beta)} \right) \right]^n \\
&\le \exp\, \frac{\beta - 1}{1+3(1-\beta)},
\end{aligned}$$

where in the fourth step we have used (3.3). Now taking the logarithm on both sides yields the desired inequality. □

Next we consider the case where a new inequality is added to Ω. Let

$$\begin{aligned}\Omega_\beta^+ := \quad & \Big\{ y \ : c_j - \langle a_j, y\rangle - \frac{1}{2}\langle y, Q_j y\rangle \geq 0, \quad j = 1, \ldots, n, \\ & \beta r + \langle a_{n+1}, y^b\rangle + \frac{1}{2}\langle y^b, Q_{n+1} y^b\rangle - \langle a_{n+1}, y\rangle - \frac{1}{2}\langle y, Q_{n+1} y\rangle \geq 0 \Big\},\end{aligned} \tag{3.4}$$

where y^b is an approximate center of Ω satisfying $\delta(y^b;\Omega) \leq 1/\beta$, $\beta \geq 2(\sqrt{2}+1)^2$,

$$r := \langle h, H^{-1}(y^b) h\rangle^{1/2}, \qquad H(y^b) := \nabla^2 \phi(y^b), \qquad h := a_{n+1} + Q_{n+1} y^b,$$

and ϕ is given by (2.2).

Lemma 3.2. *Let Ω, Ω_β^+, and r be defined as above. Suppose $\beta \geq 2(\sqrt{2}+1)^2$. Then*

$$P(\Omega_\beta^+) \geq P(\Omega) - \ln r - \ln\left[\beta + \frac{2}{\beta(2-\sqrt{2})}\right].$$

Proof. As in Lemma 3.1, we use y^a, $\bar{y}^a$ to denote the analytic centers of Ω and Ω_β^+, respectively. Also, we let $\phi(\cdot)$ and $\phi_+(\cdot)$ denote the potential functions of Ω and Ω_β^+, respectively. The gradient and the Hessian of $\phi(\cdot)$ (respectively, $\phi_+(\cdot)$) are denoted by $g(\cdot)$, $H(\cdot)$ (respectively, $g_+(\cdot)$, $H_+(\cdot)$). We start by first estimating $\langle g_+(y^b), H_+(y^b)^{-1} g_+(y^b)\rangle^{1/2}$. Since $Q_{n+1} \geq 0$, it follows that

$$H_+(y^b) = H(y^b) + \frac{1}{\beta^2 r^2}(a_{n+1} + Q_{n+1} y^b)(a_{n+1} + Q_{n+1} y^b)^T + \frac{Q_{n+1}}{\beta r} \geq H(y^b).$$

Using this and the fact that

$$\begin{aligned} g_+(y^b) &= g(y^b) + \frac{a_{n+1} + Q_{n+1} y^b}{\beta r + \langle a_{n+1}, y^b\rangle + \frac{1}{2}\langle y^b, Q_{n+1} y^b\rangle - \langle a_{n+1}, y^b\rangle - \frac{1}{2}\langle y^b, Q_{n+1} y^b\rangle} \\ &= g(y^b) + \frac{h}{\beta r},\end{aligned}$$

we obtain

$$\begin{aligned}\langle g_+(y^b), H_+(y^b)^{-1} g_+(y^b)\rangle^{1/2} &\leq \left\langle g(y^b) + \frac{h}{\beta r}, H^{-1}(y^b)\left(g(y^b) + \frac{h}{\beta r}\right)\right\rangle^{1/2} \\ &\leq \langle g(y^b), H(y^b)^{-1} g(y^b)\rangle^{1/2} + \left\langle \frac{h}{\beta r}, H^{-1}(y^b)\frac{h}{\beta r}\right\rangle^{1/2} \\ &\leq \frac{1}{\beta} + \frac{1}{\beta} \leq (\sqrt{2}-1)^2,\end{aligned}$$

where the second inequality is due to the triangle inequality and we used the definitions of $\delta(y^b;\Omega)$ and r in the third inequality. The last step follows from the fact that $\beta \geq 2(\sqrt{2}+1)^2$. By the above inequality, we can invoke Nesterov's lemma (Lemma 2.4) to conclude that

$$\langle y^b - \bar{y}^a, H_+(y^b)(y^b - \bar{y}^a)\rangle \leq \frac{2}{\beta(2-\sqrt{2})}. \tag{3.5}$$

Next we estimate $\bar{s}^a_{n+1}$:

$$\begin{aligned}\bar{s}^a_{n+1} &= \beta r + \left\langle a_{n+1}, y^b \right\rangle + \frac{1}{2}\left\langle y^b, Q_{n+1} y^b \right\rangle - \langle a_{n+1}, \bar{y}^a \rangle - \frac{1}{2}\langle \bar{y}^a, Q_{n+1}\bar{y}^a \rangle \\ &= \left\langle a_{n+1} + Q_{n+1} y^b, y^b - \bar{y}^a \right\rangle - \frac{1}{2}\langle y^a - \bar{y}^a, Q_{n+1}(y^a - \bar{y}^a)\rangle + \beta r \\ &\le \left\langle a_{n+1} + Q_{n+1} y^b, y^b - \bar{y}^a \right\rangle + \beta r \\ &= \left\langle h, y^b - \bar{y}^a \right\rangle + \beta r \\ &= \left\langle H^{-1/2}(y^b) h, H^{1/2}(y^b)(y^b - \bar{y}^a) \right\rangle + \beta r \\ &\le \left\langle h, H^{-1}(y^b) h \right\rangle^{1/2} \cdot \left\langle y^a - \bar{y}^a, H(y^b)(y^a - \bar{y}^a) \right\rangle^{1/2} + \beta r,\end{aligned}$$

where the last step is due to the Cauchy–Schwarz inequality. By the definition of r and the estimate (3.5), we further obtain

$$\begin{aligned}\bar{s}^a_{n+1} &\le r \left\langle y^b - \bar{y}^a, H_+(y^b)(y^b - \bar{y}^a) \right\rangle^{1/2} + \beta r \\ &\le \left[\frac{2}{\beta(2-\sqrt{2})} + \beta \right] r.\end{aligned} \tag{3.6}$$

In addition, we can show, by an argument similar to the one used for establishing (3.3), that

$$\sum_{j=1}^n x^a_j \bar{s}^a_j \le n.$$

Using this and the inequality (3.6), we have

$$\begin{aligned}\frac{\exp\ P(\Omega)}{\exp\ P(\Omega^+_\beta)} &= \bar{s}^a_{n+1} \prod_{j=1}^n \frac{\bar{s}^a_j}{s^a_j} \le \bar{s}^a_{n+1} \prod_{j=1}^n x^a_j \bar{s}^a_j \\ &\le \bar{s}^a_{n+1} \left(\frac{1}{n} \sum_{j=1}^n x^a_j \bar{s}^a_j \right)^n \le \bar{s}^a_{n+1} \left(\frac{1}{n} \cdot n \right)^n \\ &= r \left[\frac{2}{\beta(2-\sqrt{2})} + \beta \right],\end{aligned}$$

where in the second step we have used (3.6). Now taking the logarithm on both sides yields the desired inequality. □

4. The column generation algorithm. The column generation algorithm for finding a feasible point in Γ can be informally described as follows. Essentially, the algorithm iteratively generates a sequence of sets

$$\Omega^0 \supset \Omega^1 \supset \cdots \supset \Omega^k \supset \cdots \supset \Gamma,$$

each of which is defined by some convex quadratic inequalities. At each iteration k the algorithm finds an approximate analytic center y^k of Ω^k. If y^k is in Γ, the algorithm terminates; otherwise a new set $\Omega^{k+1} \subset \Omega^k$ is generated by either translating an existing inequality of Ω^k or adding a new inequality to Ω^k. Then the algorithm attempts to find a new approximate center y^{k+1} of Ω^{k+1} and the iteration continues.

For convenience we assume that the vectors a_j and the matrices Q_j are normalized so that

$$\max\{\|a_j\|, \|Q_j\|\} = 1, \quad \text{for all } j = 1, \ldots, n. \tag{4.1}$$

We also assume that the solution set Γ is contained in the Euclidean unit ball $\mathbb{B}(0,1) := \{y \in \mathbb{R}^m : \|y\|^2 \le 1\}$. The analytic center based column generation algorithm is as follows.

The column generation algorithm.

Step 0. Ω^0 is defined by a total of $\rho \ge 1$ quadratic inequalities of the form $\|y\|^2 \le 1$. In other words, the inequality $\|y\|^2 \le 1$ is repeated ρ times in the definition of Ω^0. (This is necessary to ensure that the subsequent technical analysis can go through, and it also provides a convenient analytical center for Ω^0.) Formally, let

$$\Omega^0 = \{y : 1 - \|y\|^2 \ge 0,\ 1 - \|y\|^2 \ge 0, \ldots,\ 1 - \|y\|^2 \ge 0\}$$

and let

$$y^0 = 0 \in \mathbb{R}^m \quad \text{and} \quad \delta(y^0; \Omega^0) = 0.$$

Clearly, we have

$$s^0 = e \in \mathbb{R}^\rho.$$

Also, it can be seen that $\phi^0(y) = -\rho \ln(1 - \|y\|^2)$. Thus, we have

$$H^0(y) = \nabla^2 \phi^0(y) = \frac{2\rho}{1 - \|y\|^2} I + \frac{4\rho y y^T}{(1 - \|y\|^2)^2} \ge 2\rho I \quad \text{for all } y \in \Omega^0. \tag{4.2}$$

Step 1. Let

$$\Omega^k = \left\{ y \in \mathbb{R}^m : c_j^k - \left\langle a_j^k, y \right\rangle - \frac{1}{2} \left\langle y, Q_j^k y \right\rangle \ge 0,\ j = 1, \ldots, n_k \right\}$$

and let y^k be an approximate analytic center of Ω^k such that $\delta(y^k; \Omega^k) \le 1/20$. Check whether or not y^k satisfies all the quadratic inequalities in Γ. If yes, stop; otherwise choose an index i such that $c_i - \left\langle a_i, y^k \right\rangle - \frac{1}{2} \left\langle y^k, Q_i y^k \right\rangle < 0$. There are two cases depending on whether or not this ith inequality has been previously considered in Ω^k.

Case 1. The ith inequality of Γ has already been violated in the previous iterations. In this case there is a $0 < \ell \le n_k$, such that $Q_\ell^k = Q_i$, $a_\ell^k = a_i$. We translate the ℓth inequality of Ω^k to obtain

$$\begin{aligned} \Omega^{k+1} := \Big\{ y \ : \ & c_j^k - \left\langle a_j^k, y \right\rangle - \frac{1}{2} \left\langle y, Q_j^k y \right\rangle \ge 0, \quad 0 \le j \le n_k, \quad j \ne \ell \\ & \beta s_\ell^k + \left\langle a_\ell, y^k \right\rangle + \frac{1}{2} \left\langle y^k, Q_\ell y^k \right\rangle - \left\langle a_\ell, y \right\rangle - \frac{1}{2} \left\langle y, Q_\ell y \right\rangle \ge 0 \Big\}, \end{aligned} \tag{4.3}$$

where $\beta = 19/20$ and $s_\ell^k := c_\ell - \left\langle a_\ell, y^k \right\rangle - \frac{1}{2} \left\langle y^k, Q_\ell y^k \right\rangle$. Set

$$Q_j^{k+1} = \begin{cases} Q_j^k & \text{if } j \ne \ell \\ Q_i & \text{if } j = \ell \end{cases}, \qquad a_j^{k+1} = \begin{cases} a_j^k & \text{if } j \ne \ell \\ a_i & \text{if } j = \ell \end{cases}$$

and

$$c_j^{k+1} = \begin{cases} c_j^k & \text{if } j \neq \ell \\ \beta s_\ell^k + \langle a_\ell, y^k \rangle + \frac{1}{2} \langle y^k, Q_\ell y^k \rangle & \text{if } j = \ell \end{cases}, \qquad n_{k+1} = n_k.$$

Case 2. The ith inequality of Γ has never been violated in the previous iterations. In this case we add a new inequality to Ω^k to obtain

$$\begin{aligned} \Omega^{k+1} := \quad \Big\{ y \ :& c_j^k - \langle a_j^k, y \rangle - \frac{1}{2} \langle y, Q_j^k y \rangle \geq 0, \qquad 0 \leq j \leq n_k, \\ & \beta r^k + \langle a_i, y^k \rangle + \frac{1}{2} \langle y^k, Q_i y^k \rangle - \langle a_i, y \rangle - \frac{1}{2} \langle y, Q_i y \rangle \geq 0 \Big\}, \end{aligned} \tag{4.4}$$

where $\beta = 20$,

$$r^k := \langle h, H^{-1}(y^k) h \rangle^{1/2}, \quad H(y^k) := \nabla^2 \phi^k(y^k), \quad h := a_i + Q_i y^k,$$

and $\phi^k(\cdot)$ is the potential function for the set Ω^k. Set

$$Q_j^{k+1} = \begin{cases} Q_j^k & \text{if } 0 \leq j \leq n_k \\ Q_i & \text{if } j = n_k + 1 \end{cases}, \qquad a_j^{k+1} = \begin{cases} a_j^k & \text{if } 0 \leq j \leq n_k \\ a_i & \text{if } j = n_k + 1 \end{cases}$$

and

$$c_j^{k+1} = \begin{cases} c_j^k & \text{if } 0 \leq j \leq n_k \\ \beta r^k + \langle a_i, y^k \rangle + \frac{1}{2} \langle y^k, Q_i y^k \rangle & \text{if } j = n_k + 1 \end{cases}, \quad n_{k+1} = n_k + 1.$$

Step 2. Let $\phi^{k+1}(\cdot)$ be the potential function of Ω^{k+1}. Take the following Newton iterations

$$y^{new} := y - (\nabla^2 \phi^{k+1}(y))^{-1} \nabla \phi^{k+1}(y), \tag{4.5}$$

starting from y^k, until a new approximate analytic center y^{k+1} of Ω^{k+1} has been obtained with $\delta(y^{k+1}; \Omega^{k+1}) \leq 1/20$. Set $k := k + 1$ and return to Step 1.

Notice that in Step 2, we did not specify how many Newton iterations must be performed. It will be shown in section 6 that only a constant number (no more than 55) of Newton iterations are needed to generate the next iterate y^{k+1}.

We close this section by making some easy observations about the column generation algorithm. First of all, the total number of inequalities in Ω^k is at most $\rho + n$ for all k, where n is the total number of inequalities in Γ. Second, since $\|y^k\| \leq 1$, $\|a_i\| \leq 1$, and $\|Q_i\| \leq 1$ (the normalization assumption), it follows that $\|h\| := \|a_i + Q_i y^k\| \leq 2$. Thus, we have

$$\begin{aligned} r^k &= \langle h, \nabla^2 \phi^k (y^k)^{-1} h \rangle^{1/2} \\ &\leq \langle h, H^0(y^k)^{-1} h \rangle^{1/2} \\ &\leq \frac{1}{\sqrt{2\rho}} \|h\| \\ &\leq \sqrt{\frac{2}{\rho}}, \quad \text{for all } k, \end{aligned} \tag{4.6}$$

where the second and third inequalities follow from the fact that (see (4.2))

$$\nabla^2 \phi^k(y^k) \geq H^0(y^k) \geq 2\rho I. \tag{4.7}$$

Third, obviously we have

$$\Omega^0 \supset \Omega^1 \supset \cdots \supset \Omega^k \supset \cdots \supset \Gamma.$$

5. Convergence analysis. In this section we shall analyze the convergence of the column generation algorithm. The basic idea is to show that the potential $P(\Omega^k)$ of the set Ω^k is increased by a constant amount each time when either an existing inequality is translated or a new inequality is added to Ω^k. This, coupled with the fact that the potential $P(\Omega^k)$ is bounded from above, makes it possible to estimate the total number of iterations required to find a feasible point of Γ.

We start by upper bounding the potential $P(\Omega^k)$ of the set Ω^k.

LEMMA 5.1. *Assume that* Γ *is given by*

$$\Gamma = \left\{ y \in \mathbb{R}^m : c_j - \langle a_j, y \rangle - \frac{1}{2} \langle y, Q_j y \rangle \geq 0, \; j = 1, \ldots, n \right\},$$

with a_j, Q_j, $j = 1, \ldots, n$ *normalized as* (4.1). *Suppose that* Γ *contains an* ε*-ball* ($\varepsilon \leq 1$). *Then we have*

$$P(\Omega^k) \leq 2(n+\rho) \ln \frac{1}{\varepsilon} + (n+\rho) \ln 2, \tag{5.1}$$

for all $k \geq 0$.

Proof. Suppose the ε-ball is centered at y^*. Since $\Gamma \subseteq \Omega^k$, it follows that this ε-ball is contained in Ω^k. Since $\varepsilon \leq 1$, we have $\varepsilon > \varepsilon^2/2$. It follows from Lemma 2.5 that

$$c_j - \langle a_j, y^* \rangle - \frac{1}{2} \langle y^*, Q_j y^* \rangle \geq \frac{\varepsilon^2}{2},$$

for all $j = 1, \ldots, n_k$. Then we have

$$\begin{aligned} P(\Omega^k) &= \min_{y \in \Omega^k} - \sum_{j=1}^{n_k} \ln \left(c_j^k - \langle a_j^k, y \rangle - \frac{1}{2} \langle y, Q_j^k y \rangle \right) \\ &\leq \sum_{j=1}^{n_k} - \ln \left(c_j^k - \langle a_j^k, y^* \rangle - \frac{1}{2} \langle y^*, Q_j^k y^* \rangle \right) \\ &\leq n_k \ln \frac{2}{\varepsilon^2} \\ &\leq (n+\rho) \ln \frac{2}{\varepsilon^2}, \end{aligned}$$

where the first inequality is due to $y^* \in \Omega_k$. The proof is complete. □

We now use this bound to establish the following convergence theorem.

THEOREM 5.2. *Assume* Γ *is defined by* n *convex quadratic inequalities and contains a ball of radius* ε. *Then the column generation algorithm will find a feasible point in* Γ *in* $O(n \ln \frac{1}{\varepsilon})$ *iterations.*

Proof. There are two cases in the algorithm. In the first case Ω^{k+1} is obtained by translating an existing inequality of Ω^k. Since we have $1 - \beta = 1/20$ in this case it follows from Lemma 3.1 that

$$P(\Omega^{k+1}) \geq P(\Omega^k) + \left(1 + \frac{3}{20}\right)^{-1} \frac{1}{20} = P(\Omega^k) + \frac{1}{23}.$$

In the second case we add a new inequality to Ω^k and have $\beta = 20$. It follows from Lemma 3.2 that

$$P(\Omega^{k+1}) \geq P(\Omega^k) - \ln r^k - \ln \left[20 + \frac{2}{20(2-\sqrt{2})} \right].$$

By the bound (4.6), we have

$$\begin{aligned} P(\Omega^{k+1}) &\geq P(\Omega^k) - \ln\sqrt{\frac{2}{\rho}} - \ln\left[20 + \frac{2}{20(2-\sqrt{2})}\right] \\ &\geq P(\Omega^k) - \left|\ln\sqrt{\frac{2}{\rho}}\right| - \ln\left[20 + \frac{2}{20(2-\sqrt{2})}\right]. \end{aligned}$$

Note that the second case can happen at most n times. Thus, after k iterations the potential is increased by

$$P(\Omega^k) - P(\Omega^0) \geq \frac{k-n}{23} - n\left|\ln\sqrt{\frac{2}{\rho}}\right| - n\ln\left[20 + \frac{2}{20(2-\sqrt{2})}\right].$$

On the other hand, it follows from $P(\Omega^0) = 0$ and (5.1) that

$$k \leq n\left\{1 + 23\left[\left|\ln\sqrt{\frac{2}{\rho}}\right| + \ln\left(20 + \frac{2}{20(2-\sqrt{2})}\right)\right]\right\} + 23(n+\rho)\ln\frac{2}{\varepsilon^2},$$

which shows that the algorithm terminates in at most $O(n\ln\frac{1}{\varepsilon})$ iterations. □

We point out that the values of the potential $P(\Omega^k)$ and the exact analytic centers are not needed in the column generation algorithm; they are needed only in the proof of Theorem 5.2. Also, note that Theorem 5.2 only gives the estimate of the total number of iterations required to solve the convex quadratic feasibility problem; it does not estimate the amount of work required to update the center y^k to the new center y^{k+1}. We shall provide such an estimate in the next section.

6. Updating to a new center.

In each step of the analytic center column generation algorithm we need to compute an approximate analytic center y^{k+1} of Ω^{k+1}. In this section, we show that y^{k+1} can be computed by the Newton procedure (4.5) starting from y^k in a constant number of iterations.

Throughout this section all the slacks are evaluated at y^k, and therefore for simplicity, we shall drop the superscripts k and $k+1$ in our notation, and denote y^k, y^{k+1}, a_j^k, Q_j^k, $s^{k+1}(y^k)$, and $s^k(y^k)$ by y, y^+, a_j, Q_j, s^+, and s, respectively. We assume for convenience that the translated inequality is labelled n_k in Case 1. Clearly, we have $s_j = s_j^+$, for $j = 1, \ldots, n_k - 1$. We also let n denote n_k. Furthermore, we denote the Hessians $\nabla^2\phi^{k+1}(y^k)$, $\nabla^2\phi^k(y^k)$ and the gradients $\nabla\phi^{k+1}(y^k)$, $\nabla\phi^k(y^k)$ by H_+, H, and g_+, g, respectively. With this simplified notation, we have, for the iteration given by Case 1 of the column generation algorithm,

$$\begin{aligned} H_+ &:= \sum_{j=1}^{n}\left[\frac{(a_j + Q_j y)(a_j + Q_j y)^T}{(s_j^+)^2} + \frac{Q_j}{s_j^+}\right] \\ &= \sum_{j=1}^{n-1}\left[\frac{(a_j + Q_j y)(a_j + Q_j y)^T}{(s_j)^2} + \frac{Q_j}{s_j}\right] \\ &\quad + \left[\frac{(a_n + Q_n y)(a_n + Q_n y)^T}{\beta^2(s_n)^2} + \frac{Q_n}{\beta s_n}\right], \end{aligned} \tag{6.1}$$

$$H := \sum_{j=1}^{n}\left[\frac{(a_j + Q_j y)(a_j + Q_j y)^T}{(s_j)^2} + \frac{Q_j}{s_j}\right]$$

and

$$
\begin{aligned}
g_+ &:= \sum_{j=1}^{n} \frac{a_j + Q_j y}{s_j^+} \\
&= \sum_{j=1}^{n-1} \frac{a_j + Q_j y}{s_j} + \frac{a_n + Q_n y}{\beta s_n}, \qquad (6.2)\\
g &:= \sum_{j=1}^{n} \frac{a_j + Q_j y}{s_j};
\end{aligned}
$$

while for Case 2, with Ω^{k+1} updated by (4.4), we have

$$
\begin{aligned}
H_+ &:= \sum_{j=1}^{n+1} \left[\frac{(a_j + Q_j y)(a_j + Q_j y)^T}{(s_j^+)^2} + \frac{Q_j}{s_j^+} \right] \\
&= \sum_{j=1}^{n} \left[\frac{(a_j + Q_j y)(a_j + Q_j y)^T}{(s_j)^2} + \frac{Q_j}{s_j} \right] \qquad (6.3)\\
&\quad + \left[\frac{(a_{n+1} + Q_{n+1} y)(a_{n+1} + Q_{n+1} y)^T}{\beta^2 r^2} + \frac{Q_{n+1}}{\beta r} \right], \\
H &= \sum_{j=1}^{n} \left[\frac{(a_j + Q_j y)(a_j + Q_j y)^T}{(s_j)^2} + \frac{Q_j}{s_j} \right] \qquad (6.4)
\end{aligned}
$$

and

$$
\begin{aligned}
g_+ &:= \sum_{j=1}^{n+1} \frac{a_j + Q_j y}{s_j^+} \\
&= \sum_{j=1}^{n} \frac{a_j + Q_j y}{s_j} + \frac{a_{n+1} + Q_{n+1} y}{\beta r}, \qquad (6.5)\\
g &:= \sum_{j=1}^{n} \frac{a_j + Q_j y}{s_j}.
\end{aligned}
$$

Lemma 6.1. *Consider Case* 1 *of the column generation algorithm whereby the iteration* (4.3) *is performed. Let* H_+, H, g_+, *and* g *be given by* (6.1) *and* (6.2). *Then*

$$
\left\langle g_+, H_+^{-1} g_+ \right\rangle^{1/2} \le (1 - \beta) + \left\langle g, H^{-1} g \right\rangle^{1/2}.
$$

Proof. By (6.2) we have

$$
g_+ = g - \left(1 - \frac{1}{\beta}\right) \frac{a_n + Q_n y}{s_n},
$$

and by (6.1) we have

$$\begin{aligned} H_+ &= \sum_{j=1}^{n-1} \left[\frac{(a_j+Q_jy)(a_j+Q_jy)^T}{(s_j)^2} + \frac{Q_j}{s_j} \right] + \left[\frac{(a_n+Q_ny)(a_n+Q_ny)^T}{\beta^2(s_n)^2} + \frac{Q_n}{\beta s_n} \right] \\ &\geq 2\rho I + \left[\frac{(a_n+Q_ny)(a_n+Q_ny)^T}{\beta^2(s_n)^2} + \frac{Q_n}{\beta s_n} \right] \\ &\geq 2\rho I + \frac{(a_n+Q_ny)(a_n+Q_ny)^T}{\beta^2(s_n)^2}, \qquad (6.6) \end{aligned}$$

where ρ is the constant given in the initialization of the column algorithm (see (4.7)). Note that the last step is due to the fact that Q_n is positive semidefinite. Moreover, since $\beta \in [0,1]$, it follows that $H_+ \geq H$, implying that $H_+^{-1} \leq H^{-1}$. Now we can use (6.6) to obtain

$$\begin{aligned} \langle g_+, H_+^{-1} g_+ \rangle^{1/2} &= \left\langle g - \left(1 - \frac{1}{\beta}\right) \frac{a_n + Q_ny}{s_n}, H_+^{-1} \left[g - \left(1 - \frac{1}{\beta}\right) \frac{a_n + Q_ny}{s_n} \right] \right\rangle^{1/2} \\ &\leq \langle g, H_+^{-1} g \rangle^{1/2} + \left(\frac{1}{\beta} - 1\right) \left\langle \frac{a_n + Q_ny}{s_n}, H_+^{-1} \left(\frac{a_n + Q_ny}{s_n}\right) \right\rangle^{1/2} \\ &\leq \langle g, H^{-1} g \rangle^{1/2} + (1-\beta) \Bigg\langle \left(\frac{a_n + Q_ny}{\beta s_n}\right), \\ &\qquad \left[2\rho I + \frac{(a_n+Q_ny)(a_n+Q_ny)^T}{\beta^2(s_n)^2} \right]^{-1} \left(\frac{a_n + Q_ny}{\beta s_n}\right) \Bigg\rangle^{1/2} \\ &= \langle g, H^{-1} g \rangle^{1/2} + (1-\beta) \frac{\|a_n + Q_ny\|}{\sqrt{2\rho\beta^2(s_n)^2 + \|a_n + Q_ny\|^2}} \\ &\leq \langle g, H^{-1} g \rangle^{1/2} + (1-\beta), \end{aligned}$$

where the first inequality is due to the triangular inequality and the last equality follows from Lemma 2.6. □

In section 4, we chose $\beta = 19/20$ in the case of translation of an existing inequality. Since we have $\delta(y^k;\Omega^k) = \langle g, H^{-1} g \rangle^{1/2} \leq 1/20$, it follows from Lemma 6.1 that $\delta(y^k;\Omega^{k+1}) \leq 1/20 + (1 - 19/20) = 1/10$, implying that y^k is "relatively close" to the analytic center of Ω^{k+1}. This makes it possible for the Newton procedure (4.5) to pull y^k even closer to the analytic center of Ω^{k+1}.

The next lemma considers Case 2 of the column generation algorithm whereby a new inequality is added to Ω^k.

LEMMA 6.2. *Let H_+, H, g_+, and g be given by* (6.3)–(6.5). *Then*

$$\langle g_+, H_+^{-1} g_+ \rangle^{1/2} \leq \frac{1}{\beta} + \langle g, H^{-1} g \rangle^{1/2}.$$

Proof. Recall that

$$r = \langle h, H^{-1} h \rangle^{1/2}, \quad \text{where } h := a_{n+1} + Q_{n+1}y.$$

It follows from (6.5) and (6.3) that

$$g_+ = g + \frac{h}{\beta r} \quad \text{and} \quad H_+^{-1} \leq H^{-1}.$$

By using these relations and the triangular inequality, we have

$$\begin{aligned} \langle g_+, H_+^{-1} g_+ \rangle &= \left\langle g + \frac{h}{\beta r}, H_+^{-1}\left(g + \frac{h}{\beta r}\right)\right\rangle^{1/2} \\ &\leq \langle g, H_+^{-1} g\rangle^{1/2} + \frac{1}{\beta r}\langle h, H^{-1} h\rangle^{1/2} \\ &= \langle g, H_+^{-1} g\rangle^{1/2} + \frac{1}{\beta} \\ &\leq \langle g, H^{-1} g\rangle^{1/2} + \frac{1}{\beta}, \end{aligned}$$

where the second equality follows from the definition of r. □

Recall from section 4 that we chose $\beta = 20$ in Case 2 when a new inequality is added to Ω^k. Since the algorithm always maintains $\delta(y^k;\Omega^k) \leq 1/20$, it follows from Lemma 6.2 that $\delta(y^k;\Omega^{k+1}) \leq 1/20 + 1/20 = 1/10$. In other words, after either a translation of an existing inequality or an addition of a new inequality, the current iterate y^k still remains "relatively close" to the analytic center of the new system Ω^{k+1} in the sense that $\delta(y^k;\Omega^{k+1}) \leq 1/10$. Once Ω^{k+1} is defined, the column generation algorithm goes on to find a new approximate center for Ω^{k+1} by performing a sequence of Newton iterations starting from y^k. The following theorem estimates the total number of Newton iterations needed to generate a new iterate y^{k+1} such that $\delta(y^{k+1};\Omega^{k+1}) \leq 1/20$.

THEOREM 6.3. *Let $\phi^{k+1}(\cdot)$ denote the potential function for Ω^{k+1}. Suppose that the following Newton iteration is initialized at y^k:*

$$y^{new} := y - \left(\nabla^2\phi^{k+1}(y)\right)^{-1}\nabla\phi^{k+1}(y). \tag{6.7}$$

Then, after a constant number of iterations, we shall obtain an iterate y^{k+1} with

$$\delta(y^{k+1};\Omega^{k+1}) \leq \frac{1}{20},$$

where $\delta(\cdot\,;\cdot)$ is the proximity measure (2.4).

Proof. Let $\bar{y}^0$, $\bar{y}^1$, ... denote the sequence of Newton iterates generated by (6.7). Thus, $\bar{y}^0 := y^k$. Let y^a denote the analytic center of Ω^{k+1}. By the discussion following Lemma 6.2 we have $\delta(\bar{y}^0;\Omega^{k+1}) = \delta(y^k;\Omega^{k+1}) \leq 1/10$. Then it follows from Lemma 2.2 that

$$\phi^{k+1}(\bar{y}^0) - \phi^{k+1}(y^a) \leq 4(1/10)^2 = 0.04. \tag{6.8}$$

By Lemma 2.3, we obtain

$$\left\langle g_+(\bar{y}^0), H_+^{-1}(\bar{y}^0) g_+(\bar{y}^0)\right\rangle = \delta(\bar{y}^0;\Omega^{k+1})^2 \leq 0.14,$$

where $g_+(\cdot)$, $H_+(\cdot)$ denote the gradient and the Hessian of ϕ^{k+1}. This implies that

$$\begin{aligned} \left\langle \bar{y}^1 - \bar{y}^0, H_+(\bar{y}^0)(\bar{y}^1 - \bar{y}^0)\right\rangle &= \left\langle g_+(\bar{y}^0), H_+^{-1}(\bar{y}^0) g_+(\bar{y}^0)\right\rangle \\ &\leq 0.14 < 1. \end{aligned}$$

Thus, Lemma 2.1 can be applied to the points $x = \bar{y}^0$, $y = \bar{y}^1$ by identifying $\phi(\cdot) := \phi^{k+1}(\cdot)$, and $\Omega := \Omega^{k+1}$. Consequently, we have for $t = 0$

$$\begin{aligned}\phi^{k+1}(\bar{y}^t) - \phi^{k+1}(\bar{y}^{t+1}) &\geq \left\langle g_+(\bar{y}^t), H_+^{-1}(\bar{y}^t)g_+(\bar{y}^t)\right\rangle - \frac{1}{2}\left\langle g_+(\bar{y}^t), H_+^{-1}(\bar{y}^t)g_+(\bar{y}^t)\right\rangle \\ &\quad - \frac{\left\langle g_+(\bar{y}^t), H_+^{-1}(\bar{y}^t)g_+(\bar{y}^t)\right\rangle^{3/2}}{3\left[1 - \left\langle g_+(\bar{y}^t), H_+^{-1}(\bar{y}^t)g_+(\bar{y}^t)\right\rangle^{1/2}\right]} \\ &\geq 0.295\left\langle g_+(\bar{y}^t), H_+^{-1}(\bar{y}^t)g_+(\bar{y}^t)\right\rangle. \qquad (6.9)\end{aligned}$$

As a by-product, we get $\phi^{k+1}(\bar{y}^0) \geq \phi^{k+1}(\bar{y}^1)$. Therefore (6.8) is still valid if we replace $\bar{y}^0$ with $\bar{y}^1$. Repeating the entire argument from (6.8) to (6.9) for $t = 1, 2, \cdots$, we conclude that

$$\phi^{k+1}(\bar{y}^t) - \phi^{k+1}(\bar{y}^{t+1}) \geq 0.295\left\langle g_+(\bar{y}^t), H_+^{-1}(\bar{y}^t)g_+(\bar{y}^t)\right\rangle \quad \text{for all } t = 0, 1, \cdots.$$

Suppose τ is the largest integer such that $\delta(\bar{y}^t; \Omega^{k+1}) > 1/20$ for all $t = 0, \ldots, \tau$. Then for each iteration $t \leq \tau$, we have

$$\phi^{k+1}(\bar{y}^t) - \phi^{k+1}(\bar{y}^{t+1}) \geq 0.295\left(\frac{1}{20}\right)^2.$$

Thus,

$$\begin{aligned}\phi^{k+1}(\bar{y}^{t+1}) - \phi^{k+1}(y^a) &\leq \left(\phi^{k+1}(\bar{y}^0) - \phi^{k+1}(y^a)\right) - \tau \times 0.295\left(\frac{1}{20}\right)^2 \\ &= \left(\phi^{k+1}(y^k) - \phi^{k+1}(y^a)\right) - \tau \times 0.295\left(\frac{1}{20}\right)^2 \\ &\leq 4\left(\frac{1}{10}\right)^2 - \tau \times 0.295\left(\frac{1}{20}\right)^2,\end{aligned}$$

where in the last step we have used (6.8). Since $\phi^{k+1}(\bar{y}^{t+1}) \geq \phi^{k+1}(y^a)$, it follows that

$$\tau \leq \frac{4 \times 20^2}{0.295 \times 10^2} < 55.$$

Thus, after no more than 55 Newton iterations (6.7) we shall have an iterate $\bar{y}^{\tau+1}$ with the property $\delta(\bar{y}^{\tau+1}; \Omega^{k+1}) \leq 1/20$. □

Since we have not optimized the choice of the various constants, the number of Newton iterations in Theorem 6.3 seems high. We expect this number to be small in the practical implementations of the column generation algorithm.

Combining Theorem 6.3 with Theorem 5.2 yields the following polynomial complexity result.

THEOREM 6.4. *Suppose that Γ is defined by n convex quadratic inequalities and contains an ε-ball. Then the column generation algorithm can find a feasible point in Γ in at most $O(n \ln \frac{1}{\varepsilon})$ Newton iterations.*

When all the data describing Γ (i.e., the vectors a_j, the matrices Q_j, and the scalars c_j) are rational and their total length in binary coding is L, then it can be shown that the set Γ, if nonempty, always contains a ball of size $O(2^{-L})$. Thus, Theorem 6.4 shows that the convex quadratic feasibility problem can be solved by the column generation algorithm in $O(nL)$ Newton iterations.

REFERENCES

[1] A. Altman and K. C. Kiwiel, *A note on some analytic center cutting plane methods for convex feasibility and minimization problems*, Comput. Optim. Appl., 5 (1996), pp. 175–180.

[2] D. S. Atkinson, *Scaling and Interior Point Methods in Optimization*, Ph.D. thesis, Coordinated Science Laboratory, College of Engineering, University of Illinois at Urbana-Champaign, 1992.

[3] D. S. Atkinson and P. M. Vaidya, *A cutting plane algorithm for convex programming that uses analytic centers*, Math. Programming, 69 (1995), pp. 1–44.

[4] O. Bahn, O. Du Merle, J.-L. Goffin, and J.-P. Vial, *Experimental behavior of an interior point cutting plane algorithm for convex programming: An application to geometric programming,* Discrete Appl. Math., 49 (1994), pp. 3–23.

[5] O. Bahn, O. Du Merle, J.-L. Goffin, and J.-P. Vial, *A cutting plane method from analytic centers for stochastic programming*, Math. Programming, 69 (1995), pp. 45–74.

[6] D. den Hertog, C. Roos, and T. Terlaky, *On the classical logarithmic barrier function method for a class of smooth convex programming problems*, J. Optim. Theory Appl., 73 (1992), pp. 1–25.

[7] J.-L. Goffin, Z.-Q. Luo, and Y. Ye, *Complexity analysis of an interior cutting plane method for convex feasibility problems*, SIAM J. Optim., 6 (1996), pp. 638–652.

[8] J.-L. Goffin, Z.-Q. Luo, and Y. Ye, *On the complexity of a column generation algorithm for convex or quasiconvex feasibility problems*, in Large Scale Optimization: State of the Art, W. W. Hager, D. W. Hearn, and P. M. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA, 1994, pp. 182–189.

[9] F. Jarre, *On the convergence of the method of analytic centers when applied to convex quadratic programs*, Math. Programming, 49 (1991), pp. 341–358.

[10] Z.-Q. Luo, *Analysis of a cutting plane method that uses weighted analytic center and multiple cuts*, SIAM J. Optim., 7 (1997), pp. 697–716.

[11] S. Mehrotra and J. Sun, *A method of analytic centers for quadratically constrained convex quadratic programs*, SIAM J. Numer. Anal., 28 (1991), pp. 529–544.

[12] S. Mehrotra and J. Sun, *On computing the center of a convex quadratically constrained set*, Math. Programming, 50 (1991), pp. 81–89.

[13] Y. Nesterov, *Cutting plane algorithms from analytic centers: Efficiency estimates*, Math. Programming, 69 (1995), pp. 149–176.

[14] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*, SIAM Studies in Applied Mathematics, SIAM, Philadelphia, PA, 1994.

[15] G. Sonnevend, *New algorithms in convex programming based on a notion of "centre"* (*for systems of analytic inequalities*) *and on rational extrapolation*, Trends in Mathematical Optimization: Proceedings of the 4th French-German Conference on Optimization in Irsee, West Germany, 1986, Internat. Schriftenreiche Numer. Math. 84, K. H. Hoffmann, J.-B. Hiriart-Urruty, C. Lemarechal, and J. Zowe, eds., Birkhäuser, Boston, pp. 311–326.

[16] Y. Ye, *A Potential reduction algorithm allowing column generation*, SIAM J. Optim., 2 (1992), pp. 7–20.

[17] Y. Ye, *Complexity analysis of the analytic center cutting plane method that uses multiple cuts*, Math. Programming, 78 (1997), pp. 85–104.

SIAM J. OPTIM.
Vol. 9, No. 1, pp. 236–269

DECOMPOSING MATRICES INTO BLOCKS[*]

RALF BORNDÖRFER[†], CARLOS E. FERREIRA[‡], AND ALEXANDER MARTIN[†]

Abstract. In this paper we investigate whether matrices arising from linear or integer programming problems can be decomposed into so-called *bordered block diagonal form*. More precisely, given some matrix A, we try to assign as many rows as possible to some number β of blocks of size κ such that no two rows assigned to different blocks intersect in a common column. Bordered block diagonal form is desirable because it can guide and speed up the solution process for linear and integer programming problems. We show that various matrices from the linear programming and mixed integer programming libraries Netlib and Miplib can indeed be decomposed into this form by computing optimal decompositions or decompositions with proven quality. These computations are done with a branch-and-cut algorithm based on polyhedral investigations of the matrix decomposition problem. In practice, however, one would use heuristics to find a good decomposition. We present several heuristic ideas and test their performance. Finally, we investigate the usefulness of optimal matrix decompositions into bordered block diagonal form for integer programming by using such decompositions to guide the branching process in a branch-and-cut code for general mixed integer programs.

Key words. block structure of a sparse matrix, matrix decomposition, integer programming, polyhedral combinatorics, cutting planes

AMS subject classifications. 90C10, 65F50

PII. S1052623497318682

1. Introduction.

In this paper we consider the following *matrix decomposition problem*. Given some matrix A, some number β of *blocks* (sets of rows), and some *capacity* κ (maximum block-size), try to assign as many rows as possible to the blocks such that (i) each row is assigned to at most one block, (ii) each block contains at most κ rows, and (iii) no two rows in different blocks have a common nonzero entry in a column. The set of rows that are not assigned to any block is called the *border*.

An equivalent statement of the problem in matrix terminology is as follows: Try to decompose the matrix into *bordered block diagonal form* with β blocks of capacity at most κ. The smaller the number of rows in the border, the better the decomposition is considered; in the best case the border will be empty and the matrix decomposes into block diagonal form. The left side of Figure 1.1 shows the structure of a 55×55 nonsymmetric matrix, namely, an optimal basis matrix of the Netlib problem `recipe`. The right side of Figure 1.1 shows an optimal decomposition of this matrix into four blocks of capacity $\lceil 55/4 \rceil = 14$. To make the block structure of the decomposition visible, we have permuted not only the rows (such that those assigned to the same block appear consecutively) but also the columns. In this case, the blocks turn out to be almost square with sizes 13×13, 13×13, 14×14, and 14×15, but in general this does not need to be the case. The border consists of only one row that could not be

[*] Received by the editors March 24, 1997; accepted for publication (in revised form) October 27, 1997; published electronically December 2, 1998. This work was supported by the Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and the German Academic Exchange Service (DAAD), program PROBRAL.

http://www.siam.org/journals/siopt/9-1/31868.html

[†] Konrad-Zuse-Zentrum Berlin, Takustraße 7, 14195 Berlin, Germany (borndoerfer@zib.de, martin@zib.de).

[‡] Universidade de São Paulo, Rua do Matão, 1010, 05508-970 São Paulo, SP, Brazil (cef@ime.usp.br).

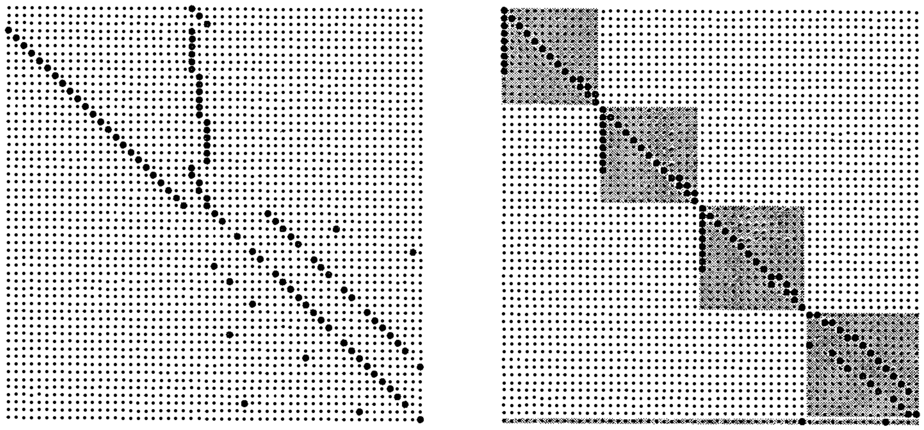


FIG. 1.1. *Decomposing a matrix into bordered block diagonal form.*

assigned to any block.

The matrix decomposition problem fits into the general context of reordering matrices to *special forms.* Special forms are well studied in the literature because they can be exploited by solution methods for linear equation systems, for example by LU- or Cholesky factorization, or by conjugate gradient methods. The two main points of interest are that special forms allow (i) the control of fill-in (bordered block diagonal form, in particular, restricts fill-in to the blocks and the border) and (ii) independent processing of individual blocks by parallel algorithms.

Methods to obtain special forms, including (bordered) block diagonal form, are widely discussed in the literature of computational linear algebra; see, for instance, Duff, Erisman, and Reid [5], Kumar et al. [19], or Gallivan et al. [10]. The matrices studied in this context mainly arise from the discretization of partial differential equations. Some newer publications deal with matrices that appear in interior point algorithms for linear programs; see Gupta [15] and Rothberg and Hendrickson [26].

The applications we have in mind are different. We are interested in matrices arising from *(mixed) integer programs* (MIPs). Such matrices often have potential for decomposition into bordered block diagonal form for two reasons. First, such matrices are sparse and generally have a small number of nonzero entries in each column. Second, bordered block diagonal form comes up in a natural way in many real-world MIPs. The problems are often composed of small blocks to model decisions in a division of a company, in a technical unit, or in a time period. These individual blocks are linked by a couple of constraints that model possible interactions to yield an appropriate model that covers the whole company, technical process, or time horizon. Examples of this type are production planning problems like the unit commitment problem in power generation, where the blocks correspond to single unit subproblems and the border is given by load balance and reserve constraints (see Sheble and Fahd [28] for a literature synopsis and Dentcheva et al. [4] for a recent application); multicommodity flow problems that come up, for example, in vehicle scheduling (see Löbel [21]); classes of combinatorial programs like the Steiner-tree packing problem (see Grötschel, Martin, and Weismantel [14]); or, recently, scenario decompositions of stochastic MIPs (see Carøe and Schultz [2]).

Bordered block diagonal form helps to *accelerate* the solution process of integer programs in several ways. The first is to solve the LP-relaxation of an integer program;

here bordered block diagonal form can speed up the required linear algebra (both if a simplex-type method or if an interior point method is used). Second, it can be used to improve the polyhedral description of the set of feasible points of an MIP. For example, given a block decomposition, taking one constraint from each block plus an according number from the border results in the structure of a generalized assignment or multiple knapsack problem (see Gottlieb and Rao [12] and Ferreira, Martin, and Weismantel [8]) whose facets are valid inequalities for the MIP under consideration. Third, block decomposition can be used in a branch-and-bound or -cut algorithm to guide branching decisions: Decomposing the transposed constraint matrix A^T will identify the columns in the border as linking columns that are interesting candidates for branching.

In this paper we develop a branch-and-cut algorithm for solving the matrix decomposition problem. Of course the expected running time of such an algorithm will not permit its usage within a parallel LU-factorization or within a branch-and-cut algorithm for general MIPs. Our aim is rather to have a tool at hand that in principle obtains an *optimal* bordered block diagonal form. We can then evaluate whether this special matrix structure indeed helps in solving general integer programs, and we can evaluate the success of decomposition heuristics that try to obtain (bordered) block diagonal form and that could be used, for instance, within a parallel LU-factorization framework.

The paper is organized as follows. In section 2 we formulate the matrix decomposition problem as a 0/1 linear program and discuss connections to related combinatorial optimization problems, namely, node separation problems in graphs, the set packing, and the set covering problem. Section 3 is devoted to a polyhedral investigation of the matrix decomposition problem and presents (new) valid and facet defining inequalities. In the branch-and-cut section 4 we present our matrix decomposition algorithm, including separation routines, primal heuristics, preprocessing, and other aspects of the implementation. We use this code in section 5 to decompose optimal basis matrices of linear programs taken from the Netlib (available by anonymous ftp from ftp://netlib2.cs.utk.edu), to decompose matrices arising from general MIPs from the Miplib (available from the URL http://www.caam.rice.edu:80/∼bixby/miplib/miplib.html), and to solve some equipartition problems investigated by Nicoloso and Nobili [22].

2. Integer programming formulation and related problems. Consider an instance (A, β, κ) of the matrix decomposition problem where $A \in \mathbb{R}^{m \times n}$ is some real matrix, $\beta \in \mathbb{N}$ is the number of blocks, and $\kappa \in \mathbb{N}$ is the block capacity. We introduce for each row $i = 1, \dots, m$ and block $b = 1, \dots, \beta$ a binary variable x_i^b that has value 1 if row i is assigned to block b and 0 otherwise. Then the matrix decomposition problem (A, β, κ) can be formulated as the 0/1 linear program (IP) that is stated on the facing page.

Inequalities (i) guarantee that each row is assigned to at most one block. Constraints (ii) ensure that the number of rows assigned to a particular block b does not exceed its capacity. Finally, (iii) expresses that two rows i and j must not be assigned to different blocks if both have a nonzero entry in some common column. These three sets of inequalities plus the bounds (iv) and the integrality constraints (v) establish a one-to-one correspondence between feasible solutions of (IP) and block decompositions of the matrix A into β blocks of capacity κ. In the following discussion we will also call a vector $x \in \mathbb{R}^{m \times \beta}$ a *block decomposition* if it is feasible for (IP). Note that formulation (IP) as it stands is not polynomial, since the number of variables $m\beta$ is not polynomial in the encoding length of β. However, we may assume without loss of generality that $\beta \le m$, because no more than m rows will be assigned. We also assume that the block capacity is at least one ($\kappa \ge 1$) and that we have at least two blocks ($\beta \ge 2$):

$$\text{(IP)}\qquad \begin{array}{rll} \max & \displaystyle\sum_{i=1}^{m}\sum_{b=1}^{\beta} x_i^b & \\ \text{(i)} & \displaystyle\sum_{b=1}^{\beta} x_i^b \le 1 & \text{for } i = 1,\dots,m; \\ \text{(ii)} & \displaystyle\sum_{i=1}^{m} x_i^b \le \kappa & \text{for } b = 1,\dots,\beta; \\ \text{(iii)} & x_i^b + x_j^{b'} \le 1 & \text{for } b,b' = 1,\dots,\beta,\ b \ne b',\ \text{and} \\ & & \text{for } i,j = 1,\dots,m,\ i \ne j,\ \text{such that} \\ & & a_{ik} \ne 0 \ne a_{jk} \text{ for some } k \in \{1,\dots,n\}; \\ \text{(iv)} & 0 \le x_i^b \le 1 & \text{for } i = 1,\dots,m,\ b = 1,\dots,\beta; \\ \text{(v)} & x_i^b \text{ integer} & \text{for } i = 1,\dots,m,\ b = 1,\dots,\beta. \end{array}$$

A first observation about (IP) is that different matrices A can give rise to the same integer program or, in other words, different matrices can be decomposed in exactly the same way. In fact, such matrices form equivalence classes as can be seen by considering the *(column) intersection graph* $G(A)$ of an $m \times n$-matrix A as introduced by Padberg [24]. $G(A)$ has the set $\{1,\dots,n\}$ of column indices of A as its node set, and there is an edge ij between two columns i and j if they have a common nonzero entry in some row. Applying this concept to the transposed matrix A^T, we obtain the *row intersection graph* $G(A^T)$ of A where two rows i and j are joined by an edge ij if and only if they have nonzero entries in a common column. But then the edges of $G(A^T)$ give rise to the inequalities (IP) (iii) and we have that, for fixed β and κ, two matrices A and A' have the same row intersection graph if and only if the corresponding integer programs (IP) are equal.

The matrix decomposition problem is related to several other combinatorial optimization problems. First, the problem can be interpreted in terms of the row intersection graph as a *node separator problem.* To see this, let $G(A^T) = (V, E)$ and consider some block decomposition x. The set $S := \{i \in V : \sum_{b=1}^{\beta} x_i^b = 0\}$ of rows in the border is a node separator in $G(A^T)$ such that the graph obtained by deleting all nodes in S and their adjacent edges decomposes into at most β parts, each of cardinality of at most κ. Conversely, each node separator in $G(A^T)$ with these properties gives rise to a block decomposition for (A, β, κ). Various node separator problems have been studied in the literature. Lengauer [20] gives a survey and discusses applications in VLSI design, and Duff, Erisman, and Reid [5] and Gallivan et al. [10] emphasize heuristic methods for use in computational linear algebra. Lower bounds on the size of a node separator in a general graph are rather rare. The only results we are aware of are from Pothen, Simon, and Liou [25] and Helmberg et al. [16], who use eigenvalue methods to derive nontrivial lower bounds on the size of a node separator for $\beta = 2$ if lower bounds on the size of the blocks are imposed.

A second connection exists to *set packing*, and this relationship is twofold. On the one hand, matrix decomposition is a generalization of set packing, because feasible solutions (stable sets) of some set packing problem $\max\{\mathbb{1}^T x : Ax \le \mathbb{1}, x \in$

$\{0,1\}^n\}$, $A \in \{0,1\}^{m\times n}$ correspond to solutions of the matrix decomposition problem $(A^T, m, 1)$ of the same objective value, and vice versa. This shows that the matrix decomposition problem is $\mathcal{NP}$-hard. On the other hand, we obtain a *set packing relaxation* of the matrix decomposition problem by deleting the block capacity constraints (ii) from the formulation (IP). All inequalities that are valid for this relaxation are also valid for the matrix decomposition problem, and we will use some of them (namely clique- and cycle-inequalities) as cutting planes in our branch-and-cut algorithm. Note, however, that the set packing relaxation allows assignment of all rows to any single block, and our computational experiments seem to indicate that these cuts are rather weak.

A close connection exists also to *set covering* via complementing variables. To see this we rewrite (IP), substituting each capacity constraint (ii) with $\binom{m}{\kappa+1}$ inequalities that sum over all subsets of cardinality $\kappa+1$ of variables $\{x_1^b, \ldots, x_m^b\}$ for some block b, and each constraint in (i) with $\binom{\kappa}{2}$ inequalities that sum over all pairs of variables in $\{x_i^1, \ldots, x_i^\beta\}$. Replacing all variables x_i^b with $1 - y_i^b$, one obtains the following set covering problem:

$$(\mathrm{IP}_c)\qquad \begin{array}{lll}
\min & \displaystyle\sum_{i=1}^{m}\sum_{b=1}^{\beta} y_i^b & \\
\text{(i)} & y_i^b + y_i^{b'} \geq 1 & \text{for } b, b' = 1, \ldots, \beta,\ b \neq b',\ \text{and} \\
 & & \text{for } i = 1, \ldots, m; \\
\text{(ii)} & \displaystyle\sum_{i\in I} y_i^b \geq 1 & \text{for } b = 1, \ldots, \beta \text{ and} \\
 & & \text{for } I \subseteq \{1, \ldots, m\} \text{ with } |I| = \kappa + 1; \\
\text{(iii)} & y_i^b + y_j^{b'} \geq 1 & \text{for } b, b' = 1, \ldots, \beta,\ b \neq b',\ \text{and} \\
 & & \text{for } i, j = 1, \ldots, m,\ i \neq j,\ \text{such that} \\
 & & a_{ik} \neq 0 \neq a_{jk} \text{ for some } k \in \{1, \ldots, n\}; \\
\text{(iv)} & 0 \leq y_i^b \leq 1 & \text{for } i = 1, \ldots, m,\ b = 1, \ldots, \beta; \\
\text{(v)} & y_i^b \text{ integer} & \text{for } i = 1, \ldots, m,\ b = 1, \ldots, \beta.
\end{array}$$

This shows that the matrix decomposition problem is a (special) set covering problem. For the case of two blocks, this formulation has been used by Nicoloso and Nobili [22] for the solution of the *matrix equipartition problem*. The matrix equipartition problem is the matrix decomposition problem for $\beta = 2$ and $\kappa = \lfloor m/2 \rfloor$, plus the additional equipartition constraint

$$\sum_{i=1}^{m} x_i^1 = \sum_{i=1}^{m} x_i^2 \quad \text{or, in complemented variables,} \quad \sum_{i=1}^{m} y_i^1 = \sum_{i=1}^{m} y_i^2,$$

which states that the two blocks of the decomposition must have equal size.

3. Polyhedral investigations. Associated with the IP-formulation (IP) of the matrix decomposition problem is the polytope

$$P(A, \beta, \kappa) := \operatorname{conv}\{x \in \mathbb{R}^{m\times\beta} : x \text{ satisfies (IP) (i)–(v)}\}, \tag{3.1}$$

given by the convex hull of all block decompositions. In this section we study the structure of $P(A, \beta, \kappa)$ to derive classes of valid and facet defining inequalities for later use as cutting planes. We start by determining its dimension.

Proposition 3.1 (dimension). *$P(A, \beta, \kappa)$ is full dimensional.*

Proof. The vector 0 and all unit vectors $e_i^b \in \mathbb{R}^{m\times\beta}$ are feasible, i.e., are in $P(A, \beta, \kappa)$, and affinely independent. □

This means that the facets of $P(A, \beta, \kappa)$ are uniquely determined up to a scalar factor (see Schrijver [27]). Two further easy observations are gathered in the following remark.

Remark 3.1.

(i) The nonnegativity inequalities $x_i^b \geq 0$ are facet defining for all $i = 1, \dots, m$ and all $b = 1, \dots, \beta$.

(ii) All facet defining inequalities $a^T x \leq \alpha$ that are not nonnegativity constraints satisfy $a \geq 0$ and $\alpha > 0$.

Remark 3.1 (i) is proven in the same way as Proposition 3.1, and Remark 3.1 (ii) is a consequence of the down monotonicity of $P(A, \beta, \kappa)$.

Facet defining inequalities have another interesting property. Consider some vector $x \in \mathbb{R}^{m\times\beta}$ and some *permutation σ of the blocks* $\{1, \dots, \beta\}$, and define the vector $\bar{x} \in \mathbb{R}^{m\times\beta}$ by

$$\bar{x}_i^b := x_i^{\sigma(b)}$$

for $i = 1, \dots, m, b = 1, \dots, \beta$. In the following discussion we will use the symbol $\sigma(x)$ to denote the vector $\bar{x}$ that arises from x by applying the block permutation σ. Then $\sigma(x) = \bar{x}$ is a feasible block decomposition if and only if x is. This simple observation has two consequences. First, it implies that $a^T x \leq b$ is a facet of $P(A, \beta, \kappa)$ if and only if its blockwise permutation $\sigma(a)^T x \leq b$ is a facet of $P(A, \beta, \kappa)$. Facets arising from each other via block permutations can thus be viewed as forming a single class that can be represented by a single member, or, to put it in a more negative way, each facet can and will be "blown up" by block permutations to a whole set of combinatorially essentially identical conditions. Second, the objective function of the matrix decomposition problem is invariant under block permutation, and thus the matrix decomposition problem is dual degenerate (has multiple optima). Both dual degeneracy and the large number of permutable facets cause difficulties in our branch-and-cut algorithm, and we will have to control the number of cuts generated and handle stalling of the objective value.

The next two subsections list the results of our polyhedral investigations in the form of valid and facet defining inequalities. We distinguish between inequalities $a^T x \leq b$ that are *invariant under block permutations* or, equivalently, have the same coefficients $a_i^b = a_i^{b'}$ for all blocks $b \neq b'$ and row indices i, and *block-discernible inequalities* that do not have this property and distinguish different blocks. It will turn out that most of the block-discernible inequalities will be inherited from the stable set relaxation of the matrix decomposition problem, while the block-invariant constraints are related to an "aggregated" version of the problem. In both subsections we want to assume $\kappa \geq 2$, because otherwise the matrix decomposition problem is a (special) set packing problem.

3.1. Block-discernible inequalities.

We saw in section 2 that we obtain a set packing relaxation of the matrix decomposition problem by dropping the block capacity constraints (ii) from the integer program (IP). The column intersection graph associated with the matrix $\text{IP}_{(\text{i}),(\text{iii})}$ formed by the left-hand sides of the constraints (IP) (i) and (iii) has the set of possible row assignments $\{1, \dots, m\} \times \{1, \dots, \beta\}$ as its node set. A (conflict) edge exists between two assignments (i, b) and (j, b') if rows i and j cannot be simultaneously assigned to the blocks b and b', i.e., if either $i = j$

and $b \neq b'$ or if $i \neq j$, $b \neq b'$, and rows i and j have a common nonzero entry in some column of A. We want to call this graph the *conflict graph* associated with the matrix decomposition problem (A, β, κ) and denote it by $G_c(A, \beta)$. In formulas, $G_c(A, \beta) = G(\mathrm{IP}_{(\mathrm{i}),(\mathrm{iii})})$. This graph allows us to interpret the inequality classes (i) and (iii) of (IP) as *clique inequalities* of the set packing relaxation corresponding to the matrix decomposition problem as also introduced by Padberg [24].

THEOREM 3.2 (clique). *Let $G_c(A, \beta) = (V, E)$ and $Q \subseteq V$. The inequality*

$$\sum_{(i,b)\in Q} x_i^b \leq 1$$

is valid for $P(A, \beta, \kappa)$ if and only if Q is a clique in $G_c(A, \beta)$. It is facet defining if and only if Q is a maximal clique in $G_c(A, \beta)$.

Proof. The validity part is obvious. It remains to show that it is facet defining if and only if Q is a maximal clique.

Suppose first that Q is not maximal but contained in a larger clique Q'. But then $\sum_{(i,b)\in Q} x_i^b \leq 1$ is the sum of the inequality $\sum_{(i,b)\in Q'} x_i^b \leq 1$ and the nonnegativity constraints $x_i^b \geq 0$ for $(i, b) \in Q' \setminus Q$ and cannot be facet defining.

Assume now that Q is maximal. We will construct a set of $m\beta$ affinely independent block decompositions for which the inequality is tight. $|Q|$ such affinely independent vectors are the unit vectors e_i^b with $(i, b) \in Q$. For each other assignment $(j, b') \notin Q$ there exists some assignment (i, b) in Q that is not in conflict with (j, b'), since Q is a maximal clique. Thus, the vector $e_j^{b'} + e_i^b$ is the incidence vector of a feasible block decomposition for which the inequality is tight. (Note that we assumed $\kappa \geq 2$ at the beginning of this section for the case $b = b'$.) The resulting $m\beta - |Q|$ characteristic vectors obtained in this way plus the $|Q|$ vectors constructed in the beginning are affinely independent. □

In the spirit of Theorem 3.2, (IP) (i) and (iii) are both clique inequalities and do not represent two different types of inequalities. The separation problem for clique inequalities is a maximum-weight clique problem and thus $\mathcal{NP}$-hard; see Garey and Johnson [11]. However, some subclasses can be separated efficiently. One such class that we use in our implementation are the *two-partition inequalities*

$$\sum_{b\in B} x_i^b + \sum_{b'\notin B} x_j^{b'} \leq 1,$$

which are defined for all sets of blocks $B \subseteq \{1, \dots, \beta\}$ and all pairs of nondisjoint rows i, j. Polynomial separation of this class is by inspection: Given i and j, we examine for each block b the variables x_i^b and x_j^b. If $x_i^b > x_j^b$, we add b to the set B; otherwise we add it to its complement. Note that for the case of two blocks ($\beta = 2$), the two-partition inequalities are exactly the inequalities (IP) (i) and (iii) and, moreover, these are already all clique inequalities. In particular, separation of clique inequalities is polynomial for $\beta = 2$. In general, maximal cliques in $G_c(A, \beta)$ are of the form $\{(i_1, b_1), \dots, (i_\beta, b_\beta)\}$, where the blocks $b_k, k = 1, \dots, \beta$, are mutually different and the set of rows $\{i_1, \dots, i_\beta\}$ forms a clique in $G(A^T)$. Thus all maximal cliques in $G_c(A, \beta)$ are of size β.

Another class inherited from the set packing relaxation are the *cycle inequalities.*

THEOREM 3.3 (odd cycle). *If C is an odd cycle in $G_c(A, \beta)$, then the cycle inequality*

$$\sum_{(i,b)\in C} x_i^b \leq \lfloor |C|/2 \rfloor$$

is valid for $P(A, \beta, \kappa)$.

Analogously to the set packing case (see again Padberg [24]), the odd cycle inequality is facet defining for its support if C is an odd hole (has no chords) and $|C|/2 \leq \kappa$. These conditions are, however, not necessary. Cycle inequalities can be separated in polynomial time using the algorithm of Lemma 9.1.11 in Grötschel, Lovász, and Schrijver [13].

Along the same lines as for the clique and cycle inequalities, the matrix decomposition polytope clearly also inherits all other packing inequalities. However, not only set packing but also set covering inequalities for (IP$_c$) can be applied (note that complementing variables preserves validity and dimension of the induced face); see Nobili and Sassano [23]. We do not use any of them for our computations, however.

We close this section investigating the *block capacity constraints* (IP) (ii) which are not inherited from the set packing polytope or the set covering polytope.

THEOREM 3.4 (block capacity). *The block capacity constraint*

$$\sum_{i=1}^{m} x_i^b \leq \kappa$$

is facet defining for $P(A, \beta, \kappa)$ *if and only if* $|\gamma(i)| \leq m - \kappa$ *holds for every row* i *(where* $\gamma(i)$ *denotes all nodes adjacent to* i *in* $G(A^T)$*).*

Proof. We first show that the inequality is facet defining if the above mentioned condition holds. To this purpose, let $a^T x \leq \alpha$ be a valid inequality that induces a facet such that $\{x \in P(A, \beta, \kappa) | \sum_{i=1}^{m} x_i^b = \kappa\} \subseteq \{x \in P(A, \beta, \kappa) | a^T x = \alpha\}$. We will show that the two inequalities are the same up to a positive scalar multiplicative factor.

Define x by

$$x_i^{b'} = \begin{cases} 1 & \text{if } 1 \leq i \leq \kappa,\ b' = b, \\ 0 & \text{otherwise.} \end{cases}$$

x is a feasible block decomposition that assigns the first κ rows to block b. x satisfies the block capacity constraint with equality and thus $a^T x = \alpha$. Now observe that, for all $1 \leq i \leq \kappa < j \leq m$, the vector $x - e_i^b + e_j^b$ is also a feasible assignment that is tight for the block capacity inequality. It follows that $a_i^b = a_j^b$ for all $1 \leq i, j \leq m$.

Now consider assigning some row j to a block $b' \neq b$. By the assumption $|\gamma(j)| \leq m - \kappa$, there is a set $R(j)$ of κ rows not adjacent to j, but then $\sum_{i \in R(j)} e_i^b$ and $\sum_{i \in R(j)} e_i^b + e_j^{b'}$ are both feasible decompositions that satisfy the block capacity constraint with equality, and thus $a_j^{b'} = 0$, completing the first part of the proof.

It remains to prove the converse direction. If there is some row j with $|\gamma(j)| > m - \kappa$, the inequality $\sum_{i=1}^{m} x_i^b + \sum_{b' \neq b} x_j^{b'} \leq \kappa$ is valid. But then the block capacity constraint can be obtained by summing up this inequality with $\sum_{b' \neq b} x_j^{b'} \geq 0$, and therefore it cannot be facet defining. □

3.2. Block-invariant inequalities. In this section we investigate inequalities for the matrix decomposition polytope that are invariant under block permutation. Consider for each block decomposition x the "aggregated" vector

$$z(x) := \left(\sum_{b=1}^{\beta} x_1^b, \ldots, \sum_{b=1}^{\beta} x_m^b \right) \in \mathbb{R}^m.$$

$z(x)$ records only whether the matrix rows are assigned to some block or not, but it no longer records to which block they are assigned. From a polyhedral point of view, the aggregated block decompositions give rise to an "aggregated" version of the block decomposition polytope

$$P_z(A, \beta, \kappa) := \operatorname{conv}\{z \in \mathbb{R}^m : \text{ there is } x \in P(A, \beta, \kappa) \text{ with } z = z(x)\}.$$

The aggregated polytope is interesting because any valid inequality $\sum_{i=1}^m a_i z_i \leq \alpha$ for $P_z(A, \beta, \kappa)$ can be "expanded" into an inequality $\sum_{i=1}^m a_i \sum_{b=1}^\beta x_i^b \leq \alpha$ that is valid for $P(A, \beta, \kappa)$. All inequalities in this subsection are of this type. Obviously, the expansion process yields inequalities that are invariant under block permutations, hence the name.

From a computational point of view, block-invariant inequalities are promising cutting planes, because the objective of the matrix decomposition problem can be written in terms of aggregated z-variables as $\mathbb{1}^T x = \mathbb{1}^T z(x)$. Thus, a complete description of $P_z(A, \beta, \kappa)$ would already allow us to determine the correct objective function value of the matrix decomposition problem and z-cuts will help to raise the lower bound of an LP-relaxation.

The aggregated polytope $P_z(A, \beta, \kappa)$ provides a model of the matrix decomposition problem that rules out degeneracy due to block permutations. While this is a very desirable property of the aggregated z-formulation, its drawback is that it is already $\mathcal{NP}$-complete to decide whether a given vector $z \in \{0, 1\}^m$ is an aggregated block decomposition or not. (It can be shown that this is a bin-packing problem.) Our choice to use z-cuts within the x-model tries to circumvent this difficulty and combines the strengths of both formulations. We remark that degeneracy problems of this type arise also in block-indexed formulations of grouping problems in cellular manufacturing, where the difficulty can be resolved by means of alternative formulations; see Crama and Oosten [3].

We already know one example of an expanded aggregated constraint: Expanding the inequality $z_i \leq 1$ for the aggregated block decomposition polytope yields the block assignment constraint (IP) (i) $\sum_{b=1}^\beta x_i^b \leq 1$ that we have analyzed in the previous subsection. More inequalities are derived from the observation that adjacent rows (with respect to $G(A^T)$) can only be assigned to the same block. A first example of this sort of inequalities are the *z-cover inequalities.*

THEOREM 3.5 (z-cover). *Let $G(A^T) = (V, E)$ and let $W \subseteq V$ be a set of rows of cardinality $\kappa + 1$. Then the z-cover inequality*

$$\sum_{i \in W} \sum_{b=1}^{\beta} x_i^b \leq \kappa$$

is valid for $P(A, \beta, \kappa)$ if and only if $(W, E(W))$ is connected (where $E(W)$ denotes all edges with both endpoints in W). It is facet defining for $P(A, \beta, \kappa)$ if and only if, for each row $i \notin W$, the graph $(W \cup \{i\}, E(W \cup \{i\}))$ has an articulation point different from i.

Proof. The validity part is easy. Since $|W| = \kappa + 1$, not all rows can be assigned to the same block. If some rows of W are assigned to different blocks, there must be at least one row in W that is not assigned because $(W, E(W))$ is connected. Conversely, if W is not connected, one easily finds a partition of W into two subsets that can be assigned to different blocks.

The proof that this inequality is facet defining if and only if, for each row $i \notin W$, the graph $(W \cup \{i\}, E(W \cup \{i\}))$ has an articulation point different from i is analogous to the proof of Theorem 3.4. The condition guarantees that if row i is assigned to some block, the assignment can be extended in such a way that κ rows from W can be assigned to at least two blocks. On the other hand, if the condition is not satisfied for some $j \notin W$, the inequality $\sum_{i \in W \cup \{j\}} \sum_{b=1}^{\beta} x_i^b \leq \kappa$ is valid, and thus the z-cover inequality cannot be facet defining. □

In the set covering model, z-cover inequalities correspond to constraints of the form $\sum_{i \in W} \sum_{b=1}^{\beta} y_i^b \geq 1$ that have been used by Nicoloso and Nobili [22] for their computations. The separation problem is to find a tree of size $\kappa + 1$ of maximum node weight. This problem has been studied by Ehrgott [6] and was shown to be $\mathcal{NP}$-hard using a reduction to the node-weighted Steiner-tree problem.

The z-cover inequalities are induced by trees, but it is possible to generalize them for subgraphs of higher connectivity.

THEOREM 3.6 (generalized z-cover). *Let $G(A^T) = (V, E)$ and let $W \subseteq V$ be a set of rows of cardinality $\kappa + k$ with $k \geq 1$. Then the (generalized) z-cover inequality*

$$\sum_{i \in W} \sum_{b=1}^{\beta} x_i^b \leq \kappa$$

is valid for $P(A, \beta, \kappa)$ if and only if $(W, E(W))$ is k-node connected. It is facet defining for $P(A, \beta, \kappa)$ if and only if, for each row $i \notin W$, there exists some node cut N in $(W \cup \{i\}, E(W \cup \{i\}))$ of cardinality k with $i \notin N$.

The proof of this generalization follows exactly the lines of the proof of Theorem 3.5. In our branch-and-cut algorithm we restrict attention to the cases $k = 1$ and $k = 2$.

Closely related to the z-cover inequality is the *z-clique inequality*. Here, we consider some node set W that not only is k-node connected for some fixed k but induces a complete subgraph. In this case the condition for being facet defining slightly changes.

THEOREM 3.7 (z-clique). *If Q is a clique in $G(A^T)$, then the z-clique inequality*

$$\sum_{i \in Q} \sum_{b=1}^{\beta} x_i^b \leq \kappa$$

is valid for $P(A, \beta, \kappa)$. It is facet defining if and only if $|Q| \geq \kappa + 1$ and, for each row $i \notin Q$, there exists a set of rows $R(i) \subseteq Q$, $|R(i)| = \kappa$, such that i is not adjacent in $G(A^T)$ to any node in $R(i)$.

Proof. The inequality is clearly valid. To show that it is facet defining given the mentioned conditions, let $a^T x \leq \alpha$ define a facet such that

$$\left\{ x \in P(A, \beta, \kappa) | \sum_{i \in Q} \sum_{b=1}^{\beta} x_i^b = \kappa \right\} \subseteq \{x \in P(A, \beta, \kappa) | a^T x = \alpha\}.$$

We will show that the two inequalities are the same up to a positive scalar multiplicative factor. To this purpose, consider any κ rows of Q. The block decomposition obtained by assigning these rows to some block b is feasible and tight for the z-clique inequality. Since $|Q| \geq \kappa + 1$, we can use these solutions to show that $a_i^b = a_j^{b'}$ for all $i, j \in Q$ and for all blocks $b, b' \in \{1, \ldots, \beta\}$. Assuming that for each row $i \notin Q$ there exists a set of nodes $R(i) \subseteq Q$, $|R(i)| = \kappa$, that are not adjacent to i, we observe that for all $b' \neq b$, the vectors $\sum_{j \in R(i)} e_j^b$ and $\sum_{j \in R(i)} e_j^b + e_i^{b'}$ are valid block decompositions that satisfy the z-clique inequality with equality. It follows that $a_i^{b'} = 0$ for all

$i \notin Q$, for all $b' \neq b$, and even for all blocks b', since b was arbitrary. This completes the first part of the proof.

If, on the other hand, Q has size less than or equal to κ, we obtain from (IP) (i) that the left-hand side of the inequality is at most $|Q|$. Thus, the inequality is redundant and cannot define a facet. Now suppose the second condition is not satisfied, i.e., there is some $j \notin Q$ such that j is incident to at least $|Q| - \kappa + 1$ nodes in Q. This implies that $Q \cup \{j\}$ is at least $(|Q| - \kappa + 1)$-node connected. Theorem 3.6 states that $\sum_{i \in Q \cup \{j\}} \sum_{b=1}^{\beta} x_i^b \leq \kappa$ is valid and this implies that the z-clique inequality is redundant. □

The z-clique separation problem is again a max-clique problem and thus is $\mathcal{NP}$-hard. In our implementation we check easily detectable special cases like the so-called *big-edge inequalities*

$$\sum_{i \in \operatorname{supp}(A_{\cdot j})} \sum_{b=1}^{\beta} x_i^b \leq \kappa$$

for all columns j, where $A_{\cdot j}$ denotes the jth column of A and $\operatorname{supp}(A_{\cdot j})$ denotes its nonzero row indices. These inequalities can be separated by inspection.

Another way to generalize the z-cover inequalities is by looking at node induced subgraphs that consist of several components. This idea, which gives rise to the class of *bin-packing inequalities*, came up in our computational experiments. The starting point is again a set of rows W that induces a subgraph of $G(A^T) = (V, E)$. Suppose $(W, E(W))$ consists of l connected components of sizes (in terms of nodes) $a_1, \ldots, a_l$. We can then associate a *bin-packing problem* with $(W, E(W))$, β, and κ in the following way: There are l items of sizes $a_1, \ldots, a_l$, and β bins of capacity κ each. The problem is to put all the items into the bins such that no bin holds items of a total size that exceeds the capacity κ. If this is not possible, we can derive a valid inequality for $P(A, \beta, \kappa)$.

Theorem 3.8. *Let $G(A^T) = (V, E)$ and $W \subseteq V$ be some subset of rows. If the bin-packing problem associated with $(W, E(W))$, β, and κ has no solution, the bin-packing inequality*

$$\sum_{i \in W} \sum_{b=1}^{\beta} x_i^b \leq |W| - 1$$

is valid for $P(A, \beta, \kappa)$.

Proof. Consider some block decomposition x. If at least one row in W is not assigned to some block, the inequality is obviously satisfied. Otherwise all rows that belong to the same (connected) component of $(W, E(W))$ must be assigned to the same block. This yields a solution to the bin-packing problem associated with $(W, E(W))$, β, and κ, a contradiction. □

We do not know any reasonable conditions that characterize when the bin packing inequalities are facet defining. Bin-packing separation is $\mathcal{NP}$-hard; see Garey and Johnson [11].

Next we give another class of *z-cycle inequalities* that generalize the cycle inequalities of the set packing polytope.

Theorem 3.9 (z-cycle). *Let $G(A^T) = (V, E)$ and $C \subseteq V$ be a cycle in $G(A^T)$ of* *cardinality at least $\kappa + 1$. Then the z-cycle inequality*

$$\sum_{i \in C} \sum_{b=1}^{\beta} x_i^b \le |C| - \left\lceil \frac{|C|}{\kappa + 1} \right\rceil$$

is valid for $P(A, \beta, \kappa)$.

The z-cycle inequality is valid because at least every $(\kappa + 1)$st node cannot be assigned to a block. One can also show that the inequality is facet defining for its support under certain rather restrictive conditions—for example, if C is an odd hole, $|C| \neq 0 \bmod (\kappa + 1)$, and the right-hand side is less than $\beta\kappa$. z-Cycle separation can be reduced to the TSP and is thus $\mathcal{NP}$-hard.

Our next class of inequalities comes up in several instances in our test set.

THEOREM 3.10 (composition of cliques (COQ)). *Let $G(A^T) = (V, E)$ and consider p mutually disjoint cliques $Q_1, \ldots, Q_p \subseteq V$ of size q, and q mutually disjoint cliques $P_1, \ldots, P_q \subseteq V$ of size p, such that $|P_i \cap Q_j| = 1$ for all i, j. Let $W = \cup_{i=1}^{p} Q_i$. Then the following inequality is valid for $P(A, \beta, \kappa)$:*

$$\sum_{i \in W} \sum_{b=1}^{\beta} x_i^b \le \max_{\left\{\substack{r \in \mathbb{N}^\beta, s \in \mathbb{N}^\beta: \\ \sum r_b = p, \sum s_b = q}\right\}} \sum_{b=1}^{\beta} \min\{\kappa, r_b s_b\} =: \alpha(p, q, \beta, \kappa). \tag{3.2}$$

Proof. Consider a block decomposition x and let

$$\begin{aligned} \overline{r}_b &:= \left| \left\{ j : \sum_{i \in Q_j} x_i^b \ge 1, j \in \{1, \ldots, p\} \right\} \right|, \\ \overline{s}_b &:= \left| \left\{ j : \sum_{i \in P_j} x_i^b \ge 1, j \in \{1, \ldots, q\} \right\} \right| \end{aligned}$$

for $b = 1, \ldots, \beta$. Because Q_j and P_j are all cliques, we have that $\sum_{b=1}^{\beta} \overline{r}_b \le p$ and $\sum_{b=1}^{\beta} \overline{s}_b \le q$. Since $|P_i \cap Q_j| = 1$ for all i, j it follows that $\sum_{i \in W} x_i^b \le \overline{r}_b \overline{s}_b$. Thus,

$$\begin{aligned} \sum_{i \in W} \sum_{b=1}^{\beta} x_i^b &= \sum_{b=1}^{\beta} \sum_{i \in W} x_i^b \\ &\le \sum_{b=1}^{\beta} \min\{\kappa, \overline{r}_b \overline{s}_b\} \\ &\le \max_{\left\{\substack{r \in \mathbb{N}^\beta, s \in \mathbb{N}^\beta: \\ \sum r_b \le p, \sum s_b \le q}\right\}} \sum_{b=1}^{\beta} \min\{\kappa, r_b s_b\} \\ &= \max_{\left\{\substack{r \in \mathbb{N}^\beta, s \in \mathbb{N}^\beta: \\ \sum r_b = p, \sum s_b = q}\right\}} \sum_{b=1}^{\beta} \min\{\kappa, r_b s_b\}, \end{aligned}$$

showing the statement. □

The right-hand side of (3.2) is quite complicated, and we do not even know whether it can be computed in polynomial time. For $\beta = 2$ the right-hand side looks more tractable:

$$\sum_{i\in W}\sum_{b=1}^{\beta} x_i^b \leq \max_{\substack{r=0,\dots,p\\ s=0,\dots,q}} \left(\min\{\kappa, rs\} + \min\{\kappa, (p-r)(q-s)\}\right). \tag{3.3}$$

However, we do not know a closed formula in this case either. An interesting special case is $p = 2$. Here the graph $(W, E(W))$ consists of two disjoint cliques that are joint by a perfect matching. Suppose further that $q < \kappa < 2q$. Then the right-hand side of (3.3) reads

$$\begin{aligned}&\max\{0, \max_{s=0,\dots,q}(\min\{\kappa, s\} + \min\{\kappa, q-s\}), \min\{\kappa, 2q\}\}\\ &= \max\{0, q, \kappa\} = \kappa.\end{aligned}$$

In this case (3.2) turns out to be even facet defining if we require in addition that each node $i \notin W$ has at most $2q - \kappa$ neighbors in W, i.e., $|\gamma(i) \cap W| \leq 2q - \kappa$.

The development of our heuristic separation routine for COQ inequalities resulted in a slight generalization of this class. The support graphs of the left-hand sides of these *extended composition of clique inequalities* are COQs where some nodes have been deleted; the right-hand sides are left unchanged.

THEOREM 3.11 (extended composition of cliques (xCOQ)). *Let $G(A^T) = (V, E)$, and consider p mutually disjoint nonempty cliques $Q_1, \dots, Q_p \subseteq V$ of size at most q, and q mutually disjoint nonempty cliques $P_1, \dots, P_q \subseteq V$ of size at most p, such that*

(i) *$|P_i \cap Q_j| \leq 1$ for all i, j, and*

(ii) *$\sum_{i=1}^{q}\sum_{j=1}^{p} |P_i \cap Q_j| = \sum_{i=1}^{q} |P_i| = \sum_{j=1}^{p} |Q_j|$;*

i.e., every element in one of the sets P_i appears in exactly one of the sets Q_j, and vice versa. Let $W = \cup_{i=1}^{p} Q_i$. Then the following inequality is valid for $P(A, \beta, \kappa)$:

$$\sum_{i\in W}\sum_{b=1}^{\beta} x_i^b \leq \alpha(p, q, \beta, \kappa). \tag{3.4}$$

Proof. The proof works by turning $P_1, \dots, P_q$ and $Q_1, \dots, Q_p$ into a proper COQ by adding some nodes that correspond to "artificial rows" and projecting the resulting inequality down to the original space of variables.

Let

$$\delta := \sum_{i=1}^{q}\sum_{j=1}^{p}(1 - |P_i \cap Q_j|)$$

be the number of nodes that "miss" to turn $P_1, \dots, P_q$ and $Q_1, \dots, Q_p$ into a COQ and add a row $\mathbb{1}^T$ of all ones to A for each of them to obtain a matrix $\bar{A}$ such that

$$\bar{A}_{i\cdot} = A_{i\cdot}, \quad i = 1, \dots, m \qquad \text{and} \qquad \bar{A}_{i\cdot} = \mathbb{1}^T, \quad i = m+1, \dots, m+\delta.$$

Consider the matrix decomposition problem $(\bar{A}, \beta, \kappa)$. Its row intersection graph $G(\bar{A}^T)$ contains $G(A^T)$ as a subgraph, and the additional artificial nodes in $G(\bar{A}^T)$ are incident to every node of $G(\bar{A}^T)$ that corresponds to a row that is not all zero (except itself).

The sets $P_1, \dots, P_q$ and $Q_1, \dots, Q_p$ are again cliques in $G(\bar{A}^T)$. Associating each of the artificial nodes $i = m+1, \dots, m+\delta$ with a different index pair ij such that

$|P_i \cap Q_j| = 0$ and adding this node to both P_i and Q_j, we can extend $P_1, \ldots, P_q$ and $Q_1, \ldots, Q_p$ to a COQ $\overline{P}_1, \ldots, \overline{P}_q$ and $\overline{Q}_1, \ldots, \overline{Q}_p$ in $G(\bar{A}^T)$ with $\overline{W} := \cup_{j=1}^{p} \overline{Q}_j = W \cup \{m+1, \ldots, m+\delta\}$. Then the COQ inequality

$$\sum_{i \in \overline{W}} \sum_{b=1}^{\beta} x_i^b \leq \alpha(p, q, \beta, \kappa) \tag{3.5}$$

is valid for $P(\bar{A}, \beta, \kappa)$ and, of course, also for

$$P(\bar{A}, \beta, \kappa) \cap \{x_i^b = 0 : \ i = m+1, \ldots, m+\delta, \ b = 1, \ldots, \beta\}.$$

Since the artificial variables in this polytope attain only values of zero, this remains true if one sets their coefficients in (3.5) also to zero, but as this results in the desired extended COQ inequality (3.4) and

$$P(\bar{A}, \beta, \kappa) \cap \{x_i^b = 0 : \ i = m+1, \ldots, m+\delta, \ b = 1, \ldots, \beta\} = P(A, \beta, \kappa) \times \{0\}^{\delta \times \beta},$$

the theorem follows by a projection on the space of the original variables. □

The last *star inequality* that we present in this section is special in the sense that it is the only one with non-0/1 coefficients. It was designed to deal with rows with many neighbors.

THEOREM 3.12 (star). *Let $G(A^T) = (V, E)$ and consider some row $i \in V$ with $|\gamma(i)| > \kappa$. Then the star inequality*

$$(|\gamma(i)| - \kappa + 1) \sum_{b=1}^{\beta} x_i^b + \sum_{j \in \gamma(i)} \sum_{b=1}^{\beta} x_j^b \leq |\gamma(i)|$$

is valid for $P(A, \beta, \kappa)$.

Proof. If i is assigned to some block b, then all rows in $\gamma(i)$ can be assigned only to b, but at most $\kappa - 1$ of them. The case where i is not assigned is trivial. □

The star inequality can be viewed as a lifting of the (redundant) inequality

$$\sum_{j \in \gamma(i)} \sum_{b=1}^{\beta} x_j^b \leq |\gamma(i)|,$$

and we want to close this section with another simple lifting theorem for block-invariant inequalities with 0/1 coefficients.

THEOREM 3.13 (strengthening). *Let $G(A^T) = (V, E)$, W be a subset of V, and $\sum_{i \in W} \sum_{b=1}^{\beta} x_i^b \leq \alpha$ be a valid inequality for $P(A, \beta, \kappa)$. If, for some row $j \notin W$, the condition*

$$|W \setminus \gamma(j)| + \kappa \leq \alpha$$

holds, then $\sum_{i \in W \cup \{j\}} \sum_{b=1}^{\beta} x_i^b \leq \alpha$ is also valid for $P(A, \beta, \kappa)$.

Proof. If j is assigned to some block b, the rows in $\{j\} \cup \gamma(j)$ can be assigned only to b, but at most κ of them. □

4. A branch-and-cut algorithm.

The polyhedral investigations of the last section form the basis for the implementation of a branch-and-cut algorithm for the solution of the matrix decomposition problem. This section describes the four main ingredients of this code: separation and LP-management, heuristics, problem reduction, and searchtree management.

4.1. Separation and LP-management. We use all of the inequalities described in section 3 as cutting planes in our algorithm. It will turn out that some of them appear in large numbers. We thus opted for the following separation strategy: We try to identify many inequalities using fast heuristics but add only selected ones to the LP. More expensive separation methods are used depending on their success.

We classify our separation routines according to their basic algorithmic principles: inspection (enumeration), greedy heuristics, other heuristics, exact polynomial methods, and "hybrid" methods (combinations of exact and heuristic methods).

The most simple methods are used for *big-edge*, *two-partition*, and *star inequalities*. These classes can be separated by simple *inspection*; the details for two-partition inequalities were already described in section 3.

Clique, *z-clique*, and *z-cover inequalities* are separated using *greedy heuristics*. In the last two cases, these methods start by sorting the rows of A with respect to increasing z-value, i.e., such that $z(x)_{i_1} \geq z(x)_{i_2} \geq \cdots \geq z(x)_{i_m}$. Then the greedy heuristic is called m times, once for each row i_j. In each call, i_j is used to initialize a clique/tree with respect to $G(A^T)$ that is iteratively extended greedily in the order of the z-sorting of the rows until z_{i_j} becomes zero and the growing procedure stops. There is also a second variant for z-cliques that is an adaptation of a similar routine by Hoffman and Padberg [17]. Here we call the greedy heuristic once for each column of A and initialize the clique with the support of this column. Having detected a violated clique inequality in one of these ways, we lift randomly determined additional rows with zero z-value sequentially into the inequality. This is done by applying the strengthening procedure of Theorem 3.13, which in this case amounts to a further growth of the clique by randomly determined rows of zero z-value. We tried to strengthen cover inequalities, but the computational effort was not justified by the one or two coefficients that were usually lifted, but, as was already mentioned in section 3, we (heuristically) keep track of the connectivity of the growing graph. If the connectivity is 2 after the graph reached size $\kappa + 1$, we add another two-connected node if possible. Figure 4.1 gives more detailed pseudocode for z-cover separation. Separation of clique inequalities is done using exactly the same routine as for z-cliques but applied to $G_c(A, \beta)$ with node weights given by the x-variables.

z-Cycle inequalities are separated in the following heuristic way. We look at some path P with end-nodes u and v, where initially u and v coincide. In each iteration we extend the path at one of its end-nodes by a neighbor w with maximal $z(x)_w$-value. Let j_w be a column of A that connects w to the path P. Since j_w forms a clique in $G(A^T)$, there are additional nodes that can be potentially added to the path if the support of j_w is greater than two, i.e., $|\operatorname{supp}(A_{\cdot j_w})| > 2$. We store these additional nodes in a buffer which will be exploited later in the heuristic. Now we test whether the new path P extended by w can be closed to a cycle C that satisfies $|C| > \kappa$ and $|C| \neq 0 \bmod (\kappa + 1)$. This is done by looking for a column j of A that contains both end-nodes of P (one of them w). $\operatorname{supp}(A_{\cdot j})$ again forms a clique, and the additional nodes in this clique, together with the nodes in the buffer, give the flexibility to add further nodes to the cycle. This freedom is exploited in our routine. We try the procedure for several starting nodes $u = v$, whose number depends on the success of the heuristic.

Separation of the *composition of clique* inequalities is not easy: We do not even know a way to compute the right-hand side $\alpha(p, q, \beta, \kappa)$ in polynomial time! But there are problems in our test set, e.g., `pipex` (see section 5.2), where compositions of cliques occur and there seems to be no way to solve this (small!) problem without

```
z-cover separation
Input: $z(x) \in \mathbb{R}^m$, $G(A^T)$
Output: Node set $T$ of a one- or two-connected subgraph of $G(A^T)$

begin
    sort $z(x)$ such that $z(x)_{i_1} \geq z(x)_{i_2} \geq \cdots \geq z(x)_{i_m}$;
    for $k := 1$ to $m$
        $T \leftarrow \{i_k\}$;
        connectivity $\leftarrow 2$;
        for $j := 1$ to $m$
            if $j = k$ or $\gamma(i_j) \cap T = \emptyset$    continue;
            if $|T| = \kappa + 1$ and $|\gamma(i_j) \cap T| = 1$    continue;
            if $|T| \geq 2$ and $|\gamma(i_j) \cap T| = 1$    connectivity $\leftarrow 1$;
            $T \leftarrow T \cup \{i_j\}$;
            if $|T| \geq \kappa +$ connectivity    break;
        endfor;
    endfor;
    return $T$ and connectivity;
end;
```

FIG. 4.1. *Separating z-cover inequalities with a greedy heuristic.*

them. Our heuristic was developed to capture these cases. It led to the development of the more general class of extended COQ inequalities, which are easier to find. The idea is as follows.

Let us start with a composition of cliques $Q_1, \dots, Q_p$ and $P_1, \dots, P_q$ as stated in Theorem 3.10. Suppose that these cliques are contained in the columns $1, \dots, p, p+1, \dots, p+q$ of the matrix A, i.e., $\operatorname{supp}(A_{\cdot i}) \supseteq Q_i$, $i = 1, \dots, p$, and $\operatorname{supp}(A_{\cdot i+p}) \supseteq P_i$, $i = 1, \dots, q$. Consider a *column/column-incidence matrix* S of A defined by

$$s_{ij} = \begin{cases} k & \text{for } k \in \{l : \ a_{li} \neq 0 \neq a_{lj}\} \text{ arbitrary, but fixed,} \\ 0 & \text{if } A_{\cdot i}^T A_{\cdot j} = 0, \end{cases}$$

i.e., $s_{ij} = k \neq 0$ if and only if columns i and j intersect in some row k, and in case there is no unique k we pick an arbitrary but fixed one. Suppose for the moment that all entries in the submatrix $S_{\{1,\dots,p\}\times\{p+1,\dots,p+q\}}$ of S are mutually different; that is, there is no row index k that appears more than once. Then the composition of cliques corresponds to the rectangle submatrix $S_{\{1,\dots,p\}\times\{p+1,\dots,p+q\}}$ of S that is completely filled with nonzeros: The rows that appear on the left-hand side of the COQ inequality (3.2) are exactly those appearing in the matrix $S_{\{1,\dots,p\}\times\{p+1,\dots,p+q\}}$. In other words, the node set W in (3.2) is $W = \{s_{ij} : \ i = 1, \dots, p, \ j = p+1, \dots, p+q\}$. Thus, given some vector $x \in \mathbb{R}^{m \times \beta}$, the left-hand side of (3.2) is $\sum_{i=1}^{p} \sum_{j=p+1}^{p+q} \sum_{b=1}^{\beta} x_{s_{ij}}^b$ and a final calculation of the right-hand side allows one to check for a possible violation of the inequality.

Our heuristic tries to go in the reverse direction: It identifies large filled rectangles in S and derives COQ and xCOQ inequalities from them. There are three difficulties. First, a clique in a composition can be not only a subset of a column of A but any clique in $G(A^T)$. However, we have not incorporated this generality in our heuristic, because we do not know how to select a promising set of cliques in $G(A^T)$. Second, columns in A that form a composition of cliques may not appear in the right order: The rectangle identifies itself only after a suitable permutation of S. In this case, we have to reorder the columns and rows of S. We obtain a filled rectangle submatrix $S_{I\times J}$ of S by starting with each of column j of S once, extend this $1 \times |\operatorname{supp}(A_{\cdot j})|$ rectangle submatrix by columns that fit best in a greedy way, sort its rows lexicographically, and consider all maximal filled submatrices in the upper left corner as potential COQ-rectangles. A third serious problem arises when two columns of A intersect in more than one row. In this case the entries of the matrix S are no longer uniquely determined, and it can happen that the entries of the rectangular submatrix $S_{I\times J}$ under consideration are no longer mutually different. Then $S_{I\times J}$ corresponds no longer to a composition of cliques, and the inequality $\sum_{ij\in I\times J}\sum_{b=1}^{\beta} x^b_{s_{ij}} \le \alpha\big(|I|,|J|,\beta,\kappa\big)$ is in general not valid. However, one can set duplicate entries in S to zero until, for every row k, there is only one representative $s_{ij} = k$ left; denote the resulting matrix by S'. Then the sets

$$\overline{Q}_i := \{s'_{ij} : \; s'_{ij} \neq 0, \; j \in J\}, \quad i \in I \qquad \text{and} \qquad \overline{P}_j := \{s'_{ij} : \; s'_{ij} \neq 0, \; i \in I\}, \quad j \in J,$$

of nonzero entries in the rows and columns of S' form an extended composition of cliques, and the corresponding xCOQ inequality

$$\sum_{k\in \operatorname{im} S_{I\times J}} \sum_{b=1}^{\beta} x^b_k \le \alpha(|I|,|J|,\beta,\kappa)$$

is valid for $P(A,\beta,\kappa)$, where $\operatorname{im} S_{I\times J} = \{s_{ij} : \; ij \in I \times J\}$ denotes the set of row indices that appear in the submatrix $S_{I\times J}$. The interesting feature of separating extended COQ inequalities instead of COQs is that the generalization gives us the algorithmic freedom to handle multiple occurrences of rows in filled rectangles of S, and this is the key to a successful heuristic separation of an otherwise rigid structure. The price for this, of course, is a reduced support in the left-hand side. To pay this price only when necessary, we heuristically determine a column/column-intersection matrix S with a large variety of rows in $\operatorname{im} S$. The right-hand side itself is computed in amortized (pseudopolynomial) time of $O(\beta\kappa n^2)$ steps by a dynamic program (for our tests $\beta \le 4$ and $\kappa = O(n)$, and thus this effectively amounts to $O(n^3)$).

The reader might have noticed that several degrees of freedom in this separation routine can be used to search for rectangles with large z-value, and this is what we would like to find. However, the running time of the method is too large to apply it after each LP, and when we did, we did not find additional cuts. We thus call the routine only once, determine some promising COQs by combinatorial criteria, store them in memory, and separate them by inspection.

To separate *clique inequalities* (for $\beta > 2$), we use an *exact branch-and-bound algorithm* for the maximum weight clique problem. Although in principle exponential, this algorithm works fast for the separation problems coming up in our matrix decomposition instances because the maximum clique size is bounded by β. We have also tried to separate z-cliques exactly, but we never observed that additional cuts were found: In the small examples, the greedy heuristic is good enough, while in the larger ones with high capacities, cliques of size κ don't seem to exist. Another exact, but this time polynomial, algorithm is used to separate *cycle inequalities*: We apply the odd-cycle algorithm described in Lemma 9.1.11 in Grötschel, Lovász, and Schrijver [13].

Finally, a mixture of exact and heuristic ideas is used in a hybrid algorithm to separate the *bin-packing inequalities.* We start by determining a node set W that can result in a violated inequality. A necessary condition for this is

$$\sum_{i\in W} z(x)_i > |W| - 1 \iff 1 > \sum_{i\in W} (1 - z(x)_i),$$

and it is reasonable to construct W by iteratively adding rows that have a z-value close to one. We thus sort the nodes with respect to increasing z-value and add them to W in this order as long as the condition stated above is satisfied. This node set W induces a subgraph $(W, E(W))$ of $G(A^T)$, and we determine the components of this subgraph. The resulting bin-packing problem (see above Theorem 3.8) is solved using an exact dynamic programming algorithm (with a time bound).

In addition to these classical types of cutting planes we also use a number of *"tie-breaking" inequalities* to cut off decompositions that are identical up to block permutations or give rise to multiple optima for other reasons as a means to counter dual degeneracy and stalling. These inequalities are in general not valid for $P(A, \beta, \kappa)$ but are valid for at least one optimal solution. The simplest kind of these cuts are the *permutation inequalities*

$$\sum_{i=1}^{m} x_i^b \le \sum_{i=1}^{m} x_i^{b+1}, \quad b = 1, \dots, \beta - 1,$$

stating that blocks with higher indices are of larger size. To break further ties, we supplement them with inequalities stipulating that, in case of equal sized blocks, the row with the smallest index will be assigned to the block with smaller index. These *strengthened permutation inequalities* read

$$x_k^{b+1} + \sum_{i=1}^{m} x_i^b - \sum_{i=1}^{m} x_i^{b+1} \le \sum_{i=0}^{k-1} x_i^b, \quad b = 1, \dots, \beta - 1,\ k = 2, \dots, m - 1.$$

If $\sum_{i=1}^m x_i^b - \sum_{i=1}^m x_i^{b+1} < 0$, the inequality is redundant, but in case of equality, the row with the smallest index in blocks b and $b+1$ must be in block b. The case $k = m$ is left out because it yields a redundant inequality. Both permutation and strengthened permutation inequalities can be separated by *inspection.*

Another idea that we use to eliminate multiple optima is based on the concept of *row preference.* We say that row i is *preferred* to row j or, in symbols, $i \prec j$, if

$$\gamma(i) \subseteq \gamma(j)$$

with respect to the row intersection graph $G(A^T)$. In this situation we may not know whether or not row i or j can be assigned to a block in some optimal solution, but we can say that for any decomposition x with $z(x)_j = 1$ (say, $x_j^b = 1$) either $z(x)_i = 1$ or we can get a feasible decomposition $x' = x - e_j^b + e_i^b$ with the same number of rows assigned. In this sense, row i is more attractive than row j. If we break ties on row preference by indices (i.e., $i \prec j \iff \gamma(i) \subsetneqq \gamma(j) \vee (\gamma(i) = \gamma(j) \wedge i < j)$), row preferences induce a partial order that we represent in a transitive and acyclic digraph

$$D(A) := (V, \{(i, j) : i \prec j\}).$$

Since the number of row preferences tends to be quadratic in the number of rows, we thin out this digraph by removing all transitive (or implied) preferences. The remaining row preferences are forced in our code by adding the *row preference inequalities*

$$\sum_{b=1}^{\beta} x_i^b \geq \sum_{b=1}^{\beta} x_j^b \quad \text{for } (i,j) \text{ with } i \prec j.$$

These can sometimes be strengthened to

$$x_i^b \geq x_j^b \quad \text{for all } b = 1, \ldots, \beta$$

if we can be sure that rows i and j cannot be assigned to different blocks in any decomposition. This will be the case, for example, if i and j are adjacent in $G(A^T) = (V, E)$ or if both i and j are adjacent to some third row k preferable to both of them (i.e., $i \prec j$, $k \prec i$, $k \prec j$, $ik \in E$ and $jk \in E$). Once $D(A)$ is set up, row preference inequalities can be separated by *inspection*.

Our last separation routine uses a *cut pool* that stores all inequalities found by the hitherto explained algorithms: The *pool separation* routine just checks all inequalities in the pool for possible violation.

The separation algorithms described in the previous paragraphs turned out to be very successful: Not only block-discernible (permutable) inequalities like two-partitions are found in large numbers but also block-invariant cuts like z-covers occur in abundance. Controlling the growth of the LP-relaxation is thus the main goal of our *separation and LP-maintenance strategy.* We start with a minimal LP-relaxation containing (besides the bounds) only the block assignment and block capacity constraints plus the $\beta - 1$ permutation inequalities. The cuts that are separated by inspection, i.e., big-edge inequalities, star inequalities, tie-breaking inequalities, and composition of clique inequalities are placed in a *cut pool*; they will be found by pool separation. The separation algorithms are called dynamically throughout the course of the branch-and-cut algorithm. After an LP is solved, we call the pool separation routine, followed by two-partition inequality separation and a couple of heuristics: The z-cover heuristic is called as it is, but application of the more expensive z-clique and z-cycle algorithms is controlled by a simple time- and success-evaluation. This control mechanism is motivated by the observation that our test set fell into two groups of examples, where one of these routines either was indispensable or essentially did not find a single cut. We empirically try to adapt to these situations by calling the separation routines only if their past success is proportional to the running time or, more precisely, if after the first call

$$\frac{\text{\# of successful calls} + 1}{\text{\# of calls}} > \frac{\text{time spent in routine}}{\text{total time spent in separation}}.$$

A call is counted as successful if a violated cut is found. If $\beta > 2$, there can be clique inequalities that are not two-partition constraints, and in this case we next call the exact clique separation routine that returns at most one cut. The branch-and-bound algorithm used there turned out to be fast enough to be called without any further considerations. Finally, we separate bin-packing inequalities. To avoid excessive running times due to the dynamic program, the routine is called with a time limit: The dynamic program will be stopped if the time spent in bin-packing separation exceeds the cumulated separation time of all other separation routines.

All violated cuts determined in this separation process are not *added* directly to the LP-relaxation but stored in a *cut buffer* first. This buffer is saved to the pool, and then a couple of promising cuts are selected to strengthen the LP-relaxation. Our criteria here have an eye on the amount of violation and on the variety of the cuts. Since inequalities of one particular type tend to have similar support, we restrict the number of cuts per type and prefer to add inequalities of other types, even if they are not among the most violated. To accomplish this we add the

$$\frac{\text{\# of cuts in cut buffer}}{\text{\# number of types of cuts}}$$

most violated cuts of each type to the LP-relaxation. We also *delete* cuts from the LP-relaxation if they become nonbinding by a slack of at least 10^{-3}, but we keep them in the cut pool for a possible later pool separation.

Another feature of our code that aims for small LPs is to *locally set up* the LP-relaxation prior to computation at any node of the searchtree. This means that when branching on some node v, we store at each of its sons a description of the last LP solved at v and of the optimal basis obtained. When we start to process v's sons, we set up this LP from scratch and load the associated (dual feasible) basis. In this way, we continue the computation exactly at the point where it stopped, and the LP will be the result of a contiguous process independent of the node selection strategy. We have compared this approach to one where the start-LP at each newly selected node is just the last LP in memory, and this leads to larger LPs and larger running times.

While these strategies were sufficient to keep the size of the LP-relaxation under control, explosive growth of the cut pool was a serious problem in our computations until we implemented the following *cut pool management*. We distinguish between disposable and indisposable cuts in the pool. *Indisposable cuts* are inequalities that are needed to set up the LP-relaxation at some node in the searchtree yet to be processed and all big-edge, star, and tie-breaking inequalities. All other cuts are *disposable* and can potentially be deleted from the cut pool, possibly having to be recomputed later. In order to control the pool size we restrict the number of disposable cuts in the pool by eliminating cuts that have not been in any LP for a certain number of iterations. This number depends on the size of the pool and the ratio of disposable to indisposable cuts.

The LPs themselves are solved with the `CPLEX 4.0` dual simplex algorithm using steepest edge pricing; see the `CPLEX` documentation [18].

4.2. Heuristics. Raising the lower bound using cutting planes is one important aspect in a branch-and-cut algorithm; finding good feasible solutions early to enable fathoming of branches of the searchtree is another, and we have implemented several primal heuristics for our matrix decomposition code. Since different nodes in a branch-and-bound tree correspond to different fixings of variables to zero or one, the heuristics should respect these fixings to increase the probability of finding different solutions. Applied at the root node where (at least initially) no variables are fixed, our methods can be seen as LP-based or pure combinatorial heuristics for the matrix decomposition problem.

Our heuristics fall into three groups: "primal" methods that iteratively fix block assignments, "dual" methods that iteratively exclude assignments until decisions become mandatory due to a lack of alternatives, and an improvement method that is applied as an "afterburner" to enhance the quality of the two groups of opening heuristics.

The *primal methods* consist of a *greedy algorithm* and a *bin-packing heuristic*; both are LP-based. The greedy algorithm starts by ordering the x_i^b variables. With probability $\frac{1}{2}$, a random ordering is chosen; otherwise a sorting according to increasing x-value is used. The rows are assigned greedily to the blocks in this order. This heuristic is similar in spirit to the popular "LP-plunging" method, i.e., the iterative rounding of some fractional LP-value to an integer followed by an LP-reoptimization, but much faster. We have also tried LP-plunging, but for the matrix decomposition problem the results were not better than with the simple greedy method, while the running time was much larger. The bin-packing heuristic starts by determining a set of nodes W that will be assigned to the blocks and used to set up a corresponding bin-packing problem. In order to find a better decomposition than the currently best known with, say, $z^\star$ rows assigned, W should be of cardinality at least $z^\star + 1$, and therefore we take the $z^\star + 1$ rows with the largest $z(x)$-values to be the members of W. The corresponding bin-packing problem is set up and solved with the same dynamic program that we used for the separation of the bin-packing inequalities; it is also called with a time limit, namely, 10 times as much as all other primal heuristics (which are very fast) together. Clearly, we also watch out for better solutions that might be detected in bin-packing separation.

The *dual methods* also respect variable fixings but are not LP-based. The idea behind them is not to assign rows to blocks but to iteratively eliminate assignments of "bad" rows. Suppose that a decision was made to assign certain rows (assigned rows) to certain blocks and to exclude other rows from assignment (unassigned rows), while for the remaining rows a decision has yet to be made (free rows). Removing the unassigned nodes from the row intersection graph $G(A^T)$ leaves us with a number of connected components, some of them larger than the maximum block capacity κ, some smaller. Both variants of the dual method will break up the components that are larger than the block capacity κ by unassigning free rows until no more such components exist. At this point, a simple first-fit decreasing heuristic is called to solve the corresponding bin-packing problem. The two variants differ in the choice of the next bad row to remove. Variant I chooses the free row in some component of size larger than κ with the largest degree with respect to the row intersection graph $G(A^T)$; variant II excludes assignment of a free row of some component with size larger than κ with the largest indegree with respect to $D(A)$, or, in other words, the least preferable row. We have also tried to use a dynamic program to solve the bin-packing problems, but it did not provide better results in our tests.

Our *improvement heuristic* is a variation of a local search technique presented by Fiduccia and Mattheyses [9]. Given some block decomposition, it performs a sequence of local exchange steps each of the following type. Some assigned row is chosen to be made unassigned, opening up possibilities to assign its unassigned neighbors. These assignments are checked and feasible assignments are executed. The details are as follows. The heuristic performs a number of passes (10 in our implementation). At the beginning of each pass, all rows are eligible for unassignment in the basic exchange step. Each row may be selected only once for unassignment in each pass and will then be "locked." Candidates for becoming unassigned are all currently assigned and unlocked rows. These candidates are rated according to the number of possible new assignments (computed heuristically) and we choose the one that is best with respect to this rating. As a special annealing-like feature, the algorithm will also perform the exchange step if it leads to a change of the current solution to the worse. If no exchange step is possible because all assigned rows are already locked, the pass ends and the next pass is started.

The *strategy to call the heuristics* is as follows. The primal methods are called after each individual LP, whereas the dual heuristics are called only once at each node in the branch-and-bound tree, because they behave in a different way only due to changes in the variable fixings.

4.3. Problem reduction. We use a couple of problem reduction techniques to eliminate redundant data. First we apply an *initial preprocessing* to the matrix A before the branch-and-cut algorithm is initiated. The purpose of this preprocessing step is to eliminate columns from the matrix A without changing the row intersection graph. We first perform a couple of straightforward tests to identify columns that are contained in other columns and can thus be deleted: We remove empty columns, then unit columns, duplicate columns, and finally, by enumeration, columns that are contained in others.

These simple initial preprocessing steps are amazingly effective, as we will see in the section on computational results. In principle, the number of rows can be reduced also. For example, empty rows could be eliminated and later used to fill excess capacity in any block, and duplicate rows or rows with just one nonzero entry could be eliminated by increasing the capacity requirements of one of its adjacent rows. These reductions, however, led to changes in the IP model and affect all separation routines discussed so far, so that we refrained from implementing them.

In addition to this initial preprocessing we do *local fixings* at the individual nodes of the branch-and-bound searchtree after each call to the LP-solver. Apart from *reduced cost fixing* and *fixing by logical implication* (i.e., if x_i^b is fixed to one, $x_i^{b'}$ will be fixed to zero for all blocks $b' \neq b$), we try to identify rows that cannot be assigned to any block given the current state of fixings. To this purpose we look at all rows W that are currently fixed for assignment to some block. We then check for each unassigned row i whether the subgraph $(W \cup \{i\}, E(W \cup \{i\}))$ of $G(A^T) = (V, E)$ contains a component with more than κ rows. If so, row i can be fixed to be unassigned.

4.4. Searchtree management. Despite our efforts to understand the polyhedral combinatorics of the matrix decomposition problem, we do not have a strong grip on the corresponding polytope, and after an initial phase of rapid growth of the lower bound, stalling occurs in the presence of significant duality gaps. We believe that—up to a certain point—it is favorable in this situation to resort to branching early, even if there is still slow progress in the cutting plane loop. In fact, we apply a rather "aggressive" *branching strategy*, splitting the currently processed node if the duality gap could not be reduced by at least 10% in any 4 consecutive LPs. On the other hand, we *pause* a node (put it back into the list of nodes yet to be processed) if the local lower bound exceeds the global lower bound by at least 10%.

Branching itself is guided by the fractional LP-values. We first look for a most fractional $z(x)$-value. If, e.g., $z(x)_i$ is closest to 0.5 (breaking ties arbitrarily), we create $\beta + 1$ new nodes corresponding to the variable fixings

$$x_i^1 = 1, \quad x_i^2 = 1, \ldots, x_i^\beta = 1, \quad \text{and} \quad \sum_{b=1}^{\beta} x_i^b = 0.$$

In other words, we branch on the block assignment constraint corresponding to row i. The advantage of this scheme is that it leads to only $\beta + 1$ new nodes instead of 2^β-nodes in an equivalent binary searchtree. If all $z(x)$-values are integral, we identify a row with a most fractional x-variable and perform the same branching step. We have also tried other branching rules by taking, for instance, the degree of a row in $G(A^T)$ into account, but the performance was inferior to the current scheme.

5. Computational results. In this section we report on computational experiences with our branch-and-cut algorithm for the solution of matrix decomposition problems arising in linear and integer programming problems. Our aim is to find answers to two complexes of *questions*. First, we would like to evaluate our branch-and-cut approach: What are the limits in terms of the size of the matrices that we can solve with our algorithm? What is the quality of the cuts, do they provide a reasonable solution guarantee? Second, we want to discuss our concept of decomposition into bordered block diagonal form: Do the test instances have this structure or are most integer programming matrices not decomposable in this way? Does knowledge of the decomposition help to solve them? And do our heuristics provide reasonable decompositions that could be used within a parallel LU-factorization framework or an integer programming code?

Our *test set* consists of matrices from real-world linear programs from the Netlib and integer programming matrices from the Miplib. In addition, we consider two sets of problems with "extreme" characteristics as benchmark problems: some small matrices arising from Steiner-tree packing problems (see Grötschel, Martin, and Weismantel [14]) and equipartition problems introduced in Nicoloso and Nobili [22]. The Steiner-tree problems are known to be in bordered block diagonal form, and we wanted to see whether our code is able to discover this structure. The equipartition problems, on the other hand, are randomly generated. Our complete test data are available via anonymous ftp from ftp.zib.de at /pub/Packages/mp-testdata/madlib and at the URL ftp://ftp.zib.de/pub/Packages/mp-testdata/madlib/index.html.

In the following subsections, we report the *results of our computations* on the different sets of problems. Our algorithm is implemented in `C` and consists of about 36,000 lines of code. The test runs were performed on a Sun Ultra Sparc 1 Model 170E and we used a time limit of 1,800 CPU seconds. The format of the upcoming tables is as follows: Column 1 provides the name of the problem, and Columns 2–4 contain the number of rows, columns, and nonzeros of the matrix to be decomposed. The two succeeding columns give the number of columns and nonzeros after presolve. Comparing Column 3 with 5 and 4 with 6 shows the performance of our preprocessing. The succeeding five columns give statistics about the number of cuts generated by our code. There are, from left to right, the number of initial cuts (*Init*) including block assignment, block capacity, big-edge, star, and tie-breaking inequalities (but not composition of clique inequalities, although they are also separated from the pool); the number of z-cover (*Cov*); the number of two-partition (2*part*); the sum of the number of bin-packing, cycle, z-cycle, clique, z-clique, and composition of clique inequalities (*BCC*); and finally the number of violated inequalities separated from the pool (*pool*). The following two columns (Columns 12 and 13) show the number of branch-and-bound nodes (*Nod*) and the number of LPs (*Iter*) solved by the algorithm. The second part of the tables starts again with the name of the problem. The next eight columns give solution values. We do not report the number of assigned rows but the number of rows in the border, because it is easier to see whether the matrix could be decomposed into block diagonal form (in this case the value is zero) or close to this form (then the value is a small positive integer). *Lb* gives the global lower bound provided by the algorithm. It coincides with the value of the upper bound *Ub* (next column) when the problem is solved to *proven optimality*. Skipping two columns for a moment, the next four columns refer to the heuristics. G, $D1$, $D2$, and B stand for the *greedy*, the *dual* *(variant* I *and* II*)*, and the *bin-packing* heuristic. The corresponding columns show the best solutions obtained by these heuristics throughout the computations at the root node. If this value coincides with the number of rows of the matrix, all rows are in the border and the heuristic failed. The two (skipped) columns right after *Ub* show which heuristic *He* found the best solution after *No* many branch-and-bound nodes (1 means it was found in the root node; 0 means that preprocessing solved the problem). The additional letter *I* indicates that the value was obtained by a succeeding call to the *improvement heuristic*. *BS* means that the bin-packing separation routine found the best solution, an asterisk $*$ shows that the LP solution provided an optimal block decomposition. The remaining five columns show timings. The last of these columns *Tot* gives the total running time measured in CPU seconds. The first four columns show the percentage of the total time spent in cut-management (*Cm*), i.e., local setup, LP-, cut-buffer, and pool-management, the time to solve the linear programs (*LP*), the time of the separation algorithms (*Sep*), and the time for the heuristics (*Heu*).

5.1. The Netlib problems. The first test set that we are going to consider consists of matrices that arise from *linear programming problems* taken from the Netlib.[1] We investigated whether *basis matrices* corresponding to optimal solutions of these linear programs can be decomposed into (bordered) block diagonal form. These bases were taken from the dual simplex algorithm of `CPLEX` [18]. Analyzing the decomposibility of such matrices gives insight into the potential usefulness of parallel LU-factorization methods within a simplex-type solver. In this context β reflects the number of processors that are available. We have chosen $\beta = 4$, since this is somehow the first interesting case where parallelization might pay. As a heuristic means for good load balancing we aim at equal-sized blocks and have set the capacity to $\kappa := \frac{\#rows}{4}$ rounded up. We tested all instances with up to 1,000 rows.

Table 5.1 shows the *results* of these experiments. The problems up to 100 rows are easy. The range of 100–200 rows is where the limits of our code become visible, and this is the most interesting "hot area" of our table: The problems here are already difficult, but because of the combinatorial structure and not because of sheer size. The results for the problems with more than 200 rows are of limited significance, because these matrix decomposition problems are large-scale and the algorithm solves too few LPs within the given time limit. We can solve only a couple of readily decomposable large instances, but it is worth noticing that a significant number of such instances exists: The difficulty of matrix decomposition problems depends as much on the structure of the matrix as on the number of rows, columns, or nonzeros.

Let us first investigate the "*dual side*" of the results. We observe that we solve very few problems at the root node (only 9 out of 77), and that the number of cuts is very large, in particular in the hot area of the table. The reason for this is basically the symmetry of the problem, as can be seen from the pool separation column (*Pool*) that counts, in particular, all violated tie-breaking cuts. Unfortunately, we don't see a way to get around this phenomenon, but we believe that the symmetry mainly prevents us from solving difficult instances of larger size. The quality of the cuts is in our opinion reasonable, as can be seen from the size of the branch-and-bound tree and the number of LPs solved. It is true, however, that the lower bound improves fast at first while stalling occurs in later stages of the computation although still large numbers of cuts are found and the problem is finished by branch-and-bound. The same behavior has been reported for similar problems like the node capacitated graph

[1] Available by anonymous ftp from ftp://netlib2.cs.utk.edu/lp/data.

Table 5.1
Decomposing LP-basis matrices (part I*).*

Name	Original			Presolved		Cuts					B&B	
	rows	col	nz	col	nz	Init	Cov	2part	BCC	Pool	Nod	Iter
seba	2	2	2	0	0	0	0	0	0	0	1	0
afiro	20	20	34	10	24	106	12	6	5	2	1	2
fit1d	24	24	178	9	126	298	97	38	265	105	6	18
fit2d	25	25	264	11	147	379	549	458	1504	892	41	99
sc50b	28	28	84	27	82	124	2304	1291	152	5452	56	182
sc50a	29	29	88	24	77	137	1062	654	53	1487	26	79
kb2	39	39	213	16	164	328	38	21	2	8	1	3
vtpbase	51	51	198	27	149	415	9130	4796	407	32630	561	1025
bore3d	52	52	311	29	262	579	2241	1025	2333	2706	51	116
adlittle	53	53	203	35	167	454	818	244	20	515	11	31
blend	54	54	313	28	274	547	3357	1553	2757	6143	61	166
recipe	55	55	100	24	69	317	0	14	3	2	1	3
scagr7	58	58	242	34	210	439	770	331	718	468	16	30
sc105	59	59	220	48	203	318	6200	3415	151	19447	66	207
stocfor1	62	62	180	25	143	497	2395	801	55	2977	21	65
scsd1	77	77	215	70	207	500	4264	1570	60	6331	21	74
beaconfd	90	90	618	48	576	1626	4618	1403	3231	6296	121	263
share2b	93	93	482	37	340	825	7335	4636	212	14713	626	860
share1b	102	102	485	56	364	721	3769	2135	294	8204	806	1142
forplan	104	104	575	70	535	815	43011	16754	4783	145556	316	681
scorpion	105	105	383	46	300	812	48144	12649	1178	113261	1206	2485
sc205	113	113	691	104	676	561	24727	15842	272	116813	136	299
brandy	113	113	874	83	774	1270	33012	13041	5037	107955	271	664
lotfi	122	122	349	60	267	1026	85763	17243	1110	203255	1076	2217
boeing2	122	122	435	70	383	1038	18823	5794	200	41101	71	237
tuff	137	137	820	84	748	1795	17387	4823	200	32293	116	306
grow7	140	140	1660	51	243	948	33529	12866	320	182151	206	424
scsd6	147	147	383	138	373	804	41055	12440	297	116468	111	331
e226	148	148	954	88	807	1469	33011	14063	272	112223	206	355
israel	163	163	1321	37	1014	2217	537	289	2115	237	11	25
agg	164	164	669	52	551	1492	37378	12257	325	174665	341	539
capri	166	166	826	102	718	1625	9291	4015	396	9008	71	94
wood1p	171	171	2393	55	1340	1650	6253	2457	56	7758	46	94
bandm	180	180	1064	90	815	1690	11854	6042	405	31838	66	96
scrs8	181	181	887	110	675	1400	11399	5375	72	30637	56	85
ship04s	213	213	573	176	536	1530	1084	72	7	190	1	7
scagr25	221	221	1627	91	1470	1831	14509	8239	6884	23224	121	261
scfxm1	242	242	1064	154	922	2182	16607	3887	68	38232	31	80
stair	246	246	3402	216	3154	1552	7192	3058	33	1005	51	44
shell	252	252	493	235	476	1495	14086	1311	85	5878	46	101
standata	258	258	513	109	364	1733	0	0	0	0	1	1
sctap1	269	269	640	77	373	1957	25028	4187	110	37924	61	143
agg2	280	280	1468	109	1275	2270	9856	4124	34	13961	31	45
agg3	282	282	1444	99	1162	2259	9572	4674	32	13575	26	43
ship08s	284	284	699	201	616	1909	7352	480	33	3403	11	35
boeing1	284	284	1384	174	1271	2670	13291	4198	48	22032	31	56
grow15	300	300	3680	102	489	2044	17134	5225	72	28665	21	77
fffff800	306	306	1382	182	1237	3045	9590	3320	30	15509	36	51
etamacro	307	307	1005	215	907	1824	8431	3985	33	10512	16	36
ship04l	313	313	868	274	829	2321	4387	520	125	1918	356	491
gfrdpnc	322	322	623	276	576	1569	13308	2328	53	4899	31	59
ship12s	344	344	858	247	761	2313	1676	108	7	62	6	8
finnis	350	350	831	178	653	2371	13570	3020	55	17997	26	61
pilot4	352	352	3157	265	2988	3014	8563	4104	25	13824	16	29
standmps	360	360	836	217	691	2785	32633	4731	183	37695	71	210
degen2	382	382	2440	262	2230	2717	6922	1959	16	2396	26	25
scsd8	397	397	1113	394	1109	1755	9128	4799	24	14671	6	25
grow22	440	440	5272	161	732	2984	10976	1880	26	10062	11	28
bnl1	448	448	1656	307	1481	3788	6984	1623	16	6521	11	19
czprob	475	475	939	464	928	3640	1227	25	4	9	1	4
scfxm2	485	485	2179	313	1906	4317	7582	1712	17	10240	6	18
perold	500	500	3277	424	3077	3448	5983	3669	13	6810	6	14
ship08l	520	520	1404	436	1320	3767	11855	703	76	4080	71	129
maros	545	545	2637	301	2323	4384	7750	2718	15	2539	11	18
ganges	576	576	3002	466	2892	4098	19490	2429	40	10251	16	42
pilotwe	613	613	2982	561	2833	3804	5489	4184	10	4238	6	11
nesm	622	622	1925	419	1461	3720	5929	3407	11	4785	6	12
fit1p	627	627	4992	1	627	5013	0	0	0	0	1	1
25fv47	677	677	3750	442	3278	5699	4644	1315	7	482	6	9
ship12l	686	686	1883	589	1786	5026	3212	127	6	74	1	6
woodw	711	711	3044	528	2849	4517	3736	2395	6	399	11	9
scfxm3	728	728	3285	469	2876	6411	4969	1203	8	3541	6	9
pilotja	745	745	4738	548	4121	5708	4462	2129	7	670	6	8
pilotnov	783	783	4428	498	3606	5479	3859	2466	6	1159	6	7
bnl2	940	940	3284	489	2509	6981	4930	880	7	634	6	8
sctap2	977	977	1491	161	642	5506	1256	0	0	156	6	6
truss	1000	1000	3564	986	3540	4792	4999	4137	6	1810	6	7
$\sum$	22911	22911	108546	14614	82679	169450	867384	285672	37498	1909629	8022	15550

partitioning problem discussed in Ferreira et al. [7].

Investigating the "*primal side*," we see that the greedy heuristic seems to be most reliable. The two dual methods perform exactly the same and yield solutions of the same quality as the greedy. Bin-packing is either very good (a rather rare event) or a catastrophe, but it complements the other heuristics. If we look at the quality of

Table 5.1
Decomposing LP-basis matrices (part II*).*

Name	Best Solutions				Heuristics at Root				Time				
	Lb	Ub	He	No	G	D1	D2	B	Cm	LP	Sep	Heu	Tot
seba	0	**0**	-	0	-	-	-	-	0%	0%	0%	0%	0.0
afiro	2	**2**	B	1	3	3	3	2	0%	40%	10%	0%	0.1
fit1d	16	**16**	B	1	16	17	17	16	6%	50%	25%	6%	0.7
fit2d	18	**18**	D1	1	20	18	18	25	12%	54%	22%	5%	5.5
sc50b	11	**11**	IG	1	11	11	11	28	6%	71%	12%	4%	12.7
sc50a	8	**8**	ID1	1	8	8	8	29	3%	78%	11%	2%	6.8
kb2	14	**14**	IG	1	14	14	14	39	4%	69%	14%	2%	0.5
vtpbase	14	**14**	IG	1	14	17	17	15	7%	57%	20%	8%	94.6
bore3d	23	**23**	D1	9	24	24	24	52	7%	69%	17%	2%	24.0
adlittle	10	**10**	D1	1	10	10	10	53	6%	72%	15%	2%	5.0
blend	20	**20**	IG	2	21	25	25	21	6%	66%	20%	3%	35.9
recipe	1	**1**	IB	1	3	4	4	1	0%	51%	18%	10%	0.4
scagr7	21	**21**	B	4	25	25	25	58	6%	46%	32%	10%	5.2
sc105	16	**16**	B	20	17	17	17	59	4%	82%	9%	2%	85.1
stocfor1	10	**10**	B	7	12	12	12	62	6%	68%	17%	4%	15.2
scsd1	8	**8**	IB	5	10	10	10	77	3%	81%	10%	3%	53.1
beaconfd	26	**26**	D1	21	28	28	28	90	4%	67%	21%	4%	115.7
share2b	9	**9**	ID1	5	10	10	10	93	4%	77%	8%	6%	282.5
share1b	7	**7**	B	4	8	8	8	102	5%	65%	13%	9%	203.6
forplan	31	36	ID1	18	39	44	44	104	2%	85%	8%	3%	1801.1
scorpion	11	**11**	B	7	14	14	14	105	4%	67%	14%	9%	940.7
sc205	28	39	IG	9	45	45	45	113	1%	92%	4%	1%	1806.6
brandy	34	**34**	*	214	51	51	51	113	2%	82%	11%	2%	1200.7
lotfi	19	**19**	B	279	23	23	23	122	4%	76%	12%	4%	1643.3
boeing2	19	**19**	B	33	31	32	32	122	2%	84%	7%	3%	411.9
tuff	26	**26**	B	43	28	35	35	137	2%	85%	7%	3%	635.3
grow7	10	18	IG	9	22	22	22	140	3%	80%	9%	6%	1801.2
scsd6	10	22	IG	4	23	23	23	147	1%	87%	5%	4%	1805.1
e226	22	30	IG	15	44	44	44	148	1%	84%	7%	5%	1800.5
israel	98	**98**	IG	2	101	119	118	163	4%	32%	51%	10%	42.1
agg	19	24	IG	56	43	43	43	164	3%	81%	9%	5%	1802.3
capri	23	53	D1	7	59	64	64	166	0%	90%	4%	3%	1801.8
wood1p	28	**28**	B	29	32	32	32	171	2%	73%	15%	7%	277.6
bandm	19	43	IG	11	58	58	58	180	1%	93%	4%	1%	1835.0
scrs8	17	43	IG	9	56	56	56	181	0%	92%	3%	2%	1802.1
ship04s	4	**4**	IB	1	8	8	8	4	5%	55%	31%	2%	10.8
scagr25	62	69	B	53	81	81	81	221	1%	52%	21%	23%	1814.7
scfxm1	10	30	IG	7	50	50	50	242	1%	87%	5%	5%	1803.7
stair	58	122	IG	3	123	133	133	246	0%	83%	10%	5%	1914.6
shell	4	**4**	B	17	8	8	8	252	3%	58%	16%	18%	187.8
standata	1	**1**	D1	1	17	1	1	258	0%	52%	35%	3%	5.3
sctap1	9	19	D1	13	20	20	20	269	1%	86%	6%	5%	1801.2
agg2	13	78	IG	5	110	112	112	280	0%	89%	5%	3%	1810.8
agg3	12	79	IG	6	92	116	116	282	0%	91%	4%	2%	1808.6
ship08s	4	**4**	B	3	10	10	10	284	3%	76%	14%	4%	144.0
boeing1	12	55	IG	2	56	56	56	284	1%	91%	3%	4%	1812.6
grow15	7	20	ID1	1	20	20	20	300	1%	84%	5%	8%	1817.1
fffff800	12	52	IG	6	79	104	104	306	0%	89%	5%	3%	1820.2
etamacro	6	83	IG	3	91	91	91	307	0%	93%	2%	2%	1832.6
ship04l	5	**5**	B	14	8	8	8	313	1%	66%	3%	25%	1727.2
gfrdpnc	4	**4**	BS	7	8	8	8	322	2%	65%	9%	21%	398.4
ship12s	3	**3**	IB	2	10	10	10	344	4%	61%	24%	6%	35.4
finnis	7	27	B	4	31	31	31	350	1%	90%	4%	3%	1842.8
pilot4	6	106	IG	1	106	109	109	352	0%	87%	6%	4%	1881.6
standmps	7	10	D1	4	11	11	11	360	2%	79%	9%	7%	1802.0
degen2	12	114	ID1	1	114	114	114	382	0%	92%	3%	2%	1922.1
scsd8	4	63	IG	2	85	98	89	397	1%	90%	2%	5%	1912.1
grow22	4	24	ID1	3	25	31	25	440	1%	83%	4%	11%	1808.9
bnl1	6	68	D1	1	68	68	68	448	0%	93%	2%	3%	2046.8
czprob	3	**3**	D1	1	3	3	3	475	4%	53%	35%	1%	42.0
scfxm2	5	65	IG	2	113	120	115	485	1%	80%	4%	13%	1849.6
perold	5	166	D1	1	166	166	166	500	0%	90%	3%	4%	2079.5
ship08l	4	5	D1	9	12	12	12	520	1%	48%	7%	40%	1807.1
maros	8	103	D1	1	103	103	103	545	1%	85%	4%	8%	1832.8
ganges	6	15	IB	3	36	39	39	576	2%	65%	10%	21%	1805.4
pilotwe	3	181	IG	2	182	186	182	613	0%	83%	3%	11%	1911.8
nesm	3	151	D1	2	152	152	152	622	1%	86%	4%	7%	1803.8
fit1p	470	**470**	IG	1	470	470	470	627	0%	92%	5%	0%	726.4
25fv47	4	190	IG	1	190	195	195	677	0%	90%	3%	5%	2102.4
ship12l	3	**3**	B	1	4	18	18	3	3%	57%	27%	8%	169.8
woodw	5	184	IG	2	186	187	187	711	0%	89%	4%	5%	2069.4
scfxm3	3	104	D1	2	105	105	105	728	0%	90%	3%	4%	1924.6
pilotja	3	226	ID1	1	226	226	226	745	0%	88%	4%	6%	1989.3
pilotnov	3	250	D1	1	250	250	250	783	0%	85%	3%	9%	1855.4
bnl2	3	199	D1	1	199	199	199	940	0%	81%	4%	11%	2051.2
sctap2	1	**1**	D1	2	2	4	4	2	0%	8%	16%	73%	834.5
truss	3	286	ID1	2	288	288	288	1000	0%	70%	4%	23%	2347.0
$\sum$	1455	4423		1026	4841	4987	4962	20893	1%	82%	6%	8%	85517.1

the solutions found at the root node as a measure of the method as a stand-alone decomposition heuristic, the table shows pretty good results for the small problems. For the larger instances the situation is a bit different. We detect larger gaps; see, for instance, `scfxm1` or `ship12s`. In fact, we have often observed in longer runs on larger examples that the best solution could steadily be improved and the optimal solution was found late. A reason might be that the heuristics are closely linked to the LP-fixings and essentially always find the same solutions until the branching process forces them strongly into another direction. We believe (and for some we know) that many of the larger problems can be decomposed much better than the *Ub*-column indicates and that there might be potential to further improve stand-alone primal heuristics.

The answer to the final question of whether LP basis matrices are decomposable into four blocks is both yes and no. Some like `recipe` or `standata` are decomposable (only one row is in the border), but others are not; for example, `israel`: 98 out of 163 are in the border. However, the results leave the possibility that larger LP-matrices, which are generally sparser than small ones, can be decomposed better so that we cannot give a final answer to this question.

5.2. The Miplib problems. In this test series we examine whether matrices arising from *integer programs* can be decomposed into (bordered) block diagonal form. There are two applications here.

First, decomposing the original constraint matrix A of some general integer program can be useful to *tighten* its LP-relaxations within a branch-and-cut algorithm. The structure of the decomposed matrix is that of a multiple knapsack or general assignment problem, and inequalities known for the associated polytopes (see Gottlieb and Rao [12], Ferreira, Martin, and Weismantel [8]) are valid for the MIP under consideration. The first interesting case in this context involves two blocks, and we set $\beta := 2$. We used $\kappa := \frac{(\#\text{rows})\cdot 1.05}{2}$ rounded up as the block capacity, which allows a deviation of 10% of the actual block sizes in the decomposition.

Table 5.2 shows the *results* that we obtained for matrices of mixed integer programs taken from the Miplib[2] and preprocessed with the presolver of the general purpose MIP-solver `SIP` that is currently under development at the Konrad-Zuse-Zentrum. We again considered all instances with up to 1,000 rows.

The picture here is a bit different from the one for the linear programming problems. Since the number of blocks is $\beta = 2$ instead of $\beta = 4$, the IP-formulation is much smaller: The number of variables is only one half, the number of conflicting assignments for two adjacent rows is only 4 instead of 12. In addition, there is much less symmetry in the problem. Consequently, the number of cuts does not increase to the same extent; the LPs are smaller and easier to solve (the percentage of LP-time in column *LP* decreases). We can solve instances up to 200 rows and many of the small ones within seconds. Note that $\beta = 2$, on the other hand, leads to doubled block capacities. This means that it becomes much more difficult to separate inequalities that have this number as their right-hand side and have a large or combinatorially restrictive support like z-clique, bin-packing, or composition of clique inequalities; see column *BCC*.

On the *primal side* the results are similar to the Netlib instances, but the heuristics seem to perform a little better for two blocks than for four.

How decomposable are the MIP matrices? We see that not all, but many, of the larger problems can be brought into bordered block diagonal form (the small ones cannot). Of course, there are also exceptions like the `air`-problems which were expected to be not decomposable. Anyway, there seems to be potential for the multiple-knapsack approach and further research in this direction, especially because there are only very few classes of cutting planes known for general MIPs.

[2] Available from the URL http://www.caam.rice.edu:80/∼bixby/miplib/miplib.html.

Table 5.2
Decomposing matrices of MIPs (part I*).*

Name	Original			Presolved		Cuts					B&B	
	rows	col	nz	col	nz	Init	Cov	2part	BCC	Pool	Nod	Iter
mod008	6	319	1243	1	6	23	0	0	0	0	1	1
stein9	13	9	45	9	45	50	115	64	1	23	16	34
p0040	13	40	70	30	60	16	0	0	0	0	1	1
p0033	15	32	97	11	41	41	23	3	0	1	1	3
gt1	15	46	92	46	92	19	262	87	0	61	13	33
flugpl	16	16	40	6	21	46	16	0	1	0	1	2
bm23	20	27	478	1	20	79	0	0	0	0	1	1
enigma	21	100	289	99	287	40	922	216	0	315	28	83
air01	23	771	4215	18	125	88	0	0	0	0	1	1
rgn	24	180	460	33	102	64	724	163	1	324	13	47
pipex	25	48	192	48	192	46	421	180	0	201	13	33
lseu	28	88	308	50	185	68	98	43	0	19	1	5
gt2	28	173	346	173	346	39	1333	340	0	290	28	86
sentoy	30	60	1800	1	30	119	0	0	0	0	1	1
stein15	36	15	120	15	120	142	17654	13145	0	26682	5458	7595
misc02	43	55	405	46	368	153	2455	1065	0	1612	34	110
sample2	45	64	140	55	131	81	976	171	1	331	7	29
air02	50	6774	61555	76	897	270	4	0	34	0	1	3
misc01	54	79	729	66	678	218	4556	2762	0	4861	367	581
mod013	62	96	192	48	144	254	1810	108	1	722	13	43
mod014	74	86	172	43	129	246	148	0	1	9	1	3
lp4l	85	1086	4677	791	3462	277	23233	16941	0	45877	1129	1779
bell5	87	101	257	73	215	171	2418	473	1	800	13	38
p0291	92	103	373	63	279	365	2943	366	0	757	19	54
misc03	96	154	2023	133	1934	360	50431	32949	1	85357	12745	15609
l152lav	97	1989	9922	695	3712	308	32271	23459	0	71535	1525	2276
khb05250	100	1299	2598	1275	2574	196	12634	1917	1	3421	73	227
harp2	100	1373	2598	1225	2450	116	250	158	0	3	1	4
bell4	101	114	293	84	248	191	7162	1041	1	2157	49	145
bell3a	107	121	311	89	263	203	2718	492	1	802	10	32
bell3b	107	121	311	89	263	203	2718	492	1	802	10	32
p0201	113	195	1677	177	1527	200	9230	2390	1	5757	46	128
stein27	118	27	378	27	378	353	280101	82635	3	382512	1753	3874
air03	124	10757	91028	656	5878	830	21108	9064	0	24479	130	377
p0808a	136	240	480	120	304	392	14298	1463	1	4711	49	175
mod010	146	2655	11203	1973	8404	453	53084	33036	1	118368	727	1170
blend2	169	319	1279	88	1039	515	0	0	0	0	4	4
noswot	182	127	732	50	455	745	84957	34285	2	533391	1006	1646
10teams	210	1600	9600	1600	9600	210	67719	25710	1	127827	130	418
misc07	224	254	8589	229	8474	894	10476	8516	0	13110	118	119
vpm1	234	378	749	203	574	906	18959	446	1	4060	34	117
vpm2	234	378	917	203	574	906	18959	446	1	4060	34	117
p0808acuts	246	240	839	230	828	612	15096	1731	1	3058	22	84
p0548	257	477	1522	250	1009	523	94934	20509	1	94842	175	462
misc05	266	131	2873	129	2869	581	16094	4604	0	5545	43	80
modglob	289	420	966	356	902	865	75843	1685	1	29153	148	422
gams	291	556	2431	540	2400	1622	41190	11	0	6897	169	295
fiber	297	1232	2644	418	1254	593	32222	7410	1	19253	31	123
p0282	305	202	1428	168	841	609	56425	17488	1	52114	91	252
stein45	331	45	1034	45	1034	992	25305	11011	0	22779	28	88
qnet1_o	369	1454	4040	837	2474	699	34759	9401	1	19450	37	111
qnet1	407	1454	4405	924	2843	653	24614	8345	1	16223	22	69
air05	408	7195	50762	3104	23458	600	5677	1286	0	207	13	19
fixnet3	478	878	1756	498	1374	1990	23068	145	1	959	34	83
fixnet4	478	878	1756	498	1374	1990	26165	192	1	1860	34	97
set1ch	477	697	1382	445	1130	1437	36069	2756	1	9871	25	89
fast0507	484	63001	406865	4927	31419	659	9549	4335	0	2135	10	24
set1al	492	712	1412	460	1160	1452	27188	2041	1	5790	22	65
set1cl	492	712	1412	460	1160	1452	26700	2001	1	5711	22	64
gen	622	797	2064	360	1303	2076	15337	304	1	3914	13	30
mod015	622	797	2064	360	1303	2076	14722	299	1	3398	13	28
danoint	664	521	3232	521	3232	1016	13895	5307	1	4119	7	24
misc06	696	1572	5126	949	3106	1298	15987	3428	1	4911	13	28
air06	763	8572	67571	4092	34800	1145	7622	3531	1	732	4	12
air04	782	8904	70189	4154	35000	1052	6242	3243	1	434	4	10
adrud	795	998	15876	495	8479	6342	3955	287	0	209	25	32
$\sum$	14814	134914	876632	35938	221378	43230	1395844	405976	75	1778801	26610	39627

The second application of matrix decomposition to integer programming is a *new branching rule*. Decomposing the transposed constraint matrix will identify the variables in the border as linking variables that are interesting candidates for branching. Since most MIP-codes create a binary searchtree, we try to decompose these matrices into $\beta := 2$ blocks. As block capacity we use $\kappa := \frac{(\#\text{rows})\cdot 1.05}{2}$ rounded up to obtain two subproblems of roughly the same size. The test set consists of all problems with up to 1,000 rows (1,000 columns in the original problem).

Table 5.3 shows the *results* of our computations. Surprisingly, the performance of our algorithm not only is similar to the "primal" case but in fact is even better! We can solve almost all problems with up to 400 rows. One reason for this is that

Table 5.2
Decomposing matrices of MIPs (part II*).*

Name	Best Solutions				Heuristics at Root				Time				
	Lb	Ub	He	No	G	D1	D2	B	Cm	LP	Sep	Heu	Tot
mod008	3	**3**	IG	1	3	3	3	6	0%	6%	6%	0%	0.1
stein9	7	**7**	IG	1	7	7	7	13	6%	34%	24%	6%	0.3
p0040	3	**3**	*	1	13	13	13	13	0%	0%	40%	0%	0.1
p0033	4	**4**	D1	1	4	4	4	15	0%	50%	16%	0%	0.1
gt1	6	**6**	D1	1	6	6	6	15	4%	47%	26%	4%	0.4
flugpl	1	**1**	ID2	1	3	3	1	16	0%	25%	0%	0%	0.0
bm23	10	**10**	IG	1	10	10	10	20	0%	22%	0%	11%	0.1
enigma	10	**10**	IG	1	10	10	10	21	7%	41%	28%	14%	2.0
air01	3	**3**	IG	1	3	3	3	23	0%	7%	0%	0%	0.4
rgn	5	**5**	IG	1	5	5	5	24	12%	48%	18%	10%	1.2
pipex	9	**9**	IG	1	9	9	9	25	12%	42%	23%	7%	0.8
lseu	7	**7**	D1	1	7	7	7	28	8%	30%	43%	4%	0.2
gt2	11	**11**	IG	1	11	11	11	28	8%	46%	28%	8%	2.8
sentoy	15	**15**	IG	1	15	15	15	30	0%	18%	12%	0%	0.2
stein15	18	**18**	IG	1	18	18	18	36	18%	27%	22%	16%	90.1
misc02	15	**15**	IG	9	17	21	21	43	6%	59%	18%	11%	9.2
sample2	4	**4**	D1	1	4	4	4	45	6%	61%	21%	6%	2.0
air02	22	**22**	IG	1	22	24	24	50	0%	6%	18%	0%	6.0
misc01	24	**24**	IG	17	25	26	26	54	7%	42%	29%	14%	31.4
mod013	6	**6**	D1	1	6	6	6	62	6%	68%	17%	3%	5.1
mod014	2	**2**	D1	1	2	2	2	74	5%	32%	35%	8%	0.3
lp4l	35	**35**	IG	487	39	41	41	85	4%	13%	27%	52%	481.2
bell5	4	**4**	ID1	4	6	7	7	87	5%	67%	15%	9%	10.6
p0291	7	**7**	D1	1	7	7	7	92	7%	52%	21%	15%	10.7
misc03	43	44	IG	9	46	46	46	96	5%	11%	39%	38%	1800.0
l152lav	36	**36**	IG	1102	43	43	43	97	3%	16%	24%	53%	754.5
khb05250	25	**25**	IG	1	25	25	25	100	3%	25%	42%	25%	89.0
harp2	17	**17**	D1	1	17	17	17	100	0%	2%	94%	0%	20.4
bell4	5	**5**	IB	4	11	11	11	101	6%	55%	17%	17%	31.7
bell3a	4	**4**	B	6	13	13	13	107	5%	63%	16%	12%	14.5
bell3b	4	**4**	B	6	13	13	13	107	5%	63%	15%	13%	14.6
p0201	21	**21**	IG	24	30	30	30	113	3%	64%	12%	19%	125.4
stein27	32	56	IG	1	56	56	56	118	12%	44%	34%	6%	1800.1
air03	49	**49**	IG	13	58	59	58	124	1%	57%	27%	12%	1146.7
p0808a	8	**8**	D1	1	8	8	8	136	6%	62%	16%	13%	83.2
mod010	47	59	IG	28	61	66	66	146	2%	24%	25%	46%	1802.5
blend2	10	**10**	IG	1	10	10	10	169	2%	57%	22%	1%	1.7
noswot	10	15	IG	10	37	42	42	182	10%	47%	33%	6%	1800.4
10teams	43	90	D1	1	90	90	90	210	3%	58%	30%	6%	1804.0
misc07	57	93	IG	11	101	107	107	224	0%	23%	22%	52%	1836.5
vpm1	7	**7**	IG	1	7	14	14	234	3%	80%	9%	5%	322.7
vpm2	7	**7**	IG	1	7	14	14	234	3%	79%	9%	6%	317.4
p0808acuts	8	**8**	D1	1	8	8	8	246	4%	65%	12%	17%	231.7
p0548	25	49	IG	35	68	68	68	257	5%	69%	12%	12%	1800.1
misc05	28	116	IG	13	120	126	126	266	1%	89%	5%	3%	1835.0
modglob	8	**8**	IG	18	12	12	12	289	3%	74%	11%	9%	1512.6
gams	15	21	B	12	36	36	36	291	1%	84%	9%	3%	1801.1
fiber	9	22	ID1	3	38	38	38	297	2%	86%	4%	6%	1807.5
p0282	23	36	D1	1	36	36	36	305	3%	71%	9%	14%	1803.2
stein45	11	154	IG	2	157	157	157	331	1%	89%	5%	2%	1820.3
qnet1_o	12	28	D1	1	28	28	28	369	2%	74%	6%	15%	1812.7
qnet1	8	35	D1	1	35	35	35	407	2%	77%	5%	14%	1801.5
air05	19	189	IG	1	189	194	194	408	0%	67%	20%	10%	1868.9
fixnet3	12	13	D1	1	13	13	13	478	1%	88%	4%	4%	1809.4
fixnet4	13	**13**	D1	1	13	13	13	478	1%	85%	5%	6%	1789.9
set1ch	7	12	D1	1	12	12	12	477	3%	72%	7%	16%	1804.4
fast0507	5	183	IG	3	217	217	217	484	1%	51%	15%	26%	1800.1
set1al	6	12	D1	1	12	12	12	492	2%	78%	5%	13%	1816.9
set1cl	6	12	D1	1	12	12	12	492	2%	77%	5%	14%	1805.3
gen	5	19	ID1	1	19	19	19	622	1%	68%	5%	24%	1809.8
mod015	5	19	ID1	1	19	19	19	622	1%	68%	4%	24%	1807.0
danoint	4	188	ID1	2	189	189	189	664	1%	77%	6%	14%	1861.4
misc06	4	70	ID1	1	70	70	70	696	2%	62%	7%	27%	1835.3
air06	3	354	IG	2	356	363	362	763	1%	43%	35%	18%	1934.3
air04	3	355	IG	1	355	355	355	782	0%	34%	26%	37%	1913.6
adrud	10	32	D1	1	32	32	32	795	0%	77%	10%	10%	1835.6
$\sum$	905	2729		1863	2931	2990	2986	14814	2%	63%	15%	17%	56337.8

MIPs tend to have sparse columns but not necessarily sparse rows. Dense columns in the transposed matrices (dense rows in the original ones) leave less freedom for row assignments; there are fewer possibilities for good decompositions, and the LP-relaxations are tighter than in the primal case.

For the reason just mentioned we expected that the transposed problems would be less decomposable than the originals, and it came as a surprise to us that the "dual" problems decompose nearly as well as the primal ones. The transposed matrices have on average about 60 rows in the border, in comparison to only 40 for the originals. However, the percentage of rows in the border (i.e., the sum of the *Ub* column divided by the sum of the *rows* column) is 18.4% (12.8%) in the primal and 20.6% (25.3%)

TABLE 5.3
Decomposing transposed matrices of MIPs (part I*).*

Name	Original			Presolved		Cuts					B&B	
	rows	col	nz	col	nz	Init	Cov	2part	BCC	Pool	Nod	Iter
stein9	9	13	45	1	9	35	0	0	0	0	1	1
stein15	15	36	120	1	15	59	0	0	0	0	1	1
flugpl	16	16	40	11	30	36	16	2	1	0	1	2
bm23	27	20	478	3	78	109	0	0	0	0	1	1
stein27	27	118	378	1	27	107	0	0	0	0	1	1
p0033	32	15	97	9	59	120	205	51	0	31	7	16
p0040	40	13	70	13	70	100	47609	7530	10	27421	1522	3361
stein45	45	331	1034	1	45	179	0	0	0	0	1	1
gt1	46	15	92	15	92	46	119471	29195	15	71563	4924	8786
pipex	48	25	192	19	96	48	142	76	16	0	4	7
misc02	55	43	405	18	229	279	547	517	0	562	25	60
sentoy	60	30	1800	1	60	239	0	0	0	0	1	1
sample2	64	45	140	45	140	82	2223	638	1	1308	13	45
misc01	79	54	729	23	359	435	2138	1121	0	1229	28	79
mod014	86	74	172	74	172	172	6137	697	1	1927	28	109
lseu	88	28	308	20	198	332	10336	3532	0	8335	76	216
mod013	96	62	192	62	192	192	448495	45097	11	171613	4711	8604
enigma	100	21	289	12	199	375	3695	1802	0	1575	40	110
bell5	101	87	257	87	257	185	5598	969	1	1645	40	108
p0291	103	92	373	65	319	261	7980	2306	0	3581	61	170
bell4	114	101	293	101	293	204	11829	2005	1	3979	109	277
bell3a	121	107	311	107	311	217	4322	942	1	1294	10	46
bell3b	121	107	311	107	311	217	4322	942	1	1294	10	46
noswot	127	182	732	103	531	427	55038	4869	1	12790	778	1605
misc05	131	266	2873	79	535	289	255523	30000	0	124752	2236	3462
misc03	154	96	2023	39	769	840	3666	4828	0	8008	328	565
gt2	173	28	346	28	346	173	173371	81355	37	280323	1807	2468
rgn	180	24	460	24	460	730	38963	6126	1	22421	112	347
p0201	195	113	1677	77	699	579	181201	53287	74	223647	625	1295
p0282	202	305	1428	189	1196	407	4311	1449	0	479	13	36
p0808a	240	136	480	136	480	480	106598	14308	1	70632	151	507
p0808acuts	240	246	839	191	729	674	107094	7435	1	58191	202	525
misc07	254	224	8589	53	1367	1456	11803	14984	0	28234	757	1117
blend2	319	169	1279	160	478	955	103558	8951	1	21986	226	680
vpm1	378	234	749	234	749	784	37116	4150	1	14141	37	136
vpm2	378	234	917	66	581	1072	25484	2483	1	6355	19	87
modglob	420	289	966	289	966	1014	42137	6770	1	19025	40	116
p0548	477	257	1522	153	823	1525	36798	3048	1	8883	46	96
danoint	521	664	3232	544	2336	649	21785	10004	1	12677	13	47
gams	556	291	2431	91	1096	4003	14962	2869	0	4200	19	35
set1ch	697	477	1382	477	1382	1657	20852	1581	1	2680	10	34
set1al	712	492	1412	492	1412	1672	16354	1713	1	2886	7	26
set1cl	712	492	1412	492	1412	1672	16354	1713	1	2886	7	26
air01	771	23	4215	22	4185	7186	0	0	0	0	4	2
gen	797	622	2064	298	1416	2113	16984	1847	1	2813	10	26
mod015	797	622	2064	298	1416	2113	16984	1847	1	2813	10	26
fixnet3	878	478	1756	478	1756	1638	13145	3715	1	1757	7	18
fixnet4	878	478	1756	478	1756	1638	13130	3611	1	2272	7	18
fixnet6	878	478	1756	478	1756	1638	10505	2989	1	901	7	15
adrud	998	795	15876	1	998	3991	0	0	0	0	1	1
$\sum$	14556	10168	72362	6766	35191	45404	2018781	373354	188	1233109	19094	35364

in the dual case (the values in parentheses count only instances that were solved to optimality). Note, however, that the dual values are biased heavily by the (not decomposable) adrud instance. An explanation may be that many of these problems contain a few important global variables that migrate into the border.

We used optimal decompositions of transposed MIP matrices to test our idea to branch on variables in the border first. The computations were performed using the above mentioned MIP-solver SIP. As our *test set* we selected all problems that are not extremely easy (less than 10 CPU seconds for SIP) and that decompose into bordered block diagonal form with a "relatively small" border: mod014, bell3a, bell3b, bell4, bell5, noswot, blend2, vpm1, vpm2, set1ch, set1al, and set1cl. Unfortunately, and contrary to what we had expected, it turned out that mod014, blend2, vpm1, vpm2, set1ch, set1al, and set1cl have only *continuous* variables in the border that do not qualify for branching variables! All border variables in noswot are integer variables; in the bell-examples 1 (2) out of 5 (6) are integer variables. We tried to extend the test set by decomposing only the integer part of all these matrices but it failed, because the integer parts turned out to have block diagonal form, i.e., no variables in the border.

For the remaining four bell* and the noswot example, we performed the following tests. The MIPs were solved with SIP using four *different branching strategies.*

TABLE 5.3
Decomposing transposed matrices of MIPs (part II).

Name	Best Solutions				Heuristics at Root				Time				
	Lb	Ub	He	No	G	D1	D2	B	Cm	LP	Sep	Heu	Tot
stein9	5	**5**	IG	1	5	5	5	9	0%	9%	0%	0%	0.1
stein15	8	**8**	IG	1	8	8	8	15	0%	18%	0%	0%	0.2
flugpl	1	**1**	IG	1	1	1	1	16	0%	8%	8%	0%	0.1
bm23	13	**13**	IG	1	13	13	13	27	0%	25%	5%	5%	0.2
stein27	13	**13**	IG	1	13	13	13	27	0%	23%	4%	0%	0.2
p0033	6	**6**	IG	1	6	6	6	32	9%	46%	16%	5%	0.5
p0040	13	**13**	IG	1	13	14	14	40	18%	38%	27%	8%	90.5
stein45	22	**22**	IG	1	22	22	22	45	2%	27%	8%	0%	0.4
gt1	16	**16**	IG	20	18	22	22	46	20%	30%	33%	8%	283.8
pipex	20	**20**	IG	2	21	22	22	48	2%	22%	33%	8%	0.4
misc02	27	**27**	D1	1	27	27	27	55	7%	43%	27%	10%	3.8
sentoy	29	**29**	IG	1	29	29	29	60	1%	33%	9%	1%	0.5
sample2	4	**4**	*	9	6	6	6	64	7%	63%	18%	6%	6.6
misc01	38	**38**	IG	1	38	38	38	79	7%	44%	32%	10%	10.3
mod014	8	**8**	ID2	1	8	9	8	86	9%	52%	23%	11%	16.7
lseu	26	**26**	*	71	36	42	39	88	7%	64%	15%	10%	65.4
mod013	14	20	B	32	23	23	23	96	14%	25%	47%	9%	1800.2
enigma	44	**44**	IG	12	46	48	48	100	6%	40%	37%	11%	19.4
bell5	5	**5**	IG	4	9	9	9	101	7%	56%	21%	12%	21.5
p0291	21	**21**	IG	5	24	29	29	103	4%	70%	11%	12%	78.6
bell4	6	**6**	IG	6	9	9	9	114	6%	58%	17%	15%	64.5
bell3a	5	**5**	ID1	1	5	5	5	121	5%	63%	17%	11%	23.8
bell3b	5	**5**	ID1	1	5	5	5	121	5%	64%	17%	10%	23.7
noswot	15	**15**	IG	2	19	19	19	127	6%	40%	21%	28%	337.5
misc05	35	49	IG	64	53	58	58	131	8%	38%	30%	19%	1800.0
misc03	73	**73**	IG	3	74	74	74	154	3%	28%	38%	19%	159.2
gt2	57	67	IG	158	76	83	83	173	7%	56%	23%	10%	1800.2
rgn	20	**20**	IG	1	20	78	42	180	5%	67%	14%	11%	474.6
p0201	42	76	IG	192	86	93	93	195	8%	54%	26%	8%	1800.0
p0282	20	**20**	ID1	1	20	20	20	202	2%	74%	9%	13%	122.7
p0808a	11	18	IG	12	32	36	32	240	5%	76%	12%	5%	1802.1
p0808acuts	12	17	ID2	28	36	36	36	240	5%	69%	12%	11%	1800.7
misc07	119	**119**	IG	1	119	121	121	254	2%	18%	40%	29%	939.8
blend2	38	**38**	D1	1	38	38	38	319	8%	33%	32%	22%	753.6
vpm1	7	**7**	IG	8	51	61	61	378	3%	80%	7%	7%	1402.3
vpm2	7	**7**	IG	3	21	69	69	378	3%	76%	8%	11%	982.0
modglob	9	29	IG	12	39	56	39	420	3%	79%	7%	8%	1807.3
p0548	29	49	IG	10	79	116	116	477	3%	48%	9%	39%	1803.7
danoint	6	210	IG	4	219	238	219	521	2%	79%	8%	9%	1808.8
gams	47	170	IG	1	170	170	170	556	2%	65%	12%	19%	1800.8
set1ch	4	33	IG	4	42	91	91	697	2%	68%	6%	22%	1843.7
set1al	3	40	IG	2	64	101	101	712	2%	76%	5%	15%	1813.6
set1cl	3	40	IG	2	64	101	101	712	2%	76%	5%	15%	1800.2
air01	184	367	IG	1	367	367	367	771	0%	95%	1%	1%	1893.0
gen	4	70	ID2	1	70	84	70	797	2%	70%	7%	19%	1821.7
mod015	4	70	ID2	1	70	84	70	797	2%	70%	7%	19%	1853.2
fixnet3	3	190	IG	1	190	221	209	878	2%	72%	6%	17%	1850.5
fixnet4	3	197	IG	2	209	221	209	878	2%	77%	6%	13%	1873.8
fixnet6	3	186	IG	2	209	221	209	878	1%	80%	5%	12%	2087.3
adrud	475	**475**	IG	1	475	475	475	998	0%	75%	17%	0%	750.9
$\sum$	1582	3007		693	3297	3737	3593	14556	4%	66%	13%	14%	41494.7

TABLE 5.4
Comparing integer programming branching rules.

Name	SIP without MAD			SIP with MAD			CPLEX		
	Gap	Nodes	Time	Gap	Nodes	Time	Gap	Nodes	Time
bell3a	96.6%	100000	236.0	67.9%	100000	288.2	0.0%	42446	54.9
bell3b	-	100000	142.1	-	100000	120.3	3.8%	100000	120.6
bell4	-	100000	137.4	2.1%	100000	164.5	3.2%	100000	119.0
bell5	6.3%	100000	184.0	2.7%	100000	153.6	6.1%	100000	113.6
noswot	-	100000	219.8	-	100000	195.2	10.3%	100000	228.1

Maximum infeasibility branching

bell3a	0.0%	33350	92.1	0.0%	55763	221.4	0.0%	19100	108.9
bell3b	0.0%	25909	308.4	0.0%	25706	301.4	0.0%	6537	70.8
bell4	1.5%	91240	1800.0	2.2%	96628	1800.0	0.0%	47001	710.5
bell5	0.1%	100000	806.4	0.1%	100000	824.0	0.1%	100000	586.9
noswot	27.9%	63474	1800.0	20.9%	90743	1800.0	7.5%	43038	1800.1

Strong branching

Maximum infeasibility branching (i.e., branching on a variable that is closest to 0.5) and *strong branching* (cf. [1]) are two standard branching strategies for solving mixed integer programs. The other two branching rules result from extending these two methods by our idea of branching on a variable that belongs to the border first. The limit of the computation was 1,800 CPU seconds or 100,000 branch-and-bound nodes, whatever came first.

Table 5.4 summarizes our results. The version without MAD (MAD stands for

TABLE 5.5
Decomposing Steiner-tree packing problems.

Name	Original			Presolved		Cuts					B&B	
	rows	col	nz	col	nz	Init	Cov	2part	BCC	Pool	Nod	Iter
g353	81	48	276	48	276	336	19768	6311	240	32467	121	385
g444	114	39	253	37	251	756	11324	4251	156	22828	56	195
d677	324	183	984	174	975	3533	13005	7370	56	54768	25	59
d688	383	230	1350	223	1341	4506	10719	6871	37	32015	19	40
$\sum$	902	500	2863	482	2843	9131	54816	24803	489	142078	221	679

Name	Best Solutions				Heuristics at Root				Time				
	Lb	Ub	He	No	G	D1	D2	B	Cm	LP	Sep	Heu	Tot
g353	16	**16**	IG	4	23	23	23	81	5%	77%	11%	3%	180.7
g444	13	**13**	B	36	22	22	22	114	3%	80%	11%	3%	186.9
d677	7	79	IG	2	98	98	98	324	0%	97%	1%	0%	11168.4
d688	7	108	IG	3	134	140	134	383	0%	97%	1%	0%	10901.3
$\sum$	43	216		45	277	283	277	902	0%	97%	1%	0%	22437.2

the MAtrix Decomposition) uses the original SIP-code and the original branching rules; the MAD version branches on a variable from the border first. For comparison, we also list the results that can be obtained using CPLEX. The column labeled *gap* reports the duality gap on termination (- means that no feasible solution was found).

The *results* are mixed. When *strong branching* is used, CPLEX is best overall and SIP with MAD is basically always worst (note that for example noswot, the value of the LP-relaxation already provides the value of the optimal integer solution, and so the main difficulty here is to find a good feasible solution). If we branch on a most infeasible variable the situation changes a bit in favor of SIP with MAD. In fact, there are two examples where this strategy performs best in terms of the *gap*. Of course, our limited tests cannot provide a definite evaluation of our idea to branch on border variables, but we think that the results show that this idea might have the potential to become a good branching strategy for certain MIP problems.

5.3. The Steiner-tree packing problems. We also tested some problem instances for which we know in advance that they have bordered block diagonal from. The problems are integer programming formulations for Steiner-tree packing problems, a problem where, in some given graph, edge sets (so-called Steiner-trees), each spanning some given subset of the node set, have to be simultaneously packed in the graph under capacity restrictions on the edges, see Grötschel, Martin, and Weismantel [14]. Unfortunately, our branch-and-cut algorithm performs very badly on these examples, although we extended the time limit to 10,800 CPU seconds; see Table 5.5. One reason for that might be that the rows that are supposed to be in the border have less nonzero entries than those that are expected to be in the blocks. The heuristics and the LP solutions, however, tend to put the rows into the border that have the most nonzeros entries. The "natural decompositions" result in borders of sizes 22, 24, 71, and 82 for problems g353, g444, d677, and d688, respectively.

5.4. The equipartition problems. Our last test set consists of equipartition problems introduced by Nicoloso and Nobili [22]. These have been generated randomly prescribing a certain matrix density. We have modified our code to handle the additional equipartition constraint. The results are given in Table 5.6: We can solve all problems within 10 CPU seconds, and as was already known from Nicoloso and Nobili [22], random matrices of this type do not decompose well.

6. Summary and conclusions. We have shown in this paper that it is possible to decompose typical linear and integer programming matrices with up to 200 and more rows to proven optimality using a cutting plane approach based on polyhedral

TABLE 5.6
Equipartitioning matrices.

Name	Original			Presolved		Cuts					B&B	
	rows	col	nz	col	nz	Init	Cov	2part	BCC	Pool	Nod	Iter
m22	9	6	21	6	21	30	37	19	0	12	16	31
m25	9	6	23	5	22	44	6	20	0	18	13	24
m33	14	9	40	8	37	38	210	90	0	68	13	33
m34	14	9	41	9	41	53	181	111	0	69	22	53
m41	18	9	43	9	43	41	654	241	1	253	28	76
m44	18	9	55	8	51	89	333	122	0	133	25	56
m51	21	14	61	12	58	52	977	349	1	441	52	143
m54	21	14	90	12	83	93	424	211	0	278	34	77
m61	21	17	69	15	65	52	742	204	0	245	28	73
m64	21	17	108	15	100	106	417	285	0	376	40	89
m71	28	15	71	14	69	64	2353	713	0	1171	76	196
m74	28	15	128	14	122	135	823	575	0	714	88	171
m81	28	21	86	20	85	74	3169	979	0	1717	268	476
m84	28	21	177	19	167	172	475	392	617	347	49	116
$\sum$	278	182	1013	166	964	1043	10801	4311	619	5842	752	1614

Name	Best Solutions				Heuristics at Root				Time				
	Lb	Ub	He	No	G	D1	D2	B	Cm	LP	Sep	Heu	Tot
m22	5	**5**	G	1	5	9	9	5	6%	46%	20%	6%	0.1
m25	7	**7**	G	2	9	9	9	9	16%	33%	8%	8%	0.1
m33	6	**6**	G	4	8	10	10	8	8%	40%	17%	20%	0.3
m34	8	**8**	G	1	8	10	10	8	23%	28%	21%	4%	0.5
m41	8	**8**	G	1	8	14	14	10	10%	57%	12%	6%	0.9
m44	10	**10**	G	1	10	10	10	10	11%	35%	35%	5%	0.6
m51	11	**11**	G	11	15	15	15	15	9%	53%	18%	8%	2.1
m54	13	**13**	G	6	15	15	15	15	8%	40%	23%	17%	1.3
m61	9	**9**	G	4	11	11	11	11	14%	49%	22%	4%	1.3
m64	15	**15**	G	5	17	17	17	17	10%	33%	23%	22%	1.6
m71	12	**12**	G	5	16	16	16	16	13%	54%	17%	7%	5.0
m74	20	**20**	G	1	20	22	22	20	8%	33%	14%	35%	4.2
m81	14	**14**	IG	3	16	16	16	16	12%	47%	18%	10%	7.7
m84	22	**22**	*	18	28	28	28	28	7%	28%	23%	35%	3.4
$\sum$	160	160		63	186	202	202	188	11%	43%	19%	16%	29.2

investigations of the matrix decomposition problem. It turned out that a substantial number of, but not all, LPs decompose well into four blocks, while even many MIPs, as well as some of their transposes, can be brought into bordered block diagonal form with two blocks. We think that these results show a significant potential for methods that can exploit this structure to solve general MIPs. Our decomposition heuristics work well for small instances, but there is room for improvement for problems of large scale, in particular, if more than two blocks are considered.

Acknowledgments. We are grateful to the Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and the German Academic Exchange Service (DAAD) for supporting this work. We enjoyed the cooperation a lot and intend to continue. We want to thank Robert E. Bixby and CPLEX Optimization, Inc., who made the newest β-versions of `CPLEX` and the `CPLEX` Callable Library available to us, and Paolo Nobili for sending a collection of his equipartition problems.

REFERENCES

[1] R. E. BIXBY, *private communication*.

[2] C. C. CARØE AND R. SCHULTZ, *Dual Decomposition in Stochastic Integer Programming*, Preprint SC 96-46, Konrad Zuse Zentrum für Informationstechnik Berlin, 1996; also available online from http://www.zib.de/ZIBbib/Publications; Oper. Res. Lett., submitted.

[3] Y. CRAMA AND M. OOSTEN, *Models for machine-part grouping in cellular manufacturing*, Internat. J. Prod. Res., 34 (1996), pp. 1693–1713.

[4] D. DENTCHEVA, R. GOLLMER, A. MÖLLER, W. RÖMISCH, AND R. SCHULTZ, *Solving the unit commitment problem in power generation by primal and dual methods*, in Proc. 9th Conf. of the European Consortium for Math. in Industry (ECMI), Copenhagen, 1996, M. Bendsøe and M. Sørensen, eds., Teubner Verlag, Stuttgart, 1997.

[5] I. DUFF, A. ERISMAN, AND J. REID, *Direct Methods for Sparse Matrices*, Oxford Univ. Press, Oxford, UK, 1986.

[6] M. Ehrgott, *Optimierungsprobleme in Graphen unter Kardinalitätsrestriktionen*, Master's thesis, Dept. of Mathematics, Univ. Kaiserslautern, Kaiserslautern, Germany, 1992.

[7] C. E. Ferreira, A. Martin, C. C. de Souza, R. Weismantel, and L. A. Wolsey, *The node capacitated graph partitioning problem: A computational study*, Math. Programming, 81 (1998), pp. 229–256.

[8] C. E. Ferreira, A. Martin, and R. Weismantel, *Solving multiple knapsack problems by cutting planes*, SIAM J. Optim., 6 (1996), pp. 858–877.

[9] C. Fiduccia and R. Mattheyses, *A linear-time heuristic for improving network partitions*, Proc. 19th IEEE Design Automation Conference, Las Vegas, NV, 1982, pp. 175–181.

[10] K. A. Gallivan, M. T. Heath, E. Ng, J. M. Ortega, B. W. Peyton, R. J. Plemmons, C. H. Romine, A. H. Sameh, and R. G. Voigt, *Parallel Algorithms for Matrix Computations*, SIAM, Philadelphia, 1990.

[11] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.

[12] E. Gottlieb and M. Rao, *The generalized assignment problem: Valid inequalities and facets*, Math. Programming, 46 (1990), pp. 31–52.

[13] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.

[14] M. Grötschel, A. Martin, and R. Weismantel, *Packing Steiner trees: A cutting plane algorithm and computational results*, Math. Programming, 72 (1996), pp. 125–145.

[15] A. Gupta, *Fast and Effective Algorithms for Graph Partitioning and Sparse Matrix Ordering*, Tech. report RC 20496, IBM T. J. Watson Res. Center, Yorktown Heights, NY, 1996.

[16] C. Helmberg, B. Mohar, S. Poljak, and F. Rendl, *A spectral approach to bandwidth and separator problems in graphs*, Linear and Multilinear Algebra, 39 (1995), pp. 73–90.

[17] K. L. Hoffman and M. W. Padberg, *Solving airline crew-scheduling problems by branch-and-cut*, Mgmt. Sci., 39 (1993), pp. 657–682.

[18] ILOG CPLEX Division, *Using the CPLEX Callable Library*, Incline Village, NV, 1997. Information available online from http://www.cplex.com.

[19] V. Kumar, A. Grama, A. Gupta, and G. Karypis, *Introduction to Parallel Computing*, Benjamin-Cummings, Menlo Park, CA, 1994.

[20] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, B.G. Teubner, Stuttgart, and John Wiley & Sons, Chichester, 1990.

[21] A. Löbel, *Optimal Vehicle Scheduling in Public Transit*, Shaker Verlag, Aachen, 1998.

[22] S. Nicoloso and P. Nobili, *A set covering formulation of the matrix equipartition problem*, in System Modelling and Optimization, Proc. 15th IFIP Conf., Zürich, Sept. 1991, P. Kall, ed., Springer-Verlag, Berlin, Heidelberg, New York, 1992, pp. 189–198.

[23] P. Nobili and A. Sassano, *Facets and lifting procedures for the set covering polytope*, Math. Programming, 45 (1989), pp. 111–137.

[24] M. W. Padberg, *On the facial structure of set packing polyhedra*, Math. Programming, 5 (1973), pp. 199–215.

[25] A. Pothen, H. D. Simon, and K.-P. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal., 11 (1990), pp. 430–452.

[26] E. Rothberg and B. Hendrickson, *Sparse Matrix Ordering Methods for Interior Point Linear Programming*, Tech. report SAND96-0475J, Sandia National Laboratories, Albuquerque, NM, 1996.

[27] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley and Sons, Chichester, 1986.

[28] G. Sheble and G. Fahd, *Unit commitment literature synopsis*, IEEE Trans. Power Systems, 9 (1994), pp. 128–135.

SIAM J. Optim.
Vol. 9, No. 1, pp. 270–290

SEQUENTIAL STOPPING RULES FOR RANDOM OPTIMIZATION METHODS WITH APPLICATIONS TO MULTISTART LOCAL SEARCH[*]

WILLIAM E. HART[†]

Abstract. Sequential stopping rules are described for several stochastic algorithms that estimate the global minimum of a function. Stopping rules are described for pure random search and stratified random search. These stopping rules use an estimate of the probability measure of the ϵ-close points to terminate these algorithms when a specified confidence has been achieved. Numerical results indicate that these stopping rules require fewer samples and are more reliable than the previous stopping rules for these algorithms. These stopping rules can also be applied to multistart local search and stratified multistart local search. Numerical results on a standard test set show that these stopping rules can perform as well as Bayesian stopping rules for multistart local search.

Key words. optimization, pure random search, stratified random search, stopping rules, multistart

AMS subject classifications. 65K10, 60F99, 62L15

PII. S1052623494277317

1. Introduction.

Consider the global optimization problem in which we search for

$$y^* = \min_{x\in\Omega} f(x),$$

where $f : \Omega \to \mathbf{R}$ satisfies some regularity conditions and the global minimum y^* is assumed to be finite. In practice, we often need to account for the fact that numerical procedures can only produce approximate answers. Hence we consider the problem solved if, given $\epsilon > 0$, we find a solution $x \in \Omega_\epsilon$, where

$$\Omega_\epsilon = \{x \mid x \in \Omega, f(x) \le y^* + \epsilon\}.$$

Törn and Žilinskas [18] describe a variety of methods that have been used to generate sequential stopping rules for stochastic global optimization algorithms, particularly *pure random search* (PRS) and *multistart local search* (MS). PRS selects points $(\xi_1, \dots , \xi_n)$ from a common distribution in the domain Ω and estimates the global optimum with $Y_n \;=\; \min\{f(\xi_1), \dots , f(\xi_n)\}$. Only mild assumptions need to be made about f and Ω to ensure that $Y_n \overset{\text{a.s.}}{\to} y^*$. MS selects points $(\xi_1, \dots , \xi_n)$ from a common distribution and estimates the global optimum with $Y_n = \min\{f(L(\xi_1)), \dots , f(L(\xi_n))\}$, where $L : \Omega \to \Omega$ is a local search method that finds a local minimum given a point in the search domain.

Using ideas introduced by Zieliński [21], Boender and Rinnooy Kan [4] present a Bayesian decision-theoretic framework that is used to define stopping rules for MS. This approach assumes that the number of local minima and relative proportions

[*] Received by the editors November 14, 1994; accepted for publication (in revised form) August 19, 1997; published electronically December 23, 1998. This work was supported by the Applied Mathematical Sciences program, U.S. Department of Energy, Office of Energy Research, and was performed at Sandia National Laboratories, operated for the U.S. Department of Energy under contract DE-AC04-94AL85000.

http://www.siam.org/journals/siopt/9-1/27731.html

[†] Sandia National Labs, P.O. Box 5800, Albuquerque, NM 87185-1110 (wehart@cs.sandia.gov).

of the local minima are random variables for which an a priori distribution can be specified. A uniform prior distribution is chosen for these random variables, which is appropriate when the user knows little or nothing about the values of the random variables [4]. This framework uses a cost structure that associates a penalty for not sampling before all local optima have been sampled. Extensions proposed by Piccioni and Ramponi [15] and Boender and Rinnooy Kan [5] take into account the values of the objective function at the local optima. Betrò and Schoen [1, 2, 3] propose a related framework that models the function to be optimized as a stochastic process.

An alternative approach used to design stopping rules for stochastic optimization algorithms are asymptotic analyses of sampling statistics. The analysis of these rules assumes that a large number of samples have been collected, which may affect their empirical performance. However, these stopping rules do not need to make assumptions about the a priori distributions of parameters of the objective function.

Sequential stopping rules for PRS have been defined using statistics of extreme values to construct a posteriori confidence intervals for the estimate of the global optimum, which are used to stop the search [6, 7, 9, 10, 19]. These methods require only that f be properly regularized, but most are applicable only when there is a unique global optimum. Dorea [11] develops stopping rules for PRS using estimates of p_ϵ, the probability that the search procedure samples a point in Ω_ϵ in a single step. This estimator of p_ϵ is more conservative than the standard estimator $\bar{p}_\epsilon$, the fraction of points for which $Y_i \leq Y_n + \epsilon$. Thus Dorea's stopping rules require more samples to terminate than those based on $\bar{p}_\epsilon$. However, the estimator proposed by Dorea is much easier to compute; to update $\bar{p}_\epsilon$ all points for which $Y_i \leq Y_n + \epsilon$ must be maintained in a sorted order, while updating Dorea's estimator, requires only maintaining a subset of these points in the order in which they are sampled (so no ordering is required). Dorea's stopping rules require no assumptions about f beyond the conditions required to insure that $Y_n \overset{\text{a.s.}}{\to} y^*$.

In this paper we use asymptotic analyses of sampling statistics to derive stopping rules that are closely related to those described by Dorea [11]. We describe an alternative method to approximate p_ϵ that is nearly as inexpensive as the method described by Dorea. Further, we propose modified stopping rules that insure that n is large enough for the estimate of p_ϵ to be accurate. These modifications overcome a number of weaknesses of Dorea's stopping rules, including the failure to reliably terminate for certain functions even if $\bar{p}_\epsilon$ is used to estimate p_ϵ.

We then generalize these modified stopping rules to the *stratified random search* algorithm described by Ermakov, Zhigyavskii, and Kondratovich [12]. Stratified random search (SRS) partitions the search domain Ω into a finite set of subdomains, and samples are selected from every subdomain according to a fixed distribution. A comparison between the stopping rules for PRS and SRS describes conditions for which SRS terminates before PRS. Finally, we describe how these stopping rules can be used for MS and the *stratified multistart* algorithm (SMS) [20, 14]. Like SRS, SMS applies local search to points that are selected using partitioned random search.

2. PRS.

2.1. Review. Dorea [11] considers the following variant of PRS.

Algorithm A. Let $\xi_1, \xi_2, \ldots$ be i.i.d. random vectors with a common distribution G on Ω. Let $(X_1, Y_1), (X_2, Y_2), \ldots$ be defined by

```
Step 1.    $X_1 = \xi_1$ and $Y_1 = f(X_1)$
Step k+1.  if $f(\xi_{k+1}) \leq Y_k$ then $X_{k+1} = \xi_{k+1}$ and $Y_{k+1} = f(X_{k+1})$
           else $X_{k+1} = X_k$ and $Y_{k+1} = Y_k$
```

Dorea describes stopping rules for Algorithm A that depend on $p_\epsilon = G(\Omega_\epsilon)$. This value can be estimated by the fraction of samples that are within ϵ of the best function value sampled. Dorea's stopping rules use a lower bound on this fraction that is easily computed, $\rho_n(\epsilon)/n$. Formally,

$$\rho_n(\epsilon) = \sup\{k \mid \tau_k > 0, Y_{\tau_k} \leq Y_n + \epsilon\},$$

and for $j = 1, \ldots, n-1$, we define

$$\begin{aligned}\tau_{j+1} = \tau_{j+1}(n) &= \sup\{k \mid 1 \leq k < \tau_j, Y_k \neq Y_{\tau_j}\} \\ &= 0 \ \text{ if } \ \sup\{k \mid 1 \leq k < \tau_j, Y_k \neq Y_{\tau_j}\} = \emptyset\end{aligned}$$

with $\tau_1(n) = n$. The $\tau_i(n)$ are indices that indicate the samples just before a sample with lower function value occurred. That is, $Y_{\tau_{i+1}(n)} > Y_{\tau_{i+1}(n)+1} = \cdots = Y_{\tau_i(n)-1} = Y_{\tau_i(n)}$.

Because $\rho_n(\epsilon)$ is defined in terms of an index of τ_i, it may underestimate the number of samples that are within ϵ of the best. For example, suppose that samples are chosen with values 4,2,3 and 1. Then $\tau_1 = 4$, $\tau_2 = 2$, and $\tau_3 = 1$. Thus $\rho_n(2) = 2$, although the number of samples within two of the best samples is three.

Dorea defines the following stopping rules for Algorithm A. Let ϵ be the desired accuracy of the best solution found and let $1-\beta$ be the required probability of success.

Stopping rule 1. For given $\epsilon > 0$ and $\beta \in]0, 1[$, terminate Algorithm A

1a) for $n \geq 2$ such that

$$1 - \left(1 - \frac{\rho_n(\epsilon)}{n}\right)^n \geq 1 - \beta;$$

1b) for $n \geq 2$ whenever a value Y_{n-m} has been repeated for m steps, so $Y_{n-m} = Y_{n-m+j}$, $j = 1, \ldots, m$, and

$$1 - \left(1 - \frac{\rho_{n-m}(\epsilon)}{n-m}\right)^n \geq 1 - \beta.$$

Dorea shows that if rule 1a is applied then

$$P(Y_n - y^* \leq \epsilon) \gtrsim 1 - \beta,$$

and if rule 1b is applied then

$$P(Y_n - y^* \leq \epsilon \mid Y_{n-m} = Y_{n-m+j}, j = 1, \ldots, m) \gtrsim 1 - \beta.$$

Dorea justifies these stopping rules by showing that for sufficiently large n, $\rho_n(\epsilon)/n$ is less than p_ϵ with high probability. Table 1 shows the performance of PRS for stopping rules 1a and 1b when applied to $f(x) = x$. If these stopping rules worked perfectly, then the percentage of sequences that were terminated would equal β in each column of Table 1.

Unfortunately, these empirical results do not generally confirm this prediction. Only rule 1a comes close when ϵ is very small. There are several reasons for this. First, $\rho_n(\epsilon)/n$ may severely underestimate p_ϵ, thereby forcing PRS to search much longer than necessary to find an ϵ-close point. For example, consider $f(x)$ and suppose $\xi_1 = \epsilon$. For small ϵ, the probability of sampling a point less than ξ_1 is small. However, for small β rule 1a will not terminate until at least one other point less than ξ_1 is

Table 1

Performance of PRS with stopping rules 1a *and* 1b *on* $f(x) = x$ *for different values of* ϵ *and* β*. Results are averages for* 1000 *random sequences of samples, showing average number of samples until termination and percentage of sequences which were terminated with* $Y_n \le y^* + \epsilon$*.*

	Rule 1a			Rule 1b		
	β: 0.025	0.05	0.10	0.025	0.05	0.10
ϵ: 0.01	35962 (0.98)	12045 (0.90)	11978 (0.87)	19 (0.10)	11 (0.07)	5 (0.05)
0.02	13605 (0.98)	2353 (0.90)	2170 (0.84)	13 (0.16)	8 (0.12)	5 (0.08)
0.05	5112 (0.94)	1023 (0.85)	992 (0.77)	9 (0.27)	6 (0.22)	4 (0.18)
0.1	2483 (0.88)	528 (0.80)	500 (0.70)	6 (0.36)	4 (0.32)	3 (0.29)

sampled. The expected number of iterations until a point less than ξ_1 is sampled is $\lceil 1/\epsilon \rceil$. As a result, the "luckier" the sampling is in earlier iterations, the greater the number of iterations the algorithm requires.

Also, $\rho_n(\epsilon)/n$ may be larger than p_ϵ for small n. This can cause these stopping rules to terminate prematurely. For example, consider $f(x) = x$ on $\Omega = [0, 1]$. If $|\xi_2 - \xi_1| \le \epsilon$ and $\xi_2 \ne \xi_1$, then rule 1a will terminate. If $\epsilon \in [0, 1/2]$, then the probability that $\xi_1 > \epsilon$, $\xi_2 > \epsilon$, and $|\xi_2 - \xi_1| \le \epsilon$ is $\epsilon(2 - 3\epsilon)$, which can be as high as $1/3$ when $\epsilon = 1/3$. Note that in this example $\rho_2(\epsilon)/2 = \bar{p}_\epsilon$, so this weakness of this stopping rule exists even when the more accurate estimator $\bar{p}_\epsilon$ is used!

Finally, note that rule 1b is much less reliable than rule 1a. This can be attributed to the fact that rule 1b estimates p_ϵ with $\rho_{n-m}(\epsilon)/(n - m)$. For small n, this may be quite different from $\rho_n(\epsilon)/n$, which is used by rule 1a to estimate p_ϵ.

2.2. Modified stopping rules. To account for the two problems described in the previous section, we propose stopping rules for Algorithm A that differ from rules 1a and 1b in two ways. First, they estimate p_ϵ with $\hat{\rho}_n(\epsilon)/n$, which includes the number of points Y_i that are within ϵ of Y_n for $i > \tau_2$. Formally, $\hat{\rho}_n(\epsilon) = \rho_n(\epsilon) + \Gamma_n(\epsilon)$,

$$\Gamma_n(\epsilon) = \begin{cases} |\{Y_i \mid Y_i \le Y_n + \epsilon, i = \tau_2 + 1, \dots, n - 1\}| & \text{if } \tau_2 + 1 < n, \\ 0 & \text{otherwise.} \end{cases}$$

This estimate of p_ϵ is more accurate than $\rho_n(\epsilon)/n$ and should therefore improve the performance of these stopping rules. Furthermore, $\hat{\rho}_n(\epsilon)$ requires only a small amount of additional computation and no additional memory requirements.

Second, these stopping rules include an additional term that ensures that n is sufficiently large for p_ϵ to be reliably estimated. The additional term ensures that $P(|\hat{\rho}_n(\epsilon)/n - p_\epsilon| \le \delta)$ is large. Consequently, these rules have an additional parameter δ that reflects how confident our estimate of p_ϵ must be before we terminate. This analysis applies equally well when $\bar{p}_\epsilon$ is used to estimate p_ϵ, so this additional term should allow a stopping rule that uses $\bar{p}_\epsilon$ to terminate reliably.

Let $\Phi(x)$ be the standard normal distribution function

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2} dy.$$

Consider the following stopping rules for Algorithm A.

Stopping rule 2. For given $\epsilon > 0$, $\delta > 0$ and $\beta \in]0, 1[$, terminate Algorithm A

2a) for $n \geq 2$ such that

$$\Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n}) - (1 - \hat{\rho}_n(\epsilon)/n)^n \geq 1 - \beta;$$

2b) for $n \geq 2$ whenever a value Y_n has been repeated for m steps, so $Y_n = Y_{n-j}$, $j = 1, \dots, m$, and

$$\Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n}) - (1 - \hat{\rho}_n(\epsilon)/n)^{n+m} \geq 1 - \beta;$$

2c) for $n \geq 2$ such that

$$\Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n}) - (1 - \bar{p}_\epsilon)^n \geq 1 - \beta.$$

The analysis in section 2.2.1 justifies the application of these rules. In section 2.2.2, we discuss the relative merits of rules 1 and 2.

2.2.1. Analysis. Let $Z_j = f(\xi_j)$ for $j = 1, \dots, n$. The Z_j's are i.i.d. random variables with a common distribution

$$F(x) = P(Z_1 \leq x) = P(f(\xi_1) \leq x) = \int_{\{u | f(u) \leq x\}} dG(u).$$

Let $(Z_{(1,n)}, \dots, Z_{(n,n)})$ be the order statistics of $(Z_1, \dots, Z_n)$. Thus $Z_{(j,n)}$ is the jth smallest element in $(Z_1, \dots, Z_n)$, so $Z_{(1,n)} = \min\{Z_1, \dots, Z_n\}$ and $Z_{(n,n)} = \max\{Z_1, \dots, Z_n\}$. For $\epsilon \geq 0$, let

$$\begin{aligned}\gamma_n(\epsilon) &= \sup\{k \mid Z_{(k,n)} \leq y^* + \epsilon\} \\ &= 0 \text{ if } \sup\{k \mid Z_{(k,n)} \leq y^* + \epsilon\} = \emptyset.\end{aligned}$$

Since the Z_j are i.i.d. random variables and $P(Z_1 \leq y^* + \epsilon) = p_\epsilon$, $\gamma_n(\epsilon)$ is binomially distributed with parameters n and p_ϵ.

We first show how to use rule 2a to bound $P(Y_n - y^* \leq \epsilon, |\gamma_n(\epsilon)/n - p_\epsilon| \leq \delta)$ from below. This bound ensures that our estimate of p_ϵ is sufficiently large, as well as ensuring that the sample we terminate with is likely to be in Ω_ϵ. Note that

$$\begin{aligned}&P(Y_n - y^* \leq \epsilon, |\gamma_n(\epsilon)/n - p_\epsilon| \leq \delta) \\ &\quad = 1 - P(Y_n - y^* > \epsilon \text{ or } |\gamma_n(\epsilon)/n - p_\epsilon| > \delta) \\ &\quad = 1 - P(Y_n - y^* > \epsilon) - P(|\gamma_n(\epsilon)/n - p_\epsilon| > \delta) \\ &\qquad + P(Y_n - y^* > \epsilon, |\gamma_n(\epsilon)/n - p_\epsilon| > \delta) \\ &\quad \geq 1 - P(Y_n - y^* > \epsilon) - P(|\gamma_n(\epsilon)/n - p_\epsilon| > \delta) \\ &\quad = 1 - (1 - p_\epsilon)^n - P(|\gamma_n(\epsilon)/n - p_\epsilon| > \delta).\end{aligned}$$

Since $\gamma_n(\epsilon)$ is binomially distributed, we can apply the DeMoivre–Laplace limit theorem [13] to get

$$\begin{aligned}P(|\gamma_n(\epsilon)/n - p_\epsilon| \leq \delta) &\approx \Phi\left(\frac{\delta\sqrt{n}}{\sqrt{p_\epsilon(1-p_\epsilon)}}\right) - \Phi\left(\frac{-\delta\sqrt{n}}{\sqrt{p_\epsilon(1-p_\epsilon)}}\right) \\ &\geq \Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n}).\end{aligned} \tag{1}$$

Thus

$$P(|\gamma_n(\epsilon)/n - p_\epsilon| > \delta) = 1 - P(|\gamma_n(\epsilon)/n - p_\epsilon| \leq \delta) \lesssim 1 - \left(\Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n})\right).$$

If we use $\hat{\rho}_n(\epsilon)/n$ to approximate p_ϵ, then we have

$$P(Y_n - y^* \le \epsilon, |\gamma_n(\epsilon)/n - p_\epsilon| \le \delta) \gtrsim \Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n}) - (1 - \hat{\rho}_n(\epsilon)/n)^n.$$

Rule 2a guarantees that this is greater than $1-\beta$. Similarly, if we use $\bar{p}_\epsilon$ to approximate p_ϵ, then rule 2c guarantees that

$$P(Y_n - y^* \le \epsilon, |\gamma_n(\epsilon)/n - p_\epsilon| \le \delta) \gtrsim (1 - (1 - \bar{p}_\epsilon)^n) \left(\Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n})\right).$$

The derivations of rule 2b is similar. When this rule is applied,

$$P(Y_n - y^* \le \epsilon \mid Y_n = Y_{n-j}, \quad j = 1, \dots, m, \text{ and } |\gamma_n(\epsilon)/n - p_\epsilon| \le \delta) \gtrsim 1 - \beta.$$

To derive this rule, we use the following result from Dorea [11],

$$P(Y_n - y^* \le \epsilon \mid Y_{n-m} = Y_{n-m+j}, j = 1, \dots, m) = 1 - (1 - p_\epsilon)^{n+m}, \tag{2}$$

and observe that

$$\begin{aligned} &P(Y_{\dot{n}} - y^* \le \epsilon \mid Y_{n-m} = Y_{n-m+j}, j = 1, \dots, m) \\ &\qquad = P(Y_n - y^* \le \epsilon \mid Y_n = Y_{n-j}, j = 1, \dots, m). \end{aligned}$$

Rules 2a and 2b bound $P(|\gamma_n(\epsilon)/n - p_\epsilon| \le \delta)$ and not $P(|\hat{\rho}_n(\epsilon)/n - p_\epsilon| \le \delta)$, although $\hat{\rho}_n(\epsilon)/n$ is actually used in the stopping rules. The bound on $\gamma_n(\epsilon)/n$ is used because $\gamma_n(\epsilon)$ is binomially distributed, which makes a simple bound easy to establish. In practice, these stopping rules should be reasonable since $\hat{\rho}_n(\epsilon)/n$ is an approximation of $\gamma_n(\epsilon)/n$. The following proposition justifies the use of $\hat{\rho}_n(\epsilon)/n$ to approximate p_ϵ. This proposition is an extension of the proposition in Dorea [11] because this proposition applies when $F(f(x) \le y^*) > 0$. The proof of this proposition is in Appendix A.

PROPOSITION 1. *Let $\beta \in]0,1[$. Assume that for all $\epsilon > 0$, $p_\epsilon = P(f(x) \le y^* + \epsilon) > 0$. Then for small $\epsilon > 0$ we have*

$$\lim_{\eta \downarrow 0} \lim_{n \to \infty} P\left(\frac{\hat{\rho}_n(\epsilon)}{n} \le p_\epsilon + \eta\right) = 1 \tag{3}$$

and

$$\lim_{\eta \downarrow 0} \lim_{n \to \infty} P\left(Y_n - y^* \le \epsilon, \delta_n(\epsilon) \ge \hat{\delta}_\eta(\epsilon)\right) = 1, \tag{4}$$

where

$$\delta_n(\epsilon) = \frac{\log \beta}{\log (1 - \hat{\rho}_n(\epsilon)/n)} \text{ and } \hat{\delta}_\eta(\epsilon) = \frac{\log \beta}{\log (1 - (p_\epsilon + \eta))}.$$

2.2.2. Comparison of rules for PRS. In this section we discuss the relative merits of rules 1 and 2. We delay an empirical comparison of these rules until section 4. Here we discuss properties of these rules that are raised by the derivations in the previous section.

Rule 2b differs from rule 1b in the manner in which (2) is interpreted. Dorea [11] uses (2) to create rule 1b by approximating p_ϵ with $\rho_{n-m}(\epsilon)/(n-m) = \rho_{\tau_2+1}(\epsilon)/(\tau_2 + 1)$. The advantage of this interpretation is that this value needs to be calculated only when a different value of Y_n is found. Our interpretation approximates p_ϵ with $\hat{\rho}_n(\epsilon)/n$. This value needs to be calculated at every iteration, but this interpretation allows us to combine $\Gamma_n(\epsilon)$ with the calculation of $\rho_n(\epsilon)$.

As we noted at the start of section 2.2, the use of $\hat{\rho}_n(\epsilon)/n$ to approximate p_ϵ should improve the performance of rule 2 over rule 1 simply because it is a better estimate of p_ϵ than $\rho_n(\epsilon)/n$. In addition, this approximation avoids certain worst case scenarios in the application of rule 1. For example, on a constant function, rule 1a fails to terminate if $\beta < 1/4$. After $n \geq 2$ samples, $\tau_1 = n$ and $\tau_2 = 0$, so $\rho_n(\epsilon)/n = 1/n$ for all $\epsilon > 0$. Rule 1a terminates if $1 - (1 - 1/n)^n \geq 1 - \beta$. Now $1 - (1 - 1/n)^n$ monotonically increases to $1 - 1/e$ beginning at $3/4$ for $n = 2$. Thus if $\beta < 1/4$, then $1-(1-1/n)^n < 1-\beta$ for all $n \geq 2$. Rule 1b avoids this difficulty by using the number of repetitions to terminate. For constant functions, $\hat{\rho}_n(\epsilon)/n = 1 = p_\epsilon$, so rule 2 also terminates successfully. More generally, the estimators used by rule 2 are guaranteed to have a nonzero probability of terminating because they take into consideration the number of samples that are within ϵ of the best point sampled so far; the difference between the estimators used is how they update their estimate of p_ϵ when a new best point is sampled.

The previous example also serves to illustrate the need for the confidence factor δ. If the confidence factor was not added to rule 2, then it might be misled into thinking that a function is constant after only two samples. In fact, this can happen to rule 1b because it does not include this confidence factor even if $\bar{p}_\epsilon$ was used to estimate p_ϵ.

This example also points to a strength of rule 2b relative to rule 2a. For small p_ϵ, it is not uncommon that a sample in Ω_ϵ occurs very infrequently. Consequently, the sampling procedure is likely to sample for a very long time before sampling twice in Ω_ϵ. In this case, rule 2b can use the number of repetitions after the first sample from Ω_ϵ to terminate. By contrast, the values for rule 2a converge to a constant that is likely to be smaller than $1 - \beta$ during the interval between samples from Ω_ϵ. Thus, it may take many samples from Ω_ϵ before rule 2a terminates.

Finally, note that the lower bound for $P(|\gamma_n(\epsilon)/n - p_\epsilon| \leq \delta)$ defined in (1) is conservative, so rule 2 may run longer than necessary, especially when p_ϵ is far from $1/2$. The bound

$$(5) \qquad P(|\gamma_n(\epsilon)/n - p_\epsilon| \leq \delta) \geq \Phi\left(\frac{\delta\sqrt{n}}{\sqrt{p'_n(1-p'_n)}}\right) - \Phi\left(\frac{-\delta\sqrt{n}.}{\sqrt{p'_n(1-p'_n)}}\right),$$

where

$$p'_n = \frac{1}{2n} + \left(1 - \frac{1}{n}\right)\hat{\rho}_n(\epsilon),$$

is less conservative. This lower bound allows the current estimate of p_ϵ to modify the stopping rule, which allows the stopping rule to terminate PRS earlier. This may, however, affect the reliability of the stopping rules when $\hat{\rho}_n(\epsilon)/n$ is a poor approximation to p_ϵ, so the more conservative approximation was chosen for rule 2.

3. SRS.

The following algorithm is a variant of the SRS algorithm considered by Ermakov, Zhigyavskii, and Kondratovich [12].

ALGORITHM B. Let $\Omega = \bigcup_{i=1}^{K} \Omega_i$ such that $\Omega_i \cap \Omega_j = \emptyset$ if $j \neq i$. Let ξ_j^i be i.i.d. random vectors with a common distribution G on Ω such that $\xi_j^i \in \Omega_i$. Let $(X_1^i, Y_1^i), (X_2^i, Y_2^i), \ldots$ for $i = 1, \ldots, K$, and $(X_1, Y_1), (X_2, Y_2), \ldots$ be defined by the following.

```
Step 1.
  for $i$ in $1:K$
    $X_1^i = \xi_1^i$ and $Y_1^i = f(X_1^i)$
  endfor
  $X_1 = \arg\min_{i=1,\dots,K} f(\xi_1^i)$ and $Y_1 = f(X_1)$
Step $k+1$.
  for $i$ in $1:K$
    if $f(\xi_{k+1}^i) \le Y_k^i$ then $X_{k+1}^i = \xi_{k+1}^i$ and $Y_{k+1}^i = f(X_{k+1}^i)$
                       else $X_{k+1}^i = X_k^i$ and $Y_{k+1}^i = Y_k^i$
  endfor
  $X_{k+1} = \arg\min_{i=1,\dots,K} f(X_{k+1}^i)$ and $Y_{k+1} = f(X_{k+1})$
```

Ermakov, Zhigyavskii, and Kondratovich [12] analyze Algorithm B when G is the uniform distribution and $\mu(\Omega_1) = \cdots = \mu(\Omega_K)$, where μ is the Lebesgue measure on Ω. Their analysis compares the performance of SRS to PRS. Using a simple extension of their analysis, it can be shown that SRS is probabilistically more powerful than PRS. Specifically, they prove the following proposition.

Proposition 2. *Let A_n and B_n be the solutions generated by Algorithm A and Algorithm B, respectively, after n iterations. If $G(\Omega_1) = \cdots = G(\Omega_K)$, then $P(A_n - y^* \le \epsilon) \le P(B_n - y^* \le \epsilon)$.*

3.1. Stopping rules. Let $p_\epsilon^i = G(\Omega_i \cap \Omega_\epsilon)/G(\Omega_i)$. To approximate p_ϵ^i after n iterations, we use $\bar{p}_\epsilon^i$, the fraction of samples in partition i for which $Y_j^i \le Y_n^i + \epsilon$, and $\hat{\rho}_n^i(\epsilon)/n$, where $\hat{\rho}_n^i(\epsilon) = \rho_n^i(\epsilon) + \Gamma_n^i(\epsilon)$,

$$\rho_n^i(\epsilon) = \begin{cases} 0 & \text{if } \sup\{k \mid \tau_k^i > 0, Y_{\tau_k^i}^i \le Y_n + \epsilon\} = \emptyset, \\ \sup\{k \mid \tau_k^i > 0, Y_{\tau_k^i}^i \le Y_n + \epsilon\} & \text{otherwise,} \end{cases}$$

$$\Gamma_n^i(\epsilon) = \begin{cases} \left|\{Y_j^i \mid Y_j^i \le Y_n + \epsilon, j = \tau_2^i + 1, \dots, n-1\}\right| & \text{if } \tau_2^i + 1 < n, \\ 0 & \text{otherwise,} \end{cases}$$

and for $j = 1, \dots, n-1$, we define

$$\begin{aligned} \tau_{j+1}^i = \tau_{j+1}^i(n) &= \sup\{k \mid 1 \le k < \tau_j^i, Y_k^i \ne Y_{\tau_j^i}^i\} \\ &= 0 \ \text{ if } \ \sup\{k \mid 1 \le k < \tau_j^i, Y_k^i \ne Y_{\tau_j^i}^i\} = \emptyset \end{aligned}$$

with $\tau_1^i(n) = n$.

Consider the following stopping rules for Algorithm B.

Stopping rule 3. For given $\epsilon > 0$, $\delta > 0$ and $\beta \in]0,1[$, terminate Algorithm B

3a) for $n \ge 2$ such that

$$\left(\Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n})\right)^K - \prod_{i=1}^K (1 - \hat{\rho}_n^i(\epsilon)/n)^n \ge 1 - \beta;$$

3b) for $n \ge 2$ whenever a value of Y_n has been repeated for m steps, so $Y_n = Y_{n-j}, j = 1, \dots, m$, and

$$\left(\Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n})\right)^K - \prod_{i=1}^K (1 - \hat{\rho}_n^i(\epsilon)/n)^{n+m} \ge 1 - \beta;$$

3c) for $n \geq 2$ such that

$$\left(\Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n})\right)^K - \prod_{i=1}^{K}(1-\bar{p}_\epsilon^i)^n \geq 1-\beta.$$

Numerical comparisons of these rules are deferred until section 4. The remainder of this section presents the derivation of these rules.

Let $Z_j^i = f(\xi_j^i)$ for $j = 1, \ldots, n$ and $i = 1, \ldots, K$. Note that $Y_n^i = \min\{Z_1^i, \ldots, Z_n^i\}$. The Z_j^i are i.i.d. random variables with a common distribution given by

$$F_i(x) = P(Z_1^i \leq x) = G(\Omega_{x-y^*} \cap \Omega_i)/G(\Omega_i),$$

where

$$G(\Theta) = \int_{\{u \in \Theta\}} dG(u).$$

For $\epsilon \geq 0$, let

$$\begin{aligned}\gamma_n^i(\epsilon) &= \sup\{k \mid Z_{(k,n)}^i \leq y^* + \epsilon\} \\ &= 0 \ \text{ if } \ \{k \mid Z_{(k,n)}^i \leq y^* + \epsilon\} = \emptyset,\end{aligned}$$

since the Z_j^i are i.i.d. random variables, and $P(Z_1^i \leq y^* + \epsilon) = p_\epsilon^i$, $\gamma_n^i(\epsilon)$ is binomially distributed with parameters n and p_ϵ^i.

We first show how to use rule 3a to bound

$$P\left(Y_n - y^* \leq \epsilon \ \text{ and } \ \left|\gamma_n^i(\epsilon)/n - p_\epsilon^i\right| \leq \delta, i = 1, \ldots, K\right).$$

Note that

$$\begin{aligned}&P\left(Y_n - y^* \leq \epsilon \ \text{ and } \ \left|\gamma_n^i(\epsilon)/n - p_\epsilon^i\right| \leq \delta, i = 1, \ldots, K\right) \\ &\quad \geq 1 - P\left(Y_n - y^* > \epsilon\right) - \left(1 - P\left(\left|\gamma_n^i(\epsilon)/n - p_\epsilon^i\right| \leq \delta, i = 1, \ldots, K\right)\right) \\ &\quad \approx \left(\Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n})\right)^K - \prod_{i=1}^{K}(1-\hat{\rho}_n^i(\epsilon)/n)^n.\end{aligned}$$

Rule 3a guarantees that this is greater than $1-\beta$. Similarly, if $\bar{p}_\epsilon^i$ is used to approximate p_ϵ^i, then rule 3c guarantees that

$$\begin{aligned}&P\left(Y_n - y^* \leq \epsilon \ \text{ and } \ \left|\gamma_n^i(\epsilon)/n - p_\epsilon^i\right| \leq \delta, i = 1, \ldots, K\right) \\ &\quad \gtrsim \left(\Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n})\right)^K - \prod_{i=1}^{K}(1-\bar{p}_\epsilon^i)^n.\end{aligned}$$

The derivation of rule 3b is similar. When this rule is applied,

$$P\left(Y_n - y^* \leq \epsilon \mid Y_n = Y_{n-j}, j = 1, \ldots, m \ \text{ and } \ \left|\gamma_n^i(\epsilon)/n - p_\epsilon^i\right| \leq \delta, i = 1, \ldots, K\right) \gtrsim 1-\beta.$$

The derivation of rule 3b uses the following lemma, whose proof is in Appendix B.

Lemma 3.1.

$$P(Y_n - y^* \leq \epsilon \mid Y_n = Y_{n-j}, j = 1, \ldots, m) = 1 - \prod_{i=1}^{K}(1-p_\epsilon^i)^{m+n}.$$

Note that we could use $\gamma_n^i(\epsilon)$ to estimate p_ϵ^i. Recall that $\gamma_n^i(\epsilon)$ is binomially distributed with parameters n and p_ϵ^i. It follows from the strong law of large numbers that $\gamma_n^i(\epsilon)/n \overset{\text{a.s.}}{\to} p_\epsilon^i$. Since y^* is unknown and the $Z_{(k,n)}^i$'s are not recorded, we approximate y^* by Y_n and $Z_{(k,n)}^i$ by $Y_{\tau_k^i}$ and $\Gamma_n(\epsilon)$. Thus, we approximate $\gamma_n^i(\epsilon)/n$ by $\hat{\rho}_n^i(\epsilon)/n$. The following proposition verifies that this approximation is reasonable when n is large.

PROPOSITION 3. *Let $\beta \in]0,1[$. Assume that for all $\epsilon > 0$, $p_\epsilon = P(f(x) \le y^* + \epsilon) > 0$. Then for $i = 1, \dots, K$ and for small ϵ, we have*

$$\lim_{\eta \downarrow 0} \lim_{n \to \infty} P\left(\frac{\hat{\rho}_n^i(\epsilon)}{n} \le p_\epsilon^i + \eta\right) = 1, \tag{6}$$

$$\lim_{\eta \downarrow 0} \lim_{n \to \infty} P\left(\delta_n(\epsilon) \ge \hat{\delta}_\eta(\epsilon)\right) = 1, \tag{7}$$

and

$$\lim_{\eta \downarrow 0} \lim_{n \to \infty} P\left(Y_n - y^* \le \epsilon, \delta_n(\epsilon) \ge \hat{\delta}_\eta(\epsilon)\right) = 1, \tag{8}$$

where

$$\delta_n(\epsilon) = \frac{\log \beta}{\sum_{i=1}^K \log\left(1 - \hat{\rho}_n^i(\epsilon)/n\right)} \quad \textit{and} \quad \hat{\delta}_\eta(\epsilon) = \frac{\log \beta}{\sum_{i=1}^K \log\left(1 - (p_\epsilon^i + \eta)\right)}.$$

Proof. Let $y_i^* = \min_{x \in \Omega_i} f(x)$. If $y_i^* = y^*$ for a given i, then (6) follows from Proposition 1. Now suppose $y_i^* > y^*$. If $Y_n < y_i^*$, then $\hat{\rho}_n^i(\epsilon) = 0$. Consequently

$$P\left(\frac{\hat{\rho}_n^i(\epsilon)}{n} \le p_{\epsilon+\eta}^i\right) \ge P(Y_n < y_i^*) = P(Y_n - y^* < \eta_1),$$

where $\eta_1 = y_i^* - y^* > 0$. Thus (6) follows from $Y_n \overset{\text{a.s.}}{\to} y^*$. From the definitions of $\delta_n(\epsilon)$ and $\hat{\delta}_\eta(\epsilon)$, we have (7) and (8) by using (6) and the fact that $Y_n \overset{\text{a.s.}}{\to} y^*$. □

3.2. Comparison with PRS. Let n_A be the number of steps specified by stopping rule 2a for Algorithm A, and let n_B be the number of steps specified by stopping rule 3a for Algorithm B. If $G(\Omega_1) = \cdots = G(\Omega_K)$, then the following argument shows that $n_A \ge K n_B$.

Suppose that $\hat{\rho}_n(\epsilon)/n = p_\epsilon$, $\hat{\rho}_n^i(\epsilon)/n = p_\epsilon^i$ and n is sufficiently large that $\Phi(2\delta\sqrt{n}) - \Phi(-2\delta\sqrt{n}) \approx 1$. Then $n_A \ge K n_B$ if

$$\frac{\log \beta}{\log(1 - p_\epsilon)} \ge K \frac{\log \beta}{\sum_{i=1}^K \log(1 - p_\epsilon^i)},$$

which is true if

$$\sum_{i=1}^K \log(1 - p_\epsilon^i) \le K \log(1 - p_\epsilon).$$

Exponentiating both sides gives us

$$\prod_{i=1}^K (1 - p_\epsilon^i) \le (1 - p_\epsilon)^K.$$

Recall that

$$1 - p_\epsilon^i = 1 - \frac{G(\Omega_\epsilon \cap \Omega_i)}{G(\Omega_i)} \text{ and } 1 - p_\epsilon = 1 - G(\Omega_\epsilon) = 1 - \sum_{i=1}^{K} G(\Omega_\epsilon \cap \Omega_i).$$

Thus we have

$$\prod_{i=1}^{K} \left[1 - \frac{G(\Omega_\epsilon \cap \Omega_i)}{G(\Omega_i)} \right]^{1/K} \leq 1 - \sum_{i=1}^{K} G(\Omega_\epsilon \cap \Omega_i).$$

If $G(\Omega_i) = G(\Omega_j)$ for all $i, j = 1, \dots, K$, then we have

$$\prod_{i=1}^{K} \left[1 - KG(\Omega_\epsilon \cap \Omega_i)\right]^{1/K} \leq 1 - \sum_{i=1}^{K} G(\Omega_\epsilon \cap \Omega_i)$$

$$\prod_{i=1}^{K} \left[\frac{1}{K} - G(\Omega_\epsilon \cap \Omega_i) \right]^{1/K} \leq \frac{1}{K} \sum_{i=1}^{K} \left[\frac{1}{K} - G(\Omega_\epsilon \cap \Omega_i) \right],$$

which is true because of the arithmetic-geometric mean inequality.

Note that equality is only achieved when $p_\epsilon^1 = \cdots = p_\epsilon^K$. Also, if $G(\Omega_i) \neq G(\Omega_j)$ for some $i, j \in \{1, \dots, K\}$, then this inequality may not hold. For example, suppose $\Omega = \Omega_1 \cup \Omega_2$ and $G(\Omega_\epsilon \cap \Omega_1) = G(\Omega_\epsilon \cap \Omega_2) > 0$. If $G(\Omega_1)/G(\Omega) = \lambda$, then $G(\Omega_2)/G(\Omega) = 1 - \lambda$. Some simple algebra shows that for all $\lambda \in]0, 1[$

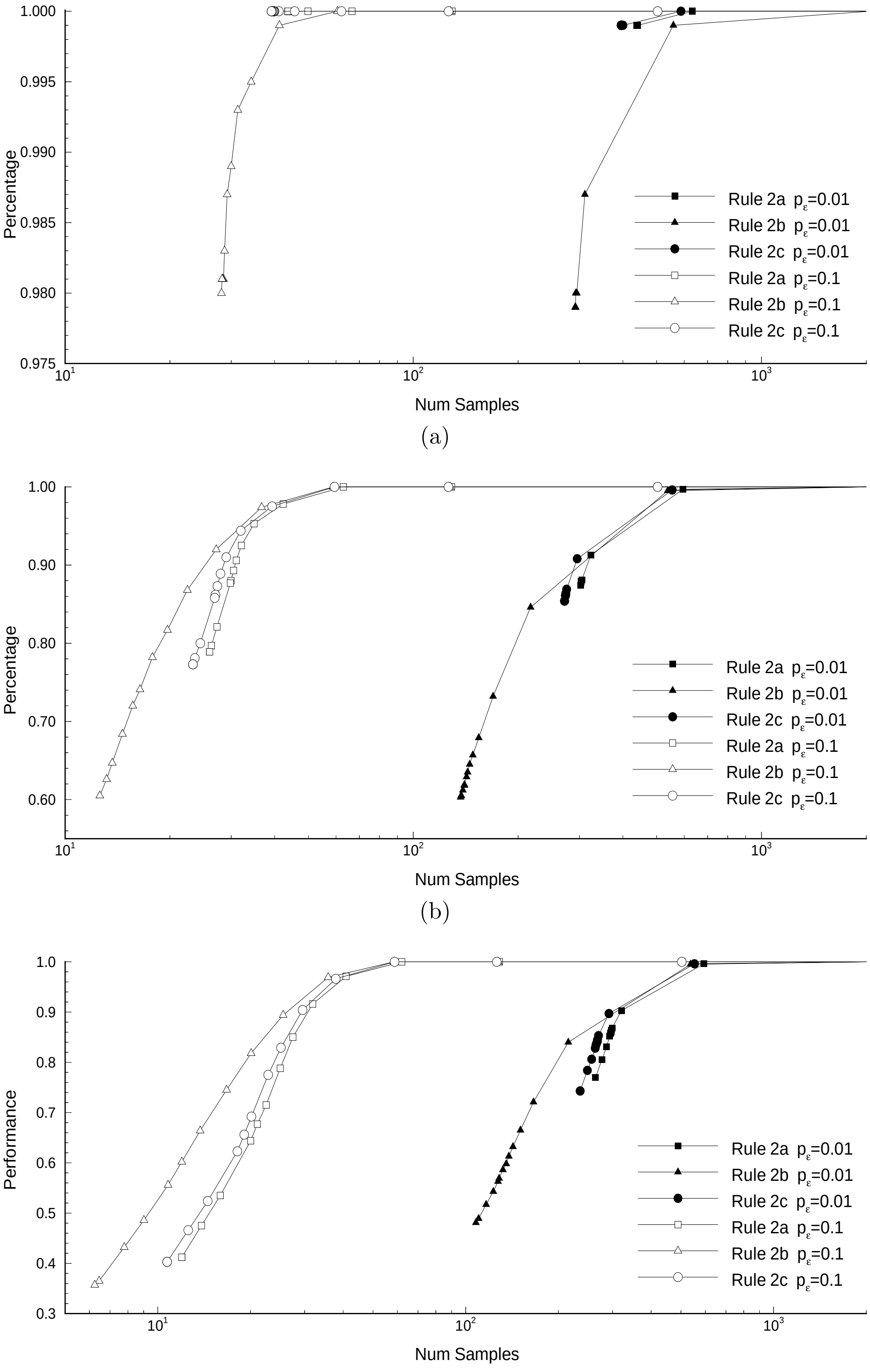
$$\left(1 - \frac{G(\Omega_\epsilon \cap \Omega_i)}{\lambda}\right)^{1/2} \left(1 - \frac{G(\Omega_\epsilon \cap \Omega_i)}{1 - \lambda}\right)^{1/2} > 1 - \sum_{i=1}^{2} G(\Omega_\epsilon \cap \Omega_i) = 1 - 2G(\Omega_\epsilon \cap \Omega_1).$$

which implies that $Kn_B > n_A$.

4. Numerical evaluation of rules for random search. In this section we numerically compare the performance of the stopping rules that we have described for PRS and SRS. Our experiments compare PRS and SRS on the three functions $f_1(x) = x^4$, $f_2(x) = x$, and $f_3(x) = 1 - (1-x)^4$ over the interval $[0, 1]$. The stopping rules that we have defined only utilize information about the distribution $F(x) = P(Z_1 \leq x)$. Consequently, these test functions can be used to make general predictions about the utility of these stopping rules. The function $f_1(x)$ is related to functions that have many solutions with values very near that of the global optimum. The function $f_3(x)$ is related to functions that have very few solutions with values very near that of the global optimum. The function $f_2(x)$ represents an intermediate between these extremes.

For rules 2 and 3, the value of δ chosen affects both the performance of the stopping rule (in terms of percentage of trials that successfully find an ϵ-optimal solution) as well as the number of samples used before terminating. Consequently, comparisons between these rules are affected by the particular values of δ chosen for the comparison. To make a fair comparison of the stopping rules, we performed repeated trials with each stopping rule using values of δ from $\{0.025, 0.05, 0.1, 0.15, \dots, 0.7\}$. For a given value of ϵ and β, 1000 trials were executed for each value of δ, giving a curve that marks the average probability that the stopping rule successfully terminates.

Figure 1 compares the average performance of rules 2a and 2b on the test functions when $\beta = 0.025$; qualitatively similar results were observed when these experiments



(a)

(b)

(c)

FIG. 1. *Summary of performance results for PRS:* (a) f_1, (b) f_2, *and* (c) f_3. *Curves represent differences as δ varies in the number of samples until termination and the average percentage of sequences that successfully terminated.*

Table 2

Performance of PRS with stopping rules 1a *and* 1b *on the three test functions for different values of ϵ and with $\beta = 0.025$. Results are averages for* 1000 *random sequences of samples, showing average number of samples until termination and percentage of sequences which were terminated with $Y_n \leq y^* + \epsilon$.*

	Rule 1a			Rule 1b		
	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$
ϵ: 0.01	36833 (1.00)	35692 (0.98)	31121 (0.85)	33 (0.14)	19 (0.10)	12 (0.07)
0.1	2773 (1.00)	2483 (0.88)	2041 (0.61)	10 (0.46)	6 (0.36)	3 (0.23)

were replicated for larger values of β. To allow comparisons between functions, we selected values of ϵ such that p_ϵ was the same for each of the functions (either 0.1 or 0.01). These graphs compare the logarithm of the number of samples with the percentage of trials that were terminated with $Y_n \leq y^* + \epsilon$. For each curve in the graphs, percentages closer to 1 correspond to values of δ closer to zero.

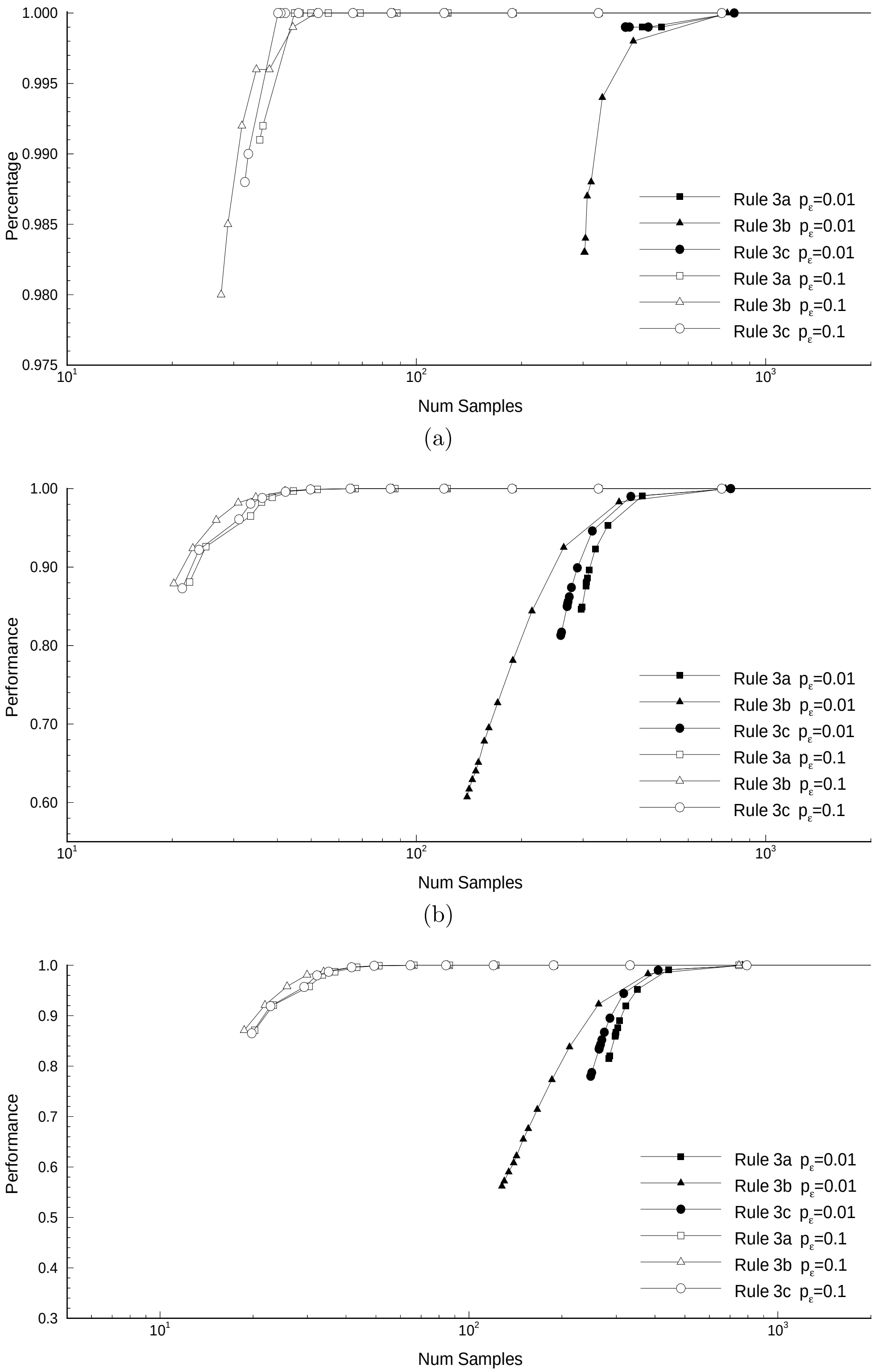
When comparing rules for a given value of p_ϵ, note that performance for a rule is better if its curve is left and above the curve of another rule. A careful inspection of this data shows that for small values of δ, rule 2b is better than rule 2a for $f_2(x)$ and $f_3(x)$, while for $f_1(x)$ rule 2a is better. However, as δ increases, Figure 1 clearly shows that the performance of rule 2b declines more quickly than the other rules. Since it may be difficult to determine a good value of δ a priori, this suggests that rule 2a and rule 2c are more robust in practice. Rule 2c is uniformly better than rule 2a and it is better than rule 2b when δ is small.

Figure 2 compares the performance of rules 3 on the test functions when $\beta = 0.025$. These results are qualitatively similar to those for rules 2. For small values of δ, rule 3b is better than rule 3a for all functions (including $f_1(x)$). When p_ϵ is small, as δ increases, the performance of rule 3b declines more quickly than rule 3a. However, for larger values of p_ϵ rule 3b remains better for all values of δ on functions $f_2(x)$ and $f_3(x)$. While rule 3b may perform better in certain contexts, the difference between rule 3a and 3b remains relatively small. Thus these experiments suggest that rule 3a is more robust than rule 3b. Finally, rule 3c is again uniformly better than rule 3a and is better than rule 3b when δ is small.

Figure 3 compares the performance of rules 2a and 3a for the three test functions. Figures 3a and 3c show that rule 3a has better performance than rule 2a on $f_2(x)$ and $f_3(x)$. Furthermore, rule 3a is more robust when p_ϵ is small on these functions. On $f_1(x)$, the performance of these stopping rules is roughly the same, though for large p_ϵ it appears that rule 2a is slightly better than rule 3a.

These observations confirm our analysis showing that SRS is more powerful than PRS. They also show that the reliability of the stopping rules decreases as the fraction of solutions with values near that of the global optimum decreases. These experiments show that the stopping rules are less reliable to changes in δ on function $f_3(x)$ than on function $f_1(x)$. One property of the experiments that the figures do not express is that bottom of the curves for rules 2a, 2c, 3a, and 3c represent the performance for a variety of values of δ. This confirms our intuitive argument that even modest efforts to ensure the reliability of the estimate for p_ϵ are valuable in avoiding premature termination of the stopping rules.

Finally, Table 2 shows the performance of rules 1a and 1b on the test functions. These results show that the performance of rules 2 and 3 is clearly superior to that of rules 1a and 1b.

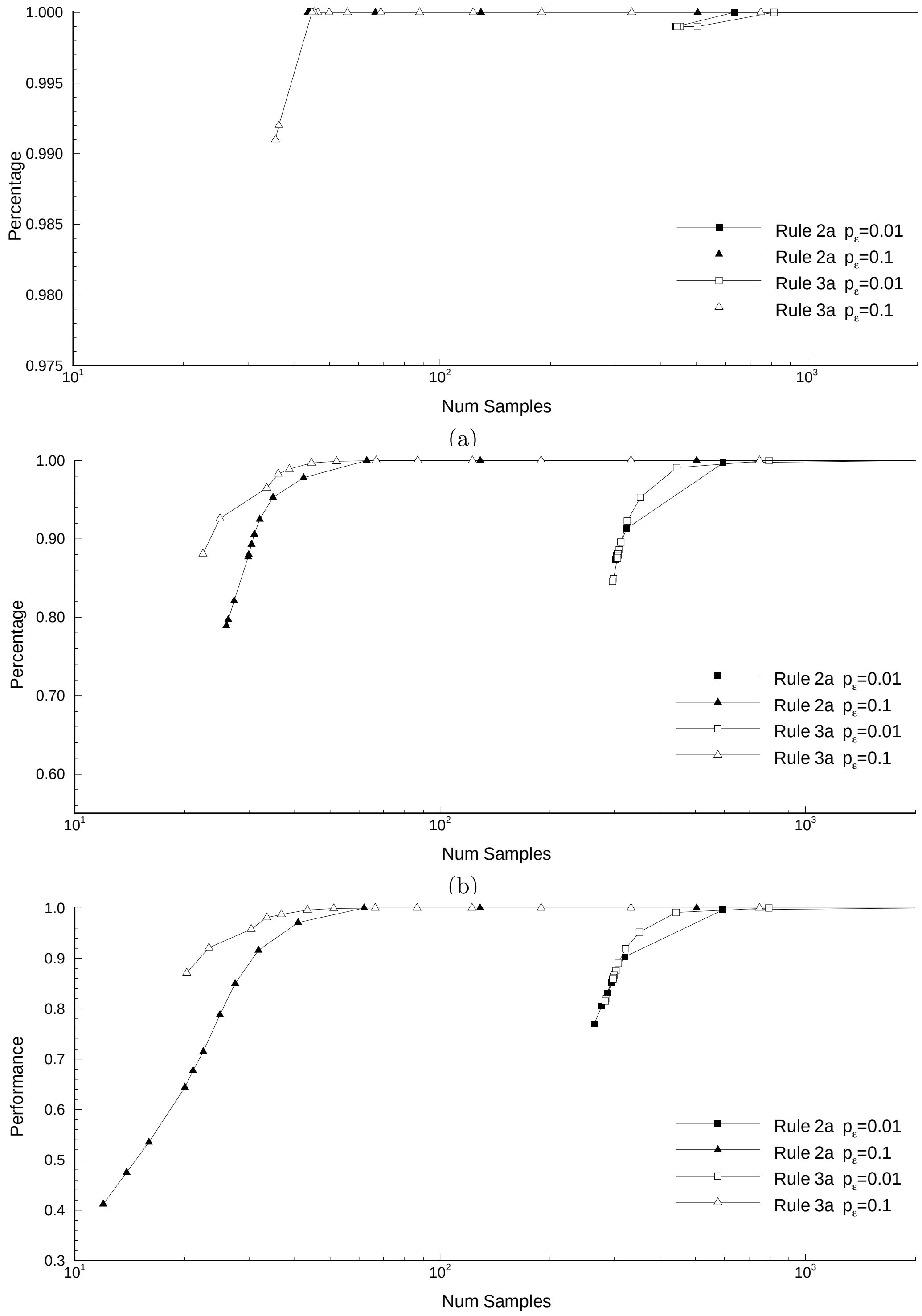


(a)

(b)

(c)

Fig. 2. *Summary of performance results for SRS:* (a) f_1, (b) f_2, *and* (c) f_3. *Curves represent differences as δ varies in the number of samples until termination and the average percentage of sequences that successfully terminated.*



(a)

(b)

(c)

FIG. 3. *A comparison of the performance of PRS and SRS:* (a) f_1, (b) f_2, *and* (c) f_3. *Curves represent differences as* δ *varies in the number of samples until termination and the average percentage of sequences that successfully terminated.*

5. MS. We now describe how stopping rules for PRS and SRS can be applied to multistart algorithms. MS algorithms iteratively apply a local search method to randomly selected samples in a domain. Let $L : \Omega \to \Omega$ be a local search algorithm which takes a sample in Ω and finds a sample that is a local minimum of f. The (pure) multistart algorithm (MS) can be defined as follows.

Algorithm C. Let $\xi_1, \xi_2, \ldots$ be i.i.d. random vectors with a common distribution G on Ω. Let $(X_1, Y_1), (X_2, Y_2), \ldots$ be defined by

```
Step 1. $X_1 = L(\xi_1)$ and $Y_1 = f(X_1)$
Step k+1.
  if $f(L(\xi_{k+1})) \leq Y_k$ then $X_{k+1} = L(\xi_{k+1})$ and $Y_{k+1} = f(X_{k+1})$
                         else $X_{k+1} = X_k$ and $Y_{k+1} = Y_k$
```

If we define $h = f \circ L$, then it becomes apparent that Algorithm C can be equated with the PRS algorithm applied to a modified function h. Consequently, stopping rules 2 and 3 can be used with Algorithm C.

Similarly, consider the SMS described by Morris and Wong [20, 14].

Algorithm D. Let $\Omega = \bigcup_{i=1}^{K} \Omega_i$ such that $\Omega_i \cap \Omega_j = \emptyset$ if $j \neq i$. Let ξ_j^i be i.i.d. random vectors with a common distribution G on Ω such that $\xi_j^i \in \Omega_i$. Let $(X_1^i, Y_1^i), (X_2^i, Y_2^i), \ldots$ for $i = 1, \ldots, K$, and $(X_1, Y_1), (X_2, Y_2), \ldots$ be defined by

```
Step 1.
  for $i$ in $1 : K$
    $X_1^i = L(\xi_1^i)$ and $Y_1^i = f(\xi_1^i)$
  endfor
  $X_1 = \arg\min_{i=1,\ldots,K} f(\xi_1^i)$ and $Y_1 = f(X_1)$
Step $k+1$.
  for $i$ in $1 : K$
    if $f(L(\xi_{k+1}^i)) \leq Y_k^i$ then $X_{k+1}^i = L(\xi_{k+1}^i)$ and $Y_{k+1}^i = f(X_{k+1}^i)$
                            else $X_{k+1}^i = X_k^i$ and $Y_{k+1}^i = Y_k^i$
  endfor
  $X_{k+1} = \arg\min_{i=1,\ldots,K} f(X_{k+1}^i)$ and $Y_{k+1} = f(X_{k+1})$.
```

Morris and Wong [20, 14] analyze Algorithm D. Using a proof technique analogous to the proof in Ermakov, Zhigyavskii, and Kondratovich [12], they show that SMS is probabilistically more powerful than MS. Like MS, SMS can be equated with SRS applied to a modified function h. Consequently, stopping rule 3 can be used with Algorithm D.

To illustrate the application of these stopping rules to MS and SMS, we optimized the standard global optimization test functions described by Dixon and Szegö [8, 18], which have previously been used to measure the performance of stopping rules for MS [1, 4, 5, 15]. This test set includes the following functions:

#	Description	# dimensions	# minima
1	Goldstein–Price	2	4
2	Branin	2	3
3	Hartman-3	3	3
4	Hartman-6	6	2
5	Shekel-5	4	5
6	Shekel-7	4	7
7	Shekel-10	4	10

Tables 3 and 4 show the performance of rules 2 and 3 on these functions. Since these are multidimensional functions, SRS partitioned the first dimension into four partitions. Results from Betrò and Schoen [1] are provided for comparison. The results

Table 3

Performance of MS with stopping rule 2 *and Betrò and Schoen's Bayesian rule. Rules* 2 *used values* $\beta = 0.025$, $\epsilon = 0.01$, *and* $\delta = 0.4$. *Results are averages for* 1000 *random sequences of samples (*100 *for the Bayesian stopping rule), showing average number of samples until termination and percentage of sequences for which were terminated with* $Y_n \leq y^* + \epsilon$.

Func	MS			
	Rule 2a	Rule 2b	Rule 2c	Bayesian
1	10.1 (1.00)	9.0 (1.00)	9.29 (1.00)	12.4 (1.00)
2	8.9 (1.00)	8.5 (1.00)	8.38 (1.00)	11.0 (1.00)
3	8.6 (1.00)	8.4 (1.00)	8.14 (1.00)	10.0 (1.00)
4	8.8 (1.00)	8.5 (1.00)	8.27 (1.00)	10.0 (1.00)
5	11.7 (0.99)	9.7 (0.99)	10.84 (0.99)	6.7 (0.97)
6	11.1 (1.00)	9.6 (1.00)	10.12 (1.00)	6.1 (0.99)
7	12.6 (1.00)	10.4 (1.00)	11.48 (1.00)	7.0 (0.96)

Table 4

Performance of SMS with stopping rule 3, *using* $\beta = 0.025$, $\epsilon = 0.01$, *and* $\delta = 1.0$. *Results are averages for* 1000 *random sequences of samples, showing average number of samples until termination summed over all partitions and percentage of sequences for which were terminated with* $Y_n \leq y^* + \epsilon$.

Func	SMS		
	Rule 3a	Rule 3b	Rule 3c
1	9.7 (0.99)	9.3 (0.99)	9.59 (0.99)
2	8.0 (1.00)	8.0 (1.00)	8.00 (1.00)
3	8.1 (1.00)	8.1 (1.00)	8.00 (1.00)
4	8.0 (1.00)	8.0 (1.00)	8.00 (1.00)
5	8.0 (1.00)	8.0 (1.00)	8.00 (1.00)
6	8.9 (0.99)	8.7 (0.99)	9.03 (0.99)
7	8.9 (0.99)	8.7 (0.99)	9.03 (0.99)

included here are for 1-sla with $a = 0$, $b = 5$, and $c = 0.5$; this configuration of their stopping rule had the best overall performance on the test set. The results in Betrò and Schoen [1] indicate that MS requires very few function evaluations to find the global optima, so high values of δ were chosen for MS and SMS.

All of the stopping rules had high reliability, and in all cases MS and SMS terminated within an average of 12 local searches. These results do not confirm the analysis of Morris and Wong [20, 14], which predicts that SMS will perform better than MS, though there may be little difference in the performance of these algorithms on functions for which so few local searches are needed. Also, these results do not clearly recommend rules 2 or 3 over the rule proposed by Betrò and Schoen. They do, however, suggest that these stopping rules can perform as well as Bayesian stopping rules.

6. Discussion. The stopping rules for PRS and SRS described in sections 2.2 and 3.1 retain the computational simplicity of the rules described by Dorea [11] while providing considerably better performance in both the number of samples required and the reliability of the final answer. Stopping rules for SRS have not been examined before. The experimental results confirm that the stopping rules for SRS can terminate more quickly than PRS, while retaining the same reliability.

The stopping rules that we have defined need no additional memory, and require a minimal amount of additional computation. They do require an additional parameter,

δ, which controls the desired accuracy of the estimate of p_ϵ. Our analysis of stopping rules that use $\bar{p}_\epsilon$ suggests that such a parameter is appropriate for stopping rules based on the type of asymptotic analysis that we have considered; stopping rules that use $\bar{p}_\epsilon$ without a confidence parameter may terminate prematurely. If the user has a priori estimates of the size of p_ϵ, then δ can be selected to match this information; small values of δ should be used when p_ϵ is small and larger values of δ should be used when p_ϵ is large. When no a priori estimates are available, rules using small values of δ will terminate conservatively. Our experiments recommend the use of rules 2a or 3a over rules 2b or 3b, because the former are more robust to choices of δ that are too large. The experimental results in section 5 also indicate that the stopping rules developed for PRS and SRS can be effectively applied to MS.

The empirical results in this paper provide preliminary evidence that the stopping rules we have proposed can be effectively applied. However, a more careful comparison is needed before making clear recommendations for one stopping rule over another. This is especially true when making comparisons between the proposed rules and the Bayesian methods; it is notoriously difficult to make comparisons between different types of stopping rules [3, 5]. More challenging test functions are needed to make such a comparison; the test functions described by Dixon and Szegö [8] have been used in most previous research on stopping rules for MS, but these functions are infamous for being an easy test set for global optimization. This comparison will also need to evaluate the sensitivity of the reliability of the stopping rules to their control parameters. Our experiments indicate that our stopping rules become less sensitive to the value of δ when p_ϵ is small, which is a reasonable assumption in many practical applications. However, selecting good values of δ may be more difficult than selecting control parameters for Bayesian methods.

To conclude, we note several potential advantages of the stopping rules we have proposed over Bayesian stopping rules. First, they can be applied to algorithms like SMS that do not perform a uniform sampling of the search domain. Second, their simplicity makes them easy to apply. Törn and Žilinskas [18, p. 93] note "the implementation (of Bayesian stopping rules) is rather complicated and therefore not easily available to potential users." Our stopping rules are simple and add little overhead to MS and SMS.

Third, these stopping rules can be applied when using local search algorithms that do not necessarily terminate at a local minimum. For example, the direct search methods like generalized pattern search [17] provide only asymptotic guarantees that the final solution is at a local minimum (or stationary point) of the objective function. The Bayesian stopping rules for MS [1, 4, 3, 15] implicitly require that the local search algorithm terminate at a local minimum, so it would be difficult to use Bayesian stopping rules with these types of local search algorithms.

Finally, our stopping rules may perform better than Bayesian stopping rules that utilize a priori for the number of local minima when optimizing functions that contain many local minima. Rinnooy Kan and Timmer [16, p. 76] briefly describe the application of a Bayesian stopping rule on a highly multimodal objective function. They observe that this stopping rule terminates prematurely because of the large number of local minima. The stopping rules that we have defined terminate based on an estimate of the ϵ-close points, which is invariant to the number of local minima in the function. Consequently, we do not expect this to adversely affect these stopping rules.

7. Appendix A: Proof of Proposition 1. From the definitions of $\delta_n(\epsilon)$ and $\hat{\delta}_\eta(\epsilon)$, we have (4) by using (3) and the fact that $Y_n \overset{\text{a.s.}}{\to} y^*$.

To prove (3), we consider two cases. First, suppose $P(f(x) \leq y^*) = 0$. Note that $p_\epsilon = F(y^* + \epsilon)$. The right continuity of F gives $\lim_{\eta \downarrow 0} p_{\epsilon+\eta} = p_\epsilon$. Moreover, since $p_\epsilon > 0$, $\epsilon > 0$ and $F(y^*) = 0$, we have p_ϵ strictly increasing on ϵ for small ϵ.

Now let $\epsilon > 0$ and $\eta > 0$. Let $\hat{\eta} > 0$ such that $p_{\epsilon+\hat{\eta}} = p_\epsilon + \eta$. For small $\epsilon > 0$, there exists $\eta_1 \in]0, \hat{\eta}[$ such that $p_{\epsilon+\hat{\eta}} > p_{\epsilon+\eta_1}$. Let $\eta_2 > 0$ such that $p_{\epsilon+\eta} \geq p_{\epsilon+\eta_1} + \eta_2$. Thus

$$\begin{aligned}P\left(\frac{\hat{\rho}_n(\epsilon)}{n} \leq p_\epsilon + \eta\right) &= P\left(\frac{\hat{\rho}_n(\epsilon)}{n} \leq p_{\epsilon+\hat{\eta}}\right) \geq P\left(\frac{\hat{\rho}_n(\epsilon)}{n} \leq p_{\epsilon+\eta_1} + \eta_2\right) \\ &\geq P\left(\frac{\hat{\rho}_n(\epsilon)}{n} \leq \frac{\gamma_n(\epsilon+\eta_1)}{n}, \left|\frac{\gamma_n(\epsilon+\eta_1)}{n} - p_{\epsilon+\eta_1}\right| < \eta_2\right).\end{aligned}$$

Now $\gamma_n(\eta_1 + \epsilon)$ is the number of samples that are within $\eta_1 + \epsilon$ of y^*. If $Y_n - y^* \leq \eta_1$, then $Y_n + \epsilon \leq y^* + \eta_1 + \epsilon$, so $\gamma_n(\eta_1 + \epsilon)$ is greater than the number of samples that are within $Y_n + \epsilon$ of y^*. This itself is an upper bound on the value of $\hat{\rho}_n(\epsilon)$. Thus we have $P(Y_n \leq y^* + \eta_1) \leq P(\hat{\rho}_n(\epsilon) \leq \gamma_n(\epsilon + \eta_1))$. Consequently,

$$P\left(\frac{\hat{\rho}_n(\epsilon)}{n} \leq p_\epsilon + \eta\right) \geq P\left(\left|\frac{\gamma_n(\epsilon+\eta_1)}{n} - p_{\epsilon+\eta_1}\right| < \eta_2, Y_n \leq y^* + \eta_1\right).$$

Since $\gamma_n(\epsilon + \eta_1)/n \overset{\text{a.s.}}{\to} p_{\epsilon+\eta_1}$ and $Y_n \overset{\text{a.s.}}{\to} y^*$, we have (3) by taking the proper limits.

Now suppose that $P(f(x) \leq y^*) = p_0 > 0$. We prove (3) by showing that $\hat{\rho}_n(\epsilon)/n \overset{\text{a.s.}}{\to} p_\epsilon$. Since $\hat{\rho}_n(\epsilon) = \rho_n(\epsilon) + \Gamma_n(\epsilon)$, it suffices to show that $\rho_n(\epsilon)/n \overset{\text{a.s.}}{\to} 0$ and $\Gamma(\epsilon)/n \overset{\text{a.s.}}{\to} p_\epsilon$.

Consider a sequence of samples from Ω for which there exists n_1 such that $Y_{n_1} = y^*$. For $n \geq n_1$, $\rho_n(\epsilon) = C$. Thus $\lim_{n\to\infty} \rho_n(\epsilon)/n = 0$. Because $p_0 > 0$, the set of sequences of samples from Ω for which $\not\exists n_1$ such that $Y_{n_1} = y^*$ has measure zero. Therefore, $\rho_n(\epsilon)/n \overset{\text{a.s.}}{\to} 0$.

Now let A_k be the set of sequences of samples from Ω for which $Y_{k-1} > Y_k = y^*$. For sequences in A_k, when $n > k$, $\Gamma_n(\epsilon)$ is binomially distributed with parameters $n-k$ and p_ϵ. Let $\bar{A}_k$ be the subset of A_k such that $\lim_{n\to\infty} \Gamma_n(\epsilon)/n = \lim_{n\to\infty} \Gamma_n(\epsilon)/(n - k) = p_\epsilon$ exists. Since $\Gamma_n(\epsilon)/(n-k) \overset{\text{a.s.}}{\to} p_\epsilon$, $P(\bar{A}_k) = P(A_k)$ and

$$P\left(\lim_{n\to\infty} \Gamma_n(\epsilon)/n = p_\epsilon\right) = \bigcup_{k=1}^{\infty} P(\bar{A}_k) = \bigcup_{k=1}^{\infty} P(A_k) = p_\epsilon \sum_{k=1}^{\infty} (1-p_\epsilon)^{k-1} = 1.$$

Thus $\Gamma_n(\epsilon)/n \overset{\text{a.s.}}{\to} p_\epsilon$.

8. Appendix B: Proof of Lemma 3.1.

Suppose that a certain value Y_n has been repeated for m steps of the algorithm. To estimate $P(Y_n = Y_{n-j}, j = 1, \ldots, m)$, note that Y_n has the distribution $H(x) = P(Y_n \leq x) = 1 - L(x)^n$, where $L(x) = \prod_{i=1}^{K}(1 - F_i(x))$. Thus

$$\begin{aligned}P(Y_n = Y_{n-j}, j = 1, \ldots, m) &= P(Z^i_{n-m+j} > Y_{n-m}, j = 1, \ldots, m) \\ &= \int_{y^*}^{\infty} L(x)^m dH(x) \\ &= -n \int_{y^*}^{\infty} L(x)^{m+n-1} L'(x) dx \\ &= -n \int_1^0 u^{m+n-1} du = \frac{n}{n+m}.\end{aligned}$$

Similarly,

$$P(Y_n = Y_{n-j}, j = 1, \ldots, m, Y_n > y^* + \epsilon) = \int_{y^*+\epsilon}^{\infty} L(x)^m dH(x) = \frac{n}{n+m} \prod_{i=1}^{K} (1 - p_\epsilon^i)^{m+n}.$$

It follows that

$$P(Y_n - y^* \leq \epsilon \mid Y_n = Y_{n-j}, j = 1, \ldots, m) = 1 - \prod_{i=1}^{K} (1 - p_\epsilon^i)^{m+n}.$$

Acknowledgments. I wish to thank John DeLaurentis for his many helpful discussions. I am also grateful for the comments from two anonymous reviewers that led to a much clearer presentation of these results.

REFERENCES

[1] B. Betrò and F. Schoen, *Sequential stopping rules for the multistart algorithm in global optimization*, Math. Programming, 38 (1987), pp. 271–286.

[2] B. Betrò and F. Schoen, *A stochastic technique for global optimization*, Comput. Math. Appl., 21 (1991), pp. 127–133.

[3] B. Betrò and F. Schoen, *Optimal and suboptimal stopping rules for the multistart algorithm in global optimisation*, Math. Programming, 57 (1992), pp. 445–458.

[4] C. Boender and A. Rinnooy Kan, *Bayesian stopping rules for multistart global optimization methods*, Math. Programming, 37 (1987), pp. 59–80.

[5] C. Boender and A. Rinnooy Kan, *On when to stop sampling for the maximum*, J. Global Optim., 1 (1991), pp. 331–340.

[6] L. de Haan, *Estimation of the minimum of a function using order statistics*, J. Amer. Statist. Assoc., 76 (1981), pp. 467–469.

[7] A. L. Dekkers and L. de Haan, *On the estimation of the extreme-value index and large quantile esimation*, Ann. Statist., 17 (1989), pp. 1795–1832.

[8] L. Dixon and G. Szegö, *The global optimization problem: An introduction*, in Towards Global Optimization 2, L. Dixon and G. Szegö, eds., North–Holland, Amsterdam, 1978, pp. 1–15.

[9] C. Dorea, *Limiting distribution for random optimization methods*, SIAM J. Control Optim., 24 (1986), pp. 76–82.

[10] C. Dorea, *Estimation of the extreme value and the extreme points*, Ann. Inst. Statist. Math., 39 (1987), pp. 37–48.

[11] C. Dorea, *Stopping rules for a random optimization method*, SIAM J. Control Optim., 28 (1990), pp. 841–850.

[12] S. Ermakov, A. Zhigyavskii, and M. Kondratovich, *Comparison of some random search procedures for a global extremum*, U.S.S.R. Comput. Math. Math. Phys., 29 (1989), pp. 112–117.

[13] W. Feller, *Introduction to Probability Theory and Its Applications*, John Wiley and Sons, New York, 1950.

[14] R. J. Morris and W. S. Wong, *Systematic choice of initial points in local search: Extensions and application to neural networks*, Inform. Process. Lett., 39 (1991), pp. 213–217.

[15] M. Piccioni and A. Ramponi, *Stopping rules for the multistart method when different local minima have different function values*, Optimization, 21 (1990), pp. 697–707.

[16] A. Rinnooy Kan and G. Timmer, *Stochastic global optimization methods - part* II*: Multi level methods*, Math. Programming, 39 (1987), pp. 57–78.

[17] V. Torczon, *On the convergence of pattern search methods*, SIAM J. Optim., 7 (1997), pp. 1–25.

[18] A. Törn and A. Žilinskas, *Global Optimization*, Lecture Notes in Comput. Sci. 350, Springer-Verlag, New York, 1989.

[19] M. R. Veall, *Testing for a global maximum in an econometric context*, Econometrica, 58 (1990), pp. 1459–1465.

[20] W. S. Wong and R. J. Morris, *A new approach to choosing initial points in local search*, Inform. Process. Lett., 30 (1989), pp. 67–72.

[21] R. Zieliński, *A statistical estimate of the structure of multiextremal functions*, Math. Programming, 21 (1981), pp. 348–356.

SIAM J. OPTIM.
Vol. 9, No. 2, pp. 291–315

A TRUST-REGION APPROACH TO NONLINEAR SYSTEMS OF EQUALITIES AND INEQUALITIES[*]

J. E. DENNIS, JR.[†], MAHMOUD EL-ALEM[‡], AND KAREN WILLIAMSON[§]

Abstract. In this paper, two new trust-region algorithms for the numerical solution of systems of nonlinear equalities and inequalities are introduced. The formulation is free of arbitrary parameters and possesses sufficient smoothness to exploit the robustness of the trust-region approach. The proposed algorithms are one-sided least-squares trust-region algorithms. The first algorithm is a single-model algorithm, and the second one is a multimodel algorithm where the Cauchy point computation is a model selection procedure.

Global convergence analysis for the two algorithms is presented. Our analysis generalizes to nonlinear systems of equalities and inequalities the well-developed theory for nonlinear least-squares problems.

Numerical experiments on the two algorithms are also presented. The performance of the two algorithms is reported. The numerical results validate the effectiveness of our approach.

Key words. fraction of Cauchy decrease, global convergence, multimodel algorithm, nonlinear systems, nonlinear least squares, one-sided least squares, system of inequalities, trust-region methods, active-set strategies

AMS subject classifications. 65K05, 49D37

PII. S1052623494276208

1. Introduction.

In this paper, we present two new trust-region algorithms for the numerical solution of a system of nonlinear equalities and inequalities defined by

$$\begin{aligned} c_i(x) &= 0, \quad i \in E, \\ c_i(x) &\le 0, \quad i \in I, \end{aligned} \tag{1.1}$$

where $c_i : \Re^n \to \Re$, $I \cup E = \{1, \ldots, m\}$, and $I \cap E = \emptyset$. In particular, we study trust-region methods for the following least-squares problem:

$$\min_{x \in \Re^n} \frac{1}{2} \left\{ \sum_{i \in E} c_i(x)^2 + \sum_{i \in I} [\max\{c_i(x), 0\}]^2 \right\}. \tag{1.2}$$

In practice, it is often useful to include weights on each term of the objective function (1.2), but here we omit them for simplicity.

Systems of nonlinear equalities and inequalities appear in a wide variety of problems in applied mathematics. These systems play a central role in the model formulation design and analysis of numerical techniques employed in solving problems arising in optimization, complementarity, and variational inequalities.

Best one-sided approximations have the form (1.2) (Taylor [35], Kaufman and Taylor [19], etc.). Another interest in problem (1.2) is when (1.1) is the constraint

[*] Received by the editors October 23, 1994; accepted for publication (in revised form) February 24, 1998; published electronically March 17, 1999. This research was supported by grants DOE DE-FG005-86ER25017, CRPC CCR-9120008, and AFOSR-F49620-9310212.

http://www.siam.org/journals/siopt/9-2/27620.html

[†] Department of Computational and Applied Mathematics and Center for Research on Parallel Computation, Rice University, P.O. Box 1892, Houston, TX 77251 (dennis@rice.edu).

[‡] Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt, and Center for Research on Parallel Computation, Rice University, P.O. Box 1892, Houston, TX 77251 (elalem@alex.eun.eg, mahmoud@rice.edu).

[§] TDA Research Inc., 12345 West 52nd Avenue, Wheat Ridge, CO 80033 (kaw@rice.edu).

set in a nonlinear programming problem. For this reason, it is important that we design our algorithms to handle any relationship between m and n, whereas, if we were considering only the pure one-sided approximation generalization of nonlinear least squares, then it would be reasonable to assume that $m \geq n$. Also, the general form of (1.2) we consider here is of interest in its own right (see Burke [5] and Burke and Han [6]).

Newton's method is a well-known and very powerful technique for solving nonlinear systems of equations. See, for example, Dennis and Schnabel [13]. Pshenichnyi [31], Robinson [32], and Daniel [12] extended Newton's method to nonlinear systems of equalities and inequalities. Robinson [32] generalized Newton's method to solve problems in the form find $x_\star$ such that $f(x_\star) \in K$, where K is a nonempty closed convex cone. Polyak [28] used gradient methods for solving problem (1.1).

Burke and Han [6] considered a Gauss–Newton approach to solving generalized inequalities; $C(x) \leq_K 0$, where C maps between normed linear spaces and $\leq_K$ denotes the partial order induced by the closed convex cone K. Burke and Ferris [7] considered an extension of the Gauss–Newton method to convex composite optimization. Using tools from nondifferentiable optimization, they were able to establish a local quadratic rate of convergence. By using a backtracking line search they were able to prove global convergence. Many authors, including Garcia-Palomares and Restuccia [16], Garcia-Palomares [15], and Burke [5], consider globally convergent algorithms for solving problem (1.1). None of these theories are based on a trust-region globalization strategy.

In this paper, we present two new trust-region-based algorithms for solving problem (1.1). By using an indicator matrix that will be presented in the next section, we are able to transform our problem into one that possesses sufficient smoothness to exploit the robustness of our algorithms. This allows us to use well-developed tools and algorithms that require differentiability. In addition to that, the proposed active set subproblems in this paper are much simpler than those proposed by Garcia-Palomares and Restuccia [16] and Burke [5]. When we present our algorithms, it will be clear that they are active set-type methods that try to identify the inequalities likely to be violated at a solution to (1.2). Based on this property, we plan in future research to use the ideas underpinning the algorithms developed here to develop an ℓ_2 trust-region active set algorithm for nonlinear programming.

The two algorithms we present in this paper are one-sided least-squares trust-region algorithms. The first one is a single-model algorithm. The second one is a multimodel algorithm, where the Cauchy point computation is a model selection procedure. Most minimization algorithms use a local quadratic model where the Hessian matrix may not be accurate. (See, e.g., Powell [30] and Toint [36].) Carter [9], on the other hand, studied the case when the gradient might be inaccurate and needs to be corrected during the step calculation. In this algorithm, we go all the way to a model in which the function value may even be wrong, and it may need to be corrected to find a step. See also Conn et al. [10].

We present global convergence results for the two algorithms. The two algorithms were tested and compared on some test problems. The results are presented.

The rest of the paper is organized as follows. In section 2, we establish the problem formulation and some notation. In section 3, we review the nonlinear least squares problem and the concept of a fraction of the Cauchy decrease. In section 4, we state some assumptions and prove a lemma that shows the required smoothness properties of the problem formulation. In sections 5 and 6, we describe our two trust-region algorithms. Section 7 is devoted to the global convergence theory for the first algorithm. In section 8, we present the global convergence theory for the multimodel algorithm. Section 9 contains our numerical results, and finally, we make some concluding remarks in section 10.

2. Preliminaries. It will be useful to establish some notation. Let $C(x) = (c_1(x), \ldots, c_m(x))^T$, and define the vector functions $C_E : \Re^n \to \Re^{|E|}$ to be the vector function whose components are $c_i(x)$ for $i \in E$ and $C_I : \Re^n \to \Re^{|I|}$ to be the vector function whose components are $c_i(x)$ for $i \in I$. Then, (1.1) can be written as

$$\begin{aligned} C_E(x) &= 0, \\ C_I(x) &\leq 0\,. \end{aligned}$$

We define a 0–1 diagonal indicator matrix $W(x) \in \Re^{m \times m}$ whose diagonal entries are

$$w_i(x) = \begin{cases} 1, & i \in E, \\ 1, & i \in I \text{ and } c_i(x) \geq 0, \\ 0, & i \in I \text{ and } c_i(x) < 0. \end{cases} \tag{2.1}$$

It is also useful to identify the square submatrix $W_I(x)$ whose diagonal entries $w_i(x)$ correspond to $i \in I$. Now, we define the functions

$$\Phi_E(x) = \frac{1}{2} C_E(x)^T C_E(x)\,, \tag{2.2}$$

$$\Phi_I(x) = \frac{1}{2} C_I(x)^T W_I(x) C_I(x)\,, \tag{2.3}$$

and

$$\Phi(x) = \Phi_E(x) + \Phi_I(x)\,. \tag{2.4}$$

The definition of $W(x)$ allows us to write $\Phi(x)$ as

$$\Phi(x) = \frac{1}{2} C(x)^T W(x) C(x),$$

and problem (1.2) can then be written as

$$\min_{x \in \Re^n} \Phi(x)\,.$$

It is easy to see that the function $t_+ = \max\{t, 0\}$ is continuous and that $(t_+)^2$ satisfies

$$\frac{d}{dt}\left(\frac{1}{2}\, t_+^2\right) = t_+\,.$$

Hence, $(t_+)^2 \in \mathcal{C}^1$. Thus, if each $c_i(x)$, for $i = 1, \ldots, m$, is continuously differentiable,

$$\begin{aligned} \nabla\Phi(x) &= \nabla\Phi_E(x) + \nabla\Phi_I(x) \\ &= C_E'(x)^T C_E(x) + C_I'(x)^T W_I C_I(x) \end{aligned} \tag{2.5}$$

is well defined and continuous. This allows us to write

$$\nabla \Phi(x) = C'(x)^T W(x) C(x).$$

Throughout the rest of the paper, the sequence of points generated by an algorithm will be denoted by $\{x_k\}$. Subscripted functions indicate that the function is evaluated at a particular point. For example, $W_k \equiv W(x_k)$, $C_k \equiv C(x_k)$, and so on. The expression $f \in Lip(S)$ is used to mean that the function f is Lipschitz continuous at every point of the set S. Finally, unless otherwise specified, all the norms used in this paper will be ℓ_2-norms.

3. Nonlinear least-squares trust-region algorithms.

The nonlinear least-squares problem is traditionally written only for equalities as

$$\min_{x \in \Re^n} \Phi_E(x), \tag{3.1}$$

where $\Phi_E(x)$ is given by (2.2).

A trust-region method for solving (3.1) is an iterative method that computes, at each iteration, a trial step s_k by minimizing a quadratic model of the objective function in the region in which we "trust" the model. First, we build a linear model of C_E around the current iterate x_k, namely, $C_E(x_k) + C'_E(x_k)s$. Then we compute a trial step s_k that (approximately) solves the trust-region subproblem

$$\begin{aligned} &\underset{s \in \Re^n}{\text{minimize}} \;\; m_k(s) \;=\; \frac{1}{2}\| \; C'_E(x_k)s + C_E(x_k) \; \|^2 \\ &\text{subject to } \|s\| \leq \Delta_k, \end{aligned}$$

where $\Delta_k > 0$ is the radius of the trust region.

The trust-region approach was first suggested by Levenberg [20]. Later, Marquardt [21] used a different formulation of this technique, and the method is now known as the Levenberg–Marquardt method. More details about problem (3.1) and the trust-region subproblem can be found in Moré [22] and Dennis and Schnabel [14].

The global convergence analysis for problem (3.1) has been well established. To insure global convergence, the step can be required to satisfy a fraction of the Cauchy decrease condition. The Cauchy step minimizes the quadratic model along the negative gradient direction inside the trust region; i.e., $s_k^{\mathsf{cp}} = -\alpha_k^{\mathsf{cp}} \, C'_E(x_k)^T C_E(x_k)$, where the step length is given by

$$\alpha_k^{\mathsf{cp}} = \begin{cases} \frac{\|C'_E(x_k)^T C_E(x_k)\|^2}{\|C'_E(x_k) C'_E(x_k)^T C_E(x_k)\|^2} & \text{if } \frac{\|C'_E(x_k)^T C_E(x_k)\|^3}{\|C'_E(x_k) C'_E(x_k)^T C_E(x_k)\|^2} \leq \Delta_k, \\ \frac{\Delta_k}{\|C'_E(x_k)^T C_E(x_k)\|} & \text{otherwise.} \end{cases} \tag{3.2}$$

The fraction of the Cauchy decrease condition means that the step s_k must predict via the quadratic model of the function $m_k(s)$ at least as much as a fraction of the decrease given by the Cauchy step s_k^{cp} on $m_k(s)$; that is, there exists a constant $\sigma > 0$ fixed across all iterations, such that

$$m_k(0) - m_k(s) \geq \sigma \, [m_k(0) - m_k(s_k^{\mathsf{cp}})]. \tag{3.3}$$

Later, we will find it useful to work with the following condition instead of (3.3). If the step satisfies a fraction of Cauchy decrease, i.e., inequality (3.3), then

$$m_k(0) - m_k(s_k) \geq \frac{\sigma}{2} \|C'_E(x_k)^T C_E(x_k)\| \min \left\{ \frac{\|C'_E(x_k)^T C_E(x_k)\|}{\|C'_E(x_k)^T C'_E(x_k)\|} \, , \; \Delta_k \right\}. \tag{3.4}$$

More details can be found in Carter [8], Moré [23], and Powell [29].

4. The continuity property. Throughout this paper, we will require the following continuity and boundedness assumptions about the problem being solved.

Assumption 1. C'_E and $C'_I \in Lip(\Omega)$, where $\Omega \in \Re^n$ is an open convex set.

Assumption 2. $C_E(x)$, $C_I(x)$, $C'_E(x)$, and $C'_I(x)$ are all bounded in norm for $x \in \Omega$.

Equivalent to these assumptions is the existence of constants $\gamma_E, \gamma_I \geq 0$, $\beta \geq \beta_i \geq 0$ for $i \in \{1, \ldots, m\}$, and $b \geq 0$, such that for all $x, y \in \Omega$,

$$\|C'_E(x) - C'_E(y)\| \leq \gamma_E \|x - y\|, \tag{4.1}$$

$$\|C'_I(x) - C'_I(y)\| \leq \gamma_I \|x - y\|, \tag{4.2}$$

$$\|C'(x)\| \leq \beta, \tag{4.3}$$

$$\|c'_i(x)\| \leq \beta_i, \quad i \in \{1, \ldots, m\}, \tag{4.4}$$

$$\|C_E(x)\| \leq b, \quad \text{and}$$

$$\|C_I(x)\| \leq b.$$

The following lemma establishes the Lipschitz continuity of $\nabla\Phi(x)$ under Assumptions 1 and 2.

LEMMA 4.1. *Let Assumptions* 1 *and* 2 *hold. Then, for every* x, $y \in \Omega$,

$$\|\nabla\Phi(x) - \nabla\Phi(y)\| \leq a_0 \|x - y\|, \tag{4.5}$$

where a_0 *is a positive constant.*

Proof. We have

$$\|\nabla\Phi(x) - \nabla\Phi(y)\| \leq \|\nabla\Phi_E(x) - \nabla\Phi_E(y)\| + \|\nabla\Phi_I(x) - \nabla\Phi_I(y)\| .$$

Now,

$$\begin{aligned}\|\nabla\Phi_E(x) - \nabla\Phi_E(y)\| &= \left\|C'_E(x)^T C_E(x) - C'_E(y)^T C_E(y)\right\| \\ &\leq \left\|\left(C'_E(x) - C'_E(y)\right)^T C_E(x)\right\| + \left\|C'_E(y)^T \left(C_E(x) - C_E(y)\right)\right\| \\ &\leq \gamma_E \|x - y\| \cdot \|C_E(x)\| + \|C'_E(y)\| \cdot \|C_E(y) - C_E(x)\| \\ &\leq \left(\gamma_E\, b + \beta_E \left(\sum_{i \in E} \beta_i\right)\right) \|x - y\|, \end{aligned} \tag{4.6}$$

where β_E bounds $\|C'_E(y)\|$, which establishes the Lipschitz continuity of $\nabla\Phi_E(x)$.

Also, we can bound $\|C'_I(y)\|$ by β_I. Hence, using an argument similar to what we used in (4.6), we have

$$\|\nabla\Phi_I(x) - \nabla\Phi_I(y)\| \leq b \cdot \gamma_I \cdot \|x - y\| + \beta_I \|W_I(y)C_I(y) - W_I(x)C_I(x)\| .$$

We will complete the proof by showing that $W_I(\cdot)C_I(\cdot)$ is Lipschitz continuous. Consider a fixed $i \in I$, and let $Z_i = \{z \in \Re^n : c_i(z) = 0\}$. If $c_i(x) \cdot c_i(y) < 0$, then we can choose $z_i \in Z_i \cap [x, y]$. Thus,

$$w_i(x)c_i(x) - w_i(y)c_i(y) = \begin{cases} c_i(x) - c_i(z_i) & \text{if} \quad c_i(y) < 0\,, \\ c_i(z_i) - c_i(y) & \text{if} \quad c_i(y) > 0\,, \end{cases}$$

and so by (4.4),

$$|w_i(x)c_i(x) - w_i(y)c_i(y)| \leq \beta_i \max\left\{\|x - z_i\|, \|z_i - y\|\right\} \leq \beta_i \|x - y\| .$$

If $c_i(x) \cdot c_i(y) \geq 0$, then for $c_i(y) \neq 0$,

$$w_i(x)c_i(x) - w_i(y)c_i(y) = \begin{cases} 0 & \text{if} \quad c_i(y) < 0\,, \\ c_i(x) - c_i(y) & \text{if} \quad c_i(y) > 0\,, \end{cases}$$

and so $|w_i(x)c_i(x) - w_i(y)c_i(y)| \leq \beta_i \|x-y\|$. If $c_i(y) = 0$, then similarly, $|w_i(x) c_i(x) - w_i(y)c_i(y)| \leq \beta_i \|x-y\|$. Putting the components together yields

$$\|W_I(y)C_I(y) - W_I(x)C_I(x)\| \leq \|W_I(y)C_I(y) - W_I(x)C_I(x)\|_1 \leq \left(\sum_{i \in I} \beta_i\right) \|x-y\|\,.$$

Finally, we obtain $\|\nabla\Phi_I(x) - \nabla\Phi_I(y)\| \leq \left(\gamma_I\, b + \beta_I \left(\sum_{i\in I} \beta_i\right)\right) \|x-y\|$. This completes the proof. □

5. Single-model algorithm.

We describe the single-model algorithm for (1.2) in four sections. In section 5.1, we discuss the model and the trust-region subproblem, and section 5.2 describes the method of solving this subproblem. Section 5.3 is devoted to presenting the trial-step acceptance mechanism and the trust-region updating strategy. Finally, in section 5.4 we summarize the single-model algorithm.

5.1. The trust-region subproblem.

The idea here is to use a standard trust-region algorithm for nonlinear least squares on

$$\min_{x \in \Re^n} \Phi(x)\,,$$

where $\Phi(x)$ is given by (2.4).

At the current iterate k, the set of binding or violated inequalities at x_k is identified, and the 0–1 diagonal matrix $W(x_k)$ defined by (2.1) is assembled. Next, we build a quadratic model

$$q(x_k + s) = \frac{1}{2}\|W(x_k)(C'(x_k)s + C(x_k))\|^2 \tag{5.1}$$

of Φ around the current iterate x_k, as in the Gauss–Newton approach, where

$$\Phi(x_k + s) = \frac{1}{2}C(x_k+s)^T W(x_k+s)C(x_k+s).$$

Thus, the quadratic model contains information about only those inequalities which are violated or active at x_k.

At each iteration, a trial step s_k is computed as an approximate solution to the trust-region subproblem

$$\begin{aligned} &\text{minimize} \;\; q(x_k+s) \equiv \frac{1}{2}\|W(x_k)\left(C'(x_k)s + C(x_k)\right)\|^2 \\ &\text{subject to} \; \|\; s \;\| \; \leq \Delta_k \end{aligned} \tag{5.2}$$

where $\Delta_k > 0$ is the current trust-region radius.

5.2. Solution of the single-model subproblem.

We want to compute an approximate solution to the trust-region subproblem (5.2). The most obvious strategy would be to use an algorithm such as the one used in the MINPACK routine LMDER [24], which is based on the Levenberg–Marquardt approach [22]. However, this routine will not solve underdetermined systems; i.e., it requires that the dimension of C be at least as large as the dimension of x. In the context of finding a solution to a set of equalities and inequalities, we have not assumed any relationship between n and m. Another alternative to consider is the completely general Moré–Sorensen routine GQTPAR [1]. However, this routine is unsuitable for $n < m$ because of its strategy for the hard case. (See Moré and Sorensen [25] for the definition of the hard case.) This routine always steps to the boundary of the trust region even when a zero-residual step is safely inside. We emphasize that this is not a criticism of GQTPAR; it is a statement of our special needs. Hence, we will use a dogleg algorithm to solve the subproblem (5.2).

The dogleg algorithm approximates the solution curve to problem (5.2) by a piecewise linear function connecting the Cauchy point to the "Newton" point. The Cauchy step is defined to be $s_k^{\mathsf{cp}} = -\alpha_k^{\mathsf{cp}}\, C'(x_k)^T W_k C(x_k)$, where

$$\alpha_k^{\mathsf{cp}} = \begin{cases} \frac{\|C'(x_k)^T W(x_k) C(x_k)\|^2}{\|W(x_k) C'(x_k) C'(x_k)^T W(x_k) C(x_k)\|^2} & \text{if } \frac{\|C'(x_k)^T W(x_k) C(x_k)\|^3}{\|W(x_k) C'(x_k) C'(x_k)^T W(x_k) C(x_k)\|^2} \le \Delta_k, \\ \frac{\Delta_k}{\|C'(x_k)^T W(x_k) C(x_k)\|} & \text{otherwise.} \end{cases}$$

If s_k^{cp} lies inside the trust region, then we compute the step that will play the role of the Newton step, and it is the minimum norm solution to

$$\text{(5.3)} \qquad \text{minimize } \frac{1}{2}\|W(x_k)\,(C'(x_k)s + C(x_k))\|^2.$$

We will refer to this step as s_k^{lf}. To compute this step, one can use the routine GELSX from LAPACK [1], which can handle both over- and underdetermined systems.

When n is large, iterative methods might have to be used to obtain the minimum norm solution of problem (5.3). A truncation procedure might also be needed. For example, a Steihaug–Toint-type algorithm can be used [34], [37].

The algorithm described in Golub and von Matt [17] is also of interest, especially for the large-scale case. This algorithm is applicable to the over- and underdetermined cases and can be applied directly to solve the trust-region subproblem (5.2).

Using the minimum norm solution ensures that if the computed step is outside the trust region, then there are no other solutions to (5.3) that are inside the trust region. If s^{lf} is inside the trust region, then we take it as the solution to the subproblem. Otherwise, we compute the dogleg step between the Cauchy point and s^{lf} with length Δ_k, and take it as the trial step.

ALGORITHM 5.1. Computing a Trial Step.

```
Compute the Cauchy step, $s^{\mathsf{cp}} = -\alpha_k^{\mathsf{cp}} (C_k')^T W_k C_k$.
If ( $\|s^{\mathsf{cp}}\| = \Delta_k$ ), then set $s_k = s^{\mathsf{cp}}$
Else, if ( $\nabla q_k(s^{\mathsf{cp}}) = 0$ ), then set $s_k = s^{\mathsf{cp}}$.
    Else, compute $s^{\mathsf{lf}}$, the minimum norm solution to
            minimize $\frac{1}{2}\ \|W_k C_k + W_k (C_k')^T s\|^2$
      If ( $\|s^{\mathsf{lf}}\| \le \Delta_k$ ), then set $s_k = s^{\mathsf{lf}}$.
      Else, dogleg between $s^{\mathsf{cp}}$ and $s^{\mathsf{lf}}$.
```

Since our theory is based on the fraction of the Cauchy decrease condition (see section 3), any method that computes a trial step that gives at least a fraction of the Cauchy decrease can be used. Therefore, in the case when n is large, a generalized dogleg algorithm introduced by Steihaug [34] and Toint [37] can be used to compute the trial step s_k. This algorithm is based on the linear conjugate gradient method and is known to be suitable for large problems for which effective preconditioners are known.

5.3. Accepting the step and updating the trust-region. Once we have computed a trial step s_k, we decide if the step is acceptable by comparing the amount of reduction s_k predicts in the model (5.1) to the amount of reduction we actually obtain in $\Phi(x)$. The actual reduction in $\Phi(x)$ is given by

$$\begin{aligned} Ared_k &= \Phi(x_k) - \Phi(x_k + s_k) \\ &= \frac{1}{2}\|W_k\, C_k\|^2 - \frac{1}{2}\|W(x_k + s_k)\, C(x_k + s_k)\|^2, \end{aligned}$$

and the predicted reduction in the model is given by

$$\begin{aligned} Pred_k &= q(x_k) - q(x_k + s_k) \\ &= \frac{1}{2}\|W_k\, C_k\|^2 - \frac{1}{2}\|W_k\, ({C_k}' s_k + C_k)\|^2. \end{aligned}$$

The trust-region algorithm should produce steps that decrease Φ and make progress toward the feasible region. To guarantee this, the actual reduction in Φ has to be greater than some fraction of the predicted reduction in the model for the step to be deemed acceptable. The computation of the trial step (specifically, the solution of the trust-region subproblem, discussed in section 5.2) ensures that the step predicts at least a fraction of the Cauchy decrease in the model, and hence, $Pred_k > 0$.

The step is accepted if $\eta_1 \le \frac{Ared_k}{Pred_k}$, where $\eta_1 \in (0, 1)$ is a small fixed constant, say 10^{-4}. If the step is judged acceptable, then we proceed to the next iteration. Otherwise, the trial step is rejected, the trust-region radius is decreased by setting $\Delta_k = \alpha_1 \|s_k\|$ for some $\alpha_1 \in (0, 1)$, and another trial step is computed from x_k in the smaller trust region.

If the step is accepted, then the trust-region radius is updated by comparing the value of $Ared_k$ with $Pred_k$. Namely, if $\eta_2 \le \frac{Ared_k}{Pred_k} < \eta_3$, where $\eta_2 \in (\eta_1, 1)$, then the radius of the trust region is kept the same. If the agreement between the actual reduction and the predicted reduction is poor ($\frac{Ared_k}{Pred_k} < \eta_2$, where η_2 is less than or equal to 0.1), then we allow possible reduction in the radius of the trust region. We set $\Delta_{k+1} = \min(\Delta_k, \alpha_2 \|s_k\|)$ for some $\alpha_2 \ge 1$. If, on the other hand, the agreement between the actual reduction and the predicted reduction is fair, $\eta_3 \le \frac{Ared_k}{Pred_k} < \eta_4$, possibly increase the trust region. Set $\Delta_{k+1} = \max(\Delta_k, \alpha_2 \|s_k\|)$. Typical values for η_3 and η_4 are 0.25 and 0.75. If the agreement between the actual reduction and the predicted reduction is good, $\eta_4 \le \frac{Ared_k}{Pred_k}$, then increase the trust region radius. Set $\Delta_{k+1} = \max(\alpha_2 \Delta_k, \alpha_3 \|s_k\|)$, where $\alpha_3 \ge \alpha_2$.

Further details concerning the basic trust-region framework can be found, for instance, in Moré [23] or Dennis and Schnabel [14].

5.4. Summary of the single-model algorithm. Putting the pieces together, we can now outline the single-model algorithm for finding a local minimizer of $\Phi(x)$.

ALGORITHM 5.2. The Single-Model Algorithm.

Initialization: *Given x_0 and $\Delta_0 > 0$, compute $C(x_0)$, $W(x_0)$, and $C'(x_0)$.*

At every iteration, do

Step 1. *Check for convergence.*

Step 2. *Compute a trial step s_k by approximately solving*

$$\begin{aligned} &\text{minimize} \quad q(x_k + s) = \frac{1}{2}\|W_k\, (C_k' s + C_k)\,\|^2 \\ &\text{subject to } \|s\| \ \le \Delta_k. \end{aligned}$$

Step 3. *Compute* $C(x_k+s_k)$ *and* $W(x_k+s_k)$.

Step 4. *Decide if the step is acceptable based on the ratio of* $Ared_k$ *to* $Pred_k$ *(as discussed in section* 5.3*) and update* Δ_k *(by the mechanism given in section* 5.3*).*

Step 5. *If the step* s_k *is acceptable, set* $x_{k+1} = x_k + s_k$*, compute* $C'(x_{k+1})$*, and go to Step* 1.
Else set $k := k+1$ *and go to Step* 2.

6. Multimodel algorithm. We describe our multimodel algorithm in four sections. In sections 6.1–6.3, we describe our way of computing the trial step. Section 6.4 is devoted to presenting the trial-step acceptance mechanism and how to update the trust-region radius.

6.1. The trust-region subproblem. At each iteration k, the set of binding or violated inequalities at x_k is identified and the 0–1 diagonal matrix $W(x_k)$, defined by (2.1), is assembled.

Next, a generalized Cauchy point is computed to be the point $x_k + s_k^{\mathsf{gcp}}$ where s_k^{gcp} solves the one-dimensional piecewise-quadratic convex minimization problem

$$\begin{aligned} &\underset{s\in\Re^n}{\text{minimize}} \ \frac{1}{2}\|V_k(s)(\ C'_k s + C_k\ )\|^2 \\ &\text{subject to } \|s\| \le \Delta_k, \quad s = -\alpha {C'_k}^T W_k C_k, \end{aligned}$$

where $V_k(s)$ is another 0–1 diagonal indicator matrix, whose diagonal elements v_i are

$$v_i(s) = \begin{cases} 1, & i \in E, \\ 1, & i \in I,\ w_i(x_k) = 1, \text{ and } c_i(x_k) + c'_i(x_k)\,s \ge 0, \\ 0, & \text{otherwise.} \end{cases} \tag{6.1}$$

The algorithm then computes a trial step by solving the following standard trust-region subproblem:

$$\begin{aligned} &\underset{s\in\Re^n}{\text{minimize}} \ \frac{1}{2}\|V_k(s_k^{\mathsf{gcp}})(\ C'_k s + C_k\ )\|^2 \\ &\text{subject to } \|s_k\| \le \Delta_k. \end{aligned}$$

It will be useful later if we point out here that

$$\Psi_k(s) = \frac{1}{2}\|V_k(s)(\ C'_k s + C_k\ )\|^2 \tag{6.2}$$

is a local form of Φ for the linearization of C about x_k. It is easy to see that $\Psi_k(0) = \Phi(x_k)$, $\nabla_s \Psi_k(s) = {C'_k}^T V_k(s)(\ C'_k s + C_k\ )$, and $\nabla_s \Psi_k(0) = {C'_k}^T W_k C_k$.

The mappings W_k and V_k are very reminiscent of the structure functionals used by Osborne and Womersley. Interested readers are referred to [26] and [27].

6.2. Computing the generalized Cauchy point. In this section, we present the algorithm that we use to compute the generalized Cauchy point and thus to determine which inequalities will be included in the trial step calculation. At the kth iteration, given $C(x_k)$, we form the *nonlinear* indicator matrix $W(x_k)$, given by (2.1).

The generalized Cauchy point s^{gcp} solves the trust-region subproblem

$$\begin{array}{lcl} & \underset{\alpha \in \Re}{\text{minimize}} & q_k(s) \equiv \frac{1}{2}\| V_k(s)\ (C_k + C_k'\, s)\ \|^2 \\ (6.3) & & \\ & \text{subject to} & s = -\alpha (C_k')^T W_k\, C_k, \\ & & \|s\| \le \Delta_k, \end{array}$$

where $V_k(s)$ is a diagonal 0–1 indicator matrix that designates which of the *linearized* inequalities that are binding or violated at $s = 0$ are still violated or binding for a particular step s. As with $W(x)$, the diagonal elements corresponding to equalities are always one, and $V_k(s)$ is given by (6.1)

In problem (6.3), the step is restricted to the negative gradient direction of the model at x_k. Normalizing this direction makes the trust-region constraint invariant under the resulting change of variables, and so we will refer to

$$d_k^{\mathsf{cp}} = \frac{-(C_k')^T W_k\, C_k}{\|(C_k')^T W_k\, C_k\|}$$

as the Cauchy direction.

It is easy to see that $V_k(s)$ evaluated at $s = 0$ is just W_k. Furthermore, the only components of $V_k(s)$ that actually depend on s are those v_i corresponding to inequalities that are binding or violated at x_k. Consequently, if all of the inequalities are strictly satisfied at x_k (or if there are no inequalities), then $V(s) \equiv W_k$ for all s. In this case, $V_k(s^{\mathsf{gcp}}) = W_k$, and problem (6.3) reduces to the standard Cauchy point computation with $s_k^{\mathsf{gcp}} = \alpha_k\, d_k^{\mathsf{cp}}$, where

$$(6.4) \qquad \alpha_k = \min\left\{ \frac{\|(C_k')^T W_k\, C_k\|}{\|W_k\, C_k'\, d_k^{\mathsf{cp}}\,\|^2}\ ,\ \Delta_k \right\}.$$

For the remainder of this section, we will assume that there is at least one violated or binding inequality at x_k. Thus, we have a one-dimensional trust-region subproblem of the form

$$\begin{array}{lcl} & \underset{\alpha \in \Re}{\text{minimize}} & q_k(\alpha) = \frac{1}{2}\| V_k(\alpha)\ (C_k + \alpha\, C_k'\, d_k^{\mathsf{cp}})\ \|^2 \\ (6.5) & & \\ & \text{subject to} & |\alpha| \le \Delta_k. \end{array}$$

Note that we obtain V_k as a function of α alone through the substitution $s = \alpha\, d_k^{\mathsf{cp}}$. The objective function of problem (6.5) is a one-dimensional piecewise quadratic, and it is convex and continuously differentiable.

It is worth noting that the number of ones in $V_k(\alpha)$ decreases as α increases and that for $\alpha_1 \le \alpha_2$, the following inequality holds for all α:

$$\| V_k(\alpha_1)\ (C_k + \alpha\, C_k'\, d_k^{\mathsf{cp}})\ \| \ge \| V_k(\alpha_2)\ (C_k + \alpha\, C_k'\, d_k^{\mathsf{cp}})\ \|.$$

We will use a “piecewise” form of Newton’s method to solve problem (6.5). Starting from $\alpha_0 = 0$, we fix the indicator matrix at $V_k(\alpha_j)$ and form the jth quadratic model

$$(6.6) \qquad q_k^j(\alpha) = \frac{1}{2}\| V_k(\alpha_j)\ (C_k + \alpha\, C_k'\, d_k^{\mathsf{cp}})\ \|^2.$$

The jth quadratic model $q_k^j(\alpha)$ is equal to the composite quadratic model $q_k(\alpha)$ given in (6.6) at the point α_j. Then, we minimize this model subject to the trust-region constraint to obtain the next iterate α_{j+1}. The Newton step for (6.6) is

$$\alpha_{j+1} = \frac{-C_k^T \; V_k(\alpha_j) \; C_k' \; d_k^{\mathsf{cp}}}{\|V_k(\alpha_j) \; C_k' \; d_k^{\mathsf{cp}}\|^2}. \tag{6.7}$$

Since the quadratic model is one-dimensional and convex, either the Newton step is the solution to the trust-region subproblem or the trust-region is binding, in which case $\alpha_{j+1} = \Delta_k$.

Given the new iterate, α_{j+1}, we must determine if it is the solution to (6.5). First, we evaluate $V_k(\alpha_{j+1})$ to determine if the set of linearly violated or binding inequalities has changed. The following lemma gives sufficient conditions for α_{j+1} to solve (6.5).

LEMMA 6.1. *Let α_{j+1} minimize the jth quadratic model $q_k^j(\alpha)$ inside the trust region. Then, α_{j+1} solves the trust-region subproblem* (6.5) *if one of the following conditions holds:*

i. $V_k(\alpha_{j+1}) = V_k(\alpha_j)$.

ii. *The trust-region is binding at α_{j+1}.*

iii. *The gradient of the new quadratic model,*

$$\nabla q_k^{j+1}(\alpha_{j+1}) = (C_k + \alpha_{j+1} C_k' d_k^{\mathsf{cp}})^T \, V_k(\alpha_{j+1}) \, C_k' \, d_k^{\mathsf{cp}} \,, \tag{6.8}$$

is equal to zero.

Furthermore, the algorithm must terminate with α_{j+1} that satisfies either i, ii, *or* iii *in a finite number of iterations.*

Proof. i. Since $V_k(\alpha_{j+1}) = V_k(\alpha_j)$, we have

$$\begin{aligned} q_k^j(\alpha_{j+1}) &= \frac{1}{2} \| \, V_k(\alpha_j) \; (C_k + \alpha_{j+1} \, C_k' \, d_k^{\mathsf{cp}}) \; \|^2 \\ &= \frac{1}{2} \| \, V_k(\alpha_{j+1}) \; (C_k + \alpha_{j+1} \, C_k' \, d_k^{\mathsf{cp}}) \; \|^2 = q_k^{j+1}(\alpha_{j+1}) \, . \end{aligned}$$

Since $q_k^{j+1}(\alpha_{j+1})$ is the composite quadratic q_k at α_{j+1}, α_{j+1} solves (6.5).

ii. Now assume that $V_k(\alpha_{j+1}) \neq V_k(\alpha_j)$. Then from the definition of V_k the only possibility is that at least one of the linear inequalities that was violated or binding at α_j is strictly satisfied at α_{j+1}. Without loss of generality, we will assume that there is only one such inequality with index l such that $V_{lk}(\alpha_j) = 1$ and $V_{lk}(\alpha_{j+1}) = 0$.

We need to show that if $\alpha_{j+1} = \Delta_k$ minimizes $q_k^j(\alpha)$ subject to the trust-region constraint, then it also minimizes $q_k^{j+1}(\alpha)$ in the trust region. Thus, α_{j+1} will be a solution to (6.5) because $q_k^{j+1}(\alpha_{j+1})$ is the composite quadratic model at α_{j+1}.

The only difference between $q_k^j(\alpha_{j+1})$ and $q_k^{j+1}(\alpha_{j+1})$ is in the lth term, with $V_{lk}(\alpha_j) = 1$ and $V_{lk}(\alpha_{j+1}) = 0$. So,

$$\nabla q_k^j(\alpha_{j+1}) = \nabla q_k^{j+1}(\alpha_{j+1}) + (c_{lk} + \alpha_{j+1} \, c_{lk}' \, d_k^{\mathsf{cp}}) \; c_{lk}' \, d_k^{\mathsf{cp}} \, . \tag{6.9}$$

Since $V_{lk}(\alpha_j) = 1$, the definition of V_k indicates that $W_l(x_k) = 1$, which implies that $c_{lk} \geq 0$. Also, $V_{lk}(\alpha_{j+1}) = 0$ yields $c_{lk} + \alpha_{j+1} \, c_{lk}' \, d_k^{\mathsf{cp}} < 0$.

We can conclude that $\alpha_{j+1} \in [0, \Delta_k]$ from the fact that $\nabla_\alpha q_k(0) \leq 0$ because either $d_k^{\mathsf{cp}} = 0$ and $\alpha = 0$ solves (6.5) or αd_k^{cp} is a descent direction for q_k at $\alpha = 0$, i.e.,

$$\nabla_\alpha q_k(0) = C_k^T W_k C_k' d_k^{\mathsf{cp}} = -\|(C_k')^T W_k C_k\| \leq 0.$$

Thus, $\alpha_{j+1} \geq 0$, and so $c'_{lk}\, d_k^{\mathsf{cp}} \leq 0$. From this we obtain

$$(c_{lk} + \alpha_{j+1}\, c'_{lk}\, d_k^{\mathsf{cp}})\; c'_{lk}\, d_k^{\mathsf{cp}} \geq 0. \tag{6.10}$$

In other words, one previously violated linear inequality became strictly satisfied or binding on the interval $[\alpha_j, \alpha_{j+1}]$.

For α_{j+1} to minimize $q_k^j(\alpha)$ with the trust-region constraint binding, we know that

$$\nabla_\alpha q_k^j(\alpha_{j+1}) \leq 0.$$

Combining this with (6.9) and (6.10), we can conclude that $\nabla q_k^{j+1}(\alpha_{j+1}) \leq 0$, and since the trust region is binding, α_{j+1} solves (6.5).

iii. If $\nabla q_k^{j+1}(\alpha_{j+1}) = 0$, then clearly α_{j+1} solves (6.5).

Finally, the algorithm must terminate in at most $\|W_I(x_k)\|_F^2 + 1$ iterations. At every iteration, at least one linear inequality, with $w_i(x_k) = 1$, must become strictly satisfied, or i indicates that the algorithm will terminate at that iteration. We started with $\|W_I(x_k)\|_F^2$ violated or binding linear inequalities, so by iteration $\|W_I(x_k)\|_F^2$, either we have found the solution or all of the inequalities with $w_{ik} = 1$ now have $v_{ik}(\alpha) = 0$. Then, only the equalities, $i \in E$, have $v_{ik} = 1$, and one more iteration could be required to solve (6.5). □

Now after Lemma 6.1 has established constructive stopping criteria, we can state our algorithm for computing the generalized Cauchy point.

ALGORITHM 6.2. Generalized Cauchy Point Algorithm.

Initialization.

Given x_k, C_k, W_k, C'_k, and $\Delta_k > 0$.

Set $j = 0$, $\alpha_0 = 0$, and $V_{0k} = W_k$.

Compute s^{gcp} and $V_k(s^{\mathsf{gcp}})$ as follows:

Step 1. *If all inequalities are strictly satisfied, i.e., $\sum_{i\in I} w_i(x_k) = 0$, then $V_k(s^{\mathsf{gcp}}) \equiv W_k$ and $s^{\mathsf{gcp}} = s_k^{\mathsf{cp}} = \alpha_k d_k^{\mathsf{cp}}$, where α_k is given by* (6.4). **Return**.

Step 2. *Compute the normalized negative gradient direction:*

$$d_k^{\mathsf{cp}} = \frac{-(C'_k)^T\, W_k\, C_k}{\|(C'_k)^T\, W_k\, C_k\|}.$$

Step 3. *Solve*

$$\begin{aligned} &\textit{minimize } q_k^j(\alpha) = \frac{1}{2}\|V_{jk}\,(\alpha\, C'_k\, d_k^{\mathsf{cp}} + C_k)\,\|^2 \\ &\textit{subject to } |\alpha| \leq \Delta_k \end{aligned}$$

for the new iterate α_{j+1}.

- *Compute the Newton step*

$$\alpha_{j+1} = \frac{-C_k^T\; V_{jk}\; C'_k\; d_k^{\mathsf{cp}}}{(d_k^{\mathsf{cp}})^T\; C'_k\; V_{jk}\; C'_k\; d_k^{\mathsf{cp}}}.$$

- *Determine if the trust region is binding; if $(\alpha_{j+1} > \Delta_k)$, then $\alpha_{j+1} = \Delta_k$.*

Step 4. *Evaluate $V_k(\alpha_{j+1})$ as in* (6.1).

```
Step 5. Check for convergence.
    • If ($V_k(\alpha_{j+1}) = V_k(\alpha_j)$) or ($\alpha_{j+1} = \Delta_k$), then $\alpha_{j+1}$ solves (6.5).
      Else
      — Compute the gradient of the new quadratic model,
            $\nabla q_k^{j+1}(\alpha_{j+1}) = (C_k + \alpha_{j+1} C'_k d_k^{\mathsf{cp}})^T \, V_k(\alpha_{j+1}) \, C'_k \, d_k^{\mathsf{cp}}$ .
      — If ($\nabla q_k^{j+1}(\alpha_{j+1}) = 0$), then $\alpha_{j+1}$ solves (6.5).
      Else go to Step 3.
Step 6. If ($V_k(s^{\mathsf{gcp}}) \neq W_k$), then compute $s^{\mathsf{gcp}} = -\alpha_k \frac{(C'_k)^T V_k(s^{\mathsf{gcp}}) C_k}{\|(C'_k)^T V_k(s^{\mathsf{gcp}}) C_k\|}$.
```

6.3. Solution of the trust-region subproblem. We now consider an algorithm that approximates the solution to the following trust-region subproblem:

$$\begin{aligned} &\underset{s \in \Re^n}{\text{minimize}} \ \frac{1}{2} \| V_k(s_k^{\mathsf{gcp}})(\ C'_k s + C_k \ ) \|^2 \\ &\text{subject to } \|s\| \leq \Delta_k. \end{aligned}$$

Algorithm 6.3. Multimodel Dogleg Step.

```
Compute the generalized Cauchy step $s^{\mathsf{gcp}}$ and $V(s^{\mathsf{gcp}})$;
If ( $\|s^{\mathsf{gcp}}\| = \Delta_k$ ), then $s_k = s^{\mathsf{gcp}}$.            (* trust region was binding *)
    Else if ( $\nabla q_k(s^{\mathsf{gcp}}) = 0$ ), then $s = s^{\mathsf{gcp}}$.
    Else
      * Compute $s^{\mathsf{lf}}$, the minimum norm solution to
            minimize $\frac{1}{2}\|V(s^{\mathsf{gcp}})(C + C's)\|^2$
      * If ( $\|s^{\mathsf{lf}}\| \leq \Delta_k$ ), then $s_k = s^{\mathsf{lf}}$
      * Else dogleg between $s^{\mathsf{gcp}}$ and $s^{\mathsf{lf}}$
```

6.4. Accepting the steps. Let s_k be a trial step computed by the algorithm. We test whether the point $x_{k+1} = x_k + s_k$ is making progress toward the feasible region. We define the actual reduction in moving from x_k to x_{k+1} to be

$$\begin{aligned} Ared_k &= \Phi_k - \Phi_{k+1}, \\ &= \frac{1}{2} \left[\ \|W_k C_k\|^2 - \|W_{k+1} C_{k+1}\|^2 \ \right]. \end{aligned}$$

The predicted reduction will be

$$Pred_k = \frac{1}{2} \left[\ \|W_k C_k\|^2 - \|V_k(s_k^{\mathsf{gcp}})(\ C_k + C'_k s_k \ )\|^2 \ \right].$$

From the way of computing the trial step, the predicted reduction is defined to produce a fraction of the Cauchy decrease in Φ at x_k, which means that $Pred_k > 0$. Hence, the step is accepted if $\frac{Ared_k}{Pred_k} \geq \eta_1$ where $\eta_1 \in (0, 1)$.

Our rule for accepting the step and updating the trust-region radius for this algorithm is the same as in section 5.3.

7. Convergence results for the single-model algorithm.

In this section, we will use the convergence theory for trust-region methods provided in Moré [23] to show that the Levenberg–Marquardt approach to the solution of (1.1) is globally convergent to a first-order stationary point under reasonable assumptions on $C(x)$.

We make the following assumption on the sequence of iterates $\{x_k\}$ generated by the single-model algorithm.

Assumption 3. For all k, x_k and $x_k + s_k \in \Omega$.

In order to apply Theorem 4.14 from Moré [23], we need to establish that $\Phi(x)$ is bounded below, $\nabla\Phi(x)$ is uniformly continuous, and the model Hessian is uniformly bounded. From (2.2) and (2.3), it is obvious that $\Phi(x)$ is bounded below by 0. We obtain the following convergence result.

LEMMA 7.1. *Let Assumptions* 1, 2, *and* 3 *hold. Then,*

$$\lim_{k\to\infty} \| \nabla\Phi(x_k) \| = 0.$$

Proof. A straightforward calculation from (5.1) yields

$$\nabla q(x_k) = C'(x_k)W(x_k)C(x_k) = \nabla\Phi(x_k)\,.$$

Lemma 4.1 shows that $\nabla\Phi$ is Lipschitz continuous, and thus, uniformly continuous.

Boundedness of the Hessian of the quadratic model can be established in the Frobenius norm in the following manner:

$$\begin{aligned} \nabla^2 q(x_k) &= C^{'}(x_k)^T W(x_k) C^{'}(x_k), \\ \| \nabla^2 q(x_k) \|_F &\le \| {C_E^{'}}^T C_E^{'} \|_F + \| {C_I^{'}}^T (x_k) W_I(x_k) C_I^{'}(x_k) \|_F \\ &\le \| C_E^{'} \|_F^2 + \| C_I^{'} \|_F^2 \\ &\le \beta_E^2 + \beta_I^2\,. \end{aligned}$$

The lemma then follows from Theorem 4.14 in Moré [23]. □

8. Convergence results for the multimodel algorithm. In this section we start by proving some intermediate lemmas needed for global convergence. Then we prove our main global convergence results.

We add to our list of assumptions the following assumption on the sequence of iterates $\{x_k\}$ generated by the multimodel algorithm.

Assumption 3′. For all k, $x_k + s_k^{\mathsf{cp}}$, $x_k + s_k^{\mathsf{gcp}}$, and $x_k + s_k \in \Omega$.

We start with the following lemma, which is needed in the proof of Lemma 8.2.

LEMMA 8.1. *Let Assumptions* 1, 2, *and* 3′ *hold. Suppose that at any given iteration k there exists an $i \in I$ such that $|c_{ik}| > 0$. If Δ_k satisfies*

$$\Delta_k \le \frac{1}{2\beta} \min_{c_{ik}\neq 0} |c_{ik}|, \tag{8.1}$$

where β is as in (4.3)*, then*

$$W_{k+1} = W_k - B_k \tag{8.2}$$

and

$$V_k(s_k^{\mathsf{gcp}}) = W_k - \bar{B}_k, \tag{8.3}$$

where B_k is a 0–1 *diagonal matrix whose diagonal elements are*

$$b_i = \begin{cases} 1 & \textit{if} \quad i \in I,\ c_{ik} = 0,\ \textit{and} \quad c_{ik+1} < 0, \\ 0 & \textit{otherwise,} \end{cases}$$

and $\bar{B}_k$ *is a* 0–1 *diagonal matrix whose diagonal elements are*

$$\bar{b}_i = \begin{cases} 1 & \textit{if} \quad i \in I,\ c_{ik} = 0,\ \textit{and} \ \ c'_{ik} s_k^{\mathsf{gcp}} < 0, \\ 0 & \textit{otherwise.} \end{cases}$$

Proof. Let i index a constraint such that $c_{ik} \neq 0$. We will show that, if (8.1) holds, c_{ik+1} has the same sign as c_{ik}.

From the hypothesis, $c_{ik+1} = c_{ik} + c'_i(x_k + \xi s_k)s_k$, where $\xi \in (0, 1)$. Therefore, if $c_{ik} > 0$, then $c_{ik+1} \geq c_{ik} - \|c'_i(x_k + \xi s_k)\| \|s_k\|$, and because $\|s_k\| \leq \Delta_k$ satisfies (8.1), we have $c_{ik+1} > \frac{c_{ik}}{2} > 0$. On the other hand, if $c_{ik} < 0$, then $c_{ik+1} \leq c_{ik} + \|c'_i(x_k + \xi s_k)\| \|s_k\|$, and again, because $\|s_k\|$ satisfies (8.1), we have $c_{ik+1} < \frac{c_{ik}}{2} < 0$. Therefore, for all the above cases, if $w_{ik} = 1$, then $w_{ik+1} = 1$ and hence $b_{ik} = 0$.

Now consider the case where $c_{ik} = 0$. In this case $w_{ik} = 1$. Therefore, if $c_{ik+1} \geq 0$, then $w_{ik+1} = 1$, and hence $b_{ik} = 0$. Thus, the only case where $b_{ik} = 1$ is when $c_{ik} = 0$ and $c_{ik+1} < 0$. Hence, (8.2) easily follows. Similarly, we can show that (8.3) holds. This completes the proof. □

The following lemma shows, for a fixed iterate k, how accurate our definition of predicted reduction is as an approximation to the actual reduction. This lemma is used in the proof of Theorem 8.5.

LEMMA 8.2. *Let Assumptions* 1, 2, *and* 3′ *hold. For a fixed* k *and varying* Δ_k, *there exists a positive constant* a_1, *which depends on* k, *such that, if* Δ_k *satisfies* (8.1), *then*

$$|Ared_k - Pred_k| \leq a_1 \|s_k\|^2. \tag{8.4}$$

Proof. If there is no index i such that $i \in I$ and $c_{ik} > 0$ or $i \in E$ and $|c_{ik}| > 0$, then the point x_k is a solution and the algorithm is terminated and there is nothing to prove.

Consider the case when there is at least one index $i \in I$ with $c_{ik} > 0$ or $i \in E$ with $|c_{ik}| > 0$. Because of the assumption that Δ_k satisfies (8.1) and using the above lemma, we have

$$\begin{aligned} Ared_k &= \frac{1}{2}\left[C_k^T W_k C_k - C_{k+1}^T W_{k+1} C_{k+1}\right] \\ &= \frac{1}{2}\left[C_k^T W_k C_k - C_{k+1}^T (W_k - B_k) C_{k+1}\right] \end{aligned}$$

and

$$\begin{aligned} Pred_k &= \frac{1}{2}\left[C_k^T W_k C_k - (C_k + C'_k s_k)^T V_k(s_k^{\mathsf{gcp}})(C_k + C'_k s_k)\right] \\ &= \frac{1}{2}\left[C_k^T W_k C_k - (C_k + C'_k s_k)^T (W_k - \bar{B}_k)(C_k + C'_k s_k)\right]. \end{aligned}$$

Thus,

$$\mid Ared_k - Pred_k \mid = \frac{1}{2} \left| \ C_{k+1}^T (W_k - B_k) C_{k+1} - (C_k + C'_k s_k)^T (W_k - \bar{B}_k)(C_k + C'_k s_k) \ \right|.$$

But, because $B_k C_k = 0$ and $\bar{B}_k C_k = 0$, we have

$$\begin{aligned} \mid Ared_k - Pred_k \mid = \frac{1}{2} \Big| \ & (C_k + C'(x_k + \xi s_k)s_k)^T W_k (C_k + C'(x_k + \xi s_k)s_k \\ & - s_k^T C'(x_k + \xi s_k)^T B_k C'(x_k + \xi s_k) s_k \\ & - (C_k + C'_k s_k)^T W_k (C_k + C'_k s_k) + s_k^T C'^T_k \bar{B}_k C'_k s_k \ \Big|. \end{aligned}$$

Using Assumptions 1, 2, and $3'$ we can easily obtain

$$\mid Ared_k - Pred_k \mid \leq a_1 \|s_k\|^2.$$

This completes the proof. □

In the above lemma we bound the difference between $Ared_k$ and $Pred_k$ by a constant a_1 times $\|s_k\|^2$. This constant depends on k. This lemma can be used for a fixed iterate k, but it cannot be used, with the same constant a_1, across all iterates.

The following lemma gives a bound to $|Ared_k - Pred_k|$ that can be used across all iterates. It also shows how accurate our definition of predicted reduction is as an approximation to the actual reduction. It says that $\mid Ared_k - Pred_k \mid = o\,(\Delta_k)$. This lemma is used in the proof of Theorem 8.6.

Lemma 8.3. *Let Assumptions* 1, 2, *and* $3'$ *hold. Then, as* $\Delta_k \rightarrow 0$,

$$|Ared_k - Pred_k| \leq o(\Delta_k). \tag{8.5}$$

Proof. From the definition of $Ared_k$ and $Pred_k$, we have

$$\begin{aligned}
\mid Ared_k - Pred_k \mid &= \frac{1}{2} \Big| \; \|W_k C_k\|^2 - \|W_{k+1} C_{k+1}\|^2 - \|W_k C_k\|^2 \\
&\qquad\qquad + \|V_k(s_k^{\mathsf{gcp}})(C_k + C_k' s_k)\|^2 \; \Big| \\
&\leq \Big| \; \|W_k C_k\|^2 - \|W_{k+1} C_{k+1}\|^2 - \|W_k C_k\|^2 \\
&\qquad\qquad + \|W_k(C_k + C_k' s_k)\|^2 \; \Big| \\
&\quad + \Big| \; \|W_k(C_k + C_k' s_k)\|^2 - \|V_k(s_k^{\mathsf{gcp}})(C_k + C_k' s_k^{\mathsf{gcp}})\|^2 \; \Big| \\
&\quad + \Big| \; \|V_k(s_k^{\mathsf{gcp}})(C_k + C_k' s_k^{\mathsf{gcp}})\|^2 - \|V_k(s_k^{\mathsf{gcp}})(C_k + C_k' s_k)\|^2 \; \Big| \\
&= \mathrm{I} + \mathrm{II} + \mathrm{III},
\end{aligned}$$

where

$$\begin{aligned}
\mathrm{I} &= \Big| \; \|W_{k+1} C_{k+1}\|^2 - \|W_k C_k\|^2 + \|W_k C_k\|^2 - \|W_k(C_k + C_k' s_k)\|^2 \; \Big|, \\
\mathrm{II} &= \Big| \; \|W_k(C_k + C_k' s_k)\|^2 - \|V_k(s_k^{\mathsf{gcp}})(C_k + C_k' s_k^{\mathsf{gcp}})\|^2 \; \Big|, \quad \text{and} \\
\mathrm{III} &= \Big| \; \|V_k(s_k^{\mathsf{gcp}})(C_k + C_k' s_k^{\mathsf{gcp}})\|^2 - \|V_k(s_k^{\mathsf{gcp}})(C_k + C_k' s_k)\|^2 \; \Big|.
\end{aligned}$$

From the continuity, we can easily show that

$$\mathrm{I} \leq O(\|s_k\|^2) = O(\Delta_k^2) = o(\Delta_k). \tag{8.6}$$

On the other hand,

$$\begin{aligned}
\mathrm{II} &= \Big| \; \|W_k(C_k + C_k' s_k)\|^2 - \|V_k(s_k^{\mathsf{gcp}})(C_k + C_k' s_k^{\mathsf{gcp}})\|^2 \; \Big|, \\
&= \Big| \; \|W_k(C_k + C_k' s_k)\|^2 - 2\Psi_k(0) + 2\Psi_k(0) - 2\Psi_k(s_k^{\mathsf{gcp}}) \; \Big|,
\end{aligned}$$

where Ψ_k is given by (6.2). Using the mean-value theorem on the real function $\Psi_k(t s_k^{gcp})$ on [0,1], we have

$$\begin{aligned}
\mathrm{II} &\leq \Big| \; \|W_k(C_k + C_k' s_k)\|^2 - \|W_k C_k\|^2 + 2\nabla\Psi_k(\xi s_k^{\mathsf{gcp}})^T s_k^{\mathsf{gcp}} \; \Big| \\
&= \Big| \; 2(C_k + \bar{\xi} C_k' s_k)^T W_k C_k' s_k + 2\nabla\Psi_k(\xi s_k^{\mathsf{gcp}})^T s_k^{\mathsf{gcp}} \; \Big| \\
&\leq 2 \Big| \; \nabla\Psi_k(0)^T s_k - \nabla\Psi_k(\xi s_k^{\mathsf{gcp}})^T s_k^{\mathsf{gcp}} \; \Big| + O\left(\|s_k\|^2\right) \\
&\leq 2 \Big| \; \nabla\Psi_k(0)^T s_k - \nabla\Psi_k(0)^T s_k^{\mathsf{gcp}} + \nabla\Psi_k(0)^T s_k^{\mathsf{gcp}} - \nabla\Psi_k(\xi s_k^{\mathsf{gcp}})^T s_k^{\mathsf{gcp}} \; \Big| \\
&\qquad + O\left(\|s_k\|^2\right) \\
&\leq 2 \Big| \; \nabla\Psi_k(0)^T (s_k - s_k^{\mathsf{gcp}}) \; \Big| + O\left(\|s_k\|^2 + \|s_k^{\mathsf{gcp}}\|^2\right) \\
&\leq 2\|\nabla\Psi_k(0)\| \; \|s_k - s_k^{\mathsf{gcp}}\| + O\left(\|s_k\|^2 + \|s_k^{\mathsf{gcp}}\|^2\right),
\end{aligned}$$

where $\xi, \bar{\xi} \in (0,1)$. Also,

$$\|s_k - s_k^{\mathsf{gcp}}\| = \left\| \frac{s_k}{\|s_k\|} - \frac{s_k^{\mathsf{gcp}}}{\|s_k\|} \right\| \|s_k\| = \|u_k - \nu_k \bar{u}_k\| \|s_k\|, \tag{8.7}$$

where u_k is a unit vector in the direction of the vector s_k, $\bar{u}_k$ is a unit vector in the direction of the vector s_k^{gcp}, and $\nu_k = \|s_k^{\mathsf{gcp}}\| / \|s_k\|$. We have $\|\nabla \Psi_k(0)\| = \|{C'_k}^T W_k C_k\|$. Let $\Delta_k \to 0$, $s_k \to 0$. First consider the case $\|{C'_k}^T W_k C_k\| \to 0$. The definition of the dogleg step implies that $\|s_k^{\mathsf{gcp}}\| \leq \|s_k\|$ (see also Dennis and Schnabel [13], Sec. 6.4.2). Hence, $\nu_k \leq 1$ and

$$\|{C'_k}^T W_k C_k\| \; \|u_k - \nu_k \bar{u}_k\| \leq 2 \|{C'_k}^T W_k C_k\| \to 0.$$

On the other hand, if $\|{C'_k}^T W_k C_k\|$ is bounded away from zero, then $\|s_k^{\mathsf{gcp}}\| = \Delta_k$ for all sufficiently small Δ_k (Algorithm 6.2). Since a dogleg strategy is employed to compute s_k, this implies $s_k = s_k^{\mathsf{gcp}}$ for all sufficiently small Δ_k (Algorithm 6.3). Thus, $\|u_k - \nu_k \bar{u}_k\| = 0$ for all sufficiently small Δ_k. Therefore, in either case,

$$\|\nabla \Psi_k(0)\| \; \|s_k - s_k^{\mathsf{gcp}}\| = \|{C'_k}^T W_k C_k\| \|u_k - \nu_k \bar{u}_k\| \|s_k\| = o(\|s_k\|). \tag{8.8}$$

Hence, we have $\|{C'_k}^T W_k C_k\| \|s_k - s_k^{\mathsf{gcp}}\| \leq \bar{\varepsilon}_k \|s_k\|$, where $\bar{\varepsilon}_k \to 0$ as $\Delta_k \to 0$. Therefore, we can write

$$\mathrm{II} \leq \bar{\varepsilon_k} \|s_k\| + O\left(\|s_k^{\mathsf{gcp}}\|^2\right) + O\left(\|s_k\|^2\right) = o\left(\Delta_k\right). \tag{8.9}$$

Finally,

$$\begin{aligned}
\mathrm{III} &= \left| \; \|V_k(s_k^{\mathsf{gcp}})(C_k + C'_k s_k^{\mathsf{gcp}})\|^2 - \|V_k(s_k^{\mathsf{gcp}})(C_k + C'_k s_k)\|^2 \; \right| \\
&= \left| \; C_k^T V_k(s_k^{\mathsf{gcp}}) C_k + 2 C_k^T V_k(s_k^{\mathsf{gcp}}) C'_k s_k^{\mathsf{gcp}} + (s_k^{\mathsf{gcp}})^T {C'_k}^T V_k(s_k^{\mathsf{gcp}}) C'_k s_k^{\mathsf{gcp}} \right. \\
&\qquad \left. - C_k^T V_k(s_k^{\mathsf{gcp}}) C_k - 2 C_k^T V_k(s_k^{\mathsf{gcp}}) C'_k s_k - s_k^T {C'_k}^T V_k(s_k^{\mathsf{gcp}}) C'_k s_k \; \right| \\
&\leq 2 \|{C'_k}^T V_k(s_k^{\mathsf{gcp}}) C_k\| \|s_k - s_k^{\mathsf{gcp}}\| + O(\|s_k^{\mathsf{gcp}}\|^2) + O(\|s_k\|^2).
\end{aligned}$$

Now using an argument similar to the one we used in (8.8), we can also write

$$\|{C'_k}^T V_k(s_k^{\mathsf{gcp}}) C_k\| \|s_k - s_k^{\mathsf{gcp}}\| \leq \hat{\varepsilon}_k \|s_k\|,$$

where $\hat{\varepsilon}_k \to 0$ as $\Delta_k \to 0$. So,

$$\mathrm{III} \leq \hat{\varepsilon}_k \|s_k\| + O\left(\Delta_k^2\right) = o\left(\Delta_k\right). \tag{8.10}$$

Combining (8.6), (8.9), and (8.10), we obtain (8.5). This completes the proof of the lemma. $\square$

The following lemma shows that, at any iteration k, the predicted reduction $Pred_k$ satisfies the fraction of the Cauchy decrease condition obtained by the generalized Cauchy point.

LEMMA 8.4. *At any iteration k, we have*

$$Pred_k \geq \|{C'_k}^T W_k C_k\| \min\left\{\Delta_k, a_2 \|{C'_k}^T W_k C_k\|\right\}, \tag{8.11}$$

where a_2 is a constant that does not depend on k.

Proof. From the definition of $Pred_k$ and $V_k(s_k^{\mathsf{gcp}})$, we have

$$\begin{aligned} Pred_k &= \frac{1}{2}\left[\|W_k C_k\|^2 - \|V_k(s_k^{\mathsf{gcp}})(C_k + C_k' s_k)\|^2\right] \\ &\geq \frac{1}{2}\left[\|W_k C_k\|^2 - \|V_k(s_k^{\mathsf{gcp}})(C_k + C_k' s_k^{\mathsf{gcp}})\|^2\right] \\ &\geq \frac{1}{2}\left[\|W_k C_k\|^2 - \|W_k(C_k + C_k' s_k^{\mathsf{cp}})\|^2\right]. \end{aligned}$$

The rest of the proof is straightforward. See, for example, Powell [29]. □

The following theorem shows that the algorithm is well defined in the sense that it will never loop ad infinitum without finding an acceptable step.

THEOREM 8.5. *Let Assumptions* 1, 2, *and* 3′ *hold. At any iteration* k, *either the point* x_k *satisfies* $\nabla\Phi(x_k) = 0$ *or an acceptable step will be found.*

Proof. In the proof of this lemma we use the notation k^j to mean the jth unacceptable trial step of iteration k.

If the point x_k satisfies $\|C'^T_k W_k C_k\| = 0$, then it is a stationary point to the problem and there is nothing to prove.

Assume that $\nabla\Phi(x_k) \equiv \|C_k'^T W_k C_k\| \neq 0$ and suppose that the algorithm loops infinitely without finding an acceptable step. Hence all the trial steps are rejected and we obtain, for all j,

$$(1 - \eta_1) < \left|\frac{Ared_{k^j}}{Pred_{k^j}} - 1\right|. \tag{8.12}$$

For sufficiently large j, and because $\|C_{k^j}'^T W_{k^j} C_{k^j}\| = \|C_k'^T W_k C_k\| \neq 0$, the trust region will have been reduced sufficiently so that inequality (8.11) will have the form

$$Pred_{k^j} \geq \|C_k'^T W_k C_k\| \Delta_{k^j}$$

and Δ_{k^j} will satisfy (8.1). Hence, Lemma 8.2 implies

$$|Ared_k - Pred_k| \leq a_1 \|s_k\|^2.$$

The above two inequalities imply that for j sufficiently large, we have

$$\left|\frac{Ared_{k^j}}{Pred_{k^j}} - 1\right| = \frac{|Ared_{k^j} - Pred_{k^j}|}{Pred_{k^j}} \leq O(\Delta_{k^j}).$$

This means that, as $j \to \infty$, $\Delta_{k^j} \to 0$ and $|\frac{Ared_{k^j}}{Pred_{k^j}} - 1| \to 0$. This contradicts (8.12). Hence, j cannot go to infinity. But this contradicts the supposition that the algorithm loops infinitely without finding an acceptable step and means that, after finitely many rejected trial steps, an acceptable one will be found. This completes the proof. □

Now we present our main global convergence result. We show that the algorithm will converge to a stationary point of problem (1.2). In particular, we show that $\lim_{k\to\infty} \nabla\Phi(x_k) = 0$, where $\Phi(x)$ is given by (2.4). This is equivalent to proving that $\lim_{k\to\infty} \|C_k'^T W_k C_k\| = 0$.

THEOREM 8.6. *Let Assumptions* 1, 2, *and* 3′ *hold. The algorithm generates a sequence of points* $\{x_k\}$ *that satisfies*

$$\lim_{k\to\infty} \nabla\Phi(x_k) = 0.$$

Proof. Suppose that $\limsup_{k\to\infty} \|C_k'^T W_k C_k\| = \varepsilon_0 > 0$. Then there exists an infinite sequence of indices $\{k_j\}$, such that $\|C_k'^T W_k C_k\| > \frac{\varepsilon_0}{2}$ for all $k \in \{k_j\}$.

Let $\hat{k}$ be such that $\hat{k} \in \{k_j\}$. Since from Lemma 4.1, $\nabla\Phi$ is Lipschitz continuous in Ω, we have, for any $x \in \Omega$,

$$\|C'(x)^T W(x) C(x)\| \geq \|C_{\hat{k}}'^T W_{\hat{k}} C_{\hat{k}}\| - a_0 \|x - x_{\hat{k}}\|.$$

This implies that for all x that satisfy $\|x - x_{\hat{k}}\| \leq \frac{\|C_{\hat{k}}'^T W_{\hat{k}} C_{\hat{k}}\|}{2a_0} \equiv \sigma$, we have

$$\|C'(x)^T W(x) C(x)\| \geq \frac{1}{2} \|C_{\hat{k}}'^T W_{\hat{k}} C_{\hat{k}}\| > \frac{\varepsilon_0}{4}.$$

Consider the ball $\mathcal{B}_\sigma = [x : \|x - x_{\hat{k}}\| \leq \sigma]$.

If $x_k \in \mathcal{B}_\sigma$ for all $k \geq \hat{k}$, then from (8.11), we have

$$Pred_k \geq \|C_k'^T W_k C_k\| \min\left\{\Delta_k, a_2 \|C_k'^T W_k C_k\|\right\}. \tag{8.13}$$

Because $x_k \in \mathcal{B}_\sigma$, we have $\|C_k'^T W_k C_k\| \geq \frac{\varepsilon_0}{4}$. Hence, for all $k \geq \hat{k}$

$$Pred_k \geq \frac{\varepsilon_0}{4} \min\left[\Delta_k, a_2 \frac{\varepsilon_0}{4}\right]. \tag{8.14}$$

If there were no acceptable steps for all $k \geq \hat{k}$, a contradiction to Theorem 8.5 would arise. Hence there exists an infinite sequence of acceptable steps. For any such k,

$$\Phi_k - \Phi_{k+1} = Ared_k \geq \eta_1 Pred_k.$$

Since Φ_k is bounded below and decreasing, $\Phi_k - \Phi_{k+1} \to 0$, and we obtain, using the above inequality and (8.14),

$$\liminf_{k\to\infty} \Delta_k = 0. \tag{8.15}$$

Hence, using Lemmas 8.3 and 8.4, the above limit implies that

$$\lim_{k\to\infty} \left| \frac{Ared_k}{Pred_k} - 1 \right| = \lim_{\Delta_k \to 0} \left| \frac{Ared_k - Pred_k}{Pred_k} \right| = 0.$$

However, the updating rules for Δ_k increase Δ_k if $\frac{Ared_k}{Pred_k} > \eta_2$. Thus Δ_k cannot converge to zero. But this contradicts (8.15) and means that all the iterates cannot stay inside $\mathcal{B}_\sigma$.

Let $l+1$ be the first index greater than $\hat{k}$ such that x_{l+1} does not lie inside the ball $\mathcal{B}_\sigma$. Hence

$$\begin{aligned}
\Phi_{\hat{k}} - \Phi_{l+1} &= \sum_{k=\hat{k}}^{l} \{\Phi_k - \Phi_{k+1}\} = \sum_{k=\hat{k}}^{l} Ared_k \\
&\geq \eta_1 \sum_{k=\hat{k}}^{l} Pred_k \geq \eta_1 \frac{\varepsilon_0}{4} \min\left[\sum_{k=\hat{k}}^{l} \Delta_k, a_2 \frac{\varepsilon_0}{4}\right] \\
&\geq \eta_1 \frac{\varepsilon_0}{4} \min\left[\frac{\varepsilon_0}{4a_0}, a_2 \frac{\varepsilon_0}{4}\right].
\end{aligned}$$

As l and $\hat{k}$ go to infinity, $\Phi_{\hat{k}} - \Phi_{l+1}$ goes to zero, which contradicts the supposition that $\varepsilon_0 > 0$. This proves the theorem. □

9. Numerical examples. In this section, we report our preliminary numerical experience with the two algorithms. The numerical experiments were done on a Sun 4/490 workstation running SunOS operating system release 4.1.3 with 64 megabytes of memory. The programs were written in MATLAB and run under MATLAB version 4.2a with machine epsilon about 10^{-16}. More numerical investigation with larger dimensional problems is needed to fully understand the behavior of the two algorithms, but the results given here are very encouraging when compared to LANCELOT on the same problems.

9.1. Algorithm parameters. Successful termination means that the termination condition of the algorithm, $\|C_k'^T W_k C_k\| \le \varepsilon_{tol} = 10^{-6}$, is met. On the other hand, if the algorithm generates a step of length less than 10^{-10}, the number of iterations is greater than 75, or the number of function evaluations is greater than 100, then it is considered an unsuccessful termination.

The initial trust-region radius for the two algorithms is set to $\Delta_0 = \|s_0^{\mathsf{cp}}\|$. The values of the parameters used for updating the trust-region radius in sections 5.3 and 6.4 are $\eta_1 = 10^{-4}$, $\eta_2 = 0.1$, $\eta_3 = 0.25$, $\eta_4 = 0.75$, $\alpha_1 = 0.3$, $\alpha_2 = 2$, and $\alpha_3 = 4$.

9.2. Numerical results. In this section, we report the numerical results for our two trust-region algorithms described in sections 5 and 6. The problems that we tested are the constraint sets of a subset of the Constrained and Unconstrained Testing Environment (the CUTE collection [3]). Some of these problems are from Schittkowski [33]. See also Hock and Schittkowski [18] and Schittkowski [33] for more test problems. We used a MATLAB interface written by Branch (see [4]).

The symbol * is added to the name of some problems to indicate that the corresponding problems have been modified by adding infeasible simple bounds on their variables. The results are summarized in Tables 9.1 and 9.2. Note that $m > n$ for several problems.

In Table 9.1, columns 1–4 give the data of the problem. In particular, the first column gives the problem name. The second column gives the dimension (number of variables) of the problem. The third and fourth columns give the number of equalities and the number of inequalities, respectively. In the fifth and sixth columns we list, respectively, the average number of iterations and the average number of function evaluations needed by the single-model algorithm to converge from different starting points to points that satisfy the stopping criterion. In the seventh and eighth columns we list the corresponding results for the multimodel algorithm. The average number of starting points tried for each test problem was about seven points, and they were chosen randomly.

We also compared our two trust-region algorithms against LANCELOT (release A). LANCELOT is a Fortran package for large-scale nonlinear optimization written by Conn, Gould, and Toint [11].

The test problems that we used are from the CUTE collection with the default starting points. To make the comparison simpler, we selected the subset of the CUTE test problems that have no objective function. LANCELOT in this case will find a feasible point. We note here that we used LANCELOT with all its parameters set to their default values.

In Table 9.2, we report the results of the single-model algorithm and the results obtained using LANCELOT. For the test problems used in Table 9.2, the results obtained by the multimodel algorithm are either the same as those obtained by the single-model algorithm or there is a slight edge in favor of the multimodel algorithm, so the results would even be slightly more in our favor had we given the multimodel results. Remember that our single and multimodel algorithms are identical when there are no inequalities.

In the second table, we report the number of function evaluations and the number of gradient evaluations used. Indeed, the numbers of function and gradient evaluations are good measures for the number of trial steps and the number of acceptable steps. In particular, the number of function evaluations is greater by one than the number of trial steps computed by the algorithm. The number of gradient evaluations is greater by one than the number of acceptable steps used by the algorithm to converge from the default starting points to points that satisfy the stopping criterion.

Again, in Table 9.2, columns 1–4 give the data of the problem. In columns 5 and 6, we list, respectively, the number of function evaluations and the number of gradient evaluations for the single-model algorithm. In columns 7 and 8, we list the corresponding results for the LANCELOT. Note that the maximum number of function evaluations allowed by LANCELOT is 1000.

In most of the test problems reported in Table 9.2, the number of function evaluations and the number of gradient evaluations obtained by our trust-region algorithm are better than those obtained by LANCELOT. This gives some indication about the viability of our approach. However, we believe that our two algorithms need to be refined with efficiency in mind to be suitable for large-scale problems. We also expect that the roughly 10% better performance of the multimodel over the single-model algorithm would be larger in the large-scale case with many inequality constraints (see section 10).

10. Concluding remarks. We have introduced two new trust-region algorithms for finding a feasible point of a set of equalities and inequalities. A one-sided least-squares formulation of the problem is described. The formulation is free of arbitrary parameters and possesses sufficient smoothness to exploit the robustness of the two algorithms. The first algorithm is a single-model algorithm. The second one is a multimodel algorithm. Global convergence results for the two algorithms are presented. It is shown that these two algorithms are globally convergent to a first-order stationary point. Another novelty is that we have given a global convergence analysis for an algorithm based on a local model that does not match function or gradient information.

We point out that our global results give us only first-order stationary point convergence. Since we are using a least-squares formulation of problem (1.1), (i.e., solving problem (1.2)), there is the possibility that the algorithm will converge to a stationary point x_* with $\Phi(x_*) > 0$. This can happen when the rank of the matrix $W_*C'(x_*)$ is less than the number of equalities and the active inequalities at the solution x_*.

We have reported preliminary numerical results with the two algorithms. For large-scale problems, computing an accurate minimum norm solution in high dimensions is probably too costly. Nevertheless, adapting an iterative method with a truncation procedure can reduce the cost of the minimum norm solution. We believe that there is considerable scope for modifying and adapting the basic ideas presented in this paper to the large-scale setting. A more comprehensive computational investigation of the two algorithms, particularly for large problems, needs to be done. It will be presented in a subsequent paper.

For future work, there are some questions that we would like to answer:

The algorithms that were developed in this paper are for finding a feasible point of a set of equalities and inequalities. An important question, which is also a research

Table 9.1
Computational results for the two algorithms.

Problem data				Single-model		Multimodel	
Prob. name	n	$\|E\|$	$\|I\|$	iter.	nfunc	iter.	nfunc
AGG	163	36	615	42	61	51	76
BDVALUE	12	10	4	5	6	5	6
DALLASM	196	151	392	8.5	11.3	7	8.8
DALLASS	46	31	92	8.3	11.2	6.3	8
EIGENA	110	110	110	18	24	17	21.5
HATFLDG	25	25	40	7	8	7	8
HS6*	2	1	2	18	37.6	18	37.6
HS7*	2	1	4	16.6	17.6	16.6	17.6
HS10	2	0	1	11.2	12.2	11.2	12.2
HS11	2	0	1	9.8	12.2	9.8	12.2
HS12	2	0	1	7.6	8.6	7.6	8.6
HS14	2	0	1	9.6	11.2	11	14
HS22	2	0	2	10.6	13	9.6	12
HS29	3	0	1	7.2	8.2	7.2	8.2
HS40*	4	3	8	16.8	19.6	16.6	20
HS43	4	0	3	10.8	14	4	5
HS60	3	1	6	22.2	26.4	14.4	17.6
HS78*	5	2	10	16	22.8	15.2	21.4
HS80	5	3	6	14	17.2	15.4	19
HS99EXP	31	21	20	14	15	16	18
HS113	10	0	8	10	12.4	14.8	19.8
HYDCAR20	99	99	0	17	22	17	22
HYDCAR6	29	29	0	8	9	8	9
LEAKNET	156	153	82	7.7	10	7.7	10
LINSPANH	97	33	194	34.3	46.7	38.3	54
METHANB8	31	31	0	7.6	8.6	7.6	8.6
NET1	48	38	75	14	17.5	14.5	18
PRODPL0	60	20	69	23.5	37.5	20	19.5
PRODPL1	60	20	69	34	56.5	39	49.5
QPNBLEND	83	43	114	36.5	57.5	17.5	23.5
HS 226	2	0	4	8.6	13	3	4
S227	2	0	2	11.4	13.2	8.8	10.4
S262	4	1	7	4	6	3.4	4.6
S263	4	2	2	12.6	14.6	11.4	13
S353	4	1	6	8	14.4	11.4	21.4
S354	4	0	5	2	3	2	3
SMBANK	117	64	234	12	17	8	11.5
SPANHYD	97	33	194	34	42.5	39	54.5
SSEBLIN	194	48	388	7	11	11	16
SSEBNLN	194	72	388	42	61	32	48
		Totals		665.9	915	619.8	822.5
		Averages		15.5	21.3	14.4	19.1

topic, is how to use any of the two algorithms suggested in this paper as an active-set strategy in a trust-region algorithm for nonlinear programming. Certainly, this is an important topic that deserves to be investigated because problems with more constraints than variables arise in engineering applications. See, for example, [2]. Our numerical results encourage us to extend this approach of treating inequalities to an active set scheme for nonlinear programming.

A related important question is how to generalize the multimodel theory developed in this paper to general nonlinear programming trust-region algorithms.

Acknowledgments. We wish to thank Amr El-Bakry for many useful discussions and comments and Luis Vicente for his careful reading of an earlier version

TABLE 9.2
The single-model algorithm versus LANCELOT.

Problem data				Single-model		LANCELOT	
Prob. name	n	$\|E\|$	$\|I\|$	nfunc	ngrad	nfunc	ngrad
AIRCFTA	8	5	6	5	5	5	5
ARGAUSS	3	15	0	3	3	3	3
ARGLINA	100	200	0	2	2	3	3
ARGLINB	10	20	0	2	2	2	2
ARGTRIG	10	10	0	6	6	9	8
ARTIF	12	10	4	6	6	14	11
BDVALUE	12	10	4	5	5	3	3
BOOTH	2	2	0	3	3	4	4
BRATU2D	49	25	48	6	6	5	5
BRATU3D	27	1	52	5	5	5	5
BROYDNBD	10	10	0	7	7	12	11
BROYDN3D	10	10	0	6	6	6	6
CBRATU2D	32	8	48	5	5	5	5
CBRATU3D	54	2	104	5	5	5	5
CHANDHEQ	10	10	10	11	11	14	14
CHEMRCTA	10	10	10	7	7	10	9
CHEMRCTB	10	10	10	8	8	9	8
CLUSTER	2	2	0	8	8	12	11
CHEMRCTB	10	10	10	8	8	9	8
EIGENA	110	110	110	9	8	17	16
GOTTFR	2	2	0	6	6	31	26
HATFLDG	25	25	0	8	8	16	15
HIMMELBC	2	2	0	2	2	5	5
HIMMELBD	2	2	0	62	39	39	34
HIMMELBE	3	3	0	4	4	4	4
HYDCAR20	99	99	0	21	18	1000	982
HYDCAR6	29	29	0	9	9	1000	981
HYPCIR	2	2	0	6	6	8	7
INTEGREQ	52	50	4	4	4	4	4
METHANB8	31	31	0	8	8	194	194
METHANL8	31	31	0	9	9	630	622
MSQRTB	9	9	0	6	6	14	12
POWELLSQ	2	2	0	12	10	5	5
RCCIPE	3	3	0	11	11	17	17
SEMICON2	12	10	30	30	21	40	37
SPMSQRT	28	44	0	10	8	13	11
TRIGGER	7	6	2	74	49	22	20
ZANGWIL3	3	3	0	3	3	8	8

of this paper. We also wish to thank three referees and the editor for their helpful reports.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammaling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK User's Guide*, SIAM, Philadelphia, PA, 1992.

[2] J. T. Betts, W. P. Huffman, and D. P. Young, *An investigation of algorithm performance for aerodynamic design optimization*, Tech. report BCSTECH-94-061, Boeing Computer Services, Boeing Company, Seattle, WA, 1994.

[3] I. Bongartz, A. R. Conn, N. Gould, and Ph. L. Toint, *CUTE: Constrained and unconstrained testing environment*, Tech. report 18860, IBM T.J. Watson Research Center, Yorktown, NY, 1993.

[4] M. A. Branch, *Getting CUTE with Matlab*, Tech. report CTC94TR194, Department of Com-

puter Science, Cornell University, Ithaca, NY, 1994.

[5] J. Burke, *Algorithms for Solving Finite Dimensional Systems of Nonlinear Equations and Inequalities that have Both Global and Quadratic Convergence Properties*, Tech. report ANL/MCS-TM-54, Mathematics and Computer Science Division, Argonne National Laboratory, Chicago, IL, 1985.

[6] J. Burke and S. P. Han, *A Gauss-Newton approach to solving generalized inequalities*, Math. Oper. Res., 11 (1986), pp. 632–643.

[7] J. V. Burke and M. C. Ferris, *A Gauss-Newton method for convex composite optimization*, Tech. report 1176, Center for Parallel Optimization, Computer Sciences, University of Wisconsin, Madison, WI, 1993.

[8] R. Carter, *Multi-Model Algorithms for Optimization*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1986.

[9] R. Carter *On the global convergence of trust region algorithms using inexact gradient information*, SIAM J. Numer. Anal., 28 (1991), pp. 251–265.

[10] A. R. Conn, N. I. Gould, A. Sartenaer, and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints*, SIAM J. Optim., 3 (1993), pp. 164–221.

[11] A. R. Conn, N. I. Gould, and Ph. L. Toint, *LANCELOT: a Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer Ser. Comput. Math. 17, Springer-Verlag, Heidelberg, Berlin, New York, 1992.

[12] J. W. Daniel, *Newton's method for nonlinear inequalities*, Numer. Math., 40 (1973), pp. 381–387.

[13] J. E. Dennis, Jr., and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[14] J. E. Dennis, Jr., and R. B. Schnabel, *A View of Unconstrained Optimization*, Handbooks Oper. Res. Management Sci. 1, North–Holland, Amsterdam, 1989.

[15] U. M. Garcia-Palomares, *On the Minimax Solution of a Nonlinear System of Mixed Equalities and Inequalities*, Tech. report, Departmento de procesos y sistemas, Universidad Simon Bolivar, Caracas, Venezuela, 1983.

[16] U. M. Garcia-Palomares and A. Restuccia, *A global quadratic algorithm for solving a system of mixed equalities and inequalities*, Math. Programming, 21 (1981), pp. 290–300.

[17] G. H. Golub and U. von Matt, *Quadratic constrained least squares and quadratic problems*, Numer. Math., 59 (1991), pp. 561–580.

[18] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Econom. and Math. Systems 187, Springer-Verlag, Berlin, 1981.

[19] E. H. Kaufman and G. D. Taylor, *Linearly constrained generalized rational approximation*, in Approximation Theory VI: Vol. 2, C. K. Chui, L. L. Schumaker, and J. D. Ward, eds., Academic Press, New York, 1989, pp. 353–356.

[20] K. Levenberg, *A method for the solution of certain problems in least squares*, Quart. Appl. Math., 2 (1944), pp. 164–168.

[21] D. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters*, SIAM J. Appl. Math., 11 (1963), pp. 431–441.

[22] J. J. Moré, *The Levenberg-Marquardt algorithm: Implementation and theory*, in Proceedings of the Dundee Conference on Numerical Analysis, G. A. Watson, ed., Springer-Verlag, Berlin, New York, 1978, pp. 105–116.

[23] J. J. Moré, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming, The State of the Art, A. Bachem, M. Grotschel, and B. Korte, eds., Springer-Verlag, New York, 1983, pp. 258–287.

[24] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *User Guide for MINPACK-1*, Tech. report ANL-80-74, Argonne National Laboratory, Argonne, IL, 1980.

[25] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[26] M. R. Osborne, *Finite Algorithms for Optimization and Data Analysis*, John Wiley, New York, 1985.

[27] M. R. Osborne, S. A. Pruess, and R. S. Womersley, *Concise representation of generalized gradients*, J. Austral. Math. Soc. Ser. B, 28 (1986), pp. 57–74.

[28] B. T. Polyak, *Gradient methods for solving equations and inequalities*, USSR Comput. Math., 4 (1964), pp. 17–32.

[29] M. J. D. Powell, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. Mangasarian, R. Meyer, and S. Robinson, eds., Academic Press, New York, 1975, pp. 1–27.

[30] M. J. D. Powell, *On the global convergence of trust region algorithms for unconstrained*

optimization, Math. Programming, 29 (1984), pp. 297–303.

[31] B. N. Pshenichnyi, *Newton's method for the solution of systems of equalities and inequalities*, Math. Notes Acad. Sci. USSR, 8 (1970), pp. 827–830.

[32] S. M. Robinson, *Extension of Newton's method to nonlinear functions with values in a cone*, Numer. Math., 19 (1972), pp. 341–347.

[33] K. Schittkowski, *More Test Examples for Nonlinear Programming Codes*, Lecture Notes in Econom. and Math. Systems 282, Springer-Verlag, Berlin, 1987.

[34] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[35] G. D. Taylor, *Uniform approximation with side conditions*, in Approximation Theory, G. G. Lorentz, ed., Academic Press, New York, 1973, pp. 481–484.

[36] Ph. L. Toint, *Global convergence of a class of trust region methods for non-convex minimization in Hilbert spaces*, IMA J. Numer. Anal., 8 (1988), pp. 231–252.

[37] Ph. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, Tech. report 80/4, Departément de Mathématique, Facultés Universitaires de Namur, Belgium, 1980.

SIAM J. OPTIM.
Vol. 9, No. 2, pp. 316–326

OPTIMAL SIGNAL SETS FOR NON-GAUSSIAN DETECTORS[*]

MARK S. GOCKENBACH[†] AND ANTHONY J. KEARSLEY[‡]

Abstract. Identifying a maximally separated set of signals is important in the design of modems. The notion of optimality is dependent on the model chosen to describe noise in the measurements; while some analytic results can be derived under the assumption of Gaussian noise, no such techniques are known for choosing signal sets in the non-Gaussian case. To obtain numerical solutions for non-Gaussian detectors, minimax problems are transformed into nonlinear programs, resulting in a novel formulation yielding problems with relatively few variables and many inequality constraints. Using sequential quadratic programming, optimal signal sets are obtained for a variety of noise distributions.

Key words. optimal design, inequality constraints, sequential quadratic programming

AMS subject classifications. 90C90, 94A12, 94A13

PII. S1052623496306553

1. Introduction.

The transmission of digital information requires signals (finite time series) that can be distinguished from one another in the presence of noise. These signals may be constrained by bounds on their energy or amplitude; the degree to which they can be distinguished depends on the distribution of noise.

We study the design of optimal signal sets under amplitude constraints and in the presence of non-Gaussian noise. We will call a signal set optimal when, roughly speaking, the *largest* probability of mistaking any one signal for any other is minimal. For this reason, as we show below, this problem is naturally formulated as a smooth and twice continuously differentiable minimax problem.

We shall assume that M signals

$$s_0, s_1, \ldots, s_{M-1}$$

are to be constructed, where each signal is to be a linear combination of K given signals

$$\phi_0, \phi_1, \ldots, \phi_{K-1}.$$

Moreover, it is assumed that each ϕ_k is a time series of length N and that $\{\phi_k\}$ is an orthonormal set under the Euclidean inner product. We denote the components of ϕ_k by $\phi_k(n)$, $n = 0, 1, \ldots, N-1$, and the components of s_m similarly. The unknowns to be determined are the weights $\{\alpha_{mk}\}$ defining the signals:

$$s_m = \sum_{k=0}^{K-1} \alpha_{mk}\phi_k, \quad m = 0, 1, \ldots, M-1.$$

[*]Received by the editors July 12, 1996; accepted for publication (in revised form) August 15, 1997; published electronically March 17, 1999. This work was performed by an employee of the U.S. Government or under U.S. Government contract. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights. Contributions of the National Institute of Standards and Technology are not subject to copyright in the United States.

http://www.siam.org/journals/siopt/9-2/30655.html

[†]Department of Mathematics, University of Michigan, Ann Arbor, MI 48109 (gock@caam.rice.edu).

[‡]Mathematical Sciences Department, Carnegie Mellon University, Pittsburgh, PA 15213-3890 (anthonyk@andrew.cmu.edu).

A signal set is often referred to as a *constellation*.

In the special case of only two basis functions, ϕ_0, ϕ_1, the signals can be represented in the plane by the coefficients $(\alpha_{00}, \alpha_{01}), \ldots, (\alpha_{M-1,0}, \alpha_{M-1,1})$. In the case of Gaussian noise, it turns out that the problem reduces to maximizing the minimum Euclidean distance between any two signals (subject to constraints on the energy or amplitudes of the signals). For this reason, heuristic methods have been used to design good signal sets.

Typically, these heuristics have taken the form of choosing the points lying on a lattice and somehow densely packing them within a fixed region of the plane. Identifying these lattice-based constellations associated with low average energies has been an active area of research (examples and pictures of these constellations can be found in [8], [10], and [11]).

Although we are not aware of previous attempts to find optimal constellations according to the criteria we describe below, related problems have been investigated. The most famous is the sphere-packing problem of communication theory (see [1]); this requires a constellation which maximizes the probability of detection under Gaussian noise. Modern research into this question has focused on the case in which the signals are chosen from a large dimensional space (in particular, an important research topic has been the strong simplex conjecture, which deals with the case $M = K + 1$; see [9]). We are concerned with the case in which the signals are chosen from a small dimensional space ($K = 2$ or 3).

In the remainder of this paper, we show that this problem can be formulated as a smooth nonlinear programming problem with relatively few variables but many inequality constraints. This problem is solved using a sequential quadratic programming (SQP) algorithm.

In section 2, we explain the formulation of the optimization problem that describes optimal signal sets. In section 3, we describe the SQP algorithm employed to solve these problems. The noise distributions used in our computations are described in section 4, and in section 5, numerical tests and results are presented. We conclude the paper with observations and comments on future work in section 6.

2. Problem formulation. As mentioned above, we wish to find a signal set which minimizes the largest probability of mistaking any one for any other. This notion of optimality is therefore based on *hypothesis testing* (see, for example, [14]). We consider for the moment that one of two signals, s_0 and s_1, is to be transmitted and that the received signal is denoted by y. We assume further that the transmitted signal is corrupted by independent, identically distributed (i.i.d.) additive noise drawn from some fixed distribution with probability density function (pdf) p_N. In other words,

$$y = s_m + \eta, \ m = 0 \text{ or } 1.$$

We assume that the a priori probabilities of s_0 and s_1 being transmitted are P_0 and P_1, respectively, and that there is a cost C_m associated with detecting signal s_0 when s_1 is actually present (a *miss*) and a cost C_f associated with detecting s_1 when s_0 is present (a *false alarm*). It is then easy to show (see [14]) that the expected cost, or *risk*, is minimized by detecting s_0 whenever

$$\frac{p(y|s_0)}{p(y|s_1)} > \frac{P_1 C_m}{P_0 C_f},$$

and otherwise detecting s_1. By taking the logarithm of both sides and using the fact that the noise is i.i.d., we obtain the following *optimal detector*: detect s_0 whenever

$$\frac{1}{N}\sum_{n=0}^{N-1}\log\frac{p_N(y(n)-s_0(n))}{p_N(y(n)-s_1(n))}>\gamma \tag{2.1}$$

(where γ is the threshold determined by the a priori probabilities and the costs of errors); otherwise detect s_1.

Now assume that s_0 was actually transmitted (so that $y=s_0+\eta$) and let $\Delta s=s_0-s_1$; the optimal detector then computes

$$\frac{1}{N}\sum_{n=0}^{N-1}\log\frac{p_N(\eta(n))}{p_N(\eta(n)+\Delta s(n))}.$$

The expected value of the nth term is

$$K_N(\Delta s(n))=\int\log\left[\frac{p_N(\tau)}{p_N(\tau+\Delta s(n))}\right]p_N(\tau)d\tau;$$

this quantity is the Kullback–Leibler distance between the noise density and the noise density shifted by $\Delta s(n)$ (see [15]).

Thus, if s_0 is actually transmitted, the expected value of the sum in (2.1) is

$$\frac{1}{N}\sum_{n=0}^{N-1}K_N(\Delta s(n)). \tag{2.2}$$

If one assumes that p_N is symmetric, as we will, it is easy to show that if s_1 is transmitted, the expected value of the sum in (2.1) is the negative of (2.2). Therefore, the probability of detecting the correct signal increases with (2.2).

From this discussion, we see that we wish to choose the signals $s_0, s_1, \dots, s_m$ so that

$$\min_{m_1\neq m_2}\sum_{n=0}^{N-1}K_N(s_{m_1}(n)-s_{m_2}(n))$$

is maximized. For physical reasons, either the average power (energy—L^2 norm) or the peak power (amplitude—L^∞ norm) of the signals must be constrained. In this paper, we are concerned with amplitude constraints.

We are thus faced with a constrained minimax problem. Due to the difficulty of solving such problems, we rewrite it as a smooth nonlinear program (NLP) by introducing an auxiliary variable t:

$$\min_{t,\alpha} -t^2 \tag{2.3}$$

$$\text{s.t.}\sum_{n=0}^{N-1}K_N(s_{m_1}(n)-s_{m_2}(n))\geq t^2,\ m_1<m_2, \tag{2.4}$$

$$s_m(n)^2\leq C^2,\ n=0,\dots,N-1,\ m=0,\dots,M-1, \tag{2.5}$$

$$t\geq 0, \tag{2.6}$$

where $C>0$ is the bound on the amplitudes of the signals. This problem has $MK+1$ variables and $M(M+1)/2+MN+1$ inequality constraints. A typical problem would have $K=2, M=16, N=50$, giving 33 variables and 937 constraints.

Difficulties arise when one tries to solve the above NLP (2.3) with standard algorithms. The fact that there are far fewer variables than constraints results in three specific difficulties:

- There are many “almost” binding constraints at the solution.
- The linearized constraints are often inconsistent.
- The boundary of the feasible region is noticeably nonlinear.

The proximity of near-binding constraints to the solution suggests that correct identification of the active set becomes more difficult near the solution. In turn this suggests that the region of rapid local convergence of the iteration sequence $\{x_k, \lambda_k\}$ will be “small.” The lack of consistent linear inequalities complicates the calculation of the SQP step (the solution of the quadratic subproblem). As observed in previous works, ([12] and others), nonlinear feasible regions can result in small acceptable steps if penalty parameters become small. Iterates will follow too closely a feasible region boundary that leads away from optimality. All three of these issues were relevant to the solution of the optimal signal set problem presented here.

3. Nonlinearly constrained optimization. Many algorithms have been developed for the solution of smooth, inequality-constrained optimization problems. Among the most popular methods is the SQP family of algorithms (see [6] for a review of these methods). Given an estimate of the solution, an SQP algorithm progresses by solving a quadratic program (QP), which is defined by a local quadratic model of the objective function and linearized constraints. The solution to the quadratic program is then used to construct an improved estimate of the solution. Many different SQP algorithms can be constructed by varying the algorithm for solving the QP, the Lagrange multiplier estimates, and the globalization strategy.

One version of the SQP algorithm, proposed by Boggs, Kearsley, and Tolle (see, for example, [4]), appears to be well suited for solving this class of NLPs. The algorithm employs a combined trust-region and merit function (see [5]) line search procedure for globalization. A modern interior point method called O3D, **O**ptimal **3**-**D**imensional subspace method (see Boggs et al. [3]), is used to solve the quadratic programming subproblems. It appears that O3D is quite compatible with the globalization procedure (e.g., the steps produced by the O3D algorithm decrease the merit function and do not impede convergence).

A nonstandard feature of this algorithm is useful because of the need to solve NLPs with very small and highly nonlinear feasible regions. A perturbation is added to the right-hand side of the system of linearized constraints to guarantee that this linear system is always consistent. Similar constraint relaxations have appeared in the literature before (see, for instance, the papers by Biggs [2], Tone [19], and Powell [17], among others). The relaxation procedure we employ is similar to methods contained in papers mentioned above, with minor modifications (and can be found in [16]). When far from feasibility, violated linearized constraints are relaxed enough to guarantee that they form a consistent system of inequalities. This relaxation is obtained by solving a linear programming problem; moreover, the solution to the linear programming problem can then be used as a feasible starting point for the QP. Because O3D is designed to solve either linear or quadratic programming problems, this two-step process can be carried out using one algorithm (and code). This procedure ensures that no “phase I” or infeasible calculations are needed (i.e., no “Big-M” method is needed) for the calculation of the SQP descent direction. Details of the constraint perturbation procedure can be found in [16].

To test the efficacy of the constraint relaxation procedure, we solved the problems

TABLE 1
Noise densities and associated Kullback–Leibler distances.

Name	Density	$K(\Delta s)$
Gaussian	$\frac{\exp(-\tau^2/2\sigma^2)}{\sqrt{2\pi\sigma^2}}$	$\frac{(\Delta s)^2}{2\sigma^2}$
Laplacian	$\frac{\exp(-\lvert\tau\rvert/(\sigma/\sqrt{2}))}{\sqrt{2}\sigma}$	$\frac{\lvert\Delta s\rvert}{\sigma/\sqrt{2}} + \exp(-\frac{\lvert\Delta s\rvert}{\sigma/\sqrt{2}}) - 1$
Hyperbolic Secant	$\frac{\mathrm{sech}(\pi\tau/2\sigma)}{2\sigma}$	$-2\ln(\mathrm{sech}(\pi\Delta s/4\sigma))$
Generalized Gaussian	$\frac{1}{2\Gamma(5/4)A}\exp(-\frac{\tau^4}{A^4})$	$\frac{\Gamma^2(3/4)}{\Gamma^2(1/4)}\left(6\frac{(\Delta s)^2}{\sigma^2} + \frac{(\Delta s)^2}{\sigma^4}\right)$
Cauchy	$\frac{1}{\pi\sigma(1+(\tau/\sigma)^2)}$	$\ln(1 + (\Delta s)^2/r\sigma^2)$

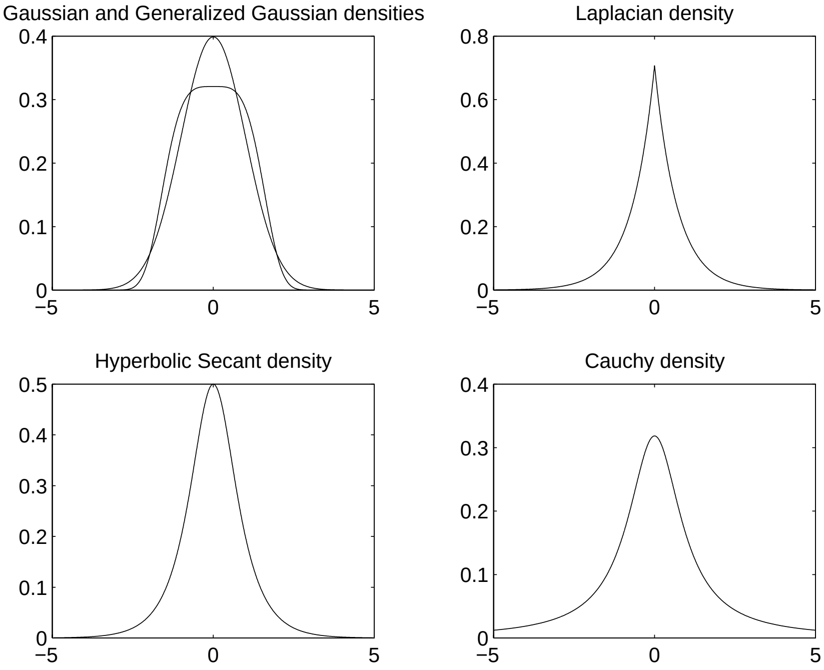


FIG. 1. *Noise densities studied.*

described in this paper twice, first without the relaxation and then with it. The advantages of relaxing the constraints are shown by the numerical results presented in the following section.

4. Noise distributions. The primary purpose of this paper is to investigate non-Gaussian noise distributions. Following Johnson and Orsak (see [15]), we selected the five densities shown in Table 1 (including the Gaussian density for comparison). These densities are graphed in Figure 1, while the associated Kullback–Leibler distances are found in Figure 2.

These densities are chosen to illustrate different possibilities. For example, the Kullback–Leibler distance associated with the Gaussian density is smaller than that of the Laplacian when Δs is small; for large Δs, this relationship is reversed. The hyperbolic secant density leads to a distance function that is similar to that of the

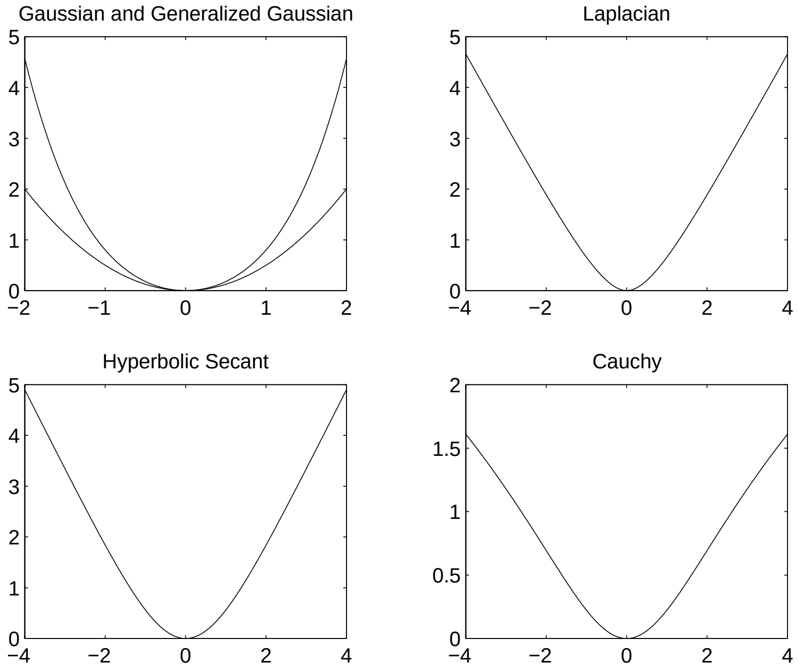


FIG. 2. *Kullback–Leibler distances associated with the various noise densities.*

Gaussian near the origin but close to that of the Laplacian for large Δs. The Kullback–Leibler distance for the generalized Gaussian density grows very rapidly with Δs, while that of the Cauchy density grows very slowly.

5. Numerical tests and results. In this section we summarize the performance of the SQP algorithms on a suite of problems, corresponding to various choices of the noise distribution, the basis signals, and the number of signals, M. For the purpose of these numerical examples, we fix the length of the signals at $N = 50$, the amplitude at $C = \sqrt{10}$, and the number of basis functions at $K = 2$. The bases used were

$$\left\{ \sqrt{\frac{2}{N}} \sin\left(2\pi\omega_1 n/N\right), \sqrt{\frac{2}{N}} \sin\left(2\pi\omega_2 n/N\right) \right\}$$

and

$$\left\{ \sqrt{\frac{2}{N}} \sin\left(2\pi\omega_1 n/N\right), \sqrt{\frac{2}{N}} \cos\left(2\pi\omega_1 n/N\right) \right\},$$

with $\omega_1 = 10$ and $\omega_2 = 11$.

We use analytic first and second derivatives and the least-squares estimate of the Lagrange multipliers (see, for example, Gill, Murray, and Wright [13]).

Problems involving the sine-cosine basis are fundamentally more difficult than those involving the sine-sine basis; this is because of the rotational symmetries in the solution space, and also because of the geometry of certain signal sets. For example,

Table 2

Performance of the unperturbed SQP algorithm on a collection of constellation problems (sine-sine basis).

(M,N,K)	Density	(n,m)	Outer	Inner	min	#/10	$\|\nabla_x L\|$
(8,50,2)	Gaussian	(17,429)	16	201 (34)	-69.7937	0 (214)	2e-12
(8,50,2)	Generalized Gaussian	(17,429)	16	179 (36)	-189.0968	4	7e-10
(8,50,2)	Hyperbolic Secant	(17,429)	65	730 (94)	-61.0936	5	5e-14
(8,50,2)	Laplacian	(17,429)	10	121 (28)	-63.1230	0 (30)	5e-14
(8,50,2)	Cauchy	(17,429)	36	499 (76)	-22.7308	0 (24)	1e-9
(16,50,2)	Gaussian	(33,937)	27	388 (55)	-29.3142	4	5e-12
(16,50,2)	Generalized Gaussian	(33,937)	29	478 (80)	-57.8296	3	7e-11
(16,50,2)	Hyperbolic Secant	(33,937)	10	119 (22)	-29.5770	3	1e-1
(16,50,2)	Laplacian	(33,937)	42	646 (121)	-32.3708	0 (12)	5e-5
(16,50,2)	Cauchy	(33,937)	33	501 (94)	-11.4304	2	6e-2

Table 3

Performance of the unperturbed SQP algorithm on a collection of constellation problems (sine-cosine basis).

(M,N,K)	Density	(n,m)	Outer	Inner	min	#/10	$\|\nabla_x L\|$
*(8,50,2)	Gaussian	(17,429)	48	631(22)	-97.5518	0(21)	5.e-11
*(8,50,2)	Generalized Gaussian	(17,429)	52	679(31)	-264.0240	0(12)	6.e-12
*(8,50,2)	Hyperbolic Secant	(17,429)	89	1199(30)	-83.1955	0(14)	9.e-4
*(8,50,2)	Laplacian	(17,429)	49	645 (21)	-84.4632	1	1 8.e-11
*(8,50,2)	Cauchy	(17,429)	67	888(19)	-22.7308	1	7.e-10
(16,50,2)	Gaussian	(33,937)	26	343 (44)	-39.7452	1	7e-12
(16,50,2)	Generalized Gaussian	(33,937)	100	1170 (110)	-76.1390	1	1e-4
(16,50,2)	Hyperbolic Secant	(33,937)	10	119 (22)	-29.5770	3	1e-1
(16,50,2)	Laplacian	(33,937)	51	611 (121)	-32.3708	0 (12)	5e-5
(16,50,2)	Cauchy	(33,937)	41	417 (31)	-15.6892	1	2e-6

when the signal set contains eight signals, optimality requires that one of the signals lie in the center, with seven signals on a circle around it (see Figure 5). The signal in the center is actually free to move in a small open set without affecting optimality. Part of the numerical difficulty can be alleviated with a small amount of regularization introduced to the objective function as follows:

$$\min_{t,\alpha} - \left(\frac{1}{2}t^2 - \frac{\epsilon}{2}\|s\|_2^2\right) \tag{5.1}$$

$$\text{s.t.} \sum_{n=0}^{N-1} K_N(s_{m_1}(n) - s_{m_2}(n)) \geq t^2,\ m_1 < m_2, \tag{5.2}$$

$$s_m(n)^2 \leq C^2,\ n = 0, \ldots, N-1,\ m = 0, \ldots, M-1, \tag{5.3}$$

$$t \geq 0. \tag{5.4}$$

A value $\epsilon \approx 10^{-6}$ worked well. The problems where this version was employed are denoted by an asterisk.

In Tables 2 and 3 we report the performance of the algorithm described in [4] without the constraint relaxation. Likewise, Tables 4 and 5 give the performance of the algorithm with constraint relaxation. The first column in each table contains the values for the number of signals M, the number of time samples N, and the number of basis functions K. The noise distribution is given in the second column, while the number of variables (n) and constraints (m) in the resulting nonlinear program is

Table 4

Performance of the constraint perturbed SQP algorithm on a collection of constellation problems (sine-sine basis).

(M, N, K)	Density	(n, m)	Outer	Inner	min	#/10	$\|\nabla_x L\|$
(8,50,2)	Gaussian	(17,429)	14	209	-69.7937	1	7e-12
(8,50,2)	Generalized Gaussian	(17,429)	14	198	-189.0968	1	4e-9
(8,50,2)	Hyperbolic Secant	(17,429)	59	801	-61.0936	1	1e-9
(8,50,2)	Laplacian	(17,429)	10	133	-63.1230	2	3e-12
(8,50,2)	Cauchy	(17,429)	31	440	-22.7308	2	8e-8
(16,50,2)	Gaussian	(33,937)	25	292	-29.3142	5	4e-11
(16,50,2)	Generalized Gaussian	(33,937)	27	401	-57.8296	3	5.e-11
(16,50,2)	Hyperbolic Secant	(33,937)	10	122	-29.5770	3	2.e-13
(16,50,2)	Laplacian	(33,937)	40	537	-32.3708	1	1.e-9
(16,50,2)	Cauchy	(33,937)	30	419	-11.4304	2	1.e-9

Table 5

Performance of the constraint perturbed SQP algorithm on a collection of constellation problems (sine-cosine basis).

(M, N, K)	Density	(n, m)	Outer	Inner	min	#/10	$\|\nabla_x L\|$
*(8,50,2)	Gaussian	(17,429)	40	123	-69.7937	4	7e-8
*(8,50,2)	Generalized Gaussian	(17,429)	39	121	-189.0968	4	3.e-8
*(8,50,2)	Hyperbolic Secant	(17,429)	70	1612	-61.0936	8	2.e-10
*(8,50,2)	Laplacian	(17,429)	46	479	-63.1230	1	4.e-7
*(8,50,2)	Cauchy	(17,429)	58	1333	-22.7308	1	3.e-7
(16,50,2)	Gaussian	(33,937)	25	325	-39.7452	2	6.e-10
(16,50,2)	Generalized Gaussian	(33,937)	91	1066	-76.1390	4	4.e-8
(16,50,2)	Hyperbolic Secant	(33,937)	10	122	-29.5770	3	5e-5
(16,50,2)	Laplacian	(33,937)	41	416	-32.3708	1	1.e-6
(16,50,2)	Cauchy	(33,937)	29	410	-15.6892	5	1.e-9

found in the third column. The number of nonlinear or outer iterations required to find the solution is recorded in the fourth column, while the fifth column gives the number of QP iterations required (with the number of phase I iterations in parentheses). In the sixth column we give the value of the putative global minimum, and in the seventh column the number of times this minimum was found in 10 tries, each starting from a randomly generated starting point. In the event that the minimizer was not found in 10 tries, we record in parentheses the number of tries it took to find it. Finally, the last column contains the size of the gradient of the Lagrangian at the computed solution. The algorithm halted when either the 2-norm of the gradient of the Lagrangian became less than or equal to 10^{-6} or the 2-norm of the solution to the quadratic subproblem (the SQP step) fell below 10^{-12}.

The optimal constellations for a subset of our test problems are shown in Figures 3, 4, and 5 (with $M = 8$) and Figures 6, 7, and 8 (with $M = 16$). It is interesting to observe that the symmetric nature of constellations conjectured to be present in optimal solutions (see [15]) is apparent in the current estimates of the solutions.

These problems have features that make them difficult to solve numerically. Not only is the number of variables much smaller than the number of constraints, but there are many constraints that are nearly binding at the solution. Many algorithms, including SQP, give rise to rapid local convergence when iterates enter a neighborhood of the solution and the correct collection of active sets has been identified (see the paper by Robinson [18] for a discussion). This rapid local convergence is especially apparent in the event that one can provide an accurate approximation to the true

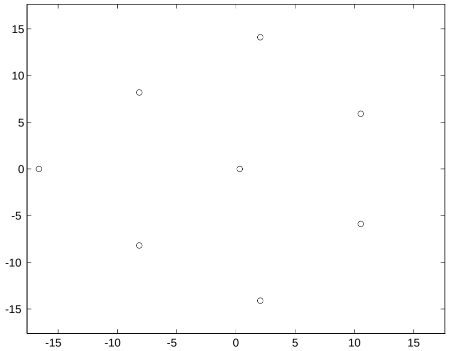


Fig. 3. *Optimal constellation for $M = 8$ with sine-sine basis functions using a Gaussian density.*

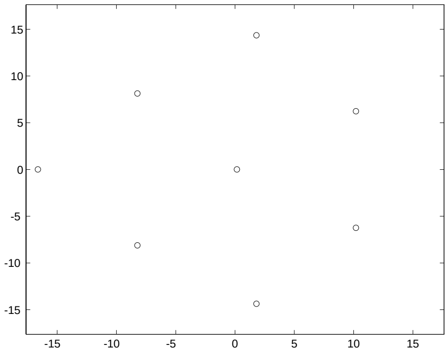


Fig. 4. *Optimal constellation for $M = 8$ with sine-sine basis functions using a generalized Gaussian density.*

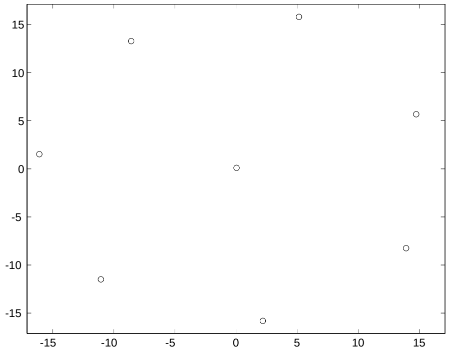


Fig. 5. *Optimal constellation for $M = 8$ with sine-cosine basis functions using a hyperbolic-secant Density.*

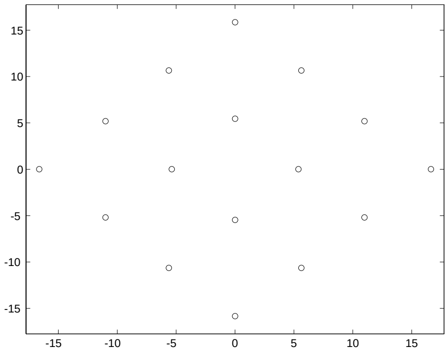


Fig. 6. *Optimal constellation for $M = 16$ with sine-sine basis functions using a Cauchy density.*

Hessian matrix at every iteration, as is the case with our problem. Even though the notion of active sets is less important to our algorithm because our interior point method quadratic program solver, O3D, does not compute active sets, the fact that many of the constraints are nearly binding at the solution has an effect on the size of the neighborhood around the solution where fast local convergence is realized.

In these numerical tests the constraint relaxation substantially improved the performance of the algorithm. This is probably due to the fact that the number of inconsistent subproblems encountered was unusually high. It is worth commenting that the additional cost of employing the relaxation procedure is not large.

6. Conclusions. In this paper we have presented an interesting collection of difficult optimization problems and an NLP formulation of them. This formulation allows a broad arsenal of numerical optimization algorithms and modern enhancements to be employed. While these problems are not “large-scale” by modern computing standards, they are, nonetheless, difficult problems to solve efficiently.

Numerical solutions to these problems were located using an SQP method with and without the constraint relaxation procedure described in [16]. Numerous numerical tests (and the summary of these tests that appear in Tables 2–5) suggest that this constraint relaxation procedure can significantly improve the performance of this

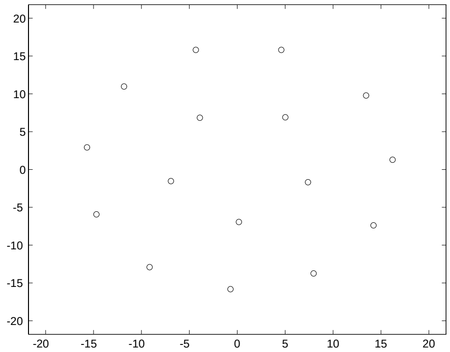


FIG. 7. *Optimal constellation for $M = 16$ with sine-cosine basis functions using a Cauchy density.*

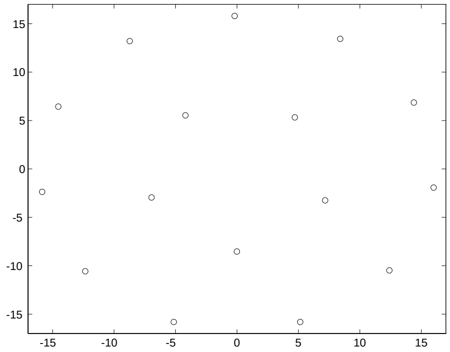


FIG. 8. *Optimal constellation for $M = 16$ with sine-cosine basis functions using a generalized Gaussian density.*

SQP method in the event that linearizations are inconsistent, which may be the case when there are far more constraints than variables.

Because there are so many different algorithms and implementations for the solution of the nonlinear programming problem, there is a need to create an accepted collection of test problems (see the paper by Bongartz et al. [7]). Because of the difficulties it poses, this family of problems is a natural candidate for such a collection.

Acknowledgments. The authors are grateful to Professor Donald Johnson and his graduate student Dong-Mei Li at Rice University for many patient discussions. The authors also thank Dr. L. C. Cowsar from Lucent Technologies. Finally, the authors greatly appreciated careful reading by an anonymous referee who suggested numerous beneficial changes to this paper.

REFERENCES

[1] A. V. BALAKRISHNAN, *A contribution to the sphere-packing problem of communication systems*, J. Math. Anal. Appl., 3 (1961), pp. 485–506.

[2] M. C. BARTHOLOMEW-BIGGS, *Opsqp and Opalqp—New Implementations of the Sequential Quadratic Programming Approach to Constrained Optimisation*, Technical Report 208, Hatfield Polytechnic, Numerical Optimisation Centre, Hatfield, Hertfordshire, England, 1989.

[3] P. T. BOGGS, P. D. DOMICH, J. E. ROGERS, AND C. WITZGALL, *An interior point method for general large scale quadratic programming problems*, Ann. Oper. Res., 62 (1996), pp. 419–437.

[4] P. T. BOGGS, A. J. KEARSLEY, AND J. W. TOLLE, *A practical algorithm for general large scale nonlinear optimization problems*, SIAM J. Optim., to appear.

[5] P. T. BOGGS, A. J. KEARSLEY, AND J. W. TOLLE, *A global convergence analysis of an algorithm for large scale nonlinear programming problems*, SIAM J. Optim., to appear.

[6] P. T. BOGGS AND J. TOLLE, *Sequential quadratic programming*, Acta Numer., (1995), pp. 1–54.

[7] I. BONGARTZ, A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Cute: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[8] C. N. CAMPOPIANO AND B. G. GLAZER, *A coherent digital amplitude and phase modulation scheme*, IRE Trans. Comm. Syst., 10 (1962), pp. 90–95.

[9] T. M. COVER AND B. GOPINATH, *Open Problems in Comminication Computation*, Springer-Verlag, New York, 1987.

[10] G. D. FORNEY, R. G. GALLAGER, G. R. LANG, F. M. LONGSTAFF, AND S. U. QURESHI, *Efficient modulation for band-limited channels*, IEEE J. Sel. Areas Comm., Sac-2 (1984), pp. 632–647.

[11] G. J. Foschini, R. D. Gitlin, and S. B. Weinstein, *Optimization of two-dimensional signal constellations in the presence of Gaussian noise*, IEEE Trans. Comm., Com-22 (1974), pp. 28–38.

[12] P. E. Gill, W. Murray, M. Saunders, and M. H. Wright, *Some Theoretical Properties of an Augmented Lagrangian Merit Function*, Technical Report 86-6, Department of Operations Research, Stanford University, Stanford, CA, 1986.

[13] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, New York, 1981.

[14] W. W. Harman, *Principles of the Statistical Theory of Communication*, McGraw-Hill, New York, 1963.

[15] D. H. Johnson and G. C. Orsak, *Performance of optimal non-Gaussian detectors*, IEE Trans. Comm., 41 (1993), pp. 1319–1328.

[16] A. J. Kearsley, *The Use of Optimization Techniques in the Solution of Partial Differential Equations from Science and Engineering*, Technical Report and Ph.D. thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1996.

[17] M. J. D. Powell, *A fast algorithm for nonlinearly constrained optimization calculations*, in Proceedings of the 1977 Dundee Biennial Conference on Numerical Analysis, Springer-Verlag, Berlin, 1977, pp. 144–157.

[18] S. Robinson, *Perturbed Kuhn–Tucker points and rates of convergence for a class of nonlinear-programming algorithms*, Math. Programming, 7 (1974), pp. 1–16.

[19] K. Tone, *Revisions of constraint approximations in the successive QP method for nonlinear programming problems*, Math. Programming, 26 (1983), pp. 144–152.

SIAM J. Optim.
Vol. 9, No. 2, pp. 327–341

ESTIMATES FOR THE NASH–SOFER PRECONDITIONER FOR THE REDUCED HESSIAN FOR SOME ELLIPTIC VARIATIONAL INEQUALITIES[*]

T. D. CHOI[†] AND C. T. KELLEY[†]

Abstract. The purpose of this paper is to present a class of examples to show how the quality of the Nash–Sofer preconditioner can be directly estimated. This class of examples includes certain discretized elliptic variational inequalities. We use sparsity and locality properties of discretizations of elliptic operators and smoothing properties of their inverses to estimate the quality of the preconditioner. One consequence of our results is that if the Hessian is the five-point discretization of a certain type of strongly elliptic operator with homogeneous Dirichlet boundary conditions on an $n \times n$ mesh and the preconditioner is a fast Poisson solver for that discretization, then the condition number of the reduced Hessian can be lowered from $O(n^2)$ to $O(n \ln(n))$. We illustrate these theoretical results with calculations.

Key words. preconditioners, reduced Hessian, elliptic variational inequalities

AMS subject classifications. 65J15, 65J10, 65K05, 65K10, 65N22

PII. S1052623497323364

1. Introduction.

In this paper we consider a class of preconditioners for constrained optimization that was recently proposed by Nash and Sofer in [26]. The motivation is the hope that a preconditioner that is good for the unconstrained problem can be projected to construct a useful preconditioner for the reduced Hessian in the constrained problem. The purpose of this paper is to present a class of examples to show how the quality of the simplest of this class of preconditioners can be directly estimated.

For these examples, the preconditioner for the unconstrained problem is a fast solver for an elliptic partial differential equation in a simple geometry, and the reduced Hessian corresponds to the same differential equation or one of the same order, but in a much more complex geometry for which fast solvers are more difficult to construct. One consequence of our results is that if the Hessian is the five-point discretization of a certain type of strongly elliptic operator with homogeneous Dirichlet boundary conditions on an $n \times n$ mesh and the preconditioner is a fast Poisson solver for that discretization, then the condition number of the reduced Hessian can be lowered from $O(n^2)$ to $O(n \ln(n))$.

Our class of problems, discretized bound constrained elliptic variational inequalities, can be solved more efficiently by multilevel methods, and we are not claiming that using the preconditioners under study in this paper will lead to optimal solvers. Our goal is not to construct optimal solvers, but rather to show that the Nash–Sofer can be analyzed for a nontrivial class of problems and that that analysis is consistent with numerical observations. Moreover, the multilevel methods discussed in the literature are more complicated to implement than, say, a multigrid Poisson solver on a regular domain, and conjugate gradient with a good preconditioner may be signifi-

[*] Received by the editors June 23, 1997; accepted for publication (in revised form) June 3, 1998; published electronically March 17, 1999. This research was supported by National Science Foundation grants DMS-9321938 and DMS-9700569 and a U.S. Department of Education GAANN fellowship.

http://www.siam.org/journals/siopt/9-2/32336.html

[†] North Carolina State University, Center for Research in Scientific Computation and Department of Mathematics, Box 8205, Raleigh, NC 27695-8205 (tdchoi@unity.ncsu.edu, Tim_Kelley@ncsu.edu).

cantly easier to program than a multilevel code for the complete problem. We provide a brief discussion of several alternative approaches to the problem in section 1.2.

1.1. Notation. In this section we specify the notation, describe the preconditioner we will study, and discuss some known convergence results for the projected Newton method. In section 3 we show how the sparsity and locality of the finite element discretization of elliptic partial differential operators and known properties of the spectrum of these operators lead to quantitative results on the effectiveness of the preconditioner. In section 4 we give numerical results based on problems from [10] and [2].

We begin with the gradient projection [4] and projected Newton [5] methods for bound constrained optimization problems of the form

$$\min_{u\in U} f(u), \tag{1.1}$$

where f is twice Lipschitz continuously differentiable and $U \subset R^N$ is given by

$$U = \{u \,|\, u_{low} \le u \le u_{high}\} \tag{1.2}$$

with the inequalities understood to be componentwise. If we let u^i be the ith component of $u \in R^N$, the l^2 projection onto U is $\mathcal{P}$, where

$$\mathcal{P}(u)^i = \begin{cases} u^i_{low} & \text{if } u^i \le u^i_{low}, \\ u^i & \text{if } u^i_{low} < u^i < u^i_{high}, \\ u^i_{high} & \text{if } u^i \ge u^i_{high}. \end{cases} \tag{1.3}$$

For sets $\mathcal{S} \subset Z^N$ we define $\mathcal{P}_\mathcal{S} u$ by

$$(\mathcal{P}_\mathcal{S} u)^i = \begin{cases} u^i, & i \in \mathcal{S}, \\ 0, & i \notin \mathcal{S}. \end{cases}$$

Throughout this paper we make the following assumption.

Assumption 1.1. Problem (1.1) has a solution $u^* \in U$ such that

1. $\nabla f(u^*)^i \ne 0$ if

$$i \in \mathcal{A}^* = \{i \,|\, (u^*)^i = u^i_{low} \text{ or } (u^*)^i = u^i_{high}\};$$

2. the reduced Hessian

$$\mathcal{R}^* = (I - \mathcal{P}_{\mathcal{A}^*})\nabla^2 f(u^*)(I - \mathcal{P}_{\mathcal{A}^*}) + \mathcal{P}_{\mathcal{A}^*}$$

is positive definite.

This paper is about local convergence and we will assume that our iterates are near u^*.

We consider iterative methods of the form

$$u_+ = \mathcal{P}(u_c - \alpha\mathcal{R}^{-1}\nabla f(u_c)). \tag{1.4}$$

In (1.4) $\mathcal{R}$ is a symmetric positive definite matrix and α is a steplength control parameter. For the gradient projection method, $\mathcal{R} = I$. For the projected Newton method, which is of interest here, $\mathcal{R}$ is the reduced Hessian and is built from the Hessian $\nabla^2 f(u_c)$ of f and an approximation of the active set at the current point

$$\mathcal{A}_c \approx \{i \,|\, u^i_c \in \{u^i_{low}, u^i_{high}\}\} \tag{1.5}$$

by

$$\mathcal{R}_c = \mathcal{P}_{\mathcal{I}_c}\nabla^2 f(u_c)\mathcal{P}_{\mathcal{I}_c} + \mathcal{P}_{\mathcal{A}_c}. \tag{1.6}$$

In (1.6), $\mathcal{I}_c$, the inactive set, is the complement of $\mathcal{A}_c$. It is known [5] that if Assumption 1.1 holds and $\mathcal{A}_c$ is carefully constructed, then the projected Newton iteration is locally q-superlinearly convergent and full steps (i.e., $\alpha = 1$) are taken in the terminal phase of the iteration. One possible choice of $\mathcal{A}_c$ is

(1.7)

$$\mathcal{A}_c = \mathcal{A}^\epsilon(u_c) = \left\{ i \,|\, u_c^i < u_{low}^i + \epsilon, \frac{\partial f(u_c)}{\partial u^i} > 0 \right\} \cup \left\{ i \,|\, u_c^i < u_{high}^i - \epsilon, \frac{\partial f(u_c)}{\partial u^i} < 0 \right\}.$$

Note that part 1 of Assumption 1.1 implies that the conditions on $\partial f / \partial u^i$ in (1.7) are redundant if u_c is sufficiently near u^*. To compute the projected Newton step the linear system

$$\mathcal{R}_c d = -\nabla f(u_c) \tag{1.8}$$

must be solved for the search direction d. If we apply $\mathcal{P}_{\mathcal{A}_c}$ and $\mathcal{P}_{\mathcal{I}_c}$ to (1.8) we obtain

$$\mathcal{P}_{\mathcal{I}_c}\nabla^2 f(u_c)\mathcal{P}_{\mathcal{I}_c} d = -\mathcal{P}_{\mathcal{I}_c}\nabla f(u_c) \text{ and } \mathcal{P}_{\mathcal{A}_c} d = -\mathcal{P}_{\mathcal{A}_c}\nabla f(u_c). \tag{1.9}$$

The idea of [26] is that if $\mathcal{M}$ is a good preconditioner, for example, a fast solver for the full Hessian or a simpler problem of the same order [19], [23], then

$$\mathcal{M}_{\mathcal{R}} = \mathcal{P}_{\mathcal{I}_c}\mathcal{M}\mathcal{P}_{\mathcal{I}_c} + \mathcal{P}_{\mathcal{A}_c} \tag{1.10}$$

can be an effective preconditioner for $\mathcal{R}$. Some numerical evidence is presented in [26] but we believe that this paper contains the first theoretical results on the application of this preconditioner to problems involving partial differential operators.

In the case of the elastoplastic torsion problem that we use as an example in section 2, $\nabla^2 f(u_c)$ is a discretization of the Laplacian operator on a simply connected region in R^2 or R^3, and fast solvers can be easily constructed using, for example, Fourier methods in simple geometries [29] or multigrid methods [12] in more complex situations. Since $\mathcal{A}$ and $\mathcal{I}$ change with the iteration and (1.9) is the discretization of an elliptic equation on a complex geometry determined by the boundary between $\mathcal{A}$ and $\mathcal{I}$, it is not clear how to implement fast solvers for the reduced problem (1.9) in an efficient way.

1.2. Alternative methods.

For quadratic problems, as most of our examples are, a related preconditioner has been proposed in [27]. This preconditioner is constructed using knowledge of a good preconditioner for $\nabla^2 f$, which is exactly the view taken in this paper. In [27] one assumes that one can find a matrix $L \approx (\nabla^2 f)^{-1}$ such that

$$L_{\mathcal{R}} = \left(P_{\mathcal{I}_c} L^{-1} P_{\mathcal{I}_c} + P_{\mathcal{A}_c}\right)^{-1}$$

can be computed easily. In that case, the condition number of $L_{\mathcal{R}}\mathcal{R}$ is no greater than that of $L\nabla^2 f$. However, $L_{\mathcal{R}}$ may be difficult to compute. For example, if $L = (\nabla^2 f)^{-1}$, then $L_{\mathcal{R}} = \mathcal{R}^{-1}$, and computing $L_{\mathcal{R}}$ would be equivalent to solving the linear system we are preconditioning. Thus, one needs a good preconditioner for

$\nabla^2 f$ as well as some insight into $\mathcal{R}$ so that $L_{\mathcal{R}}$ is easily computed. This insight is not needed in the preconditioner considered here.

The obstacle and elastoplastic torsion problems are, after being discretized, essentially box-constrained quadratic programming problems. The main methods for solving them include the successive overrelaxation method with projection (SORP) [22], multigrid methods [6], [14], [15], [16], [17], [20], methods which combine SORP or projected gradient with conjugate gradient [21], [24], [25], and methods based on the Polyak algorithm [9], [27]. SORP is an early method that is still widely used because of its simplicity and robustness. In [6], [14], [15] the multigrid methods are shown in numerical tests to be superior to SORP, but in [20] Kornhuber states that they still either lack robustness or have poor convergence rates. Convergence of these multigrid methods is proven, but there are no results on the optimality of the methods.

In [17], Hoppe and Kornhuber propose a multilevel method that uses an active set strategy to reduce the problem to a series of subproblems which are solved using preconditioned (by additive Schwarz [28]) conjugate gradient. They prove convergence of the algorithm and estimates on the condition number of the subproblem. The theory and numerical results show that the number of conjugate gradient iterations is linear with respect to j (the refinement level) when j is large enough. In [20] Kornhuber presents a multigrid method which in numerical tests has the same efficiency as multigrid in the unconstrained case. He also proves in addition to convergence of the method that the asymptotic convergence rates are bounded by $1-O(j^{-3})$, where the minimal diameter of the discretization triangles is of $O(2^{-j})$.

The nonmultigrid methods are not as fast as the multilevel methods; however, they are often simpler to implement and do not require additional data for auxiliary problems that most multigrid methods require [21]. Numerical experiments show that these methods are faster than SORP. The convergence of the methods is proven, but no results on the optimality of the methods are given. Two of the methods build on the algorithm that we essentially use in this paper, which is the preconditioned conjugate gradient method combined with an active set strategy (PCGA). One major problem with PCGA is that the method is effective only when the initial active set is sufficiently "good." In [25] Moré and Toraldo modify this method by combining a projected gradient step with the step from PCGA. That is, at each step they take one or more projected gradient steps and then they take the step from PCGA. In [21] Kočvara and Zowe use a SORP step in place of the projected gradient step. The idea behind these methods is that the projected gradient or SORP step will help to quickly locate the active set at the solution. Finding the active set at the solution is the limiting factor in the efficiency of PCGA.

2. Example: Elastoplastic torsion problem.

As an example, we consider the elastoplastic torsion problem on the unit square in two dimensions [1], [2], [10]. We will consider one other example from these sources in section 4. Let

$$\Omega = [0,1] \times [0,1] \subset R^2.$$

We use the formulation

$$\min_{u \in U} f(u), \tag{2.1}$$

where

$$f(u) = \frac{1}{2}\int_{\Omega} \|D_x u(x)\|^2 \, dx - c\int_{\Omega} u(x) \, dx \tag{2.2}$$

and

$$U = \{u \in H_0^1(\Omega) \,|\, |u(x)| \leq \mathrm{dist}(x, \partial\Omega) \text{ a.e. in } \Omega\}. \tag{2.3}$$

In (2.2) D_x denotes the gradient with respect to x. The purpose of this notation is to distinguish D_x from ∇f, which will denote the derivative of f with respect to u. Similarly, D_x^2 will denote the Laplacian in x and $\nabla^2 f$ the Hessian of f.

For this infinite-dimensional problem, the independent variable x plays the role of the index i and, in terms of the formulation in section 1,

$$u_{high}(x) = \mathrm{dist}(x, \partial\Omega) \quad \text{and} \quad u_{low}(x) = -\mathrm{dist}(x, \partial\Omega).$$

With respect to the L^2 inner product, the gradient and Hessian of f are

$$\nabla f(u) = -D_x^2 u - c \quad \text{and} \quad \nabla^2 f(u) = -D_x^2. \tag{2.4}$$

With this choice of inner product, the projection onto U is the obvious analog of (1.3):

$$\mathcal{P}u(x) = \begin{cases} u_{high}(x), & u(x) \geq u_{high}(x), \\ u(x), & u_{low}(x) < u(x) < u_{high}(x), \\ u_{low}(x), & u(x) \geq u_{low}(x). \end{cases} \tag{2.5}$$

This choice of scalar product is the one inherited by the discretization in [2], which we use here.

Ideally, one would use the inner product in H_0^1, the space where the problem is naturally posed [10]. If we do this, then

$$\nabla f(u) = u + e \quad \text{and} \quad \nabla^2 f(u) = I, \tag{2.6}$$

where e is the weak solution in H_0^1 of $-D_x^2 e = -c$. Therefore, if the H_0^1 inner product is used, the projected Newton and gradient projection methods are identical. The catch is that the projection onto the feasible set is not given by a simple and easy-to-compute expression like (2.5). In fact, if w is the projection of u onto U relative to the H_0^1 norm, then by definition

$$\|w - u\|_{H_0^1}^2 \leq \|v - u\|_{H_0^1}^2 \tag{2.7}$$

for all $v \in U$. This is also an elliptic variational inequality of the first kind and is no easier to solve than the original problem.

2.1. Discretization of elastoplastic torsion problem. Discretization of a two-dimensional problem is most simply presented if the discrete variable is doubly indexed. However, the general theory that we develop in section 3 is more clear with single indexing. To make the distinction more clear, we double index in the next few paragraphs with uppercase letters and then single index with lowercase letters.

We use a uniform mesh of width $h = 1/(n+1)$ in the horizontal and vertical directions. Hence, the unknown vector $\{u^{I,J}\}_{I,J=1}^n$ will have size $N = n^2$. The nodal points for the discretization are

$$x^{I,J} = (Ih, Jh), \quad 0 \leq I, \quad J \leq n+1,$$

with values of I, J of 0 or $n+1$ denoting boundary nodes.

We discretize (2.1) with piecewise linear finite elements to obtain [10], [2]

$$\min_{u \in U_h} f_h(u), \tag{2.8}$$

where

$$f_h(u) = \frac{1}{2}\sum\nolimits_{I,J}\left(q_L^{I,J}(u) + q_U^{I,J}(u)\right) - c\sum\nolimits_{I,J} u^{I,J}, \tag{2.9}$$

$$\begin{aligned} q_L^{I,J}(u) &= \frac{1}{2}\left[\left(\frac{u^{I+1,J} - u^{I,J}}{h}\right)^2 + \left(\frac{u^{I,J+1} - u^{I,J}}{h}\right)^2\right] \\ &\text{and} \\ q_U^{I,J}(u) &= \frac{1}{2}\left[\left(\frac{u^{I-1,J} - u^{I,J}}{h}\right)^2 + \left(\frac{u^{I,J-1} - u^{I,J}}{h}\right)^2\right], \end{aligned} \tag{2.10}$$

and

$$U_h = \{u \in R^{n\times n} \,|\, |u^{I,J}| \le d^{I,J} = \operatorname{dist}(x^{I,J}, \partial\Omega)\}. \tag{2.11}$$

The gradient of f is

$$\nabla f_h(u) = -D_h^2 u - \mathbf{c},$$

where $\mathbf{c}$ is the vector having all components equal to c and D_h^2 is the discrete Laplacian. We describe the Hessian of f, the negative discrete Laplacian with homogeneous Dirichlet boundary data, in terms of its action on a vector:

$$(-D_h^2 f_h(u) w)^{I,J} = \frac{-w^{I-1,J} - w^{I,J-1} - w^{I,J+1} - w^{I+1,J} + 4w^{I,J}}{h^2}.$$

The discrete active set at u $\mathcal{A}(u)$ will be constructed to approximate

$$\{I, J \,|\, |u^{I,J}| = d^{I,J}\} \subset Z^{n\times n}$$

and let $\mathcal{I}(u)$ be the complement of $\mathcal{A}(u)$. Note that for the discrete problem the active set is expressed in terms of indices, integer pairs, whereas for the continuous problem the active set would be a subset of Ω. For the discrete problems under study here, our notion of boundary for $\mathcal{A}$ and $\mathcal{I}$ must take into account the locality of the operator $D_x^2 f$.

Definition 2.1. *The indices (I, J) and (K, L) are* adjacent *if the corresponding nodes $x^{I,J}$, $x^{K,L}$ satisfy*

$$\|x^{I,J} - x^{K,L}\| \le h.$$

(I, J) is adjacent to a set $\mathcal{S} \subset Z^{n\times n}$ if (I, J) is adjacent to any point in $\mathcal{S}$. If $\mathcal{S} \subset Z^{n\times n}$ and $\mathcal{S}'$ is the complement of $\mathcal{S}$, then

$$\partial\mathcal{S} = \{(I, J) \in \mathcal{S} \,|\, (I, J) \text{ adjacent to } \mathcal{S}'\}.$$

Note that the indices adjacent to (I, J) are exactly those that play a role in the five-point approximation to the Laplacian of u at $x^{I,J}$.

We now order the nodes with a single index. For the elastoplastic torsion problem, f_h is quadratic, and the Hessian is the negative discrete Laplacian, $-D_h^2$. It is well known that $-D_h^2$ is positive definite [10], [18]. It is also known (see [13] with $C = 1/\log(4)$), and easily verifiable using the diagonalization of the discrete Laplacian, that $(D_h^2)^{-1}$ satisfies the componentwise estimate

$$((D_h^2)^{-1})^{i,j} \le Ch^2 \log(1/h) \tag{2.12}$$

for some $C > 0$ which is independent of h.

3. Convergence results.

The general assumptions made in this section are motivated by the results for the elastoplastic torsion problem that we stated in section 2 and the examples from section 4. Our assumptions on the preconditioner $\mathcal{M}$ are based on known properties of a Poisson solver such as (2.12). We consider the problem given by (1.1) and (1.2).

3.1. Assumptions and notation.

We assume that the indices can be grouped in such a way that notions of adjacency and interior make sense. This means that our problem is based on a physical grid and hence that the mesh width plays a role in the assumption. The underlying assumption in this section is that a sequence of problems is under study; hence the bounds in the assumptions are stated in terms of the problem size N.

Assumption 3.1. There is a metric δ on Z^N and $M, p > 0$ such that

- i and j are adjacent if $\delta(i,j) \le N^{-p}$ and
- the number of points adjacent to i is at most M.

We can now define the boundary and interior of a set and the support of a vector.

DEFINITION 3.1. *Let $\mathcal{S} \in Z^N$.*

- *$\mathcal{S}'$ is the complement of $\mathcal{S}$.*
- *The boundary $\partial\mathcal{S}$ of $\mathcal{S}$ is*

$$\partial\mathcal{S} = \{i \in \mathcal{S} \,|\, i \text{ is adjacent to } \mathcal{S}'\}.$$

- *The interior $\mathcal{S}^o$ of $\mathcal{S}$ is*

$$\mathcal{S}^o = \{i \in \mathcal{S} \,|\, i \notin \partial\mathcal{S}\}.$$

- *The support of $u \in R^N$ is*

$$\text{supp}(u) = \{i \,|\, u^i \ne 0\}.$$

Our assumptions on $\mathcal{M}$ and $\nabla^2 f$ are as follows.

Assumption 3.2.

- Locality: For all $u \in R^N$, $\mathcal{S} \subset Z^N$, and $w \in R^N$ with $\text{supp}(w) \subset \mathcal{S}^o$,

$$\text{supp}(\mathcal{M}^{-1}w), \text{supp}(\nabla^2 f(u)w) \subset \mathcal{S}.$$

- Sparsity: There are at most M nonzeros in each row and column of $\mathcal{M}^{-1}$ and $\nabla^2 f(u)$. Moreover, there are $C_1, d > 0$ such that

$$|(\mathcal{M}^{-1})^{ij}|, |(\nabla^2 f(u))^{ij}| \le C_1 N^{2/d}.$$

- Smoothing: There is a function $\phi \ge 0$ such that $\lim_{N\to\infty} \phi(N) = 0$ and

$$|\mathcal{M}^{ij}| \le \phi(N)$$

for all $1 \le i, j \le N$.

As an example of Assumptions 3.1 and 3.2 we point to the two-dimensional elastoplastic torsion problem with

$$\nabla^2 f = -D_h^2 \quad \text{and} \quad \mathcal{M} = (-D_h^2)^{-1} = (\nabla^2 f)^{-1}.$$

Here we see that the exponent in Assumption 3.1 $p = 1/2$. We may set $M = C_1 = 5$, and if i and j correspond to the two-dimensional mesh points (I, J) and (K, L),

$$\delta(i,j) = \frac{n+1}{n} \| x^{I,J} - x^{K,L} \|_2,$$

since $h = 1/(n+1)$ and $N = n^2$. The five-point Laplacian has $d = 2$ and $M = 5$. The estimate (2.12) implies that, for some $C_0 > 0$,

$$|\mathcal{M}^{ij}| = |((D_h^2)^{-1})^{ij}| = O(h^2 \log(1/h)) = O(N^{-1} \log(N)) \leq C_0 N^{-1} \log(N).$$

Hence we may use $\phi(N) = C_0 N^{-1} \log(N)$.

Local convergence of any Newton-like method requires regularity and nonsingularity assumptions. In the present context, we will also require regularity of $\partial\mathcal{A}$ in a certain sense.

Assumption 3.3. Let d be the exponent from Assumption 3.2. There are an open neighborhood $\mathcal{N}$ of u^* and $C_2, \epsilon_0 > 0$ so that for all $\epsilon < \epsilon_0$, $u \in \mathcal{N}$, $\mathcal{A} = \mathcal{A}^\epsilon(u)$, and $\mathcal{I} = \mathcal{A}'$,

- $\mathcal{R}(u)$ is positive definite;
- $\mathcal{M}$ is symmetric and positive definite and there is $C_3 \geq 0$ such that

$$\|\mathcal{M}\nabla^2 f(u)\| \leq C_3 N^{1/d};$$

- the cardinality $|\partial\mathcal{I}|$ of $\partial\mathcal{I}$ satisfies

$$|\partial\mathcal{I}| < C_2 N^{(d-1)/d}.$$

The third part of Assumption 3.3 states that the interface between the inactive and active sets is, in a sense, of lower dimension $(d-1)$ than Ω, which has dimension d. This is a natural assumption for the discretized elliptic variational inequalities under consideration in this paper if $\partial\mathcal{I}$ is a family of curves in the two-dimensional set $\overline{\Omega}$.

3.2. Main results. In order to estimate the effectiveness of $\mathcal{M}$ we need lower and upper bounds for the spectrum of

$$\mathcal{M}_{\mathcal{R}}^{1/2} \mathcal{R}(u) \mathcal{M}_{\mathcal{R}}^{1/2} = (\mathcal{P}_{\mathcal{I}} \mathcal{M} \mathcal{P}_{\mathcal{I}})^{1/2} \nabla^2 f(u) (\mathcal{P}_{\mathcal{I}} \mathcal{M} \mathcal{P}_{\mathcal{I}})^{1/2} + \mathcal{P}_{\mathcal{A}}.$$

We obtain these bounds by estimating the spectrum of the similar matrix

$$E(u) = \mathcal{P}_{\mathcal{I}} \mathcal{M} \mathcal{P}_{\mathcal{I}} \nabla^2 f(u) \mathcal{P}_{\mathcal{I}} + \mathcal{P}_{\mathcal{A}}. \tag{3.1}$$

THEOREM 3.2. *Let Assumptions* 3.1*,* 3.2*, and* 3.3 *hold. Let $u \in \mathcal{N}$. Then there is $C > 0$, independent of N, such that*

$$0 < \lambda \leq C(N^{d^{-1}} + \phi(N) N^{d^{-1}+1}) \tag{3.2}$$

for all eigenvalues λ of $E(u)$.

Proof. It clearly suffices to prove that (3.2) holds for $\lambda \neq 1$. Let $E(u)w = \lambda w$ with $\lambda \neq 1$. Then

$$\lambda \mathcal{P}_{\mathcal{A}} w = \mathcal{P}_{\mathcal{A}} E(u) w = \mathcal{P}_{\mathcal{A}} w$$

and hence $\mathcal{P}_{\mathcal{A}} w = 0$.

Now write

$$w = \mathcal{P}_{\mathcal{I}^o} w + \mathcal{P}_{\partial \mathcal{I}} w.$$

By Assumption 3.2 we see that

$$\nabla^2 f(u) \mathcal{P}_{\mathcal{I}^o} w = \mathcal{P}_{\mathcal{I}} \nabla^2 f(u) \mathcal{P}_{\mathcal{I}^o} w$$

and hence

$$\mathcal{P}_{\mathcal{I}} \mathcal{M} \mathcal{P}_{\mathcal{I}} \nabla^2 f(u) \mathcal{P}_{\mathcal{I}^o} w = \mathcal{P}_{\mathcal{I}} \mathcal{M} \nabla^2 f(u) \mathcal{P}_{\mathcal{I}^o} w.$$

Hence,

$$\|E(u) \mathcal{P}_{\mathcal{I}^o} w\| \leq \|\mathcal{M} \nabla^2 f(u)\| \|\mathcal{P}_{\mathcal{I}^o} w\| \leq C_3 N^{1/d} \|\mathcal{P}_{\mathcal{I}^o} w\|. \tag{3.3}$$

Our sparsity assumption implies that there is a set $\mathcal{L}$ with $|\mathcal{L}| \leq M|\partial \mathcal{I}|$ such that for all $i \in \operatorname{supp}(\mathcal{P}_{\partial \mathcal{I}} w) \subset \partial \mathcal{I}$,

$$(E(u) \mathcal{P}_{\partial \mathcal{I}} w)^i = \sum_{j \in \mathcal{L}} \mathcal{M}^{ij} \sum_{k \in \partial \mathcal{I}} (\nabla^2 f(u))^{jk} (\mathcal{P}_{\partial \mathcal{I}} w)^k.$$

Hence,

$$\|E(u) \mathcal{P}_{\partial \mathcal{I}} w\|_\infty \leq M^2 C_1 |\partial \mathcal{I}| \phi(N) N^{2/d} \|\mathcal{P}_{\partial \mathcal{I}} w\|_\infty.$$

And so, since

$$\|\mathcal{P}_S\|_\infty = 1$$

for all $S \subset Z^N$,

$$\begin{aligned} |\lambda| \|w\|_\infty = \|E(u) w\|_\infty &\leq \|E(u) \mathcal{P}_{\mathcal{I}^o} w\|_\infty + \|E(u) \mathcal{P}_{\partial \mathcal{I}} w\|_\infty \\ &\leq C_3 N^{1/d} \|\mathcal{P}_{\mathcal{I}^o} w\|_\infty + M^2 C_1 |\partial \mathcal{I}| \phi(N) N^{2/d} \|\mathcal{P}_{\partial \mathcal{I}} w\|_\infty \\ &\leq (C_3 N^{1/d} + M^2 C_1 C_2 \phi(N) N^{2/d+1-1/d}) \|w\|_\infty. \end{aligned}$$

This completes the proof with $C = C_3 + M^2 C_1 C_2$. □

Our remaining task is to prove a lower bound on λ. We begin with a basic lemma.

LEMMA 3.3. *Let $\mathcal{M}$ be symmetric and positive definite. Let $\mathcal{A} \subset Z^N$ and $\mathcal{I}$ be the complement of $\mathcal{A}$. Then*

$$\sigma(\mathcal{P}_{\mathcal{I}} \mathcal{M} \mathcal{P}_{\mathcal{A}} \mathcal{M}^{-1} \mathcal{P}_{\mathcal{I}}) \subset (-\infty, 0).$$

Proof. Let $\mu \neq 0$ be an eigenvalue of $\mathcal{P}_{\mathcal{I}}\mathcal{M}\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}\mathcal{P}_{\mathcal{I}}$ and $v \neq 0$ a corresponding eigenvector. So

$$\mathcal{P}_{\mathcal{I}}\mathcal{M}\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}\mathcal{P}_{\mathcal{I}}v = \mu v.$$

Hence

$$\mathcal{M}\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}\mathcal{P}_{\mathcal{I}}v = \mu v + y,$$

where $\mathrm{supp}(y) \subset \mathcal{A}$.

Now, using the symmetry and positivity of $\mathcal{M}$ and the fact that $v^T y = 0$, we have

$$\begin{aligned} y^T\mathcal{M}^{-1}v &= y^T\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}v = (y+\mu v)^T\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}v \\ &= (\mathcal{M}\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}v)^T\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}v \\ &= (\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}v)^T\mathcal{M}(\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}v) > 0. \end{aligned} \tag{3.4}$$

We complete the proof by noting that

$$\mathcal{M}\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}v = \mu v + y$$

implies

$$\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}v = \mu\mathcal{M}^{-1}v + \mathcal{M}^{-1}y$$

and so, since $\mathcal{P}_{\mathcal{A}}v = 0$,

$$0 = v^T\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}v = \mu v^T\mathcal{M}^{-1}v + v^T\mathcal{M}^{-1}y.$$

Hence, $\mu v^T\mathcal{M}^{-1}v = -v^T\mathcal{M}^{-1}y < 0$ by (3.4), which proves the assertion because $v^T\mathcal{M}^{-1}v > 0$. □

In order to complete our lower estimate we require one more assumption, which is trivially satisfied in the examples in section 4.

Assumption 3.4. There is $\delta_- \in (0,1)$, independent of N, such that $\nabla^2 f(u) - \delta_-\mathcal{M}^{-1}$ is symmetric and positive definite.

THEOREM 3.4. *Let Assumptions* 3.1*,* 3.2*,* 3.3*, and* 3.4 *hold. Let* $u \in \mathcal{N}$*. Then if* λ *is an eigenvalue of* $\mathcal{M}_{\mathcal{R}}^{1/2}\mathcal{R}(u)\mathcal{M}_{\mathcal{R}}^{1/2}$*,*

$$\lambda \geq \delta_-.$$

Proof. Note that our assumptions imply that if

$$B(u) = (\mathcal{P}_{\mathcal{I}}\mathcal{M}\mathcal{P}_{\mathcal{I}})^{1/2}\mathcal{M}^{-1}(\mathcal{P}_{\mathcal{I}}\mathcal{M}\mathcal{P}_{\mathcal{I}})^{1/2} + \mathcal{P}_{\mathcal{A}},$$

then

$$\mathcal{M}_{\mathcal{R}}^{1/2}\mathcal{R}(u)\mathcal{M}_{\mathcal{R}}^{1/2} - \delta_- B(u)$$

is symmetric and positive definite. Hence if ν is the smallest eigenvalue of $B(u)$, then $\lambda \geq \delta_-\nu$.

As in the proof of Theorem 3.2 we consider the matrix

$$C(u) = \mathcal{P}_{\mathcal{A}} + \mathcal{P}_{\mathcal{I}}\mathcal{M}\mathcal{P}_{\mathcal{I}}\mathcal{M}^{-1}\mathcal{P}_{\mathcal{I}},$$

which is similar to $B(u)$. Since

$$C(u) = I - \mathcal{P}_{\mathcal{I}}\mathcal{M}\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}\mathcal{P}_{\mathcal{I}}$$

and all eigenvalues of $\mathcal{P}_{\mathcal{I}}\mathcal{M}\mathcal{P}_{\mathcal{A}}\mathcal{M}^{-1}\mathcal{P}_{\mathcal{I}}$ are negative by Lemma 3.3, we must have $\nu \geq 1$. □

COROLLARY 3.5. *Let Assumptions* 3.1, 3.2, 3.3, *and* 3.4 *hold. Let $u \in \mathcal{N}$. Then*

$$\kappa(E(u)) \leq C\delta_-^{-1}\left(N^{d^{-1}} + \phi(N)N^{d^{-1}+1}\right).$$

Returning to elliptic variational inequalities in two space dimensions, the condition number of $E(u)$, by the theorem above, is, using $N = n \times n \approx h^{-2}$, $d = 2$, $\nabla^2 f(u) = \mathcal{M}^{-1}$ (so any $0 < \delta_- < 1$ will satisfy Assumption 3.4), and $\phi(N) = N^{-1}\ln(N)$,

$$O(N^{d^{-1}}\ln(N)) = O(N^{1/2}\ln(N)),$$

which is substantially better than the $O(h^{-2}) = O(N)$ condition number [3] of $\mathcal{R}$ itself.

4. Numerical results.

4.1. Problems. In addition to the elastoplastic torsion problem, we report computations on the journal bearing problem [10], [2]. As was the case with the elastoplastic torsion problem, the journal bearing problem has the form

$$\min_{u\in U} f(u), \tag{4.1}$$

where

$$f(u) = \frac{1}{2}\int_\Omega w_q(x)(D_x u(x))^T(D_x u(x))dx - \int_\Omega w_l(x)u(x)dx. \tag{4.2}$$

In (4.2) $\Omega = (0, 2\pi) \times (0, 2b)$ with $b > 0$, $\epsilon \in (0, 1)$,

$$w_q(x_1, x_2) = (1 + \epsilon\cos x_1)^3 \quad \text{and} \quad w_l(x_1, x_2) = \epsilon\sin x_1.$$

The feasible set is

$$U = \{u \in H_0^1(\Omega) \,|\, u \geq 0 \text{ a.e.}\}. \tag{4.3}$$

The finite element discretization is described in [2]. The fact that $w_q > 0$ implies that if $\mathcal{M}$ is the inverse of the discrete Laplacian, then Assumption 3.4 holds with $\delta_- = \inf w_q$.

We also report computations on the minimal surface problem [2]. It has the following form:

$$\min_{u\in U} f(u), \tag{4.4}$$

where

$$f(u) = \int_\Omega \left(1 + (D_x u(x))^T(D_x u(x))\right)^{1/2} dx. \tag{4.5}$$

In (4.5) $\Omega = (-\frac{1}{2}, \frac{1}{2}) \times (-\frac{1}{2}, \frac{1}{2})$. The feasible set is

$$(4.6) \quad U = \left\{ u \in H^1(\Omega) : v(x) = v_\Omega(x) \text{ for } x \in \partial\Omega, v(x) \geq 3 \operatorname{dist}(x, \partial\Omega) - \frac{1}{2} \right\},$$

where

$$v_\Omega(x_1, x_2) = u^2 - v^2$$

and u, v are the unique solutions to the equations

$$x_1 = u + uv^2 - \frac{1}{3}u^3,$$

$$x_2 = -v - u^2 v + \frac{1}{3}v^3.$$

The finite element discretization is described in [2]. The fact that the Gateaux derivative $f'(u)$ exists and

$$(f'(u), w) = \int_\Omega \left(1 + (D_x u(x))^T (D_x u(x))\right)^{-1/2} (D_x u(x))^T (D_x w(x)) \, dx,$$

where $(\cdot, \cdot)$ denotes the L^2 inner product and $w \in H^1$, implies that if $\mathcal{M}$ is the inverse of the discrete Laplacian, then Assumption 3.4 holds with

$$\delta_- = \inf \left(1 + (D_x u(x))^T (D_x u(x))\right)^{-1/2}.$$

4.2. Predictions of the theory. We will apply the theory developed in the previous sections and the well-known estimate for convergence of conjugate gradient iterations for a positive definite A [7], [11],

$$(4.7) \qquad \|x - x_k\|_A \leq \|x - x_0\|_A \left(\frac{1 - \kappa(A)^{1/2}}{1 + \kappa(A)^{1/2}} \right)^{2k},$$

to explain the numerical observations in this section. If, as in the case of the elliptic operators considered in this paper, there is no special clustering of the spectrum, the number of iterations needed to reduce the A-norm of the error by a factor of η is well estimated by

$$\frac{\eta}{2(\ln(1 + \kappa(A)^{1/2}) - \ln(1 - \kappa(A)^{1/2}))} = O\left(\frac{\eta}{\kappa(A)^{1/2}} \right).$$

In computations, only residual norms are observed, not errors, but the estimate of convergence rates in (4.7) should also reflect the reduction in residuals.

In the context of the two-dimensional elliptic variational inequalities considered here,

$$\kappa(\mathcal{R}) = O(h^{-2}) = O(N)$$

for the unpreconditioned problem and

$$\kappa(\mathcal{M}_\mathcal{R}^{1/2} \mathcal{R}(u) \mathcal{M}_\mathcal{R}^{1/2}) = O(h^{-1} \ln(1/h)) = O(N^{1/2} \ln(N)).$$

In the reports on numerical results in this section we tabulate linear iteration counts as a function of the mesh size h. We expect the count to double as $h \to h/2$ for the unpreconditioned problem and, neglecting the factor of $\ln(N)$, to increase by a factor of $\sqrt{2}$ for the preconditioned problem.

4.3. Observations. For all of the problems, we compute solutions of the discrete problems having $N = n^2$ unknowns with mesh sizes of

$$h = \frac{1}{n+1} = 2^{-k}, \qquad k = 4, \ldots, 8.$$

We terminate the linear iteration when the relative residual is small, i.e.,

$$\|\mathcal{R}d + \nabla f(u)\| \leq \eta \|\nabla f(u)\| \tag{4.8}$$

for some small η. We use (4.8) rather than the standard inexact Newton criterion [8]

$$\|\mathcal{R}d + \nabla f(u)\| \leq \eta \|u - \mathcal{P}(u - \nabla f(u))\|$$

in order to clearly see the effects of the preconditioner. We set $\eta = 10^{-5}$ for all h. The reason for making the termination criterion for the linear iteration independent of the mesh is to be able to compare the increased cost of the linear iteration as the mesh is refined to the prediction in section 4.2 in a direct way. The nonlinear iteration was terminated when

$$\|u - \mathcal{P}(u - \nabla f(u))\| \leq h^2.$$

Since this is dependent on h, we tabulate the average number of linear iterations per nonlinear iteration in the tables.

$\mathcal{M}$ was the fast Poisson solver from [29] and the functions, gradients, and Hessians were computed using the MINPACK-2 codes [2]. The computations were done on an IBM RS6000 Model 250 with the AIX XL Fortran Compiler under AIX operating system version 3.2.

One might suspect that the size of the active set plays a role in the performance in that for a small active set the preconditioner should perform more like that for an unconstrained problem. To examine this we varied the value of the parameter c for the elastoplastic torsion problem. The tables indicate that unless the active set is empty and the preconditioner is the inverse ($c = 3$ and some cases with coarse meshes), the performance of the preconditioner is as the theory predicts.

In Tables 4.1 and 4.2 we tabulate the average number of linear iterations per nonlinear iteration for the three problems, the elastoplastic torsion problem (EPT), the journal bearing problem (JBP), and the minimal surface problem (MIN), and several mesh sizes. We also tabulate the ratios of these values from one mesh size to the next. For the preconditioned results of the EPT, four different values of the constant c are used ($c = 5, 4, 3.5, 3$). The percentage of indices that are active at the solution is approximately 32% for $c = 5$, 17% for $c = 4$, 9% for $c = 3.5$, and 0% for $c = 3$. In Table 4.2 the value of c is given in parentheses. For the numerical results of the MIN we wish to minimize the nonlinear effects of the problem so that we can clearly see the results of the theory. We do so by choosing a very good starting point at each mesh, namely, the solution of the minimal surface problem on the $h = 1/16$ mesh. The predicted ratios for all three problems, 2 for the unpreconditioned case and $\sqrt{2}$ for the preconditioned, are well approximated by the numerical results. The exception, of course, is EPT(3), where the preconditioner is the inverse of the Hessian and only one iteration is needed.

TABLE 4.1
Unpreconditioned iterations.

h	1/16	1/32	1/64	1/128	1/256
EPT(5)	19.0	36.2	68.6	129.7	247.3
		1.9	1.9	1.9	1.9
JBP	29.3	56.3	112.4	214.4	403.4
		1.9	2.0	1.9	1.9
MIN	—	82	178	346	681.3
			2.2	1.9	2.0

TABLE 4.2
Preconditioned iterations.

h	1/16	1/32	1/64	1/128	1/256
EPT(5)	3.7	5.6	8.4	13.2	19.1
		1.5	1.5	1.6	1.5
EPT(4)	1	4.3	6.6	9.7	15.4
		4.3	1.5	1.5	1.6
EPT(3.5)	1	1	4	7.3	11.6
		1	4	1.8	1.6
EPT(3)	1	1	1	1	1
		1	1	1	1
JBP	14.7	17.3	18.9	22.5	28.7
		1.2	1.1	1.2	1.3
MIN	—	32	44.5	57.5	69.8
			1.4	1.3	1.2

Acknowledgments. The authors are most grateful to Loyce Adams for her wise counsel on the smoothing assumptions.

REFERENCES

[1] B. M. Averick, R. G. Carter, and J. J. Moré, *The MINPACK-2 Test Problem Collection (Preliminary Version)*, Tech. Rep. ANL/MCS-TM-150, Math. and Comp. Science Div. Report, Argonne National Laboratory, Argonne, IL, May 1991.

[2] B. M. Averick and J. J. Moré, *User Guide for the MINPACK-2 Test Problem Collection*, Tech. Rep. ANL/MCS-TM-157, Math. and Comp. Science Div. Report, Argonne National Laboratory, Argonne, IL, October 1991.

[3] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK, 1994.

[4] D. P. Bertsekas, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Trans. Automat. Control, 21 (1976), pp. 174–184.

[5] D. P. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.

[6] A. Brandt and C. Cryer, *Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems*, SIAM J. Sci. Stat., 4 (1983), pp. 655–684.

[7] P. Concus, G. H. Golub, and D. P. O'Leary, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, in Sparse Matrix Computations, J. R. Bunch and D. J. Rose, eds., Academic Press, New York, 1976, pp. 309–332.

[8] R. Dembo, S. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[9] Z. Dostál, *Box constrained quadratic programming with proportioning and projections*, SIAM J. Optim., 7 (1997), pp. 871–887.

[10] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, 1984.

[11] G. H. Golub and C. G. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1983.

[12] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer Series in Computational Mathematics 4, Springer-Verlag, New York, 1985.

[13] W. Hackbusch, *Elliptic Differential Equations: Theory and Numerical Treatment*, Springer-Verlag, New York, 1992.

[14] W. Hackbusch and H. Mittelmann, *On multigrid methods for variational inequalities*, Numer. Math., 42 (1983), pp. 65–76.

[15] R. Hoppe, *Multigrid algorithms for variational inequalities*, SIAM J. Numer. Anal., 24 (1987), pp. 1046–1065.

[16] R. Hoppe, *Multigrid solutions to the elastic plastic torsion problem in multiply connected domains*, Internat. J. Numer. Meth. Engrg., 26 (1988), pp. 631–646.

[17] R. Hoppe and R. Kornhuber, *Adaptive multilevel methods for obstacle problems*, SIAM J. Numer. Anal., 31 (1994), pp. 301–323.

[18] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, Cambridge, UK, 1987.

[19] W. Joubert, T. A. Manteuffel, S. Parter, and S.-P. Wong, *Preconditioning second-order elliptic operators: Experiment and theory*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 259–288.

[20] R. Kornhuber, *Monotone multigrid methods for elliptic variational inequalities* I, Numer. Math., 69 (1994), pp. 167–184.

[21] M. Kočvara and J. Zowe, *An iterative two-step algorithm for linear complementarity problems*, Numer. Math., 68 (1994), pp. 95–106.

[22] O. Mangasarian, *Solution of symmetric linear complementarity problems by iterative methods*, J. Optim. Theory Appl., 22 (1977), pp. 4656–485.

[23] T. A. Manteuffel and S. Parter, *Preconditioning and boundary conditions*, SIAM J. Numer. Anal., 27 (1990), pp. 656–694.

[24] H. Mittelmann, *On the efficient solution of nonlinear finite element equations* II, Numer. Math., 36 (1981), pp. 375–387.

[25] J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.

[26] S. G. Nash and A. Sofer, *Preconditioning reduced matrices*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 47–68.

[27] D. P. O'Leary, *A generalized conjugate gradient algorithm for solving a class of quadratic programming problems*, Linear Algebra Appl., 34 (1980), pp. 371–399.

[28] B. Smith, P. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1996.

[29] P. N. Swarztrauber and R. A. Sweet, *Algorithm* 541: *Efficient FORTRAN subprograms for the solution of elliptic partial differential equations*, ACM Trans. Math. Software, 5 (1979), pp. 352–364.

SIAM J. OPTIM.
Vol. 9, No. 2, pp. 342–373

JACOBIAN SMOOTHING METHODS FOR NONLINEAR COMPLEMENTARITY PROBLEMS*

CHRISTIAN KANZOW† AND HEIKO PIEPER‡

Abstract. We present a new algorithm for the solution of general (not necessarily monotone) complementarity problems. The algorithm is based on a reformulation of the complementarity problem as a nonsmooth system of equations by using the Fischer–Burmeister function. We use an idea by Chen, Qi, and Sun and apply a Jacobian smoothing method (which combines nonsmooth Newton and smoothing methods) to solve this system. In contrast to that of Chen, Qi, and Sun, however, our method is at least well defined for general complementarity problems. Extensive numerical results indicate that the new algorithm works very well. In particular, it can solve all nonlinear complementarity problems from the MCPLIB and GAMSLIB libraries.

Key words. nonlinear complementarity problem, nonsmooth Newton method, smoothing method, global convergence, quadratic convergence

AMS subject classification. 90C33

PII. S1052623497328781

1. Introduction.

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be continuously differentiable. The *nonlinear complementarity problem* is to find a solution of the following system of equations and inequalities:

$$x_i \geq 0, \;\; F_i(x) \geq 0, \;\; x_i F_i(x) = 0 \quad \forall i \in I := \{1, \dots, n\}.$$

We denote this problem by NCP(F). It has a large number of important applications, and we refer the interested reader to the survey papers by Harker and Pang [22] and Ferris and Pang [17].

The basic idea of most algorithms for the solution of NCP(F) is to reformulate this problem as a nonlinear system of equations, as an optimization problem, or as a parametric problem. Here we concentrate on the equation-based approach, where problem NCP(F) is written equivalently as

$$\Phi(x) = 0 \tag{1.1}$$

for a suitable equation operator $\Phi : \mathbb{R}^n \to \mathbb{R}^n$. For certain reasons, the operator Φ is usually nonsmooth, so that we cannot apply the classical Newton method in order to solve the problem (1.1). Nevertheless, recent research shows that one can still design globally and locally fast convergent methods for the solution of (1.1). In the following, we give a short summary of the basic ideas of some of the methods that are related to this paper.

Nonsmooth Newton Methods. Instead of solving problem (1.1) by the classical Newton method, one can apply a nonsmooth Newton method based, e.g., on Clarke's [12] generalized Jacobian $\partial\Phi(x)$ of Φ at the point $x \in \mathbb{R}^n$. For example, the nonsmooth

*Received by the editors October 14, 1997; accepted for publication (in revised form) May 8, 1998; published electronically March 17, 1999.

http://www.siam.org/journals/siopt/9-2/32878.html

†University of Hamburg, Institute of Applied Mathematics, Bundesstrasse 55, 20146 Hamburg, Germany (kanzow@math.uni-hamburg.de). The research of this author was supported by Deutsche Forschungsgemeinschaft.

‡Department of Engineering-Economic Systems and Operations Research, Terman Engineering Center, Stanford University, Stanford, CA 94305-4023 (pieper@stanford.edu).

Newton methods by Kummer [30] and Qi and Sun [37] solve at each iteration the generalized Newton equation

$$V_k d = -\Phi(x^k), \tag{1.2}$$

where $V_k \in \partial\Phi(x^k)$. This method is locally superlinearly/quadratically convergent under certain assumptions but (in contrast to the classical Newton method for smooth systems of equations) cannot be globalized in a simple way for general operators Φ. However, by using special functions Φ, several authors have recently presented globally and locally fast convergent nonsmooth Newton-type methods; see, e.g., [25, 16, 13, 28, 5].

One of the main advantages of most of these methods is the fact that they are usually well defined for an arbitrary complementarity problem NCP(F).

Smoothing Methods. Another way to deal with the nonsmoothness of Φ is to approximate this function by a smooth operator $\Phi_\mu : \mathbb{R}^n \to \mathbb{R}^n$, where $\mu > 0$ denotes the smoothing parameter. The basic idea of the class of smoothing methods is then to solve a sequence of problems

$$\Phi_\mu(x) = 0 \tag{1.3}$$

and to force μ to go to 0. The advantage of this approach is that one can apply the standard Newton method for solving problem (1.3) so that one has to solve at each iteration the smoothing Newton equation

$$\Phi'_\mu(x^k)d = -\Phi_\mu(x^k). \tag{1.4}$$

Smoothing methods of this kind were considered, e.g., by Chen and Harker [6, 7], Chen and Mangasarian [9], Kanzow [26], Gabriel and Moré [20], Burke and Xu [3, 43], Xu [41, 42], Hotta and Yoshise [23], Chen and Ye [11], Chen and Chen [4], Chen and Xiu [8], Jiang [24], Qi and Sun [36], and Tseng [40]. In particular, the paper [3] by Burke and Xu initiated much of the recent research in this area.

The disadvantage of smoothing methods is that they usually require F to be at least a P_0-function in order to guarantee that the linear systems (1.4) are solvable. It seems difficult to make smoothing methods work on general complementarity problems, where the Jacobian in (1.4) might be singular. This problem is also illustrated by the fact that smoothing methods try to follow the so-called smoothing path, which may not exist for non-P_0- or nonmonotone problems.

Nevertheless, a sophisticated implementation, like in the SMOOTH code by Chen and Mangasarian [9], seems to work quite well also for nonmonotone problems; see [2].

Jacobian Smoothing Methods. The third class of algorithms for the solution of (1.1) is due to Chen, Qi, and Sun [10]. They call it a smoothing Newton method, but we prefer the name Jacobian smoothing method in order to distinguish it better from the class of smoothing methods. These methods try to solve at each iteration the mixed Newton equation

$$\Phi'_\mu(x^k)d = -\Phi(x^k). \tag{1.5}$$

This linear system combines the nonsmooth Newton equation (1.2) and the smoothing Newton equation (1.4): it uses the unperturbed right-hand side from (1.2) but the smooth matrix from (1.4).

The algorithm and convergence theory developed by Chen, Qi, and Sun [10] still relies on the fact that the linear systems (1.5) are solvable at each iteration, and, similarly to the class of smoothing methods, this assumption is intimately related to F being a P_0-function. Hence also, this Jacobian smoothing method is not well defined for general complementarity problems.

Note that the Jacobian smoothing idea is also used in a couple of recent smoothing papers as a kind of hybrid step; see, e.g., [11, 4]. The main reason for doing this is that the Jacobian smoothing method helps to prove (or simplifies the proof of) local fast convergence.

Despite the fact that Jacobian smoothing methods are often viewed as a variation of smoothing methods, we take a different point of view: We view a Jacobian smoothing method as a suitable perturbation of a nonsmooth Newton method. In fact, the Jacobian smoothing method seems to be much closer to nonsmooth Newton methods than to smoothing methods, since the Jacobian smoothing methods do not try to follow any smoothing path. Instead, they also try to solve the unperturbed problem (1.1) directly by replacing the matrix $V_k \in \partial\Phi(x^k)$ in (1.2) by a suitable approximation $\Phi'_\mu(x^k)$.

With this in mind, it seems reasonable to ask if one can modify the Jacobian smoothing method by Chen, Qi, and Sun [10] in such a way that it becomes well defined for general complementarity problems. This is actually the main motivation for this paper, and the answer is positive.

In order to do this, however, we cannot consider the general class of smoothing methods used by Chen, Qi, and Sun [10]. Instead, we concentrate on one particular reformulation of the complementarity problem NCP(F) and fully exploit the (additional) properties of this special reformulation. It is based on the Fischer–Burmeister function $\varphi : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$\varphi(a,b) := \sqrt{a^2+b^2} - a - b;$$

see [18]. Then it is well known and easy to see that problem NCP(F) is equivalent to problem (1.1) with Φ being defined by

$$\Phi(x) := \begin{pmatrix} \varphi(x_1, F_1(x)) \\ \vdots \\ \varphi(x_n, F_n(x)) \end{pmatrix}.$$

The globalization strategy for our algorithm is heavily based on the natural merit function $\Psi : \mathbb{R}^n \to \mathbb{R}$ given by

$$\Psi(x) := \frac{1}{2}\Phi(x)^T\Phi(x).$$

The corresponding smooth operator $\Phi_\mu : \mathbb{R}^n \to \mathbb{R}^n$ is defined similarly by

$$\Phi_\mu(x) := \begin{pmatrix} \varphi_\mu(x_1, F_1(x)) \\ \vdots \\ \varphi_\mu(x_n, F_n(x)) \end{pmatrix},$$

where $\varphi_\mu : \mathbb{R}^2 \to \mathbb{R}$ denotes Kanzow's [26] smooth approximation

$$\varphi_\mu(a,b) := \sqrt{a^2+b^2+2\mu} - a - b, \quad \mu > 0,$$

of the Fischer–Burmeister function.

The basic idea of the Jacobian smoothing method to be presented in this paper is to solve the nonlinear complementarity problem NCP(F) by minimizing the merit function Ψ. Unfortunately, given an iterate x^k, the search direction d^k computed from the mixed Newton equation (1.5) is not necessarily a descent direction for Ψ at the point x^k; instead, this search direction is used in order to reduce the related merit function

$$\Psi_\mu(x) := \frac{1}{2}\Phi_\mu(x)^T\Phi_\mu(x).$$

In order to make the algorithm at least well defined for an arbitrary nonlinear complementarity problem, we use a gradient step for the merit function Ψ in case the linear system (1.5) does not have a solution or gives a poor search direction for Ψ_μ. Besides the fact that the introduction of such a gradient step is a rather simple idea, it complicates the global convergence analysis considerably. Basically, this is due to the fact that we now minimize different merit functions, and a reduction in one merit function does not necessarily correspond to a reduction in the other merit function. The global convergence analysis is therefore somewhat more difficult than for many nonsmooth Newton and smoothing methods; in particular, it is based on a rather sophisticated updating rule for the smoothing parameter μ.

The organization of this paper is as follows: The mathematical background and some preliminary results are summarized in section 2. The Jacobian smoothing idea is discussed in more detail in section 3. The algorithm, together with some of its elementary properties, is presented in section 4. The global and local convergence analysis is part of sections 5 and 6, respectively. Extensive and very encouraging numerical results are reported in section 7, and section 8 concludes this paper with some final remarks.

Some words about our notation: Let $G : \mathbb{R}^n \to \mathbb{R}^m$ be continuously differentiable. Then $G'(x) \in \mathbb{R}^{m\times n}$ denotes the Jacobian of G at a point $x \in \mathbb{R}^n$, whereas the symbol $\nabla G(x)$ is used for the transposed Jacobian. In particular, if $m = 1$, the gradient $\nabla G(x)$ is viewed as a column vector. If $G : \mathbb{R}^n \to \mathbb{R}^m$ is only locally Lipschitzian, we can define Clarke's [12] *generalized Jacobian* as follows:

$$\partial G(x) := \operatorname{conv}\left\{H \in \mathbb{R}^{m\times n} \,|\, \exists \{x^k\} \subseteq D_G : x^k \to x \text{ and } G'(x^k) \to H\right\};$$

here, D_G denotes the set of differentiable points of G and conv$\mathcal{A}$ is the convex hull of a set $\mathcal{A}$. If $m = 1$, we call $\partial G(x)$ the *generalized gradient* of G at x for obvious reasons.

Usually, $\partial G(x)$ is difficult to compute, especially for $m > 1$. Instead, Proposition 2.6.2 (e) in Clarke [12] provides the overestimation

$$\partial G(x)^T \subseteq \partial G_1(x) \times \cdots \times \partial G_m(x),$$

where the right-hand side denotes the set of matrices in $\mathbb{R}^{n\times m}$ whose ith column is given by the generalized gradient of the ith component function G_i. Since this right-hand side is often easier to compute and was motivated by the recent paper [34] by Qi, we write

$$\partial_C G(x)^T := \partial G_1(x) \times \cdots \times \partial G_m(x)$$

and call $\partial_C G(x)$ the *C-subdifferential* of G at x. For the purpose of this paper, the C-subdifferential is considerably more important than the more familiar generalized Jacobian.

If $x \in \mathbb{R}^n$, we denote by $\|x\|$ the Euclidian norm of x. Similarly, $\|A\|$ denotes the spectral norm of a matrix $A \in \mathbb{R}^{n\times n}$ which is the induced matrix norm of the Euclidian vector norm. Occasionally, we will also write $\|\cdot\|_2$ in order to avoid any possible confusion. Sometimes we also need the Frobenius norm $\|A\|_F$ of a matrix $A \in \mathbb{R}^{n\times n}$.

If $A \in \mathbb{R}^{n\times n}$ is any given matrix and $\mathcal{A} \subseteq \mathbb{R}^{n\times n}$ is a nonempty set of matrices, we denote by $\operatorname{dist}(A, \mathcal{A}) := \inf_{B\in\mathcal{A}} \|A - B\|$ the distance between A and $\mathcal{A}$. This is sometimes also written as $\operatorname{dist}_2(A, \mathcal{A})$ in order to emphasize that the distance is measured using the spectral norm. Similarly, we write $\operatorname{dist}_F(A, \mathcal{A})$ if the distance is calculated by using the Frobenius norm. The (Euclidian) distance between a vector and a set of vectors of the same dimension is defined in an analogous way.

Finally, we make use of the Landau symbols $o(\cdot)$ and $O(\cdot)$: Let $\{\alpha_k\}$ and $\{\beta_k\}$ be two sequences of positive numbers such that $\beta_k \to 0$. Then we write $\alpha_k = o(\beta_k)$ if $\alpha_k/\beta_k \to 0$ and $\alpha_k = O(\beta_k)$ if $\limsup_{k\to\infty} \alpha_k/\beta_k < \infty$, i.e., if there exists a constant $c > 0$ such that $\alpha_k \le c\beta_k$ for all $k \in \mathbb{N} := \{0, 1, 2, \ldots\}$.

2. Preliminaries. In this section, we summarize some of the known properties of the functions Φ, Φ_μ, and Ψ, which will be important for our subsequent analysis. In addition, we prove some preliminary results which will also be used later.

The first result follows directly from the definition of the C-subdifferential and Proposition 3.1 in [16].

PROPOSITION 2.1. *For an arbitrary $x \in \mathbb{R}^n$, we have*

$$\partial_C \Phi(x)^T = D_a(x) + \nabla F(x) D_b(x), \tag{2.1}$$

where $D_a(x) = \operatorname{diag}(a_1(x), \ldots, a_n(x))$, $D_b(x) = \operatorname{diag}(b_1(x), \ldots, b_n(x)) \in \mathbb{R}^{n\times n}$ are diagonal matrices whose ith diagonal element is given by

$$a_i(x) = \frac{x_i}{\sqrt{x_i^2 + F_i(x)^2}} - 1, \qquad b_i(x) = \frac{F_i(x)}{\sqrt{x_i^2 + F_i(x)^2}} - 1$$

if $(x_i, F_i(x)) \neq (0, 0)$ and by

$$a_i(x) = \xi_i - 1, \qquad b_i(x) = \rho_i - 1$$

for every $(\xi_i, \rho_i) \in \mathbb{R}^2$ such that $\|(\xi_i, \rho_i)\| \le 1$ if $(x_i, F_i(x)) = (0, 0)$.

The next result follows from [16, 19] together with known results for (strongly) semismooth functions [37] and the recent theory of C-differentiable functions by Qi [34].

PROPOSITION 2.2. *Assume that $\{x^k\} \subseteq \mathbb{R}^n$ is any convergent sequence with limit point $x^* \in \mathbb{R}^n$. Then the following statements hold:*

(a) *The function Φ is semismooth so that*

$$\|\Phi(x^k) - \Phi(x^*) - H_k(x^k - x^*)\| = o(\|x^k - x^*\|)$$

for any $H_k \in \partial_C \Phi(x^k)$.

(b) *If F is continuously differentiable with a locally Lipschitzian Jacobian, then Φ is strongly semismooth so that*

$$\|\Phi(x^k) - \Phi(x^*) - H_k(x^k - x^*)\| = O(\|x^k - x^*\|^2)$$

for any $H_k \in \partial_C \Phi(x^k)$.

The following result can be verified similarly to Lemma 3.7 in [27].

PROPOSITION 2.3. *The function φ_μ satisfies the inequality*

$$|\varphi_{\mu_1}(a,b) - \varphi_{\mu_2}(a,b)| \le \sqrt{2}|\sqrt{\mu_1} - \sqrt{\mu_2}|$$

for all $(a,b) \in \mathbb{R}^2$ and all $\mu_1, \mu_2 \ge 0$. In particular, we have

$$|\varphi_\mu(a,b) - \varphi(a,b)| \le \sqrt{2}\sqrt{\mu}$$

for all $(a,b) \in \mathbb{R}^2$ and all $\mu > 0$.

As an immediate consequence of Proposition 2.3, we obtain the following corollary.

COROLLARY 2.4. *The function Φ_μ satisfies the inequality*

$$\|\Phi_{\mu_1}(x) - \Phi_{\mu_2}(x)\| \le \kappa\,|\sqrt{\mu_1} - \sqrt{\mu_2}\,| \tag{2.2}$$

for all $x \in \mathbb{R}^n$ and $\mu_1, \mu_2 \ge 0$, where $\kappa := \sqrt{2n}$. In particular, we have

$$\|\Phi_\mu(x) - \Phi(x)\| \le \kappa\,\sqrt{\mu}$$

for all $x \in \mathbb{R}^n$ and all $\mu \ge 0$.

We next state a result which is a minor extension of Proposition 3.4 of [16]. We omit its proof here since it can be carried out in a similar way as the one in [16].

PROPOSITION 2.5. *The merit function Ψ is continuously differentiable with $\nabla\Psi(x) = V^T\Phi(x)$ for an arbitrary $V \in \partial_C\Phi(x)$.*

The following technical result will be used in the proof of our main global convergence result, Theorem 5.8 below.

LEMMA 2.6. *Let $\{x^k\} \subseteq \mathbb{R}^n$ and $\{\mu_k\} \subseteq \mathbb{R}$ be two sequences with $\{x^k\} \to x^*$ for some $x^* \in \mathbb{R}^n$ and $\{\mu_k\} \downarrow 0$. Then*

$$\lim_{k\to\infty} \nabla\Psi_{\mu_k}(x^k) = \nabla\Psi(x^*)$$

and

$$\lim_{k\to\infty} \Phi'_{\mu_k}(x^k)^T\Phi(x^k) = \nabla\Psi(x^*).$$

Proof. Since Ψ_μ is differentiable for all $\mu > 0$, we have

$$\nabla\Psi_{\mu_k}(x^k) = \Phi'_{\mu_k}(x^k)^T\Phi_{\mu_k}(x^k) = \sum_{i\in I}\varphi_{\mu_k}(x_i^k, F_i(x^k))\nabla\Phi_{\mu_k,i}(x^k),$$

where $\Phi_{\mu_k,i}$ denotes the ith component function of Φ_{μ_k}. On the other hand, for arbitrary $V \in \partial_C\Phi(x^*)$, we obtain from Proposition 2.5

$$\nabla\Psi(x^*) = V^T\Phi(x^*) = \sum_{i\in I}\varphi(x_i^*, F_i(x^*))V_i^T,$$

where V_i^T denotes the ith column of the matrix V^T. Now let

$$\beta(x^*) := \{i \mid x_i^* = F_i(x^*) = 0\}.$$

We consider two cases.

Case 1. $i \notin \beta(x^*)$.

Then the Fischer–Burmeister function is continuously differentiable at $(x_i^*, F_i(x^*))$, and the ith column of V^T is single valued and equal to $\nabla \Phi_i(x^*)$ (cf. Proposition 2.1). In particular, all limits exist, and from the continuity of φ and ∇F, we obtain

$$\lim_{k\to\infty} \varphi_{\mu_k}(x_i^k, F_i(x^k))\nabla \Phi_{\mu_k,i}(x^k) = \varphi(x_i^*, F_i(x^*))\nabla \Phi_i(x^*) = \varphi(x_i^*, F_i(x^*))V_i^T.$$

Case 2. $i \in \beta(x^*)$.

Since

$$\frac{\partial \varphi_\mu}{\partial a}(a,b) \in (-2,0) \quad \text{and} \quad \frac{\partial \varphi_\mu}{\partial b}(a,b) \in (-2,0)$$

for all $(a,b) \in \mathbb{R}^2$ and $\mu > 0$, the sequence $\{\nabla \Phi_{\mu_k,i}(x^k)\}$ is bounded for $k \to \infty$. Since

$$\lim_{k\to\infty} \varphi_{\mu_k}(x_i^k, F_i(x^k)) = \varphi(x_i^*, F_i(x^*)) = 0,$$

we therefore have

$$\lim_{k\to\infty} \varphi_{\mu_k}(x_i^k, F_i(x^k))\nabla \Phi_{\mu_k,i}(x^k) = 0.$$

Since we also have $\varphi(x_i^*, F_i(x^*))V_i^T = 0$ for all $i \in \beta(x^*)$, the first statement follows from Cases 1 and 2.

The second statement is easier to establish than the first one since we multiply by $\Phi(x^k)$ and not by $\Phi_{\mu_k}(x^k)$. The proof would be similar to the one just given. □

We conclude this section by stating another technical result that will also be utilized in our global convergence analysis.

LEMMA 2.7. *Let $\{x^k\}, \{d^k\} \subseteq \mathbb{R}^n$ and $\{t_k\} \subseteq \mathbb{R}$ be sequences with $x^{k+1} := x^k + t_k d^k$ such that $\{x^k\} \to x^*$, $\{d^k\} \to d^*$, and $\{t_k\} \downarrow 0$ for certain vectors $x^*, d^* \in \mathbb{R}^n$. Furthermore, let $\{\mu_k\} \subseteq \mathbb{R}$ be a sequence with $\{\mu_k\} \downarrow 0$. Then*

$$\lim_{k\to\infty} \frac{\Psi_{\mu_k}(x^k + t_k d^k) - \Psi_{\mu_k}(x^k)}{t_k} = \nabla \Psi(x^*)^T d^*.$$

Proof. From Proposition 2.5 and the mean value theorem, we obtain that, for each $k \in \mathbb{N}$, there exists a vector $\xi^k \in \mathbb{R}^n$ on the line segment between x^k and x^{k+1} (that is, $\xi^k = x^k + \theta_k d^k$ for some $\theta_k \in [0, t_k]$) such that

$$\Psi_{\mu_k}(x^k + t_k d^k) - \Psi_{\mu_k}(x^k) = t_k \nabla \Psi_{\mu_k}(\xi^k)^T d^k.$$

Dividing by t_k gives

$$\frac{\Psi_{\mu_k}(x^k + t_k d^k) - \Psi_{\mu_k}(x^k)}{t_k} = \nabla \Psi_{\mu_k}(\xi^k)^T d^k.$$

Since ξ^k lies between x^k and x^{k+1}, it follows that $\{\xi^k\} \to x^*$. Therefore, we can apply the first statement of Lemma 2.6, so that passing to the limit, we get

$$\lim_{k\to\infty} \frac{\Psi_{\mu_k}(x^k + t_k d^k) - \Psi_{\mu_k}(x^k)}{t_k} = \lim_{k\to\infty} \nabla \Psi_{\mu_k}(\xi^k)^T d^k = \nabla \Psi(x^*)^T d^*.$$

This completes the proof. □

3. Jacobian smoothing. The basic idea of our algorithm, to be presented in section 4, is to replace the generalized Newton equation

$$V_k d = -\Phi(x^k), \quad V_k \in \partial_C \Phi(x^k),$$

by the linear system

$$\Phi'_{\mu_k}(x^k) d = -\Phi(x^k);$$

i.e., we replace the element V_k from the C-subdifferential $\partial_C \Phi(x^k)$ by the (existing) Jacobian $\Phi'_{\mu_k}(x^k)$ of the smoothed operator Φ_{μ_k}. In order to guarantee local fast convergence of this iteration, we have to control the difference between $\Phi'_{\mu_k}(x^k)$ and the set $\partial_C \Phi(x^k)$. A first result in this direction is established in the following lemma.

Lemma 3.1. *Let $x \in \mathbb{R}^n$ be arbitrary but fixed. Then we have*

$$\lim_{\mu \downarrow 0} \operatorname{dist}(\Phi'_\mu(x), \partial_C \Phi(x)) = 0. \tag{3.1}$$

Proof. From the definition of Φ_μ, we have for all $\mu > 0$,

$$\Phi'_\mu(x) = \operatorname{diag}\left(\frac{x_i}{\sqrt{x_i^2 + F_i(x)^2 + 2\mu}} - 1\right) + \operatorname{diag}\left(\frac{F_i(x)}{\sqrt{x_i^2 + F_i(x)^2 + 2\mu}} - 1\right) F'(x).$$

We consider the distance between the columns of the transposed Jacobians.

To this end, let us define

$$\beta(x) := \{i \,|\, x_i = F_i(x) = 0\}.$$

If we denote the ith component function of Φ_μ by $\Phi_{\mu,i}$, we obtain

$$\lim_{\mu \downarrow 0} \nabla \Phi_{\mu,i}(x) = \begin{cases} \left(\frac{x_i}{\sqrt{x_i^2+F_i(x)^2}} - 1\right) e_i + \left(\frac{F_i(x)}{\sqrt{x_i^2+F_i(x)^2}} - 1\right) \nabla F_i(x) & \text{for } i \notin \beta(x), \\ -e_i - \nabla F_i(x) & \text{for } i \in \beta(x). \end{cases}$$

Hence the assertion follows from Proposition 2.1 (with $(\xi_i, \rho_i) = (0,0)$ for $i \in \beta(x)$). □

It is an immediate consequence of Lemma 3.1 that we can find, for every fixed $\delta > 0$, a parameter $\bar{\mu} = \bar{\mu}(x, \delta) > 0$ such that

$$\operatorname{dist}(\Phi'_\mu(x), \partial_C \Phi(x)) \leq \delta$$

for all $0 < \mu \leq \bar{\mu}$. However, it does not follow from Lemma 3.1 how we can choose this threshold value $\bar{\mu}$. On the other hand, it is important for the design of our algorithm to have an explicit expression of a possible value of $\bar{\mu}$. This is made more precise in Proposition 3.4 below, whose proof is based on the following two observations.

Lemma 3.2. *Let $x \in \mathbb{R}^n$ and $\mu > 0$ be arbitrary but fixed. Then*

$$\left[\operatorname{dist}_F \left(\nabla \Phi_\mu(x), \partial_C \Phi(x)^T\right)\right]^2 = \sum_{i=1}^n \left[\operatorname{dist}_2 \left(\nabla \Phi_{\mu,i}(x), \partial \Phi_i(x)\right)\right]^2.$$

Proof. Let V_i be the ith column of a matrix V. Then, using the definition of the C-subdifferential, it is easy to see that

$$\inf_{V \in \partial_C \Phi(x)^T} \sum_{i=1}^n \|\nabla \Phi_{\mu,i}(x) - V_i\|_2^2 = \sum_{i=1}^n \inf_{H_i \in \partial \Phi_i(x)} \|\nabla \Phi_{\mu,i}(x) - H_i\|_2^2.$$

Using this and the definition of the Frobenius norm, we obtain

$$\begin{aligned}
[\operatorname{dist}_F(\nabla\Phi_\mu(x), \partial_C\Phi(x)^T)]^2 &= \inf_{V\in\partial_C\Phi(x)^T} \|\nabla\Phi_\mu(x) - V\|_F^2 \\
&= \inf_{V\in\partial_C\Phi(x)^T} \sum_{i=1}^n \|\nabla\Phi_{\mu,i}(x) - V_i\|_2^2 \\
&= \sum_{i=1}^n \inf_{H_i\in\partial\Phi_i(x)} \|\nabla\Phi_{\mu,i}(x) - H_i\|_2^2 \\
&= \sum_{i=1}^n [\operatorname{dist}_2(\nabla\Phi_{\mu,i}(x), \partial\Phi_i(x))]^2 .
\end{aligned}$$

This completes the proof. □

LEMMA 3.3. *Let $\mu > 0$ be arbitrary but fixed. Then the function $f : (0,\infty) \to \mathbb{R}$, defined by*

$$f(\tau) := \frac{1}{\sqrt{\tau}} - \frac{1}{\sqrt{\tau + 2\mu}},$$

is strictly decreasing in $\tau > 0$.

Proof. The function f is continuously differentiable with

$$f'(\tau) = -\frac{1}{2}\frac{1}{(\sqrt{\tau})^3} + \frac{1}{2}\frac{1}{\sqrt{\tau+2\mu}^3} = -\frac{1}{2}\left(\frac{1}{(\sqrt{\tau})^3} - \frac{1}{\sqrt{\tau+2\mu}^3}\right).$$

Hence we have $f'(\tau) < 0$ for all $\tau > 0$. This implies our assertion. □

We now come to the main result of this section.

PROPOSITION 3.4. *Let $x \in \mathbb{R}^n$ be arbitrary but fixed. Assume that x is not a solution of NCP(F). Let us define the constants*

$$\gamma(x) := \max_{i\notin\beta(x)} \{\|x_i e_i + F_i(x)\nabla F_i(x)\|\} \geq 0$$

and

$$\alpha(x) := \min_{i\notin\beta(x)} \{x_i^2 + F_i(x)^2\} > 0,$$

where $\beta(x) := \{i|\, x_i = F_i(x) = 0\}$. Let $\delta > 0$ be given, and define

$$\bar{\mu}(x,\delta) := \begin{cases} 1 & \text{if } \left(\frac{n\gamma(x)^2}{\delta^2} - \alpha(x)\right) \leq 0, \\ \frac{\alpha(x)^2}{2}\left(\frac{\delta^2}{n\gamma(x)^2 - \delta^2\alpha(x)}\right) & \text{otherwise.} \end{cases}$$

Then

$$\operatorname{dist}_F(\Phi'_\mu(x), \partial_C\Phi(x)) \leq \delta$$

for all μ such that $0 < \mu \leq \bar{\mu}(x,\delta)$.

Proof. We first note that $\{1,\dots,n\}\setminus\beta(x) \neq \emptyset$ since x is not a solution of NCP(F) by assumption. Hence $\alpha(x) > 0$. Furthermore, since $\|A\|_F = \|A^T\|_F$ for an arbitrary matrix $A \in \mathbb{R}^{n\times n}$, we obtain

$$\begin{aligned}
\operatorname{dist}_F\left(\Phi'_\mu(x), \partial_C\Phi(x)\right) &= \operatorname{dist}_F(\nabla\Phi_\mu(x), \partial_C\Phi(x)^T) \\
&= \sqrt{\textstyle\sum_{i=1}^n [\operatorname{dist}_2(\nabla\Phi_{\mu,i}(x), \partial\Phi_i(x))]^2}
\end{aligned} \tag{3.2}$$

from Lemma 3.2. Hence it is sufficient to consider the distance between the ith columns of $\nabla\Phi_\mu(x)$ and $\partial_C\Phi(x)^T$. To this end, we recall that these columns are given by

$$\nabla\Phi_{\mu,i}(x) = \frac{\partial\varphi_\mu}{\partial a}(x_i, F_i(x))e_i + \frac{\partial\varphi_\mu}{\partial b}(x_i, F_i(x))\nabla F_i(x)$$

and

$$\partial\Phi_i(x) = \begin{cases} \frac{\partial\varphi}{\partial a}(x_i, F_i(x))e_i + \frac{\partial\varphi}{\partial b}(x_i, F_i(x))\nabla F_i(x) & \text{if } i \notin \beta(x), \\ (\xi_i - 1)e_i + (\rho_i - 1)\nabla F_i(x) & \text{if } i \in \beta(x), \end{cases}$$

respectively, where $(\xi_i, \rho_i) \in \mathbb{R}^2$ denotes any vector such that $\|(\xi_i, \rho_i)\| \leq 1$; see Proposition 2.1. We distinguish two cases:

Case 1. $i \in \beta(x)$.

Then $(x_i, F_i(x)) = (0, 0)$ and therefore

$$\nabla\Phi_{\mu,i}(x) = -e_i - \nabla F_i(x).$$

Hence, taking $(\xi_i, \rho_i) = (0, 0)$, we see that

$$\nabla\Phi_{\mu,i}(x) \in \partial\Phi_i(x)$$

so that

$$\operatorname{dist}_2\left(\nabla\Phi_{\mu,i}(x), \partial\Phi_i(x)\right) = 0 \tag{3.3}$$

for all $i \in \beta(x)$.

Case 2. $i \notin \beta(x)$.

In this case, we have

$$\partial\Phi_i(x) = \{\nabla\Phi_i(x)\}.$$

By a simple calculation, we therefore get

$$\begin{aligned}
&\operatorname{dist}_2\left(\nabla\Phi_{\mu,i}(x), \partial\Phi_i(x)\right) \\
&= \|\nabla\Phi_{\mu,i}(x) - \nabla\Phi_i(x)\| \\
&= \left\| \left(\frac{x_i}{\sqrt{x_i^2 + F_i(x)^2 + 2\mu}} - 1 \right) e_i + \left(\frac{F_i(x)}{\sqrt{x_i^2 + F_i(x)^2 + 2\mu}} - 1 \right) \nabla F_i(x) \right. \\
&\quad \left. - \left(\frac{x_i}{\sqrt{x_i^2 + F_i(x)^2}} - 1 \right) e_i - \left(\frac{F_i(x)}{\sqrt{x_i^2 + F_i(x)^2}} - 1 \right) \nabla F_i(x) \right\| \\
&= \left\| x_i e_i \left(\frac{1}{\sqrt{x_i^2 + F_i(x)^2 + 2\mu}} - \frac{1}{\sqrt{x_i^2 + F_i(x)^2}} \right) \right. \\
&\quad \left. + F_i(x)\nabla F_i(x) \left(\frac{1}{\sqrt{x_i^2 + F_i(x)^2 + 2\mu}} - \frac{1}{\sqrt{x_i^2 + F_i(x)^2}} \right) \right\| \\
&= \left\| \left(\frac{1}{\sqrt{x_i^2 + F_i(x)^2 + 2\mu}} - \frac{1}{\sqrt{x_i^2 + F_i(x)^2}} \right) (x_i e_i + F_i(x)\nabla F_i(x)) \right\| \\
&= \left(\frac{1}{\sqrt{x_i^2 + F_i(x)^2}} - \frac{1}{\sqrt{x_i^2 + F_i(x)^2 + 2\mu}} \right) \|x_i e_i + F_i(x)\nabla F_i(x)\|.
\end{aligned}$$

In view of the definitions of the constants $\alpha(x)$ and $\gamma(x)$, we therefore obtain by using Lemma 3.3

$$\begin{aligned}\operatorname{dist}_2(\nabla\Phi_{\mu,i}(x), \partial\Phi_i(x)) &\leq \left(\frac{1}{\sqrt{\alpha(x)}} - \frac{1}{\sqrt{\alpha(x)+2\mu}}\right)\gamma(x)\\ &= \left(\frac{\sqrt{\alpha(x)+2\mu}-\sqrt{\alpha(x)}}{\sqrt{\alpha(x)}\sqrt{\alpha(x)+2\mu}}\right)\gamma(x)\\ &\leq \left(\frac{\sqrt{2\mu}}{\sqrt{\alpha(x)}\sqrt{\alpha(x)+2\mu}}\right)\gamma(x),\end{aligned}$$

where the latter inequality follows from the elementary fact that $\sqrt{a+b} \leq \sqrt{a}+\sqrt{b}$ for all $a, b \geq 0$. We now want to show that

$$\left(\frac{\sqrt{2\mu}}{\sqrt{\alpha(x)}\sqrt{\alpha(x)+2\mu}}\right)\gamma(x) \leq \frac{\delta}{\sqrt{n}} \tag{3.4}$$

for all $0 < \mu \leq \bar{\mu}(x,\delta)$, which then implies

$$\operatorname{dist}_2\left(\nabla\Phi_{\mu,i}(x), \partial\Phi_i(x)\right) \leq \frac{\delta}{\sqrt{n}}. \tag{3.5}$$

If $\gamma(x) = 0$, then inequality (3.4) holds trivially (for arbitrary $\mu > 0$). Hence we assume that $\gamma(x) > 0$. Then an easy calculation shows that (3.4) is equivalent to

$$\alpha(x)^2 \geq 2\mu\left(\frac{n\gamma(x)^2}{\delta^2} - \alpha(x)\right). \tag{3.6}$$

Hence, if $\frac{n\gamma(x)^2}{\delta^2} - \alpha(x) \leq 0$, inequality (3.4) is satisfied for any $\mu > 0$, in particular for all $\mu \in (0,1]$. Otherwise we obtain the following upper bound from (3.6):

$$\mu \leq \frac{\alpha(x)^2}{2}\left(\frac{\delta^2}{n\gamma(x)^2 - \delta^2\alpha(x)}\right) =: \bar{\mu}(x,\delta).$$

Putting together (3.2), (3.3), and (3.5), we therefore obtain

$$\operatorname{dist}_F(\Phi_\mu'(x), \partial_C\Phi(x)) \leq \sqrt{\sum_{i=1}^n \frac{\delta^2}{n}} = \delta$$

for all $0 < \mu \leq \bar{\mu}(x,\delta)$. □

The constant $\bar{\mu}(x,\delta)$ defined in Proposition 3.4 will play a central role in the design of our algorithm, to be described in the following section.

We also note that, since $\|A\| \leq \|A\|_F$ for an arbitrary matrix $A \in \mathbb{R}^{n\times n}$, it follows from Proposition 3.4 that

$$\operatorname{dist}(\Phi_\mu'(x), \partial_C\Phi(x)) \leq \delta$$

for all μ with $0 < \mu \leq \bar{\mu}(x,\delta)$.

4. Algorithm.

In this section, we give a detailed description of our Jacobian smoothing method and state some of its elementary properties. In particular, we show that the algorithm is well defined for an arbitrary complementarity problem.

Basically, we try to take the Jacobian smoothing method from Chen, Qi, and Sun [10]. In addition, we incorporate a gradient step in a way similar to (but slightly different from) the way this is done by some nonsmooth Newton methods [13, 28, 5]. Unfortunately, the introduction of these gradient steps makes the updating rules for our smoothing parameter μ_k, as well as the convergence theory, considerably more technical and complicated. However, it is this gradient step which makes the algorithm applicable to a general nonlinear complementarity problem.

In fact, this is also the reason why we concentrate on the Fischer–Burmeister function: Its merit function Ψ is smooth due to Proposition 2.5, whereas the same does not hold for the general class of smoothing functions considered in [10].

We now state our algorithm formally.

ALGORITHM 4.1 (Jacobian Smoothing Method).

(S.0) *Choose* $x^0 \in \mathbb{R}^n$, $\lambda, \alpha, \eta, \rho \in (0,1)$, $\gamma > 0$, $\sigma \in (0, \frac{1}{2}(1-\alpha))$, $p > 2$, *and* $\epsilon \geq 0$. *Set* $\beta_0 := \|\Phi(x^0)\|$, $\kappa := \sqrt{2n}$, $\mu_0 := (\frac{\alpha}{2\kappa}\beta_0)^2$, *and* $k := 0$.

(S.1) *If* $\|\nabla\Psi(x^k)\| \leq \epsilon$: *STOP.*

(S.2) *Find a solution* $d^k \in \mathbb{R}^n$ *of the linear system*

$$\Phi'_{\mu_k}(x^k)d = -\Phi(x^k). \quad \text{(Newton step)} \tag{4.1}$$

If the system (4.1) *is not solvable or if the condition*

$$\Phi(x^k)^T \Phi'_{\mu_k}(x^k)d^k \leq -\rho\|d^k\|^p \tag{4.2}$$

is not satisfied, set

$$d^k := -\nabla\Psi(x^k). \quad \text{(Gradient step)} \tag{4.3}$$

(S.3) *Find the smallest* m_k *in* $\{0, 1, 2, \ldots\}$ *such that*

$$\Psi_{\mu_k}(x^k + \lambda^{m_k} d^k) \leq \Psi_{\mu_k}(x^k) - 2\sigma\lambda^{m_k}\Psi(x^k) \tag{4.4}$$

if d^k *is given by* (4.1), *and such that*

$$\Psi(x^k + \lambda^{m_k} d^k) \leq \Psi(x^k) - \sigma\lambda^{m_k}\|d^k\|^2 \tag{4.5}$$

if d^k *is given by* (4.3). *Set* $t_k := \lambda^{m_k}$ *and* $x^{k+1} := x^k + t_k d^k$.

(S.4) *If*

$$\|\Phi(x^{k+1})\| \leq \max\left\{\eta\beta_k, \frac{1}{\alpha}\|\Phi(x^{k+1}) - \Phi_{\mu_k}(x^{k+1})\|\right\}, \tag{4.6}$$

then set

$$\beta_{k+1} := \|\Phi(x^{k+1})\|$$

and choose μ_{k+1} *such that*

$$0 < \mu_{k+1} \leq \min\left\{\left(\frac{\alpha}{2\kappa}\beta_{k+1}\right)^2, \frac{\mu_k}{4}, \bar{\mu}(x^{k+1}, \gamma\beta_{k+1})\right\}. \tag{4.7}$$

If (4.6) *is not satisfied and* $d^k = -\nabla\Psi(x^k)$, *then set*

$$\beta_{k+1} := \beta_k$$

and choose μ_{k+1} *such that*

$$0 < \mu_{k+1} \le \min\left\{ \left(\frac{\alpha}{2\kappa}\|\Phi(x^{k+1})\|\right)^2, \left(\frac{\|\Phi(x^k)\| - \|\Phi(x^{k+1})\|}{2\kappa}\right)^2, \frac{\mu_k}{4} \right\}. \tag{4.8}$$

If none of the above conditions is met, set $\beta_{k+1} := \beta_k$ *and* $\mu_{k+1} := \mu_k$.

(S.5) *Set* $k \leftarrow k+1$, *and return to step* (S.1).

For convenience of presentation, we assume implicitly throughout the theoretical part of this paper that the termination parameter ϵ is equal to 0 and that the algorithm does not terminate after a finite number of iterations.

Before we start to investigate the properties of Algorithm 4.1, we give some comments on it: In step (S.2), we try to solve the (mixed) Newton equation (4.1) which is the main computational effort of our method. If the solution of this linear system does not provide a direction of sufficient decrease (in the sense of (4.2)), we switch to the steepest descent direction of the merit function Ψ.

In step (S.3), we perform a line search. The line search rule depends on the search direction chosen in step (S.2): If d^k is the Newton direction, the line search in (4.4) is used as a globalization strategy. Note that this line search condition is exactly the same as in Chen, Qi, and Sun [10]. On the other hand, if d^k is a gradient step, we use the standard Armijo rule in (4.5).

The complicated part of the algorithm is in step (S.4), where we update the parameter μ_k. The first part of the updating rules (where condition (4.6) is satisfied) is also used by Chen, Qi, and Sun [10]. The second part is due to the gradient step. In the following list, we give some more detailed comments on the role of these two updating rules:

(a) In both updating rules, namely, in (4.7) and (4.8), we reduce μ_k by at least a factor of 1/4. This is reasonable since we want to force μ_k to go to 0.

(b) The last part of the updating rule (4.7) controls the distance between our smooth Jacobian and the C-subdifferential; see Lemma 4.2 (b) below.

(c) The remaining parts of the updating rules (4.7) and (4.8) are important in order to guarantee that Algorithm 4.1 is well defined and globally convergent. We will exploit these rules several times in our convergence proofs.

We now turn to the analysis of Algorithm 4.1. To this end, we introduce the index set

$$K = \{0\} \cup \left\{ k \in \mathbb{N} \,\middle|\, \|\Phi(x^k)\| \le \max\left\{ \eta\beta_{k-1}, \frac{1}{\alpha}\|\Phi(x^k) - \Phi_{\mu_{k-1}}(x^k)\| \right\} \right\}. \tag{4.9}$$

We stress that, compared to the updating rule (4.6), there is a shift of the indices in the definition of the index set K!

We can prove the following result.

LEMMA 4.2. *The following two statements hold:*

(a) *We have*

$$\|\Phi(x^k) - \Phi_{\mu_k}(x^k)\| \le \alpha\|\Phi(x^k)\| \tag{4.10}$$

for all $k \ge 0$.

(b) *We have*

$$\operatorname{dist}_F(\Phi'_{\mu_k}(x^k), \partial_C \Phi(x^k)) \leq \gamma \|\Phi(x^k)\| \tag{4.11}$$

for all $k \in K$ with $k \geq 1$.

Proof. (a) We distinguish three cases.

Case 1. $k \in K$.

Then we obtain from (4.7) and Corollary 2.4

$$\|\Phi(x^k) - \Phi_{\mu_k}(x^k)\| \leq \kappa\sqrt{\mu_k} \leq \frac{\alpha}{2}\beta_k \leq \alpha\beta_k = \alpha\|\Phi(x^k)\|.$$

Case 2. $k \notin K$ and the $(k-1)$st step is a Newton step (i.e., μ_k is not updated by (4.8)).

In this case, we have $\mu_k = \mu_{k-1}$, so that we obtain from (4.6)

$$\|\Phi(x^k) - \Phi_{\mu_k}(x^k)\| = \|\Phi(x^k) - \Phi_{\mu_{k-1}}(x^k)\| < \alpha\|\Phi(x^k)\|.$$

Case 3. $k \notin K$ and the $(k-1)$st step is a gradient step (i.e., μ_k is updated by (4.8)).

Then we obtain from Corollary 2.4 and (4.8)

$$\|\Phi(x^k) - \Phi_{\mu_k}(x^k)\| \leq \kappa\sqrt{\mu_k} \leq \frac{\alpha}{2}\|\Phi(x^k)\| \leq \alpha\|\Phi(x^k)\|.$$

Statement (a) now follows from these three cases.

(b) Statement (b) follows immediately from the definition of the threshold value $\bar{\mu}(x,\delta)$ in Proposition 3.4 and the updating rule (4.7). □

As a consequence of Lemma 4.2, we obtain the following theorem.

THEOREM 4.3. *Algorithm* 4.1 *is well defined.*

Proof. We only have to show that the exponent m_k in the line search rules (4.4)/(4.5) is finite for any $k \in \mathbb{N}$. In case of a gradient step, this is well known since we use the standard Armijo rule. In case of a Newton step, we can use part (a) of Lemma 4.2 and prove the finiteness of m_k in essentially the same way as was done in [10, Lemma 3.1]. □

5. Global convergence.

The aim of this section is to show that any accumulation point of a sequence generated by Algorithm 4.1 is at least a stationary point of Ψ. Unfortunately, the analysis is somewhat technical due to the different updating rules for Newton and gradient steps in Algorithm 4.1. We therefore need a couple of preliminary results. Some of them, however, are of interest on their own.

We begin our global convergence analysis with the following observation.

LEMMA 5.1. *Let $\{x^k\} \subseteq \mathbb{R}^n$ be a sequence generated by Algorithm* 4.1. *Assume that $\{x^k\}$ has an accumulation point x^*, which is a solution of NCP(F). Then the index set K is infinite and $\{\mu_k\} \to 0$.*

Proof. Assume that K is finite. Then it follows from (4.6) and the updating rules for β_k in step (S.4) of Algorithm 4.1 that there is a $k_0 \in \mathbb{N}$ such that

$$\beta_k = \beta_{k_0}$$

and

$$\|\Phi(x^{k+1})\| > \max\left\{\eta\beta_k, \frac{1}{\alpha}\|\Phi(x^{k+1}) - \Phi_{\mu_k}(x^{k+1})\|\right\} \geq \eta\beta_k = \eta\beta_{k_0}$$

for all $k \in \mathbb{N}$ with $k \geq k_0$. However, this contradicts the fact that x^* is a solution of NCP(F), so that we have $\Phi(x^*) = 0$.

Hence K is an infinite set. The updating rules for μ_k therefore immediately imply that the whole sequence $\{\mu_k\}$ converges to 0. □

We will also need the following simple result.

LEMMA 5.2. *The following two statements hold:*

(a) *If d^k is given by* (4.1)*, we have*

$$\|\Phi_{\mu_k}(x^{k+1})\| < \|\Phi_{\mu_k}(x^k)\|.$$

(b) *If $d^k = -\nabla\Psi(x^k)$ and if μ_k is updated by* (4.8)*, then*

$$\|\Phi_{\mu_{k+1}}(x^{k+1})\| \leq \|\Phi_{\mu_{k+1}}(x^k)\|.$$

(Note the difference between the indexes μ_k and μ_{k+1} in statements (a) *and* (b)*.)*

Proof. Part (a) follows immediately from the line search rule (4.4).

(b) Let $d^k = -\nabla\Psi(x^k)$ and assume (4.6) is not satisfied. From (4.5), we have $\|\Phi(x^k)\| - \|\Phi(x^{k+1})\| =: c_k > 0$. Therefore, together with Corollary 2.4, we get

$$\begin{aligned}
\|\Phi_{\mu_{k+1}}(x^{k+1})\| &\leq \|\Phi_{\mu_{k+1}}(x^{k+1}) - \Phi(x^{k+1})\| + \|\Phi(x^{k+1})\| \\
&\leq \kappa\sqrt{\mu_{k+1}} + \|\Phi(x^k)\| - c_k \\
&\leq \|\Phi_{\mu_{k+1}}(x^k)\| + \|\Phi(x^k) - \Phi_{\mu_{k+1}}(x^k)\| + \kappa\sqrt{\mu_{k+1}} - c_k \\
&\leq \|\Phi_{\mu_{k+1}}(x^k)\| + 2\kappa\sqrt{\mu_{k+1}} - c_k \\
&\leq \|\Phi_{\mu_{k+1}}(x^k)\|,
\end{aligned}$$

where the last inequality follows from the special choice of μ_{k+1} made in (4.8). □

As a simple consequence of this result, we obtain the following corollary.

COROLLARY 5.3. *If $k \notin K$, then*

$$\|\Phi_{\mu_k}(x^k)\| \leq \|\Phi_{\mu_k}(x^{k-1})\|.$$

Proof. First assume that $k \notin K$ and the updating rule (4.8) is active (i.e., d^{k-1} is a gradient step). Taking into account the shift of indices in the definition of the set K, we directly obtain from Lemma 5.2 (b)

$$\|\Phi_{\mu_k}(x^k)\| \leq \|\Phi_{\mu_k}(x^{k-1})\|.$$

On the other hand, if (4.8) is not active (i.e., d^{k-1} is a Newton direction), then we have $\mu_k = \mu_{k-1}$ and therefore

$$\|\Phi_{\mu_k}(x^k)\| = \|\Phi_{\mu_{k-1}}(x^k)\| < \|\Phi_{\mu_{k-1}}(x^{k-1})\| = \|\Phi_{\mu_k}(x^{k-1})\|$$

by Lemma 5.2 (a). This completes the proof. □

Using these preliminary results, we are now able to show that the iterates x^k stay in a certain level set. To this end, we first note that, in all standard descent methods, the iterates would stay in the level set belonging to the level $\Psi(x^0)$ of Ψ at the initial iterate x^0. This is no longer true for our algorithm basically because we minimize different merit functions in our line search rules, namely, Ψ when using a gradient step and Ψ_{μ_k} when using a Newton step. (Note that a decrease in one merit function does not necessarily imply a decrease in the other.) Fortunately, our following result shows that the possible increase in Ψ can't be too dramatic. In fact, this result shows that all iterates x^k stay in a level set whose level can be made arbitrarily close to the level $\Psi(x^0)$.

PROPOSITION 5.4. *The sequence* $\{x^k\}$ *generated by Algorithm* 4.1 *remains in the level set*

$$\mathcal{L}_0 := \{x \in \mathbb{R}^n \mid \Psi(x) \le (1+\alpha)^2\Psi(x^0)\}. \tag{5.1}$$

Proof. We define the following two index sets:

$$K_1 := \left\{ k \in K \,\middle|\, \eta\beta_{k-1} \ge \frac{1}{\alpha}\|\Phi(x^k) - \Phi_{\mu_{k-1}}(x^k)\| \right\} \tag{5.2}$$

and

$$K_2 := \left\{ k \in K \,\middle|\, \eta\beta_{k-1} < \frac{1}{\alpha}\|\Phi(x^k) - \Phi_{\mu_{k-1}}(x^k)\| \right\}. \tag{5.3}$$

Then $K = \{0\} \cup K_1 \cup K_2$, where K is defined in (4.9). Assume that K consists of $k_0 = 0 < k_1 < k_2 < \cdots$ (notice that K is not necessarily infinite). Let $k \in \mathbb{N}$ be an arbitrary but fixed index and k_j the largest number in K such that $k_j \le k$. Then we have

$$\mu_k \le \mu_{k_j} \quad \text{and} \quad \beta_k = \beta_{k_j}$$

in view of the updating rules in step (S.4) of Algorithm 4.1. We divide the proof into three parts.

(a) In this part, we show that the following inequality holds:

$$\|\Phi(x^k)\| \le \beta_{k_j} + 2\kappa\sqrt{\mu_{k_j}}. \tag{5.4}$$

If $k_j = k$, this inequality is obviously true since $\beta_{k_j} = \|\Phi(x^{k_j})\|$ in this case. Hence we assume that $k_j < k$ in the following. From Corollary 5.3, we obtain

$$\|\Phi_{\mu_l}(x^l)\| \le \|\Phi_{\mu_l}(x^{l-1})\|$$

for all $k_j < l < k_{j+1}$. Since $k < k_{j+1}$, this implies

$$\|\Phi_{\mu_l}(x^l)\| \le \|\Phi_{\mu_l}(x^{l-1})\|$$

for all $k_j < l \le k$ or, equivalently,

$$\|\Phi_{\mu_{l+1}}(x^{l+1})\| \le \|\Phi_{\mu_{l+1}}(x^l)\|$$

for all l such that $k_j \le l \le k-1$. Then, by Corollary 2.4, we get for all l such that $k_j \le l \le k-1$:

$$\begin{aligned} \|\Phi_{\mu_{l+1}}(x^{l+1})\| + \kappa\sqrt{\mu_{l+1}} &\le \|\Phi_{\mu_{l+1}}(x^l)\| + \kappa\sqrt{\mu_{l+1}} \\ &\le \|\Phi_{\mu_l}(x^l)\| + \|\Phi_{\mu_{l+1}}(x^l) - \Phi_{\mu_l}(x^l)\| + \kappa\sqrt{\mu_{l+1}} \\ &\le \|\Phi_{\mu_l}(x^l)\| + \kappa(\sqrt{\mu_l} - \sqrt{\mu_{l+1}}) + \kappa\sqrt{\mu_{l+1}} \\ &= \|\Phi_{\mu_l}(x^l)\| + \kappa\sqrt{\mu_l}. \end{aligned} \tag{5.5}$$

This inequality, together with Corollary 2.4, gives

$$\begin{aligned} \|\Phi(x^k)\| &\le \|\Phi_{\mu_k}(x^k)\| + \|\Phi(x^k) - \Phi_{\mu_k}(x^k)\| \\ &\le \|\Phi_{\mu_k}(x^k)\| + \kappa\sqrt{\mu_k} \\ &\le \|\Phi_{\mu_{k-1}}(x^{k-1})\| + \kappa\sqrt{\mu_{k-1}} \\ &\;\;\vdots \\ &\le \|\Phi_{\mu_{k_j}}(x^{k_j})\| + \kappa\sqrt{\mu_{k_j}} \\ &\le \|\Phi(x^{k_j})\| + \|\Phi_{\mu_{k_j}}(x^{k_j}) - \Phi(x^{k_j})\| + \kappa\sqrt{\mu_{k_j}} \\ &\le \|\Phi(x^{k_j})\| + \kappa\sqrt{\mu_{k_j}} + \kappa\sqrt{\mu_{k_j}} \\ &= \beta_{k_j} + 2\kappa\sqrt{\mu_{k_j}}, \end{aligned} \tag{5.6}$$

where the dots indicate the repeated use of (5.5). This shows that (5.4) holds for arbitrary $k \in \mathbb{N}$.

(b) In this part, we show that

$$\sqrt{\mu_{k_j}} \le \frac{1}{2^{j+1}}\frac{\alpha}{\kappa}\|\Phi(x^0)\|$$

and

$$\beta_{k_j} \le r^j\|\Phi(x^0)\|,$$

where

$$r := \max\left\{\frac{1}{2}, \eta\right\}.$$

Indeed, for $j = 0$, we have $k_0 = 0$ and therefore

$$\sqrt{\mu_{k_0}} = \sqrt{\mu_0} = \frac{\alpha}{2\kappa}\|\Phi(x^0)\|$$

and

$$\beta_{k_0} = \beta_0 = r^0\|\Phi(x^0)\|$$

by the definitions of μ_0 and β_0. For $j \ge 1$, step (S.4) of Algorithm 4.1 shows that

$$\beta_{k_j} \le \eta\beta_{k_j-1} = \eta\beta_{k_{j-1}} \le r\beta_{k_{j-1}} \quad \text{for } k_j \in K_1,$$

and, using Corollary 2.4,

$$\beta_{k_j} \le \frac{1}{\alpha}\|\Phi(x^{k_j}) - \Phi_{\mu_{k_j-1}}(x^{k_j})\| \le \frac{\kappa}{\alpha}\sqrt{\mu_{k_j-1}} \le \frac{\kappa}{\alpha}\sqrt{\mu_{k_{j-1}}} \le \frac{1}{2}\beta_{k_{j-1}} \le r\beta_{k_{j-1}} \quad \text{for } k_j \in K_2.$$

Similarly, we obtain

$$\mu_{k_j} \le \frac{1}{4}\mu_{k_j-1} \le \frac{1}{4}\mu_{k_{j-1}}.$$

From the definitions of μ_0 and β_0, we thus have

$$\sqrt{\mu_{k_j}} \le \frac{1}{2^j}\sqrt{\mu_0} = \frac{1}{2^{j+1}}\frac{\alpha}{\kappa}\|\Phi(x^0)\| \tag{5.7}$$

and

$$\beta_{k_j} \le r^j \beta_0 = r^j \|\Phi(x^0)\|. \tag{5.8}$$

This completes the proof of part (b).

(c) In this part, we now want to verify the statement of our proposition. Using parts (a) and (b), we obtain

$$\begin{aligned} \|\Phi(x^k)\| &\le \beta_{k_j} + 2\kappa\sqrt{\mu_{k_j}} \\ &\le r^j \|\Phi(x^0)\| + \frac{\alpha}{2^j}\|\Phi(x^0)\| \\ &\le r^j(1+\alpha)\|\Phi(x^0)\| \\ &\le (1+\alpha)\|\Phi(x^0)\|. \end{aligned} \tag{5.9}$$

Hence $x^k \in \mathcal{L}_0$. □

Note that the level set $\mathcal{L}_0$ as defined in Proposition 5.4 is known to be compact if F is a uniform P-function or, more generally, an R_0-function [19].

REMARK 5.5. *We explicitly point out that the proof of Proposition* 5.4 *showed that the following inequality holds for all $k \in \mathbb{N}$:*

$$\|\Phi(x^k)\| \le r^j(1+\alpha)\|\Phi(x^0)\|,$$

where, if $K = \{k_0, k_1, k_2, \dots\}$ with $k_0 = 0$, the index $j \in \mathbb{N}$ is defined to be the largest integer $k_j \in K$ such that $k_j \le k$.

As an immediate consequence of Remark 5.5, we obtain the following proposition.

PROPOSITION 5.6. *Let $\{x^k\}$ be a sequence generated by Algorithm* 4.1 *and assume that the index set K is infinite. Then each accumulation point of the sequence $\{x^k\}$ is a solution of NCP(F).*

Proof. Let x^* be an accumulation point of the sequence $\{x^k\}$, and let $\{x^k\}_L$ be a subsequence converging to x^*. Since K is infinite by assumption, we obtain from Remark 5.5

$$\|\Phi(x^*)\| = \lim_{k\in L} \|\Phi(x^k)\| \le \lim_{j\to\infty} r^j(1+\alpha)\|\Phi(x^0)\| = 0,$$

where the exponent $j \in \mathbb{N}$ is defined as in Remark 5.5. Hence x^* is a solution of NCP(F). □

In our next result, we consider the situation in which x^* is a limit point of a subsequence which consists of gradient steps only.

PROPOSITION 5.7. *Let $\{x^k\}$ be a sequence generated by Algorithm* 4.1 *and let $\{x^k\}_L$ be a subsequence converging to a point $x^* \in \mathbb{R}^n$. If $d^k = -\nabla\Psi(x^k)$ for all $k \in L$, then x^* is a stationary point of Ψ.*

Proof. If the index set K is infinite, the accumulation point x^* is a solution of NCP(F) by Proposition 5.6. Hence x^* is a global minimum and therefore a stationary point of Ψ.

So let K be finite. Then, without loss of generality, we can assume that $K \cap L = \emptyset$ so that the updating rule (4.8) is active for all $k \in L$. This, in particular, implies that $\{\mu_k\} \to 0$.

Let $\hat{k}$ be the largest number in K (which exists since K is finite). Then we obtain from the updating rules in step (S.4) of Algorithm 4.1 for all $k > \hat{k}$:

$$\mu_k \le \mu_{\hat{k}}, \quad \beta_k = \beta_{\hat{k}} = \|\Phi(x^{\hat{k}})\|, \tag{5.10}$$

$$\|\Phi(x^k)\| > \eta\beta_{k-1} = \eta\|\Phi(x^{\hat{k}})\| > 0 \tag{5.11}$$

and

$$\alpha\|\Phi(x^k)\| > \|\Phi(x^k) - \Phi_{\mu_{k-1}}(x^k)\|. \tag{5.12}$$

From (5.11), we get

$$\Psi(x^k) > \eta^2\Psi(x^{\hat{k}}) > 0 \tag{5.13}$$

for all $k > \hat{k}$.

The proof is by contradiction: Assume that $\nabla\Psi(x^*) \neq 0$. Our first aim is to show that $\liminf_{k\in L} t_k = 0$. Suppose that $\liminf_{k\in L} t_k = t_* > 0$. Since $d^k = -\nabla\Psi(x^k)$ for all $k \in L$, we obtain from the Armijo rule (4.5)

$$\Psi(x^{k+1}) - \Psi(x^k) \leq -\sigma t_k\|\nabla\Psi(x^k)\|^2 \leq -\frac{c}{2} \tag{5.14}$$

for all $k \in L$ sufficiently large, where $c := \sigma t_*\|\nabla\Psi(x^*)\|^2 > 0$. Since $\{\mu_k\} \to 0$, Corollary 2.4 shows that

$$|\Psi_{\mu_k}(x^{k+1}) - \Psi(x^{k+1})| \leq \frac{c}{4} \quad \text{and} \quad |\Psi_{\mu_k}(x^k) - \Psi(x^k)| \leq \frac{c}{4}$$

for all $k \in \mathbb{N}$ sufficiently large. Using $\{\mu_k\} \to 0$ once again and taking into account that the sequence $\{\|\Phi(x^k)\|\}$ is bounded by Proposition 5.4, we also have

$$2\kappa\sqrt{\mu_k}\|\Phi(x^k)\| + 2\kappa^2\mu_k \leq \frac{c}{4} \tag{5.15}$$

for all $k \in \mathbb{N}$ large enough. Let L consist of $l_0, l_1, l_2, \ldots$. Then, for all l_j sufficiently large, we obtain in a similar way as in the proof of Proposition 5.4 (see (5.6) and recall that K is finite)

$$\begin{array}{rcl} \Psi(x^{l_{j+1}}) & = & \frac{1}{2}\|\Phi(x^{l_{j+1}})\|^2 \\ & \leq & \frac{1}{2}\left(\|\Phi(x^{l_j+1})\| + 2\kappa\sqrt{\mu_{l_j+1}}\right)^2 \\ & = & \Psi(x^{l_j+1}) + 2\kappa\sqrt{\mu_{l_j+1}}\|\Phi(x^{l_j+1})\| + 2\kappa^2\mu_{l_j+1} \\ & \leq & \Psi(x^{l_j+1}) + \frac{c}{4}, \end{array} \tag{5.16}$$

where the last inequality follows from (5.15). Using (5.14) and (5.16), we obtain

$$\Psi(x^{l_{j+1}}) - \Psi(x^{l_j}) = \underbrace{\Psi(x^{l_{j+1}}) - \Psi(x^{l_j+1})}_{\leq \frac{c}{4}} + \underbrace{\Psi(x^{l_j+1}) - \Psi(x^{l_j})}_{\leq -\frac{c}{2}} \leq -\frac{c}{4}$$

for all l_j large enough. Hence $\{\Psi(x^{l_j})\} \to -\infty$ for $j \to \infty$, but this contradicts the fact that $\Psi(x) \geq 0$ for all $x \in \mathbb{R}^n$. Hence we have $\liminf_{k\in L} t_k = 0$.

Subsequencing if necessary, we can assume that $\lim_{k\in L} t_k = 0$. We now want to derive a contradiction to our assumption that $\nabla\Psi(x^*) \neq 0$. Since $\lim_{k\in L} t_k = 0$, the full stepsize is never accepted for all $k \in L$ sufficiently large. Hence we obtain from the Armijo rule (4.5)

$$\Psi(x^k + \lambda^{m_k-1}d^k) > \Psi(x^k) - \sigma\lambda^{m_k-1}\|d^k\|^2$$

or, equivalently,

$$\frac{\Psi(x^k + \lambda^{m_k-1}d^k) - \Psi(x^k)}{\lambda^{m_k-1}} > -\sigma\|d^k\|^2. \tag{5.17}$$

By taking the limit $k \to \infty$ on L, we obtain from (5.17), the continuous differentiability of Ψ, $d^k = -\nabla\Psi(x^k)$ for all $k \in L$, and the fact that $\lambda^{m_k-1} \to 0$ for $k \to_L \infty$

$$-\nabla\Psi(x^*)^T\nabla\Psi(x^*) \geq -\sigma\nabla\Psi(x^*)^T\nabla\Psi(x^*).$$

This yields $1 \leq \sigma$, a contradiction to our choice of the parameter σ. Hence we must have $\nabla\Psi(x^*) = 0$, and this completes the proof of Proposition 5.7. □

We are now able to prove the main global convergence result for Algorithm 4.1.

THEOREM 5.8. *Let $\{x^k\}$ be a sequence generated by Algorithm* 4.1. *Then each accumulation point of the sequence $\{x^k\}$ is a stationary point of Ψ.*

Proof. If K is infinite, the conclusion follows immediately from Proposition 5.6. Hence we can assume that K contains only finitely many indices.

Similar to the proof of Proposition 5.7, we denote by $\hat{k}$ the largest index in K. Then (5.10), (5.11), (5.12), and (5.13) hold for all $k > \hat{k}$.

Let x^* be an accumulation point of the sequence $\{x^k\}$, and let $\{x^k\}_L$ be a subsequence converging to x^*. If $d^k = -\nabla\Psi(x^k)$ for infinitely many $k \in L$, then x^* is a stationary point of Ψ by Proposition 5.7. Hence we can assume without loss of generality that d^k is the Newton direction computed as a solution of the linear system (4.1) for all $k \in L$, so that

$$\|\Phi(x^k)\| = \|\Phi'_{\mu_k}(x^k)d^k\| \leq \|\Phi'_{\mu_k}(x^k)\|\,\|d^k\| \tag{5.18}$$

holds for all $k \in L$. Since K is finite, we can further assume without loss of generality that $k \notin K$ for all $k \in L$; i.e., neither the updating rule (4.7) nor the updating rule (4.8) is active for $k \in L$.

The proof is by contradiction: Assume that x^* is not a stationary point of Ψ. Since the sequence $\{\mu_k\}$ is monotonically decreasing and bounded from below, it converges to some $\mu_* \geq 0$. If $\mu_* > 0$, then it follows from the updating rules of step (S.4) in Algorithm 4.1 that μ_k is actually constant for all k sufficiently large.

The remaining part of this proof is divided into three steps.

(a) We first show that there exist positive constants m and M such that

$$0 < m \leq \|d^k\| \leq M \quad \text{for all } k \in L. \tag{5.19}$$

In fact, if $\{\|d^k\|\}_{\tilde{L}} \to 0$ on a subset $\tilde{L} \subseteq L$, we would have from (5.18) that $\{\|\Phi(x^k)\|\}_{\tilde{L}} \to 0$ because the sequence $\{\Phi'_{\mu_k}(x^k)\}_{\tilde{L}}$ is obviously bounded on the convergent sequence $\{x^k\}_{\tilde{L}}$. But then the continuity of Φ would imply that $\Phi(x^*) = 0$, so that K would be infinite by Lemma 5.1. This, however, would contradict our assumption that K is finite.

On the other hand, we have from (4.2) for all $k \in L$

$$-\|\Phi'_{\mu_k}(x^k)^T\Phi(x^k)\|\,\|d^k\| \leq \Phi(x^k)^T\Phi'_{\mu_k}(x^k)d^k \leq -\rho\|d^k\|^p. \tag{5.20}$$

Since $\{\Phi'_{\mu_k}(x^k)^T\Phi(x^k)\}_L$ is convergent (either by Lemma 2.6 or because μ_k is eventually constant) and therefore bounded, there exists a constant $C > 0$ such that

$$\|\Phi'_{\mu_k}(x^k)^T\Phi(x^k)\| \leq C$$

for all $k \in L$. With (5.20), we have

$$\rho\|d^k\|^p \leq \|\Phi'_{\mu_k}(x^k)^T\Phi(x^k)\|\|d^k\| \leq C\|d^k\|$$

for all $k \in L$. Since $p > 1$, this shows that $\{\|d^k\|\}_L$ is bounded. This completes the proof of part (a).

(b) We now show that $\liminf_{k \in L} t_k = 0$. Suppose that $\liminf_{k \in L} t_k =: t_* > 0$. Then from (5.13) and the line search rule (4.4), we have for all $k \in L$ sufficiently large

$$\Psi_{\mu_k}(x^{k+1}) - \Psi_{\mu_k}(x^k) \leq -2\sigma t_k \Psi(x^k) \leq -\sigma t_* \eta^2 \Psi(x^{\hat{k}}) < 0. \tag{5.21}$$

We define $c := \sigma t_* \eta^2 \Psi(x^{\hat{k}}) > 0$ and consider two cases.

Case 1. $\{\mu_k\} \to \mu_* > 0$.

Then we have $\mu_k = \mu_*$ constant for $k \in \mathbb{N}$ sufficiently large. Hence we obtain from (5.21) for all $k \in L$ large enough

$$\Psi_{\mu_*}(x^{k+1}) - \Psi_{\mu_*}(x^k) = \Psi_{\mu_k}(x^{k+1}) - \Psi_{\mu_k}(x^k) \leq -c. \tag{5.22}$$

Since μ_k is eventually constant, the updating rule (4.8) excludes the existence of gradient steps for $k \in \mathbb{N}$ sufficiently large. Hence, if we assume that L consists of $l_0, l_1, l_2, \ldots$, we obtain from Lemma 5.2 (a) for all l_j sufficiently large

$$\Psi_{\mu_*}(x^{l_{j+1}}) - \Psi_{\mu_*}(x^{l_j}) \leq \Psi_{\mu_*}(x^{l_j+1}) - \Psi_{\mu_*}(x^{l_j}) \leq -c.$$

This implies

$$\Psi_{\mu_*}(x^{l_j}) \to -\infty$$

for $j \to \infty$, a contradiction to $\Psi_{\mu_*}(x) \geq 0$ for all $x \in \mathbb{R}^n$.

Case 2. $\{\mu_k\} \to 0$.

Then we obtain from Corollary 2.4 that

$$|\Psi_{\mu_k}(x^{k+1}) - \Psi(x^{k+1})| \leq \frac{c}{4} \quad \text{and} \quad |\Psi_{\mu_k}(x^k) - \Psi(x^k)| \leq \frac{c}{4} \tag{5.23}$$

for all $k \in \mathbb{N}$ sufficiently large. Again, let the sequence L consist of $l_0, l_1, l_2, \ldots$. Then the following inequality holds for all l_j large enough:

$$\begin{aligned} \Psi(x^{l_j+1}) - \Psi(x^{l_j}) &= -(\Psi_{\mu_{l_j}}(x^{l_j+1}) - \Psi(x^{l_j+1})) + (\Psi_{\mu_{l_j}}(x^{l_j}) - \Psi(x^{l_j})) \\ &\quad + \Psi_{\mu_{l_j}}(x^{l_j+1}) - \Psi_{\mu_{l_j}}(x^{l_j}) \\ &\leq \underbrace{|\Psi_{\mu_{l_j}}(x^{l_j+1}) - \Psi(x^{l_j+1})|}_{\leq \frac{c}{4} \text{ by } (5.23)} + \underbrace{|\Psi_{\mu_{l_j}}(x^{l_j}) - \Psi(x^{l_j})|}_{\leq \frac{c}{4} \text{ by } (5.23)} \\ &\quad + \underbrace{\Psi_{\mu_{l_j}}(x^{l_j+1}) - \Psi_{\mu_{l_j}}(x^{l_j})}_{\leq -c \text{ by } (5.21)} \\ &\leq -\frac{c}{2}. \end{aligned} \tag{5.24}$$

The remaining part of the proof for Case 2 is now similar to the one for Proposition 5.7. In particular, for l_j large enough, we can prove the following inequality in essentially the same way as in the proof of Proposition 5.7 (see (5.16) and recall that K is finite):

$$\Psi(x^{l_{j+1}}) \leq \Psi(x^{l_j+1}) + \frac{c}{4}. \tag{5.25}$$

Combining (5.24) and (5.25), we obtain

$$\Psi(x^{l_{j+1}}) - \Psi(x^{l_j}) = \Psi(x^{l_{j+1}}) - \Psi(x^{l_j+1}) + \Psi(x^{l_j+1}) - \Psi(x^{l_j}) \leq \frac{c}{4} - \frac{c}{2} = -\frac{c}{4}.$$

This implies $\Psi(x^{l_j}) \to -\infty$ for $j \to \infty$, contradicting the fact that $\Psi(x) \geq 0$ for all $x \in \mathbb{R}^n$.

Since both Case 1 and Case 2 lead to a contradiction, the proof of part (b) is also completed.

(c) We now turn back to the main part of our proof; i.e., we will now derive a contradiction to our assumption that $\nabla\Psi(x^*) \neq 0$.

Because of part (b), we have $\liminf_{k\in L} t_k = 0$. Let L_0 be a subsequence of L such that $\{t_k\}_{L_0}$ converges to 0. Then $m_k > 0$ for all $k \in L_0$ sufficiently large, where $m_k \in \mathbb{N}$ denotes the exponent from the line search rule (4.4). By this line search rule, we therefore have

$$-2\sigma\lambda^{m_k-1}\Psi(x^k) < \Psi_{\mu_k}(x^k + \lambda^{m_k-1}d^k) - \Psi_{\mu_k}(x^k)$$

for all $k \in L_0$ large enough. Dividing both sides by λ^{m_k-1}, we obtain

$$-2\sigma\Psi(x^k) < \frac{\Psi_{\mu_k}(x^k + \lambda^{m_k-1}d^k) - \Psi_{\mu_k}(x^k)}{\lambda^{m_k-1}}.$$

Let μ_* be the limit of $\{\mu_k\}$, and if $\mu_* = 0$, we write $\nabla\Psi_{\mu_*}(x^*)$ for the gradient of the unperturbed function Ψ at the limit point x^*. By (5.19) we can assume, subsequencing if necessary, that $\{d^k\}_{L_0} \to d^* \neq 0$, so that, passing to the limit, we get

$$-2\sigma\Psi(x^*) \leq \nabla\Psi_{\mu_*}(x^*)^T d^*. \tag{5.26}$$

For $\mu_* = 0$ this follows from Lemma 2.7, and if $\mu_* > 0$, then $\mu_k = \mu_*$ for sufficiently large k, so that (5.26) follows from the mean value theorem.

Using (4.1), (5.12), and Corollary 2.4, we further have for $k \in L_0$:

$$\begin{aligned}
\nabla\Psi_{\mu_k}(x^k)^T d^k &= -\Phi(x^k)^T\Phi_{\mu_k}(x^k) \\
&= -2\Psi(x^k) + \Phi(x^k)^T(\Phi(x^k) - \Phi_{\mu_k}(x^k)) \\
&\leq -2\Psi(x^k) + \|\Phi(x^k)\|\,\|\Phi(x^k) - \Phi_{\mu_{k-1}}(x^k)\| \\
&\quad + \|\Phi(x^k)\|\,\|\Phi_{\mu_{k-1}}(x^k) - \Phi_{\mu_k}(x^k)\| \\
&\leq -2\Psi(x^k) + 2\alpha\Psi(x^k) + \kappa\|\Phi(x^k)\|(\sqrt{\mu_{k-1}} - \sqrt{\mu_k}) \\
&= -2(1-\alpha)\Psi(x^k) + \kappa\|\Phi(x^k)\|(\sqrt{\mu_{k-1}} - \sqrt{\mu_k}).
\end{aligned} \tag{5.27}$$

By taking the limit $k \to_{L_0} \infty$ in (5.27), we obtain from (5.26) (and Lemma 2.6 if $\mu_* = 0$)

$$-2\sigma\Psi(x^*) \leq \nabla\Psi_{\mu_*}(x^*)^T d^* \leq -2(1-\alpha)\Psi(x^*), \tag{5.28}$$

since $\{\|\Phi(x^k)\|\}$ is bounded (by Proposition 5.4), and $(\sqrt{\mu_{k-1}} - \sqrt{\mu_k}) \to 0$ (because $\{\mu_k\}$ converges). We have $\Psi(x^*) > 0$, because otherwise K would be infinite. Therefore (5.28) gives $\sigma \geq (1-\alpha)$, which is a contradiction to $\sigma < \frac{1}{2}(1-\alpha)$. This, finally, completes the proof of Theorem 5.8. □

Note that Theorem 5.8 is a subsequential convergence result to stationary points of Ψ only. However, it is well known that such a stationary point x^* is already a solution of NCP(F) if, e.g., the Jacobian $F'(x^*)$ is a P_0-matrix; see [16, 13]. Moreover, Proposition 5.6 provides another sufficient condition for an accumulation point to be a solution of the complementarity problem. In particular, Algorithm 4.1 is guaranteed to converge to a solution of the nonlinear complementarity problem if F is a P_0- and R_0-function.

6. Local convergence.

In this section, we want to show that Algorithm 4.1 is locally Q-superlinearly/Q-quadratically convergent under certain assumptions. As a first step in this direction, we show that the whole sequence $\{x^k\}$ generated by Algorithm 4.1 converges to a unique point x^* if certain conditions hold. The proof of this result is based on the following Proposition by Moré and Sorensen [31]. (Note that their result is fairly general and completely independent of any specific algorithm.)

PROPOSITION 6.1. *Assume that $x^* \in \mathbb{R}^n$ is an isolated accumulation point of a sequence $\{x^k\} \subseteq \mathbb{R}^n$ (not necessarily generated by Algorithm* 4.1*) such that $\{\|x^{k+1} - x^k\|\}_L \to 0$ for any subsequence $\{x^k\}_L$ converging to x^*. Then the whole sequence $\{x^k\}$ converges to x^*.*

Proposition 6.1 enables us to establish the following result.

THEOREM 6.2. *Let $\{x^k\}$ be a sequence generated by Algorithm* 4.1*. If one of the accumulation points of the sequence $\{x^k\}$, let us say x^*, is an isolated solution of NCP(F), then $\{x^k\} \to x^*$.*

Proof. Let x^* be an isolated solution of NCP(F). We want to verify the assumptions of Proposition 6.1. To this end, we first show that x^* is also an isolated accumulation point of the sequence $\{x^k\}$.

Since x^* solves NCP(F), Lemma 5.1 shows that the index set K is infinite and $\{\mu_k\}$ converges to 0. Hence Proposition 5.6 shows that each accumulation point of the sequence $\{x^k\}$ is already a solution of NCP(F). Thus x^* is necessarily an isolated accumulation point of the sequence $\{x^k\}$.

Now let $\{x^k\}_L$ be an arbitrary subsequence of $\{x^k\}$ converging to x^*. From the updating rule in step (S.3) of Algorithm 4.1, we have

$$\|x^{k+1} - x^k\| = \lambda^{m_k} \|d^k\| \le \|d^k\|. \tag{6.1}$$

Therefore it suffices to show that $\{\|d^k\|\}_L \to 0$. Since Ψ is continuously differentiable and since the solution x^* of NCP(F) is, in particular, a stationary point of Ψ, we have

$$\{\nabla\Psi(x^k)\}_L \to \nabla\Psi(x^*) = 0. \tag{6.2}$$

Suppose the sequence $\{d^k\}_L$ contains only a finite number of Newton directions. Then $\{\|d^k\|\}_L \to 0$ follows immediately. Assume therefore that there is a subsequence $\{d^k\}_{L_0}$ of $\{d^k\}_L$ such that d^k is the solution of the linear system (4.1) for all $k \in L_0$.

From (4.2), we obtain

$$\rho\|d^k\|^p \le -(\Phi'_{\mu_k}(x^k)^T\Phi(x^k))^T\, d^k \le \|\Phi'_{\mu_k}(x^k)^T\Phi(x^k)\|\, \|d^k\|$$

for all $k \in L_0$, from which we get

$$\|d^k\| \le \left(\frac{\|\Phi'_{\mu_k}(x^k)^T\Phi(x^k)\|}{\rho}\right)^{\frac{1}{p-1}} \tag{6.3}$$

because $p > 1$. Since $\{\mu_k\} \to 0$, we obtain

$$\lim_{k\to\infty, k\in L_0} \Phi'_{\mu_k}(x^k)^T\Phi(x^k) \to \nabla\Psi(x^*) = 0$$

from Lemma 2.6. Hence the right-hand side of (6.3) converges to 0, so that $\{d^k\}_{L_0} \to 0$. We obviously also have $\{d^k\}_{L\setminus L_0} \to 0$ from (6.2) (if the set $L \setminus L_0$ is infinite). Hence (6.1) shows that

$$\{\|x^{k+1} - x^k\|\}_L \to 0.$$

The assertion now follows from Proposition 6.1. $\square$

REMARK 6.3. *We explicitly point out that, in the proof of Theorem* 6.2, *we have actually shown that if the sequence* $\{x^k\}$ *generated by Algorithm* 4.1 *converges to a solution of NCP*(F), *then* $\{\|d^k\|\} \to 0$. *This fact will be important in the proof of Theorem* 6.6 *below.*

In order to verify that Algorithm 4.1 eventually takes the full stepsize $t_k = 1$, we state the following lemma which was shown by Chen, Qi, and Sun [10, Lemma 3.2].

LEMMA 6.4. *If there exists a scalar*

$$\omega \in \left[\frac{1}{2} - \frac{(1-\alpha-2\sigma)^2}{2(2+\alpha)^2}, \frac{1}{2}\right]$$

such that

$$\Psi(y) \leq \Psi(x^k) - 2\omega\Psi(x^k) \tag{6.4}$$

for some $k \in K$ *and* $y \in \mathbb{R}^n$, *then it holds that*

$$\Psi_{\mu_k}(y) \leq \Psi_{\mu_k}(x^k) - 2\sigma\Psi(x^k), \tag{6.5}$$

where μ_k *is the smoothing parameter in the kth step.*

In the proof of our main local convergence result, we will also utilize the following proposition, which was originally shown by Facchinei and Soares [16]. An alternative proof of this result was given by Kanzow and Qi [29] under slightly different assumptions. Here we restate the result from [29].

PROPOSITION 6.5. *Let* $G: \mathbb{R}^n \to \mathbb{R}^n$ *be locally Lipschitzian and* $x^* \in \mathbb{R}^n$ *with* $G(x^*) = 0$ *such that all elements in* $\partial G(x^*)$ *are nonsingular, and assume that there are two subsequences* $\{x^k\} \subseteq \mathbb{R}^n$ *and* $\{d^k\} \subseteq \mathbb{R}^n$ *with*

$$\lim_{k\to\infty} x^k = x^* \quad \textit{and} \quad \|x^k + d^k - x^*\| = o(\|x^k - x^*\|).$$

Then

$$\|G(x^k + d^k)\| = o(\|G(x^k)\|).$$

Before stating our local convergence result, we recall that a solution x^* of NCP(F) is called *R-regular* if the submatrix $F'(x^*)_{\alpha\alpha}$ is nonsingular and the Schur complement

$$F'(x^*)_{\beta\beta} - F'(x^*)_{\beta\alpha}F'(x^*)_{\alpha\alpha}^{-1}F'(x^*)_{\alpha\beta} \in \mathbb{R}^{|\beta|\times|\beta|}$$

is a P-matrix; see Robinson [38]. Here, we have used the standard index set notation

$$\begin{aligned}\alpha &:= \{i|\, x_i^* > 0 = F_i(x^*)\},\\ \beta &:= \{i|\, x_i^* = 0 = F_i(x^*)\},\\ \gamma &:= \{i|\, x_i^* = 0 < F_i(x^*)\}.\end{aligned}$$

THEOREM 6.6. *Let* $\{x^k\}$ *be a sequence generated by Algorithm* 4.1. *If one of the limit points of the sequence* $\{x^k\}$, *let us say* x^*, *is an R-regular solution of NCP*(F), *then* $\{x^k\} \to x^*$, *and the convergence rate is at least Q-superlinear. If* $F: \mathbb{R}^n \to \mathbb{R}^n$ *is continuously differentiable with a locally Lipschitzian Jacobian, then the convergence rate is Q-quadratic.*

Proof. We first note that the assumed R-regularity of the solution x^* implies that all elements of the C-subdifferential $\partial_C \Phi(x^*)$ are nonsingular; see [16]. Hence Proposition 2.5 in [33], together with Proposition 2.2 of this paper, shows that x^* is an isolated solution of $\Phi(x) = 0$ and therefore also of NCP(F). Hence, by Theorem 6.2, the whole sequence $\{x^k\}$ converges to x^*. Let K be again the set defined by (4.9), which, by Lemma 5.1, is infinite since the sequence $\{x^k\}$ converges to a solution of NCP(F). In particular, we have $\{x^k\}_K \to x^*$.

We now divide the proof into four steps.

(a) In this part, we show that, for all $k \in K$ sufficiently large, the matrix $\Phi'_{\mu_k}(x^k)$ is nonsingular and satisfies the inequality

$$\|\Phi'_{\mu_k}(x^k)^{-1}\| \leq 2c$$

for a certain constant $c > 0$.

Since $\{x^k\}$ converges to x^*, the assumed R-regularity together with the upper semicontinuity of the C-subdifferential implies that, for all $k \in \mathbb{N}$ sufficiently large, all matrices $V_k \in \partial_C \Phi(x^k)$ are nonsingular with $\|V_k^{-1}\| \leq c$ for some constant $c > 0$. We now want to show that the same is true for $\Phi'_{\mu_k}(x^k)$. Let $H_k \in \partial_C \Phi(x^k)$ such that

$$\operatorname{dist}_F(\Phi'_{\mu_k}(x^k), \partial_C \Phi(x^k)) = \|\Phi'_{\mu_k}(x^k) - H_k\|_F$$

(note that such an element exists since the set $\partial_C \Phi(x^k)$ is nonempty and compact). With (4.11) we have

$$\|H_k - \Phi'_{\mu_k}(x^k)\| \leq \|H_k - \Phi'_{\mu_k}(x^k)\|_F \leq \gamma\beta_k \tag{6.6}$$

for all $k \in K$. Hence it follows that

$$\begin{aligned} \|I - H_k^{-1}\Phi'_{\mu_k}(x^k)\| &= \|H_k^{-1}(H_k - \Phi'_{\mu_k}(x^k))\| \\ &\leq \|H_k^{-1}\| \, \|H_k - \Phi'_{\mu_k}(x^k)\| \\ &\leq \gamma\beta_k c. \end{aligned} \tag{6.7}$$

Since K is infinite, we have $\beta_k \to 0$ in view of the updating rules in step (S.4) of Algorithm 4.1. Therefore, for $k \in K$ large enough such that $\beta_k \leq \frac{1}{2\gamma c}$, we have

$$\|I - H_k^{-1}\Phi'_{\mu_k}(x^k)\| \leq \frac{1}{2}.$$

From the perturbation lemma [14, Theorem 3.1.4], we obtain that $\Phi'_{\mu_k}(x^k)$ is nonsingular for all $k \in K$ large enough with

$$\|\Phi'_{\mu_k}(x^k)^{-1}\| \leq 2\|H_k^{-1}\| \leq 2c. \tag{6.8}$$

Hence system (4.1) admits a solution for all $k \in K$ sufficiently large, and the proof of part (a) is completed.

(b) We next want to show that, for all $k \in K$ sufficiently large, the solution d^k of the linear system (4.1) satisfies the descent condition (4.2).

To this end, we first note that the linear system (4.1) has a unique solution for all $k \in K$ sufficiently large by part (a). We now show that these d^k satisfy the inequality

$$\Phi(x^k)^T \Phi'_{\mu_k}(x^k) d^k \leq -\rho_1 \|d^k\|^2 \tag{6.9}$$

for a certain positive constant ρ_1. Indeed, this follows from the fact that

$$\|d^k\| \leq \|\Phi'_{\mu_k}(x^k)^{-1}\| \, \|\Phi(x^k)\|$$

by (4.1), so that (6.8) implies

$$\Phi(x^k)^T \Phi'_{\mu_k}(x^k) d^k = -\|\Phi(x^k)\|^2 \leq -\frac{\|d^k\|^2}{4c^2} \tag{6.10}$$

for all $k \in K$ large enough. Hence (6.9) follows from (6.10) by taking $\rho_1 = 1/(4c^2)$. Since $\{\|d^k\|\} \to 0$ by Remark 6.3, it is now easy to see that (6.9) eventually implies (4.2) for any $\rho > 0$ und $p > 2$. Hence, for all $k \in K$ sufficiently large, the search direction d^k is always given by (4.1).

(c) In view of parts (a) and (b), the search direction d^k is given by (4.1) for all $k \in K$ large enough. In this step, we want to show that there is an index $\bar{k} \in K$ such that if $k \in K$ is any index with $k \geq \bar{k}$, then the index $k+1$ also belongs to the set K and $x^{k+1} = x^k + d^k$. Repeating this argument, it then follows that eventually all iterations k belong to the set K, and that the full step $t_k = 1$ is always accepted.

In order to prove this statement, we recall from part (a) that there is a constant $c > 0$ such that $\|\Phi'_{\mu_k}(x^k)^{-1}\| \leq 2c$ for all $k \in K$ sufficiently large. From Algorithm 4.1 and (6.6), we therefore obtain for all $k \in K$ large enough:

$$\begin{aligned}
&(6.11)\\
&\|x^k + d^k - x^*\|\\
&\quad = \|x^k - x^* - \Phi'_{\mu_k}(x^k)^{-1}\Phi(x^k)\|\\
&\quad = \|\Phi'_{\mu_k}(x^k)^{-1}(\Phi'_{\mu_k}(x^k)(x^k - x^*) - \Phi(x^k) + \Phi(x^*))\|\\
&\quad \leq \|\Phi'_{\mu_k}(x^k)^{-1}\| \left(\|(\Phi'_{\mu_k}(x^k) - H_k)(x^k - x^*)\| + \|H_k(x^k - x^*) - \Phi(x^k) + \Phi(x^*)\|\right)\\
&\quad \leq 2c(\gamma\beta_k\|x^k - x^*\| + \|H_k(x^k - x^*) - \Phi(x^k) + \Phi(x^*)\|),
\end{aligned}$$

where, again, $H_k \in \partial_C \Phi(x^k)$ is chosen in such a way that

$$\mathrm{dist}_F(\Phi'_{\mu_k}(x^k), \partial_C \Phi(x^k)) = \|\Phi'_{\mu_k}(x^k) - H_k\|_F;$$

see part (a) of this proof. Using Proposition 2.2 (a) and taking into account that $\beta_k \to 0$, we have

$$\|x^k + d^k - x^*\| = o(\|x^k - x^*\|) \quad \text{for } k \to \infty, \ k \in K. \tag{6.12}$$

Hence (6.12) and Proposition 6.5 show that

$$\|\Phi(x^k + d^k)\| = o(\|\Phi(x^k)\|) \quad \text{for } k \to \infty, \ k \in K. \tag{6.13}$$

Let $\omega := \max\left\{\frac{1}{2} - \frac{(1-\alpha-2\sigma)^2}{2(2+\alpha)^2}, \frac{1-\eta^2}{2}\right\}$. Then (6.13) implies that there exists an index $\bar{k} \in K$ such that

$$\Psi(x^k + d^k) \leq \Psi(x^k) - 2\omega\Psi(x^k) \tag{6.14}$$

for all $k \in K$ with $k \geq \bar{k}$. Hence, by Lemma 6.4, we therefore have

$$\Psi_{\mu_k}(x^k + d^k) \leq \Psi_{\mu_k}(x^k) - 2\sigma\Psi(x^k) \tag{6.15}$$

for all $k \in K$ with $k \geq \bar{k}$. Hence the full stepsize of 1 will eventually be accepted for all $k \geq \bar{k}$, $k \in K$. In particular, $x^{\bar{k}+1} = x^{\bar{k}} + d^{\bar{k}}$, and from (6.14) and the definition of ω, we obtain

$$\|\Phi(x^{\bar{k}+1})\| \leq \sqrt{1-2\omega}\|\Phi(x^{\bar{k}})\| \leq \eta\|\Phi(x^{\bar{k}})\| = \eta\beta_{\bar{k}},$$

which implies that $\bar{k}+1 \in K$; cf. (4.9). Repeating the above process, we may prove that for all $k \geq \bar{k}$, we have

$$k \in K$$

and

$$x^{k+1} = x^k + d^k.$$

This completes the proof of part (c).

(d) We now turn to the final part of the proof where we want to verify the Q-superlinear/Q-quadratic rate of convergence. Since $k \in K$ and $t_k = 1$ for all $k \in \mathbb{N}$ sufficiently large by part (c), the Q-superlinear convergence follows immediately from (6.12).

If $F\colon \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable with a locally Lipschitzian Jacobian, then Proposition 2.2 (b) shows that

$$\|H_k(x^k - x^*) - \Phi(x^k) + \Phi(x^*)\| = O(\|x^k - x^*\|^2).$$

Since Φ is obviously locally Lipschitzian, we further have

$$\beta_k = \|\Phi(x^k)\| = \|\Phi(x^k) - \Phi(x^*)\| = O(\|x^k - x^*\|).$$

Hence the Q-quadratic rate of convergence of $\{x^k\}$ to x^* follows from (6.11) by using similar arguments as for the proof of the local Q-superlinear convergence. □

7. Numerical results. We implemented the Jacobian smoothing method from Algorithm 4.1 in MATLAB and tested it on a Sun SPARC20 station. As test problems, we use all complementarity problems and all available starting points from the MCPLIB and GAMSLIB collections [15].

The implemented version of the algorithm differs from the one described before in two main aspects: On the one hand, we replaced the monotone Armijo rule by a nonmonotone variant [21]. For the details of the implementation of this nonmonotone Armijo rule, we refer the interested reader to [32].

On the other hand, we incorporated a heuristic backtracking strategy in our implementation in order to avoid domain violations which occur quite often since the mapping F in many examples of the test libraries is not defined everywhere. To this end, we first compute

$$\hat{t}_k := \max\{\nu_k^l \,|\, l = 0, 1, 2, \ldots\}$$

in such a way that $F(x^k + \hat{t}_k d^k)$ is well defined, and then we take $\hat{t}_k$ as the initial steplength, with which we go into the nonmonotone line search test. Note that we allow the backtracking factor ν_k to vary in each iteration. In our implementation we choose ν_k between 0.5 and 0.75; i.e., we increase ν_k gradually in case $l \leq 1$ and decrease it for $l > 1$. This procedure leads to fewer function evaluations and slightly faster convergence for some of the `pgvon105` and `pgvon106` test problems.

The algorithm terminates if one of the following conditions is satisfied:

$$\Psi(x^k) \le \epsilon_1, \;\; \|\nabla\Psi(x^k)\| \le \epsilon_2, \;\; k > k_{\max} \text{ or } t_k < t_{\min}.$$

In the implementation we used the following parameter settings:

$$\rho = 10^{-18}, \;\; p = 2.1, \;\; \lambda = 0.5, \;\; \sigma = 10^{-4}, \;\; \gamma = 30, \;\; \alpha = 0.95, \;\; \eta = 0.9,$$

and

$$\epsilon_1 = 10^{-12}, \;\; \epsilon_2 = 10^{-6}, \;\; k_{\max} = 300, \;\; t_{\min} = 10^{-16}.$$

We report the results for all complementarity problems in the MCPLIB and GAMSLIB libraries and all available starting points in Tables 7.1 and 7.2, respectively. The columns in these tables have the following meanings:

Problem:	name of the test problem in the specific test library,
n:	dimension of the test problem,
SP:	number of starting point,
k:	number of iterations,
F-ev.:	number of function evaluations,
N:	number of Newton steps taken,
G:	number of gradient steps taken,
$\Psi(x^f)$:	$\Psi(x)$ at the final iterate $x = x^f$,
$\lVert\nabla\Psi(x^f)\rVert$:	$\lVert\nabla\Psi(x)\rVert$ at the final iterate $x = x^f$,
B:	number of backtracking steps.

From the definition of the algorithm it follows that the number of Jacobian evaluations is one more than the number of iterations k.

Looking at Tables 7.1 and 7.2, the most obvious observation is that we do not have a single failure; i.e., the main termination criterion

$$\Psi(x^k) \le 10^{-12}$$

is satisfied for all test problems including the difficult ones like `billups`, `colvdual`, `vonthmcp`, and `vonthmge`, to mention just a few.

As known to the authors, there is currently only one other algorithm available which also has no failures on these problems, namely the semismooth Newton-type method by Chen, Chen, and Kanzow [5]. Compared to that algorithm, it seems that our Jacobian smoothing method sometimes needs fewer iterations, whereas the number of function evaluations is usually higher. This may indicate that the stepsize rule (4.4) is not "optimal" and may be improved. However, function evaluations are, in general, considerably cheaper than, e.g., the solution of the linear system (4.1). We also stress that the philosophy of these two methods is different, so it is difficult to compare them with each other.

On the other hand, however, we could try to compare our algorithm with its underlying semismooth Newton method from De Luca, Facchinei, and Kanzow [13]. It turns out that our algorithm is more reliable and that we use considerably fewer gradient steps. In fact, we have just one gradient step, namely, on example `vonthmge`. We believe that this indicates that the smoothing parameter μ regularizes the Jacobian matrix $\Phi'_\mu(x)$ to some extent. This is also reflected by some known theoretical results; e.g., the Jacobian $\Phi'_\mu(x)$ is nonsingular if $F'(x)$ is a P_0-matrix (see [26]), whereas an element from the C-subdifferential $\partial_C\Phi(x)$ is nonsingular only under a slightly stronger assumption (see [13]).

Table 7.1
Numerical results for MCPLIB test problems.

Problem	n	SP	k	F-ev.	N	G	$\Psi(x^f)$	$\|\nabla\Psi(x^f)\|$	B
bertsekas	15	1	34	271	34	0	1.5e-19	3.5e-08	0
bertsekas	15	2	37	353	37	0	1.4e-16	1.0e-06	0
bertsekas	15	3	42	406	42	0	2.5e-19	4.4e-08	0
billups	1	1	27	389	27	0	4.1e-17	1.8e-08	0
colvdual	20	1	15	37	15	0	1.0e-17	4.8e-07	0
colvdual	20	2	26	64	26	0	9.4e-16	4.4e-06	0
colvnlp	15	1	16	39	16	0	2.7e-17	7.8e-07	0
colvnlp	15	2	14	26	14	0	6.8e-15	1.7e-05	0
cycle	1	1	3	5	3	0	8.1e-16	4.0e-08	0
explcp	16	1	5	6	5	0	2.8e-15	7.5e-08	0
hanskoop	14	1	9	14	9	0	2.9e-16	3.3e-08	0
hanskoop	14	2	9	12	9	0	1.4e-17	1.8e-08	0
hanskoop	14	3	8	12	8	0	9.5e-16	1.5e-07	0
hanskoop	14	4	9	13	9	0	4.2e-18	1.0e-08	0
hanskoop	14	5	10	16	10	0	3.3e-18	8.9e-09	1
josephy	4	1	8	11	8	0	1.3e-19	1.7e-09	0
josephy	4	2	7	12	7	0	1.7e-18	1.5e-08	0
josephy	4	3	13	18	13	0	1.0e-14	4.8e-07	0
josephy	4	4	5	6	5	0	2.6e-20	7.6e-10	0
josephy	4	5	5	6	5	0	2.4e-13	2.6e-06	0
josephy	4	6	6	8	6	0	8.1e-21	9.9e-10	0
kojshin	4	1	10	17	10	0	3.3e-24	1.6e-11	0
kojshin	4	2	9	21	9	0	2.9e-15	1.2e-07	0
kojshin	4	3	7	10	7	0	1.8e-15	2.0e-07	0
kojshin	4	4	12	26	12	0	8.0e-17	1.6e-07	0
kojshin	4	5	5	7	5	0	5.0e-18	8.8e-09	0
kojshin	4	6	6	8	6	0	4.7e-25	8.5e-12	0
mathinum	3	1	7	11	7	0	1.7e-24	3.8e-12	0
mathinum	3	2	5	6	5	0	4.4e-15	2.6e-07	0
mathinum	3	3	5	6	5	0	9.2e-18	8.6e-09	0
mathinum	3	4	7	8	7	0	5.1e-23	2.8e-11	0
mathisum	4	1	5	7	5	0	4.1e-19	2.1e-09	0
mathisum	4	2	6	7	6	0	1.5e-13	1.3e-06	0
mathisum	4	3	8	10	8	0	9.0e-17	2.3e-08	0
mathisum	4	4	6	7	6	0	1.5e-22	4.1e-11	0
nash	10	1	8	9	8	0	5.3e-20	2.4e-08	0
nash	10	2	11	25	11	0	1.8e-22	6.9e-10	0
pgvon105	105	1	33	81	33	0	1.1e-13	4.7e-03	33
pgvon105	105	2	33	98	33	0	1.3e-14	7.3e-03	31
pgvon105	105	3	69	251	69	0	6.2e-17	5.0e-04	68
pgvon106	106	1	23	49	23	0	4.6e-14	4.0e-07	23
powell	16	1	13	41	13	0	3.3e-17	9.1e-08	4
powell	16	2	14	36	14	0	2.4e-14	3.3e-06	4
powell	16	3	23	45	23	0	1.3e-13	1.5e-06	4
powell	16	4	16	45	16	0	9.7e-16	5.7e-07	6
scarfanum	13	1	10	13	10	0	1.7e-16	1.7e-07	0
scarfanum	13	2	12	15	12	0	1.7e-16	1.7e-07	0
scarfanum	13	3	12	16	12	0	1.7e-16	1.7e-07	1
scarfasum	14	1	8	11	8	0	1.1e-18	3.1e-08	0
scarfasum	14	2	10	14	10	0	9.6e-17	2.8e-07	0
scarfasum	14	3	11	14	11	0	2.5e-19	1.4e-08	0
scarfbnum	39	1	23	36	23	0	1.7e-14	3.4e-05	0
scarfbnum	39	2	24	42	24	0	2.4e-14	3.7e-05	0
scarfbsum	40	1	20	56	20	0	1.2e-16	1.9e-06	0
scarfbsum	40	2	26	72	26	0	9.1e-20	5.2e-08	0
sppe	27	1	7	8	7	0	4.8e-14	4.4e-07	0
sppe	27	2	6	7	6	0	4.8e-25	2.9e-12	0
tobin	42	1	9	12	9	0	4.8e-13	9.9e-07	0
tobin	42	2	11	15	11	0	4.8e-24	3.1e-12	0

Table 7.2
Numerical results for GAMSLIB test problems.

Problem	n	SP	k	F-ev.	N	G	$\Psi(x^f)$	$\|\nabla\Psi(x^f)\|$	B
cafemge	101	1	11	19	11	0	7.9e-25	1.7e-09	0
cammge	128	1	0	1	0	0	5.1e-13	3.1e-04	0
co2mge	208	1	1	2	1	0	1.3e-14	1.0e-07	0
dmcmge	170	1	88	523	88	0	1.3e-21	2.1e-07	1
etamge	114	1	20	49	20	0	1.6e-15	3.6e-05	0
finmge	153	1	0	1	0	0	2.2e-14	7.6e-06	0
hansmcp	43	1	17	31	17	0	3.3e-14	7.8e-07	0
hansmge	43	1	14	30	14	0	4.9e-13	9.1e-07	0
harkmcp	32	1	13	16	13	0	2.0e-16	2.9e-08	0
kehomge	9	1	10	12	10	0	1.7e-20	7.9e-09	0
mr5mcp	350	1	10	17	10	0	1.7e-18	2.8e-07	1
nsmge	212	1	12	19	12	0	5.6e-18	2.9e-07	0
oligomcp	6	1	6	7	6	0	7.1e-17	1.5e-07	0
sammge	23	1	0	1	0	0	0.0	0.0	0
scarfmcp	18	1	9	12	9	0	9.2e-17	1.3e-07	1
scarfmge	18	1	11	15	11	0	5.3e-13	1.1e-05	0
shovmge	51	1	1	2	1	0	5.6e-14	5.7e-05	0
threemge	9	1	0	1	0	0	0.0	0.0	0
transmcp	11	1	13	22	13	0	3.1e-16	2.5e-08	0
two3mcp	6	1	8	12	8	0	4.8e-13	2.0e-05	0
unstmge	5	1	8	9	8	0	1.6e-13	7.6e-07	0
vonthmcp	125	1	54	280	54	0	6.1e-15	2.9e-02	37
vonthmge	80	1	31	97	30	1	4.5e-13	1.3e-04	0

We finally stress that we also tested some other parameter settings; there, we usually had some more gradient steps but still fewer than for the method from [13]. This fact may explain why our Jacobian smoothing method seems to be superior to its underlying semismooth Newton method from [13], since it is well accepted in the community that taking as many Newton steps as possible usually improves the overall behavior of the algorithm. On the other hand, we stress that we were not able to solve the `vonthmge` example without using a gradient step.

8. Final remarks. In this paper, we introduced a new algorithm for the solution of a general (i.e., not necessarily monotone) complementarity problem. We call this algorithm a Jacobian smoothing method since, basically, it is a perturbation of a semismooth Newton method being applied to a reformulation of the complementarity problem as a nonsmooth system of equations $\Phi(x) = 0$. In this perturbation, we replace an element from the generalized Jacobian by a standard Jacobian of a smooth operator Φ_μ which approximates Φ for $\mu \to 0$.

The basic idea of this Jacobian smoothing method is taken from the recent paper [10] by Chen, Qi, and Sun. We modified their algorithm in such a way that it becomes applicable to general complementarity problems. Although this modification makes the convergence analysis rather technical (especially the global one), the main convergence results are quite nice. Moreover, the numerical performance is extremely promising. In fact, we are able to solve all complementarity problems from the MCPLIB and GAMSLIB test problem collections. In particular, our Jacobian smoothing method is considerably more reliable than the semismooth method by De Luca, Facchinei, and Kanzow [13], which is the underlying semismooth Newton method for our algorithm.

It would be interesting to see how our perturbation technique would work if we applied it to other equation reformulations of the nonlinear complementarity problem like those presented in [28, 35, 5]. Finally, it would also be interesting to see how the Jacobian smoothing method would work on mixed complementarity problems. An extension to this more general class of problems seems possible by using, e.g., an idea from Billups [1]; see also Qi [35] and Sun and Womersley [39]. We leave this as a future research topic.

REFERENCES

[1] S.C. Billups, *Algorithms for Complementarity Problems and Generalized Equations*, Ph.D. Thesis, Computer Sciences Department, University of Wisconsin, Madison, WI, August 1995.

[2] S.C. Billups, S.P. Dirkse, and M.C. Ferris, *A comparison of algorithms for large scale mixed complementarity problems*, Comput. Optim. Appl., 7 (1997), pp. 3–25.

[3] J. Burke and S. Xu, *The global linear convergence of a non-interior path-following algorithm for linear complementarity problems*, Math. Oper. Res., 23 (1998), pp. 719–734.

[4] B. Chen and X. Chen, *A global and local superlinear continuation-smoothing method for $P_0 + R_0$ and monotone NCP*, SIAM J. Optim., to appear.

[5] B. Chen, X. Chen, and C. Kanzow, *A Penalized Fischer-Burmeister NCP-Function: Theoretical Investigation and Numerical Results*, Preprint 126, Institute of Applied Mathematics, University of Hamburg, Germany, September 1997 (revised May 1998).

[6] B. Chen and P.T. Harker, *A non-interior-point continuation method for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1168–1190.

[7] B. Chen and P.T. Harker, *Smooth approximations to nonlinear complementarity problems*, SIAM J. Optim., 7 (1997), pp. 403–420.

[8] B. Chen and N. Xiu, *A Global Linear and Local Quadratic Non-Interior Continuation Method for Nonlinear Complementarity Problems Based on Chen-Mangasarian Smoothing Function*, Tech. Report, Department of Management and Systems, Washington State University, Pullman, WA, 1997.

[9] C. Chen and O.L. Mangasarian, *A class of smoothing functions for nonlinear and mixed complementarity problems*, Comput. Optim. Appl., 5 (1996), pp. 97–138.

[10] X. Chen, L. Qi, and D. Sun, *Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities*, Math. Comp., 67 (1998), pp. 519–540.

[11] X. Chen and Y. Ye, *On Homotopy-Smoothing Methods for Variational Inequalities*, Tech. Report AMR 96/39, School of Mathematics, The University of New South Wales, Sydney, Australia, December 1996.

[12] F.H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley, New York, 1983 (reprinted by SIAM, Philadelphia, 1990).

[13] T. De Luca, F. Facchinei, and C. Kanzow, *A semismooth equation approach to the solution of nonlinear complementarity problems*, Math. Programming, 75 (1996), pp. 407–439.

[14] J.E. Dennis, Jr. and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983 (reprinted by SIAM, Philadelphia, 1996).

[15] S.P. Dirkse and M.C. Ferris, *MCPLIB: A collection of nonlinear mixed complementarity problems*, Optim. Methods Software, 5 (1995), pp. 123–156.

[16] F. Facchinei and J. Soares, *A new merit function for nonlinear complementarity problems and a related algorithm*, SIAM J. Optim., 7 (1997), pp. 225–247.

[17] M.C. Ferris and J.-S. Pang, *Engineering and economic applications of complementarity problems*, SIAM Rev., 39 (1997), pp. 669–713.

[18] A. Fischer, *A special Newton-type optimization method*, Optimization, 24 (1992), pp. 269–284.

[19] A. Fischer, *Solution of monotone complementarity problems with locally Lipschitzian functions*, Math. Programming, 76 (1997), pp. 513–532.

[20] S.A. Gabriel and J.J. Moré, *Smoothing of mixed complementarity problems*, In Complementarity and Variational Problems. State of the Art, M.C. Ferris and J.-S. Pang, eds., SIAM, Philadelphia, 1997, pp. 105–116.

[21] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone linesearch technique for Newton's method*, SIAM J. Numer. Anal., 23 (1986), pp. 707–716.

[22] P.T. Harker and J.-S. Pang, *Finite dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 161–220.

[23] K. Hotta and A. Yoshise, *Global Convergence of a Class of Non-Interior-Point Algorithms Using Chen-Harker-Kanzow Functions for Nonlinear Complementarity Problems*, Tech. Report 708, Institute of Policy and Planning Sciences, University of Tsukuba, Tsukuba, Ibaraki, Japan, December 1996.

[24] H. Jiang, *Smoothed Fischer-Burmeister Equation Methods for the Complementarity Problem*, Tech. Report, Department of Mathematics, The University of Melbourne, Parkville, Victoria, Australia, June 1997.

[25] H. Jiang and L. Qi, *A new nonsmooth equations approach to nonlinear complementarity problems*, SIAM J. Control Optim., 35 (1997), pp. 178–193.

[26] C. Kanzow, *Some noninterior continuation methods for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 851–868.

[27] C. Kanzow, *A new approach to continuation methods for complementarity problems with uniform P-functions*, Oper. Res. Lett., 20 (1997), pp. 85–92.

[28] C. Kanzow and H. Kleinmichel, *A new class of semismooth Newton-type methods for nonlinear complementarity problems*, Comput. Optim. Appl., to appear.

[29] C. Kanzow and H.-D. Qi, *A QP-free constrained Newton-type method for variational inequality problems*, Math. Programming, to appear.

[30] B. Kummer, *Newton's method for nondifferentiable functions*, In Advances in Mathematical Optimization, J. Guddat et al., eds., Akademie-Verlag, Berlin, Germany, 1988, pp. 114–125.

[31] J.J. Moré and D.C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Stat. Comput., 4 (1983), pp. 553–572.

[32] H. Pieper, *Ein Glättungsverfahren zur Lösung von nichtlinearen Komplementaritätsproblemen*, Diploma Thesis, Institute of Applied Mathematics, University of Hamburg, Germany, May 1997.

[33] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.

[34] L. Qi, *C-Differentiability, C-Differential Operators and Generalized Newton Methods*, Tech. Report, School of Mathematics, The University of New South Wales, Sydney, Australia, January 1996.

[35] L. Qi, *Regular Pseudo-Smooth NCP and BVIP Functions and Globally and Quadratically Convergent Generalized Newton Methods for Complementarity and Variational Inequality Problems*, Tech. Report AMR 97/14, School of Mathematics, The University of New South Wales, Sydney, Australia, July 1997 (revised September 1997).

[36] L. Qi and D. Sun, *Globally Linearly, and Globally and Locally Superlinearly Convergent Versions of the Hotta-Yoshise Non-Interior Point Algorithm for Nonlinear Complementarity Problems*, Tech. Report, School of Mathematics, The University of New South Wales, Sydney, Australia, May 1997.

[37] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–367.

[38] S.M. Robinson, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.

[39] D. Sun and R.S. Womersley, *A new unconstrained differentiable merit function for box constrained variational inequality problems and a damped Gauss–Newton method*, SIAM J. Optim., 9 (1999), pp. 409–434.

[40] P. Tseng, *Analysis of a Non-Interior Continuation Method Based on Chen-Mangasarian Smoothing Functions for Complementarity Problems*, Tech. Report, Department of Mathematics, University of Washington, Seattle, WA, July 1997.

[41] S. Xu, *The Global Linear Convergence of an Infeasible Non-Interior Path-Following Algorithm for Complementarity Problems with Uniform P-Functions*, Tech. Report, Department of Mathematics, University of Washington, Seattle, WA, December 1996.

[42] S. Xu, *The Global Linear Convergence and Complexity of a Non-Interior Path-Following Algorithm for Monotone LCP Based on Chen-Harker-Kanzow-Smale Smooth Functions*, Tech. Report, Department of Mathematics, University of Washington, Seattle, WA, February 1997.

[43] S. Xu and J.V. Burke, *A polynomial time interior-point path-following algorithm for LCP based on Chen-Harker-Kanzow smoothing techniques*, Math. Programming, to appear.

SIAM J. OPTIM.
Vol. 9, No. 2, pp. 374–387

OPTIMALITY CONDITIONS FOR OPTIMIZATION PROBLEMS WITH COMPLEMENTARITY CONSTRAINTS[*]

J. J. YE[†]

Abstract. Optimization problems with complementarity constraints are closely related to optimization problems with variational inequality constraints and bilevel programming problems. In this paper, under mild constraint qualifications, we derive some necessary and sufficient optimality conditions involving the proximal coderivatives. As an illustration of applications, the result is applied to the bilevel programming problems where the lower level is a parametric linear quadratic problem.

Key words. optimization problems, complementarity constraints, optimality conditions, bilevel programming problems, proximal normal cones

AMS subject classifications. 49K99, 90C, 90D65

PII. S1052623497321882

1. Introduction.

The main purpose of this paper is to derive necessary and sufficient optimality conditions for the *optimization problem with complementarity constraints* (OPCC) defined as follows:

$$\begin{array}{llll} & \text{(OPCC)} & \min & f(x,y,u) \\ (1.1) & & \text{s.t.} & \langle u, \psi(x,y,u)\rangle = 0, \quad u \geq 0, \quad \psi(x,y,u) \leq 0 \\ & & & L(x,y,u) = 0, \quad g(x,y,u) \leq 0, \quad (x,y,u) \in \Omega, \end{array}$$

where $f : R^{n+m+q} \to R$, $\psi : R^{n+m+q} \to R^q$, $L : R^{n+m+q} \to R^l$, $g : R^{n+m+q} \to R^d$, and Ω is a nonempty subset of R^{n+m+q}.

(OPCC) is an optimization problem with equality and inequality constraints. However, due to the complementarity constraint (1.1), the Karush–Kuhn–Tucker (KKT) necessary optimality condition is rarely satisfied by (OPCC) since it can be shown as in [9, Proposition 1.1] that there always exists a nontrivial abnormal multiplier. This is equivalent to saying that the usual constraint qualification conditions, such as the Mangasarian–Fromovitz condition, will never be satisfied (see [8, Proposition 3.1]). The purpose of this paper is to derive necessary and sufficient optimality conditions under mild constraint qualifications that are satisfied by a large class of OPCCs.

To motivate our main results, we formulate problem (OPCC), where $\Omega = R^{n+m+q}$, as the following optimization problem with a generalized equation constraint:

$$\begin{array}{llll} & \text{(GP)} & \min & f(x,y,u) \\ (1.2) & & \text{s.t.} & 0 \in -\psi(x,y,u) + N(u, R^q_+), \\ & & & L(x,y,u) = 0, \quad g(x,y,u) \leq 0, \end{array}$$

where

$$N(u,C) := \begin{cases} \text{the normal cone of } C \text{ at } y & \text{if } u \in C, \\ \emptyset & \text{if } u \notin C \end{cases}$$

[*] Received by the editors May 26, 1997; accepted for publication (in revised form) May 4, 1998; published electronically March 17, 1999. This work was supported by the Natural Sciences and Engineering Research Council of Canada and a University of Victoria internal research grant.

http://www.siam.org/journals/siopt/9-2/32188.html

[†] Department of Mathematics and Statistics, University of Victoria, Victoria, BC V8W 3P4, Canada (janeye@uvic.ca).

is the normal cone operator in the sense of convex analysis.

Let $(\bar{x}, \bar{y}, \bar{u})$ be a solution of (OPCC), where $\Omega = R^{n+m+q}$. If $N(u, R^q_+)$ were single-valued and smooth, then the generalized equation constraint (1.2) would reduce to an ordinary equation. Using the KKT condition, we could deduce that if a constraint qualification is satisfied for (GP) and the problem data are smooth, then there exist KKT multipliers $\xi \in R^l$, $\zeta \in R^d$, $\eta \in R^q$ such that

$$\begin{aligned} 0 &= \nabla f(\bar{x}, \bar{y}, \bar{u}) + \nabla L(\bar{x}, \bar{y}, \bar{u})^\top \xi + \nabla g(\bar{x}, \bar{y}, \bar{u})^\top \zeta \\ &\quad - \nabla \psi(\bar{x}, \bar{y}, \bar{u})^\top \eta + \{0\} \times \{0\} \times \nabla N_{R^q_+}(\bar{u})^\top \eta, \\ 0 &= \langle \zeta, g(\bar{x}, \bar{y}, \bar{u}) \rangle, \ \zeta \geq 0, \end{aligned}$$

where ∇ denotes the usual gradient, $M^\top$ denotes the transpose of the matrix M, and N_C denotes the map $y \to N(y, C)$. However, $u \rightrightarrows N(u, R^q_+)$ is in general a set-valued map. Naturally, we hope to replace $\nabla N_{R^q_+}(\bar{u})^\top \eta$ by the image of some derivatives of the set-valued map $u \rightrightarrows N(u, R^q_+)$ acting on the vector η. The natural candidate for such a derivative of set-valued maps is the Mordukhovich coderivative (see Definition 2.3) since the Mordukhovich coderivatives have a good calculus, and in the case when the set-valued map is single-valued and smooth, the image of the Mordukhovich coderivative acting on a vector coincides with the usual gradient operator acting on the vector (see [6, Proposition 2.4]). Indeed, as in [7], we can show that if $(\bar{x}, \bar{y}, \bar{u})$ is an optimal solution of (OPCC) and a constraint qualification holds, then there exist $\xi \in R^l$, $\zeta \in R^d$, $\eta \in R^q$ such that

$$\begin{aligned} 0 &\in \nabla f(\bar{x}, \bar{y}, \bar{u}) + \nabla L(\bar{x}, \bar{y}, \bar{u})^\top \xi + \nabla g(\bar{x}, \bar{y}, \bar{u})^\top \zeta \\ &\quad - \nabla \psi(\bar{x}, \bar{y}, \bar{u})^\top \eta + \{0\} \times \{0\} \times D^* N_{R^q_+}(\bar{u}, \psi(\bar{x}, \bar{y}, \bar{u}))(\eta), \\ 0 &= \langle \zeta, g(\bar{x}, \bar{y}, \bar{u}) \rangle, \qquad \zeta \geq 0, \end{aligned}$$

where D^* denotes the Mordukhovich coderivative (see Definition 2.3). Recall from [7, Definition 2.8] that a set-valued map $\Phi : R^n \rightrightarrows R^q$ with a closed graph is said to be pseudo-upper-Lipschitz continuous at $(\bar{z}, \bar{v})$ with $\bar{v} \in \Phi(\bar{z})$ if there exist a neighborhood U of $\bar{z}$, a neighborhood V of $\bar{v}$, and a constant $\mu > 0$ such that

$$\Phi(z) \cap V \subset \Phi(\bar{z}) + \mu \|z - \bar{z}\| B \quad \forall z \in U.$$

The constraint qualification for the above necessary condition involving the Mordukhovich coderivative turns out to be the pseudo-upper-Lipschitz continuity of the set-valued map

$$\Sigma(v_1, v_2, v_3) := \{(x, y, u) : v_1 \in -\psi(x, y, u) + N(u, R^q_+), L(x, y, u) = v_2, g(x, y, u) + v_3 \leq 0\}$$

at $(\bar{x}, \bar{y}, \bar{u}, 0)$. This constraint qualification is very mild since the pseudo-upper-Lipschitz continuity is weaker than both the upper-Lipschitz continuity and the pseudo-Lipschitz continuity (the so-called Aubin property). However, the Mordukhovich normal cone involved in the necessary condition may be too large sometimes. For example, in [7, Example 4.1], both $(0, 0)$ and $(1, 1)$ satisfy the above necessary conditions, but only $(1, 1)$ is the unique optimal solution. Can one replace the Mordukhovich normal cone involved in the necessary condition by the potentially smaller proximal normal cone? The answer is negative in general, since the proximal coderivative as defined in Definition 2.3 usually has only a “fuzzy” calculus. Consider the following optimization problem:

$$\begin{array}{ll} \min & -y \\ \text{s.t.} & y-u=0, \quad yu=0, \quad y\geq 0, \quad u\geq 0. \end{array}$$

The unique optimal solution $(0,0)$ does not satisfy the KKT condition but satisfies the necessary condition involving the Mordukhovich coderivatives. It does not satisfy the necessary condition with the Mordukhovich normal cone replaced by the proximal normal cone. This example shows that some extra assumptions are needed for the necessary condition involving the proximal coderivatives to hold. In this paper such a condition is found. Moreover, we show that the proximal normal cone involved in the necessary condition can be represented by a system of linear and nonlinear equations, and the necessary optimality conditions involving the proximal coderivatives turn out to be sufficient under some convexity assumptions on the problem data.

Although the optimization problems with complementarity constraints are a class of optimization problems with independent interest, the incentive to study (OPCC) mainly comes from the following optimization problem with variational inequality constraints (OPVIC), where the constraint region of the variational inequality is a system of inequalities:

$$\begin{array}{lll} \text{(OPVIC)} & \min & f(x,y) \\ & \text{s.t.} & y\in S(x), \quad g(x,y)\leq 0, \quad (x,y)\in\Omega, \end{array}$$

where $f: R^{n+m}\to R$, Ω is a nonempty subset of R^{m+n} and $S(x)$ is the *solution set of a variational inequality with parameter* x; i.e.,

$$S(x)=\{y\in R^m : \psi(x,y)\leq 0 \text{ and } \langle F(x,y), z-y\rangle \geq 0 \;\; \forall z \text{ s.t. } \psi(x,z)\leq 0\},$$

where $F: R^{n+m}\to R^m$ and $\psi: R^{n+m}\to R^q$. The recent monograph [4] by Luo, Pang, and Ralph has an extensive study for (OPVIC). The reader may find the references for the various optimality conditions for (OPVIC) from [4].

(OPCC) is closely related to OPVICs and bilevel programming problems. Indeed, if ψ is $C^1$and quasi convex in y and a certain constraint qualification condition holds at $\bar{y}$ for the optimization problem

$$\min \quad \langle F(\bar{x},\bar{y}), z\rangle \quad \text{ s.t. } \psi(\bar{x},z)\leq 0,$$

then by the KKT necessary and sufficient optimality condition, $(\bar{x},\bar{y})$ is a solution of (OPVIC) if and only if there exists $\bar{u}\in R^q$ such that $(\bar{x},\bar{y},\bar{u})$ is a solution of the following optimization problem:

$$\begin{array}{lll} \text{(KS)} & \min & f(x,y) \\ & \text{s.t.} & \langle u, \psi(x,y)\rangle = 0, \quad u\geq 0, \quad \psi(x,y)\leq 0, \\ & & F(x,y)+\nabla_y \psi(x,y)^\top u = 0, \\ & & g(x,y)\leq 0, \qquad (x,y)\in\Omega, \end{array}$$

which is a special case of (OPCC).

In the case where $F(x,y)=\nabla_y h(x,y)$, where $h: R^{n+m}\to R$ is differentiable and pseudoconvex in y, (KS) is equivalent to the following *bilevel programming problem* (BLPP), or so-called Stackelberg game:

$$\begin{array}{lll} \text{(BLPP)} & \min & f(x,y) \\ & \text{s.t.} & y\in S(x), \quad g(x,y)\leq 0, \quad (x,y)\in\Omega, \end{array}$$

where $S(x)$ is the set of solutions of the problem (P_x):

$$(P_x) \qquad \text{minimize} \quad h(x,y) \qquad \text{s.t. } \psi(x,y) \leq 0.$$

We organize the paper as follows. Section 2 contains background material on nonsmooth analysis and preliminary results. In section 3 we derive the necessary and sufficient optimality conditions for (OPCC). As an illustration of applications, we also apply the result to (BLPP), where the lower level is a linear quadratic programming problem.

2. Preliminaries. This section contains some background material on nonsmooth analysis and preliminary results which will be used later. We give only concise definitions that will be needed in the paper. For more detailed information on the subject, our references are Clarke [1, 2], Loewen [3], and Mordukhovich [6].

First we give some concepts for various normal cones and subgradients.

DEFINITION 2.1. *Let Ω be a nonempty subset of R^n. Given $\bar{z} \in cl\Omega$, the closure of set Ω, the convex cone*

$$N^\pi(\bar{z},\Omega) := \{\xi \in R^n : \exists M > 0 \ s.t. \ \langle \xi, z - \bar{z}\rangle \leq M\|z-\bar{z}\|^2 \ \forall z \in \Omega\}$$

is called the proximal normal cone to set Ω at point $\bar{z}$, and the closed cone

$$\hat{N}(\bar{z},\Omega) := \{\lim_{i\to\infty} \xi_i : \xi_i \in N^\pi(z_i,\Omega), z_i \to \bar{z}\}$$

is called the limiting normal cone to Ω at point $\bar{z}$.

REMARK 2.1. *It is known that if Ω is convex, then the proximal normal cone and the limiting normal cones coincide with the normal cone in the sense of convex analysis.*

DEFINITION 2.2. *Let $f : R^n \to R \cup \{+\infty\}$ be lower semicontinuous and finite at $\bar{z} \in R^n$. The limiting subgradient of f at $\bar{z}$ is defined to be the set*

$$\hat{\partial} f(\bar{z}) := \{\zeta : (\zeta, -1) \in \hat{N}(\bar{z}, epi\ f)\},$$

where $epi\ f := \{(z,v) : v \geq f(z)\}$ denotes the epigragh of f.

REMARK 2.2. *It is known that if f is a convex function, the limiting subgradient coincides with the subgradient in the sense of convex analysis. For a locally Lipschitz function f, $\partial f = co\hat{\partial} f(x)$, where ∂ denotes the Clarke generalized gradient and co denotes the convex hull. Hence the limiting subgradient is in general a smaller set than the Clarke generalized gradient.*

For set-valued maps, the definition for limiting normal cone leads to the definition of coderivative of a set-valued map introduced by Mordukhovich (see, e.g., [6]).

DEFINITION 2.3. *Let $\Phi : R^n \rightrightarrows R^q$ be an arbitrary set-valued map (assigning to each $z \in R^n$ a set $\Phi(z) \subseteq R^q$ which may be empty) and $(\bar{z},\bar{v}) \in$ cl $Gr\Phi$, where $Gr\Phi$ denotes the graph of Φ; i.e., $(z,v) \in Gr\Phi$ if and only if $v \in \Phi(z)$. The set-valued maps from R^q into R^n defined by*

$$D^*_\pi \Phi(\bar{z},\bar{v})(\eta) = \{\zeta \in R^n : (\zeta, -\eta) \in N^\pi((\bar{z},\bar{v}), Gr\Phi)\},$$
$$D^* \Phi(\bar{z},\bar{v})(\eta) = \{\zeta \in R^n : (\zeta, -\eta) \in \hat{N}((\bar{z},\bar{v}), Gr\Phi)\}$$

are called the proximal and Mordukhovich coderivatives of Φ at point $(\bar{z},\bar{v})$, respectively.

PROPOSITION 2.4. *Suppose B is closed, $\bar{x} \in A$, $\bar{x} \notin B$. Then*

$$N^{\pi}(\bar{x}, A \cup B) = N^{\pi}(\bar{x}, A).$$

Proof. Since $\bar{x} \notin B$ and B is closed, there exists a neighborhood of $\bar{x}$ that is not contained in B. Therefore, from the definition of the proximal normal cone, we have

$$N^{\pi}(\bar{x}, A \cup B) = N^{\pi}(\bar{x}, A). \qquad \square$$

In the following proposition we show that the proximal normal cone of a union of a finite number of sets is the intersection of the proximal cones.

PROPOSITION 2.5. *Let $\Omega = \cup_{i=1}^{m} \Omega_i$ and $\bar{x} \in \cap_{i=1}^{m} \Omega_i$. Suppose $\Omega_i \ \forall\, i = 1, 2, \ldots, m$ are closed. Then*

$$N^{\pi}(\bar{x}, \Omega) = \cap_{i=1}^{m} N^{\pi}(\bar{x}, \Omega_i).$$

Proof. Let $\zeta \in N^{\pi}(\bar{x}, \Omega)$. Then, by definition, there exists a constant $M > 0$ such that

$$\langle \zeta, x - \bar{x} \rangle \leq M \left\| x - \bar{x} \right\|^2 \qquad \forall x \in \Omega = \cup_{i=1}^{m} \Omega_i.$$

Since $\bar{x} \in \cap_{i=1}^{m} \Omega_i$, the above inequality implies that $\zeta \in \cap_{i=1}^{m} N^{\pi}(\bar{x}, \Omega_i)$.

Conversely, suppose $\zeta \in \cap_{i=1}^{m} N^{\pi}(\bar{x}, \Omega_i)$. Then for all $i = 1, 2, \ldots, m$, there exists $M_i > 0$ such that

$$\langle \zeta, x - \bar{x} \rangle \leq M_i \left\| x - \bar{x} \right\|^2 \qquad \forall x \in \Omega_i.$$

That is, there exists $M = \max_{i \in \{1,2,\ldots,m\}} M_i > 0$ such that

$$\langle \zeta, x - \bar{x} \rangle \leq M \left\| x - \bar{x} \right\|^2 \qquad \forall x \in \Omega = \cup_{i=1}^{m} \Omega_i,$$

which implies that $\zeta \in N^{\pi}(\bar{x}, \Omega)$. $\square$

The above decomposition formula for calculating the proximal normal cones turns out to be very useful, since when a set can be written as a union of some convex sets, the task of calculating the proximal normal cones is reduced to calculating the normal cone to convex sets which are easier to calculate. The following proposition is a nice application of the decomposition formula and will be used to calculate the proximal normal cone to the graph of the set-valued map $N_{R^q_+}$ for general q in Proposition 2.7.

PROPOSITION 2.6.

$$N^{\pi}((\bar{x}, \bar{y}), GrN_{R_+}) = \begin{cases} \{0\} \times R & \textit{if } \bar{x} > 0, \bar{y} = 0, \\ R \times \{0\} & \textit{if } \bar{x} = 0, \bar{y} < 0, \\ (-\infty, 0] \times [0, \infty) & \textit{if } \bar{x} = \bar{y} = 0. \end{cases}$$

Proof. It is easy to see that $\mathrm{Gr}N_{R_+} = \Omega_1 \cup \Omega_2$, where $\Omega_1 = [0, \infty) \times \{0\}$ and $\Omega_2 = \{0\} \times (-\infty, 0]$.

We discuss the following three cases.

Case 1. $\bar{x} > 0$, $\bar{y} = 0$.

In this case, $(\bar{x}, \bar{y}) \in \Omega_1$ and $(\bar{x}, \bar{y}) \notin \Omega_2$. Since Ω_2 is closed, by Proposition 2.4 we have in this case

$$N^{\pi}((\bar{x}, \bar{y}), GrN_{R_+}) = N((\bar{x}, \bar{y}), \Omega_1) = \{0\} \times R.$$

Case 2. $\bar{x} = 0$, $\bar{y} < 0$.

In this case, $(\bar{x}, \bar{y}) \in \Omega_2$ and $(\bar{x}, \bar{y}) \notin \Omega_1$. Since Ω_1 is closed, by Proposition 2.4 we have in this case

$$N^\pi((\bar{x}, \bar{y}), GrN_{R_+}) = N((\bar{x}, \bar{y}), \Omega_2) = R \times \{0\}.$$

Case 3. $\bar{x} = \bar{y} = 0$.

In this case, $(\bar{x}, \bar{y}) \in \Omega_1 \cap \Omega_2$. By Proposition 2.5 we have

$$\begin{aligned} N^\pi((\bar{x}, \bar{y}), GrN_{R_+}) &= N((\bar{x}, \bar{y}), \Omega_1) \cap N((\bar{x}, \bar{y}), \Omega_2) \\ &= ((-\infty, 0] \times R) \cap (R \times [0, \infty)) \\ &= (-\infty, 0] \times [0, \infty). \qquad \square \end{aligned}$$

Now we are in a position to give an expression for the proximal normal cone to the graph of the set-valued map $N_{R^q_+}$ for general q.

PROPOSITION 2.7. *For any* $(\bar{x}, \bar{y}) \in GrN_{R^q_+}$, *define*

$$\begin{aligned} L := L(\bar{x}) &: = \{i \in \{1, 2, \dots, q\} : \bar{x}_i > 0\}, \\ I_+ := I_+(\bar{x}, \bar{y}) &: = \{i \in \{1, 2, \dots, q\} : \bar{x}_i = 0, \bar{y}_i < 0\}, \\ I_0 := I_0(\bar{x}, \bar{y}) &: = \{i \in \{1, 2, \dots, q\} : \bar{x}_i = 0, \bar{y}_i = 0\}. \end{aligned}$$

Then

$$N^\pi((\bar{x}, \bar{y}), GrN_{R^q_+}) = \{(\gamma, -\eta) \in R^{2q} : \eta_{I_0} \leq 0, \eta_{I_+} = 0, \gamma_L = 0, \gamma_{I_0} \leq 0\}.$$

Proof. Since

$$\begin{aligned} GrN_{R^q_+} &= \left\{(x, y) \in R^{2q} : y \in N(x, R^q_+)\right\} \\ &= \left\{(x, y) \in R^{2q} : y \in N(x_1, R_+) \times N(x_2, R_+) \times \cdots \times N(x_q, R_+)\right\} \\ &= \left\{(x, y) \in R^{2q} : (x_i, y_i) \in GrN_{R_+} \forall i = 1, 2, \dots, q\right\}, \end{aligned}$$

we have

$$(x, y) \in \mathrm{Gr}N_{R^q_+} \quad \text{if and only if} \quad (x_1, y_1, x_2, y_2, \dots, x_q, y_q) \in \prod_{i=1}^q \mathrm{Gr}N_{R_+}.$$

Hence from the definition, it is clear that

$$(\gamma, -\eta) \in N^\pi((\bar{x}, \bar{y}), GrN_{R^q_+})$$

if and only if

$$\begin{aligned} (\gamma_1, -\eta_1, \gamma_2, -\eta_2, \dots, \gamma_q, -\eta_q) &\in N^\pi\left((\bar{x}_1, \bar{y}_1, \bar{x}_2, \bar{y}_2, \dots, \bar{x}_q, \bar{y}_q), \prod_{i=1}^q GrN_{R_+}\right) \\ &= \prod_{i=1}^q N^\pi((\bar{x}_i, \bar{y}_i), \mathrm{Gr}N_{R_+}). \end{aligned}$$

The rest of the proof follows from Proposition 2.6. $\square$

It turns out that we can express any element of $N^\pi((\bar{x}, \bar{y}),\mathrm{Gr}N_{R^q_+})$ by a system of nonlinear equations as in the following proposition.

Proposition 2.8.

$$(\gamma, -\eta) \in N^{\pi}((\bar{x}, \bar{y}), GrN_{R^q_+})$$

if and only if there exist $\alpha, \beta \in R^{2q}_+$*such that*

$$0 = \sum_{i=1}^{q} \bar{x}_i(\alpha_i + \beta_i) - \sum_{i=1}^{q} \bar{y}_i(\alpha_{q+i} + \beta_{q+i}), \tag{2.1}$$

$$\gamma_i = -\alpha_i + \bar{y}_i\beta_i \quad \forall i = 1, 2, \ldots, q, \tag{2.2}$$

$$\eta_i = -\alpha_{q+i} + \bar{x}_i\beta_{q+i} \quad \forall i = 1, 2, \ldots, q. \tag{2.3}$$

Proof. By Proposition 2.7, $(\gamma, -\eta) \in N^{\pi}((\bar{x}, \bar{y}), GrN_{R^q_+})$ if and only if

$$\eta_{I_0} \le 0, \quad \eta_{I_+} = 0, \quad \gamma_L = 0, \quad \gamma_{I_0} \le 0.$$

By the definition for the index sets I_0, I_+, L in Proposition 2.7, we have

$$\begin{aligned} \eta_{I_0} \le 0, \gamma_{I_0} \le 0 \quad & \text{if and only if } \bar{x}_i = \bar{y}_i = 0 \Longrightarrow \eta_i \le 0, \quad \gamma_i \le 0, \\ \eta_{I_+} = 0 \quad & \text{if and only if } \bar{y}_i < 0 \Longrightarrow \eta_i = 0, \\ \gamma_L = 0 \quad & \text{if and only if } \bar{x}_i > 0 \Longrightarrow \gamma_i = 0. \end{aligned}$$

Since for any $(\bar{x}, \bar{y}) \in GrN_{R^q_+}$, $\bar{x} \ge 0, \bar{y} \le 0$, for nonnegative vectors α and β, (2.1) is equivalent to

$$\bar{x}_i(\alpha_i + \beta_i) = 0, \quad \bar{y}_i(\alpha_{q+i} + \beta_{q+i}) = 0 \quad \forall i = 1, \ldots, q.$$

Hence the existence of nonnegative vectors α and β satisfying (2.1)–(2.2) is equivalent to the following condition:

$$\begin{aligned} & \bar{x}_i = \bar{y}_i = 0 \Longrightarrow \eta_i \le 0, \qquad \gamma_i \le 0, \\ & \bar{y}_i < 0 \Longrightarrow \eta_i = 0, \\ & \bar{x}_i > 0 \Longrightarrow \gamma_i = 0. \end{aligned}$$

Consequently, it is equivalent to

$$\eta_{I_0} \le 0, \quad \eta_{I_+} = 0, \quad \gamma_L = 0, \quad \gamma_{I_0} \le 0.$$

The proof of the proposition is therefore complete. □

Finally, we would like to recall the following definition of a very mild constraint qualification called "calmness," introduced by Clarke [1].

Definition 2.9. *Let* $\bar{x}$ *be a local solution to the following mathematical programming problem:*

$$\begin{aligned} \textit{minimize} \quad & f(x) \\ \textit{s.t.} \quad & g(x) \le 0, \\ & h(x) = 0, \\ & x \in C, \end{aligned}$$

where $f : R^d \to R^n$, $g : R^d \to R^m$, and C is a closed subset of R^d. The above mathematical programming problem is said to be calm at $\bar{x}$ provided that there exist positive ϵ and M such that for all $(p,q) \in \epsilon B$, for all x in $\bar{x} + \epsilon B$ satisfying $g(x) + p \leq 0, h(x) + q = 0, x \in C$, one has

$$f(x) - f(\bar{x}) + M\|(p,q)\| \geq 0,$$

where B is the open unit ball in the appropriate space.

It is well known that the calmness condition is a constraint qualification for the existence of a KKT multiplier and the sufficient conditions for the calmness condition include the linear independence condition, the Slater condition, and the Mangasarian–Fromowitz condition. Moreover, the calmness condition is satisfied automatically in the case where the feasible region is a polyhedron.

3. Optimality conditions for OPCC. Let $(\bar{x}, \bar{y}, \bar{u}) \in \Omega$ and $g(\bar{x}, \bar{y}, \bar{u}) \leq 0$, $L(\bar{x}, \bar{y}, \bar{u}) = 0$. Let

$$\begin{aligned} L(\bar{u}) &:= \{1 \leq i \leq q : \bar{u}_i > 0\}, \\ I_+(\bar{x}, \bar{y}, \bar{u}) &:= \{1 \leq i \leq q : \bar{u}_i = 0, \psi_i(\bar{x}, \bar{y}, \bar{u}) < 0\}, \\ I_0(\bar{x}, \bar{y}, \bar{u}) &:= \{1 \leq i \leq q : \bar{u}_i = 0, \psi_i(\bar{x}, \bar{y}, \bar{u}) = 0\}. \end{aligned}$$

Where there is no confusion, we simply use L, I_+, I_0 instead of $L(\bar{u}), I_+(\bar{x}, \bar{y}, \bar{u})$, $I_0(\bar{x}, \bar{y}, \bar{u})$, respectively. It is clear that $\{1, 2, \ldots, q\} = L(\bar{u}) \cup I_+(\bar{x}, \bar{y}, \bar{u}) \cup I_0(\bar{x}, \bar{y}, \bar{u})$. Let

$$F = \left\{ (x,y,u) \in \Omega : \begin{array}{c} L(x,y,u) = 0, \ g(x,y,u) \leq 0 \\ \langle u, \psi(x,y,u) \rangle = 0, u \geq 0, \psi(x,y,u) \leq 0 \end{array} \right\}$$

be the feasible region of (OPCC). For any $I \subseteq \{1, 2, \ldots, q\}$, let

$$F_I := \left\{ (x,y,u) \in \Omega : \begin{array}{l} L(x,y,u) = 0, g(x,y,u) \leq 0 \\ u_i \geq 0, \ \psi_i(x,y,u) = 0 \ \forall i \in I \\ u_i = 0, \ \psi_i(x,y,u) \leq 0 \ \forall i \in \{1, 2, \ldots, q\} \backslash I \end{array} \right\}$$

denote a piece of the feasible region F.

Taking the "piecewise programming" approach in the terminology of [4], as in Corollary 2 of [5], we observe that the feasible region of the problem (OPCC) can be rewritten as a union of all pieces $F = \cup_{I \subseteq \{1,2,\ldots,q\}} F_I$. Therefore, a local solution $(\bar{x}, \bar{y}, \bar{u})$ for (OPCC) is also a local solution for each subproblem of minimizing the objective function f over a piece which contains the point $(\bar{x}, \bar{y}, \bar{u})$. Moreover, if $(\bar{x}, \bar{y}, \bar{u})$ is contained in all pieces and all subproblems are convex, then it is a global minimum for the original problem (OPCC). Hence the following proposition follows from this observation.

PROPOSITION 3.1. *Let $(\bar{x}, \bar{y}, \bar{u})$ be a local optimal solution to* (OPCC). *Suppose that f, g, ψ, L are locally Lipschitz near $(\bar{x}, \bar{y}, \bar{u})$ and Ω is closed. If for any given index set $\alpha \subseteq I_0$, the problem of minimizing f over $F_{\alpha \cup L}$ is calm in the sense of Definition* 2.9 *at $(\bar{x}, \bar{y}, \bar{u})$, then there exist $\xi \in R^l$, $\zeta \in R^d$, $\eta \in R^q, \gamma \in R^q$ such that*

$$0 \in \hat{\partial} f(\bar{x}, \bar{y}, \bar{u}) + \sum_{i=1}^{l} \xi_i \hat{\partial} L_i(\bar{x}, \bar{y}, \bar{u}) + \sum_{i=1}^{d} \zeta_i \hat{\partial} g_i(\bar{x}, \bar{y}, \bar{u}) + \hat{N}((\bar{x}, \bar{y}, \bar{u}), \Omega)$$

$$-\sum_{i=1}^{q} \eta_i \hat{\partial} \psi_i(\bar{x}, \bar{y}, \bar{u}) + \{(0, 0, \gamma)\}, \tag{3.1}$$

$$\zeta \geq 0, \qquad \langle \zeta, g(\bar{x},\bar{y},\bar{u})\rangle = 0, \tag{3.2}$$

$$\eta_{I_0\backslash\alpha} \leq 0, \quad \eta_{I_+} = 0, \quad \gamma_L = 0, \quad \gamma_\alpha \leq 0. \tag{3.3}$$

Conversely, let $(\bar{x},\bar{y},\bar{u})$ *be a feasible solution for* (OPCC), *and for all index sets* $\alpha \subseteq I_0$*, there exist* $\xi \in R^l$*,* $\zeta \in R^d$*,* $\eta \in R^q, \gamma \in R^q$ *such that* (3.1)–(3.3) *are satisfied. If* f *is either convex or pseudoconvex,* g *is convex,* ψ, L *are affine, and* Ω *is convex, then* $(\bar{x},\bar{y},\bar{u})$ *is a minimum of* f *over all* $(x,y,u) \in \cup_{\alpha\subseteq I_0} F_{\alpha\cup L}$*. If in addition to the above assumptions* $I_0 = \{1,2,\ldots,q\}$*, then* $(\bar{x},\bar{y},\bar{u})$ *is a global solution for* (OPCC).

Proof. It is obvious that the feasible region of (OPCC) can be represented as the union of pieces $F = \cup_{I\subseteq\{1,2,\ldots,q\}} F_I$. Since $\bar{u}_i > 0 \ \forall i \in L(\bar{u})$ and $\psi_i(\bar{x},\bar{y},\bar{u}) < 0$ $\forall i \in I_+(\bar{x},\bar{y},\bar{u})$, and

$$F_{\alpha\cup L} = \left\{ (x,y,u) \in \Omega : \begin{array}{l} L(x,y,u) = 0, g(x,y,u) \leq 0 \\ u_i \geq 0, \ \psi_i(x,y,u) = 0 \ \forall i \in \alpha \\ u_i \geq 0, \ \psi_i(x,y,u) = 0 \ \forall i \in L \\ u_i = 0, \ \psi_i(x,y,u) \leq 0 \ \forall i \in I_+ \\ u_i = 0, \ \psi_i(x,y,u) \leq 0 \ \forall i \in I_0\backslash\alpha \end{array} \right\},$$

we have

$$(\bar{x},\bar{y},\bar{u}) \in \cap_{\alpha\subseteq I_0} F_{\alpha\cup L}$$

and

$$(\bar{x},\bar{y},\bar{u}) \notin F\backslash(\cup_{\alpha\subseteq I_0} F_{\alpha\cup L}).$$

Hence if $(\bar{x},\bar{y},\bar{u})$ is optimal for (OPCC), then for any given index set $\alpha \subseteq I_0, (\bar{x},\bar{y},\bar{u})$ is also a minimum for f over $F_{\alpha\cup L}$. Since this problem is calm, by the well-known nonsmooth necessary optimality condition (see, e.g., [1, 2, 3]), there exist $\xi \in R^l$, $\zeta \in R^d$, $\eta \in R^q$, $\gamma \in R^q$ such that (3.1)–(3.3) are satisfied. Conversely, suppose that for each $\alpha \subseteq I_0$ there exist $\xi \in R^l$, $\zeta \in R^d$, $\eta \in R^q$, $\gamma \in R^q$ such that (3.1)–(3.3) are satisfied and the problem is convex. By virtue of Remarks 2.1 and 2.2, the limiting subgradients and the limiting normal cones coincide with the subgradients and the normal cone in the sense of convex analysis, respectively. Hence, by the standard first-order sufficient optimality conditions, $(\bar{x},\bar{y},\bar{u})$ is a minimum of f over $F_{\alpha\cup L}$ for each $\alpha \subseteq I_0$ and hence is a minimum of f over $\cup_{\alpha\subseteq I_0} F_{\alpha\cup L}$. In the case when $I_0 = \{1,2,\ldots,q\}$, $L = \emptyset$ and the feasible region $F = \cup_{\alpha\subseteq I_0} F_{\alpha\cup L}$. Hence $(\bar{x},\bar{y},\bar{u})$ is a global optimal for (OPCC) in this case. The proof of the proposition is now complete. □

Remark 3.1. *The necessary part of the above proposition with smooth problem data is given by Luo, Pang, and Ralph in* [4] *under the so-called "basic constraint qualification."*

Note that the multipliers in Proposition 3.1 depend on the index set α through (3.3). However, if for some pair of index sets α $(\subseteq I_0)$ and $I_0\backslash\alpha$, the components $(\eta_{I_0}, \gamma_{I_0})$ of the multipliers are the same, then we would have a necessary condition that does not depend on the index set α. In this case the necessary condition turns out to be the necessary condition involving the proximal coderivatives as in (b) of the following theorem.

Theorem 3.2. *Suppose* f, g, L, ψ *are continuously differentiable. Then the following three conditions are equivalent:*

(a) *There exist* $\xi \in R^l$, $\zeta \in R^d$, $\eta, \gamma \in R^q$ *such that*

$$0 = \nabla f(\bar{x},\bar{y},\bar{u}) + \sum_{i=1}^{l} \xi_i \nabla L_i(\bar{x},\bar{y},\bar{u}) + \sum_{i=1}^{d} \zeta_i \nabla g_i(\bar{x},\bar{y},\bar{u})$$

$$- \sum_{i=1}^{q} \eta_i \nabla_i \psi_i(\bar{x},\bar{y},\bar{u}) + \{(0,0,\gamma)\}, \tag{3.4}$$

$$\zeta \geq 0, \qquad \langle \zeta, g(\bar{x},\bar{y},\bar{u}) \rangle = 0, \tag{3.5}$$

$$\eta_{I_0} \leq 0, \quad \eta_{I_+} = 0, \quad \gamma_L = 0, \quad \gamma_{I_0} \leq 0. \tag{3.6}$$

(b) *There exist* $\xi \in R^l$, $\zeta \in R^d$, $\eta \in R^q$ *such that*

$$0 = \nabla f(\bar{x},\bar{y},\bar{u}) + \sum_{i=1}^{l} \xi_i \nabla L_i(\bar{x},\bar{y},\bar{u}) + \sum_{i=1}^{d} \zeta_i \nabla g_i(\bar{x},\bar{y},\bar{u})$$

$$- \sum_{i=1}^{q} \eta_i \nabla \psi_i(\bar{x},\bar{y},\bar{u}) + \{0\} \times \{0\} \times D^*_\pi N_{R^q_+}(\bar{u}, \psi(\bar{x},\bar{y},\bar{u}))(\eta), \tag{3.7}$$

$$\zeta \geq 0, \qquad \langle \zeta, g(\bar{x},\bar{y},\bar{u}) \rangle = 0. \tag{3.8}$$

(c) *There exist* $\xi \in R^l$, $\zeta \in R^d$, $\eta, \gamma \in R^q$, $\alpha, \beta \in R^{2q}_+$ *such that* (3.4) *and* (3.5) *are satisfied and*

$$\begin{aligned} 0 &= \sum_{i=1}^{q} \bar{u}_i(\alpha_i + \beta_i) - \sum_{i=1}^{q} \psi_i(\bar{x},\bar{y},\bar{u})(\alpha_{q+i} + \beta_{q+i}), \\ \eta_i &= -\alpha_{q+i} + \bar{u}_i \beta_{q+i} \quad \forall i = 1,2,\ldots,q, \\ \gamma_i &= -\alpha_i + \psi_i(\bar{x},\bar{y},\bar{u})\beta_i \quad \forall i = 1,2,\ldots,q. \end{aligned}$$

Let $(\bar{x},\bar{y},\bar{u})$ *be a local optimal solution to* (OPCC), *where* $\Omega = R^{n+m+q}$. *Suppose that there exists an index set* $\alpha \subseteq I_0$ *such that the problem of minimizing* f *over* $F_{\alpha \cup L}$ *and the problem of minimizing* f *over* $F_{(I_0 \setminus \alpha) \cup L}$ *are calm. Furthermore, suppose that*

$$0 = \sum_{i=1}^{l} \xi_i \nabla L_i(\bar{x},\bar{y},\bar{u}) + \sum_{i=1}^{d} \zeta_i \nabla g_i(\bar{x},\bar{y},\bar{u}) - \sum_{i=1}^{q} \eta_i \nabla \psi_i(\bar{x},\bar{y},\bar{u}) + \{(0,0,\gamma)\}, \tag{3.9}$$

$$0 = \langle \zeta, g(\bar{x},\bar{y},\bar{u}) \rangle, \quad \eta_{I_+} = 0, \quad \gamma_L = 0 \tag{3.10}$$

implies that $\eta_{I_0} = 0, \gamma_{I_0} = 0$. *Then the three equivalent conditions* (a)–(c) *hold. Conversely, let* $(\bar{x},\bar{y},\bar{u})$ *be a feasible solution to* (OPCC), *where* $\Omega = R^{n+m+q}$ *and let* f *be pseudoconvex,* g *be convex,* ψ, L *be affine. If one of the equivalent conditions* (a)–(c) *holds, then* $(\bar{x},\bar{y},\bar{u})$ *is a minimum of* f *over all* $(x,y,u) \in \cup_{\alpha \subseteq I_0} F_{\alpha \cup L}$. *If in addition to the above assumptions* $I_0 = \{1,2,\ldots,q\}$, *then* $(\bar{x},\bar{y},\bar{u})$ *is a global solution for* (OPCC).

Proof. By the definition of the proximal coderivatives (Definition 2.3),

$$\gamma \in D^*_\pi N_{R^q_+}(\bar{u}, \psi(\bar{x},\bar{y},\bar{u}))(\eta)$$

if and only if

$$(\gamma, -\eta) \in N^\pi((\bar{u}, \psi(\bar{x},\bar{y}), GrN^q_+).$$

Hence the equivalence of condition (a) and condition (b) follows from Proposition 2.7. The equivalence of condition (b) and condition (c) follows from Proposition 2.8.

Let $(\bar{x}, \bar{y}, \bar{u})$ be a local optimal solution to (OPCC), where $\Omega = R^{n+m+q}$. Then it is also a local optimal solution to the problem of minimizing f over $F_{\alpha \cup L}$ and the problem of minimizing f over $F_{(I_0 \backslash \alpha) \cup L}$. By the calmness assumption for these two problems, there exist $\xi^i \in R^l$, $\zeta^i \in R^d$, $\eta^i \in R^q$, $\gamma^i \in R^q, i = 1, 2$, satisfying (3.1)–(3.3), which implies that

$$\begin{aligned} 0 = &\sum_{i=1}^{l} (\xi_i^1 - \xi_i^2) \nabla L_i(\bar{x}, \bar{y}, \bar{u}) + \sum_{i=1}^{d} (\zeta_i^1 - \zeta_i^2) \nabla g_i(\bar{x}, \bar{y}, \bar{u}) \\ &\quad - \sum_{i=1}^{q} (\eta_i^1 - \eta_i^2) \nabla \psi_i(\bar{x}, \bar{y}, \bar{u}) + \{(0, 0, \gamma^1 - \gamma^2)\}, \\ 0 = &\langle \zeta^1 - \zeta^2, g(\bar{x}, \bar{y}, \bar{u}) \rangle, \quad (\eta^1 - \eta^2)_{I_+} = 0, \quad (\gamma^1 - \gamma^2)_L = 0. \end{aligned}$$

By the assumption we arrive at $\eta_{I_0}^1 = \eta_{I_0}^2, \gamma_{I_0}^1 = \gamma_{I_0}^2$. Since by (3.3), $\eta_{I_0 \backslash \alpha}^1 \leq 0, \gamma_\alpha^1 \leq 0$ and $\eta_\alpha^2 \leq 0, \gamma_{I_0 \backslash \alpha}^2 \leq 0$, we have

$$\eta_{I_0}^1 = \eta_{I_0}^2 \leq 0, \qquad \gamma_{I_o}^1 = \gamma_{I_0}^2 \leq 0.$$

That is, condition (a) holds.

The sufficient part of the theorem follows from the sufficient part of Proposition 3.1. □

As observed in [4, Proposition 4.3.5], the necessary optimality conditions (3.4)–(3.6) happen to be the KKT condition for the relaxed problem

$$\begin{aligned} \text{(RP)} \qquad & \min f(x, y, u) \\ & \text{s.t. } u_i \geq 0, \quad \psi_i(x, y, u) = 0 \quad \forall i \in L(\bar{u}), \\ & \qquad u_i = 0, \quad \psi_i(x, y, u) \leq 0 \quad \forall i \in I_+(\bar{x}, \bar{y}, \bar{u}), \\ & \qquad u_i \geq 0, \quad \psi_i(x, y, u) \leq 0 \quad \forall i \in I_0(\bar{x}, \bar{y}, \bar{u}), \\ & \qquad L(x, y, u) = 0, \qquad g(x, y, u) \leq 0, \end{aligned}$$

and $(\xi, \zeta, \eta, \gamma)$ satisfies (3.4)–(3.6) if and only if it satisfies the KKT condition for the subproblem of minimizing f over the feasible region $F_{\alpha \cup L}$, i.e., (3.1)–(3.3) with the smooth problem data and $\Omega = R^{n+m+q}$, for all index sets $\alpha \subseteq I_0(\bar{x}, \bar{y}, \bar{u})$. Consequently, if the strict Mangasarian–Fromovitz constraint qualification (SMFCQ) holds for problem (RP) at $(\xi, \zeta, \eta, \gamma)$ which satisfies (3.4)–(3.6), then $(\xi, \zeta, \eta, \gamma)$ is the unique multiplier which satisfies (3.4)–(3.6). Since the index sets α only affect the $(\eta_{I_0}, \gamma_{I_0})$ components of the multiplier $(\xi, \zeta, \eta, \gamma)$, we observe that the existence of multipliers satisfying (3.4)–(3.6) is equivalent to the existence of multipliers satisfying (3.1)–(3.3) for all index sets $\alpha \subseteq I_0(\bar{x}, \bar{y}, \bar{u})$ with the components $(\eta_{I_0}, \gamma_{I_0})$ having the same sign. From the proof of Theorem 3.2, it is easy to see that the condition that no nonzero vectors satisfy (3.9)–(3.10) is a sufficient condition for the existence of common $(\eta_{I_0}, \gamma_{I_0})$ components of the multiplier $(\xi, \zeta, \eta, \gamma)$ for all index sets $\alpha \subseteq I_0(\bar{x}, \bar{y}, \bar{u})$. Hence this condition refines the sufficient condition of a unique multiplier such as the SMFCQ for the relaxed problem proposed in [4, Proposition 4.3.5].

We now give an example which does not have a unique multiplier satisfying (3.4)–(3.6) but does satisfy the condition proposed in Theorem 3.2.

Example 3.1 (see [4, Example 4.3.6]). *Consider the following OPCC:*

$$\begin{array}{rl} \textit{minimize} & x_3 + u_1 + u_2 \\ \textit{s.t.} & u \geq 0, \qquad \psi(x,u) := (-x_1 - u_1, -x_2 - u_2) \leq 0, \\ & \langle u, \psi(x,u)\rangle = 0, \\ & x_3 \geq 0, \qquad 2x_3 \geq 0. \end{array}$$

$(\bar{x}, \bar{u}) = (\bar{x}_1, \bar{x}_2, 0, 0, 0)$*, where* $\bar{x}_1, \bar{x}_2$ *are any real numbers, are obviously solutions to the above problem. As pointed out in* [4, *Example* 4.3.6]*, SMFCQ does not hold for this problem. However, we can verify that it satisfies our condition. Indeed, the equation* (3.9) *for this problem is*

$$\begin{aligned} 0 = {} & \zeta_1(0,0,-1,0,0) + \zeta_2(0,0,-2,0,0) - \eta_1(-1,0,0,-1,0) \\ & -\eta_2(0,-1,0,0,-1) + (0,0,0,\gamma_1,\gamma_2), \end{aligned}$$

which implies that $\eta = 0, \gamma = 0$.

Moreover, the calmness condition is satisfied since the constraint region for each subproblem $F_{\alpha\cup L}$ *is a polyhedron due to the fact that* ψ *and* g *are both affine. Hence by Theorem* 3.2*, if* $(\bar{x}, \bar{u})$ *is a local minimum to the above problem, then there exist* ζ, η, γ *such that*

$$\begin{aligned} 0 = {} & (0,0,1,1,1) + \zeta_1(0,0,-1,0,0) + \zeta_2(0,0,-2,0,0) \\ & -\eta_1(-1,0,0,-1,0) - \eta_2(0,-1,0,0,-1) + (0,0,0,\gamma_1,\gamma_2), \\ & \zeta \geq 0, \quad \zeta_1 \bar{x}_3 = 0, \quad 2\zeta_2 \bar{x}_3 = 0, \\ & \eta_{I_0} \leq 0, \quad \eta_{I_+} = 0, \quad \gamma_L = 0, \quad \gamma_{I_0} \leq 0, \end{aligned}$$

which implies $\eta_1 = \eta_2 = 0$*,* $\gamma_1 = \gamma_2 = 1$*, and* $\bar{x}_3 = 0$*. Since* $I_0(\bar{x}, \bar{u}) = \{1, 2\}$ *for* $(\bar{x}, \bar{u}) = 0$*,* 0 *is a global optimal solution according to Theorem* 3.2 *and* $(\bar{x}, 0, 0)$ *with* $\bar{x} \neq 0$ *are local optimal solutions.*

To illustrate the application of the result obtained, we now consider the following bilevel programming problem (BLQP), where the lower level problem is linear quadratic:

$$\begin{array}{lrl} \text{(BLQP)} & \min & f(x,y) \\ & \text{s.t.} & y \in S(x), \\ & & Gx + Hy + a \leq 0, \end{array}$$

where G and H are $l \times n$ and $l \times m$ matrices, respectively, $a \in R^l$, and $S(x)$ is the solution set of the quadratic programming problem with parameter x:

$$\begin{array}{lrl} (\text{QP}_x) & \min & \langle y, Px\rangle + \dfrac{1}{2}\langle y, Qy\rangle + p^t x + q^t y \\ & \text{s.t.} & Dx + Ey + b \leq 0, \end{array}$$

where $Q \in R^{m\times m}$ is a symmetric and positive semidefinite matrix, $p \in R^n$, $q \in R^m$, $P \in R^{m\times n}$, D and E are $q \times n$ and $q \times m$ matrices, respectively, and $b \in R^q$.

Replacing the bilevel constraint by the KKT condition for the lower level problem, it is easy to see that (BLQP) is equivalent to the problem

$$\begin{array}{lrl} \text{(KKT)} & \min & f(x,y) \\ & \text{s.t.} & \langle Dx + Ey + b, u\rangle = 0, \quad u \geq 0, \quad Dx + Ey + b \leq 0, \\ & & Qy + Px + q + E^\top u = 0, \\ & & Gx + Hy + a \leq 0, \end{array}$$

which is an OPCC. Let $(\bar{x}, \bar{y})$ be an optimal solution of (BLQP) and $\bar{u}$ a corresponding multiplier; i.e,

$$0 = Q\bar{y} + P\bar{x} + q + E^\top \bar{u}, \tag{3.11}$$

$$\langle D\bar{x} + E\bar{y} + b, \bar{u}\rangle = 0, \qquad u \geq 0. \tag{3.12}$$

Then

$$\begin{aligned} L &= \{1 \leq i \leq q : \bar{u}_i > 0\}, \\ I_+ &= \{1 \leq i \leq q : \bar{u}_i = 0, (D\bar{x} + E\bar{y} + b)_i < 0\}, \\ I_0 &= \{1 \leq i \leq q : \bar{u}_i = 0, (D\bar{x} + E\bar{y} + b)_i = 0\}. \end{aligned}$$

The feasible region of problem (KKT) is

$$F = \left\{ (x, y, u) \in R^{n+m+q} : \begin{array}{l} Qy + Px + q + E^\top u = 0,\ Gx + Hy + a \leq 0 \\ \langle u, Dx + Ey + b\rangle = 0, u \geq 0, Dx + Ey + b \leq 0 \end{array} \right\},$$

and for any $I \subseteq \{1, 2, \ldots, q\}$,

$$F_I = \left\{ (x, y, u) \in R^{n+m+q} : \begin{array}{l} Qy + Px + q + E^\top u = 0,\ Gx + Hy + a \leq 0 \\ u_i \geq 0,\ (Dx + Ey + b)_i = 0\ \forall i \in I \\ u_i = 0,\ (Dx + Ey + b)^i \leq 0\ \forall i \in \{1, 2, \ldots, q\} \backslash I \end{array} \right\}.$$

Since $F_{\alpha \cup L}$ for any index set $\alpha \subseteq I_0$ has linear constraints only, the problem of minimizing f over $F_{\alpha \cup L}$ is calm. Hence the following result follows from Proposition 3.1.

COROLLARY 3.3. *Let $(\bar{x}, \bar{y})$ be an optimal solution of* (BLQP) *and $\bar{u}$ a corresponding multiplier. Suppose that f is locally Lipschitz near $(\bar{x}, \bar{y})$. Then for each $\alpha \subseteq I_0$, there exist $\xi \in R^m$, $\zeta \in R^d$, $\eta \in R^q$ such that*

$$\begin{aligned} &0 \in \hat{\partial} f(\bar{x}, \bar{y}) + \{P^\top \xi\} \times Q^\top \xi + \{G^\top \zeta\} \times \{H^\top \zeta\} - \{D^\top \eta\} \times \{E^\top \eta\}, \\ &\zeta \geq 0, \qquad \langle G\bar{x} + H\bar{y} + a, \zeta\rangle = 0, \\ &\eta_\alpha \leq 0, \quad \eta_{I_+} = 0, \quad (E\xi)_L = 0, \quad (E\xi)_\alpha \geq 0. \end{aligned}$$

If f is either convex or pseudoconvex, then the above necessary condition is also sufficient for a feasible solution $(\bar{x}, \bar{y}, \bar{u})$ of (KKT) *to be a minimum of f over all $(x, y, u) \in \cup_{\alpha \subseteq I_0} F_{\alpha \cup L}$. In particular, if f is either convex or pseudoconvex and $I_0 = \{1, 2, \ldots, q\}$, then the above condition is sufficient for a feasible solution $(\bar{x}, \bar{y})$ to be a global optimum for* (BLQP).

The following result follows from Theorem 3.2.

COROLLARY 3.4. *Let $(\bar{x}, \bar{y})$ be an optimal solution of* (BLQP) *and $\bar{u}$ a corresponding multiplier. Suppose that f is C^1 and*

$$0 = P^\top \xi + G^\top \zeta - D^\top \eta, \tag{3.13}$$

$$0 = Q^\top \xi + H^\top \zeta - E^\top \eta, \tag{3.14}$$

$$0 = \langle G\bar{x} + H\bar{y} + a, \zeta\rangle, \tag{3.15}$$

$$\eta_{I_+} = 0, \qquad (E\xi)_L = 0$$

implies $\eta_{I_0} = (E\xi)_{I_0} = 0$. Then there exist $\xi \in R^m$, $\zeta \in R^d$, $\eta \in R^q$ such that

$$0 = \nabla f(\bar{x}, \bar{y}) + \{P^\top \xi\} \times Q^\top \xi + \{G^\top \zeta\} \times \{H^\top \zeta\} - \{D^\top \eta\} \times \{E^\top \eta\}, \tag{3.16}$$

$$\zeta \geq 0, \qquad \langle G\bar{x} + H\bar{y} + a, \zeta\rangle = 0, \tag{3.17}$$

$$\eta_{I_0} \leq 0, \quad \eta_{I_+} = 0, \quad (E\xi)_L = 0, \quad (E\xi)_{I_0} \geq 0.$$

Equivalently, there exist $\xi \in R^l$, $\zeta \in R^d$, $\eta \in R^q$ *such that* (3.16)–(3.17) *are satisfied and*

$$(-E\xi, -\eta) \in N^{\pi}((\bar{u}, D\bar{x} + E\bar{u} + b), GrN_{R^q_+}).$$

Equivalently, there exist $\xi \in R^l$, $\zeta \in R^d$, $\eta \in R^q$, $\alpha, \beta \in R^{2q}_+$ *such that* (3.16)–(3.17) *are satisfied and*

$$\begin{aligned} 0 &= \sum_{i=1}^{q} \bar{u}_i(\alpha_i + \beta_i) - \sum_{i=1}^{q}(D\bar{x} + E\bar{y} + b)_i(\alpha_{q+i} + \beta_{q+i}), \\ \eta_i &= -\alpha_{q+i} + \bar{u}_i\beta_{q+i} \; \forall i = 1, 2, \ldots, q, \\ (E\xi)_i &= \alpha_i - (D\bar{x} + E\bar{y} + b)_i\beta_i \; \forall i = 1, 2, \ldots, q. \end{aligned}$$

Conversely, let $(\bar{x}, \bar{y})$ *be any vector in* R^{n+m} *satisfying the constraints* $G\bar{x}+H\bar{y}+a \le 0$ *and* $D\bar{x} + E\bar{y} + b \le 0$ *and* f *be pseudoconvex. If there exists* $\bar{u} \in R^q$ *that satisfies* (3.11)–(3.12) *such that one of the above equivalent conditions holds, then* $(\bar{x}, \bar{y}, \bar{u})$ *is a minimum of* f *over all* $(x, y, u) \in \cup_{\alpha \subseteq I_0} F_{\alpha \cup L}$. *In addition to the above assumptions, if* $I_0 = \{1, 2, \ldots, q\}$, *then* $(\bar{x}, \bar{y})$ *is a global minimum for* (BLQP).

Acknowledgments. The author would like to thank Dr. Qing Lin for a helpful discussion of Proposition 2.8.

REFERENCES

[1] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley-Interscience, New York, 1983; reprinted by SIAM, Philadelphia, 1990.

[2] F .H. Clarke, *Methods of Dynamic and Nonsmooth Optimization*, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 57, SIAM, Philadelphia, 1989.

[3] P. D. Loewen, *Optimal Control via Nonsmooth Analysis*, CRM Proceedings and Lecture Notes, AMS, Providence, RI, 1993.

[4] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, London, UK, 1996.

[5] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Piecewise Sequential Quadratic Programming for Mathematical Programs with Nonlinear Complementarity Constraints*, in Multilevel Optimization: Algorithms and Applications, Nonconvex Optim. Anal. 20, Kluwer Academic Publishers, Norwell, MA, 1998.

[6] B. S. Mordukhovich, *Generalized differential calculus for nonsmooth and set-valued mappings*, J. Math. Anal. Appl., 183 (1994), pp. 250–288.

[7] J. J. Ye and X. Y. Ye, *Necessary optimality conditions for optimization problems with variational inequality constraints*, Math. Oper. Res., 22 (1997), pp. 977–997.

[8] J. J. Ye and D. L. Zhu, *Optimality conditions for bilevel programming problems*, Optimization, 33 (1995), pp. 9–27.

[9] J. J. Ye, D. L. Zhu, and Q. J. Zhu. *Exact penalization and necessary optimality conditions for generalized bilevel programming problems*, SIAM J. Optim., 7 (1997), pp. 481–507.

SIAM J. OPTIM.
Vol. 9, No. 2, pp. 388–413

A NEW UNCONSTRAINED DIFFERENTIABLE MERIT FUNCTION FOR BOX CONSTRAINED VARIATIONAL INEQUALITY PROBLEMS AND A DAMPED GAUSS–NEWTON METHOD[*]

DEFENG SUN[†] AND ROBERT S. WOMERSLEY[†]

Abstract. In this paper we propose a new unconstrained differentiable merit function f for box constrained variational inequality problems $\mathrm{VIP}(l, u, F)$. We study various desirable properties of this new merit function f and propose a Gauss–Newton method in which each step requires only the solution of a system of linear equations. Global and superlinear convergence results for $\mathrm{VIP}(l, u, F)$ are obtained. Key results are the boundedness of the level sets of the merit function for any uniform P-function and the superlinear convergence of the algorithm without a nondegeneracy assumption. Numerical experiments confirm the good theoretical properties of the method.

Key words. variational inequality problems, box constraints, merit functions, Gauss–Newton method, superlinear convergence

AMS subject classifications. 90C33, 90C30, 65H10

PII. S1052623496314173

1. Introduction.

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a continuously differentiable mapping and S be a nonempty closed convex set in $\mathbb{R}^n$. The variational inequality problem, denoted by $\mathrm{VIP}(S, F)$, is to find a vector $x \in S$ such that

$$F(x)^T(y - x) \geq 0 \quad \text{for all } y \in S. \tag{1.1}$$

A box constrained variational inequality problem, denoted $\mathrm{VIP}(l, u, F)$, has

$$S = \{x \in \mathbb{R}^n | \ l \leq x \leq u\}, \tag{1.2}$$

where $l_i \in \mathbb{R} \cup \{-\infty\}$, $u_i \in \mathbb{R} \cup \{+\infty\}$, and $l_i < u_i$, $i = 1, \ldots, n$. In some papers, e.g., [2, 7], $\mathrm{VIP}(l, u, F)$ is called the mixed complementarity problem. Further, if $S = \mathbb{R}^n_+$, $\mathrm{VIP}(S, F)$ reduces to the nonlinear complementarity problem, denoted $\mathrm{NCP}(F)$, which is to find $x \in \mathbb{R}^n$ such that

$$x \geq 0, \quad F(x) \geq 0, \quad x^T F(x) = 0. \tag{1.3}$$

Two comprehensive surveys of variational inequality problems and nonlinear complementarity problems are [23] and [34].

Recently much effort has been made to derive *merit functions* for $\mathrm{VIP}(S, F)$ and then to use these functions to develop solution methods. Formally, we say that a function $h : X \to [0, \infty)$ is a merit function for $\mathrm{VIP}(S, F)$ on a set X (typically $X = \mathbb{R}^n$ or $X = S$) provided $h(x) \geq 0$ for all $x \in X$ and $x \in X$ satisfies (1.1) if and only if $h(x) = 0$. Then, we may reformulate $\mathrm{VIP}(S, F)$ as the minimization problem

$$\min_{x \in X} \quad h(x). \tag{1.4}$$

[*] Received by the editors December 27, 1996; accepted for publication (in revised form) April 17, 1998; published electronically March 17, 1999. This research was supported by the Australian Research Council.

http://www.siam.org/journals/siopt/9-2/31417.html

[†] School of Mathematics, University of New South Wales, Sydney, NSW 2052, Australia (sun@maths.unsw.edu.au, R.Womersley@unsw.edu.au).

Recent developments of this area are summarized in [20].

It is well known [9] that $x \in \mathbb{R}^n$ solves $\mathrm{VIP}(S, F)$ if and only if x is a solution of the equation

$$H(x) := x - \Pi_S[x - \alpha^{-1}F(x)] = 0 \tag{1.5}$$

for an arbitrary positive constant α. Here Π_S is the orthogonal projection operator onto S. An obvious merit function for (1.1) is

$$h(x) := \frac{1}{2}\|H(x)\|^2. \tag{1.6}$$

We can find a solution of (1.1) by solving (1.4) with $X = \mathbb{R}^n$ or $X = S$. Unfortunately the function h defined by (1.5) and (1.6) is not continuously differentiable, so gradient-based methods cannot be used directly. Nevertheless global and superlinear convergence properties have been obtained under some regularity conditions [32, 33, 35, 36]. Another approach based on nonsmooth equations and nonsmooth merit functions is Ralph's path search method [41, 7].

Recent interests have focused on (unconstrained) differentiable merit functions. Early differentiable merit functions such as the regularized gap function [19] are constrained ones. By applying the Moreau–Yosida regularization to some gap functions, Yamashita and Fukushima [48] proposed unconstrained differentiable merit functions for (1.1). These functions possess nice theoretical properties but are not easy to evaluate in general. Peng [37] showed that the difference of two regularized gap functions constitutes an unconstrained differentiable merit function for $\mathrm{VIP}(S, F)$. Later, Yamashita, Taji, and Fukushima [49] extended the idea of Peng [37] and investigated some important properties related to this merit function. Specifically, the latter authors considered the function $h_{\alpha\beta} : \mathbb{R}^n \to \mathbb{R}$ defined by

$$h_{\alpha\beta}(x) := f_\alpha(x) - f_\beta(x), \tag{1.7}$$

where α and β are arbitrary positive parameters such that $\alpha < \beta$ and f_α is the regularized gap function

$$f_\alpha(x) := \max_{y \in S}\left\{F(x)^T(x - y) - \frac{\alpha}{2}\|x - y\|^2\right\}. \tag{1.8}$$

(The function f_β is defined similarly with α replaced by β.) In the special case $\beta = \alpha^{-1}$ and $\alpha < 1$ in (1.7), the function $h_{\alpha\beta}$ reduces to the merit function studied by Peng [37]. The function $h_{\alpha\beta}$ defined by (1.7) is called the D-gap function. Based on this merit function, globally and superlinearly convergent Newton-type methods for solving $\mathrm{VIP}(S, F)$ have been proposed under the assumption that F is a strongly monotone function [44]. It was pointed out by Peng and Yuan [38] that when $S = \mathbb{R}^n_+$, $\beta = \alpha^{-1}$, and $0 < \alpha < 1$, the function $h_{\alpha\beta}$ is actually the implicit Lagrangian function

$$\begin{aligned} m_\alpha(x) := x^T F(x) + \frac{\alpha}{2}\big(&\|[x - \alpha^{-1}F(x)]_+\|^2 - \|x\|^2 \\ &+ \|[F(x) - \alpha^{-1}x]_+\|^2 - \|F(x)\|^2\big) \end{aligned} \tag{1.9}$$

introduced by Mangasarian and Solodov [30] for the nonlinear complementarity problem (1.3). Here $[z]_+$ denotes the vector with components $\max\{z_i, 0\}, i = 1, \ldots, n$. The function $m_\alpha(x)$ is one of the many unconstrained differentiable merit functions for $\mathrm{NCP}(F)$ and its various properties were further studied in [11, 24, 28, 37, 46, 47, 49].

Another well-studied unconstrained differentiable merit function for NCP(F) has the form

$$\theta(x) := \frac{1}{2}\sum_{i=1}^{n} \phi(x_i, F_i(x))^2, \tag{1.10}$$

where $\phi : \mathbb{R}^2 \to \mathbb{R}$ is the function

$$\phi(a, b) := \sqrt{a^2 + b^2} - (a + b) \tag{1.11}$$

introduced by Fischer [16] but attributed to Burmeister and called the Fischer–Burmeister function. This merit function θ has been much studied and used in solving nonlinear complementarity problems [6, 13, 14, 18, 21, 24, 25, 26, 45] (see [17] for a survey). In particular, based on this merit function, globally and superlinearly convergent Newton-type methods for NCP(F) were given in [6] under the assumption that F is a uniform P-function, which is a weaker condition than the assumption that F is a strongly monotone function. Unlike the implicit Lagrangian function m_α, the nice properties of the merit function θ based on the Fischer–Burmeister function cannot be naturally generalized to VIP(S, F).

In this paper we study new unconstrained merit functions for the box constrained variational inequality problem VIP(l, u, F) where S is of the form (1.2). Despite its special structure, VIP(l, u, F) has many applications in engineering, economics, and sciences. An available unconstrained differentiable merit function for VIP(l, u, F) is the D-gap function $h_{\alpha\beta}$. However, when reduced to NCP(F), the D-gap function $h_{\alpha\beta}$ with $\beta = \alpha^{-1}$ and $\alpha \in (0, 1)$ becomes the implicit Lagrangian function m_α. This merit function suffers from the drawback that it needs more restrictive assumptions to get globally and superlinearly convergent methods for NCP(F) than the merit function $\theta(x)$ based on the Fischer–Burmeister function does. This motivates the investigation of other unconstrained differentiable merit functions which need less restrictive assumptions. Throughout this paper we adopt the convention that $\pm\infty \times 0 = 0$. Then it is easy to see that VIP(l, u, F) is equivalent to its Karush–Kuhn–Tucker (KKT) system

$$\begin{array}{c} v - w = F(x), \\ x_i - l_i \geq 0, \quad v_i \geq 0, \quad (x_i - l_i)v_i = 0, \quad i = 1, \ldots, n, \\ u_i - x_i \geq 0, \quad w_i \geq 0, \quad (u_i - x_i)w_i = 0, \quad i = 1, \ldots, n. \end{array} \tag{1.12}$$

If $x \in \mathbb{R}^n$ solves VIP(l, u, F), then $(x, v, w) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$ with $v = [F(x)]_+$ and $w = [-F(x)]_+$ solves the KKT system (1.12). Conversely, if $(x, v, w) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$ solves the KKT system (1.12), then x solves VIP(l, u, F). Define $E : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{3n}$ as

$$E(x, v, w) := \begin{pmatrix} v - w - F(x) \\ \phi(x_i - l_i, v_i), \ i = 1, \ldots, n \\ \phi(u_i - x_i, w_i), \ i = 1, \ldots, n \end{pmatrix}.$$

Then an obvious unconstrained differentiable merit function for the KKT system (1.12) is

$$\xi(x, v, w) := \frac{1}{2}\|E(x, v, w)\|^2.$$

The merit function ξ for $\mathrm{VIP}(l, u, F)$ has many good properties; for example, see [10]. However, it also suffers from several drawbacks. A disadvantage of this merit function is that the level sets $L_c(\xi)$ of ξ are in general not bounded for all nonnegative numbers c. Here the level sets $L_c(g)$ of $g : \mathbb{R}^m \to \mathbb{R}$ are

$$L_c(g) := \{z \in \mathbb{R}^m |\ g(z) \leq c\}.$$

This can easily be shown by taking $l_i = -1$, $u_i = 1$, and $v_i = w_i \to \infty$, $i = 1, \ldots, n$. The unboundedness of the level sets could allow the sequence of iterates to diverge to infinity. This unfavorable property is caused by introducing the variables v and w. So it appears better to consider $\mathrm{VIP}(l, u, F)$ in its original space instead of in the larger-dimensional space.

We propose a new merit function which has bounded level sets for any uniform P-function (see section 3) and establish superlinear convergence of a damped Gauss–Newton algorithm without a nondegeneracy assumption (see section 7). First define $\psi : \mathbb{R}^2 \to \mathbb{R}_+$ as

$$\psi(a, b) := ([-\phi(a, b)]_+)^2 + ([-a]_+)^2, \tag{1.13}$$

where $\phi(a, b)$ is the Fischer–Burmeister function defined in (1.11).

The following proposition is simple but is essential to the discussion of this paper.

Proposition 1.1. *The function ψ defined by* (1.13) *is continuously differentiable on the whole space of* $\mathbb{R}^2$ *and has the property*

$$\psi(a, b) = 0 \Longleftrightarrow a \geq 0, \quad ab_+ = 0. \tag{1.14}$$

Proof. Since both $([-\phi(a, b)]_+)^2$ and $([-a]_+)^2$ are continuously differentiable, ψ is continuously differentiable. By considering the fact that

$$a \geq 0,\ \sqrt{a^2 + b^2} - (a + b) \geq 0 \Longleftrightarrow a \geq 0,\ ab_+ = 0,$$

we get (1.14) easily. □

After simple computation we can see that the function ψ can be rewritten as

$$\psi(a, b) = \varphi(a, b)^2, \tag{1.15}$$

where

$$\begin{aligned} \varphi(a, b) \quad &:= \begin{cases} [-\phi(a, b)]_+ & \text{if } a \geq 0, \\ a & \text{otherwise,} \end{cases} \\ &= \begin{cases} -\phi(a, [b]_+) & \text{if } a \geq 0, \\ a & \text{otherwise,} \end{cases} \qquad (1.16) \\ &= \min\{[-\phi(a, b)]_+, a\}. \end{aligned}$$

Such equivalent expressions for ψ and φ will be useful in the following discussions. Note that φ is not differentiable at $(0, b)$ for any $b \leq 0$ and at $(a, 0)$ for any $a \geq 0$, but ψ is continuously differentiable.

Since for any $b \in \mathbb{R}$

$$\lim_{a \to \infty} \phi(a, b) = -b,$$

it is natural to define

$$\phi(+\infty, b) = -b.$$

Thus, for any $b \in \mathbb{R}$ we define

$$\psi(+\infty, b) = ([b]_+)^2.$$

Based on ψ, we define $f : \mathbb{R}^n \to \mathbb{R}$ as

$$f(x) := \frac{1}{2}\left[\sum_{i=1}^{n} \psi(x_i - l_i, F_i(x)) + \sum_{i=1}^{n} \psi(u_i - x_i, -F_i(x))\right]. \tag{1.17}$$

This function is an unconstrained differentiable merit function for VIP(l, u, F) (see Theorem 2.2) and has many good properties. When VIP(l, u, F) reduces to NCP(F), i.e., $l_i = 0$ and $u_i = +\infty$, $i = 1, \dots, n$, the function (1.17) becomes

$$f(x) = \frac{1}{2}\sum_{i=1}^{n} \eta(x_i, F_i(x)), \tag{1.18}$$

where for any $(a, b) \in \mathbb{R}^2$

$$\eta(a, b) := \begin{cases} ((a + b) - \sqrt{a^2 + b^2})^2 & \text{if } a \geq 0, b \geq 0, \\ b^2 & \text{if } a \geq 0, b < 0, \\ a^2 & \text{if } a < 0, b \geq 0, \\ a^2 + b^2 & \text{if } a < 0, b < 0. \end{cases} \tag{1.19}$$

The organization of this paper is as follows. In the next section we study some preliminary properties of the new merit function. In section 3 we study the conditions under which the level sets of f are bounded. In section 4 we give conditions which ensure that a stationary point of f is a solution of VIP(l, u, F). Section 5 is devoted to the nonsingularity of the iteration matrices. In section 6 we state the algorithm. We analyze the convergence properties of the algorithm in section 7 and give numerical results in section 8. Some concluding remarks are given in section 9.

For a continuously differentiable function $F : \mathbb{R}^n \to \mathbb{R}^n$, we denote the Jacobian of F at $x \in \mathbb{R}^n$ by $F'(x)$, whereas the transposed Jacobian is $\nabla F(x)$. Throughout $\|\cdot\|$ denotes the Euclidean norm. If $\mathcal{J}$ and $\mathcal{K}$ are index sets such that $\mathcal{J}, \mathcal{K} \subseteq \{1, \dots, m\}$, we denote by $W_{\mathcal{J}\mathcal{K}}$ the $|\mathcal{J}| \times |\mathcal{K}|$ submatrix of W consisting of entries W_{jk}, $j \in \mathcal{J}$, $k \in \mathcal{K}$. If $W_{\mathcal{J}\mathcal{J}}$ is nonsingular, we denote by $W/W_{\mathcal{J}\mathcal{J}}$ the Schur complement of $W_{\mathcal{J}\mathcal{J}}$ in W, i.e., $W/W_{\mathcal{J}\mathcal{J}} := W_{\mathcal{K}\mathcal{K}} - W_{\mathcal{K}\mathcal{J}}W_{\mathcal{J}\mathcal{J}}^{-1}W_{\mathcal{J}\mathcal{K}}$, where $\mathcal{K} = \{1, \dots, m\}\backslash\mathcal{J}$. If w is an m vector, we denote by $w_{\mathcal{J}}$ the subvector with components $j \in \mathcal{J}$.

2. Some preliminaries.

By noting that $x \in \mathbb{R}^n$ solves VIP(l, u, F) if and only if $H(x) = 0$ and that $\pm\infty \times 0 = 0$, we have the following results directly.

LEMMA 2.1. *A vector $x \in \mathbb{R}^n$ solves VIP(l, u, F) if and only if it satisfies*

$$l_i \leq x_i \leq u_i,\ (x_i - l_i)[F_i(x)]_+ = 0,\ (u_i - x_i)[-F_i(x)]_+ = 0,\ i = 1, \dots, n. \tag{2.1}$$

THEOREM 2.2. *The function $f(x)$ defined by* (1.17) *is nonnegative on $\mathbb{R}^n$, and $f(x) = 0$ if and only if $x \in \mathbb{R}^n$ solves VIP(l, u, F). In addition, if F is continuously differentiable, then f is also continuously differentiable.*

Proof. Since $\psi(a,b) \geq 0$ for all $(a,b) \in \mathbb{R}^2$, $f(x)$ is nonnegative on $\mathbb{R}^n$. From Proposition 1.1 and Lemma 2.1, $x \in \mathbb{R}^n$ solves VIP(l,u,F) if and only if it satisfies

$$\psi(x_i - l_i, F_i(x)) = 0, \quad \psi(u_i - x_i, -F_i(x)) = 0, \quad i = 1, \dots, n.$$

Thus $f(x) = 0$ if and only if $x \in \mathbb{R}^n$ solves VIP(l,u,F). Moreover, it is easy to see that if F is continuously differentiable, then so is F, as ψ is continuously differentiable by Proposition 1.1. □

Note that although f is continuously differentiable, it is not twice continuously differentiable and its gradient ∇f may not be locally Lipschitz continuous. For such an example we refer to the one-dimensional function given in section 1 of [44]. So a direct use of Newton's method for minimizing $f(x)$ may fail. However, we can still expect to obtain globally and superlinearly convergent Newton-type methods for minimizing $f(x)$. The tool used here is semismoothness.

Semismoothness was originally introduced by Mifflin [31] for functionals. Convex functions, smooth functions, and piecewise linear functions are examples of semismooth functions. The composition of semismooth functions is still a semismooth function (see [31]). In [40] Qi and Sun extended the definition of semismooth functions to $G : \mathbb{R}^n \to \mathbb{R}^m$. A locally Lipschitz continuous vector valued function $G : \mathbb{R}^n \to \mathbb{R}^m$ has a generalized Jacobian $\partial G(x)$ as in Clarke [5]. G is said to be *semismooth* at $x \in \mathbb{R}^n$ if

$$\lim_{\substack{V \in \partial G(x+th') \\ h' \to h, \ t \downarrow 0}} \{Vh'\}$$

exists for any $h \in R^n$. It has been proved in [40] that G is semismooth at x if and only if all its component functions are. Also $G'(x;h)$, the directional derivative of G at x in the direction h, exists for any $h \in \mathbb{R}^n$ and is equal to the above limit if G is semismooth at x.

LEMMA 2.3 (see [40]). *Suppose that $G : \mathbb{R}^n \to \mathbb{R}^m$ is a locally Lipschitzian function and is semismooth at x. Then*

(i) *for any $V \in \partial G(x+h)$, $h \to 0$,*

$$Vh - G'(x;h) = o(\|h\|);$$

(ii) *for any $h \to 0$,*

$$G(x+h) - G(x) - G'(x;h) = o(\|h\|).$$

A stronger notion than semismoothness is strong semismoothness. G is said to be *strongly semismooth* at x if G is semismooth at x, and for any $V \in \partial G(x+h)$, $h \to 0$,

$$Vh - G'(x;h) = O(\|h\|^2).$$

(Note that in [40] and [39] different names for strong semismoothness are used.) A function G is said to be a (strongly) semismooth function if it is (strongly) semismooth everywhere.

In [39] Qi defined the generalized Jacobian

$$\partial_B G(x) := \left\{ V \in \mathbb{R}^{n \times n} | \ V = \lim_{x^k \to x} G'(x^k), G \text{ is differentiable at } x^k \text{ for all } k \right\}.$$

This concept will be used in the design of our algorithm.

Let $\varphi : \mathbb{R}^2 \to \mathbb{R}$ be the function defined by (1.16) and define $\Psi,\ \Phi : \mathbb{R}^n \to \mathbb{R}^n$ by

$$\Psi_i(x) := \varphi(x_i - l_i, F_i(x)) = \min\{[-\phi(x_i - l_i, F_i(x))]_+, x_i - l_i\}$$

and

$$\Phi_i(x) := \varphi(u_i - x_i, -F_i(x)) = \min\{[-\phi(u_i - x_i, -F_i(x))]_+, u_i - x_i\}$$

for $i = 1, \ldots, n$. Define $G : \mathbb{R}^n \to \mathbb{R}^n$ by

$$\begin{aligned} G_i(x) &:= \sqrt{\Psi_i(x)^2 + \Phi_i(x)^2} \\ &= \begin{cases} (l_i - x_i) & \text{if } x_i < l_i \ \&\ F_i(x) \geq 0, \\ -\phi(x_i - l_i, F_i(x)) & \text{if } l_i \leq x_i \leq u_i \ \&\ F_i(x) \geq 0, \\ \sqrt{(u_i - x_i)^2 + \phi(x_i - l_i, F_i(x))^2} & \text{if } x_i > u_i \ \&\ F_i(x) \geq 0, \\ \sqrt{(x_i - l_i)^2 + \phi(u_i - x_i, -F_i(x))^2} & \text{if } x_i < l_i \ \&\ F_i(x) < 0, \\ -\phi(u_i - x_i, -F_i(x)) & \text{if } l_i \leq x_i \leq u_i \ \&\ F_i(x) < 0, \\ (x_i - u_i) & \text{if } x_i > u_i \ \&\ F_i(x) < 0 \end{cases} \end{aligned} \tag{2.2}$$

for $i = 1, \ldots, n$ and where $\phi(\cdot)$ is the Fischer–Burmeister function defined in (1.11). Then the merit function $f(x)$ defined by (1.17) can be rewritten as

$$f(x) = \frac{1}{2}\|G(x)\|^2. \tag{2.3}$$

Proposition 2.4. *Suppose that F is continuously differentiable at $x \in \mathbb{R}^n$. Then G is semismooth at x. Moreover if F' is locally Lipschitz continuous around x, then G is strongly semismooth at x.*

Proof. We need only to prove that for each i, G_i is (strongly) semismooth at x under the assumptions. First note that $\phi(\cdot)$ is a strongly semismooth function [18, Lemma 20] and $[\cdot]_+ : \mathbb{R} \to \mathbb{R}_+$ is strongly semismooth everywhere. Then by Theorem 19 in Fischer [18], which states that the composition of strongly semismooth functions is a strongly semismooth function, we know that $[-\phi(\cdot)]_+$ is strongly semismooth everywhere. It is easy to see that $\min\{\cdot, \cdot\} : \mathbb{R}^2 \to \mathbb{R}$ is strongly semismooth everywhere. Thus, by using Theorem 19 in Fischer [18] again, $\varphi : \mathbb{R}^2 \to \mathbb{R}$ is a strongly semismooth function. Then by Theorem 5 in Mifflin [31], which states that the composition of semismooth functions is semismooth, we know that Ψ_i and Φ_i are semismooth at x, and because $\sqrt{\alpha^2 + \beta^2}$ is a strongly semismooth function of α and β, G_i is semismooth at x. If F' is locally Lipschitz continuous, then $(y_i - l_i, F_i(y))$ and $(u_i - y_i, -F_i(y))$ are strongly semismooth at x. Thus, by Theorem 19 in Fischer [18] we know that Ψ_i and Φ_i are strongly semismooth at x. So, G_i is strongly semismooth at x. □

We need the following definitions concerning matrices and functions.

Definition 2.5. *A matrix $W \in \mathbb{R}^{n \times n}$ is called a*

- *P_0-matrix if each of its principal minors is nonnegative;*
- *P-matrix if each of its principal minors is positive.*

Obviously a positive semidefinite matrix is a P_0-matrix and a positive definite matrix is a P-matrix.

Definition 2.6. *A function $F : \mathbb{R}^n \to \mathbb{R}^n$*

- *is a P_0-function if for every x and y in $\mathbb{R}^n$ with $x \neq y$ there is an index i such that*

$$x_i \neq y_i, \quad (x_i - y_i)(F_i(x) - F_i(y)) \geq 0;$$

- *is a P-function if for every x and y in $\mathbb{R}^n$ with $x \neq y$ there is an index i such that*

$$x_i \neq y_i, \quad (x_i - y_i)(F_i(x) - F_i(y)) > 0;$$

- *is a uniform P-function if there exists a positive constant μ such that for every x and y in $\mathbb{R}^n$ there is an index i such that*

$$(x_i - y_i)(F_i(x) - F_i(y)) \geq \mu \|x - y\|^2;$$

- *is a monotone function if for every x and y in $\mathbb{R}^n$*

$$(x - y)^T (F(x) - F(y)) \geq 0;$$

- *is a strongly monotone function if there exists a positive constant μ such that for every x and y in $\mathbb{R}^n$*

$$(x - y)^T (F(x) - F(y)) \geq \mu \|x - y\|^2.$$

It is known that every strongly monotone function is a uniform P-function and every monotone function is a P_0-function. Furthermore, the Jacobian of a continuously differentiable P_0-function (uniform P-function) is a P_0-matrix (P-matrix).

3. Bounded level sets. In this section we study the conditions under which the level sets of the merit function f are bounded. Since for any $c \in \mathbb{R} \cup \{-\infty\}$, $d \in \mathbb{R} \cup \{\infty\}$ with $c < d$ and $a \in \mathbb{R}$, $\Pi_{[c,d]\cap\mathbb{R}}(a) = \Pi_{[c,d]}(a)$, for the sake of simplicity we use $\Pi_{[c,d]}(a)$ instead of $\Pi_{[c,d]\cap\mathbb{R}}(a)$ to represent the orthogonal projection of a onto $[c, d] \cap \mathbb{R}$. Boundedness results for NCP(F) with uniform P-functions have been established by Jiang [24], Facchinei and Soares [14], and De Luca, Facchinei, and Kanzow [6]. Here these results are extended to VIP(l, u, F).

The following lemma is essential to develop conditions which ensure bounded level sets.

LEMMA 3.1. *For four given numbers $a, b \in \mathbb{R}$, $c \in \mathbb{R} \cup \{-\infty\}$, and $d \in \mathbb{R} \cup \{\infty\}$ with $c < d$, we have*

$$\gamma_1 \left|a - \Pi_{[c,d]}[a - b]\right|^2 \leq \psi(a - c, b) + \psi(d - a, -b) \leq \gamma_2 \left|a - \Pi_{[c,d]}[a - b]\right|^2 \tag{3.1}$$

with $\gamma_1 = 1/(6 + 4\sqrt{2})$ and $\gamma_2 = 12 + 8\sqrt{2}$.

Proof. First, from Tseng [45], for any two numbers $v, w \in \mathbb{R}$ we have

$$\frac{1}{2 + \sqrt{2}} |\min\{v, w\}| \leq |\phi(v, w)| \leq (2 + \sqrt{2}) |\min\{v, w\}|. \tag{3.2}$$

Then by using the second equality of (1.16), if $v \geq 0$ we have

$$\frac{1}{2 + \sqrt{2}} |\min\{v, w_+\}| \leq |\varphi(v, w)| \leq (2 + \sqrt{2}) |\min\{v, w_+\}| \tag{3.3}$$

and if $v < 0$ we have

$$|\varphi(v,w)| = |v| = |\min\{v, w_+\}|. \tag{3.4}$$

Thus, for all $(v,w) \in \mathbb{R}^2$ we have

$$\frac{1}{6+4\sqrt{2}}|\min\{v,w_+\}|^2 \le \varphi(v,w)^2 \le (6+4\sqrt{2})|\min\{v,w_+\}|^2.$$

Let

$$\begin{aligned} t &:= |\min\{a-c, b_+\}|^2 + |\min\{d-a, [-b]_+\}|^2 \\ &= \begin{cases} |\min\{a-c,b\}|^2 + (d-a)^2 & \text{if } b \ge 0 \;\&\; a \ge d, \\ |\min\{a-c,b\}|^2 & \text{if } b \ge 0 \;\&\; a < d, \\ |\min\{d-a,-b\}|^2 & \text{if } b < 0 \;\&\; a \ge c, \\ (a-c)^2 + |\min\{d-a,-b\}|^2 & \text{if } b < 0 \;\&\; a < c. \end{cases} \end{aligned} \tag{3.5}$$

Then

$$\frac{1}{6+4\sqrt{2}}t \le \psi(a-c,b) + \psi(d-a,-b) \le (6+4\sqrt{2})t. \tag{3.6}$$

Denote

$$r := |a - \Pi_{[c,d]}[a-b]|^2 = \begin{cases} (a-c)^2 & \text{if } a-b \le c, \\ b^2 & \text{if } c < a-b < d, \\ (a-d)^2 & \text{if } a-b \ge d. \end{cases}$$

Next we prove that

$$r \le t \le 2r. \tag{3.7}$$

First, if either $b \ge 0$ and $a < d$ or $b < 0$ and $a > c$, then we can directly verify that $r = t$. Next, we consider the other two cases.

Case 1. $b \ge 0$ and $a \ge d$. Then

$$t = |\min\{a-c,b\}|^2 + (d-a)^2 = (d-a)^2 + \begin{cases} (a-c)^2 & \text{if } b \ge a-c, \\ b^2 & \text{if } b < a-c. \end{cases}$$

After simple computation we get

$$r \le t \le 2r.$$

Case 2. $b < 0$ and $a \le c$. Then

$$t = (a-c)^2 + |\min\{d-a,-b\}|^2 = (a-c)^2 + \begin{cases} (d-a)^2 & \text{if } d-a \le -b, \\ b^2 & \text{if } d-a > -b. \end{cases}$$

Again, after simple computation we get

$$r \le t \le 2r.$$

Overall we have proved (3.7). By combining (3.6) and (3.7), we get (3.1). □

THEOREM 3.2. *Suppose that for any sequence $\{x^k\}$ with $\|x^k\| \to \infty$ there exists an index $i \in \{1, \ldots, n\}$ independent of k such that*

$$|x_i^k - \Pi_{[l_i,u_i]}[x_i^k - F_i(x^k)]| \to \infty. \tag{3.8}$$

Then for any $c \geq 0$, $L_c(f)$ is bounded. In particular, if S is bounded or if F is a uniform P-function, then $L_c(f)$ is bounded.

Proof. Suppose that for some given $c \geq 0$, $L_c(f)$ is unbounded. Then there exists a sequence $\{x^k\}$ diverging to infinity and satisfying

$$f(x^k) \leq c.$$

But, on the other hand, from Lemma 3.1 and the assumption that there exists an index i, independent of k, such that (3.8) holds, we have

$$f(x^k) \geq \frac{1}{2}\gamma_1 |x_i^k - \Pi_{[l_i,u_i]}[x_i^k - F_i(x^k)]|^2 \to \infty,$$

where $\gamma_1 = 1/(6+4\sqrt{2})$. This is a contradiction. So for any $c \geq 0$, $L_c(f)$ is bounded if (3.8) holds.

By noting that if S is bounded then (3.8) holds automatically, we can conclude that for any given $c \geq 0$, $L_c(f)$ is bounded.

If F is a uniform P-function, then by [14] for any sequence $\{x^k\}$ with $\|x^k\| \to \infty$ there exists an index $i \in \{1, \ldots, n\}$ independent of k such that

$$|x_i^k| \to \infty, \quad |F_i(x^k)| \to \infty,$$

which, in turn, implies (3.8). This completes the proof. □

4. Stationary point conditions. In general a stationary point of a merit function may not be a solution of the underlying problem. Many people [6, 11, 14, 21, 24, 25, 26, 29, 47] have studied the conditions under which a stationary point is a solution of NCP(F). In this section we study the conditions under which a stationary point of (1.17) is a solution of VIP(l, u, F). Similar work has been done in [10, 27] for box constrained variational inequality problems.

First let us study the structure of $\partial_B G_i(x)$, where $G_i(\cdot)$, $i = 1, \ldots, n$ are defined in (2.2). Denote by e_i the ith unit row vector of $\mathbb{R}^n$, $i = 1, \ldots, n$. For any $x \in \mathbb{R}^n$ we discuss five cases, each of which includes three subcases.

Case 1. $x_i < l_i$.

Case 1.1 $F_i(x) > 0$. Then $G_i(x) = l_i - x_i$ and $\partial_B G_i(x) = \{-e_i\}$.

Case 1.2. $F_i(x) < 0$. Then

$$G_i(x) = \sqrt{(x_i - l_i)^2 + \phi(u_i - x_i, -F_i(x))^2},$$
$$\partial_B G_i(x) = \{\alpha_i(-e_i) + \beta_i(-F_i'(x))\},$$

where

$$\alpha_i = \frac{l_i - x_i}{G_i(x)} + \frac{\phi(u_i - x_i, -F_i(x))}{G_i(x)} \left(\frac{u_i - x_i}{\sqrt{(u_i - x_i)^2 + (-F_i(x))^2}} - 1 \right),$$
$$\beta_i = \frac{\phi(u_i - x_i, -F_i(x))}{G_i(x)} \left(\frac{-F_i(x)}{\sqrt{(u_i - x_i)^2 + (-F_i(x))^2}} - 1 \right).$$

Case 1.3. $F_i(x) = 0$. Then $G_i(x) = l_i - x_i$ and $\partial_B G_i(x) = \{-e_i\}$.

Case 2. $x_i > u_i$.

Case 2.1. $F_i(x) > 0$. Then

$$\begin{aligned} G_i(x) &= \sqrt{(u_i - x_i)^2 + \phi(x_i - l_i, F_i(x))^2}, \\ \partial_B G_i(x) &= \{\alpha_i e_i + \beta_i F_i'(x)\}, \end{aligned}$$

where

$$\begin{aligned} \alpha_i &= \frac{x_i - u_i}{G_i(x)} + \frac{\phi(x_i - l_i, F_i(x))}{G_i(x)} \left(\frac{x_i - l_i}{\sqrt{(x_i - l_i)^2 + F_i(x)^2}} - 1 \right), \\ \beta_i &= \frac{\phi(x_i - l_i, F_i(x))}{G_i(x)} \left(\frac{F_i(x)}{\sqrt{(x_i - l_i)^2 + F_i(x)^2}} - 1 \right). \end{aligned}$$

Case 2.2. $F_i(x) < 0$. Then $G_i(x) = x_i - u_i$ and $\partial_B G_i(x) = \{e_i\}$.

Case 2.3. $F_i(x) = 0$. Then $G_i(x) = x_i - u_i$ and $\partial_B G_i(x) = \{e_i\}$.

Case 3. $l_i < x_i < u_i$.

Case 3.1. $F_i(x) > 0$. Then

$$\begin{aligned} G_i(x) &= -\phi(x_i - l_i, F_i(x)), \\ \partial_B G_i(x) &= \{\alpha_i e_i + \beta_i F_i'(x)\}, \end{aligned}$$

where

$$\alpha_i = 1 - \frac{x_i - l_i}{\sqrt{(x_i - l_i)^2 + F_i(x)^2}} \text{ and } \beta_i = 1 - \frac{F_i(x)}{\sqrt{(x_i - l_i)^2 + F_i(x)^2}}.$$

Case 3.2. $F_i(x) < 0$. Then

$$\begin{aligned} G_i(x) &= -\phi(u_i - x_i, -F_i(x)), \\ \partial_B G_i(x) &= \{\alpha_i(-e_i) + \beta_i(-F_i'(x))\}, \end{aligned}$$

where

$$\alpha_i = 1 - \frac{u_i - x_i}{\sqrt{(u_i - x_i)^2 + F_i(x)^2}} \text{ and } \beta_i = 1 - \frac{-F_i(x)}{\sqrt{(u_i - x_i)^2 + F_i(x)^2}}.$$

Case 3.3. $F_i(x) = 0$. Then $G_i(x) = 0$ and $\partial_B G_i(x) \subseteq \{F_i'(x), -F_i'(x)\}$.

Case 4. $x_i = l_i$.

Case 4.1. $F_i(x) > 0$. Then $G_i(x) = 0$ and $\partial_B G_i(x) \subseteq \{e_i, -e_i\}$.

Case 4.2. $F_i(x) < 0$. Then

$$\begin{aligned} G_i(x) &= -\phi(u_i - l_i, -F_i(x)), \\ \partial_B G_i(x) &= \{\alpha_i(-e_i) + \beta_i(-F_i'(x))\}, \end{aligned}$$

where

$$\alpha_i = 1 - \frac{u_i - l_i}{\sqrt{(u_i - l_i)^2 + F_i(x)^2}} \text{ and } \beta_i = 1 - \frac{-F_i(x)}{\sqrt{(u_i - l_i)^2 + F_i(x)^2}}.$$

Case 4.3. $F_i(x) = 0$. Then $G_i(x) = 0$ and

$$\partial_B G_i(x) \subseteq \{\alpha_i e_i + \beta_i F_i'(x)\} \cup \{\bar{\alpha}_i(-e_i) + \bar{\beta}_i(-F_i'(x))\},$$

where $\alpha_i, \beta_i \in [0,1]$ satisfy $(\alpha_i - 1)^2 + (\beta_i - 1)^2 = 1$ and $\bar{\alpha}_i, \bar{\beta}_i \in [0,1]$ satisfy $\bar{\alpha}_i^2 + \bar{\beta}_i^2 = 1$.

Case 5. $x_i = u_i$.

Case 5.1. $F_i(x) > 0$. Then

$$G_i(x) = -\phi(u_i - l_i, F_i(x)),$$
$$\partial_B G_i(x) = \{\alpha_i e_i + \beta_i F_i'(x)\},$$

where

$$\alpha_i = 1 - \frac{u_i - l_i}{\sqrt{(u_i - l_i)^2 + F_i(x)^2}} \text{ and } \beta_i = 1 - \frac{F_i(x)}{\sqrt{(u_i - l_i)^2 + F_i(x)^2}}.$$

Case 5.2. $F_i(x) < 0$. Then $G_i(x) = 0$ and $\partial_B G_i(x) \subseteq \{e_i, -e_i\}$.

Case 5.3. $F_i(x) = 0$. Then $G_i(x) = 0$ and

$$\partial_B G_i(x) \subseteq \{\alpha_i(-e_i) + \beta_i(-F_i'(x))\} \cup \{\bar{\alpha}_i e_i + \bar{\beta}_i F_i'(x)\},$$

where $\alpha_i, \beta_i \in [0,1]$ satisfy $(\alpha_i - 1)^2 + (\beta_i - 1)^2 = 1$ and $\bar{\alpha}_i, \bar{\beta}_i \in [0,1]$ satisfy $\bar{\alpha}_i^2 + \bar{\beta}_i^2 =1$.

For any $x \in \mathbb{R}^n$ define the index sets $\mathcal{A}_{jk}(x)$ by

$$\mathcal{A}_{jk}(x) := \{i|\text{ Case } j.k \text{ occurs at } x_i,\ i = 1, \ldots, n\},\ j = 1, \ldots, 5,\ k = 1, \ldots, 3.$$

For example, some $i \in \mathcal{A}_{42}(x)$ means that Case 4.2 occurs at x_i, i.e., $x_i = l_i$ and $F_i(x) < 0$. Furthermore let

$$\mathcal{A}_{31}^{-\infty}(x) := \{i|\ i \in \mathcal{A}_{31}(x) \text{ and } l_i = -\infty,\ i = 1, \ldots, n\},$$
$$\mathcal{A}_{32}^{\infty}(x) := \{i|\ i \in \mathcal{A}_{32}(x) \text{ and } u_i = \infty,\ i = 1, \ldots, n\},$$
$$\mathcal{A}_{42}^{\infty}(x) := \{i|\ i \in \mathcal{A}_{42}(x) \text{ and } u_i = \infty,\ i = 1, \ldots, n\},$$
$$\mathcal{A}_{51}^{-\infty}(x) := \{i|\ i \in \mathcal{A}_{51}(x) \text{ and } l_i = -\infty,\ i = 1, \ldots, n\}.$$

For convenience we define the four additional index sets

$$\mathcal{O}(x) := \mathcal{A}_{11}(x) \cup \mathcal{A}_{13}(x) \cup \mathcal{A}_{22}(x) \cup \mathcal{A}_{23}(x),$$
$$\mathcal{P}(x) := \mathcal{A}_{31}^{-\infty} \cup \mathcal{A}_{32}^{\infty} \cup \mathcal{A}_{42}^{\infty} \cup \mathcal{A}_{51}^{-\infty},$$
$$\mathcal{Q}(x) := \mathcal{A}_{33}(x) \cup \mathcal{A}_{41}(x) \cup \mathcal{A}_{43}(x) \cup \mathcal{A}_{52}(x) \cup \mathcal{A}_{53}(x),$$
$$\mathcal{R}(x) := \{1, \ldots, n\} \backslash \{\mathcal{O}(x) \cup \mathcal{P}(x) \cup \mathcal{Q}(x)\}.$$

LEMMA 4.1. *For any $x \in \mathbb{R}^n$, each $i \in \{1, \ldots, n\}$, and any $W \in \partial_B G_i(x)$ we have that*

i) *if $i \in \mathcal{O}(x)$, then either $W^T G_i(x) = G_i(x) e_i^T$ or $W^T G_i(x) = -G_i(x) e_i^T$;*

ii) *if $i \in \mathcal{P}(x)$, then either $W^T G_i(x) = G_i(x) \nabla F_i(x)$ or $W^T G_i(x) = -G_i(x) \nabla F_i(x)$;*

iii) *if $i \in \mathcal{Q}(x)$, then $W^T G_i(x) = 0$;*

iv) *if $i \in \mathcal{R}$, then there exist c_i and d_i such that $W^T G_i(x) = c_i e_i^T + d_i \nabla F_i(x)$ and $c_i d_i > 0$.*

Proof. Parts i)–iii) can be easily verified. For part iv) we need only to note that if $i \in \mathcal{R}$, then $G_i(x) \neq 0$ and there exist positive numbers α_i and β_i such that $W = \alpha_i e_i + \beta_i F_i'(x)$ or $W = \alpha_i(-e_i) + \beta_i(-F_i'(x))$. □

Without causing any confusion we will use $\mathcal{O}$, $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ to represent $\mathcal{O}(x)$, $\mathcal{P}(x)$, $\mathcal{Q}(x)$, and $\mathcal{R}(x)$, respectively. Without loss of generality, assume that $\nabla F(x)$ is partitioned in the form

$$\nabla F(x) = \begin{pmatrix} \nabla F(x)_{\mathcal{O}\mathcal{O}} & \nabla F(x)_{\mathcal{O}\mathcal{P}} & \nabla F(x)_{\mathcal{O}\mathcal{Q}} & \nabla F(x)_{\mathcal{O}\mathcal{R}} \\ \nabla F(x)_{\mathcal{P}\mathcal{O}} & \nabla F(x)_{\mathcal{P}\mathcal{P}} & \nabla F(x)_{\mathcal{P}\mathcal{Q}} & \nabla F(x)_{\mathcal{P}\mathcal{R}} \\ \nabla F(x)_{\mathcal{Q}\mathcal{O}} & \nabla F(x)_{\mathcal{Q}\mathcal{P}} & \nabla F(x)_{\mathcal{Q}\mathcal{Q}} & \nabla F(x)_{\mathcal{Q}\mathcal{R}} \\ \nabla F(x)_{\mathcal{R}\mathcal{O}} & \nabla F(x)_{\mathcal{R}\mathcal{P}} & \nabla F(x)_{\mathcal{R}\mathcal{Q}} & \nabla F(x)_{\mathcal{R}\mathcal{R}} \end{pmatrix}.$$

Now we are ready to give the main result of this section.

THEOREM 4.2. *Suppose that $x \in \mathbb{R}^n$ is a stationary point of f, i.e., $\nabla f(x) = 0$, and that $\nabla F(x)_{\mathcal{P}\mathcal{P}}$ is nonsingular and its Schur complement in*

$$\begin{pmatrix} \nabla F(x)_{\mathcal{P}\mathcal{P}} & \nabla F(x)_{\mathcal{P}\mathcal{R}} \\ \\ \nabla F(x)_{\mathcal{R}\mathcal{P}} & \nabla F(x)_{\mathcal{R}\mathcal{R}} \end{pmatrix}$$

is a P_0-matrix. Then x is a solution of VIP(l,u,F).

Proof. Since f is continuously differentiable and G is locally Lipschitz continuous, by Clarke [5] we have that for any $y \in \mathbb{R}^n$ and any $V \in \partial G(y)$

$$\nabla f(y) = V^T G(y).$$

Let V be an element of $\partial_B G(x) (\subseteq \partial G(x))$. Then for $i = 1, \ldots, n$ there exist matrices $W_i \in \partial_B G_i(x)$ such that

$$V = W_1 \times W_2 \times \cdots \times W_n.$$

Thus

$$\nabla f(x) = \sum_{i=1}^{n} W_i^T G_i(x) = 0.$$

By considering parts i) and ii) of Lemma 4.1, without loss of generality we assume that

$$W_i^T G_i(x) = G_i(x) e_i^T \text{ for } i \in \mathcal{O} \qquad \text{and} \qquad W_i^T G_i(x) = G_i(x) \nabla F_i(x) \text{ for } i \in \mathcal{P}.$$

Thus from Lemma 4.1,

$$\sum_{i\in\mathcal{O}} G_i(x) e_i^T + \sum_{i\in\mathcal{P}} G_i(x) \nabla F_i(x) + \sum_{i\in\mathcal{R}} (M_i e_i^T + N_i \nabla F_i(x)) = 0, \tag{4.1}$$

where

$$M_i := c_i G_i(x), \quad N_i := d_i G_i(x), \quad i \in \mathcal{R},$$

and c_i and d_i are numbers defined in part iv) of Lemma 4.1. Equation (4.1) can be rewritten as

$$\begin{aligned} G_{\mathcal{O}}(x) + \nabla F(x)_{\mathcal{O}\mathcal{P}} G_{\mathcal{P}}(x) + \nabla F(x)_{\mathcal{O}\mathcal{R}} N_{\mathcal{R}} &= 0, \\ \nabla F(x)_{\mathcal{P}\mathcal{P}} G_{\mathcal{P}}(x) + \nabla F(x)_{\mathcal{P}\mathcal{R}} N_{\mathcal{R}} &= 0, \\ \nabla F(x)_{\mathcal{Q}\mathcal{P}} G_{\mathcal{P}}(x) + \nabla F(x)_{\mathcal{Q}\mathcal{R}} N_{\mathcal{R}} &= 0, \\ \nabla F(x)_{\mathcal{R}\mathcal{P}} G_{\mathcal{P}}(x) + M_{\mathcal{R}} + \nabla F(x)_{\mathcal{R}\mathcal{R}} N_{\mathcal{R}} &= 0. \end{aligned} \tag{4.2}$$

From the second equality of (4.2) we have

$$G_{\mathcal{P}}(x) = -\left(\nabla F(x)_{\mathcal{P}\mathcal{P}}\right)^{-1} \nabla F(x)_{\mathcal{P}\mathcal{R}} N_{\mathcal{R}}.$$

This and the fourth equality of (4.2) give

$$M_{\mathcal{R}} + [\nabla F(x)_{\mathcal{R}\mathcal{R}} - \nabla F(x)_{\mathcal{R}\mathcal{P}}(\nabla F(x)_{\mathcal{P}\mathcal{P}})^{-1}\nabla F(x)_{\mathcal{P}\mathcal{R}}]N_{\mathcal{R}} = 0. \tag{4.3}$$

Since $\nabla F(x)_{\mathcal{R}\mathcal{R}} - \nabla F(x)_{\mathcal{R}\mathcal{P}}(\nabla F(x)_{\mathcal{P}\mathcal{P}})^{-1}\nabla F(x)_{\mathcal{P}\mathcal{R}}$ is a P_0-matrix, there exists an index $j \in \{1, \ldots, |\mathcal{R}|\}$ such that

$$(N_{\mathcal{R}})_j\{[\nabla F(x)_{\mathcal{R}\mathcal{R}} - \nabla F(x)_{\mathcal{R}\mathcal{P}}(\nabla F(x)_{\mathcal{P}\mathcal{P}})^{-1}\nabla F(x)_{\mathcal{P}\mathcal{R}}]N_{\mathcal{R}}\}_j \geq 0,$$

which, with (4.3), gives

$$(M_{\mathcal{R}})_j(N_{\mathcal{R}})_j \leq 0.$$

This contradicts part iv) of Lemma 4.1. So, we have

$$\mathcal{R} = \emptyset.$$

Then from the second and the first equalities of (4.2) we get

$$G_{\mathcal{P}}(x) = G_{\mathcal{O}}(x) = 0.$$

Thus, by Lemma 4.1 we have proved that $G(x) = 0$, so x is a solution of VIP(l, u, F) by Theorem 2.2. □

The conditions used in Theorem 4.2 are quite mild. In particular, if all $l_i = -\infty$ and all $u_i = \infty$, i.e., VIP(l, u, F) reduces to the nonlinear system of equations $F(x) = 0$, we require only that $\nabla F(x)$ is nonsingular. Also, if all l_i and u_i are bounded we require only that $\nabla F(x)_{\mathcal{R}\mathcal{R}}$ is a P_0-matrix, which is implied by assuming that F is a P_0-function.

5. Nonsingularity conditions. In this section we study the conditions under which the elements of a generalized Jacobian are nonsingular at a solution point $x^* \in \mathbb{R}^n$ of VIP(l, u, F). The basic idea follows from Facchinei and Soares [14]. Since x^* is a solution of VIP(l, u, F),

$$\mathcal{O} = \mathcal{P} = \mathcal{R} = \emptyset, \quad \mathcal{Q} = \{1, \ldots, n\},$$

where $\mathcal{O}$, $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ are abbreviations of $\mathcal{O}(x^*)$, $\mathcal{P}(x^*)$, $\mathcal{Q}(x^*)$, and $\mathcal{R}(x^*)$, respectively. For notational convenience let

$$\begin{aligned}
\mathcal{I} &:= \mathcal{A}_{33}(x^*) = \{i \in 1, \ldots, n \mid l_i < x_i^* < u_i \text{ and } F_i(x^*) = 0\},\\
\mathcal{J} &:= \mathcal{A}_{43}(x^*) \cup \mathcal{A}_{53}(x^*)\\
&\ \ = \{i \in 1, \ldots, n \mid x_i^* = l_i \text{ and } F_i(x^*) = 0\}\\
&\qquad \cup \{i \in 1, \ldots, n \mid x_i^* = u_i \text{ and } F_i(x^*) = 0\},\\
\mathcal{K} &:= \mathcal{A}_{41}(x^*) \cup \mathcal{A}_{52}(x^*)\\
&\ \ = \{i \in 1, \ldots, n \mid x_i^* = l_i \text{ and } F_i(x^*) > 0\}\\
&\qquad \cup \{i \in 1, \ldots, n \mid x_i^* = u_i \text{ and } F_i(x^*) < 0\}.
\end{aligned}$$

Then

$$\mathcal{I} \cup \mathcal{J} \cup \mathcal{K} = \{1, \ldots, n\}.$$

By rearrangement we assume that $F'(x^*)$ can be rewritten as

$$F'(x^*) = \begin{pmatrix} F'(x^*)_{\mathcal{I}\mathcal{I}} & F'(x^*)_{\mathcal{I}\mathcal{J}} & F'(x^*)_{\mathcal{I}\mathcal{K}} \\ F'(x^*)_{\mathcal{J}\mathcal{I}} & F'(x^*)_{\mathcal{J}\mathcal{J}} & F'(x^*)_{\mathcal{J}\mathcal{K}} \\ F'(x^*)_{\mathcal{K}\mathcal{I}} & F'(x^*)_{\mathcal{K}\mathcal{J}} & F'(x^*)_{\mathcal{K}\mathcal{K}} \end{pmatrix}.$$

VIP(l, u, F) is said to be R-regular at x^* if $F'(x^*)_{\mathcal{I}\mathcal{I}}$ is nonsingular and its Schur complement in the matrix

$$\begin{pmatrix} F'(x^*)_{\mathcal{I}\mathcal{I}} & F'(x^*)_{\mathcal{I}\mathcal{J}} \\ F'(x^*)_{\mathcal{J}\mathcal{I}} & F'(x^*)_{\mathcal{J}\mathcal{J}} \end{pmatrix}$$

is a P-matrix; see [14]. R-regularity coincides with the notion of regularity introduced in [42].

PROPOSITION 5.1. *Suppose that VIP(l, u, F) is R-regular at x^*. Then all $V \in \partial_B G(x^*)$ are nonsingular.*

Proof. Since

$$\partial_B G(x^*) \subseteq \partial_C G(x^*) := \partial_B G_1(x^*) \times \partial_B G_2(x^*) \times \cdots \times \partial_B G_n(x^*),$$

it is sufficient to prove the conclusion by showing that all $U \in \partial_C G(x^*)$ are nonsingular. Let U be an arbitrary element of $\partial_C G(x^*)$. By the discussion of section 4 on the structure of $\partial_B G_i(x^*)$ and as x^* is a solution of VIP(l, u, F), we have

$$U_i = \begin{cases} F_i'(x^*) \quad \text{or} \quad -F_i'(x^*) & \text{if } i \in \mathcal{I}, \\ \alpha_i e_i + \beta_i F_i'(x^*) \quad \text{or} \quad \alpha_i(-e_i) + \beta_i(-F_i'(x^*)) & \text{if } i \in \mathcal{J}, \\ e_i \quad \text{or} \quad -e_i & \text{if } i \in \mathcal{K}, \end{cases} \tag{5.1}$$

where in (5.1) α_i and β_i are nonnegative numbers satisfying $(\alpha_i - 1)^2 + (\beta_i - 1)^2 = 1$ or $(\alpha_i)^2 + (\beta_i)^2 = 1$ for $i \in \mathcal{J}$. By using standard analysis (see, for example, [14, Proposition 3.2]) we can prove that U is nonsingular under the assumptions and, so, complete the proof. □

THEOREM 5.2. *Suppose that VIP(l, u, F) is R-regular at x^*. Then there exist a neighborhood $N(x^*)$ of x^* and a constant c such that for any $x \in N(x^*)$ and any $V \in \partial_B G(x)$, V is nonsingular and satisfies*

$$\|V^{-1}\| \leq c.$$

Proof. This follows directly from Proposition 5.1 and [39, Lemma 2.6]. □

COROLLARY 5.3. *If $F'(x^*)$ is a P-matrix, then the conclusion of Theorem* 5.2 *holds.*

Proof. This corollary is established by noting that if $F'(x^*)$ is a P-matrix, then VIP(l, u, F) is R-regular at x^*. □

We note that Sun, Fukushima, and Qi [44, Theorem 3.2] proved a similar result to Corollary 5.3 for the D-gap function $h_{\alpha\beta}(x)$ defined by (1.7). For VIP(l, u, F), their condition becomes

$$\lambda_{\min}(F'(x^*) + F'(x^*)^T) \geq \alpha + \beta^{-1}\|\nabla F(x^*)\|^2, \tag{5.2}$$

where $0 < \alpha < \beta$ and $\lambda_{\min}(W)$ denotes the smallest eigenvalue of the symmetric matrix W. Condition (5.2) implies that $F'(x^*)$ must be a positive definite matrix and hence a P-matrix. It may not be satisfied if $F'(x^*)$ is only a P-matrix. For example, let $n = 2$, $S = \mathbb{R}^2_+$, and $F(x) = Wx + q$ with

$$W = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}, \quad q = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Then $x^* = (0,0)^T$, $F'(x^*) = W$ is a P-matrix, but (5.2) fails to hold because $\lambda_{\min}(F'(x^*) + F'(x^*)^T) = 0$. Thus our assumption is weaker. In fact, one of our main motivations of this paper is to pursue a simple and differentiable merit function for VIP(l, u, F) such that the iteration matrix is nonsingular if $F'(x^*)$ is only a P-matrix.

6. A damped Gauss–Newton method. In this section a damped Gauss–Newton method for solving VIP(l, u, F) is outlined. It is similar to that in Facchinei and Kanzow [12], except that a negative gradient direction is not used. The motivation for using damped Gauss–Newton methods for solving semismooth equations is discussed in [12].

Let $I \in \mathbb{R}^{n\times n}$ be the identity matrix. An outline of a damped Gauss–Newton method is as follows.

Step 0. Choose $x^0 \in \mathbb{R}^n$, $\rho \in (0,1)$, $p_1, p_2 > 0$, and $\sigma \in (0, 1/2)$. Set $k := 0$.

Step 1. If $\|\nabla f(x^k)\| = 0$, stop.

Step 2. Select an element $V_k \in \partial_B G(x^k)$. Let d^k be the solution of the linear system

$$\left(V_k^T V_k + p_1 \|G(x^k)\|^{p_2} I\right) d = -\nabla f(x^k). \tag{6.1}$$

Step 3. Let m_k be the smallest nonnegative integer m such that

$$f(x^k + \rho^m d^k) \le f(x^k) + \sigma \rho^m \nabla f(x^k)^T d^k. \tag{6.2}$$

Set $x^{k+1} := x^k + \rho^{m_k} d^k$, $k := k+1$ and go to Step 1.

The above method is different from the classical damped Gauss–Newton method for solving nonlinear least squares problems in that G is not continuously differentiable. Note that if in (6.1) p_1 is set to zero, the solution of (6.1) is exactly the solution of the linear least squares problem

$$\min_{d\in\mathbb{R}^n} \frac{1}{2}\|V_k d + G(x^k)\|^2$$

as $f(x)$ is continuously differentiable and $\nabla f(x^k) = V_k^T G(x^k)$ [5]. In (6.1), the term $p_1\|G(x^k)\|^{p_2} I$ is used to make sure that $V_k^T V_k + p_1\|G(x^k)\|^{p_2} I$ is positive definite. If x^k is not a solution of VIP(l, u, F), then $\nabla f(x^k)^T d^k < 0$, which means that the above algorithm is well defined at the kth iteration. If V_k is nonsingular, then the term $V_k^T V_k$ is positive definite and with $p_1 = 0$ the solution of (6.1) reduces to solving the linear system

$$V_k d = -G(x^k)$$

to get a generalized Newton direction.

7. Convergence analysis.

In this section we analyze the convergence properties of the damped Gauss–Newton method described in section 6, establishing superlinear convergence without any nondegeneracy assumption. The analysis builds on the work of [6, 12] for NCP(F).

First we state a global convergence theorem.

THEOREM 7.1. *Suppose that $\{x^k\}$ is a sequence generated by the damped Gauss–Newton method. Then each accumulation point x^* of $\{x^k\}$ is a stationary point of f.*

Proof. The proof is similar to that of [12, Theorem 15]. We omit the detail. □

Now we are ready to prove the superlinear (quadratic) convergence of the damped Gauss–Newton method. We proceed along the lines of the proof of [12, Theorem 17], except that for superlinear convergence we do not assume that F' is Lipschitz continuous and for quadratic convergence we do not assume that F' is continuously differentiable.

THEOREM 7.2. *Suppose that $\{x^k\}$ is a sequence generated by the damped Gauss–Newton method and x^*, an accumulation point of $\{x^k\}$, is a solution of VIP(l, u, F). If VIP(l, u, F) is R-regular at x^*, then the whole sequence $\{x^k\}$ converges to x^* Q-superlinearly. Furthermore, if F' is Lipschitz continuous around x^* and $p_2 \geq 1$, then the convergence is Q-quadratic.*

Proof. From Lemma 2.3, Proposition 2.4, and Theorem 5.2, for all x^k sufficiently close to x^* we have

$$\begin{aligned}
&\|x^k + d^k - x^*\| \\
&\quad = \|x^k - \left(V_k^T V_k + p_1\|G(x^k)\|^{p_2} I\right)^{-1} \nabla f(x^k) - x^*\| \\
&\quad \leq \| \left(V_k^T V_k + p_1\|G(x^k)\|^{p_2} I\right)^{-1} \| \, \|\nabla f(x^k) - \left(V_k^T V_k + p_1\|G(x^k)\|^{p_2} I\right)(x^k - x^*)\| \\
&\quad = O(1)\|V_k^T G(x^k) - V_k^T V_k(x^k - x^*) - p_1\|G(x^k)\|^{p_2}(x^k - x^*)\| \\
&\quad \leq O(1)\left[\|V_k^T\|\|G(x^k) - G(x^*) - V_k(x^k - x^*)\| + p_1\|G(x^k)\|^{p_2}\|x^k - x^*\|\right] \\
&\quad \leq O(1)\|G(x^k) - G(x^*) - V_k(x^k - x^*)\| + O\left(\|G(x^k)\|^{p_2}\right)\|x^k - x^*\| \\
&\quad \leq o(\|x^k - x^*\|).
\end{aligned} \tag{7.1}$$

Then, for all x^k sufficiently close to x^*,

$$\|d^k\| = \|x^k - x^*\| + o(\|x^k - x^*\|),$$

and so

$$\begin{aligned}
f(x^k + d^k) &= \frac{1}{2}\|G(x^k + d^k)\|^2 \\
&= \frac{1}{2}\|G(x^k + d^k) - G(x^*)\|^2 \\
&= O(\|x^k + d^k - x^*\|^2) \\
&= o(\|d^k\|^2).
\end{aligned}$$

Thus from Lemma 2.3 and Theorem 5.2, for all x^k sufficiently close to x^*,

$$\begin{aligned}
&f(x^k+d^k)-f(x^k)-\sigma\nabla f(x^k)^Td^k\\
&\quad=o(\|d^k\|^2)-\frac{1}{2}\|G(x^k)\|^2+\sigma(d^k)^T(V_k^TV_k+p_1\|G(x^k)\|^{p_2}I)d^k\\
&\quad=-\frac{1}{2}\|G(x^k)-G(x^*)\|^2+\sigma(d^k)^T(V_k^TV_k)d^k+o(\|d^k\|^2)\\
&\quad=-\frac{1}{2}(\|V_k(x^k-x^*)\|+o(\|x^k-x^*\|))^2+\sigma(d^k)^T(V_k^TV_k)d^k+o(\|d^k\|^2)\\
&\quad=-\frac{1}{2}\|V_k(-d_k+x^k+d^k-x^*)\|^2+\sigma(d^k)^T(V_k^TV_k)d^k+o(\|d^k\|^2)\\
&\quad=-\frac{1}{2}(d^k)^T(V_k^TV_k)d^k+\sigma(d^k)^T(V_k^TV_k)d^k+o(\|d^k\|^2)\\
&\quad=\left(\sigma-\frac{1}{2}\right)(d^k)^T(V_k^TV_k)d^k+o(\|d^k\|^2)\\
&\quad<0.
\end{aligned}$$

Then we can deduce that for all x^k sufficiently close to x^*,

$$x^{k+1}=x^k+d^k.$$

Thus from (7.1) we have proved that $\{x^k\}$ converges to x^* Q-superlinearly.

Finally, if F' is locally Lipschitz continuous around x^* and $p_2\geq 1$, we can easily modify the above arguments to get the Q-quadratic convergence of $\{x^k\}$. □

COROLLARY 7.3. *If F is a uniform P-function, then the sequence $\{x^k\}$ generated by the damped Gauss–Newton method is bounded and converges to the unique solution x^* of VIP(l,u,F) Q-superlinearly. Furthermore, if F' is locally Lipschitz continuous around x^* and $p_2\geq 1$, then the convergence is Q-quadratic.*

Proof. From Theorem 3.2 the level set $L_{f(x^0)}(f)$ is bounded. Then the sequence $\{x^k\}$ generated by the damped Gauss–Newton method is bounded and hence has at least one accumulation point, say, $\bar{x}$. According to Theorem 7.1, $\bar{x}$ is a stationary point of f. From Theorem 4.2, this stationary point $\bar{x}$ must be a solution of VIP(l,u,F) because $F'(\bar{x})$ is a P-matrix under the assumption that F is a uniform P-function. Since F is a uniform P-function, VIP(l,u,F) has a unique solution x^* (see, for example, [23, Theorem 3.9]). This means that $\bar{x}=x^*$. The conclusions of this corollary follow from Theorem 7.2 and the fact that if $F'(x^*)$ is a P-matrix, then VIP(l,u,F) is R-regular at x^*. □

Corollary 7.3 says that if F is a continuously differentiable uniform P-function, then the sequence $\{x^k\}$ generated by the damped Gauss–Newton method based on the new merit function f is well defined and converges to the unique solution of VIP(l,u,F) superlinearly. Such a result was only obtained for the nonlinear complementarity problem based on the merit function $\theta(x)$ (see, for example, [6]). In [27, 44] a similar result based on the D-gap function $h_{\alpha\beta}$ was obtained by assuming that F is a strongly monotone function, which is a stronger condition than that of a uniform P-function. Additionally, by technically choosing a sequence of smooth mappings $H_\varepsilon(x)$, $\varepsilon\to 0^+$ to approximate the nonsmooth mapping $H(x)$, Chen, Qi, and Sun [4] gave a similar result to Corollary 7.3 based on the so-called Jacobian consistency property. Here we directly construct a continuously differentiable merit function to obtain Corollary 7.3 instead of constructing a series of smooth approximating functions.

8. Numerical results. In this section we present some numerical experiments for the algorithm proposed in section 6 using the whole set of test problems from GAMS and MCP libraries (GAMSLIB and MCPLIB) [2, 8, 15]. The algorithm was implemented in MATLAB and run on a Sun SPARC Server 3002. Instead of a monotone linesearch we used a nonmonotone version as described in [10], which was originally due to Grippo, Lampariello, and Lucidi [22] and can be stated as follows. Let $\ell \geq 1$ be a prespecified constant and $\ell_k \geq 1$ be an integer which is adjusted at each iteration k. Calculate a steplength $t_k > 0$ satisfying the nonmonotone Armijo-rule

$$f(x^k + t_k d^k) \leq \mathcal{W}_k + \sigma t_k \nabla f(x^k)^T d^k, \tag{8.1}$$

where $\mathcal{W}_k := \max\{f(x^j) | j = k+1-\ell_k, \ldots, k\}$ denotes the maximal function value of f over the last ℓ_k iterations. Note that $\ell_k = 1$ corresponds to the monotone Armijo-rule. In the implementation, we used the following adjustment of ℓ_k:

1. Set $\ell_k = 1$ for $k = 0, 1, 2, 3, 4$, i.e., start the algorithm using the monotone Armijo-rule for the first four steps.
2. $\ell_{k+1} = \min\{\ell_k + 1, \ell\}$ at all remaining iterations ($\ell = 5$ in our implementation).

Throughout the computational experiments the starting points are provided by GAMSLIB or MCPLIB. The parameters used in the algorithm were $\rho = 0.5$, $p_1 = 5.0 \times 10^{-7}/\sqrt{n}$ $(n < 100), 10^{-6}/n$ $(n \geq 100)$, $p_2 = 1$, and $\sigma = 10^{-4}$. We replaced the term $p_1 \|G(x^k)\|^{p_2}$ in the algorithm by $\min\{p_0, p_1 \|G(x^k)\|^{p_2}\}$ with $p_0 = 10^{-4}$. If $n > 2500$, instead of using a Gauss–Newton direction, we simply used a pure Gauss–Newton direction $d^k = -(V_k^T V_k)^{-1} \nabla f(x^k) = -V_k^{-1} G(x^k)$, which is actually a (generalized) Newton direction. The iteration of the algorithm is stopped if either

$$f(x^k)/n \leq 10^{-12} \qquad \text{or} \qquad \|\nabla f(x^k)\|/\sqrt{n} \leq 10^{-10}$$

or if either

—the number of iterations exceeds 300, or

—the number of linesearch steps exceeds 40, giving a stepsize $t_k < 9.09 \times 10^{-13}$.

Finally we note that in our algorithm we assume that F is well defined everywhere, whereas there are a few examples in the GAMSLIB and MCPLIB where the function F may not be defined outside of S or even on the boundary of S. To partially avoid this problem our implementation used the following heuristic technique introduced in [10]. Let t denote a stepsize for which inequality (8.1) shall be tested. Before testing check whether $F(x^k + td^k)$ is well defined. If $F(x^k + td^k)$ is not well defined, then set $t := t/2$ and check again. Repeat this process until F is well defined or the limit of 40 linesearch steps is exceeded. In the first case continue with the nonmonotone Armijo linesearch. Otherwise the algorithm stops. This is equivalent to taking $f(x) = \infty$ for all points x where $F(x)$ is not defined.

The numerical results are summarized in Table 8.1 for the GAMSLIB problems and Tables 8.2–8.4 for the MCPLIB problems. In these tables the first column gives the name of the problem; n is the number of the variables in the problem; Nit denotes the number of iterations (LSF means the maximum number of linesearch steps was exceeded); NF denotes the number of evaluations of the function F; f_0 and f_F denote the value of f/n at the starting point and the final iterate, respectively; $\|\nabla f_F\|$ denotes the value of $\|\nabla f\|/\sqrt{n}$ at the final iterate; and CPU denotes the CPU time in seconds for the MATLAB implementation. Nit is equal to the number of evaluations of the Jacobian $F'(x)$ and the number of subproblems (6.1) or systems of linear equations solved. In the "Problem" column of Tables 8.2–8.4, the number after each problem specifies which starting point from the library is used. In the "f_0" column of Tables

Table 8.1
Numerical results for the problems from GAMSLIB.

Problem	n	f_0	Nit	NF	f_F	$\|\nabla f_F\|$	CPU
cafemge	101	$1.6\times10^{+1}$	7	12	7.2×10^{-16}	2.0×10^{-6}	0.6
cammcp	242	$1.6\times10^{+2}$	7	10	3.6×10^{-16}	5.5×10^{-6}	1.4
cammge	128	4.0×10^{-15}	0	1	4.0×10^{-15}	2.8×10^{-5}	0.2
cirimge	9	$1.1\times10^{+3}$	5	7	4.0×10^{-14}	3.7×10^{-5}	0.2
co2mge	208	3.1×10^{-15}	0	1	3.1×10^{-14}	1.1×10^{-5}	0.2
dmcmge	170	7.3	89	514	1.4×10^{-13}	$1..4\times10^{-3}$	23.2
ers82mcp	232	7.0	7	9	8.5×10^{-20}	4.2×10^{-8}	2.9
etamge	114	$1.0\times10^{+1}$	15	25	2.4×10^{-21}	5.8×10^{-8}	1.1
finmge	153	1.4×10^{-16}	0	1	1.4×10^{-16}	6.1×10^{-7}	0..2
gemmcp	262	9.6×10^{-14}	0	1	9.6×10^{-14}	1.1×10^{-5}	0.1
gemmge	178	1.4×10^{-13}	0	1	1.4×10^{-13}	2.3×10^{-6}	0.2
hansmcp	43	4.1	36	87	3.2×10^{-19}	2.3×10^{-9}	1.8
hansmge	43	3.7	14	41	3.2×10^{-25}	2.3×10^{-12}	1.0
harkmcp	32	$3.6\times10^{+1}$	23	44	1.6×10^{-13}	8.6×10^{-7}	0.7
harmge	11	$2.9\times10^{+2}$	24	57	1.3×10^{-17}	6.7×10^{-8}	0.7
kehomge	9	$1.9\times10^{+1}$	12	17	4.2×10^{-16}	1.2×10^{-6}	0.3
kormcp	78	$7.3\times10^{+2}$	5	6	6.3×10^{-13}	1.2×10^{-3}	0.3
mr5mcp	350	$2.9\times10^{+2}$	9	11	3.5×10^{-15}	2.3×10^{-5}	1.7
nsmge	212	$1.6\times10^{+1}$	15	25	2.1×10^{-20}	2.0×10^{-9}	3.8
oligomcp	6	$8..8\times10^{+2}$	6	9	2.7×10^{-21}	9.2×10^{-10}	0.2
sammge	23	0	0	1	0	0	0.1
scarfmcp	18	$1.2\times10^{+1}$	7	10	1.8×10^{-16}	6.9×10^{-7}	0.2
scarfmge	18	6.5	11	18	5.1×10^{-16}	3.6×10^{-7}	0.4
shovmge	51	9.4×10^{-9}	1	2	1.9×10^{-16}	9.6×10^{-7}	0.2
threemge	9	0	0	1	0	0	0.1
transmcp	11	$1.2\times10^{+4}$	6	21	2.6×10^{-5}	6.8×10^{-11}	0.3
two3mcp	6	$2.0\times10^{+2}$	7	10	2.2×10^{-17}	2.6×10^{-7}	0.2
unstmge	5	5.5×10^{-2}	8	10	7.6×10^{-19}	1.6×10^{-9}	0.2
vonthmcp	125	$2.1\times10^{+4}$	> 300	-	3.5	$2.4\times10^{+8}$	-
vonthmge	80	$3.3\times10^{+4}$	18(LSF)	-	$3.5\times10^{+2}$	$1.9\times10^{+6}$	-
wallmcp	6	1.2	4	5	8.8×10^{-26}	2.9×10^{-12}	0.2

8.1–8.4, DomainV means that the starting point is not in the domain of function or Jacobian.

Tables 8.1–8.4 show that the algorithm was able to solve most problems in GAMSLIB and MCPLIB. More precisely, for the GAMSLIB, for the problem `transmcp` our algorithm converged to a local minimum of $f(x)$ with $f_F = 2.6 \times 10^{-5}$ and $\|\nabla f_F\| = 6.8 \times 10^{-11}$. This is not strange because our algorithm can be used only to find local solutions of f, which may not be solutions of VIP(l, u, F). We also have failures on the problems `vonthmcp` and `vonthmge`. These are two von Thünen problems which are known to be very hard. By choosing different parameters we can solve `transmcp` and `vonthmge` with high precision but still fail on `vonthmcp`. On problems from the MCPLIB we have more failures. However, by using different parameters than those reported here, we can also solve all these failed problems except for `billups` and `pgvon105` with the second starting point, which violates the domain of Jacobian evaluation. The `billups` problem was constructed by Billups [1] in order to make almost all state-of-the-art methods fail on this problem. Note that the function F in the

Table 8.2

Numerical results for the problems from MCPLIB.

Problem	n	f_0	Nit	NF	f_F	$\|\nabla f_F\|$	CPU
bertsekas(1)	15	1.8×10^{-2}	30	115	1.5×10^{-21}	5.0×10^{-9}	1.1
bertsekas(2)	15	1.0×10^{-2}	30	107	1.6×10^{-21}	5.1×10^{-9}	1.1
bertsekas(3)	15	$4.3\times10^{+3}$	32	126	3.8×10^{-14}	2.5×10^{-5}	1.2
billups	1	5.0×10^{-5}	132	3494	1.0×10^{-5}	1.0×10^{-12}	4.6
bert_oc	5000	2.5×10^{-2}	4	6	2.5×10^{-31}	1.1×10^{-15}	49.2
bratu	5625	2.3×10^{-3}	12	50	1.9×10^{-17}	2.0×10^{-8}	141.0
choi	13	2.2×10^{-3}	4	5	3.2×10^{-17}	5.5×10^{-9}	0.6
colvdual(1)	20	$2.0\times10^{+1}$	> 300	-	2.5×10^{-4}	1.0×10^{-1}	-
colvdual(2)	20	$3.3\times10^{+2}$	> 300	-	2.5×10^{-4}	1.1×10^{-1}	-
colvnlp(1)	15	$2.7\times10^{+1}$	14	36	9.2×10^{-26}	3.6×10^{-11}	0.5
colvnlp(2)	15	$4.4\times10^{+1}$	11	20	1.1×10^{-13}	6.6×10^{-5}	0.4
cycle	1	4.4×10^{-1}	3	5	3.3×10^{-14}	2.6×10^{-7}	0.2
ehl_k40	41	$3.1\times10^{+3}$	38	158	1.6×10^{-18}	2.8×10^{-6}	5.1
ehl_k60	61	$9.2\times10^{+3}$	> 300	-	$3.4\times10^{+1}$	$1.3\times10^{+6}$	-
ehl_k80	81	$2.0\times10^{+4}$	134	1050	3.8×10^{-14}	1.0×10^{-3}	93.7
ehl_kost	101	$3.4\times10^{+4}$	> 300	-	$2.7\times10^{+1}$	$1.1\times10^{+5}$	-
explcp	16	5.0×10^{-1}	22	68	4.7×10^{-18}	3.1×10^{-9}	0.7
freebert(1)	15	1.8×10^{-2}	27	110	1.6×10^{-21}	5.1×10^{-9}	1.1
freebert(2)	15	$5.2\times10^{+6}$	59	108	1.7×10^{-21}	5.2×10^{-9}	1.2
freebert(3)	15	1.8×10^{-2}	26	106	1.6×10^{-21}	5.1×10^{-9}	1.1
freebert(4)	15	1.8×10^{-2}	30	115	1.5×10^{-21}	5.0×10^{-9}	1.2
freebert(5)	15	$5.2\times10^{+6}$	271	374	1.6×10^{-21}	5.1×10^{-9}	4.1
freebert(6)	15	1.8×10^{-2}	27	110	5.7×10^{-14}	3.0×10^{-5}	1.0
gafni(1)	5	5.3×10^{-2}	11	20	1.2×10^{-15}	5.7×10^{-6}	0.3
gafni(2)	5	1.4×10^{-2}	12	30	7.9×10^{-13}	1.5×10^{-4}	0.4
gafni(3)	5	5.5×10^{-2}	29	40	2.1×10^{-16}	2.4×10^{-6}	0.5
hanskoop(1)	14	3.8×10^{-1}	17	41	1.1×10^{-18}	7.7×10^{-9}	0.6
hanskoop(2)	14	1.3	18	45	2.9×10^{-15}	8.9×10^{-7}	0.6
hanskoop(3)	14	1.8×10^{-1}	57	172	2.6×10^{-19}	5.7×10^{-9}	1.9
hanskoop(4)	14	1.1×10^{-1}	22	116	2.8×10^{-19}	6.3×10^{-9}	1.2
hanskoop(5)	14	$2.1\times10^{+2}$	116	235	6.8×10^{-21}	1.4×10^{-9}	2.6
hydroc06	29	2.2×10^{-1}	5	7	2.3×10^{-17}	1.5×10^{-7}	0.2
hydroc20	99	1.6×10^{-1}	16	21	7.1×10^{-14}	1.8×10^{-6}	0.8
jel	6	$2.0\times10^{+2}$	7	10	2.2×10^{-17}	2.6×10^{-7}	0.2
josephy(1)	4	6.3	6	10	1.1×10^{-14}	1.4×10^{-6}	0.2
josephy(2)	4	4.3×10^{-1}	6	9	2.1×10^{-19}	5.9×10^{-9}	0.2
josephy(3)	4	$5.0\times10^{+3}$	> 300	-	3.8×10^{-2}	1.2	-

`billups` problem is pseudomonotone at a solution, which is exactly what is needed for some globally convergent methods [43]. To solve this problem, we can first use the method in [43] to make the iterates approximate the solution to some extent and then switch to the above algorithm. In fact, by using the method in [43], after 76 iterations and 147 function evaluations we get a final x with $|\min\{x, F(x)\}| = 3.8 \times 10^{-8}$. Note that for `pgvon106` we have $\|\nabla f_F\| = 2.1 \times 10^{+1}$ while f_F is very small. This also confirms that `pgvon106` is really a hard problem. The main focus of this paper is problems with both lower bounds and upper bounds on the variables. Some of the larger examples are `bratu` with $n = 5625$, `opt_cont127`, `opt_cont255`, and `opt_cont511`

Table 8.3
Numerical results for the problems from MCPLIB (continued).

Problem	n	f_0	Nit	NF	f_F	$\|\nabla f_F\|$	CPU
josephy(4)	4	6.0×10^{-1}	5	6	1.0×10^{-21}	4.1×10^{-10}	0.2
josephy(5)	4	1.6	4	5	4.1×10^{-22}	2.6×10^{-10}	0.2
josephy(6)	4	1.3	6	9	3.6×10^{-21}	7.6×10^{-10}	0.2
kojshin(1)	4	$1.6\times10^{+1}$	> 300	-	1.7×10^{-1}	3.8×10^{-1}	-
kojshin(2)	4	4.3×10^{-1}	7	15	1.3×10^{-13}	1.9×10^{-6}	0.2
kojshin(3)	4	$5.0\times10^{+3}$	10	14	1.6×10^{-14}	1.6×10^{-6}	0.2
kojshin(4)	4	2.5	2	3	1.2×10^{-20}	1.9×10^{-9}	0.1
kojshin(5)	4	6.1	4	5	6.4×10^{-22}	5.0×10^{-10}	0.2
kojshin(6)	4	4.4	7	9	1.5×10^{-24}	1.5×10^{-11}	0.2
mathinum(1)	3	2.9×10^{-1}	18	35	4.7×10^{-13}	1.9×10^{-6}	0.4
mathinum(2)	3	2.9×10^{-1}	18	35	4.7×10^{-13}	1.9×10^{-6}	0.4
mathinum(3)	3	9.7	25	62	3.2×10^{-13}	1.9×10^{-6}	0.6
mathinum(4)	3	2.1	6	7	1.8×10^{-20}	5.1×10^{-10}	0.9
mathisum(1)	4	2.0×10^{-1}	4	6	3.3×10^{-13}	1.9×10^{-6}	0.2
mathisum(2)	4	1.5	6	7	1.0×10^{-21}	1.1×10^{-10}	0.2
mathisum(3)	4	$3.8\times10^{+1}$	5	7	1.3×10^{-20}	1.0×10^{-10}	0.2
mathisum(4)	4	8.1×10^{-1}	5	6	2.7×10^{-19}	1.8×10^{-9}	0.2
methan08	31	1.1	4	5	2.4×10^{-21}	2.3×10^{-9}	0.2
nash(1)	10	$1.0\times10^{+4}$	6	7	1.9×10^{-17}	1.8×10^{-7}	0.2
nash(2)	10	$4.0\times10^{+1}$	9	20	6.5×10^{-17}	8.6×10^{-7}	0.3
obstacle(1)	2500	1.4×10^{-4}	10	11	$2..6\times10^{-23}$	3.2×10^{-11}	12.9
obstacle(2)	2500	1.4×10^{-2}	12	15	2.6×10^{-23}	3.2×10^{-11}	17.0
opt_cont31	1024	4.0×10^{-2}	9	16	4.4×10^{-24}	5.4×10^{-13}	10.6
opt_cont127	4096	5.8×10^{-3}	7	19	1.1×10^{-13}	1.3×10^{-6}	171.6
opt_cont255	8193	2.2×10^{-3}	10	34	1.4×10^{-15}	1.5×10^{-7}	894.7
opt_cont511	16384	8.3×10^{-4}	11	45	3.9×10^{-13}	2.5×10^{-6}	6003.8
pgvon105(1)	105	$2.5\times10^{+1}$	31	100	8.7×10^{-20}	2.2×10^{-6}	4.9
pgvon105(2)	105	DomainV	-	-	-	-	-
pgvon105(3)	105	5.3×10^{-1}	37(LSF)	-	3.6×10^{-3}	$2.0\times10^{+2}$	-
pgvon106	106	$2.6\times10^{+1}$	57	233	7.9×10^{-13}	$2.1\times10^{+1}$	10..9
pies	42	$1.7\times10^{+4}$	10	11	1.2×10^{-19}	3.4×10^{-8}	0.3
powell(1)	16	5.6×10^{-1}	4	5	1.0×10^{-20}	1.4×10^{-9}	0.2
powell(2)	16	$1.2\times10^{+1}$	10	18	2.7×10^{-25}	1.3×10^{-11}	0.4
powell(3)	16	$2.9\times10^{+3}$	10	11	1.4×10^{-21}	4.5×10^{-10}	0.3
powell(4)	16	$1.5\times10^{+2}$	9	10	2.5×10^{-17}	6.7×10^{-8}	0.3
powell_mcp(1)	8	$2.9\times10^{+1}$	5	6	3.4×10^{-13}	$4..9\times10^{-6}$	0.2
powell_mcp(2)	8	$1.4\times10^{+2}$	6	7	1.1×10^{-13}	2.8×10^{-6}	0.2
powell_mcp(3)	8	$1.8\times10^{+4}$	8	9	1.4×10^{-16}	9.9×10^{-8}	0.2
powell_mcp(4)	8	$1.1\times10^{+3}$	7	8	1.1×10^{-15}	2.7×10^{-7}	0.2

with $4096, 8193$, and $16,394$ variables, respectively. All of these problems were solved to high accuracy within 12 iterations and 50 function evaluations.

The MATLAB implementation used for these numerical experiments is not very sophisticated. The Jacobian $F'(x^k)$ and the element V_k of the generalized Jacobian are stored as a sparse matrix, but then for small problems ($n \leq 2500$) the matrix $V_k^T V_k$ is formed directly, resulting in considerable fill-in. For large problems we simply calculate the generalized Newton direction using MATLAB's direct sparse linear

TABLE 8.4
Numerical results for the problems from MCPLIB (continued).

Problem	n	f_0	Nit	NF	f_F	$\|\nabla f_F\|$	CPU
scarfanum(1)	13	3.4	8	16	2.6×10^{-20}	5.6×10^{-9}	0.3
scarfanum(2)	13	4.5	12	33	2.6×10^{-20}	5.6×10^{-9}	0.5
scarfanum(3)	13	3.0	9	12	2.5×10^{-20}	5.5×10^{-9}	0.3
scarfasum(1)	14	5.4×10^{-1}	4	6	7.3×10^{-18}	1.7×10^{-7}	0.2
scarfasum(2)	14	4.1×10^{-1}	8	19	3.5×10^{-19}	7.4×10^{-8}	0.3
scarfasum(3)	14	2.4	11	24	4.7×10^{-19}	8.5×10^{-8}	0.4
scarfbnum(1)	39	$1.0\times10^{+2}$	46	94	2.0×10^{-13}	6.0×10^{-5}	1.7
scarfbnum(2)	39	$1.1\times10^{+2}$	13	18	3.8×10^{-15}	3.2×10^{-5}	0.5
scarfbsum(1)	40	$6.1\times10^{+1}$	9	21	1.8×10^{-16}	2.5×10^{-6}	0.5
scarfbsum(2)	40	$6.8\times10^{+1}$	32(LSF)	-	4.0×10^{-1}	$5.8\times10^{+1}$	-
sppe(1)	27	$1.1\times10^{+2}$	11	19	6.8×10^{-24}	1.2×10^{-11}	0.4
sppe(2)	27	$4.8\times10^{+1}$	7	8	8.5×10^{-20}	1.4×10^{-9}	0.2
tobin(1)	42	$1.8\times10^{+2}$	9	15	1.8×10^{-16}	4.3×10^{-8}	0..4
tobin(2)	42	$1.8\times10^{+2}$	12	15	2.0×10^{-22}	6.8×10^{-11}	0.4

equation solver. In particular, note from the formulae in section 4 that the generalized Jacobian V_k has rows which consist of either the unit vector e_i, the corresponding row $F'_i(x^k)$ of the Jacobian of F, or a linear combination of these terms. Thus V_k has at least the sparsity structure of $F'(x^k)$ and often considerably more when a row $F'_i(x^k)$ is replaced by the (scaled) unit vector e_i. Thus there is considerable potential to exploit the sparsity of V_k, for example, by reordering the columns to produce more efficient matrix factorizations. In particular, if the sparsity of $F'(x)$ does not change, then this reordering could be done once rather than on every iteration.

The numerical experiments in this paper are simply meant to demonstrate the viability of the proposed merit function $f(x)$ for solving $VIP(l,u,F)$. Further work is needed to produce robust, efficient software.

9. Final remarks.

In this paper we presented a new differentiable merit function for solving a box constrained variational inequality problem. This new merit function has many desirable properties over the existing ones. The key idea is to use the fact that

$$\psi(a,b)=0 \Longleftrightarrow a\geq 0, \quad ab_+=0$$

to reformulate $\mathrm{VIP}(l,u,F)$ as the minimization of an unconstrained differentiable merit function. This reformulation allows us to construct a globally and superlinearly convergent damped Gauss–Newton method for solving $\mathrm{VIP}(l,u,F)$. One of the most important features of the damped Gauss–Newton method introduced here is that at each iteration we need only to solve a linear system of equations. Besides the formula introduced in this paper, there are other possible functions $\psi(a,b)$. For example we can let

$$\psi^{\mathrm{new}}(a,b):=([-\phi_R(a,b)]_+)^2+([-a]_+)^2, \tag{9.1}$$

where $\phi_R:\mathbb{R}^2\to\mathbb{R}$ is defined by

$$\phi_R(a,b):=\phi(a,b)-a_+b_+=\sqrt{a^2+b^2}-(a+b)-a_+b_+ \tag{9.2}$$

and can be regarded as a regularized Fischer–Burmeister function. It is not difficult to verify that $\phi_R(\cdot)^2$ and $\psi^{\text{new}}(\cdot)$ are continuously differentiable functions on $\mathbb{R}^2$ (for any $b \in \mathbb{R}$ we define $\phi_R(+\infty, b) = -b$). This modification may enhance the boundedness results of the corresponding merit function. Chen, Chen, and Kanzow [3] reported some interesting properties of the modified Fischer–Burmeister function

$$\begin{aligned} \phi_{CCK}(a,b) &= -[\lambda\phi(a,b) - (1-\lambda)a_+b_+] \\ &= -\lambda\left(\phi(a,b) - \tfrac{1-\lambda}{\lambda}a_+b_+\right), \quad \lambda \in (0,1). \end{aligned} \tag{9.3}$$

By letting $\alpha = \frac{1-\lambda}{\lambda}$ and ignoring the outside $-\lambda$ parameter, the function ϕ_{CCK} defined in (9.3) takes the form

$$\phi_{CCK}(a,b) = \phi(a,b) - \alpha a_+b_+, \quad \alpha \in (0,\infty). \tag{9.4}$$

Note that $a, b \geq 0$ and $ab = 0$ is equivalent to $\alpha a, \alpha b \geq 0$ and $(\alpha a)(\alpha b) = 0$ for any $\alpha > 0$. The function ϕ_{CCK} defined in (9.4) can be treated as a scaled form of ϕ_R. Numerically this scaling can play an important role in the behavior of the corresponding algorithm. The application of ϕ_R or ϕ_{CCK} to box constrained variational inequality problems needs further investigation.

In this paper we introduced the merit function f for VIP(l, u, F) without considering whether l_i, u_i, $i = 1, \ldots, n$ are finite or not. However, if some l_i and/or u_i are infinite, we can modify our merit function. For example we can define

$$\begin{aligned} f^{\text{new}}(x) := \frac{1}{2}\Bigg[&\sum_{i\in I_l}\phi(x_i - l_i, F_i(x))^2 + \sum_{i\in I_u}\phi(u_i - x_i, -F_i(x))^2 \\ &+ \sum_{i\in I_{lu}}\psi(x_i - l_i, F_i(x)) + \sum_{i\in I_{lu}}\psi(u_i - x_i, -F_i(x))\Bigg], \end{aligned} \tag{9.5}$$

where

$$\begin{aligned} I_l &:= \{i|\ l_i > -\infty,\ u_i = \infty,\ i = 1, \ldots, n\}, \\ I_u &:= \{i|\ l_i = -\infty,\ u_i < \infty,\ i = 1, \ldots, n\}, \\ I_{lu} &:= \{1, \ldots, n\}\backslash\{I_l \cup I_u\}. \end{aligned}$$

The function $f^{\text{new}}(x)$ has similar properties to $f(x)$ and possibly has only slightly different stationary point conditions. Roughly speaking, the stationary point conditions for f need a stronger nonsingularity condition on ∇F, while f^{new} needs a stronger P_0 property on ∇F (i.e., the set $\mathcal{P}$ in Theorem 4.2 may contain fewer elements). Note that in (9.5) the functions $\phi(\cdot)$ and $\psi(\cdot)$ can be replaced by $\phi_R(\cdot)$ and $\psi^{\text{new}}(\cdot)$, respectively.

Acknowledgments. The authors would like to thank the two anonymous referees and the associate editor for their helpful comments, which improved the presentation of this paper. Thanks also to Houduo Qi for his comments on (9.1) and (9.2).

REFERENCES

[1] S. C. Billups, *Algorithms for complementarity problems and generalized equations*, Ph.D. thesis, Computer Sciences Department, University of Wisconsin, Madison, WI, 1995.

[2] S. C. Billups, S. P. Dirkse, and M. C. Ferris, *A comparison of algorithms for large scale mixed complementarity problems*, Comput. Optim. Appl., 7 (1997), pp. 3–25.

[3] B. Chen, X. Chen, and C. Kanzow, *A Modified Fischer-Burmeister NCP-function*, Talk presented at the 1997 International Symposium on Mathematical Programming, EPFL, Lausanne, Switzerland, 1997.

[4] X. Chen, L. Qi, and D. Sun, *Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities*, Math. Comp., 67 (1998), pp. 519–540.

[5] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley, New York, 1983; reprinted by SIAM, Philadelphia, 1990.

[6] T. De Luca, F. Facchinei, and C. Kanzow, *A semismooth equation approach to the solution of nonlinear complementarity problems*, Math. Programming, 75 (1996), pp. 407–439.

[7] S. P. Dirkse and M. C. Ferris, *The PATH solver: A non-monotone stabilization scheme for mixed complementarity problems*, Optimization Methods and Software, 5 (1995), pp. 123–156.

[8] S. P. Dirkse and M. C. Ferris, *MCPLIB: A collection of nonlinear mixed complementarity problems*, Optim. Methods Softw., 5 (1995), pp. 319–345.

[9] B. C. Eaves, *On the basic theorem of complementarity*, Math. Programming, 1 (1971), pp. 68–75.

[10] F. Facchinei, A. Fischer, and C. Kanzow, *A semsimooth Newton method for variational inequalities: The case of box constraints*, in Complementarity and Variational Problems: State of the Art, M. C. Ferris and J. S. Pang, eds., SIAM, Philadelphia, PA, 1997, pp. 76–90.

[11] F. Facchinei and C. Kanzow, *On unconstrained and constrained stationary points of the implicit Lagrangian*, J. Optim. Theory Appl., 92 (1997), pp. 99–115.

[12] F. Facchinei and C. Kanzow, *A nonsmooth inexact Newton method for the solution of large-scale nonlinear complementarity problems*, Math. Programming, 76 (1997), pp. 493–512.

[13] F. Facchinei and J. Soares, *Testing a new class of algorithms for nonlinear complementarity problems*, in Variational Inequalities and Network Equilibrium Problems, F. Giannessi, ed., Plenum Press, New York, 1995, pp. 69–83.

[14] F. Facchinei and J. Soares, *A new merit function for nonlinear complementarity problems and a related algorithm*, SIAM J. Optim., 7 (1997), pp. 225–247.

[15] M. C. Ferris and T. F. Rutherford, *Accessing realistic mixed complementarity problems within MATLAB*, in Nonlinear Optimization and Applications, G. Di Pillo and F. Giannessi, eds., Plenum Press, New York, 1996, pp. 141–153.

[16] A. Fischer, *A special Newton-type optimization method*, Optimization, 24 (1992), pp. 269–284.

[17] A. Fischer, *An NCP-function and its use for the solution of complementarity problems*, in Recent Advances in Nonsmooth Optimization, D. Du, L. Qi, and R. Womersley, eds., World Scientific, River Edge, NJ, 1995, pp. 88–105.

[18] A. Fischer, *Solution of monotone complementarity problems with locally Lipschitzian functions*, Math. Programming, 76 (1997), pp. 513–532.

[19] M. Fukushima, *Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems*, Math. Programming, 53 (1992), pp. 99–110.

[20] M. Fukushima, *Merit functions for variational inequality and complementarity problems*, in Nonlinear Optimization and Applications, G. Di Pillo and F. Giannessi, eds., Plenum Press, New York, 1996, pp. 155–170.

[21] C. Geiger and C. Kanzow, *On the resolution of monotone complementarity problems*, Comput. Optim. Appl., 5 (1996), pp. 155–173.

[22] L. Grippo, F. L. Lampariello, and S. Lucidi, *A nonmonotone linesearch technique for Newton's method*, SIAM J. Numer. Anal., 23 (1986), pp. 707–716.

[23] P. T. Harker and J. S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 161–220.

[24] H. Jiang, *Unconstrained minimization approaches to nonlinear complementarity problems*, J. Global Optim., 9 (1996), pp. 169–181.

[25] H. Jiang and L. Qi, *A new nonsmooth equations approach to nonlinear complementarity problems*, SIAM J. Control Optim., 35 (1997), pp. 178–193.

[26] C. Kanzow, *Nonlinear complementarity as unconstrained optimization*, J. Optim. Theory Appl., 88 (1996), pp. 139–155.

[27] C. Kanzow and M. Fukushima, *Theoretical and numerical investigation of the D-gap function for box constrained variational inequalities*, Math. Programming, 83 (1998), pp. 55–87.

[28] Z. -Q. Luo, O. L. Mangasarian, J. Ren, and M. V. Solodov, *New error bounds for the linear complementarity problem*, Math. Oper. Res., 19 (1994), pp. 880–892.

[29] Z. -Q. Luo and P. Tseng, *A new class of merit functions for the nonlinear complementarity problem*, in Complementarity and Variational Problems: State of the Art, M. C. Ferris and J. S. Pang, eds., SIAM, Philadelphia, PA, 1997, pp. 204–225.

[30] O. L. Mangasarian and M. V. Solodov, *Nonlinear complementarity as unconstrained and constrained minimization*, Math. Programming, 62 (1993), pp. 277–297.

[31] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 15 (1977), pp. 959–972.

[32] J. S. Pang, *Newton's method for B-differentiable equations*, Math. Oper. Res., 15 (1990), pp. 331–341.

[33] J. S. Pang, *A B-differentiable equation based, globally and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational inequality problems*, Math. Programming, 51 (1991), pp. 101–131.

[34] J. S. Pang, *Complementarity problems*, in Handbook of Global Optimization, R. Horst and P. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA, 1995, pp. 271–338.

[35] J. S. Pang and S. A. Gabriel, *NE/SQP: A robust algorithm for the nonlinear complementarity problem*, Math. Programming, 60 (1993), pp. 295–337.

[36] J. S. Pang and L. Qi, *Nonsmooth equations: Motivation and algorithms*, SIAM J. Optim., 3 (1993), pp. 443–465.

[37] J. M. Peng, *Equivalence of variational inequality problems to unconstrained optimization*, Math. Programming, 78 (1997), pp. 347–355.

[38] J. M. Peng and Y. Yuan, *Unconstrained methods for generalized complementarity problems*, J. Comput. Math., 15 (1997), pp. 253–264.

[39] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.

[40] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–367.

[41] D. Ralph, *Global convergence of damped Newton's method for nonsmooth equations via the path search*, Math. Oper. Res., 19 (1994), pp. 352–389.

[42] S. M. Robinson, *Generalized equations*, in Mathematical Programming: The State of the Art, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 346–367.

[43] D. Sun, *A class of iterative methods for solving nonlinear projection equations*, J. Optim. Theory Appl., 91 (1996), pp. 123–140.

[44] D. Sun, M. Fukushima, and L. Qi, *A computable generalized Hessian of the D-gap function and Newton-type methods for variational inequality problems*, in Complementarity and Variational Problems: State of the Art, M. C. Ferris and J. S. Pang, eds., SIAM, Philadelphia, PA, 1997, pp. 452–473.

[45] P. Tseng, *Global behaviour of a class of merit functions for the nonlinear complementarity problem*, J. Optim. Theory Appl., 89 (1996), pp. 17–37.

[46] P. Tseng, N. Yamashita, and M. Fukushima, *Equivalence of complementarity problems to differentiable minimization: A unified approach*, SIAM J. Optim., 6 (1996), pp. 446–460.

[47] N. Yamashita and M. Fukushima, *On stationary points of implicit Lagrangian for nonlinear complementarity problems*, J. Optim. Theory Appl., 84 (1995), pp. 653–663.

[48] N. Yamashita and M. Fukushima, *Equivalent unconstrained minimization and global error bounds for variational inequality problems*, SIAM J. Control Optim., 35 (1997), pp. 273–284.

[49] N. Yamashita, K. Taji, and M. Fukushima, *Unconstrained optimization reformulations of variational inequality problems*, J. Optim. Theory Appl., 92 (1997), pp. 439–456.

SIAM J. Optim.
Vol. 9, No. 2, pp. 414–443

OPTIMAL SIZING FOR A CLASS OF NONLINEARLY ELASTIC MATERIALS[*]

CHRISTOPH STANGL[†]

Abstract. The intention of this paper is to develop efficient algorithms for solving sizing optimization problems for a class of nonlinearly elastic materials (including, e.g., gray cast iron that is used in car engines) within common finite element programming packages such as, e.g., MSC/NASTRAN. The nonlinear material law is such that Young's modulus depends on the stresses. Therefore it is not possible to use standard software to perform sizing optimization; new algorithms must be found.

We develop and analyze algorithms to solve the nonlinear discrete equilibrium equations and the discrete design problem. Moreover, we test our algorithms on practically relevant problems, namely to minimize the volume, respectively, to reach an even stress distribution, of a unit injector rocker arm made of gray cast iron.

Key words. sizing optimization, nonlinearly elastic materials, Newton's method, homotopy method, sensitivities, sequential quadratic programming algorithm

AMS subject classifications. 65K10, 93C20, 90C30, 65N30

PII. S1052623497319213

1. Introduction.

In this paper we will present a methodology for solving sizing optimization problems for nonlinearly elastic materials that are practically relevant, e.g., to minimize the volume of a given object under certain stress constraints or to reach an even stress distribution such that the object of interest is utilized in an optimal way.

In [15] such problems are solved for linearly elastic materials, where the sequential quadratic programming (SQP) algorithm is used together with an active set strategy in order to solve the optimization problem and the finite element programming package MSC/NASTRAN is used to compute displacements and stresses and their sensitivities.

We are interested in optimizing materials for which Young's modulus depends on the stress field. Therefore the displacements depend on the stresses, and the stress field depends on the displacements (as usual) and is given implicitly; e.g., gray cast iron (GGL25) that is used in car engines belongs to this class of nonlinearly elastic material laws. Furthermore, this stress dependence is also exhibited in magnetostrictive materials (cf. [4], [14]) with similar nonlinear material behavior noted in certain elastomers (cf. [2], [3]).

In [6] the following question is answered: under which assumptions does there exist a (unique) solution to the resulting nonlinear equilibrium equations in the mathematical theory of continuum mechanics for this class of material laws? We will use these results in the present paper as a basis to develop algorithms for solving the discrete nonlinear equilibrium equations (section 5) and to compute sensitivities (section 6). The methodology of [15] is used as a basis to solve the considered sizing optimization problems (section 7). Since we are interested only in solving the discrete problem, we

[*] Received by the editors March 31, 1997; accepted for publication (in revised form) January 25, 1998; published electronically March 17, 1999.

http://www.siam.org/journals/siopt/9-2/31921.html

[†] Institut für Industriemathematik, Johannes-Kepler-Universität, Altenbergerstrasse 69, A-4040 Linz, Austria (stangl@indmath.uni-linz.ac.at). The work of this author was supported by the Austrian Forschungsförderungsfonds für die Gewerbliche Wirtschaft and AVL LIST GmbH Graz.

will not deal with the question of discretization errors (therefore cf. [19]) and formulate the design problems as well as the equilibrium equations in finite dimensions.

In section 8 we will present some numerical results for the optimization of a unit injector rocker arm made of gray cast iron, which will show that the properties of the given object can be improved significantly by using our methodology. Analogously to [15] we couple our own finite element analysis modules and the optimization routines with the finite element programming package MSC/NASTRAN. Clearly our concept can be coupled with other common finite element codes without changes.

Before we start with our investigations, we will write down the weak formulation of the equilibrium equations of mathematical continuum mechanics for the considered class of nonlinearly elastic materials as given in [6] in two dimensions. Let ω be a bounded domain in $\mathbb{R}^2$, $\gamma = \partial\omega$ be sufficiently smooth, and $\gamma_1, \gamma_2 \subset \gamma$ such that $\bar{\gamma}_1 \cup \bar{\gamma}_2 = \bar{\gamma}$. Furthermore, let there be given body forces $\hat{F} : \omega \to \mathbb{R}^2$ and surface forces $G : \gamma_2 \to \mathbb{R}^2$, and let ω be fixed on γ_1. Then the weak equilibrium equations for the considered class of nonlinearly elastic materials are given by

$$U \in \mathcal{V}_0 \text{ such that } \; a_\sigma(U,V) = \langle \tilde{F}, V\rangle \qquad \forall\, V \in \mathcal{V}_0, \tag{1.1}$$

where

$$\mathcal{V}_0 := \{V \in \mathcal{V} := [H^1(\omega)]^2 : \; V|_{\gamma_1} \equiv 0\} \tag{1.2}$$

and

$$a_\sigma(U,V) := \int_\omega \frac{E(\sigma)}{1+\nu} \cdot \left[\frac{\nu}{1-\nu} tr(\varepsilon(U)) tr(\varepsilon(V)) + \varepsilon(U) : \varepsilon(V)\right] \, dX, \tag{1.3}$$

$$\langle \tilde{F}, V\rangle := \int_\omega \hat{F} \cdot V \; dX - \int_\omega \Delta T \cdot (\beta : (\nabla V)) \, dX + \int_{\gamma_2} G \cdot V \; dA \tag{1.4}$$

with

$$\varepsilon_{ij}(U(X)) \quad = \quad \tfrac{1}{2} \cdot (U_{i,j}(X) + U_{j,i}(X)) \quad \forall i,j = 1,2, \text{ in } \omega, \tag{1.5}$$

where $\beta = [\beta_{ij}]_{i,j=1}^2$ is the real matrix of thermal expansion coefficients; $\Delta T := T - T_{ref}$; $T_{ref} \in \mathbb{R}$ is the reference temperature (the temperature of ω before the deformation); $T = \text{const}$ is the temperature distribution over ω, which is assumed to be constant over ω throughout the whole paper; ":" denotes the matrix inner product; and $E : \mathbb{R}^3 \to \mathbb{R}$ is a given sufficiently smooth function. Moreover, σ is the solution to the equation

$$\sigma \in \mathcal{W} := [L_2(\omega)]^3 \text{ with } \sigma(X) = E(\sigma(X)) \cdot \tilde{G} \cdot \varepsilon(U(X)) + \Delta T \cdot \beta, \tag{1.6}$$

$$\text{almost everywhere (a.e.) in } \omega,$$

where $\sigma := (\sigma_{11}, \sigma_{22}, \sigma_{12})^T$, $\varepsilon := (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{12})^T$, $\beta := (\beta_{11}, \beta_{22}, \beta_{12})^T$, and $\tilde{G}$ is the modified matrix of elasticity coefficients (for the situation of plane stress)

$$\tilde{G} := \frac{1}{(1-\nu)(1+\nu)} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix},$$

where $0 \le \nu < 1/2$ is Poisson's ratio. We have stated this problem for the situation of plane stress since we will assume in section 3 that the object to be optimized is (in some sense) a thin plate.

Remark 1.1. *As posed by our industrial partner AVL for the gray cast iron GGL25, the material law function E is given by*

$$E(\underline{\sigma}) = y_0 + y_1 \cdot f(\underline{\sigma}) + y_2 \cdot (T - T_{ref}) \qquad \forall\ \underline{\sigma} \in \mathbb{R}^3, \tag{1.7}$$

where y_0, y_1, and y_2 are given real constants and $f : \mathbb{R}^3 \to \mathbb{R}$ is given by

$$f(\underline{\sigma}) := \sigma_{MAX}(\underline{\sigma}) = \frac{\sigma_{11} + \sigma_{22}}{2} + \sqrt{\left(\frac{\sigma_{11} - \sigma_{22}}{2}\right)^2 + \sigma_{12}^2}$$

(cf. also section 8). It can be shown that the mapping f is Lipschitz continuous with constant $L_f := 1 + \frac{\sqrt{2}}{2}$ and the mapping E is Lipschitz continuous with the constant $L_E := |y_1| \cdot L_f$ (cf., e.g., [6]).

2. Formulation of the discrete equilibrium equations. Since we are interested in solving design problems with the state equations (1.1) and (1.6) numerically, we will transform (1.1) and (1.6) onto finite dimensions via finite elements. Although the following considerations are quite standard, we will also present some details in order to introduce the notation used throughout this paper.

Let there be given a triangulation $\{\omega^{(i)}\}_{i \in I}$ of $\omega \subset \mathbb{R}^2$ with $I := \{1, \dots, Ne\}$, i.e.,

$$\begin{aligned} \textstyle\bigcup_{i \in I} \omega^{(i)} &= \omega, \\ \overline{\omega^{(i)}} \cap \overline{\omega^{(j)}} &= \begin{cases} \text{a common point of } \omega^{(i)} \text{ and } \omega^{(j)}, \text{ or} \\ \text{a common edge of } \omega^{(i)} \text{ and } \omega^{(j)}, \text{ or} \\ \emptyset \end{cases} \qquad \forall\ i \neq j \in I. \end{aligned} \tag{2.1}$$

We assume that all finite elements $\omega^{(i)}$ are linear triangles (our results also stay valid for bilinear parallelograms, respectively, combinations of both sorts of elements (cf. also [8, p. 193]); only some constants will change in our analysis). In order to establish a more comfortable treatment of the finite element analysis we introduce the variables

$Ng \in \mathbb{N}$ ... number of grid points,
$Ne \in \mathbb{N}$ ... number of elements,
$Nf \in \mathbb{N}$ ... number of unrestricted degrees of freedom

and the mappings ($\mathcal{P}(A)$ denotes the set of all subsets of the set A)

$$\begin{aligned} \mathbf{node} : \{1, \dots, Ne\} &\to \mathcal{P}(\{1, \dots, Ng\}), \\ i &\mapsto \{k \in \{1, \dots, Ng\} : \text{node k belongs to finite element i}\}, \\ \mathbf{dof} : \{1, \dots, Ne\} &\to \mathcal{P}(\{1, \dots, 6Ng\}), \\ i &\mapsto \{k \in \{1, \dots, 6Ng\} : D_k \text{ belongs to a node in } \mathbf{node}(i)\}, \\ \mathbf{\Delta} : \{1, \dots, Nf\} &\to \mathcal{P}(\{1, \dots, Ne\}), \\ i &\mapsto \{k \in \{1, \dots, Ne\} : D_k \text{ belongs to a node of finite element i}\}, \end{aligned}$$

where "D_k" denotes the kth component of the displacement vector as defined after (2.4). The mapping $\mathbf{node}(i)$ determines those nodes that belong to finite element i, **dof** assigns to a finite element i the set of all degrees of freedom that belong to the nodes of i, and $\mathbf{\Delta}$ maps a degree of freedom to the set of those finite elements it belongs to (sometimes we will also use $\mathbf{\Delta}$ defined on the set $\{1, \dots, Ng\}$, i.e., $\mathbf{\Delta}$ maps a grid point to the set of those finite elements it belongs to). Clearly one can determine **dof** from the knowledge of **node** and the set of the restricted degrees of freedom.

Furthermore, we will define the finite-dimensional space $\mathcal{V}_{0h}$ that should approximate the space $\mathcal{V}_0$ (cf. (1.2)) via

$$\mathcal{V}_{0h} := \operatorname{span}\{\psi^1, \dots, \psi^{Nf}\},$$

where the functions $\psi^1, \dots, \psi^{Nf} : \omega \subset \mathbb{R}^2 \to \mathbb{R}^2$ should be linearly independent and sufficiently smooth such that $\psi^k \in \mathcal{V}_0$, $k = 1, \dots, Nf$ and $\mathcal{V}_{0h} \subset \mathcal{V}_0$. (This can be realized by omitting all shape functions that correspond to grid points on the boundary γ_1.) Since the displacement field U is two-dimensional, the shape functions ψ^k are of the form

$$\psi^1(X) = \begin{pmatrix} \phi_1(X) \\ 0 \end{pmatrix}, \quad \psi^2(X) = \begin{pmatrix} 0 \\ \phi_1(X) \end{pmatrix}, \quad \dots, \quad \psi^{2Ng}(X) = \begin{pmatrix} 0 \\ \phi_{Ng}(X) \end{pmatrix},$$

where $\phi_i : \omega \subset \mathbb{R}^2 \to \mathbb{R}$ are the usual one-dimensional "hat functions" corresponding to one grid point, i.e., $\phi_i|_{\omega^{(j)}} \in \mathbb{P}_k$, $k \in \mathbb{N}, \forall j = 1, \dots, Ne$, where $\mathbb{P}_k$ denotes the set of all polynomials of degree lower than or equal to k and ϕ_i equals 1 at the grid point X_i and 0 at all other grid points (cf., e.g., [7, p. 129ff], [13, p. 560ff], or [24, p. 113ff]); for the meaning of the discretization parameter h (as, e.g., in "$\mathcal{V}_{0h}$") see Definition 4.1.

The generation of the finite element formulation of the equilibrium equations (1.1) and (1.6) works analogously to the linear case, i.e., $E = \text{const}$ (cf. [5, p. 287ff]), and is given by (we perform numerical integration with the center of gravity as the integration point and the measure of the domain of integration as the corresponding integration weight; for ease of notation use only one integration point)

$$\underline{\underline{K}}(\underline{\sigma}) \cdot \underline{D} = \underline{Q} \tag{2.2}$$

with the stiffness matrix

$$\begin{aligned} K_{ij}(\underline{\sigma}) = \sum_{k \in \mathbf{\Delta}(i) \cap \mathbf{\Delta}(j)} & meas(\omega^{(k)}) \cdot \frac{E(\underline{\sigma}^{(k)})}{1+\nu} \Bigg[\frac{\nu}{1-\nu} \operatorname{tr}(\varepsilon(\psi^j)) \operatorname{tr}(\varepsilon(\psi^i)) \\ & + \varepsilon(\psi^j) : \varepsilon(\psi^i) \Bigg] (X_*^{(k)}), \end{aligned} \tag{2.3}$$

the load vector

$$\begin{aligned} Q_i = \sum_{k \in \mathbf{\Delta}(i)} & meas(\omega^{(k)}) \cdot \left[\hat{F}(X_*^{(k)}) \cdot \psi^i(X_*^{(k)}) - \Delta T \cdot \beta : (\nabla \psi^i)(X_*^{(k)}) \right] \\ & + meas(\gamma_2^{(k)}) \cdot \left[G(Y_*^{(k)}) \cdot \psi^i(Y_*^{(k)}) \right], \end{aligned} \tag{2.4}$$

and the displacement vector $\underline{D} := [D_i]_{i=1}^{Nf}$, where each two successive components of $\underline{D}$ correspond to the first and the second degree of freedom of one grid point (unless it is restricted). In the above formulas, $X_*^{(k)}$ denotes the center of gravity of $\omega^{(k)}$ and

$Y_*^{(k)}$ is some integration point on the boundary $\gamma_2^{(k)} := \partial\omega^{(k)} \cap \gamma_2$, $k \in \{1, \dots, Ne\}$. Furthermore, the discrete stress vector $\underline{\sigma} := (\underline{\sigma}^{(1)}, \dots, \underline{\sigma}^{(Ne)})^T$ fulfills the equations

$$\underline{\sigma}^{(i)} = E(\underline{\sigma}^{(i)}) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}}^{(i)} \cdot \underline{D}^{(i)} + \Delta T \cdot \underline{\beta} \qquad \forall\ i = 1, \dots, Ne \tag{2.5}$$

with $\underline{\beta} = (\beta_{11}, \beta_{22}, \beta_{12})^T$, $\underline{D}^{(i)}$ as the local displacement vector for the ith finite element, i.e., $\underline{D}^{(i)} := [D_l]_{l \in \mathbf{dof}(i)}$, the local stress vectors

$$\underline{\sigma}^{(i)} := (\sigma_{11}^{(i)}(X_*^{(i)}), \sigma_{22}^{(i)}(X_*^{(i)}), \sigma_{12}^{(i)}(X_*^{(i)}))^T,$$

and the interpolation matrices

$$\underline{\underline{C}}^{(i)} = \begin{bmatrix} \phi_{l_1,1}(X_*^{(i)}) & 0 & \dots & \phi_{l_{n_i},1}(X_*^{(i)}) & 0 \\ 0 & \phi_{l_1,2}(X_*^{(i)}) & \dots & 0 & \phi_{l_{n_i},2}(X_*^{(i)}) \\ \frac{1}{2}\phi_{l_1,2}(X_*^{(i)}) & \frac{1}{2}\phi_{l_1,1}(X_*^{(i)}) & \dots & \frac{1}{2}\phi_{l,2}(X_*^{(i)}) & \frac{1}{2}\phi_{l_{n_i},1}(X_*^{(i)}) \end{bmatrix} \tag{2.6}$$

$\forall i = 1, \dots, Ne$ with $n_i := |\mathbf{node}(i)|$ (cf. [19, p. 64ff]).

Summing up the work of this section, we have derived the discrete equilibrium equations

$$\begin{aligned} \underline{\underline{K}}(\underline{\sigma}) \cdot \underline{D} &= \underline{Q}, \\ \underline{\sigma} &= E(\underline{\sigma}) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D} + \Delta T \cdot \underline{\beta}, \end{aligned} \tag{2.7}$$

where we have joined the Ne equations (2.5) to one equation for the discrete stress vector $\underline{\sigma} = (\underline{\sigma}^{(1)}, \dots, \underline{\sigma}^{(Ne)})^T \in \mathbb{R}^{3Ne}$ without introducing a new notation. (This should not give rise to any confusion.)

Notice that the global displacement vector $\underline{D}$ and the discrete stress vector $\underline{\sigma}$ in (2.7) depend on each other in the same way as in the weak equilibrium equations (1.1) and (1.6). Furthermore, the structure of equations (2.2) and (2.5) is quite similar to the one of (1.1) and (1.6), respectively, and therefore we will be able to apply the same techniques as in [6] to answer the question of existence and uniqueness of solutions of the discrete equilibrium equations.

3. Formulation of the discrete design problem.

In this section we want to define the two discrete design problems of interest that we will solve in section 8 for a practically relevant example. We will assume that the body to be optimized is defined as follows.

Let there be given a fixed domain $\omega \subset \mathbb{R}^2$ that is closed and bounded and the body

$$\Omega := \{(x, y, z) \in \mathbb{R}^3 :\ (x, y) \in \omega,\ z \in [-\theta(x, y), \theta(x, y)]\} \tag{3.1}$$

with a continuous *thickness function* $\theta : \omega \to \mathbb{R}^+$. We want to optimize the design of Ω as defined in (3.1), i.e., since ω is fixed, we intend to find an optimal thickness distribution θ. Since we have to transform everything on finite dimensions in order to be able to perform numerical computations, we will choose our design variables such that they define a finite-dimensional approximation of the continuous thickness distribution θ over ω. We assume again that there is given a finite element discretization

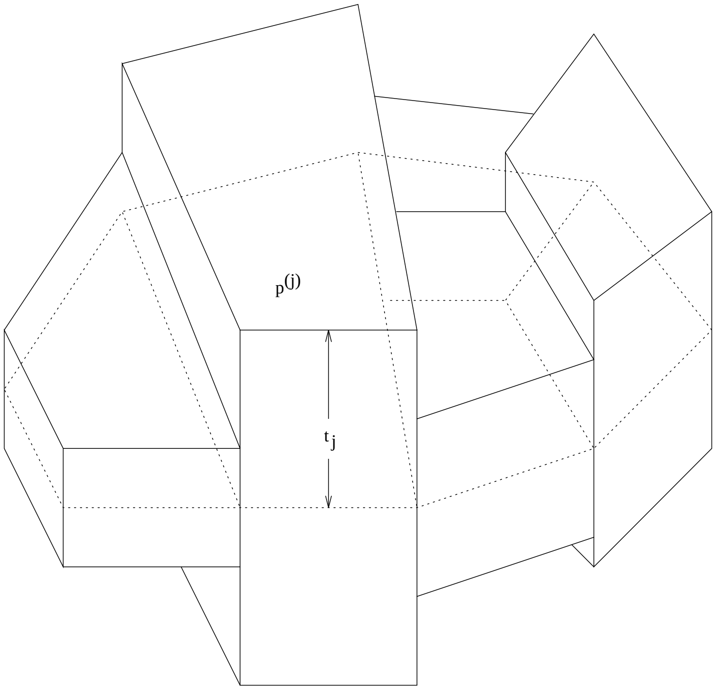


FIG. 3.1. *The design variables are the thicknesses t_j over the sets $p^{(j)}$.*

$\{\omega^{(i)}\}_{i=1,\dots,Ne}$ of $\omega \subset \mathbb{R}^2$ (cf. (2.1)). Furthermore, we suppose that there is given a partition $\{p^{(j)}\}_{j=1}^{Nd}$ of $\omega \subset \mathbb{R}^2$ with the properties

$$\begin{aligned} p^{(j)} &= \textstyle\bigcup_{i\in I_j} \omega^{(i)}, \qquad \emptyset \neq I_j \subset \{1,\dots,Ne\}, \\ \textstyle\bigcup_{j=1}^{Nd} p^{(j)} &= \omega, \\ \overset{\circ}{p}{}^{(j)} \cap \overset{\circ}{p}{}^{(k)} &= \emptyset \qquad \forall\, j \neq k, \end{aligned} \tag{3.2}$$

where $Nd \in \mathbb{N}$ denotes the number of design variables. We define the set $\mathcal{O}_{ad}$ of admissible bodies, respectively, designs, Ω via

$$\mathcal{O}_{ad} := \left\{ \Omega \subset \mathbb{R}^3 : \ \Omega = \bigcup_{j=1}^{Nd} p^{(j)} \times [-t_j, t_j],\ 0 < \underline{\tau}_j \le t_j \le \bar{\tau}_j \ \forall j = 1,\dots,Nd \right\}, \tag{3.3}$$

with some real numbers $\underline{\tau}_j, \bar{\tau}_j > 0$, $j = 1,\dots,Nd$, i.e., the set of admissible designs consists of all "skyline-like" bodies (cf. Figure 3.1) over the fixed partition $\{p^{(j)}\}_{j=1}^{Nd}$, where the corresponding heights t_j have to satisfy box constraints. Since the partition $\{p^{(j)}\}_{j=1}^{Nd}$ is fixed, the geometry of a body $\Omega \in \mathcal{O}_{ad}$ is completely defined through the *design vector* $\underline{t} := (t_1,\dots,t_{Nd})^T$. In the following we will note the dependence of an element Ω of $\mathcal{O}_{ad}$ on the design vector $\underline{t}$, sometimes explicitly by $\Omega(\underline{t})$.

REMARK 3.1. *We want to point out again that the definition* (3.3) *of admissible designs has to be understood as a finite-dimensional approximation of a continuous thickness distribution θ (cf.* (3.1)*). Clearly, no one would produce an object $\Omega \in \mathcal{O}_{ad}$ in reality since at the interface between two adjacent sets $p^{(j)}$ and $p^{(k)}$ of the partition* *of ω there occur singularities in the (infinite-dimensional) stress field (for the finite element formulation, no singularities occur).*

Therefore we will approximate the skyline-like surface of the final design in section 8 by some smoother surface, and we will show (by computation) that the resulting displacements and stresses do not differ too much from each other, respectively.

Before we state the design problems to solve over the set of admissible designs (3.3), we note some assumptions that should hold true throughout this paper.

ASSUMPTION 3.2. *Let there be given a three-dimensional object $\Omega \subset \mathbb{R}^3$ with applied body forces $\hat{F} : \Omega \to \mathbb{R}^3$ and surface forces $G : \Gamma_2 \to \mathbb{R}^3$, where $\Gamma_2 \subset \partial\Omega$. Performing sizing optimization, we assume that*

- *we have a plane stress problem, i.e., we consider the body Ω to be a plate that is thin in X_3-direction (compared to the other coordinate directions) and that can carry stresses only parallel to this plane; more precisely, we suppose that Ω can be defined as in* (3.1) *with a "small" thickness function θ and Γ_2 given by*

$$\Gamma_2 = \{(x,y,z) \in \mathbb{R}^3 : \ (x,y) \in \gamma_2, \ z \in [-\theta(x,y), \theta(x,y)]\}$$

  with some $\gamma_2 \subset \partial\omega$;
- *the applied surface tractions G and the body forces $\hat{F}$ are independent of X_3 (therefore there is no displacement and no strain in X_3-direction) and the other displacement components U_1 and U_2 are also independent of X_3.*

Actually the above assumptions have to be checked for each considered example: Can the object indeed be considered to be thin in X_3-direction? Are the occurring quantities really independent of X_3? These questions will be considered for our numerical examples in section 8, where we will perform fully three-dimensional computations to check our results.

From Assumption 3.2 we obtain that no variable depends on X_3, and therefore all integrals in (1.3) and (1.4) over a set $\Omega \in \mathcal{O}_{ad}$ can be written as

$$\int_\Omega [...] \; d(X_1,X_2,X_3) = \sum_{j=1}^{Nd} t_j \int_{p^{(j)}} [...] \; d(X_1,X_2) = \sum_{i=1}^{Ne} t^{(i)} \int_{\omega^{(i)}} [...] \; d(X_1,X_2),$$

where $t^{(i)}$ is the thickness over the finite element $\omega^{(i)}$. Thus the discrete formulation of the equilibrium equations for a body $\Omega \in \mathcal{O}_{ad}$ under Assumption 3.2 is given by (2.2) and (2.5), but the stiffness matrix $\underline{\underline{K}}$ and the load vector $\underline{Q}$ have to be redefined via

$$\begin{aligned} K_{ij}(\underline{t},\underline{\sigma}) := \sum_{k \in \mathbf{\Delta}(i) \cap \mathbf{\Delta}(j)} t^{(k)} \cdot meas(\omega^{(k)}) \cdot \frac{E(\underline{\sigma}^{(k)})}{1+\nu} \Bigg[\frac{\nu}{1-\nu} \mathrm{tr}(\varepsilon(\psi^j)) \mathrm{tr}(\varepsilon(\psi^i)) \\ + \ \varepsilon(\psi^j) : \varepsilon(\psi^i) \Bigg] (X_*^{(k)}), \end{aligned} \tag{3.4}$$

$$\begin{aligned} Q_i(\underline{t}) := \sum_{k \in \mathbf{\Delta}(i)} & t^{(k)} \cdot meas(\omega^{(k)}) \cdot \Big[\hat{F}(X_*^{(k)}) \cdot \psi^i(X_*^{(k)}) - \Delta T \cdot \beta : (\nabla \psi^i)(X_*^{(k)}) \Big] \\ & + t^{(k)} \cdot meas(\gamma_2^{(k)}) \cdot \Big[G(Y_*^{(k)}) \cdot \psi^i(Y_*^{(k)}) \Big], \end{aligned}$$

respectively (cf. (2.3) and (2.4)), where $\gamma_2^{(k)} = \gamma_2 \cap \partial\omega^{(k)}$. Moreover the bilinear form corresponding to (3.4) is given by

$$a_{\sigma,t}(U,V) := \sum_{i=1}^{Nd} t_i \cdot \int_{p^{(i)}} \frac{E(\sigma)}{1+\nu} \cdot \left[\frac{\nu}{1-\nu} \mathrm{tr}(\varepsilon(U))\mathrm{tr}(\varepsilon(V)) + \varepsilon(U) : \varepsilon(V) \right] \, dX \tag{3.5}$$

(cf. (1.3)).

Next we define the discrete design problems that we will solve in section 8 for a practically relevant problem. We want to stress again that we are only interested in numerical aspects and the development of algorithms to solve the optimization problem and we will not deal with the question of discretization errors (therefore cf. [19]).

The first design problem we want to solve is to minimize the volume of a given body under maximum stress constraints whose design can be approximated by elements Ω of the set $\mathcal{O}_{ad}$ (cf. Remark 3.1); more precisely,

$$\begin{aligned} Volume = \sum_{j=1}^{Nd} t_j \cdot meas(p^{(j)}) \longrightarrow \min_{\underline{t} \in \mathbb{R}^{Nd}}&, \\ \underline{\underline{K}}(\underline{t},\underline{\sigma}) \cdot \underline{D} = \underline{Q}(\underline{t})&, \\ \underline{\sigma} = E(\underline{\sigma}) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D} + \Delta T \cdot \underline{\beta}&, \\ \sigma_{elm}^{(i)} \le \bar{\sigma} & \qquad \forall\, i = 1, \ldots, Me, \\ \underline{\delta}_i \le D_i \le \bar{\delta}_i & \qquad \forall\, i = 1, \ldots, Mf, \\ \underline{\tau}_i \le t_i \le \bar{\tau}_i & \qquad \forall\, i = 1, \ldots, Md, \end{aligned} \tag{3.6}$$

with $1 \le Me \le Ne$, $1 \le Mf \le Nf$, $1 \le Md \le Nd$ integer numbers, $\sigma^*, \bar{\sigma} > 0$, $\underline{\delta}_i, \bar{\delta}_i \in \mathbb{R}$, $i = 1, \ldots, Mf$, $\underline{\tau}_i, \bar{\tau}_i > 0$, $i = 1, \ldots, Md$, and the element stress $\sigma_{elm}^{(i)}$ of the ith finite element $\omega^{(i)}$, $i \in \{1, \ldots, Ne\}$, given by

$$\sigma_{elm}^{(i)} := \frac{1}{meas(\omega^{(i)})} \int_{\omega^{(i)}} \sigma_{VON}(\sigma(X)) \, dX \approx \sigma_{VON}(\underline{\sigma}^{(i)}), \tag{3.7}$$

where the von Mises stress σ_{VON} is defined by

$$\sigma_{VON}(\underline{\sigma}) := \sqrt{\sigma_{11}^2 + \sigma_{22}^2 + 3\sigma_{12}^2 - \sigma_{11}\sigma_{22}}$$

$\forall \underline{\sigma} = (\sigma_{11}, \sigma_{22}, \sigma_{12})^T \in \mathbb{R}^3$ (cf. [12, p. 255]); the introduction of the limits Me, Mf, Md instead of their upper bounds Ne, Nf, Nd allows for a more general notation of the design problem. Notice that t_i denotes the ith component of the design vector and $t^{(i)}$ denotes the thickness of the ith finite element.

The second problem of interest is to find the optimal design $\Omega \in \mathcal{O}_{ad}$ (approximating a smooth three-dimensional body; cf. Remark 3.1) in order to reach an even stress distribution over the cross section ω; more precisely,

$$\begin{aligned} &Even\ Distribution = \sum_{j=1}^{Ne}\left(\frac{\sigma_{elm}^{(j)}}{\sigma_*} - 1\right)^2 \longrightarrow \ \min, \\ &\underline{\underline{K}}(\underline{t},\underline{\sigma})\cdot\underline{D} = \underline{Q}(\underline{t}), \\ &\underline{\sigma} = E(\underline{\sigma})\cdot\underline{\underline{\tilde{G}}}\cdot\underline{\underline{C}}\cdot\underline{D} + \Delta T\cdot\underline{\beta}, \\ &\sigma_{elm}^{(i)} \le \bar{\sigma} && \forall\ i = 1,\dots,Me, \\ &\underline{\delta}_i \le D_i \le \bar{\delta}_i && \forall\ i = 1,\dots,Mf, \\ &\underline{\tau}_i \le t_i \le \bar{\tau}_i && \forall\ i = 1,\dots,Md. \end{aligned} \tag{3.8}$$

4. Existence and uniqueness of solutions. We have stated in sections 2 and 3, respectively, the discrete equilibrium equations and the discrete design problem(s) we want to solve. In this section we will (briefly) deal with the question of existence and uniqueness of these problems before we develop and analyze algorithms for solving both problems.

4.1. Discrete equilibrium equations. Following the lines of [6], one can see that the same technique used there to prove the existence and uniqueness of solutions of the weak equilibrium equations (1.1) and (1.6) can be applied to (2.2) and (2.5), too. Since it is very space consuming to write down all the details and we want to concentrate on numerical aspects, we will state only the main result and some important facts (for the full details see [19, p. 68ff]). Throughout this section we assume that the parameters t_j, $j = 1,\dots,Nd$, are fixed (cf. (3.3)).

First let us define the sets of suitable stress fields, respectively, displacement fields,

$$\underline{S}(S) := \{\underline{\sigma} \in \mathbb{R}^{3Ne} :\ \|\underline{\sigma}^{(i)}\|_3 \le S\ \forall i = 1,\dots,Ne\}, \tag{4.1}$$

$$\underline{B}(R) := \{\underline{D} \in \mathbb{R}^{Nf} :\ \|\underline{D}\|_{Nf} \le R\} \tag{4.2}$$

$\forall R, S > 0$, where $\|.\|_n$ is the Euclidean norm in the space $\mathbb{R}^n$.

In order to prove several estimates, it is necessary that the triangulation of the two-dimensional cross section ω is a special kind, as shown in the following definition.

DEFINITION 4.1. *A (fixed) triangulation $\{\omega^{(i)}\}_{i=1}^{Ne}$ of $\omega \subset \mathbb{R}^2$ with discretization parameter h is said to be regular iff there exist constants $\alpha_0 > 0$ and $\varphi_0 \in (0, \frac{\pi}{2})$ such that*

$$\begin{aligned} 0 < \alpha_0\cdot h \le h_{(i),A}, h_{(i),B}, h_{(i),C} \le h, \\ 0 < \varphi_0 \le \theta_{(i),A}, \theta_{(i),B}, \theta_{(i),C} \le \pi - \varphi_0 \end{aligned}$$

$\forall i = 1,\dots,Ne$, where $h_{(i),A}$, $h_{(i),B}$, $h_{(i),C}$ denote the lengths of the three edges and $\theta_{(i),A}$, $\theta_{(i),B}$, $\theta_{(i),C}$ denote the inner angles of the ith finite element.

Moreover, we introduce the following parameters:

$\lambda = (1-\nu)^{-1}$	...	maximum eigenvalue of the matrix $\underline{\underline{\tilde{G}}}$,
c_1	...	maximum number of elements connected to a grid point,
c_2	...	maximum number of grid points of one finite element,
c_{Korn}	...	constant of Korn's inequality (cf., e.g., [22, p. 16ff]),
$\lambda_{MIN}(G_0)$	...	minimal eigenvalue of the element mass matrix $\underline{\underline{G}}_0$ with

$$\underline{\underline{G}}_0 := [G_{ij}]_{i,j=1}^6 := \left[\int_\Delta \phi_i(X)\cdot\phi_j(X)\ dX\right]_{i,j=1}^6, \tag{4.3}$$

where Δ is the master triangle with the corners $(0,0)$, $(1,0)$, $(0,1)$ (cf. [19, p. 138ff]). Furthermore, we define $t_{MIN} := \min_{i=1}^{Nd} |t_i|$ and $t_{MAX} := \max_{i=1}^{Nd} |t_i|$ (cf. (3.3)). Then we have the following theorem.

THEOREM 4.2. *Let the material law function E be Lipschitz continuous with constant $L_E > 0$ such that $E(0) > 0$, and let $\underline{t} \in \mathbb{R}^{Nd}$ be fixed. Then there exists a radius $S > 0$ such that*

$$0 < \underline{e} \le E(\underline{\sigma}) \le \bar{e} \qquad \forall\ \underline{\sigma} \in \underline{\mathcal{S}}(S)$$

with $\underline{\mathcal{S}}(S)$ as defined in (4.1)*. Moreover, there exists a radius $R > 0$ such that there exists a unique solution $\underline{\sigma}$ of* (2.5) *that lies in the set $\underline{\mathcal{S}}(S)$ $\forall \underline{D} \in \underline{\mathcal{B}}(R)$ with $\underline{\mathcal{B}}(R)$ as defined in* (4.2)*.*

Furthermore, let the load vector $\underline{Q}$ lie in the ball $B(0,r) := \{v \in \mathbb{R}^{Nf} : \|v\|_{Nf} < r\}$ with sufficiently small radius $r > 0$ such that $\underline{\underline{K}}^{-1}(\underline{t},\underline{\sigma})\underline{Q} \in \underline{\mathcal{B}}(R)$ $\forall \underline{\sigma} \in \underline{\mathcal{S}}(S)$, and let the triangulation of ω be regular.

Then there exists a solution $(\underline{D}^*, \underline{\sigma}^*) \in \underline{\mathcal{B}}(R) \times \underline{\mathcal{S}}(S)$ of the discrete equilibrium equations (2.2) *and* (2.5)*.*

Moreover, if the parameters L_E, R, ν, c_2, α_0, φ_0, h, $\bar{e}$, c_{Korn}, $\underline{e}$, $\lambda_{MIN}(G_0)$, c_1, t_{MIN}, t_{MAX} fulfill the inequality

$$L_E \cdot R \cdot \lambda \cdot c' \cdot h^{-1} \cdot \left(\frac{\bar{e} \cdot (1+\nu)}{c_{Korn}^2 \underline{e} \cdot (1-\nu)} \cdot c'' \cdot h^{-2} + 1 \right) < 1 \tag{4.4}$$

with

$$c' := \frac{2\sqrt{10 c_2}}{\alpha_0^2 \sin\varphi_0} \qquad \text{and} \qquad c'' := \frac{64 \lambda_{MIN}(G_0) c_1 c_2^2}{\alpha_0^2 \sin(\varphi_0)} \cdot \frac{t_{MAX}}{t_{MIN}},$$

then the solution $(\underline{D}^*, \underline{\sigma}^*) \in \underline{\mathcal{B}}(R) \times \underline{\mathcal{S}}(S)$ of the discrete equilibrium equations (2.2) *and* (2.5) *is unique.*

Proof. (This is only a sketch; for details see [19, p. 76ff]). Everything works analogously to [6]: Under the assumptions $E(0) > 0$ and $E \in C^{0,1}(\mathbb{R}^3)$ there exists a radius $S > 0$ such that Young's modulus E is positive uniformly over the set $\underline{\mathcal{S}}(S)$. Furthermore, for all (fixed) $\underline{\sigma} \in \underline{\mathcal{S}}(S)$ and for a sufficiently small load vector $\underline{Q}$ there exists a unique solution $\underline{D} \in \underline{\mathcal{B}}(R)$ to (2.2). Moreover, for all (fixed) $\underline{D} \in \underline{\mathcal{B}}(R)$ there exists a unique solution $\underline{\sigma} \in \underline{\mathcal{S}}(S)$ to (2.5). We define the well-defined operator

$$\begin{aligned} \mathcal{M} : (\underline{\mathcal{S}}(S), \|.\|_{3Ne}) &\to (\underline{\mathcal{S}}(S), \|.\|_{3Ne}), \\ \underline{\sigma} &\mapsto \mathcal{M}(\underline{\sigma}) := E(\underline{\sigma}) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{\underline{K}}^{-1}(\underline{t},\underline{\sigma})\underline{Q} + \Delta T \cdot \underline{\beta}. \end{aligned}$$

Under the stated assumptions it is indeed possible to prove analogously to [6] that $\mathcal{M}(\underline{\sigma}) \in \underline{\mathcal{S}}(S)$ $\forall \underline{\sigma} \in \underline{\mathcal{S}}(S)$ and that $\mathcal{M}$ is Lipschitz continuous with constant

$$L_h := L_E R \lambda \cdot \frac{2\sqrt{10 c_2}}{\alpha_0^2 \sin\varphi_0 h} \cdot \left(\frac{\bar{e} \cdot (1+\nu)}{c_{Korn}^2 \underline{e} \cdot (1-\nu)} \cdot \frac{64 \lambda_{MIN}(G_0) c_1 c_2^2}{\alpha_0^2 \sin(\varphi_0) h^2} \cdot \frac{t_{MAX}}{t_{MIN}} + 1 \right). \tag{4.5}$$

From (4.4) it follows that $L_h < 1$, and from Banach's fixed point theorem it follows that the fixed point $\underline{\sigma} \in \underline{\mathcal{S}}(S)$, and therefore the solution of the discrete equilibrium equations is unique. □

The difference between the proof of Theorem 4.2 and the proof of the existence and uniqueness of solutions of the weak equilibrium equations (1.1) and (1.6) in [6] is that in several estimates the discretization parameter h occurs, e.g., the Frobenius norm of the matrix $\underline{\underline{C}}^{(i)}$ can be estimated by

$$\|\underline{\underline{C}}^{(i)}\|_F := \sqrt{C_{kl}^{(i)} \cdot C_{kl}^{(i)}} \leq \frac{2 \cdot \sqrt{10 \cdot c_2}}{\alpha_0^2 \sin(\varphi_0) h}$$

(cf. [19, p. 72]) or the norm of the inverse of the stiffness matrix $\underline{\underline{K}}(\underline{t}, \underline{\sigma})$ can be estimated by

$$\|\underline{\underline{K}}^{-1}(\underline{t}, \underline{\sigma})\| \leq \frac{2 \cdot (1+\nu)}{\alpha_0^2 \sin(\varphi_0) \lambda_{MIN}(G_0) c_{Korn}^2 \underline{e} h^2 t_{MIN}}$$

(cf. [19, p. 73]), which yields the two multipliers "$c' \cdot h^{-1}$" and "$c'' \cdot h^{-2}$." In fact, comparing the Lipschitz constant of the fixed point operator in the infinite-dimensional case (cf. [6])

$$L = \frac{L_E \cdot R_{critical}}{1-\nu} \cdot \left(\frac{\bar{e} \cdot (1+\nu)}{c_{Korn}^2 \underline{e} \cdot (1-\nu)} + 1 \right) \tag{4.6}$$

with the Lipschitz constant L_h in (4.5), one can see that they are equal apart from these additional multipliers.

One would expect that the Lipschitz constants L_h for the discrete operator $\mathcal{M}$ converge to the Lipschitz constant L of the continuous operator as h tends to zero. Since $\lim_{h \to 0} L_h = \infty$ we must conclude that our method to estimate the Lipschitz constant L_h is not optimal (cf. also [7, p. 315], where it is mentioned that in the Lipschitz constant for the semidiscrete problem of parabolic partial differential equations negative powers of h appear, too).

4.2. Discrete design problem. In [9] the existence of solutions of shape design problems is studied extensively; notice that sizing problems are special cases of shape design problems. In the following we will generalize an existence result for state-constrained problems for linear elastic materials to our nonlinear material law $E = E(\sigma)$ (cf. [9, p. 221ff]).

If we assume that the bilinear form $a_{\sigma,t}(.,.)$ as defined in (3.5) is uniformly elliptic over the set $\underline{\mathcal{S}}(S)$ (cf. (4.1)) and the set of all admissible designs, one can extend the existence theory in [9] for linear problems to our special class of nonlinear problems without too much effort.

THEOREM 4.3. *Let*

1. X, Y, Z *be Banach spaces,*
2. $\mathcal{T} \subset X$, $\mathcal{K} \subset Y$, $\mathcal{R} \subset Z$ *be convex, closed, and nonempty sets,*
3. $A(\underline{t}, \underline{\sigma}) : Y \to Y'$ *be generated by a symmetric, uniformly elliptic, and uniformly continuous bilinear form* $a_{\underline{t},\underline{\sigma}}$*, i.e.,*

$$\langle A(\underline{t}, \underline{\sigma})\underline{u}, \underline{v}\rangle \ := \ a_{\underline{t},\underline{\sigma}}(\underline{u}, \underline{v}) \qquad \forall\ \underline{u}, \underline{v} \in Y,$$

where

$$\begin{array}{rcll} a_{\underline{t},\underline{\sigma}}(\underline{u}, \underline{v}) & = & a_{\underline{t},\underline{\sigma}}(\underline{v}, \underline{u}) & \forall\ \underline{u}, \underline{v} \in Y, \\ a_{\underline{t},\underline{\sigma}}(\underline{v}, \underline{v}) & \geq & \mu_1 \cdot \|\underline{v}\|_Y^2 & \forall\ \underline{v} \in Y\ \forall \underline{\sigma} \in \mathcal{R}\ \forall \underline{t} \in \mathcal{T}, \\ |a_{\underline{t},\underline{\sigma}}(\underline{u}, \underline{v})| & \leq & \mu_2 \cdot \|\underline{u}\|_Y \cdot \|\underline{v}\|_Y & \forall\ \underline{u}, \underline{v} \in Y\ \forall \underline{\sigma} \in \mathcal{R}\ \forall \underline{t} \in \mathcal{T} \end{array}$$

with some positive constants μ_1 *and* μ_2*, and let* $f \in Y'$ *be linear and continuous,*

4. $(\underline{t}_n, \underline{\sigma}_n) \to (\underline{t}, \underline{\sigma})$ *in* $X \times Z \Longrightarrow A(\underline{t}_n, \underline{\sigma}_n) \to A(\underline{t}, \underline{\sigma})$ *in* $L(Y, Y')$,
5. *there exist a unique solution* $(\underline{u}^*(\underline{t}), \underline{\sigma}^*(\underline{t})) \in Y \times Z \ \forall \underline{t} \in \mathcal{T}$ *(cf.* [6]*),*
6. $s : \mathcal{K} \to \mathcal{R}$ *be continuous,*
7. $J : X \times Y \times Z \to \mathbb{R}$ *be convex and lower semicontinuous,*
8. $\mathcal{T} \subset X$ *and* $\mathcal{R} \subset Z$ *be compact.*

Then the problem

$$(\mathcal{P}) \qquad \left\{ \begin{array}{rcl} J(\underline{t}, \underline{u}, \underline{\sigma}) & \to & \min\limits_{\underline{t} \in \mathcal{T}}, \\ A(\underline{t}, \underline{\sigma})\underline{u} & = & f, \\ \underline{\sigma} & = & s(\underline{u}), \\ (\underline{u}, \underline{\sigma}) & \in & \mathcal{K} \times \mathcal{R}, \\ \underline{t} & \in & \mathcal{T} \end{array} \right.$$

has at least one optimal solution $(\underline{t}^*, \underline{u}^*, \underline{\sigma}^*)$ *if problem* $(\mathcal{P})$ *has admissible pairs (i.e., if the set of all* $\underline{t} \in \mathcal{T}$ *such that* $(\underline{u}(\underline{t}), \underline{\sigma}(\underline{t})) \in \mathcal{K} \times \mathcal{R}$ *is nonempty).*

Proof. See [19, p. 80ff]. □

REMARK 4.4. *Before we apply Theorem* 4.3 *to the design problems* (3.6) *and* (3.8)*, we want to state that the theorem also remains true for design problems formulated in terms of the element stress vector* $\underline{\sigma}_{elm} := (\sigma_{elm}^{(1)}, \dots, \sigma_{elm}^{(Ne)})^T$ *instead of the stress vector* $\underline{\sigma}$*, since this is only a reformulation of the problem.*

Then we have the following theorem.

THEOREM 4.5. *We define the set of admissible design vectors via*

$$\mathcal{T} := \{\underline{t} \in X := \mathbb{R}^{Nd} : \ \underline{\tau}_i \le t_i \le \bar{\tau}_i, i = 1, \dots, Md\}. \tag{4.7}$$

Suppose that all assumptions of Theorem 4.2 *are true, but with* $\tau_{MAX} := \max_{i=1}^{Nd} |\bar{\tau}_i|$ *and* $\tau_{MIN} := \min_{i=1}^{Nd} |\underline{\tau}_i|$ *instead of* t_{MAX} *and* t_{MIN}*, respectively. If there exists a vector* $\underline{t} \in \mathcal{T}$ *such that the unique solution* $(\underline{D}(\underline{t}), \underline{\sigma}(\underline{t})) \in \mathbb{R}^{Nf} \times \mathbb{R}^{3Ne}$ *of the discrete equilibrium equations* (2.2) *and* (2.5) *fulfills the constraints for problem* (3.6)*, respectively,* (3.8)*, then there exists an optimal solution* $\underline{t}^*$*.*

Proof. (This includes only the most important steps; for details cf. [19]). We will apply Theorem 4.3 to the two design problems (3.6) and (3.8) using Remark 4.4. Therefore we have to verify assumptions 1–8:

Let $X := \mathbb{R}^{Nd}$, $Y := \mathbb{R}^{Nf}$, and $Z := \mathbb{R}^{Ne}$. Then assumption 1 is clearly fulfilled. We define

$$\begin{aligned} \mathcal{K} &:= \{\underline{u} \in Y : \underline{u} \in \underline{\mathcal{B}}(R) \text{ and } \underline{u}_i \le D_i \le \bar{u}_i, i = 1, \dots, Mf\}, \\ \mathcal{R} &:= \{\underline{\sigma}_{elm} = \underline{\sigma}_{elm}(\underline{\sigma}) \in Z : \underline{\sigma} \in \underline{\mathcal{S}}(S) \text{ and } (\underline{\sigma}_{elm})_i \le \bar{\sigma}, i = 1, \dots, Me\} \end{aligned}$$

with $\underline{\mathcal{S}}(S)$ and $\underline{\mathcal{B}}(R)$ defined as in (4.1) and (4.2), respectively. It is easy to see that $\mathcal{T}$, $\mathcal{K}$, and $\mathcal{R}$ are closed and convex sets. Furthermore, $\mathcal{T}$ and $\mathcal{R}$ are bounded and are therefore compact since X and Z are finite-dimensional spaces (cf. [10, p. 33]).

For the design problems (3.6) and (3.8) the operator A is exactly the stiffness matrix $\underline{\underline{K}}$ defined as in (3.4). From subsection 4.1 and [6] it follows that the functional f and the matrix $\underline{\underline{K}}(\underline{t}, \underline{\sigma})$ indeed have the required properties of assumption 3, where the constants μ_1 and μ_2 do not depend on the design $\underline{t}$, since the design vectors $\underline{t}$ in the set $\mathcal{T}$ are uniformly bounded from below and above. This is also the reason why one can formulate the conditions for existence and uniqueness of [6] uniformly over $\mathcal{T}$. The continuity assumption 4 can be seen as follows. Let $(\underline{t}_n, \underline{\sigma}_n) \to (\underline{t}, \underline{\sigma})$ in $X \times Z$. Then

$$\|\underline{\underline{K}}(\underline{t}_n, \underline{\sigma}_n) - \underline{\underline{K}}(\underline{t}, \underline{\sigma})\| \le \|\underline{\underline{K}}(\underline{t}_n, \underline{\sigma}_n) - \underline{\underline{K}}(\underline{t}_n, \underline{\sigma})\| + \|\underline{\underline{K}}(\underline{t}_n, \underline{\sigma}) - \underline{\underline{K}}(\underline{t}, \underline{\sigma})\|. \tag{4.8}$$

Therefore the first term in (4.8) goes to zero as n tends to infinity since the sequence $\{t_n\}_{n\in\mathbb{N}}$ is uniformly bounded over $\mathcal{T}$ and the mapping $\underline{\sigma} \in \mathbb{R}^3 \mapsto E(\underline{\sigma}) \in \mathbb{R}$ is Lipschitz continuous. Furthermore, the second term goes to zero because $\underline{\underline{K}}$ is linear with respect to $\underline{t}$, i.e., it is also continuous over $\mathcal{T}$ (cf. (3.4)).

Under the assumptions of Theorem 4.2 the continuity of the operator s can be shown as in [6] for the infinite-dimensional case (for details see [19, p. 69ff]). Moreover, the cost functionals J of problems (3.6) and (3.8) are indeed convex and lower semicontinuous.

Finally we may conclude from Theorem 4.3 that for the design problem (3.6) as well as for the design problem (3.8) there exists a solution $\underline{t}_*$. □

For problem (3.8) the cost functional $J = J(\underline{\sigma}_{elm})$ is strictly convex with respect to $\underline{\sigma}_{elm}$, but the mapping

$$\underline{\sigma} = (\sigma_{11}, \sigma_{22}, \sigma_{12})^T \in \mathbb{R}^3 \mapsto \underline{\sigma}_{elm}(\underline{\sigma}) = \sqrt{\sigma_{11}^2 + \sigma_{22}^2 + 3\sigma_{12}^2 - \sigma_{11}\sigma_{22}}$$

is not injective (it is constant on ellipsoids). Therefore the solution of problem (3.8) need not be unique. For the first design problem (3.6), J is linear, i.e., this solution need not be unique either.

5. Algorithms for solving the discrete equilibrium equations. In this section we will use the analysis of [6] and section 4 to develop algorithms to solve (2.2) and (2.5), which will be used in section 8 to solve numerical examples. We will also present convergence results and discuss difficulties that (may) occur. Throughout this section we will omit the dependence of several terms on the design vector $\underline{t}$.

5.1. Fixed point iteration. We have shown in section 4 that (2.2) and (2.5) have a unique solution under certain restrictions on the applied loads and the material law. Furthermore, we have seen that the operator

$$\begin{aligned} &\mathcal{M} : \underline{\mathcal{S}}(S) \to \underline{\mathcal{S}}(S), \\ &\qquad \underline{\sigma} \mapsto \mathcal{M}(\underline{\sigma}) := E(\underline{\sigma}) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot (\underline{\underline{K}}^{-1}(\underline{\sigma})\underline{Q}) + \Delta T \cdot \underline{\beta} \end{aligned}$$

is Lipschitz continuous with the constant L_h as given in (4.5). It follows from Banach's fixed point theorem that if L_h is smaller than one, the fixed point iteration $\underline{\sigma}^{(n+1)} := \mathcal{M}(\underline{\sigma}^{(n)})$ converges to the unique fixed point $\underline{\sigma}^* \in \underline{\mathcal{S}}(S)$ of the operator $\mathcal{M}$ for all starting points $\underline{\sigma}^{(0)} \in \underline{\mathcal{S}}(S)$, and $(\underline{D}^* := \underline{\underline{K}}^{-1}(\underline{\sigma}^*)\underline{Q}, \underline{\sigma}^*)$ is then the unique solution of the discrete equilibrium equations (2.2) and (2.5).

Therefore we define the fixed point iteration. (The superscript for the stress vector denotes the iteration number, not a finite element index.)

Algorithm 5.1.

$$\begin{aligned} &\underline{\underline{K}}(\underline{\sigma}^{(n)})\underline{D}^{(n+1)} = \underline{Q}, \\ &\qquad \underline{\sigma}^{(n+1)} = E(\underline{\sigma}^{(n)}) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D}^{(n+1)} + \Delta T \cdot \underline{\beta} \end{aligned}$$

with an arbitrary starting point of $\underline{\sigma}^{(0)} \in \underline{\mathcal{S}}(S)$.

From Theorem 4.2 and Banach's fixed point theorem, the next theorem follows immediately.

Theorem 5.2. *Let all assumptions of Theorem* 4.2 *hold, especially the condition* (4.4) *that implies that* $L_h < 1$ *with* L_h *as defined in* (4.5). *Then Algorithm* 5.1 *converges for an arbitrary starting point* $\underline{\sigma}^{(0)} \in \underline{\mathcal{S}}(S)$. *Furthermore, the estimate*

$$\|\underline{\sigma}^{(n)} - \underline{\sigma}^{(0)}\| \leq \frac{L_h^n}{1 - L_h} \cdot \|\underline{\sigma}^{(1)} - \underline{\sigma}^{(0)}\|$$

is true for all $n \in \mathbb{N}$.

REMARK 5.3. *In fact, the bound R on the displacement vector $\underline{D}$, respectively, the applied loads, and L_E have to be quite small in order to make the Lipschitz constant L_h of the operator $\mathcal{M}$ smaller than one (cf.* (4.5)*). Therefore some difficulties have occurred in finding suitable input parameters for the numerical examples to yield convergence of Algorithm* 5.1. *(We will return to that point in subsection* 5.4.*)*

Next we will investigate another algorithm that is usually the first approach toward solving a nonlinear equation.

5.2. Newton's method. If we assume that the material law function E is continuously differentiable, then we can use Newton's method in order to solve the discrete equilibrium equations (2.2) and (2.5). Therefore we define the function

$$F(\underline{D},\underline{\sigma}) := \begin{pmatrix} \underline{\underline{K}}(\underline{\sigma}) \cdot \underline{D} - \underline{Q} \\ \underline{\sigma} - E(\underline{\sigma}) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D} - \Delta T \cdot \underline{\beta} \end{pmatrix}. \tag{5.1}$$

Then (2.2) and (2.5) read as "$F(\underline{D},\underline{\sigma}) = 0$." Computing the gradient of F with respect to $\underline{D}$ and $\underline{\sigma}$ yields

$$\nabla_{(\underline{D},\underline{\sigma})} F(\underline{D},\underline{\sigma}) := \left[\begin{array}{ccc} \underline{\underline{K}}(\underline{\sigma}) & \vdots & \dfrac{\partial}{\partial \underline{\sigma}} \left(\underline{\underline{K}}(\underline{\sigma}) \cdot \underline{D} \right) \\ \cdots\cdots & & \cdots\cdots\cdots \\ -E(\underline{\sigma}) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} & \vdots & \underline{\underline{I}} - \dfrac{\partial}{\partial \underline{\sigma}} (E(\underline{\sigma}) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D}) \end{array} \right], \tag{5.2}$$

and one Newton step is given by

$$\nabla_{(\underline{D},\underline{\sigma})} F(\underline{D}^{(j)},\underline{\sigma}^{(j)}) \cdot \begin{pmatrix} \Delta \underline{D}^{(j)} \\ \Delta \underline{\sigma}^{(j)} \end{pmatrix} = -F(\underline{D}^{(j)},\underline{\sigma}^{(j)}),$$
$$(\underline{D}^{(j+1)},\underline{\sigma}^{(j+1)})^T = (\underline{D}^{(j)},\underline{\sigma}^{(j)})^T + (\Delta\underline{D}^{(j)},\Delta\underline{\sigma}^{(j)})^T.$$

Multiplying out, computing $\Delta\underline{D}^{(j)}$ explicitly from the first equation, and inserting the resulting formula into the second equation results in the following algorithm.

ALGORITHM 5.4.

$$\left\{ \begin{array}{l} \Delta \underline{D}^{(j)} = \underline{\underline{K}}^{-1} \left[\underline{Q} - \underline{\underline{K}} \cdot \underline{D}^{(j)} - \frac{\partial}{\partial \underline{\sigma}} (\underline{\underline{K}} \cdot \underline{D}^{(j)}) \Delta \underline{\sigma}^{(j)} \right], \\ \left[\underline{\underline{I}} - \frac{\partial}{\partial \underline{\sigma}} (E(\underline{\sigma}^{(j)}) \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D}^{(j)}) + E(\underline{\sigma}^{(j)}) \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{\underline{K}}^{-1} \frac{\partial}{\partial \underline{\sigma}} (\underline{\underline{K}} \cdot \underline{D}^{(j)}) \right] \Delta \underline{\sigma}^{(j)} \\ \qquad = -\underline{\sigma}^{(j)} + E(\underline{\sigma}^{(j)}) \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D}^{(j)} + \Delta T \underline{\beta} + E(\underline{\sigma}^{(j)}) \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{\underline{K}}^{-1} (\underline{Q} - \underline{\underline{K}} \cdot \underline{D}^{(j)}) \end{array} \right.$$

with some appropriate starting point $(\underline{D}^{(0)},\underline{\sigma}^{(0)})^T$. (Notice that the stiffness matrix $\underline{\underline{K}} = \underline{\underline{K}}(\underline{\sigma}^{(j)})$ depends on j, too.)

From [17, p. 148] we get the following (local) convergence result of Newton's method.

LEMMA 5.5. *Let X, Y be normed vector spaces, $D \subset X$ be open, and $F : D \to Y$ be differentiable in D. Furthermore, let $x^* \in D$ be such that $F(x^*) = 0$, $F'(x^*)$ is regular, and F' is continuous in x^*. Then Newton's method*

$$x^{k+1} = x^k - \left[F'(x^k) \right]^{-1} \cdot F(x^k)$$

is locally q-superlinearly convergent.

Then a convergence result for Algorithm 5.4 is given by the following theorem.

THEOREM 5.6. *Let all assumptions of Theorem* 4.2 *hold, and let the function E be in the space $C^1(\mathbb{R}^3)$. Then Newton's method converges q-superlinearly for sufficiently small R and L_E in a neighborhood of the solution $(\underline{D}^*, \underline{\sigma}^*)$ of the discrete equilibrium equations* (2.2) *and* (2.5), *more precisely, if*

$$L_E \cdot R < \left[c \cdot h^{-5} \cdot \frac{\bar{e}}{(1-\nu)^2} \right]^{-1}, \tag{5.3}$$

where the constant $c > 0$ depends only upon the triangulation.

Proof. (This is only a sketch; for more details see [19, p. 90ff]). We will check the assumptions of Lemma 5.5 in order to prove the (local) convergence of Algorithm 5.4.

- F is continuously differentiable: From the fact that E is continuously differentiable over $\mathbb{R}$ 3 and (2.3) we may conclude that the derivatives

  $$\frac{\partial}{\partial \underline{\sigma}} \left(\underline{\underline{K}}(\underline{\sigma}) \cdot \underline{D} \right) \text{ and } \frac{\partial}{\partial \underline{\sigma}} \left(E(\underline{\sigma}) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D} \right)$$

  exist and are continuous.
- $F(x^*) = 0$: Since we have shown in Theorem 4.2 that there exists a unique solution $(\underline{D}^*, \underline{\sigma}^*)$ to the discrete equilibrium equations, we obtain from the definition (5.1) of F that $x^* := (\underline{D}^*, \underline{\sigma}^*)$ is a root of F.
- Regularity of $F'(x^*)$: Assume that there exists a vector $\underline{c} := (\alpha, \beta)$ such that $F'(x^*) \cdot \underline{c} = 0$. Then it follows from (5.2) that

  $$\begin{aligned} \underline{\underline{K}}(\underline{\sigma}^*) \cdot \alpha + \frac{\partial}{\partial \underline{\sigma}} \left(\underline{\underline{K}}(\underline{\sigma}^*) \cdot \underline{D}^* \right) \cdot \beta &= 0, \\ -E(\underline{\sigma}^*) \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \alpha + \left[\underline{\underline{I}} - \frac{\partial}{\partial \underline{\sigma}} (E(\underline{\sigma}^*) \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D}^*) \right] \cdot \beta &= 0. \end{aligned}$$

  Computing α explicitly from the first equation and inserting the resulting formula into the second equation yields

  $$\alpha = -\underline{\underline{K}}^{-1}(\underline{\sigma}^*) \cdot \tfrac{\partial}{\partial \underline{\sigma}} \left(\underline{\underline{K}}(\underline{\sigma}^*) \cdot \underline{D}^* \right) \cdot \beta,$$

  $$\left[\underline{\underline{I}} - \left(\tfrac{\partial}{\partial \underline{\sigma}} (E(\underline{\sigma}^*) \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D}^*) - E(\underline{\sigma}^*) \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{\underline{K}}^{-1} \cdot \tfrac{\partial}{\partial \underline{\sigma}} \left(\underline{\underline{K}} \cdot \underline{D}^* \right) \right) \right] \cdot \beta = 0. \tag{5.4}$$

  We define the matrix

  $$\underline{\underline{T}} := \frac{\partial}{\partial \underline{\sigma}} \left(E(\underline{\sigma}^*) \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D}^* \right) - E(\underline{\sigma}^*) \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{\underline{K}}^{-1}(\underline{\sigma}^*) \frac{\partial}{\partial \underline{\sigma}} \left(\underline{\underline{K}}(\underline{\sigma}^*) \cdot \underline{D}^* \right). \tag{5.5}$$

  It can be shown that the Frobenius norm of the matrix $\underline{\underline{T}}$ can be estimated through

  $$\|\underline{\underline{T}}\|_F \leq R \cdot c \cdot h^{-5} \cdot L_E \cdot \frac{\bar{e}}{(1-\nu)^2} \tag{5.6}$$

  with some constant $0 < c \neq c(h)$ (cf. [19, p. 91ff]). Due to (5.3), $\|\underline{\underline{T}}\|_F$ is lower than 1, i.e., the matrix $\underline{\underline{I}} - \underline{\underline{T}}$ is invertible, which implies via (5.4) that the vectors β and α are zero. Therefore $F'(x^*)$ is indeed regular. □

In fact, computing the upper bound for the Frobenius norm of T in (5.6) for some numerical examples yields horribly large numbers. On the other hand, if one calculates the norm for the matrix $\underline{\underline{T}}$ for the numerical examples given in section 8 (with some program), it is obtained that $\|\underline{\underline{T}}\|_F \approx 10$. (In fact, the applied forces, the prescribed temperature distribution, and L_E would have to be smaller to make sure that $\|\underline{\underline{T}}\|_F$ is really lower than one.)

Thus we may conclude that the estimate (5.6) is extremely rough and the result is not useful for practical problems. On the other hand these considerations supply some insight into the nature of the problem.

REMARK 5.7. *Looking at the proof of Theorem* 5.6*, it would suffice that* 1 *is not an eigenvalue of the matrix* T *instead of proving that the norm of* $\underline{\underline{T}}$ *is lower than* 1*. This condition is extremely difficult to verify, but for numerical experiments we should keep in mind that Newton's method will probably converge for a broader class of material laws than those given by* (5.3)*.*

5.3. Modified Newton method. We assume that the matrix $\underline{\underline{T}}$ (cf. (5.5)) is small compared to the identity matrix (measured in the Frobenius norm) in each iteration step. Motivated by Algorithm 5.4 we have the following algorithm.

ALGORITHM 5.8.

$$\begin{cases} \Delta\underline{D}^{(j)} = \underline{\underline{K}}^{-1}\left[\underline{Q} - \underline{\underline{K}}\cdot\underline{D}^{(j)} - \frac{\partial}{\partial\underline{\sigma}}(\underline{\underline{K}}\cdot\underline{D}^{(j)})\Delta\underline{\sigma}^{(j)}\right], \\ \underline{\sigma}^{(j+1)} = E(\underline{\sigma}^{(j)})\underline{\underline{\tilde{G}}}\cdot\underline{\underline{C}}\cdot\underline{D}^{(j)} + \underline{\beta}\Delta T + E(\underline{\sigma}^{(j)})\underline{\underline{\tilde{G}}}\cdot\underline{\underline{C}}\cdot\underline{\underline{K}}^{-1}(\underline{Q} - \underline{\underline{K}}\cdot\underline{D}^{(j)}). \end{cases}$$

(See Algorithm 5.8; now the matrix $\underline{\underline{T}}$ is neglected.) In order to compare Algorithms 5.4 and 5.8, let the iterates $\underline{D}^{(j)}$ and $\underline{\sigma}^{(j)}$ be the same for Newton's algorithm and Algorithm 5.8. (In the following we will denote an iterate of Newton's algorithm by the subscript "NEW" and an iterate of Algorithm 5.8 by the subscript "MOD.") Then the error in the next iteration point $(\underline{D}^{(j)}, \underline{\sigma}^{(j)})$ can be estimated as follows. We define

$$\underline{g}^{(j)} := -\underline{\sigma}^{(j)} + E(\underline{\sigma}^{(j)})\underline{\underline{\tilde{G}}}\cdot\underline{\underline{C}}\cdot\underline{D}^{(j)} + \underline{\beta}\Delta T + E(\underline{\sigma}^{(j)})\underline{\underline{\tilde{G}}}\cdot\underline{\underline{C}}\cdot\underline{\underline{K}}^{-1}(\underline{Q} - \underline{\underline{K}}\cdot\underline{D}^{(j)}),$$

and $\underline{\underline{T}}^{(j)}$ should be given by (5.5), where the iterates $\underline{D}^{(j)}$ and $\underline{\sigma}^{(j)}$ have to be inserted for D and σ, respectively. If we assume that $\|\underline{\underline{T}}^{(j)}\|_F < (\|\underline{\underline{I}}\|_F)^{-1}$, then it holds that

$$\|\underline{\sigma}_{MOD}^{(j+1)} - \underline{\sigma}_{NEW}^{(j+1)}\| \leq \|\underline{\underline{I}} - (\underline{\underline{I}} - \underline{\underline{T}}^{(j)})^{-1}\|_F \cdot \|\underline{g}^{(j)}\| \leq \frac{\|\underline{\underline{I}}\|_F^2 \cdot \|\underline{\underline{T}}^{(j)}\|_F}{1 - \|\underline{\underline{I}}\|_F \cdot \|\underline{\underline{T}}^{(j)}\|_F} \cdot \|\underline{g}^{(j)}\|$$

and

$$\|\underline{D}_{MOD}^{(j+1)} - \underline{D}_{NEW}^{(j+1)}\| \leq \|\underline{\underline{K}}^{-1}\|_F \cdot \left\|\frac{\partial}{\partial\underline{\sigma}}(\underline{\underline{K}}\cdot\underline{D}^{(j)})\right\|_F \cdot \|\underline{\sigma}_{MOD}^{(j+1)} - \underline{\sigma}_{NEW}^{(j+1)}\|,$$

where we used the "perturbation lemma"

$$\|\underline{\underline{A}}^{-1} - \underline{\underline{B}}^{-1}\| \leq \frac{\|\underline{\underline{B}}^{-1}\|^2 \cdot \|\underline{\underline{A}} - \underline{\underline{B}}\|}{1 - \|\underline{\underline{B}}^{-1}\| \cdot \|\underline{\underline{A}} - \underline{\underline{B}}\|}$$

for a regular matrix $\underline{\underline{B}}$ and a matrix $\underline{\underline{A}}$ with $\|\underline{\underline{A}} - \underline{\underline{B}}\| \cdot \|\underline{\underline{B}}^{-1}\| < 1$ (cf. [21, p. 162]).

Therefore the difference in the next iterates over the jth finite element is "small" if the Frobenius norm of the matrix $\underline{\underline{T}}^{(j)}$ is sufficiently small.

This algorithm has the great advantage (compared with Algorithm 5.4) that no linear equation has to be solved for computing $\underline{\sigma}^{(j+1)}$. The dimension of the corresponding matrix $\underline{\underline{I}} - \underline{\underline{T}}$ (in Algorithm 5.4) is $Nf \times 3Ne$, and the matrix is full, which is extremely costly in terms of computing time.

We could not find a convergence result for Algorithm 5.8, but the numerical results in section 8 together with the above considerations indicate that a positive answer exists.

5.4. Homotopy method. In this section we want to improve the algorithms for solving the discrete equilibrium equations (2.2) and (2.5) developed so far for the nonlinear material law given in Remark 1.1. The method suggested can be transferred to other nonlinear material laws in the same manner.

We mentioned in subsection 5.1 that some difficulties may occur (and in practice do indeed occur) for Algorithms 5.1, 5.4, and 5.8. On the one hand, the global displacement vector $\underline{D}$, respectively, L_E, is in practice not as small as needed for the contractivity of the fixed point operator $\mathcal{M}$ and the smallness of the matrix $\underline{\underline{T}}$ as defined in (5.5). On the other hand, Algorithm 5.4 is only locally convergent, i.e., the starting point x^0 might be too far away from the solution x^*.

In fact, if Young's modulus were constant, i.e., $E = E_0 = \text{const}$, then only one step of Algorithm 5.4 would have to be performed in order to yield the solution to (2.2) and (2.5). Furthermore, the nonlinearity of the material law (1.7) lies in the parameter y_1: if y_1 were 0, the material law would be constant. Thus one can try to use some homotopy method with respect to the parameter y_1 to solve (1.1) and (1.6) for the nonlinear material law (1.7).

We define the operator $H(y; \underline{D}, \underline{\sigma}) := F_y(\underline{D}, \underline{\sigma})$, where F_y is given by (5.1) and the subindex "y" denotes that in the material law (1.7) the parameter y_1 is set to y. Then (1.1) and (1.6) can be written as

$$H(y_1; \underline{D}, \underline{\sigma}) = 0. \tag{5.7}$$

The problem "$H(0; \underline{D}, \underline{\sigma}) = 0$" corresponds to a material law, where Young's modulus is constant, namely, $E = y_0 + y_2 \cdot \Delta T$, i.e., this is a linear problem and therefore it is straightforward to find the solution $(\underline{D}^0, \underline{\sigma}^0)$. Now we can try to find the solution of (5.7) by the iteration scheme, as follows.

ALGORITHM 5.9.

$$
\begin{array}{l}
\textit{given } \tilde{y} \textit{ set } \Delta y := (y_1 - \tilde{y})/n \textit{ with } n \in \mathbb{N}; y^{(0)} := \tilde{y} \\
\left.\begin{array}{l}
\quad \textit{set } y^{(k)} = y^{(k-1)} + \Delta y \\
\quad \textit{if } y^{(k)} > y_1 \textit{ then stop} \\
\quad \textit{solve } H(y^{(k)}; \underline{D}^{(k)}, \underline{\sigma}^{(k)}) = 0, \textit{ e.g., by some Newton method} \\
\qquad\qquad \textit{with starting point } (\underline{D}^{(k-1)}, \underline{\sigma}^{(k-1)})
\end{array}\right\} \forall\, k = 1, 2, \ldots.
\end{array}
$$

In solving problem

$$H(y^{(k)}; \underline{D}^{(k)}, \underline{\sigma}^{(k)}) = 0 \tag{5.8}$$

in Algorithm 5.9 one *predictor step* is usually made to get an approximation for the solution $(\underline{D}^{(k)}, \underline{\sigma}^{(k)})$, followed by several *corrector steps* in order to get nearer to the curve $y \mapsto (\underline{D}(y), \underline{\sigma}(y))$ that is implicitly defined through (5.8) (cf. [1]). The starting homotopy parameter $\tilde{y}$ may be set to zero or to some value "nearer" to the final value y_1.

Looking at Algorithm 5.9 the following question arises: How does the increment Δy have to be chosen such that the solution point $(\underline{D}^{(k-1)}, \underline{\sigma}^{(k-1)})$ of the previous iteration is sufficiently near the solution of the problem $H(y^{(k)}, \underline{D}^{(k)}, \underline{\sigma}^{(k)}) = 0$ and Newton's method will indeed converge? An answer to this question is, e.g., given in [1], [23, pp. 249–276], and [25]. In [23] conditions for choosing a uniform stepsize are given in order to yield a converging solution algorithm.

For our numerical examples in section 8 it was sufficient to use only predictor steps and to implement a fairly simple adaption algorithm for the steplength Δy (cf. subsection 7.4).

6. Calculation of sensitivities. In the previous section we developed algorithms to solve the discrete nonlinear equilibrium equations (2.2) and (2.5). Once the solutions $\underline{D}$ and $\underline{\sigma}$ are found, the objective function and the constraint functions of the design problems (3.6) and (3.8) can be evaluated, but we also need derivatives and therefore more information.

We want to state here that in this section we will only deal with the derivatives of the discrete problem. Furthermore, we assume throughout this section that the material law function E should be continuously differentiable.

For solving the discrete design problem (3.6) or (3.8), abstractly written as

$$\begin{array}{lccl} \phi(\underline{t}, \underline{D}(\underline{t}), \underline{\sigma}(\underline{t})) & \longrightarrow & \min\limits_{\underline{t}\in\mathbb{R}^{Nd}}, & \\ c_i(\underline{t}, \underline{D}(\underline{t}), \underline{\sigma}(\underline{t})) & \leq & 0 & \forall\ i = 1, \dots, M \end{array} \tag{6.1}$$

(cf. section 3 for the definition of the objective function ϕ and the constraint functions c_i, $i = 1, \dots, M$), we will use the SQP algorithm (together with an active set strategy; cf., e.g., [18]): In each iteration step the new iterate $\underline{x}_{k+1}$ is found through a line search along a certain search direction $\underline{d}_k$, i.e.,

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \cdot \underline{d}_k.$$

The search direction $\underline{d}_k$ is determined as the solution of the quadratic optimization problem

$$\begin{array}{ll} \text{minimize } \frac{1}{2}\underline{d}^T \underline{\underline{B}}_k \underline{d} + \nabla\phi(x_k)^T \underline{d} & \text{in } \mathbb{R}^n \\ \qquad \nabla c_j(x_k)^T \underline{d} + c_j(x_k) \leq 0 & \forall\ j = 1, \dots, M, \end{array}$$

where gradient information about the objective function and the constraint functions is used. Thus we have to compute the derivatives of the objective and the constraint functions with respect to the design variables t_i, $i = 1, \dots, Nd$.

As indicated in the formulation of the design problem (6.1), the displacement vector $\underline{D}$ and the stress vector $\underline{\sigma}$ depend upon the vector of design variables $\underline{t}$ via the formulas established in section 2. Therefore it follows by the chain rule that

$$\begin{aligned} \frac{d}{dt_i}\phi(\underline{D}, \underline{\sigma}, \underline{t}) &= \frac{\partial\phi}{\partial D}(\underline{D}, \underline{\sigma}, \underline{t}) \cdot \frac{\partial D}{\partial t_i} + \frac{\partial\phi}{\partial\sigma}(\underline{D}, \underline{\sigma}, \underline{t}) \cdot \frac{\partial\sigma}{\partial t_i} + \frac{\partial\phi}{\partial t_i}(\underline{D}, \underline{\sigma}, \underline{t}), \\ \frac{d}{dt_i}c_j(\underline{D}, \underline{\sigma}, \underline{t}) &= \frac{\partial c_j}{\partial D}(\underline{D}, \underline{\sigma}, \underline{t}) \cdot \frac{\partial D}{\partial t_i} + \frac{\partial c_j}{\partial\sigma}(\underline{D}, \underline{\sigma}, \underline{t}) \cdot \frac{\partial\sigma}{\partial t_i} + \frac{\partial c_j}{\partial t_i}(\underline{D}, \underline{\sigma}, \underline{t}) \end{aligned} \tag{6.2}$$

$\forall i = 1, \ldots, Nd,\ j = 1, \ldots, M$. Thus, one has to calculate the *sensitivities*

$$\frac{\partial D}{\partial t_i} \text{ and } \frac{\partial \sigma}{\partial t_i} \qquad \forall\ i = 1, \ldots, Nd \tag{6.3}$$

in order to be able to compute the derivatives (6.2).

If the material law function E is once continuously differentiable and the Frobenius norm of the matrix $\underline{\underline{T}}$ (cf. (5.5)) is “small” (cf. subsection 5.2), the sensitivities (6.3) can be computed by applying the implicit function theorem (cf. [10, p. 295]) to the function

$$F(\underline{t}, \underline{D}, \underline{\sigma}) := \begin{pmatrix} \underline{\underline{K}}(\underline{t}, \underline{\sigma}) \cdot \underline{D} - \underline{Q}(\underline{t}) \\ \underline{\sigma} - E(\underline{\sigma})\underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D} - \Delta T \cdot \underline{\beta} \end{pmatrix},$$

i.e.,

$$\begin{pmatrix} \dfrac{\partial D}{\partial t} \\ \dfrac{\partial \sigma}{\partial t} \end{pmatrix} = -(\nabla_{\underline{D},\underline{\sigma}} F)^{-1}(\underline{t}, \underline{D}, \underline{\sigma}) \cdot \frac{F(\underline{t}, \underline{D}, \underline{\sigma})}{\partial t} \Longleftrightarrow$$

$$\begin{bmatrix} \underline{\underline{K}}(\underline{\sigma}, \underline{t}) & \frac{\partial}{\partial \sigma}\left(\underline{\underline{K}}(\underline{\sigma}, \underline{t}) \cdot \underline{D}\right) \\ -E(\underline{\sigma}) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D} & \underline{\underline{I}} - \frac{\partial}{\partial \sigma}(E(\underline{\sigma}) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D}) \end{bmatrix} \cdot \begin{pmatrix} \dfrac{\partial D}{\partial t} \\ \dfrac{\partial \sigma}{\partial t} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial Q}{\partial t} - \dfrac{\partial K}{\partial t}\underline{D} \\ 0 \end{pmatrix}.$$

Thus $\frac{\partial D}{\partial t}$ and $\frac{\partial \sigma}{\partial t}$ can be computed analogously to the displacement vector $\underline{D}$ and the stress vector $\underline{\sigma}$ in Algorithm 5.4 via Algorithm 6.1.

Algorithm 6.1.

$$\left\{ \begin{aligned} &\frac{\partial D_*}{\partial t} = \underline{\underline{K}}^{-1} \left[\frac{\partial Q}{\partial t} - \frac{\partial K}{\partial t} \cdot \underline{D}_* - \frac{\partial}{\partial \underline{\sigma}}(\underline{\underline{K}} \cdot \underline{D}_*) \frac{\partial \sigma_*}{\partial t} \right], \\ &\left[\underline{\underline{I}} - \frac{\partial}{\partial \underline{\sigma}}(E(\underline{\sigma}_*)\underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D}_*) + E(\underline{\sigma}_*)\underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{\underline{K}}^{-1} \frac{\partial}{\partial \underline{\sigma}}(\underline{\underline{K}} \cdot \underline{D}_*) \right] \frac{\partial \sigma_*}{\partial t} \\ &\qquad = E(\underline{\sigma}_*)\underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{\underline{K}}^{-1} \left(\frac{\partial Q}{\partial t} - \frac{\partial K}{\partial t} \cdot \underline{D}_* \right), \end{aligned} \right. \tag{6.4}$$

where $\underline{D}_*$ *and* $\underline{\sigma}_*$ *are the solutions to* (2.2) *and* (2.5)*; notice that* $\underline{\underline{K}} = \underline{\underline{K}}(\underline{\sigma}^*)$.

If the matrix $\underline{\underline{T}}$ (cf. (5.5)) is small compared to the identity matrix $\underline{\underline{I}}$, these equations become

$$\begin{aligned} \frac{\partial D_*}{\partial t} &= \underline{\underline{K}}^{-1} \left[\frac{\partial Q}{\partial t} - \frac{\partial K}{\partial t} \cdot \underline{D}_* - \frac{\partial}{\partial \underline{\sigma}}(\underline{\underline{K}} \cdot \underline{D}_*) \frac{\partial \sigma_*}{\partial t} \right], \\ \frac{\partial \sigma_*}{\partial t} &= E(\underline{\sigma}_*) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{\underline{K}}^{-1} \left(\frac{\partial Q}{\partial t} - \frac{\partial K}{\partial t} \cdot \underline{D}_* \right). \end{aligned} \tag{6.5}$$

If we omit the term $\frac{\partial}{\partial \underline{\sigma}}(\underline{\underline{K}} \cdot \underline{D}_*)\frac{\partial \sigma_*}{\partial t}$, we obtain the formulas

$$\begin{aligned} \frac{\partial D_*}{\partial t} &= \underline{\underline{K}}^{-1} \cdot \left[\frac{\partial Q}{\partial t} - \frac{\partial K}{\partial t} \cdot \underline{D}_* \right], \\ \frac{\partial \sigma_*}{\partial t} &= E(\underline{\sigma}_*) \cdot \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \frac{\partial D_*}{\partial t}. \end{aligned} \tag{6.6}$$

Clearly (6.5) or (6.6) can be solved with less effort than (6.4), but one must check for each example whether the matrix $\underline{\underline{T}}$ or the derivatives with respect to σ may really be neglected.

7. Algorithm for solving the discrete design problem. We already mentioned in section 6 that we solve the optimization problem

$$\begin{aligned} \phi(\underline{t}, \underline{D}(\underline{t}), \underline{\sigma}(\underline{t})) &\longrightarrow \min_{\underline{t}\in\mathbb{R}^{Nd}}, \\ c_i(\underline{t}, \underline{D}(\underline{t}), \underline{\sigma}(\underline{t})) &\leq 0 \qquad \forall\, i = 1, \dots, M \end{aligned} \tag{7.1}$$

with the sequential quadratic programming algorithm (together with an active set strategy). Therefore we need to evaluate the objective function ϕ and the constraint functions c_i, $i = 1, \dots, M$ as well as their gradients $\nabla_t f$ and $\nabla_t c_i$, $i = 1, \dots, M$.

In section 5 we developed an algorithm to compute the displacement vector $\underline{D}$ and the stress vector $\underline{\sigma}$, and in section 6 we developed an algorithm for calculating their sensitivities with respect to the design variables.

Therefore we are able to perform both function and gradient evaluations, i.e., we are ready to formulate an algorithm for solving the discrete design problem.

7.1. Statement of the algorithm. Combining Algorithms 5.1 and 5.9 to compute $\underline{D}$ and $\underline{\sigma}$, and (6.6) to calculate $\partial D/\partial t$ and $\partial\sigma/\partial t$, respectively, our algorithm to solve the discrete design problem (7.1) is given by Algorithm 7.1 (again superscripts denote the iteration index), where $\underline{D}^{(n)}$ and $\underline{\sigma}^{(n)}$ are the last iterates in the inner loop and $\underline{D}_*$ and $\underline{\sigma}_*$ are the obtained solutions of discrete equilibrium equations. The starting point $(\underline{D}^{(0)}, \underline{\sigma}^{(0)})$ can be computed by using some linear material law which approximates the given nonlinear material law (e.g., setting $E = E(0) = \text{const}$) or choosing the solution of the foregoing iteration.

In Algorithm 7.1 we have chosen (6.6) in order to compute the sensitivities of $\underline{D}$ and $\underline{\sigma}$. Of course, one should use (6.4), (6.5), or (6.6) for computing the sensitivities that correspond to the algorithm one uses for solving the discrete equilibrium equations (cf. Algorithm 5.1, 5.4, or 5.8).

In [18, p. 210ff] a convergence result is given for the SQP algorithm together with an active set strategy, just as we use it to solve the design problems (3.6) and (3.8), but the conditions given there are impossible to prove for our settings. Therefore we are not able to state a priori parameter choice rules that guarantee the convergence of Algorithm 5.8.

7.2. Stopping criteria. In order to reach convergence in the outer loop of Algorithm 7.1, both of the following stopping criteria have to be fulfilled:

1. The relative change in the objective function should be sufficiently small, i.e.,

$$\Delta_F := \frac{|\phi^{(n)} - \phi^{(n-1)}|}{\phi^{(n-1)}} < \varepsilon_F, \tag{7.2}$$

2. the sum of the constraint violations should be sufficiently small, i.e.,

$$\Delta_C := \max_{i=1}^{M}(0, c_i(t)) < \varepsilon_C \tag{7.3}$$

for some suitable positive parameters ε_F and ε_C. For the fixed point iteration, respectively, the (modified) Newton iteration, in the inner loop the relative error of two successive iteration points has to be "small enough," i.e.,

$$\|\underline{D}^{(j-1)} - \underline{D}^{(j)}\| < \varepsilon_D \cdot \|\underline{D}^{(j-1)}\| \quad \text{and} \quad \|\underline{\sigma}^{(j-1)} - \underline{\sigma}^{(j)}\| < \varepsilon_\sigma \cdot \|\underline{\sigma}^{(j-1)}\|$$

for some suitable positive parameters ε_D and ε_σ.

Algorithm 7.1.

```
given $\underline{t}_0$, $\Delta y_1$, $\tilde{y}$
DO WHILE (not converged yet)
    compute initial points $\underline{D}^{(0)}$ and $\underline{\sigma}^{(0)}$; set $y := \tilde{y}$
    DO WHILE ( $|y| \leq |y_1|$ )
        DO WHILE (not converged yet)
            $\Delta \underline{D}^{(j)} = \underline{\underline{K}}^{-1}(\underline{t}_0, \underline{\sigma}^{(j)}) \cdot \left[ \underline{Q}(\underline{t}_0) - \underline{\underline{K}}(\underline{t}_0, \underline{\sigma}^{(j)}) \cdot \underline{D}^{(j)} \right]$
            $\underline{D}^{(j+1)} = \underline{D}^{(j)} + \Delta \underline{D}^{(j)}$
            $\underline{\sigma}^{(j+1)} = E(\underline{\sigma}^{(j)}) \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \underline{D}^{(j+1)} + \underline{\beta} \cdot \Delta T$
        ENDDO
        $\underline{D}^{(0)} = \underline{D}^{(n)}, \underline{\sigma}^{(0)} = \underline{\sigma}^{(n)}$
        $y = y + \Delta y$
    ENDDO
    $\dfrac{\partial D_*}{\partial t} = \underline{\underline{K}}^{-1}(\underline{t}_0, \sigma_*) \cdot \left[ \dfrac{\partial Q(\underline{t}_0)}{\partial t} - \dfrac{\partial K(\underline{t}_0, \sigma_*)}{\partial t} \cdot \underline{D}_* \right]$
    $\dfrac{\partial \sigma_*}{\partial t} = E(\sigma_*) \underline{\underline{\tilde{G}}} \cdot \underline{\underline{C}} \cdot \dfrac{\partial D_*}{\partial t}$
    input $\underline{D}_*, \underline{\sigma}_*, \dfrac{\partial D_*}{\partial t}, \dfrac{\partial \sigma_*}{\partial t}$ into optimizer (one SQP step) $\Rightarrow \underline{t}_*$
    $\underline{t}_0 = \underline{t}_*$
ENDDO
```

Since we do not solve the nonlinear discrete equilibrium equations (2.2) and (2.5) exactly, but only up to an error $\varepsilon > 0$, the following question arises: What effect does this have on the solution of the design problem (7.1)? An answer to this question will be given in [20] using results of the theory of point-to-set mappings.

7.3. Improvement of the algorithm. In Algorithm 7.1 there are several levels of convergence:

1. the convergence of the whole design cycle,
2. the convergence of Newton's method for calculating $\underline{D}$ and $\underline{\sigma}$,
3. the convergence of the optimizer (if one uses approximations for the objective and the constraint functions; see also [16]).

Since some software library will be used for the optimization, the convergence of the optimizer is not included in the following considerations.

If the whole optimization is far away from the optimal solution, it would not be very efficient to compute the displacement vector and the stress vector up to a high accuracy. Thus at the beginning, larger relative errors in the finite element analysis will be accepted and the "level of acceptance" will be decreased when the minimum is approached, e.g., we have Algorithm 7.2.

Algorithm 7.2.

```
given $0 < \xi_1, \xi_2, \eta_1, \eta_2 < 1$ and $\kappa_F, \kappa_C > 1$
IF ($\Delta_F \le \kappa_F \cdot \varepsilon_F$ AND $\Delta_C \le \kappa_C \cdot \varepsilon_C$) THEN
    $\varepsilon_D = \xi_1 \cdot \varepsilon_D$
    $\varepsilon_\sigma = \xi_2 \cdot \varepsilon_\sigma$
    $\kappa_F = \max(\eta_1 \cdot \kappa_F, 1)$
    $\kappa_C = \max(\eta_2 \cdot \kappa_C, 1)$
ENDIF
```

It is recommended that the last optimization cycle be repeated with the desired accuracy for solving the discrete equilibrium equations, since there is no guarantee that the final accuracy achieved by coupling Algorithm 7.1 with Algorithm 7.2 is better than the desired accuracy.

The whole algorithm for solving the design problem (7.1), including the improvements of this subsection, is summarized in Figure 7.1 (the module ACONTR performs Algorithm 7.2).

7.4. Dynamic update of parameters. The parameters used for the homotopy method $(\tilde{y}, \Delta y)$ and for modifying the relative errors for the stopping criterion of Newton's method $(\kappa_F, \kappa_C, \xi_1, \xi_2, \eta_1, \eta_2)$ should probably be dynamically updated according to the actual state of the optimization process.

Since only a few iterations are needed for our numerical examples to reach the desired solution of the design problem (cf. section 8), it was not necessary to pay too much attention to choosing or adapting the "coupling parameters" κ_F, κ_C, ξ_1, ξ_2, η_1, η_2; it is important only that the relative errors ε_D and ε_σ used in the stopping criterion for Newton's method are decreased quite rapidly approaching the solution of the design problem.

The modification of the increment size Δy is coupled with the maximum number of iterations MAXLOP at solving the discrete equilibrium equations. The parameter MAXLOP plays a crucial role in the finite element analysis: Each time Newton's method terminates because the number of iterations exceeds MAXLOP, the step size Δy of the homotopy method will be decreased (e.g., halved) and a further solution to (1.1) and (1.6) will be started. If Δy becomes too small, the solution to (1.1) and (1.6) is stopped and the optimization is continued with the last iterates.

Thus it seems to be best to choose MAXLOP quite large, but on the other hand, if the number of iterations is too large and Newton's method does not converge at all, it would be better to abort this analysis as soon as possible. (We used the setting MAXLOP = 10 for numerical experiments; see section 8.)

These considerations complete our analysis of the existence and uniqueness of the discrete equilibrium equations and the discrete design problem and the convergence analysis of different algorithms to solve these problems. The only thing remaining is to test our algorithms for some numerical examples.

8. Numerical examples. In this section we will present some numerical results for a real-life problem from our industrial partner AVL. The goal is to find the optimal design of a unit injector rocker arm made of gray cast iron as the solution of the design

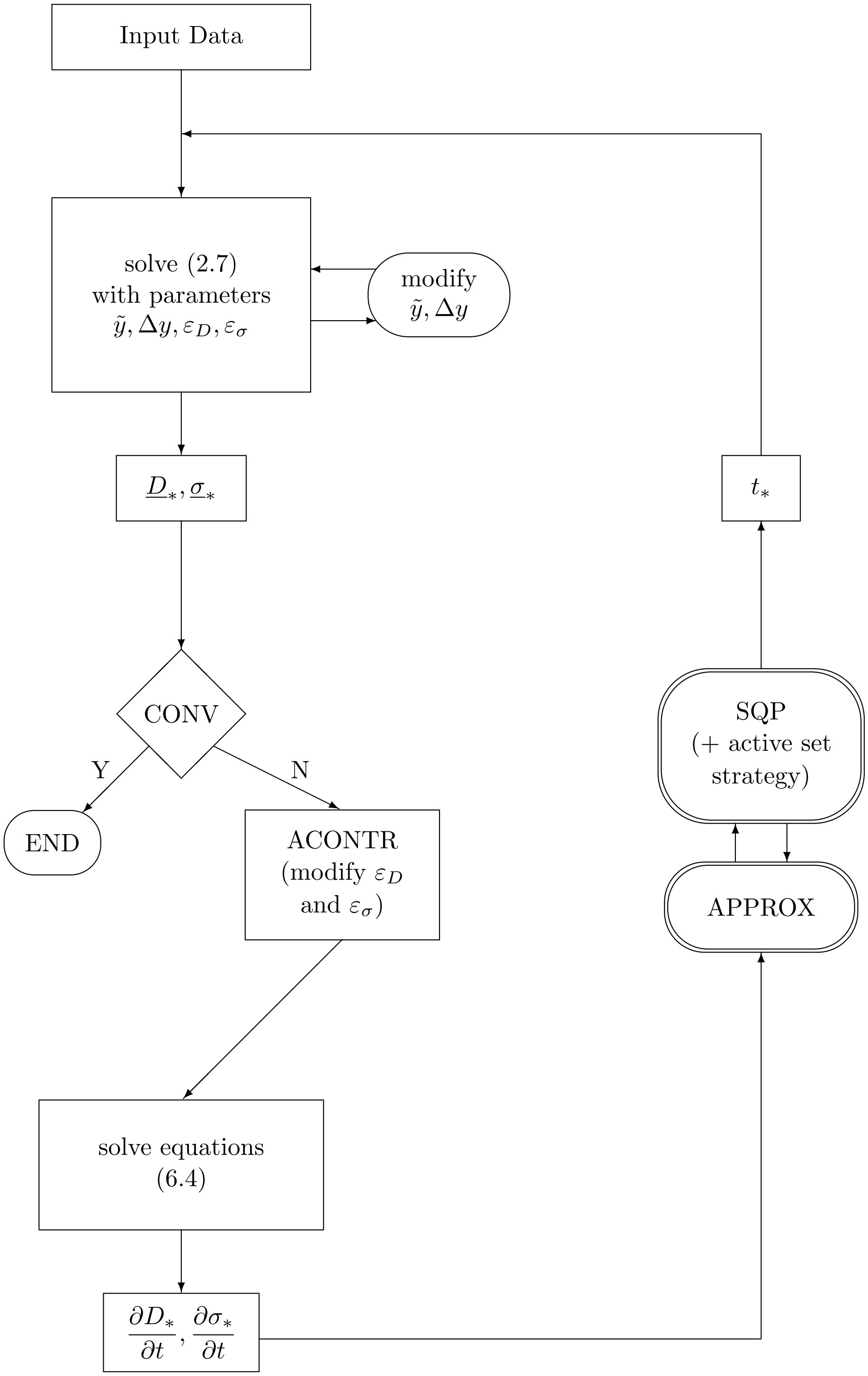

Input Data
solve (2.7)
with parameters
$\tilde{y}, \Delta y, \varepsilon_D, \varepsilon_\sigma$
modify
$\tilde{y}, \Delta y$
$\underline{D}_*, \underline{\sigma}_*$
CONV
Y
N
END
ACONTR
(modify ε_D
and ε_σ)
solve equations
(6.4)
$\frac{\partial D_*}{\partial t}, \frac{\partial \sigma_*}{\partial t}$
APPROX
SQP
(+ active set
strategy)
t_*

FIG. 7.1. *Flow diagram of the solution of the design problem for nonlinearly elastic materials.*

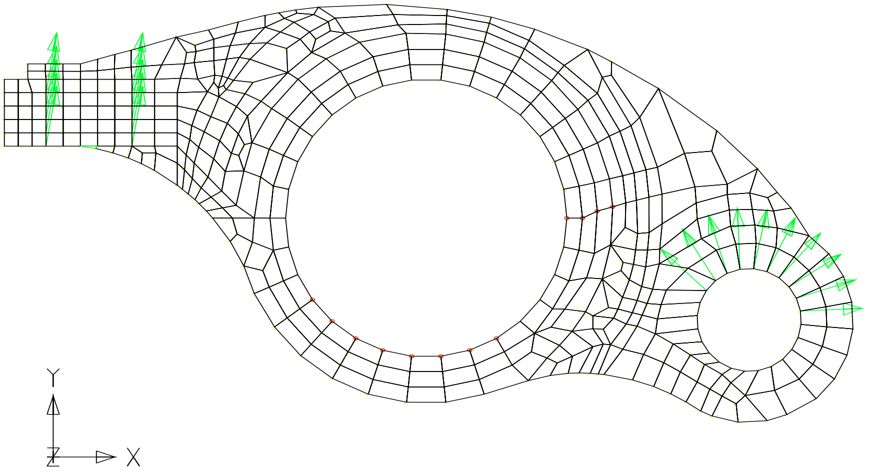


FIG. 8.1. *Finite element mesh, exerted forces and constraints of a unit injector rocker arm.*

problem (3.6), respectively (3.8). Therefore we suppose that the nonlinear material law considered is given by (1.7).

In the following the solution algorithm for the design problem developed in subsection 7.1 is modified according to [16]. There, not only one SQP step is performed within each optimization cycle, but the objective function and the constraint functions are approximated, e.g., quadratically, and then the optimization problem is solved in a user-specified trust region around the actual iteration point using these approximations (cf. Figure 7.1). The great advantage of this method is that the number of solutions of the equilibrium equations is remarkably lower, which is usually the most costly part of the whole optimization process.

For our problems we have used a mixed linear approximation for the objective function and the constraint functions, i.e. (we will write ϕ only as a function of $\underline{t}$),

$$\phi(\underline{t}) \approx \phi(\underline{t}^{(n)}) + \underline{b}^T \cdot \nabla_{\underline{t}}\phi(\underline{t}^{(n)})$$

with

$$b_i := \begin{cases} t_i - t_i^{(n)} & \text{for } (\nabla_{\underline{t}}\phi(\underline{t}^{(n)}))_i \geq 0, \\ -\frac{t_i^{(n)}}{t_i} \cdot (t_i - t_i^{(n)}) & \text{for } (\nabla_{\underline{t}}\phi(\underline{t}^{(n)}))_i < 0 \end{cases}$$

(cf. [16, p. 12]).

8.1. Problem description. First let us have a look at the finite element mesh of the unit injector rocker arm in Figure 8.1.

At the left and right side (with respect to the X-axis) of the object, there are applied surface forces, and at the lower part of the bigger inner circle the displacements of those grid points marked with a dot are prescribed (Dirichlet boundary conditions).

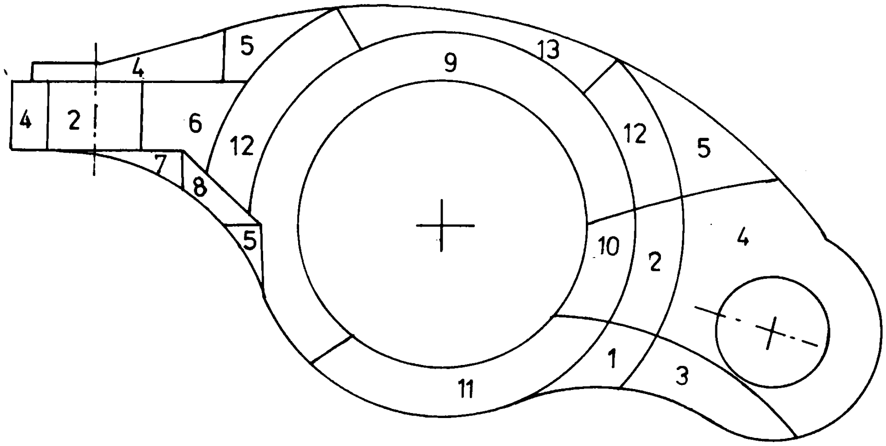


FIG. 8.2. *Partition of Ω in subdomains with constant thickness.*

The finite element mesh consists of 467 grid points and 397 elements (396 bilinear rectangles and 1 linear triangle). Thus the constants used in the previous sections for describing the finite element model are given by

$$Ne = 397, \qquad Nf = 926, \qquad Ng = 467.$$

The partition of the domain ω into subdomains $p^{(j)}$, $i = 1, \dots, Nd$, according to section 3 is given in Figure 8.2. The body is subdivided into 19 pieces, where there are 13 variable thicknesses, i.e., $Nd = 13$.

The original design and the von Mises stress contours can be seen in Figure 8.3. The surface in Figure 8.3 is an approximation of the skyline-like design defined via (3.3) that has almost the same mechanical properties (cf. also Remark 8.1). The same also holds for Figures 8.4 and 8.5.

8.2. Setting of the input parameter. In the following subsections the two design problems stated in section 3 are analyzed for the nonlinear material law (1.7) with the constants

$$y_0 = 1.3 \cdot 10^5, \quad y_1 = -320, \quad y_2 = -40.$$

Furthermore, the reference temperature T_{ref} of the body is set to 20 degrees centigrade and the temperature vector is initialized with 85 degrees centigrade, i.e., $T = 85$ all over Ω. The thermal expansion coefficients are given by $\beta_i = 1.02 \cdot 10^{-5} + 8.0 \cdot 10^{-8} \cdot (T - T_{ref})$.

The default values of the parameters controlling the finite element analysis are given by $\kappa_F = \kappa_C = 5.0$, $\xi_1 = \xi_2 = \eta_1 = \eta_2 = 0.5$, $\varepsilon_D = \varepsilon_\sigma = 0.05$, $\varepsilon_0 = 0.01$, $\varepsilon_C = \varepsilon_F = 5.0 \cdot 10^{-3}$ (cf. subsection 7.1), and the parameters for the homotopy method are set to $\tilde{y} = -260$, $\Delta y = -30$ (cf. subsection 5.4).

8.3. Objective equals minimal volume. The problem to be solved is

$$\begin{aligned} &\text{Volume} \longrightarrow \min, \\ &\sigma_{elm}^{(i)} \leq \bar{\sigma} \qquad && \forall\, i = 1, \dots, Me, \\ &\underline{\delta}_i \leq D_i \leq \bar{\delta}_i && \forall\, i = 1, \dots, Mf, \\ &\underline{\tau}_i \leq t_i \leq \bar{\tau}_i && \forall\, i = 1, \dots, Md, \end{aligned}$$

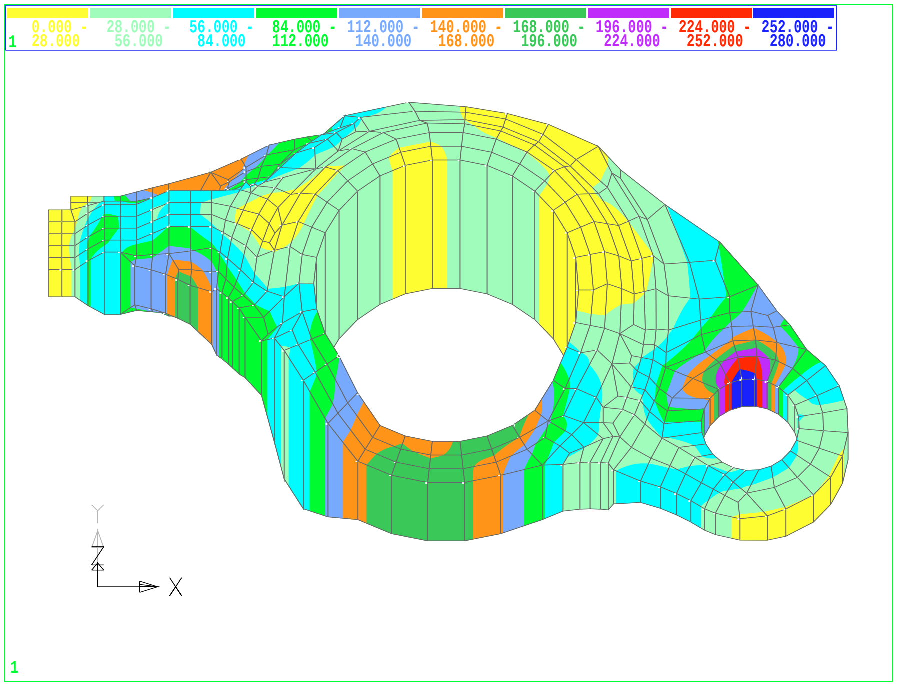


FIG. 8.3. *Original design and von Mises stress of unit injector rocker arm.*

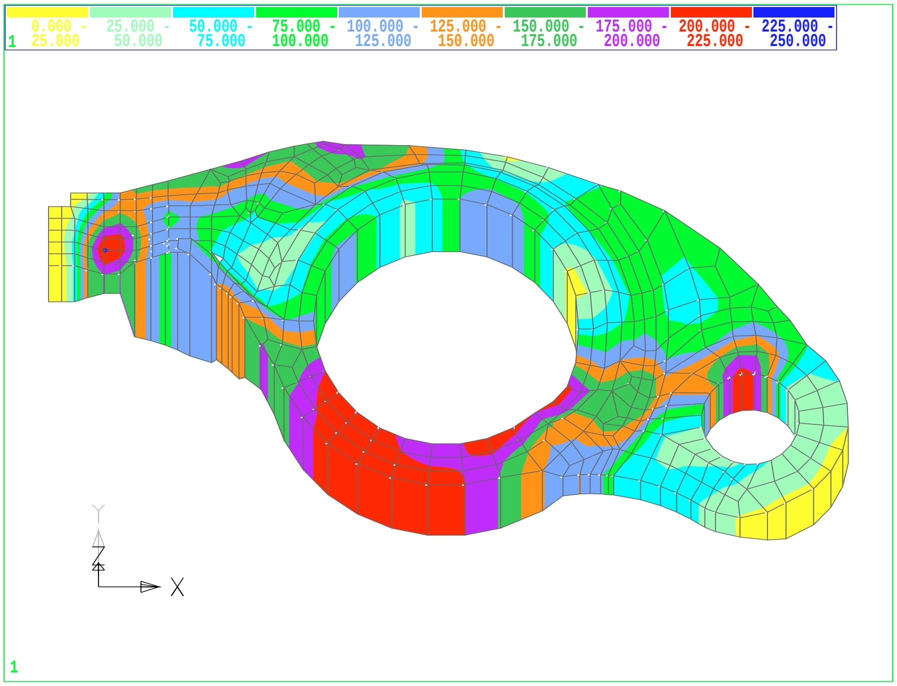


FIG. 8.4. *Optimal design (objective equals minimal volume) and von Mises stress of unit injector rocker arm (modified Newton method).*

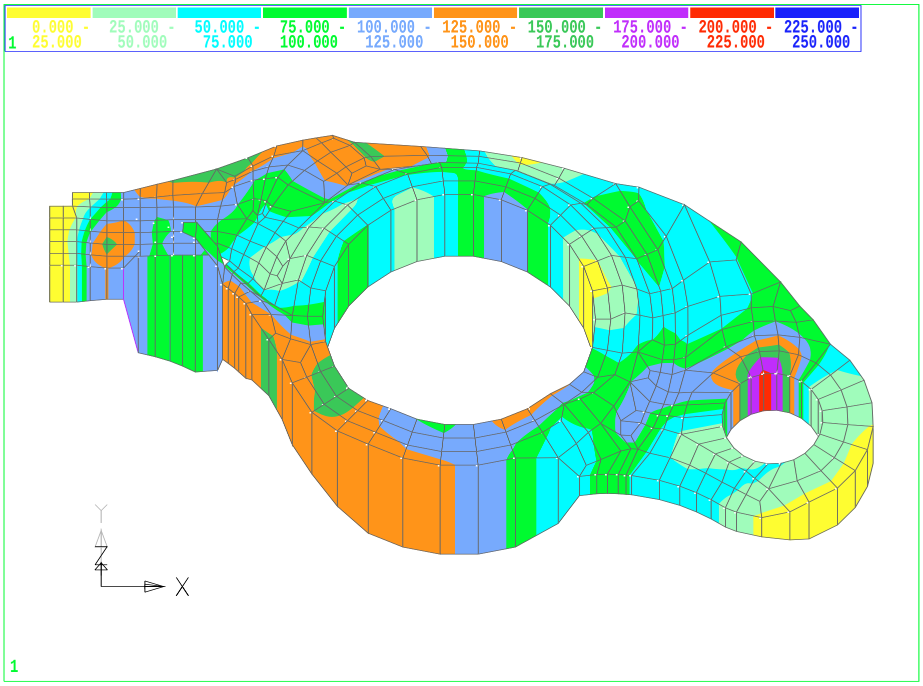


FIG. 8.5. *Optimal design (objective equals even stress distribution) and von Mises stress of unit injector rocker arm (modified Newton method).*

where $\underline{D}$ and $\underline{\sigma}$ are the solutions to (2.2) and (2.5). The obtained results (with the SQP algorithm together with an active set strategy) are given in Table 8.1.

The data in the first column correspond to the linear material law $E = y_0 + y_2 \cdot \Delta T$ (cf. [15]), where all stiffness matrices are computed by NASTRAN. The results in the second column are achieved by computing the stiffness matrix and the matrix of elasticity coefficients with polynomial shape functions. (The differences in the results of NASTRAN are due to the fact that NASTRAN uses different shape functions: relative error in the objective function is 1.4%.)

The third column contains the results for the nonlinear case described above. (Here we used $\varepsilon_D = \varepsilon_\sigma = 0.02$; Algorithm 5.1 did not converge for greater starting errors.) The value of the objective function of the original design was 26374.525; thus a decrease of 48.18% could be achieved in the nonlinear case. The corresponding design and the von Mises stress contours can be seen in Figure 8.4 (cf. Remark 8.1). On the one hand it can be seen in Figure 8.4 that the prescribed von Mises stress is not exceeded; on the other hand it is reached in some regions, since otherwise it would be possible to take away more material.

REMARK 8.1. *As mentioned in section* 3, *we assume that our problem is a plane stress problem, which has to be verified for our specific example.*

Therefore we performed some additional investigations in order to answer the question of whether the reduction of the three-dimensional object to a two-dimensional cross section is valid with sufficient accuracy: We generated a three-dimensional surface of the above calculated optimal design by assigning to the z-coordinate of a grid point (of the two-dimensional model) the arithmetic sum of the different heights of adjacent finite elements, ending up in the design drawn in Figure 8.4. (*The height is*

TABLE 8.1
Numerical results for the nonlinear material law (1.7) *for the design problems* (3.6) *and* (3.8).

	Linear case	Linear case	Nonlinear case	Nonlinear case
	NASTRAN	New routines	New routines	New routines
	$E = \text{const}$	$E = \text{const}$	$E = E(\sigma, T)$	$E = E(\sigma, T)$
		Fixed point	Fixed point	Modified Newton
F =	12933.643	12754.713	13668.458	14328.906
X =	5.000	5.000	5.000	5.000
	5.689	5.637	5.151	5.342
	5.000	5.000	5.000	5.000
	8.575	8.722	11.061	10.164
	10.573	10.708	10.826	10.303
	8.220	6.956	6.669	9.447
(3.6)	18.477	18.771	18.168	23.224
	14.481	13.752	14.993	18.576
	12.657	12.364	12.626	14.183
	7.423	7.298	6.334	5.764
	12.538	12.364	11.758	12.932
	6.740	6.209	7.921	8.548
	6.017	5.865	6.308	6.499
IT =	7	7	11	9
CPU =	00:02:19.08	00:02:59.38	00:09:22.27	00:05:41.70
F =	72.275	71.476	63.703	63.321
X =	5.103	5.124	5.091	5.324
	8.114	8.110	8.203	7.984
	5.000	5.000	5.000	5.000
	9.110	8.996	9.282	9.449
	13.493	13.553	13.996	14.280
	8.208	8.167	9.225	9.369
	33.120	34.106	34.862	33.046
(3.8)	19.334	19.300	17.677	17.113
	16.398	16.2463	15.955	16.643
	12.705	12.665	11.701	10.960
	22.568	22.571	21.604	21.598
	8.172	7.929	9.274	9.226
	6.403	6.533	6.007	5.662
IT =	6	6	7	7
CPU =	00:02:09.73	00:02:33.11	00:03:12.26	00:04:36.42

distributed symmetrically to the two-dimensional cross section; cf. (3.1) *and* (3.3).) *Solving the equilibrium equations for the resulting three-dimensional model, it turned out that the stresses are indeed almost uniformly distributed over the height and the results look very similar to those of the two-dimensional example.* (*In order to save space we will not print these comparisons here; cf.* [19, p. 117ff].)

Thus we may conclude that for our numerical examples the assumption of a plane stress problem is at least approximately justified.

The fourth column corresponds to the results obtained with the modified Newton method (cf. Algorithm 5.8). The convergence at solving (1.1) and (1.6) is improved enormously: In each design cycle only a few iterations (about 3) are needed to achieve a relative error in $\underline{D}$ and $\underline{\sigma}$ smaller than 5.10^{-7}. Furthermore, no homotopy method has to be used, i.e., $\tilde{y}$ can be set to -320. Compared to the solution of the fixed point iteration, the relative error in the objective function is 4.83% and the objective function could be decreased by 45.67%, which is only slightly worse than the result obtained with the fixed point iteration.

Thus the modified Newton method yields a similar result as the fixed point iteration, but it is preferable because there is no need to use a homotopy method (fewer parameters to choose) and it is faster.

8.4. Objective equals even stress distribution. The problem to be solved is

$$\begin{aligned} \sum_{j=1}^{Ne} \left(\frac{\sigma_{elm}^{(j)}}{\sigma_*} - 1 \right)^2 \quad &\longrightarrow \ \min, \\ \sigma_{elm}^{(i)} \leq \bar{\sigma} \quad &\forall\ i = 1, \dots, Me, \\ \underline{\delta}_i \leq D_i \leq \bar{\delta}_i \quad &\forall\ i = 1, \dots, Mf, \\ \underline{t} \leq t_i \leq \bar{t} \quad &\forall\ i = 1, \dots, Md. \end{aligned}$$

The obtained results (with SQP and an active set strategy) are given in Table 8.1 (here $\tilde{y}$ is set to -320, i.e., no homotopy method has to be used). The value of the objective function of the original design was 109.673; thus a decrease of 41.92% could be achieved in the nonlinear case. Furthermore, the volume of the body has been reduced by 36.37% (volume for optimal design is 16782.029). The corresponding design and the von Mises stress contours can be seen in Figure 8.5 (cf. Remark 8.1).

The convergence at solving (1.1) and (1.6) with the modified Newton method is improved in the same way as for the minimization of the volume. The relative error in the objective function is only 0.6%.

Since no homotopy method had to be used at the fixed point iteration, one can say that neither of the two methods is significantly superior to the other one.

9. Conclusions. In this paper we developed a methodology for solving sizing optimization problems for a class of nonlinearly elastic materials. We developed algorithms in order to solve the discrete equilibrium equations and to compute the sensitivities of the displacement and the stress vector, and we presented a concept of how to couple these modules efficiently with a finite element programming package and some optimization code.

We showed the good performance of our algorithms for practically relevant numerical examples by implementing our routines within the finite element programming package MSC/NASTRAN and using the SQP algorithm together with an active set strategy as the optimizer.

It turned out that using the modified Newton method in order to solve the nonlinear equilibrium equations yields quite the same results as the fixed point iteration within almost the same CPU time (per iteration), but more accurate gradient information is taken into account. Therefore for the modified Newton method no homotopy method has to be used, as opposed to the fixed point iteration for which there have occurred some difficulties in finding suitable input parameters in order to yield convergence.

Since this kind of finite element package and the optimizer coupled with our own routines can be treated as black boxes by our algorithms, the concept presented can be transferred to finite element codes other than MSC/NASTRAN and to optimization algorithms other than the SQP algorithm without any changes. Moreover, the theoretical results presented will also stay valid.

Acknowledgment. We want to thank Ralf Uwe Pfau at the Institute of Industrial Mathematics at the Johannes Kepler University of Linz for some hints on how to reduce the CPU time for the modified Newton method.

REFERENCES

[1] E. L. Allgower and K. Georg, *Numerical Continuation Methods: An Introduction*, Springer-Verlag, Berlin, Heidelberg, 1990.

[2] H. T. Banks, N. J. Lybeck, M. J. Gaitens, B. Muñoz, and L. Yanyo, *Computational methods for estimation in the modelling of nonlinear elastomers*, Kybernetika, 32 (1996), pp. 526–542.

[3] H. T. Banks, N. J. Lybeck, B. Muñoz, and L. Yanyo, *Nonlinear elastomers: Modelling and estimation*, in Proceedings of the 3rd IEEE Mediterranean Symposium on New Directions in Control, 1, Limassol, Cyprus, 1995, pp. 1–7.

[4] F. T. Calkins and A. B. Flatau, *Transducer-based measurements of Terfenol-D material properties*, in Smart Structures and Materials 1996: Smart Structures and Integrated Systems, I. Chopra, ed., SPIE, San Diego, CA, 1996.

[5] P. G. Ciarlet, *Mathematical Elasticity, Vol* I: *Three-Dimensional Elasticity*, Stud. Math. Appl. 20, North–Holland, Amsterdam, 1988.

[6] H. W. Engl and C. Stangl, *Existence and uniqueness of solutions of the equilibrium equations for a class of nonlinearly elastic materials*, ZAMM, 78 (1998), pp. 467–481.

[7] Ch. Grossmann and H.-G. Roos, *Numerik partieller Differentialgleichungen*, Teubner-Verlag, Stuttgart, 1992.

[8] W. Hackbusch, *Elliptic Differential Equations: Theory and Numerical Treatment*, Springer-Verlag, Berlin, 1992.

[9] J. Haslinger and P. Neittaanmäki, *Finite Element Approximation for Optimal Shape Design: Theory and Applications*, John Wiley, Chichester, 1988.

[10] H. Heuser, *Lehrbuch der Analysis, Teil* 2, 3rd ed., Teubner-Verlag, Stuttgart, 1986.

[11] L. Holzleitner, *Domain Optimization in Linearized Elasticity with the Finite Element Package MSC/NASTRAN*, Ph.D. thesis, Johannes Kepler University, Linz, Austria, 1996.

[12] E. J. Haug, K. K. Choi, and V. Komkov, *Design Sensitivity Analysis of Structural Systems*, Math. Sci. Engrg. 177, Academic Press, New York, 1986.

[13] E. J. Haug and K. K. Choi, *Methods of Engineering Mathematics*, Prentice–Hall, Englewood Cliffs, NJ, 1993.

[14] D. C. Jiles, *Theory of the magnetomechanical effect*, J. Phys. D, 28 (1995), pp. 1537–1546.

[15] K. G. Mahmoud, H. W. Engl, and L. Holzleitner, *Optimum structural design using MSC/NASTRAN and sequential quadratic programming*, Comput. & Structures, 52 (1994), pp. 437–447.

[16] K. G. Mahmoud, *Approximations in optimum structural design*, in Advances in Structural Optimization, B. H. V. Topping and M. Papadrakakis, eds., Civil-Comp Press, Edinburgh, 1994, pp. 57–67.

[17] J. M. Ortega, *Numerical Analysis*, Academic Press, New York, 1972.

[18] K. Schittkowski, *On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function*, Math. Oper. Statist., Ser. Optim., 14 (1983), pp. 197–216.

[19] C. Stangl, *Optimal Sizing for a Class of Nonlinearly Elastic Materials*, Ph.D. thesis, Johannes Kepler University, Linz, Austria, 1996.

[20] C. Stangl, *Stability of the Optimal Design with Respect to Inaccurate Solution of the Nonlinear State Problem*, submitted.

[21] J. Stoer, *Einführung in die Numerische Mathematik*, 4th ed., Springer-Verlag, Berlin, 1983.

[22] R. Temam, *Mathematical Problems in Plasticity*, Gauthier–Villars, Paris, 1985.

[23] H. Wacker, ed., *Continuation Methods*, Academic Press, London, 1978.

[24] O. C. Zienkiewicz, *Methode der Finiten Elemente*, Carl Hanser Verlag München, Wien, 1975.

[25] W. Zulehner, *Schrittweitensteuerung für Einbettungsmethoden*, Ph.D. thesis, Johannes Kepler University, Linz, Austria, 1981.

SIAM J. Optim.
Vol. 9, No. 2, pp. 444–465

A PREDICTOR-CORRECTOR INTERIOR-POINT ALGORITHM FOR THE SEMIDEFINITE LINEAR COMPLEMENTARITY PROBLEM USING THE ALIZADEH–HAEBERLY–OVERTON SEARCH DIRECTION*

MASAKAZU KOJIMA[†], MASAYUKI SHIDA[‡], AND SUSUMU SHINDOH[§]

Abstract. This paper proposes a globally convergent predictor-corrector infeasible-interior-point algorithm for the monotone semidefinite linear complementarity problem using the Alizadeh–Haeberly–Overton search direction, and shows its quadratic local convergence under the strict complementarity condition.

Key words. semidefinite linear complementarity problem, semidefinite programming, interior-point algorithm, predictor-corrector algorithm, local convergence, quadratic convergence

AMS subject classifications. 90C33, 90C05, 90C25, 65K10

PII. S1052623496300623

1. Introduction.

Several distinct search directions have been used in many interior-point algorithms [1, 2, 5, 8, 9, 15, 14, 16, 20, 22, 23, 24, 27, 31, 32, 34] for the semidefinite program (SDP). They are roughly classified into two groups. The search directions in one group [1, 5, 22, 23, 24, 31, 32] are founded on the self-concordant barrier or potential function [22] for the SDP, while each search direction in the other group [2, 8, 9, 15, 14, 16, 20, 27, 34] is derived from a certain linearization of the optimality condition for the SDP. The optimality condition consists of primal feasibility, dual feasibility, and complementarity equations. This paper is concerned with the latter group of search directions.

Among the search directions in the latter group, the one independently proposed by Helmberg et al. [8] and Kojima, Shindoh, and Hara [15], which we will call the HRVW/KSH/M search direction, has been studied extensively in recent papers [8, 9, 14, 16, 20, 27, 34]. In particular, Monteiro [20] devised a new formulation of the HRVW/KSH/M search direction. Many polynomial-time primal-dual interior-point algorithms for the linear program (LP), such as central trajectory following algorithms [10, 11, 19, 33], potential reduction algorithms [12, 17], predictor-corrector algorithms [18, 26], were extended to the SDP, and similar global (polynomial-time) computational complexities for the extended algorithms were established in those papers. Also there are a few articles [14, 27] that investigate the local convergence of interior-point algorithms for the SDP. Using Monteiro's new formulation of the HRVW/KSH/M search direction, Potra and Sheng [27] provided a sufficient condition for the superlinear convergence of an extension of the Mizuno–Todd–Ye-type predictor-corrector algorithm for the LP to the SDP. In their recent paper [14], Kojima, Shida, and Shindoh presented an example that exhibits a substantial difficulty in the local conver-

*Received by the editors March 15, 1996; accepted for publication (in revised form) July 15, 1997; published electronically March 17, 1999.

http://www.siam.org/journals/siopt/9-2/30062.html

[†]Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152, Japan (kojima@is.titech.ac.jp).

[‡]Department of Mathematics, Faculty of Engineering, Kanagawa University, Rokkakubashi, Kanagawa-ku, Yokohama, 221, Japan (shida@is.titech.ac.jp).

[§]Department of Mathematics and Physics, The National Defense Academy, Hashirimizu 1-10-20, Yokosuka, Kanagawa, 239, Japan (shindoh@cc.nda.ac.jp).

gence analysis of the Potra–Sheng extension of the Mizuno–Todd–Ye-type predictor-corrector algorithm. They deduced from the example that the Potra–Sheng extension needs an additional condition to attain the superlinear convergence. Their condition requires that the generated sequence converges to a solution of the SDP tangentially to the central path. This example gave the authors a motivation to explore the local convergence of interior-point algorithms using different search directions. It will be shown in the current paper that one of the search directions proposed by Alizadeh, Haeberly, and Overton [2], which we will call the AHO search direction (see (1.3) below), fits quadratic convergence under the strict complementarity condition quite well.

Besides what we have called the AHO search direction above, Alizadeh, Haeberly, and Overton [2] derived some other primal-dual search directions from linearization of the optimality condition for the SDP, and they reported some numerical results which showed that a primal-dual Mehrotra-type predictor-corrector interior-point algorithm using the AHO search direction worked more efficiently than the algorithms using the other primal-dual search directions. But no theoretical convergence analysis has been done on the algorithm using the AHO search direction.

Let $\mathcal{S}$ denote the set of all $n \times n$ symmetric real matrices. We regard $\mathcal{S}$ as an $n(n+1)/2$-dimensional linear space with the inner product $\boldsymbol{X} \bullet \boldsymbol{Y} = \mathrm{Tr}\, \boldsymbol{X}^T\boldsymbol{Y}$ of $\boldsymbol{X}$ and $\boldsymbol{Y}$ in $\mathcal{S}$ and the Frobenius norm $\|\boldsymbol{X}\|_F = (\boldsymbol{X} \bullet \boldsymbol{X})^{1/2}$ of $\boldsymbol{X} \in \mathcal{S}$, where $\mathrm{Tr}\, \boldsymbol{A}$ denotes the trace of an $n \times n$ matrix $\boldsymbol{A}$. We write $\boldsymbol{X} \succ \boldsymbol{O}$ if $\boldsymbol{X} \in \mathcal{S}$ is positive definite and $\boldsymbol{X} \succeq \boldsymbol{O}$ if $\boldsymbol{X} \in \mathcal{S}$ is positive semidefinite. Here $\boldsymbol{O}$ denotes the $n \times n$ zero matrix. We also use the symbols $\mathcal{S}_{++}$ and $\mathcal{S}_{+}$ for the set of positive definite symmetric matrices and the set of positive semidefinite symmetric matrices, respectively:

$$\mathcal{S}_{++} = \{\boldsymbol{X} \in \mathcal{S} \ : \ \boldsymbol{X} \succ \boldsymbol{O}\} \ \text{ and } \mathcal{S}_{+} = \{\boldsymbol{X} \in \mathcal{S} \ : \ \boldsymbol{X} \succeq \boldsymbol{O}\}.$$

Let $\mathcal{F}$ be an $n(n+1)/2$-dimensional affine subspace of $\mathcal{S} \times \mathcal{S}$, and

$$\mathcal{F}_{+} = \{(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F} \ : \ \boldsymbol{X} \succeq \boldsymbol{O}, \ \boldsymbol{Y} \succeq \boldsymbol{O}\}.$$

We are concerned with the semidefinite linear complementarity problem (SDLCP):

$$\text{find an } (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_{+} \ \text{ such that } \boldsymbol{X} \bullet \boldsymbol{Y} = 0. \tag{1.1}$$

We call an $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_{+}$ a feasible solution of the SDLCP (1.1). Throughout the paper we assume the monotonicity of the $n(n+1)/2$-dimensional affine subspace $\mathcal{F}$:

$$(\boldsymbol{U}' - \boldsymbol{U}) \bullet (\boldsymbol{V}' - \boldsymbol{V}) \geq 0 \ \text{ for every } (\boldsymbol{U}', \boldsymbol{V}'), \ (\boldsymbol{U}, \boldsymbol{V}) \in \mathcal{F}. \tag{1.2}$$

The monotone SDLCP was introduced in the paper [15] by Kojima, Shindoh, and Hara as an extension of the monotone LCP and a mathematical framework on which they founded interior-point algorithms. Besides the interior-point algorithms given in their paper [15], many of the primal-dual interior-point algorithms developed so far for the SDP can be extended to the monotone SDLCP.

If we adapt the AHO search direction [2] to the monotone SDLCP, we can describe it as a solution of the system of equations

$$\left\{ \begin{array}{l} \boldsymbol{X}d\boldsymbol{Y} + d\boldsymbol{Y}\boldsymbol{X} + d\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{Y}d\boldsymbol{X} = 2\beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{X}, \\ (\boldsymbol{X} + d\boldsymbol{X}, \boldsymbol{Y} + d\boldsymbol{Y}) \in \mathcal{F}. \end{array} \right. \tag{1.3}$$

Here $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}_{++}^2$ denotes an iterate, $\beta \in [0, 1]$ a search direction parameter, and $\mu = \boldsymbol{X} \bullet \boldsymbol{Y}/n$. It was shown in the recent paper [29] that the system (1.3) of equations above has the unique solution $(d\boldsymbol{X}, d\boldsymbol{Y})$ whenever $\boldsymbol{X} \succ \boldsymbol{O}$, $\boldsymbol{Y} \succ \boldsymbol{O}$ and $\boldsymbol{XY} + \boldsymbol{YX} \succeq \boldsymbol{O}$. See also Corollary 3.2 in section 3.

The current paper has two purposes. One is to propose a globally convergent Mizuno–Todd–Ye-type predictor-corrector infeasible-interior-point algorithm, with the use of the AHO search direction, for the monotone SDLCP. The other purpose is to demonstrate its quadratic convergence under the strict complementarity condition. Although we will describe the algorithm for the monotone SDLCP, we can easily apply it to the primal-dual pair of SDPs. See the papers [13, 14, 15, 29] for detailed relations between the primal-dual pair of SDPs and the monotone SDLCP.

In section 2, we present a globally convergent Mizuno–Todd–Ye-type predictor-corrector infeasible-interior-point algorithm using the AHO search direction for the SDLCP (1.1). Section 3 is devoted to fundamental lemmas which we will use in sections 4 and 5. We prove the global convergence of the algorithm in section 4 and derive its quadratic convergence under the strict complementarity condition in section 5. A proposition playing a key role in section 5 and its proof are based on the paper [27] by Potra and Sheng. In section 6, we will show further local convergence properties under an additional nondegeneracy assumption.

2. A predictor-corrector interior-point algorithm.

Throughout the paper we use the following notation:

$$\begin{aligned}
\zeta &: \text{ a constant not less than } 1/n, \\
\mathcal{F}_0 &= \{(\boldsymbol{U}', \boldsymbol{V}') - (\boldsymbol{U}, \boldsymbol{V}) \ : \ (\boldsymbol{U}', \boldsymbol{V}'),\ (\boldsymbol{U}, \boldsymbol{V}) \in \mathcal{F}\}, \\
\widetilde{\mathcal{N}}(\gamma, \tau) &= \left\{ (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}_+ \times \mathcal{S}_+ \ : \ \begin{array}{l} (\boldsymbol{XY} + \boldsymbol{YX})/2 \succeq (1-\gamma)\tau \boldsymbol{I}, \\ \boldsymbol{X} \bullet \boldsymbol{Y}/n \le (1 + \zeta\gamma)\tau \end{array} \right\} \\
& \quad \text{for each } \gamma \in [0,\ 1] \text{ and each } \tau \ge 0.
\end{aligned}$$

By construction, we see that

$$\begin{array}{l}
(1-\gamma)\tau \le \boldsymbol{X} \bullet \boldsymbol{Y}/n \\
\text{if } (\boldsymbol{X}, \boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma, \tau),\ \gamma \in [0,1], \text{ and } \tau \ge 0, \\
\widetilde{\mathcal{N}}(0, \tau) = \{(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}_+ \times \mathcal{S}_+ \ : \ \boldsymbol{XY} = \tau \boldsymbol{I}\} \subset \widetilde{\mathcal{N}}(\gamma, \tau) \subset \widetilde{\mathcal{N}}(\gamma', \tau) \\
\text{if } 0 < \gamma < \gamma' \le 1 \ \text{ and } \tau > 0.
\end{array}$$

Let $0 < \gamma < 1$. Then the set $\{(\boldsymbol{X}, \boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma, \tau) \ : \ \tau > 0\}$ forms a neighborhood of "the central manifold" $\{(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}_+ \times \mathcal{S}_+ \ : \ \boldsymbol{XY} = \tau \boldsymbol{I} \text{ for some } \tau > 0\}$. We call γ a *neighborhood parameter*. This set serves as the admissible region in which we confine iterates $(\boldsymbol{X}^k, \boldsymbol{Y}^k)$ $(k = 0, 1, 2, \dots)$ of Algorithm 2.1 described below. More precisely, starting from an $(\boldsymbol{X}^0, \boldsymbol{Y}^0, \theta^0, \gamma^0) = (\sqrt{\mu^0}\boldsymbol{I}, \sqrt{\mu^0}\boldsymbol{I}, 1, 0)$ for some $\mu^0 > 0$, Algorithm 2.1 generates a sequence $\{(\boldsymbol{X}^k, \boldsymbol{Y}^k, \boldsymbol{X}^k_c, \boldsymbol{Y}^k_c, \theta^k, \gamma^k)\}$ such that for every $k = 1, 2, \dots$,

$$1 \ge \theta^k \ge 0,\ \gamma > \gamma^k \ge 0, \tag{2.1}$$

$$1 = \theta^0 > \theta^k > \theta^{k+1}, \tag{2.2}$$

$$(\boldsymbol{X}^k, \boldsymbol{Y}^k) \in \widetilde{\mathcal{N}}(\gamma^k, \theta^k \mu^0),\ (\boldsymbol{X}^k, \boldsymbol{Y}^k) \in \mathcal{F} + \theta^k \left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}}) \right), \tag{2.3}$$

$$(\boldsymbol{X}^k_c, \boldsymbol{Y}^k_c) \in \widetilde{\mathcal{N}}(\gamma, \theta^{k+1} \mu^0),\ (\boldsymbol{X}^k_c, \boldsymbol{Y}^k_c) \in \mathcal{F} + \theta^{k+1} \left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}}) \right). \tag{2.4}$$

Here $(\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}})$ denotes an arbitrary pair of matrices in $\mathcal{F}$; in particular, we can take any solution of the SDLCP (1.1) for $(\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}})$ when the SDLCP (1.1) has a solution.

Note that

$$\begin{aligned} &\mathcal{F} + \theta\left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\boldsymbol{X}', \boldsymbol{Y}')\right) \\ &\quad = \mathcal{F} + \theta\left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\boldsymbol{X}, \boldsymbol{Y})\right) \\ &\qquad \text{for any } (\boldsymbol{X}', \boldsymbol{Y}') \in \mathcal{F},\ (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F} \ \text{ and } \theta \in [0, 1]. \end{aligned}$$

Among the iterates $\boldsymbol{X}^k,\ \boldsymbol{Y}^k,\ \boldsymbol{X}^k_c,\ \boldsymbol{Y}^k_c,\ \theta^k,\ \gamma^k$, the triplet $(\boldsymbol{X}^k, \boldsymbol{Y}^k, \theta^k)$ is updated to $(\boldsymbol{X}^k_c, \boldsymbol{Y}^k_c, \theta^{k+1})$ by the predictor step (Step 2), while the triplet $(\boldsymbol{X}^k_c, \boldsymbol{Y}^k_c, \gamma^k)$ is updated to $(\boldsymbol{X}^{k+1}, \boldsymbol{Y}^{k+1}, \gamma^{k+1})$ by the corrector step (Step 4). θ^{k+1} serves as a measure of both feasibility and optimality. Given an $\epsilon \geq 0$, the algorithm stops (at Step 3) when θ^{k+1} becomes equal to or smaller than ϵ. In this case, we have an approximate solution $(\boldsymbol{X}^k_c, \boldsymbol{Y}^k_c)$ of the SDLCP (1.1) such that

$$(2.5) \qquad \left\{ \begin{array}{l} \epsilon \geq \theta^{k+1} \geq 0, \\ \boldsymbol{X}^k_c \succeq \boldsymbol{O},\ \boldsymbol{Y}^k_c \succeq \boldsymbol{O},\ \boldsymbol{X}^k_c \bullet \boldsymbol{Y}^k_c / n \leq (1 + \zeta\gamma)\theta^{k+1}\mu^0, \\ (\boldsymbol{X}^k_c, \boldsymbol{Y}^k_c) \in \mathcal{F} + \theta^{k+1}\left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}})\right). \end{array} \right.$$

We call ϵ an *accuracy parameter.*

Before describing Algorithm 2.1, we introduce the hypothesis below. When the algorithm detects (at Step 1 or Step 3) that the hypothesis is false, it stops.

Hypothesis 2.1 (see [15]). Let $\omega^* \geq 1$. There exists a solution $(\boldsymbol{X}^*, \boldsymbol{Y}^*)$ of the SDLCP (1.1) such that

$$(2.6) \qquad \omega^* \boldsymbol{X}^0 \succeq \boldsymbol{X}^* \ \text{ and } \omega^* \boldsymbol{Y}^0 \succeq \boldsymbol{Y}^*.$$

ALGORITHM 2.1.

Step 0. Choose an accuracy parameter $\epsilon \geq 0$, a neighborhood parameter $\gamma \in (0,\ 1)$, and an initial point $(\boldsymbol{X}^0, \boldsymbol{Y}^0) = (\sqrt{\mu^0}\boldsymbol{I}, \sqrt{\mu^0}\boldsymbol{I})$ with some $\mu^0 > 0$. Let $\theta^0 = 1$, $\sigma = 2\omega^*/(1 - \gamma) + 1$, $\gamma^0 = 0$, and $k = 0$.

Step 1. If the inequality

$$(2.7) \qquad \theta^k(\boldsymbol{X}^0 \bullet \boldsymbol{Y}^k + \boldsymbol{X}^k \bullet \boldsymbol{Y}^0) \leq \sigma \boldsymbol{X}^k \bullet \boldsymbol{Y}^k$$

does not hold then stop.

Step 2 (predictor step). Compute a solution $(d\boldsymbol{X}^k_p, d\boldsymbol{Y}^k_p)$ of the system of equations

$$(2.8) \qquad \left. \begin{array}{l} \boldsymbol{X}^k d\boldsymbol{Y}^k_p + d\boldsymbol{Y}^k_p \boldsymbol{X}^k + d\boldsymbol{X}^k_p \boldsymbol{Y}^k + \boldsymbol{Y}^k d\boldsymbol{X}^k_p = -\boldsymbol{X}^k \boldsymbol{Y}^k - \boldsymbol{Y}^k \boldsymbol{X}^k, \\ (\boldsymbol{X}^k + d\boldsymbol{X}^k_p, \boldsymbol{Y}^k + d\boldsymbol{Y}^k_p) \in \mathcal{F}. \end{array} \right\}$$

Let

$$(2.9) \qquad \left\{ \begin{array}{lcl} \delta^k_p & = & \dfrac{\|d\boldsymbol{X}^k_p\|_F \|d\boldsymbol{Y}^k_p\|_F}{\theta^k \mu^0}, \\ \hat{\alpha}^k_p & = & \dfrac{2}{\sqrt{1 + 4\delta^k_p/(\gamma - \gamma^k)} + 1}, \\ \check{\alpha}^k_p & = & \max \left\{ \alpha' \in [0, 1] \ : \begin{array}{c} (\boldsymbol{X}^k + \alpha d\boldsymbol{X}^k_p, \boldsymbol{Y}^k + \alpha d\boldsymbol{Y}^k_p) \\ \in \widetilde{\mathcal{N}}(\gamma, (1 - \alpha)\theta^k \mu^0) \\ \text{for every } \alpha \in [0, \alpha'] \end{array} \right\}. \end{array} \right.$$

Choose a step length $\alpha_p^k \in [\hat{\alpha}_p^k, \check{\alpha}_p^k]$. Let

$$(\boldsymbol{X}_c^k, \boldsymbol{Y}_c^k) = (\boldsymbol{X}^k, \boldsymbol{Y}^k) + \alpha_p^k (d\boldsymbol{X}_p^k, d\boldsymbol{Y}_p^k) \quad \text{and } \theta^{k+1} = (1 - \alpha_p^k)\theta^k.$$

Step 3. If $\theta^{k+1} \leq \epsilon$ then stop. If the inequality

$$\theta^{k+1}(\boldsymbol{X}^0 \bullet \boldsymbol{Y}_c^k + \boldsymbol{X}_c^k \bullet \boldsymbol{Y}^0) \leq \sigma \boldsymbol{X}_c^k \bullet \boldsymbol{Y}_c^k \tag{2.10}$$

does not hold then stop.

Step 4 (corrector step). Compute a solution $(d\boldsymbol{X}_c^k, d\boldsymbol{Y}_c^k)$ of the system of equations

$$\left\{ \begin{array}{l} \boldsymbol{X}_c^k d\boldsymbol{Y}_c^k + d\boldsymbol{Y}_c^k \boldsymbol{X}_c^k + d\boldsymbol{X}_c^k \boldsymbol{Y}_c^k + \boldsymbol{Y}_c^k d\boldsymbol{X}_c^k \\ = 2\theta^{k+1}\mu^0 \boldsymbol{I} - \boldsymbol{X}_c^k \boldsymbol{Y}_c^k - \boldsymbol{Y}_c^k \boldsymbol{X}_c^k, \\ (d\boldsymbol{X}_c^k, d\boldsymbol{Y}_c^k) \in \mathcal{F}_0. \end{array} \right. \tag{2.11}$$

Let

$$\left\{ \begin{array}{lcl} \delta_c^k & = & \dfrac{\|d\boldsymbol{X}_c^k\|_F \|d\boldsymbol{Y}_c^k\|_F}{\theta^{k+1}\mu^0}, \\ \hat{\alpha}_c^k & = & \left\{ \begin{array}{ll} \gamma/(2\delta_c^k) & \text{if } \gamma \leq 2\delta_c^k, \\ 1 & \text{if } \gamma > 2\delta_c^k, \end{array} \right. \\ \check{\gamma}^{k+1} & = & \left\{ \begin{array}{ll} \gamma(1 - \gamma/(4\delta_c^k)) & \text{if } \gamma \leq 2\delta_c^k, \\ \delta_c^k & \text{if } \gamma > 2\delta_c^k, \end{array} \right. \\ \hat{\gamma}^{k+1} & = & \min \left\{ \gamma' \in [0,1] \ : \begin{array}{l} (\boldsymbol{X} + \alpha d\boldsymbol{X}^k, \boldsymbol{Y} + \alpha d\boldsymbol{Y}^k) \\ \in \widetilde{\mathcal{N}}(\gamma', \theta^{k+1}\mu^0), \\ \alpha \in [0,1] \end{array} \right\}. \end{array} \right. \tag{2.12}$$

Choose a step length $\alpha_c^k \in [0, 1]$ and γ^{k+1} such that

$$\left\{ \begin{array}{l} \hat{\gamma}^{k+1} \leq \gamma^{k+1} \leq \check{\gamma}^{k+1}, \\ (\boldsymbol{X}_c^k + \alpha_c^k d\boldsymbol{X}_c^k, \boldsymbol{Y}_c^k + \alpha_c^k d\boldsymbol{Y}_c^k) \in \widetilde{\mathcal{N}}(\gamma^{k+1}, \theta^{k+1}\mu^0). \end{array} \right. \tag{2.13}$$

(It will be shown in Lemma 3.8 that the pair of $\alpha_c^k = \hat{\alpha}_c^k$ and $\gamma^{k+1} = \check{\gamma}^{k+1}$ satisfies the relations above.) Let $(\boldsymbol{X}^{k+1}, \boldsymbol{Y}^{k+1}) = (\boldsymbol{X}_c^k, \boldsymbol{Y}_c^k) + \alpha_c^k (d\boldsymbol{X}_c^k, d\boldsymbol{Y}_c^k)$.

Step 5. Replace k by $k + 1$. Go to Step 1.

A distinctive feature of Algorithm 2.1 lies in the sophisticated step length control rule at Step 4 (the corrector step). When $\delta_c^k = \|d\boldsymbol{X}_c^k\|_F \|d\boldsymbol{Y}_c^k\|_F / (\theta^{k+1}\mu^0) < \gamma/2$, we can take the unit step length $\alpha_c^k = 1$ and a $\gamma^{k+1} \leq \gamma/2$; hence $(\boldsymbol{X}^{k+1}, \boldsymbol{Y}^{k+1}) \in \widetilde{\mathcal{N}}(\gamma/2, \theta^{k+1}\mu^0)$. When $\delta_c^k \geq \gamma/2$, however, we may not be able to take the unit step length, and as δ_c^k gets larger, we are forced to take a smaller step length α_c^k. On the other hand, the step length control rule at Step 2 (the predictor step) is based on and similar to the one used in the paper [27]. The theorem below summarizes the consistency and the global convergence of Algorithm 2.1. A proof of the theorem is given in section 4.

THEOREM 2.1 (global convergence theorem).

(i) *Algorithm* 2.1 *consistently generates a sequence* $\{(\boldsymbol{X}^k, \boldsymbol{Y}^k, \boldsymbol{X}_c^k, \boldsymbol{Y}_c^k, \theta^k, \gamma^k)\}$ *satisfying* (2.1), (2.2), (2.3), *and* (2.4).

(ii) *If Algorithm* 2.1 *stops at Step* 1 *violating the inequality* (2.7)*, then there is no solution of the SDLCP* (1.1) *satisfying* (2.6)*.*

(iii) *If Algorithm* 2.1 *stops at Step* 3 *with* $\theta^{k+1} \le \epsilon$*, then* $(\boldsymbol{X}_c^k, \boldsymbol{Y}_c^k)$ *gives an approximate solution of the SDLCP* (1.1) *satisfying* (2.5)*.*

(iv) *If Algorithm* 2.1 *stops at Step* 3 *violating the inequality* (2.10)*, then there is no solution of the SDLCP* (1.1) *satisfying* (2.6)*.*

(v) *If* $\epsilon > 0$*, Algorithm* 2.1 *stops in a finite number of iterations at either Step* 1 *or Step* 3*.*

3. Lemmas. In this section, we present a series of lemmas which we will utilize in proving both Theorem 2.1 (the global convergence theorem) in section 4 and Theorem 5.1 (the local convergence theorem) in section 5.

LEMMA 3.1. *Assume that* $(\boldsymbol{X}, \boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma, \tau)$ *for some* $\tau > 0$ *and* $\gamma \in (0, 1)$*.*

(i) $\|(\boldsymbol{XY} + \boldsymbol{YX})/2\|_F \le n(1 + \zeta\gamma)\tau$.

(ii) $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}_{++} \times \mathcal{S}_{++}$.

(iii) *Let* $(d\boldsymbol{X}, d\boldsymbol{Y})$ *be a solution of the system of equations*

$$\boldsymbol{X} d\boldsymbol{Y} + d\boldsymbol{Y}\boldsymbol{X} + d\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{Y} d\boldsymbol{X} = \boldsymbol{C} \quad \textit{and } (d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}_0, \tag{3.1}$$

where $\boldsymbol{C} \in \mathcal{S}$ *is a constant matrix. Then*

$$\|d\boldsymbol{X}\|_F \le \frac{\|\boldsymbol{X}\|_F \|\boldsymbol{C}\|_F}{((1-\gamma)\tau)} \quad \textit{and } \|d\boldsymbol{Y}\|_F \le \frac{\|\boldsymbol{Y}\|_F \|\boldsymbol{C}\|_F}{((1-\gamma)\tau)}.$$

Proof. (i) Let ν_j $(j = 1, 2, \dots, n)$ denote the eigenvalues of $(\boldsymbol{XY} + \boldsymbol{YX})/2$. Since the matrix $(\boldsymbol{XY} + \boldsymbol{YX})/2$ is symmetric and positive definite, we see that

$$\|(\boldsymbol{XY} + \boldsymbol{YX})/2\|_F = \left(\sum_{j=1}^{n} (\nu_j)^2 \right)^{1/2} \le \sum_{j=1}^{n} \nu_j = \boldsymbol{X} \bullet \boldsymbol{Y} \le n(1 + \zeta\gamma)\tau.$$

Here the last inequality follows from $(\boldsymbol{X}, \boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma, \tau)$.

(ii) By definition, we know that $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}_+ \times \mathcal{S}_+$. If $\boldsymbol{X}$ and/or $\boldsymbol{Y}$ were singular, we would have

$$\boldsymbol{u}^T(\boldsymbol{XY} + \boldsymbol{YX})\boldsymbol{u} = 0 \text{ for some nonzero } \boldsymbol{u} \in R^n,$$

which would contradict the assumption that $(\boldsymbol{X}, \boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma, \tau)$ with $\tau > 0$ and $\gamma \in (0, 1)$.

(iii) We will use the notation $\boldsymbol{A} \otimes \boldsymbol{B}$ for the Kronecker product of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, and the notation

$$\textbf{vec } \boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{\cdot 1} \\ \boldsymbol{A}_{\cdot 2} \\ \cdot \\ \boldsymbol{A}_{\cdot n} \end{pmatrix} \in R^{mn},$$

where $\boldsymbol{A}_{\cdot j}$ denotes the jth column of an $m \times n$ matrix $\boldsymbol{A}$. See, e.g., the book [6] for basic properties on the Kronecker product. Let

$$\boldsymbol{E} = (\boldsymbol{X} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{X}) \quad \text{and} \quad \boldsymbol{F} = (\boldsymbol{Y} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{Y}).$$

Then both $\boldsymbol{E}$ and $\boldsymbol{F}$ are symmetric and positive definite. Hence we can rewrite the system (3.1) of equations as

$$\textbf{vec } d\boldsymbol{Y} + \boldsymbol{E}^{-1}\boldsymbol{F}\boldsymbol{E}(\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X}) = \boldsymbol{E}^{-1}\textbf{vec } \boldsymbol{C} \text{ and } (d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}_0.$$

It follows that

$$\begin{aligned}
&\|\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X}\|\|\boldsymbol{C}\|_F \\
&\geq (\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X})^T\textbf{vec } \boldsymbol{C} \\
&\geq (\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X})^T\boldsymbol{F}\boldsymbol{E}(\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X}) \\
&\quad (\text{since } (\textbf{vec } d\boldsymbol{X})^T(\textbf{vec } d\boldsymbol{Y}) = d\boldsymbol{X} \bullet d\boldsymbol{Y} \geq 0 \text{ by the assumption (1.2)}) \\
&= (\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X})^T(\boldsymbol{Y} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{Y})(\boldsymbol{X} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{X})(\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X}) \\
&= (\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X})^T(\boldsymbol{Y}\boldsymbol{X} \otimes \boldsymbol{I} + \boldsymbol{X} \otimes \boldsymbol{Y} + \boldsymbol{X} \otimes \boldsymbol{Y} + \boldsymbol{I} \otimes \boldsymbol{Y}\boldsymbol{X})(\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X}) \\
&\geq (\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X})^T(\boldsymbol{Y}\boldsymbol{X} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{Y}\boldsymbol{X})(\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X}) \\
&\quad (\text{since both } \boldsymbol{Y} \otimes \boldsymbol{X} \text{ and } \boldsymbol{X} \otimes \boldsymbol{Y} \text{ are symmetric and positive definite}) \\
&= (\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X})^T \left(((\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{Y}\boldsymbol{X})/2) \otimes \boldsymbol{I} + \boldsymbol{I} \otimes ((\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{Y}\boldsymbol{X})/2)\right) (\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X}) \\
&\geq (\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X})^T \left(2(1-\gamma)\tau\boldsymbol{I} \otimes \boldsymbol{I}\right) (\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X}) \\
&\quad (\text{since } (\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{Y}\boldsymbol{X})/2 \succeq (1-\gamma)\tau\boldsymbol{I}) \\
&= 2(1-\gamma)\tau\|\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X}\|^2.
\end{aligned}$$

Therefore, we obtain that

$$\|\boldsymbol{C}\|_F \geq 2(1-\gamma)\tau\|\boldsymbol{E}^{-1}\textbf{vec } d\boldsymbol{X}\|.$$

On the other hand, if $\boldsymbol{W} \in \mathcal{S}$ is a solution of the system of equations $\boldsymbol{X}\boldsymbol{W} + \boldsymbol{W}\boldsymbol{X} = d\boldsymbol{X}$, then $\textbf{vec } \boldsymbol{W} = \boldsymbol{E}^{-1}(\textbf{vec } d\boldsymbol{X})$; hence

$$\begin{aligned}
\|d\boldsymbol{X}\|_F &= \|\boldsymbol{X}\boldsymbol{W} + \boldsymbol{W}\boldsymbol{X}\|_F \\
&\leq 2\|\boldsymbol{X}\|_F\|\boldsymbol{W}\|_F \\
&= 2\|\boldsymbol{X}\|_F\|\boldsymbol{E}^{-1}(\textbf{vec } d\boldsymbol{X})\| \\
&\leq \|\boldsymbol{X}\|_F\|\boldsymbol{C}\|_F/((1-\gamma)\tau).
\end{aligned}$$

We can prove similarly the inequality $\|d\boldsymbol{Y}\|_F \leq \|\boldsymbol{Y}\|_F\|\boldsymbol{C}\|_F/((1-\gamma)\tau)$. □

The corollary below ensures the existence and the uniqueness of the solution of the systems (2.8) and (2.11) of equations. This result was shown in the paper [29], but we give a proof of the corollary to make the current paper self-contained. The assumption in the corollary is slightly stronger than the one in Theorem 3.1 of the paper [29].

COROLLARY 3.2. *Assume that* $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}_+ \times \mathcal{S}_+$ *and* $\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{Y}\boldsymbol{X} \succ \boldsymbol{O}$. *For any* $\boldsymbol{C} \in \mathcal{S}$ *and any* $(\boldsymbol{X}', \boldsymbol{Y}') \in \mathcal{S} \times \mathcal{S}$, *there exists a unique solution* $(d\boldsymbol{X}, d\boldsymbol{Y})$ *of the system of equations*

$$(3.2) \qquad \boldsymbol{X}d\boldsymbol{Y} + d\boldsymbol{Y}\boldsymbol{X} + d\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{Y}d\boldsymbol{X} = \boldsymbol{C} \quad \textit{and } (d\boldsymbol{X} + \boldsymbol{X}', d\boldsymbol{Y} + \boldsymbol{Y}') \in \mathcal{F}.$$

Proof. First we take $\tau > 0$ and $\gamma \in (0,1)$ such that $(\boldsymbol{X}, \boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma, \tau)$. Let $m = n(n+1)/2$. For every $\boldsymbol{C} \in \mathcal{S}$ and $(\boldsymbol{X}', \boldsymbol{Y}') \in \mathcal{S} \times \mathcal{S}$, define

$$\begin{aligned}
\mathcal{G}(\boldsymbol{C}) &= \{(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{S} \times \mathcal{S} \ : \ \boldsymbol{X}d\boldsymbol{Y} + d\boldsymbol{Y}\boldsymbol{X} + d\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{Y}d\boldsymbol{X} = \boldsymbol{C}\}, \\
\mathcal{F}(\boldsymbol{X}', \boldsymbol{Y}') &= \mathcal{F} - (\boldsymbol{X}', \boldsymbol{Y}').
\end{aligned}$$

Since $\boldsymbol{X} \in \mathcal{S}$ is positive definite by (ii) of Lemma 3.1, for every $d\boldsymbol{X} \in \mathcal{S}$ there exists a unique $d\boldsymbol{Y} \in \mathcal{S}$ such that $(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{G}(\boldsymbol{O})$. This implies that $\mathcal{G}(\boldsymbol{O})$ forms an m-dimensional linear subspace of the $2m$-dimensional linear space $\mathcal{S} \times \mathcal{S}$. Take an $(\boldsymbol{X}'', \boldsymbol{Y}'') \in \mathcal{F}$. Then we see that $\mathcal{F}(\boldsymbol{X}'', \boldsymbol{Y}'') = \mathcal{F}_0$; hence $\mathcal{F}(\boldsymbol{X}'', \boldsymbol{Y}'')$ forms an m-dimensional linear subspace of the $2m$-dimensional linear space $\mathcal{S} \times \mathcal{S}$. Applying (iii) of Lemma 3.1 with $\boldsymbol{C} = \boldsymbol{O}$, we know that the m-dimensional linear subspaces $\mathcal{G}(\boldsymbol{O})$ and $\mathcal{F}(\boldsymbol{X}'', \boldsymbol{Y}'') = \mathcal{F}_0$ of the $2m$-dimensional linear space $\mathcal{S} \times \mathcal{S}$ transversally intersect at the single point $(\boldsymbol{O}, \boldsymbol{O})$. Therefore, for any $\boldsymbol{C} \in \mathcal{S}$ and $(\boldsymbol{X}', \boldsymbol{Y}') \in \mathcal{S} \times \mathcal{S}$, their parallel translations $\mathcal{G}(\boldsymbol{C})$ and $\mathcal{F}(\boldsymbol{X}', \boldsymbol{Y}')$ transversely intersect at a single point, and the desired result follows. □

LEMMA 3.3. *Suppose that Hypothesis* 2.1 *is true. Let* $\gamma \in (0,1)$, $\mu^0 > 0$, $(\boldsymbol{X}^0, \boldsymbol{Y}^0) = (\sqrt{\mu^0}\boldsymbol{I}, \sqrt{\mu^0}\boldsymbol{I})$ *and* $\sigma = 2\omega^*/(1-\gamma) + 1$. *If* $(\boldsymbol{X}, \boldsymbol{Y}, \theta)$ *satisfies*

$$1 \geq \theta \geq 0,\ (\boldsymbol{X}, \boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma, \theta\mu^0), \qquad (3.3)$$
$$(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F} + \theta\left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}})\right) \quad \textit{for some } (\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}}) \in \mathcal{F},$$

then the inequality

$$\theta\left(\boldsymbol{X}^0 \bullet \boldsymbol{Y} + \boldsymbol{X} \bullet \boldsymbol{Y}^0\right) \leq \sigma \boldsymbol{X} \bullet \boldsymbol{Y} \qquad (3.4)$$

holds.

Proof. This lemma follows directly from Lemma 7.6 of [15]. □

LEMMA 3.4. *Let* $\gamma \in (0,1)$, $\mu^0 > 0$, $(\boldsymbol{X}^0, \boldsymbol{Y}^0) = (\sqrt{\mu^0}\boldsymbol{I}, \sqrt{\mu^0}\boldsymbol{I})$ *and* $\sigma > 0$. *Suppose that* (3.3) *and* (3.4) *hold. Then*

$$\|\boldsymbol{X}\|_F \leq n\sigma(1+\zeta\gamma)\sqrt{\mu^0} \quad \textit{and } \|\boldsymbol{Y}\|_F \leq n\sigma(1+\zeta\gamma)\sqrt{\mu^0} \qquad (3.5)$$

hold.

Proof. We see from (3.3) and (3.4) that

$$\begin{aligned}\theta\sqrt{\mu^0}\left(\|\boldsymbol{Y}\|_F + \|\boldsymbol{X}\|_F\right) &\leq \theta\sqrt{\mu^0}\left(\mathrm{Tr}\ \boldsymbol{Y} + \mathrm{Tr}\ \boldsymbol{X}\right)\\ &= \theta(\boldsymbol{X}^0 \bullet \boldsymbol{Y} + \boldsymbol{X} \bullet \boldsymbol{Y}^0)\\ &\leq \sigma \boldsymbol{X} \bullet \boldsymbol{Y}\\ &\leq n\sigma(1+\zeta\gamma)\theta\mu^0.\end{aligned}$$

Thus (3.5) follows. □

LEMMA 3.5. *Let* $\gamma \in (0,1)$, $\mu^0 > 0$, $(\boldsymbol{X}^0, \boldsymbol{Y}^0) = (\sqrt{\mu^0}\boldsymbol{I}, \sqrt{\mu^0}\boldsymbol{I})$, *and* $\sigma > 0$. *Define*

$$\kappa_c = 2\sigma(1+\zeta\gamma)(2+\zeta\gamma)\sqrt{\mu^0}/(1-\gamma). \qquad (3.6)$$

If $(\boldsymbol{X}_c^k, \boldsymbol{Y}_c^k, \theta^{k+1})$ *satisfies* (2.1), (2.4), *and* (2.10), *then the solution* $(d\boldsymbol{X}_c^k, d\boldsymbol{Y}_c^k)$ *of the system* (2.11) *of equations satisfies that* $\|d\boldsymbol{X}_c^k\|_F \leq n^2\kappa_c$ *and* $\|d\boldsymbol{Y}_c^k\|_F \leq n^2\kappa_c$.

Proof. By applying (iii) of Lemma 3.1 with $\tau = \theta^{r+1}\mu^0$ and $\boldsymbol{C} = 2\theta^{k+1}\mu^0\boldsymbol{I} - \boldsymbol{X}_c^k\boldsymbol{Y}_c^k - \boldsymbol{Y}_c^k\boldsymbol{X}_c^k$, we have that

$$\begin{aligned}\|d\boldsymbol{X}_c^k\|_F &\leq \|\boldsymbol{X}_c^k\|_F\|2\theta^{k+1}\mu^0\boldsymbol{I} - \boldsymbol{X}_c^k\boldsymbol{Y}_c^k - \boldsymbol{Y}_c^k\boldsymbol{X}_c^k\|_F/((1-\gamma)\theta^{k+1}\mu^0)\\ &\leq \|\boldsymbol{X}_c^k\|_F\left(2\theta^{k+1}\mu^0\sqrt{n} + \|\boldsymbol{X}_c^k\boldsymbol{Y}_c^k + \boldsymbol{Y}_c^k\boldsymbol{X}_c^k\|_F\right)/((1-\gamma)\theta^{k+1}\mu^0).\end{aligned}$$

By (i) of Lemma 3.1 and Lemma 3.4, we also know that

$$\|\boldsymbol{X}_c^k\boldsymbol{Y}_c^k + \boldsymbol{Y}_c^k\boldsymbol{X}_c^k\|_F \leq 2n(1+\zeta\gamma)\theta^{k+1}\mu^0 \ \text{ and } \|\boldsymbol{X}_c^k\|_F \leq n\sigma(1+\zeta\gamma)\sqrt{\mu^0}.$$

It follows that

$$\begin{aligned}\|d\boldsymbol{X}_c^k\|_F &\leq \left(n\sigma(1+\zeta\gamma)\sqrt{\mu^0}\right)\left(2\theta^{k+1}\mu^0\sqrt{n}+n(1+\zeta\gamma)\theta^{k+1}\mu^0\right)/((1-\gamma)\theta^{k+1}\mu^0)\\ &\leq n^2\kappa_c.\end{aligned}$$

We can similarly prove that $\|d\boldsymbol{Y}_c^k\|_F \leq n^2\kappa_c$. □

LEMMA 3.6. *Let* $\gamma \in (0,1)$, $\mu^0 > 0$, $(\boldsymbol{X}^0, \boldsymbol{Y}^0) = (\sqrt{\mu^0}\boldsymbol{I}, \sqrt{\mu^0}\boldsymbol{I})$, $(\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}}) \in \mathcal{F}$, *and* $\sigma > 0$. *Define*

$$(3.7)\qquad \left\{\begin{array}{l}\varphi = 2(1+\zeta\gamma)\max\left\{\mu^0,\ \sigma\sqrt{\mu^0}\left(\|\boldsymbol{X}^0-\overline{\boldsymbol{X}}\|_F+\|\boldsymbol{Y}^0-\overline{\boldsymbol{Y}}\|_F\right)\right\},\\ \kappa_p = \dfrac{2\sigma(1+\zeta\gamma)\varphi}{(1-\gamma)\sqrt{\mu^0}}+\max\left\{\|\boldsymbol{X}^0-\overline{\boldsymbol{X}}\|_F,\ \|\boldsymbol{Y}^0-\overline{\boldsymbol{Y}}\|_F\right\}.\end{array}\right.$$

Suppose that $(\boldsymbol{X}^k, \boldsymbol{Y}^k, \theta^k)$ *satisfies* (2.2), (2.3), *and* (2.7). *Define*

$$(3.8)\qquad \left\{\begin{array}{lcl}\boldsymbol{K}_1^k &=& -\boldsymbol{X}^k\boldsymbol{Y}^k-\boldsymbol{Y}^k\boldsymbol{X}^k,\\ \boldsymbol{K}_2^k &=& -\theta^k\Big(\boldsymbol{X}^k(\boldsymbol{Y}^0-\overline{\boldsymbol{Y}})+(\boldsymbol{Y}^0-\overline{\boldsymbol{Y}})\boldsymbol{X}^k\\ && \qquad +(\boldsymbol{X}^0-\overline{\boldsymbol{X}})\boldsymbol{Y}^k+\boldsymbol{Y}^k(\boldsymbol{X}^0-\overline{\boldsymbol{X}})\Big).\end{array}\right.$$

Let $(d\boldsymbol{X}_p^k, d\boldsymbol{Y}_p^k)$ *be the solution of the system* (2.8) *of equations. For each* $j = 1, 2$, *let* $(\boldsymbol{U}_j^k, \boldsymbol{V}_j^k)$ *be the solution of the system*

$$(3.9)\qquad \boldsymbol{X}^k\boldsymbol{V}_j^k+\boldsymbol{V}_j^k\boldsymbol{X}^k+\boldsymbol{U}_j^k\boldsymbol{Y}^k+\boldsymbol{Y}^k\boldsymbol{U}_j^k = \boldsymbol{K}_j^k \quad \textit{and } (\boldsymbol{U}_j^k, \boldsymbol{V}_j^k) \in \mathcal{F}_0.$$

(i) $\|\boldsymbol{K}_j^k\|_F \leq n\varphi\theta^k$ $(j = 1, 2)$.

(ii) $(d\boldsymbol{X}_p^k, d\boldsymbol{Y}_p^k) = \displaystyle\sum_{j=1}^{2}(\boldsymbol{U}_j^k, \boldsymbol{V}_j^k)+\theta^k\left((\boldsymbol{X}^0, \boldsymbol{Y}^0)-(\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}})\right)$.

(iii) $\|d\boldsymbol{X}_p^k\|_F \leq n^2\kappa_p$ *and* $\|d\boldsymbol{Y}_p^k\|_F \leq n^2\kappa_p$.

Proof. (i) By (i) of Lemma 3.1 and Lemma 3.4, we see that

$$\begin{aligned}\|\boldsymbol{K}_1^k\|_F &= \|\boldsymbol{X}^k\boldsymbol{Y}^k+\boldsymbol{Y}^k\boldsymbol{X}^k\|_F \leq 2n(1+\zeta\gamma)\theta^k\mu^0 \leq n\varphi\theta^k,\\ \|\boldsymbol{K}_2^k\|_F &= \theta^k\|\boldsymbol{X}^k(\boldsymbol{Y}^0-\overline{\boldsymbol{Y}})+(\boldsymbol{Y}^0-\overline{\boldsymbol{Y}})\boldsymbol{X}^k+(\boldsymbol{X}^0-\overline{\boldsymbol{X}})\boldsymbol{Y}^k+\boldsymbol{Y}^k(\boldsymbol{X}^0-\overline{\boldsymbol{X}})\|_F\\ &\leq 2\theta^k\|\boldsymbol{X}^k(\boldsymbol{Y}^0-\overline{\boldsymbol{Y}})+(\boldsymbol{X}^0-\overline{\boldsymbol{X}})\boldsymbol{Y}^k\|_F\\ &\leq 2\theta^k(n\sigma(1+\zeta\gamma)\sqrt{\mu^0})\left(\|\boldsymbol{Y}^0-\overline{\boldsymbol{Y}}\|_F+\|\boldsymbol{X}^0-\overline{\boldsymbol{X}}\|_F\right)\\ &\leq n\varphi\theta^k.\end{aligned}$$

(ii) Let $(\boldsymbol{U}, \boldsymbol{V})$ denote the right-hand side of the identity to be proved. It is easily verified that $(\boldsymbol{U}, \boldsymbol{V})$ is a solution of the system (2.8) of equations. Since $(d\boldsymbol{X}_p^k, d\boldsymbol{Y}_p^k)$ is the unique solution of the system (2.8) of equations, we obtain the desired identity.

(iii) By (iii) of Lemma 3.1, Lemma 3.4, and the assertion (i) above, we see that

$$\begin{aligned}\|\boldsymbol{U}_j^k\| &\leq \|\boldsymbol{X}^k\|_F\|\boldsymbol{K}_j^k\|_F/((1-\gamma)\theta^k\mu^0)\\ &\leq \left(n\sigma(1+\zeta\gamma)\sqrt{\mu^0}\right)\left(n\varphi\theta^k\right)/((1-\gamma)\theta^k\mu^0)\\ &= n^2\sigma(1+\zeta\gamma)\varphi/\left((1-\gamma)\sqrt{\mu^0}\right) \quad \text{for } j = 1, 2.\end{aligned}$$

Hence we see by the assertion (ii) above that

$$\|d\boldsymbol{X}_p^k\|_F \le \|\boldsymbol{U}_1^k\|_F + \|\boldsymbol{U}_2^k\|_F + \theta^k\|\boldsymbol{X}^0 - \overline{\boldsymbol{X}}\|_F \le n^2\kappa_p.$$

We can similarly prove that $\|d\boldsymbol{Y}_p^k\|_F \le n^2\kappa_p$. ∎

LEMMA 3.7. *Let $\gamma \in (0,1)$, $\mu^0 > 0$, $(\boldsymbol{X}^0, \boldsymbol{Y}^0) = (\sqrt{\mu^0}\boldsymbol{I}, \sqrt{\mu^0}\boldsymbol{I})$. Suppose that $(\boldsymbol{X}^k, \boldsymbol{Y}^k, \theta^k, \gamma^k)$ satisfies* (2.1) *and* (2.3). *Let $(d\boldsymbol{X}_p^k, d\boldsymbol{Y}_p^k)$ be the solution of the system* (2.8) *of equations. Define δ_p^k, $\hat{\alpha}_p^k$ and $\check{\alpha}_p^k$ by* (2.9). *Then $0 < \hat{\alpha}_p^k \le \check{\alpha}_p^k \le 1$.*

Proof. By definition, we know that $0 < \hat{\alpha}_p^k \le 1$ and $0 \le \check{\alpha}_p^k \le 1$. Hence it suffices to show that

$$(3.10)\quad (\boldsymbol{X}^k + \alpha d\boldsymbol{X}_p^k, \boldsymbol{Y}^k + \alpha d\boldsymbol{Y}_p^k) \in \widetilde{\mathcal{N}}(\gamma, (1-\alpha)\theta^k\mu^0) \text{ for every } \alpha \in [0, \hat{\alpha}_p^k].$$

Assume that $0 \le \alpha \le \hat{\alpha}_p^k$. Then

(3.11)

$$\begin{aligned}
&\frac{(\boldsymbol{X}^k + \alpha d\boldsymbol{X}_p^k)(\boldsymbol{Y}^k + \alpha d\boldsymbol{Y}_p^k) + (\boldsymbol{Y}^k + \alpha d\boldsymbol{Y}_p^k)(\boldsymbol{X}^k + \alpha d\boldsymbol{X}_p^k)}{2}\\
&\quad -(1-\gamma)(1-\alpha)\theta^k\mu^0\boldsymbol{I}\\
&= \frac{(1-\alpha)(\boldsymbol{X}^k\boldsymbol{Y}^k + \boldsymbol{Y}^k\boldsymbol{X}^k) + \alpha^2(d\boldsymbol{X}_p^k d\boldsymbol{Y}_p^k + d\boldsymbol{Y}_p^k d\boldsymbol{X}_p^k)}{2}\\
&\quad -(1-\gamma)(1-\alpha)\theta^k\mu^0\boldsymbol{I}\\
&\succeq (1-\alpha)(1-\gamma^k)\theta^k\mu^0\boldsymbol{I} - \alpha^2\|d\boldsymbol{X}_p^k\|_F\|d\boldsymbol{Y}_p^k\|_F\boldsymbol{I}\\
&\quad -(1-\gamma)(1-\alpha)\theta^k\mu^0\boldsymbol{I}\\
&= \left(-\alpha^2\delta_p^k + (1-\alpha)(\gamma-\gamma^k)\right)\theta^k\mu^0\boldsymbol{I}.
\end{aligned}$$

On the other hand, it follows from the definition of $\hat{\alpha}_p^k$ in (2.9) that

$$\hat{\alpha}_p^k = \frac{-(\gamma-\gamma^k) + \sqrt{(\gamma-\gamma^k)^2 + 4\delta_p^k(\gamma-\gamma^k)}}{2\delta_p^k},$$

which coincides with a positive root of the quadratic polynomial

$$\left(-\alpha^2\delta_p^k + (1-\alpha)(\gamma-\gamma^k)\right)$$

in (3.11). Therefore,

$$\begin{aligned}
&\frac{(\boldsymbol{X}^k + \alpha d\boldsymbol{X}_p^k)(\boldsymbol{Y}^k + \alpha d\boldsymbol{Y}_p^k) + (\boldsymbol{Y}^k + \alpha d\boldsymbol{Y}_p^k)(\boldsymbol{X}^k + \alpha d\boldsymbol{X}_p^k)}{2}\\
&\qquad -(1-\gamma)(1-\alpha)\theta^k\mu^0\boldsymbol{I} \succeq \boldsymbol{O}\\
&\qquad\qquad \text{for every } \alpha \in [0, \hat{\alpha}_p^k],
\end{aligned}$$

which, together with $(\boldsymbol{X}^k, \boldsymbol{Y}^k) \in \mathcal{S}_{++} \times \mathcal{S}_{++}$, implies that

$$(\boldsymbol{X}^k + \alpha d\boldsymbol{X}_p^k, \boldsymbol{Y}^k + \alpha d\boldsymbol{Y}_p^k) \in \mathcal{S}_+ \times \mathcal{S}_+ \text{ for every } \alpha \in [0, \hat{\alpha}_p^k].$$

We also see that for every $\alpha \in [0,1]$,

$$(1+\zeta\gamma)(1-\alpha)\theta^k\mu^p - \frac{(\boldsymbol{X}^k + \alpha d\boldsymbol{X}_p^k) \bullet (\boldsymbol{Y}^k + \alpha d\boldsymbol{Y}_p^k)}{n}$$

$$\begin{aligned}
&= (1+\zeta\gamma)(1-\alpha)\theta^k\mu^0 - \frac{(1-\alpha)\boldsymbol{X}^k \bullet \boldsymbol{Y}^k}{n} + \frac{\alpha^2 d\boldsymbol{X}_p^k \bullet d\boldsymbol{Y}_p^k}{n} \\
&\geq \left((1+\zeta\gamma)(1-\alpha) - (1-\alpha)(1+\zeta\gamma^k) - \alpha^2\delta_p^k/n\right)\theta^k\mu^0 \\
&= \left(-\alpha^2\delta_p^k/n + \zeta(1-\alpha)(\gamma-\gamma^k)\right)\theta^k\mu^0 \\
&\geq \left(-\alpha^2\delta_p^k + (1-\alpha)(\gamma-\gamma^k)\right)\zeta\theta^k\mu^0 \text{ (since } \zeta \geq 1/n).
\end{aligned}$$

Note that the coefficient of $\zeta\theta^k\mu^0$ is the same quadratic polynomial as the one that appears in (3.11) above. Hence

$$(1+\zeta\gamma)(1-\alpha)\theta^k\mu^p - \frac{(\boldsymbol{X}^k + \alpha d\boldsymbol{X}_p^k) \bullet (\boldsymbol{Y}^k + \alpha d\boldsymbol{Y}_p^k)}{n} \geq 0 \text{ for every } \alpha \in [0, \hat{\alpha}_p^k].$$

Thus we have shown the desired relation (3.10). □

LEMMA 3.8. *Let $\gamma \in (0,1)$, $\mu^0 > 0$, $(\boldsymbol{X}^0, \boldsymbol{Y}^0) = (\sqrt{\mu^0}\boldsymbol{I}, \sqrt{\mu^0}\boldsymbol{I})$, and $\sigma > 0$. Suppose that $(\boldsymbol{X}_c^k, \boldsymbol{Y}_c^k)$ satisfies (2.4) for some $\theta^{k+1} \in [0,1]$. Let $(d\boldsymbol{X}_c^k, d\boldsymbol{Y}_c^k)$ be the solution of the system (2.11) of equations. Define δ_c^k, $\hat{\alpha}_c^k$, $\check{\gamma}^{k+1}$ and $\hat{\gamma}^k$ by (2.12). Then $0 \leq \hat{\gamma}^{k+1} \leq \check{\gamma}^{k+1} < \gamma$, and the pair of $\alpha_c^k = \hat{\alpha}_c^k$ and $\gamma^{k+1} = \check{\gamma}^{k+1}$ satisfies (2.13).*

Proof. We first observe that for every $\alpha \in [0,1]$,

$$\begin{aligned}
&\frac{(\boldsymbol{X}_c^k + \alpha d\boldsymbol{X}_c^k)(\boldsymbol{Y}_c^k + \alpha d\boldsymbol{Y}_c^k) + (\boldsymbol{Y}_c^k + \alpha d\boldsymbol{Y}_c^k)(\boldsymbol{X}_c^k + \alpha d\boldsymbol{X}_c^k)}{2} \\
&= \frac{(1-\alpha)(\boldsymbol{X}_c^k\boldsymbol{Y}_c^k + \boldsymbol{Y}_c^k\boldsymbol{X}_c^k) + 2\alpha\theta^{k+1}\mu^0\boldsymbol{I} + \alpha^2(d\boldsymbol{X}_c^k d\boldsymbol{Y}_c^k + d\boldsymbol{Y}_c^k d\boldsymbol{X}_c^k)}{2} \\
&\succeq (1-\alpha)(1-\gamma)\theta^{k+1}\mu^0\boldsymbol{I} + \alpha\theta^{k+1}\mu^0\boldsymbol{I} - \alpha^2\|d\boldsymbol{X}_c^k\|_F\|d\boldsymbol{Y}_c^k\|_F\boldsymbol{I} \\
&= \left(-\alpha^2\delta_c^k + \gamma\alpha + (1-\gamma)\right)\theta^{k+1}\mu^0\boldsymbol{I}.
\end{aligned}$$

It follows that

$$\begin{aligned}
&\frac{(\boldsymbol{X}_c^k + \alpha d\boldsymbol{X}_c^k)(\boldsymbol{Y}_c^k + \alpha d\boldsymbol{Y}_c^k) + (\boldsymbol{Y}_c^k + \alpha d\boldsymbol{Y}_c^k)(\boldsymbol{X}_c^k + \alpha d\boldsymbol{X}_c^k)}{2} \succeq \boldsymbol{O} \text{ for every } \alpha \in [0, \hat{\alpha}_c^k], \\
&\frac{(\boldsymbol{X}_c^k + \hat{\alpha}_c^k d\boldsymbol{X}_c^k)(\boldsymbol{Y}_c^k + \hat{\alpha}_c^k d\boldsymbol{Y}_c^k) + (\boldsymbol{Y}_c^k + \hat{\alpha}_c^k d\boldsymbol{Y}_c^k)(\boldsymbol{X}_c^k + \hat{\alpha}_c^k d\boldsymbol{X}_c^k)}{2} \succeq (1-\check{\gamma}^{k+1})\theta^{k+1}\mu^0\boldsymbol{I}.
\end{aligned}$$

We also see that for every $\alpha \in [0,1]$,

$$\begin{aligned}
&\frac{(\boldsymbol{X}_c^k + \alpha d\boldsymbol{X}_c^k) \bullet (\boldsymbol{Y}_c^k + \alpha d\boldsymbol{Y}_c^k)}{n} \\
&= (1-\alpha)\frac{\boldsymbol{X}_c^k \bullet \boldsymbol{Y}_c^k}{n} + \alpha\theta^{k+1}\mu^0 + \alpha^2\frac{d\boldsymbol{X}_c^k \bullet d\boldsymbol{Y}_c^k}{n} \\
&\leq \left((1-\alpha)(1+\zeta\gamma) + \alpha + \alpha^2\delta_c^k/n\right)\theta^{k+1}\mu^0 \\
&= \left(\alpha^2\delta_c^k/n - \zeta\gamma\alpha + (1+\zeta\gamma)\right)\theta^{k+1}\mu^0 \\
&\leq \left(\zeta\alpha^2\delta_c^k - \zeta\gamma\alpha + (1+\zeta\gamma)\right)\theta^{k+1}\mu^0 \text{ (since } \zeta \geq 1/n).
\end{aligned}$$

It follows that

$$\frac{(\boldsymbol{X}_c^k + \hat{\alpha}_c^k d\boldsymbol{X}_c^k) \bullet (\boldsymbol{Y}_c^k + \hat{\alpha}_c^k d\boldsymbol{Y}_c^k)}{n} \leq (1+\zeta\check{\gamma}^{k+1})\theta^{k+1}\mu^0.$$

Thus we have shown that $(\boldsymbol{X}_c^k + \hat{\alpha}_c^k d\boldsymbol{X}_c^k, \boldsymbol{Y}_c^k + \hat{\alpha}_c^k d\boldsymbol{Y}_c^k) \in \widetilde{\mathcal{N}}(\check{\gamma}^{k+1}, \theta^{k+1}\mu^0)$, and the desired result follows. □

4. Proof of Theorem 2.3.

4.1. Proof of (i), (ii), (iii), and (iv) of Theorem 2.3. By definition, $(\boldsymbol{X}_c^0, \boldsymbol{Y}_c^0, \theta^0, \gamma^0)$ satisfies (2.1), (2.3), and $\theta^0 = 1$. Let $q \geq 0$. Assume that we have computed $(\boldsymbol{X}_c^q, \boldsymbol{Y}_c^q, \theta^q, \gamma^q)$ satisfying (2.1) and (2.3) for $k = q$, we will investigate each step of the algorithm.

Step 1. Applying Lemma 3.3, we know that if the inequality (2.7) with $\theta = \theta^q$ and $(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{X}^q, \boldsymbol{Y}^q)$ at Step 1 does not hold then there is no solution of the SDLCP (1.1) satisfying (2.6). Hence we have shown (ii) of Theorem 2.1.

Step 2 (predictor step). Now suppose that the inequality (2.7) holds for $k = q$. The validity of Step 2 follows from Corollary 3.2 and Lemma 3.7. Corollary 3.2 ensures the existence and the uniqueness of the solution $(d\boldsymbol{X}_p^q, d\boldsymbol{Y}_p^q)$ of the system (2.8) of equations with $k = q$. In view of Lemma 3.7, we can consistently take a step length α_p^q satisfying

$$\begin{aligned}
&0 < \hat{\alpha}_p^q \leq \alpha_p^q \leq \check{\alpha}_p^q \leq 1, \\
&0 \leq \theta^{q+1} = (1 - \alpha_p^q)\theta^q < \theta^q, \\
&(\boldsymbol{X}_c^q, \boldsymbol{Y}_c^q) = (\boldsymbol{X}^q, \boldsymbol{Y}^q) + \alpha_p^q (d\boldsymbol{X}_p^q, d\boldsymbol{Y}_p^q) \in \widetilde{\mathcal{N}}(\gamma, \theta^{k+1}\mu^0).
\end{aligned}$$

Hence (2.2) and the first relation of (2.4) with $k = q$ follow. To derive the second relation of (2.4) with $k = q$, we observe that

$$\begin{aligned}
(\boldsymbol{X}_c^q, \boldsymbol{Y}_c^q) &= (\boldsymbol{X}^q, \boldsymbol{Y}^q) + \alpha_p^q (d\boldsymbol{X}_p^q, d\boldsymbol{Y}_p^q) \\
&= (1 - \alpha_p^q)(\boldsymbol{X}^q, \boldsymbol{Y}^q) + \alpha_p^q (\boldsymbol{X}^q + d\boldsymbol{X}_p^q, \boldsymbol{Y}^q + d\boldsymbol{Y}_p^q) \\
&\in (1 - \alpha_p^q)\left(\mathcal{F} + \theta^q \left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}})\right)\right) + \alpha_p^q \mathcal{F} \\
&\quad \text{(since (2.3) and (2.8) hold for } k = q) \\
&= \mathcal{F} + (1 - \alpha_p^q)\theta^k \left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}})\right) \\
&= \mathcal{F} + \theta^{k+1} \left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}})\right).
\end{aligned}$$

Thus we have shown the latter relation of (2.4) with $k = q$.

Step 3. If $\theta^{q+1} \leq \epsilon$, then $(\boldsymbol{X}_p^q, \boldsymbol{Y}_p^q)$ satisfies (2.5) for $k = q$; hence (iii) of Theorem 2.3 holds. If the inequality (2.10) with $k = q$ does not hold, then we have the same conclusion as in Step 1 by applying Lemma 3.3 with $\theta = \theta^{q+1}$ and $(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{X}_c^q, \boldsymbol{Y}_c^q)$. Hence we have shown (iv) of Theorem 2.1.

Step 4 (corrector step). The validity of this step follows from Corollary 3.2 and Lemma 3.8. Corollary 3.2 ensures the existence and the uniqueness of the solution $(d\boldsymbol{X}_c^q, d\boldsymbol{Y}_c^q)$ of the system (2.11) of equations with $k = q$. By Lemma 3.8, the first relation of (2.3) holds for $k = q + 1$, and we can consistently take a step length α_c^q and a γ^{q+1} satisfying (2.13) with $k = q$. Hence the second relation of (2.1) holds for $q = k + 1$. We also see by (2.4) and (2.11) with $k = q$ that

$$\begin{aligned}
(\boldsymbol{X}^{q+1}, \boldsymbol{Y}^{q+1}) &= (\boldsymbol{X}_c^q, \boldsymbol{Y}_c^q) + \alpha_c^q (d\boldsymbol{X}_c^q, d\boldsymbol{Y}_c^q) \\
&\in \mathcal{F} + \theta^{q+1} \left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}})\right) + \mathcal{F}_0 \\
&= \mathcal{F} + \theta^{q+1} \left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}})\right).
\end{aligned}$$

Therefore the latter relation of (2.3) holds for $k = q + 1$.

We have shown (2.1) and (2.3) with $k = q+1$ to proceed to the $(q+1)$th iteration consistently, and we have proved (i) of Theorem 2.1.

4.2. Proof of (v) of Theorem 2.3. Now we assume that $\epsilon > 0$. Then we have $\theta^k \geq \epsilon$, the inequalities (2.7) and (2.10) while Algorithm 2.1 is running. We see by Lemmas 3.5 and 3.6 that

$$0 \leq \delta_p^k \leq n^4\kappa_p^2/(\epsilon\mu^0) \text{ and } 0 \leq \delta_c^k \leq n^4\kappa_c^2/(\epsilon\mu^0) \text{ for every } k = 0, 1, \dots .$$

Hence, defining

$$\left\{ \begin{array}{ll} \bar{\gamma} & = \max\{\gamma(1 - \gamma\epsilon\mu^0/(4n^4\kappa_c^2)), \gamma/2\}, \\ \bar{\alpha}_p = & \dfrac{2}{\sqrt{1 + 4n^4\kappa_p^2/\left(\epsilon\mu^0(\gamma - \bar{\gamma})\right)} + 1}, \end{array} \right. \tag{4.1}$$

we obtain that

$$0 \leq \gamma^k \leq \check{\gamma}^k \leq \bar{\gamma} < \gamma \text{ and } 0 < \bar{\alpha}_p \leq \hat{\alpha}_p^k \leq \alpha_p^k \text{ for every } k = 0, 1, 2, \dots ;$$

hence

$$\epsilon \leq \theta^k = \theta^0 \prod_{j=1}^{k-1} (1 - \alpha_p^j) \leq \theta^0 (1 - \bar{\alpha}_p)^{(k-1)} \text{ for every } k = 0, 1, 2, \dots . \tag{4.2}$$

Therefore Algorithm 2.1 must stop in a finite number of iterations. This completes the proof of Theorem 2.1.

Remark 4.1. If we regard κ_p, κ_c, γ, and μ^0 as constant in (4.1), then we can take a positive constant δ such that $\delta\epsilon/n^4 \leq \bar{\alpha}_p$ for every sufficiently large n. In this case we can derive from (4.2) that Algorithm 2.1 stops in $O((n^4/\epsilon)\log(1/\epsilon))$ iterations; hence Algorithm 2.1 works as a fully polynomial-time approximation scheme [25].

5. Local convergence. In the remainder of the paper, we assume Hypothesis 2.1 and discuss the local convergence of the sequence generated by Algorithm 2.1 with $\epsilon = 0$. Hypothesis 2.1 ensures that (ii) and (iv) of Theorem 2.1 cannot occur. Since $\epsilon = 0$, if Algorithm 2.1 stops in a finite number of iterations (i.e., the sequence is finite), then we obtain an exact solution $(\boldsymbol{X}_c^{k+1}, \boldsymbol{Y}_c^{k+1})$ of the SDLCP (1.1) at Step 3 of the last iteration. Assuming that the sequence is infinite, we will establish the following.

THEOREM 5.1 (local convergence theorem). *Assume that Hypothesis* 2.1 *and Condition* 5.1 *below hold. Let* $\{(\boldsymbol{X}^k, \boldsymbol{Y}^k, \boldsymbol{X}_c^k, \boldsymbol{Y}_c^k, \theta^k, \gamma^k)\}$ *be the sequence generated by Algorithm* 2.1 *with* $\epsilon = 0$.

(i) *The* $\hat{\alpha}_p^k$ *defined in* (2.9) *satisfies that* $\hat{\alpha}_p^k \to 1$ *as* $k \to \infty$.

(ii) *There is a positive constant* η *such that* $\theta^{k+1} \leq \eta(\theta^k)^2$ *for every* $k = 0, 1, 2, \dots$, *i.e., the optimality and feasibility measure* θ^k *converges to zero quadratically.*

Condition 5.1 (strict complementarity). There is a solution $(\boldsymbol{X}^*, \boldsymbol{Y}^*)$ of the SDLCP (1.1) such that $\boldsymbol{X}^* + \boldsymbol{Y}^* \succ \boldsymbol{O}$.

Before proving Theorem 5.1, we introduce some notation. We often use the following notation for a matrix $\boldsymbol{B}$ depending on a parameter δ:

$\boldsymbol{B} = \boldsymbol{\Theta}(\delta^\beta)$ if $\boldsymbol{B}$ is a symmetric matrix (or a number), and if $\xi_2\delta^\beta\boldsymbol{I} \succeq \boldsymbol{B} \succeq \xi_1\delta^\beta\boldsymbol{I}$ for some $\xi_1 > 0$, some $\xi_2 \geq \xi_1$, and any small $\delta > 0$,

$\boldsymbol{B} = \boldsymbol{O}(\delta^\beta)$ if $\|\boldsymbol{B}\|_F \leq \xi\delta^\beta$ for some $\xi \geq 0$ and any small $\delta > 0$,

$\boldsymbol{B} = \boldsymbol{o}(\delta^\beta)$ if $\|\boldsymbol{B}\|_F/\delta^\beta \to 0$ as $\delta \to 0$.

Since $\boldsymbol{X}^*$ and $\boldsymbol{Y}^*$ commute, there exists an orthogonal matrix $\boldsymbol{Q}$ such that

$$\boldsymbol{Q}^\top \boldsymbol{X}^* \boldsymbol{Q} = \begin{pmatrix} \boldsymbol{\Lambda}_B & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{pmatrix}, \boldsymbol{Q}^\top \boldsymbol{Y}^* \boldsymbol{Q} = \begin{pmatrix} \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{\Lambda}_N \end{pmatrix},$$

where $\boldsymbol{\Lambda}_B$ and $\boldsymbol{\Lambda}_N$ are positive diagonal matrices with dimensions m and $n-m$ for some $m \in \{0,1,2,\ldots,n\}$, respectively. For each $(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S} \times \mathcal{S}$, define

$$\boldsymbol{Q}^\top \boldsymbol{X} \boldsymbol{Q} \equiv \hat{\boldsymbol{X}} = \begin{pmatrix} \hat{\boldsymbol{X}}_B & \hat{\boldsymbol{X}}_J \\ \hat{\boldsymbol{X}}_J^\top & \hat{\boldsymbol{X}}_N \end{pmatrix} \quad \text{and} \quad \boldsymbol{Q}^\top \boldsymbol{Y} \boldsymbol{Q} \equiv \hat{\boldsymbol{Y}} = \begin{pmatrix} \hat{\boldsymbol{Y}}_B & \hat{\boldsymbol{Y}}_J \\ \hat{\boldsymbol{Y}}_J^\top & \hat{\boldsymbol{Y}}_N \end{pmatrix}. \tag{5.1}$$

Define an affine subspace of $\mathcal{S} \times \mathcal{S}$ which contains the solutions of the SDLCP:

$$\widetilde{\mathcal{M}} \equiv \left\{ (\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F} \cap (\mathcal{S} \times \mathcal{S}) : \begin{array}{l} \hat{\boldsymbol{X}} = \begin{pmatrix} \boldsymbol{M}_B & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{pmatrix}, \hat{\boldsymbol{Y}} = \begin{pmatrix} \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{M}_N \end{pmatrix}, \\ \boldsymbol{M}_B \text{ is an } m \times m \text{ symmetric matrix and} \\ \boldsymbol{M}_N \text{ is an } (n-m) \times (n-m) \text{ symmetric matrix} \end{array} \right\}.$$

To simplify the argument, we here assume the following Proposition 5.2, which we will prove later.

PROPOSITION 5.2. *Under Condition* 5.1*, for every* $(\boldsymbol{X},\boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma,\tau)$*, we have*

$$\left\| (\boldsymbol{X} - \check{\boldsymbol{X}}, \boldsymbol{Y} - \check{\boldsymbol{Y}}) \right\|_F = \boldsymbol{O}(\tau), \tag{5.2}$$

where $(\check{\boldsymbol{X}},\check{\boldsymbol{Y}})$ *is the solution of the minimization problem:*

$$\min\{\|(\boldsymbol{X}-\boldsymbol{X}',\boldsymbol{Y}-\boldsymbol{Y}')\|_F : (\boldsymbol{X}',\boldsymbol{Y}') \in \widetilde{\mathcal{M}}\}. \tag{5.3}$$

Proof of Theorem 5.1. By (iii) of Lemma 3.1, the definition of δ_c^k, and the fact that $2\theta^k \boldsymbol{I} - (\boldsymbol{X}^k\boldsymbol{Y}^k + \boldsymbol{Y}^k\boldsymbol{X}^k) = \boldsymbol{O}(\theta^k)$, we have $\delta_c^k = \boldsymbol{O}(1)$; i.e., there exists a constant C such that $\delta_c \leq C$ for all iterates. This implies the existence of a positive constant $\bar{\gamma}$ such that

$$0 \leq \gamma^k \leq \bar{\gamma} < \gamma \quad \text{for every } k = 0,1,2,\ldots. \tag{5.4}$$

Let $\Delta\boldsymbol{X}^k = d\boldsymbol{X}_p^k + (\boldsymbol{X}^k - \check{\boldsymbol{X}}^k)$ and $\Delta\boldsymbol{Y}^k = d\boldsymbol{Y}_p^k + (\boldsymbol{Y}^k - \check{\boldsymbol{Y}}^k)$. Note that $(\Delta\boldsymbol{X}^k, \Delta\boldsymbol{Y}^k)$ is a solution of (3.1) with $\boldsymbol{X} = \boldsymbol{X}^k$, $\boldsymbol{Y} = \boldsymbol{Y}^k$, and $\boldsymbol{C} = (\boldsymbol{X}^k - \check{\boldsymbol{X}}^k)(\boldsymbol{Y}^k - \check{\boldsymbol{Y}}^k) + (\boldsymbol{Y}^k - \check{\boldsymbol{Y}}^k)(\boldsymbol{X}^k - \check{\boldsymbol{X}}^k)$. By (5.2), we have $\|\boldsymbol{C}\|_F = \boldsymbol{O}((\theta^k)^2)$. Hence, by (iii) of Lemma 3.1, we have $\|\Delta\boldsymbol{X}^k\|_F = \boldsymbol{O}(\theta^k)$ and $\|\Delta\boldsymbol{Y}^k\|_F = \boldsymbol{O}(\theta^k)$. Therefore, using (5.2) again, we also have $\|d\boldsymbol{X}_p^k\|_F = \boldsymbol{O}(\theta^k)$ and $\|d\boldsymbol{Y}_p^k\|_F = \boldsymbol{O}(\theta^k)$. Hence we have

$$0 \leq \delta_p^k = \frac{\|d\boldsymbol{X}_p^k\|_F \|d\boldsymbol{Y}_p^k\|_F}{\theta^k \mu^0} \leq \boldsymbol{O}(\theta^k) \quad \text{for every } k = 0,1,2,\ldots. \tag{5.5}$$

Specifically, $\delta_p^k \to 0$ as $k \to \infty$. Hence $\hat{\alpha}_p^k \to 1$ as $k \to \infty$. This implies the assertion (i). Now, using the inequalities (5.4) and (5.5), we see that for every $k = 0,1,2,\ldots$,

$$\begin{array}{rcl} 1 - \alpha_p^k & \leq & 1 - \hat{\alpha}_p^k \\ & = & 1 - \dfrac{2}{\sqrt{1 + 4\delta_p^k/(\gamma - \gamma^k)} + 1} \\ & = & \dfrac{4\delta_p^k/(\gamma - \gamma^k)}{(\sqrt{1 + 4\delta_p^k/(\gamma - \gamma^k)} + 1)^2} \\ & \leq & \delta_p^k/(\gamma - \gamma^k) \\ & \leq & \delta_p^k/(\gamma - \bar{\gamma}) \\ & = & \boldsymbol{O}(\theta^k). \end{array}$$

Finally we obtain by the construction of θ^{k+1} that

$$\theta^{k+1} = (1 - \alpha_p^k)\theta^k = \boldsymbol{O}((\theta^k)^2).$$

Thus we have shown the assertion (ii). □

The remainder of this section is devoted to proving Proposition 5.2 by a series of Lemmas. The essential idea of the proof is based on section 4 of the paper [27] by Potra and Sheng.

LEMMA 5.3. *Suppose that* $(\boldsymbol{X}, \boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma, \tau)$. *Then, under Condition* 5.1*, we have*

$$\left\{ \begin{array}{l} \hat{\boldsymbol{X}} = \boldsymbol{Q}^\top \boldsymbol{X} \boldsymbol{Q} = \begin{pmatrix} \boldsymbol{\Theta}(1) & \boldsymbol{O}(\sqrt{\tau}) \\ \boldsymbol{O}(\sqrt{\tau}) & \boldsymbol{O}(\tau) \end{pmatrix}, \\ \hat{\boldsymbol{Y}} = \boldsymbol{Q}^\top \boldsymbol{Y} \boldsymbol{Q} = \begin{pmatrix} \boldsymbol{O}(\tau) & \boldsymbol{O}(\sqrt{\tau}) \\ \boldsymbol{O}(\sqrt{\tau}) & \boldsymbol{\Theta}(1) \end{pmatrix}. \end{array} \right. \tag{5.6}$$

Proof. By Lemma 2.2 of [27], we have

$$\lambda_{min}(\boldsymbol{X}^{\frac{1}{2}} \boldsymbol{Y} \boldsymbol{X}^{\frac{1}{2}}) \geq \frac{1}{2} \lambda_{min}(\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{Y}\boldsymbol{X}),$$

where $\lambda_{min}(\boldsymbol{A})$ is the smallest eigenvalue of $\boldsymbol{A}$. Hence we have $\boldsymbol{X}^{\frac{1}{2}} \boldsymbol{Y} \boldsymbol{X}^{\frac{1}{2}} \succeq (1-\gamma)\tau \boldsymbol{I}$. Using this fact, we can easily follow the proofs of Lemmas 4.4 and 4.6 of [27] to derive (5.6). □

Let $\omega = \max\{\|\hat{\boldsymbol{X}}_J\|_F, \|\hat{\boldsymbol{Y}}_J\|_F, \tau\}$.

LEMMA 5.4. *Suppose that* $(\boldsymbol{X}, \boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma, \tau)$. *Under Condition* 5.1*, we have*

$$\|(\boldsymbol{X} - \check{\boldsymbol{X}}, \boldsymbol{Y} - \check{\boldsymbol{Y}})\|_F = \boldsymbol{O}(\omega), \tag{5.7}$$

where $(\check{\boldsymbol{X}}, \check{\boldsymbol{Y}})$ *is the solution of the minimization problem* (5.3).

Proof. Suppose that (5.7) does not hold; i.e., there exists a convergent sequence $(\boldsymbol{X}_k, \boldsymbol{Y}_k) \in \widetilde{\mathcal{N}}(\gamma, \tau)$ such that

$$\xi_k = \|(\boldsymbol{X}_k - \check{\boldsymbol{X}}_k, \boldsymbol{Y}_k - \check{\boldsymbol{Y}}_k)\|_F / \omega_k \to \infty, \omega_k \to 0 \text{ as } k \to \infty. \tag{5.8}$$

Let us define

$$(\Delta \boldsymbol{X}_k, \Delta \boldsymbol{Y}_k) = \frac{1}{\omega_k \xi_k} (\boldsymbol{X}_k - \check{\boldsymbol{X}}_k, \boldsymbol{Y}_k - \check{\boldsymbol{Y}}_k). \tag{5.9}$$

Since $\|(\Delta \boldsymbol{X}_k, \Delta \boldsymbol{Y}_k)\|_F = 1$, taking a convergent subsequence, we may assume that

$$(\Delta \boldsymbol{X}_k, \Delta \boldsymbol{Y}_k) \to (\Delta \boldsymbol{X}', \Delta \boldsymbol{Y}').$$

Then we have that $(\Delta \boldsymbol{X}', \Delta \boldsymbol{Y}') \in \mathcal{F}_0$. Let $\boldsymbol{q}_i$ be the ith column vector of $\boldsymbol{Q}^*$ $(1 \leq i \leq n)$. From (5.6), we have for any $i \in B$,

$$\begin{aligned} |\boldsymbol{q}_i^\top (\Delta \boldsymbol{Y}_k) \boldsymbol{q}_i| = |\boldsymbol{q}_i^\top \boldsymbol{Y}_k \boldsymbol{q}_i| / (\xi_k \omega_k) &\leq \|(\hat{\boldsymbol{Y}}_k)_B\|_F / (\xi_k \omega_k) \\ &\leq \boldsymbol{O}(\tau_k) / (\xi_k \omega_k) = \boldsymbol{O}(1/\xi_k) = \boldsymbol{o}(1), \end{aligned}$$

which implies $\boldsymbol{q}_i^\top \Delta \boldsymbol{Y}' \boldsymbol{q}_i = 0$ for each $i \in B$. Similarly, $\boldsymbol{q}_i^\top \Delta \boldsymbol{X}' \boldsymbol{q}_i = 0$ for each $i \in N$. For each pair i, j where $i \in B, j \in N$, or $i \in N, j \in B$, we have

$$|\boldsymbol{q}_i^\top (\Delta \boldsymbol{Y}_k) \boldsymbol{q}_j| = |\boldsymbol{q}_i^\top \boldsymbol{Y}_k \boldsymbol{q}_j| / (\xi_k \omega_k) \leq \|(\hat{\boldsymbol{Y}}_k)_J\|_F / (\omega_k \xi_k) \leq 1/\xi_k = \boldsymbol{o}(1),$$

which implies $\boldsymbol{q}_i^\top \Delta \boldsymbol{Y}' q_j = 0$. Similarly, we have $\boldsymbol{q}_i^\top \Delta \boldsymbol{X}' \boldsymbol{q}_j = 0$ for any pair i, j described above. Hence,

$$(\check{\boldsymbol{X}}_k, \check{\boldsymbol{Y}}_k) + \nu(\Delta \boldsymbol{X}', \Delta \boldsymbol{Y}') \in \widetilde{\mathcal{M}} \text{ for all } \nu \in R,$$

and

$$\begin{array}{cl} & \dfrac{\| \left(\boldsymbol{X}_k - (\check{\boldsymbol{X}}_k + \omega_k \xi_k \Delta \boldsymbol{X}'), \boldsymbol{Y}_k - (\check{\boldsymbol{Y}}_k + \omega_k \xi_k \Delta \boldsymbol{Y}') \right) \|_F}{\|(\boldsymbol{X}_k - \check{\boldsymbol{X}}_k, \boldsymbol{Y}_k - \check{\boldsymbol{Y}}_k)\|_F} \\ = & \dfrac{1}{\omega_k \xi_k} \| \left(\boldsymbol{X}_k - (\check{\boldsymbol{X}}_k + \omega_k \xi_k \Delta \boldsymbol{X}'), \boldsymbol{Y}_k - (\check{\boldsymbol{Y}}_k + \omega_k \xi_k \Delta \boldsymbol{Y}') \right) \|_F \\ = & \|(\Delta \boldsymbol{X}_k - \Delta \boldsymbol{X}', \Delta \boldsymbol{Y}_k - \Delta \boldsymbol{Y}')\|_F \\ \to & 0 \quad \text{as } k \to \infty. \end{array}$$

Therefore,

$$\begin{array}{l} \left(\boldsymbol{X}_k - (\check{\boldsymbol{X}}_k + \omega_k \xi_k \Delta \boldsymbol{X}'), \boldsymbol{Y}_k - (\check{\boldsymbol{Y}}_k + \omega_k \xi_k \Delta \boldsymbol{Y}') \right) \in \widetilde{\mathcal{M}} \text{ and} \\ \| \left(\boldsymbol{X}_k - (\check{\boldsymbol{X}}_k + \omega_k \xi_k \Delta \boldsymbol{X}'), \boldsymbol{Y}_k - (\check{\boldsymbol{Y}}_k + \omega_k \xi_k \Delta \boldsymbol{Y}') \right) \|_F < \|(\boldsymbol{X}_k - \check{\boldsymbol{X}}_k, \boldsymbol{Y}_k - \check{\boldsymbol{Y}}_k)\|_F \end{array}$$

hold for every sufficiently large k. This contradicts the assumption that $(\check{\boldsymbol{X}}_k, \check{\boldsymbol{Y}}_k)$ is the solution of the minimization problem (5.3). □

LEMMA 5.5. *Suppose that $(\boldsymbol{X}, \boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma, \tau)$. Under Condition* 5.1*, we have*

$$\hat{\boldsymbol{X}}_J = \boldsymbol{O}(\tau), \ \hat{\boldsymbol{Y}}_J = \boldsymbol{O}(\tau). \tag{5.10}$$

Proof. Suppose that (5.10) does not hold. We may assume there exists a subsequence $(\boldsymbol{X}_k, \boldsymbol{Y}_k) \in \widetilde{\mathcal{N}}(\gamma, \tau_k)$ such that

$$\chi_k = \frac{\|(\hat{\boldsymbol{X}}_k)_J\|_F}{\tau_k} \to \infty, \quad \tau_k \to 0, \quad \text{as } k \to \infty. \tag{5.11}$$

Since $(\boldsymbol{X}_k, \boldsymbol{Y}_k) \in \widetilde{\mathcal{N}}(\gamma, \tau_k)$, we have

$$\boldsymbol{X}_k \boldsymbol{Y}_k + \boldsymbol{Y}_k \boldsymbol{X}_k = O(\tau_k) = \boldsymbol{o}(\|(\hat{\boldsymbol{X}}_k)_J\|_F). \tag{5.12}$$

This yields

$$(\hat{\boldsymbol{X}}_k)_B (\hat{\boldsymbol{Y}}_k)_J + (\hat{\boldsymbol{X}}_k)_J (\hat{\boldsymbol{Y}}_k)_N = o(\|(\hat{\boldsymbol{X}}_k)_J\|_F). \tag{5.13}$$

Hence

$$(\hat{\boldsymbol{Y}}_k)_J = -(\hat{\boldsymbol{X}}_k)_B^{-1} (\hat{\boldsymbol{X}}_k)_J (\hat{\boldsymbol{Y}}_k)_N + (\hat{\boldsymbol{X}}_k)_B^{-1} o(\|(\hat{\boldsymbol{X}}_k)_J\|_F). \tag{5.14}$$

Since $(\hat{\boldsymbol{X}}_k)_B = \boldsymbol{\Theta}(1), \ (\hat{\boldsymbol{Y}}_k)_N = \boldsymbol{\Theta}(1)$ from Lemma 5.3, we have that

$$\|(\hat{\boldsymbol{Y}}_k)_J\|_F = \boldsymbol{\Theta}(\|(\hat{\boldsymbol{X}}_k)_J\|_F). \tag{5.15}$$

Take a convergent subsequence $(\boldsymbol{X}_k, \boldsymbol{Y}_k)$ such that

$$(\boldsymbol{X}_k, \boldsymbol{Y}_k) \to (\boldsymbol{X}^\infty, \boldsymbol{Y}^\infty) \text{ and } \left(\frac{(\hat{\boldsymbol{X}}_k)_J}{\|(\hat{\boldsymbol{X}}_k)_J\|_F}, \frac{(\hat{\boldsymbol{Y}}_k)_J}{\|(\hat{\boldsymbol{X}}_k)_J\|_F} \right) \to (\boldsymbol{X}'_J, \boldsymbol{Y}'_J)$$

as $k \to \infty$. By (5.13) and letting $k \to \infty$, we get

$$\hat{\boldsymbol{X}}_B^\infty \boldsymbol{Y}'_J + \boldsymbol{X}'_J \hat{\boldsymbol{Y}}_N^\infty = \boldsymbol{O}.$$

By Lemma 5.3, $\hat{\boldsymbol{X}}_B^\infty, \hat{\boldsymbol{Y}}_N^\infty$ are positive definite matrices. Note that $\|\boldsymbol{X}'_J\|_F = 1$. Hence,

$$\begin{array}{rcl} \boldsymbol{X}'_J \bullet \boldsymbol{Y}'_J & = & -\boldsymbol{X}'_J \bullet [(\boldsymbol{X}_B^\infty)^{-1} \boldsymbol{X}'_J \boldsymbol{Y}_N^\infty] \\ & = & -\mathrm{Tr}\ ([(\boldsymbol{X}_B^\infty)^{-\frac{1}{2}} \boldsymbol{X}'_J (\boldsymbol{Y}_N^\infty)^{\frac{1}{2}}][(\boldsymbol{X}_B^\infty)^{-\frac{1}{2}} \boldsymbol{X}'_J (\boldsymbol{Y}_N^\infty)^{\frac{1}{2}}]^\top) \\ & < & 0. \end{array} \tag{5.16}$$

Let $(\check{\boldsymbol{X}}_k, \check{\boldsymbol{Y}}_k) \in \widetilde{\mathcal{M}}$ be defined as in Lemma 5.4 and let

$$\boldsymbol{X}'' = \boldsymbol{X}_k - \check{\boldsymbol{X}}_k + \theta(\boldsymbol{X}^\infty - \boldsymbol{X}^0), \quad \boldsymbol{Y}'' = \boldsymbol{Y}_k - \check{\boldsymbol{Y}}_k + \theta(\boldsymbol{Y}^\infty - \boldsymbol{X}^0).$$

It is easily seen that $(\boldsymbol{X}'', \boldsymbol{Y}'') \in \mathcal{F}^0$, and therefore $\boldsymbol{X}'' \bullet \boldsymbol{Y}'' \geq 0$. By Lemma 5.4, we have

$$(\boldsymbol{X}_k - \check{\boldsymbol{X}}_k) \bullet (\boldsymbol{Y}_k - \check{\boldsymbol{Y}}_k) = \boldsymbol{O}(\omega_k \tau_k),$$

i.e.,

(5.17)

$$((\hat{\boldsymbol{X}}_k)_B - (\hat{\check{\boldsymbol{X}}}_k)_B) \bullet (\hat{\boldsymbol{Y}}_k)_B + 2(\hat{\boldsymbol{X}}_k)_J \bullet (\hat{\boldsymbol{Y}}_k)_J + (\hat{\boldsymbol{X}}_k)_N \bullet ((\hat{\boldsymbol{Y}}_k)_N - (\hat{\check{\boldsymbol{Y}}}_k)_N) = \boldsymbol{O}(\omega_k \tau_k).$$

From Lemmas 5.3 and 5.4, we have that

$$(\hat{\boldsymbol{Y}}_k)_B = \boldsymbol{O}(\tau_k), (\hat{\boldsymbol{X}}_k)_N = \boldsymbol{O}(\tau_k), \tag{5.18}$$

$$(\hat{\boldsymbol{X}}_k)_B - (\hat{\check{\boldsymbol{X}}}_k)_B = \boldsymbol{O}(\omega_k), (\hat{\boldsymbol{Y}}_k)_N - (\hat{\check{\boldsymbol{Y}}}_k)_N = \boldsymbol{O}(\omega_k). \tag{5.19}$$

Therefore,

$$(\hat{\boldsymbol{X}}_k)_J \bullet (\hat{\boldsymbol{Y}}_k)_J = O(\omega_k \tau_k), \tag{5.20}$$

and

$$\begin{array}{rcl} \dfrac{\omega_k \tau_k}{\|(\hat{\boldsymbol{X}}_k)_J\|_F^2} & = & \dfrac{\tau_k}{\|(\hat{\boldsymbol{X}}_k)_J\|_F^2} \max\{\|(\hat{\boldsymbol{X}}_k)_J\|_F, \|(\hat{\boldsymbol{Y}}_k)_J\|_F, \tau_k\} \\ & = & \dfrac{\tau_k}{\|(\hat{\boldsymbol{X}}_k)_J\|_F} \max\left\{1, \dfrac{\|(\hat{\boldsymbol{Y}}_k)_J\|_F}{\|(\hat{\boldsymbol{X}}_k)_J\|_F}, \dfrac{\tau_k}{\|(\hat{\boldsymbol{X}}_k)_J\|_F}\right\} \\ & = & o(1). \end{array} \tag{5.21}$$

Dividing both sides of (5.20) by $\|(\hat{\boldsymbol{X}}_k)_J\|_F^2$, recalling (5.21), and letting $k \to \infty$, we obtain

$$\boldsymbol{X}'_J \bullet \boldsymbol{Y}'_J = 0,$$

which contradicts (5.16). □

From Lemma 5.5, we get $\omega_k = \boldsymbol{\Theta}(\tau_k)$. Therefore, together with Lemma 5.4, we conclude Proposition 5.2, i.e.,

$$\|(\boldsymbol{X} - \check{\boldsymbol{X}}, \boldsymbol{Y} - \check{\boldsymbol{Y}})\|_F = \boldsymbol{O}(\tau).$$

6. Local convergence with nondegeneracy.

Throughout this section, we assume Hypothesis 2.1, Condition 5.1, and the following.

Condition 6.1 (nondegeneracy). $(\boldsymbol{U},\boldsymbol{V}) = (\boldsymbol{O},\boldsymbol{O})$ if $\boldsymbol{X}^*\boldsymbol{V} + \boldsymbol{U}\boldsymbol{Y}^* = \boldsymbol{O}$ and $(\boldsymbol{U},\boldsymbol{V}) \in \mathcal{F}_0$.

Under these assumptions, the solution $(\boldsymbol{X}^*,\boldsymbol{Y}^*)$ of the SDLCP (1.1) ensured by Hypothesis 2.1 is the unique one. See section 4 of [14]. Hence the sequence $\{(\boldsymbol{X}^k,\boldsymbol{Y}^k,\boldsymbol{X}^k_c,\boldsymbol{Y}^k_c,\theta^k,\gamma^k)\}$ generated by Algorithm 2.1 with $\epsilon = 0$ satisfies

$$(\boldsymbol{X}^k,\boldsymbol{Y}^k) \to (\boldsymbol{X}^*,\boldsymbol{Y}^*) \text{ and } (\boldsymbol{X}^k_c,\boldsymbol{Y}^k_c) \to (\boldsymbol{X}^*,\boldsymbol{Y}^*) \text{ as } k \to \infty \tag{6.1}$$

in addition to (2.1), (2.2), (2.3), and (2.4).

Remark. Under the strict complementarity, the nondegeneracy is equivalent to the combination of the primal and the dual nondegeneracy conditions given in the paper [3]. This fact was due to Haeberly [7].

Assuming that the sequence is infinite, we establish the following theorem.

THEOREM 6.1. *Assume that Hypothesis* 2.1 *and Condition* 6.1 *hold. Let* $\{(\boldsymbol{X}^k,\boldsymbol{Y}^k,\boldsymbol{X}^k_c,\boldsymbol{Y}^k_c,\theta^k,\gamma^k)\}$ *be the sequence generated by Algorithm* 2.1 *with* $\epsilon = 0$.

(i) *The* $\hat{\alpha}^k_c$ *defined in* (2.12) *satisfies that* $\hat{\alpha}^k_c = 1$ *for every sufficiently large* k.

(ii) *The* $\hat{\gamma}^{k+1}$ *defined in* (2.12) *satisfies that* $\hat{\gamma}^{k+1} \to 0$ *as* $k \to \infty$.

The assertions (i) and (ii) of the theorem imply the following.

(i)′ For every sufficiently large k, we can take the unit step length $\alpha^k_c = 1$ at the corrector step.

(ii)′ The sequence $\{(\boldsymbol{X}^k,\boldsymbol{Y}^k)\}$ converges to the solution $(\boldsymbol{X}^*,\boldsymbol{Y}^*)$ tangentially to the central manifold in the sense that

$$\|(\boldsymbol{X}^k\boldsymbol{Y}^k + \boldsymbol{Y}^k\boldsymbol{X}^k)/2 - \left(\boldsymbol{X}^k \bullet \boldsymbol{Y}^k/n\right)\boldsymbol{I}\|_F/(\boldsymbol{X}^k \bullet \boldsymbol{Y}^k/n) \to 0 \text{ as } k \to \infty.$$

We need some lemmas to prove the theorem.

LEMMA 6.2. *Assume that*

$$\boldsymbol{X}^*\boldsymbol{V} + \boldsymbol{V}\boldsymbol{X}^* + \boldsymbol{U}\boldsymbol{Y}^* + \boldsymbol{Y}^*\boldsymbol{U} = \boldsymbol{O} \text{ and } (\boldsymbol{U},\boldsymbol{V}) \in \mathcal{F}_0.$$

Then $(\boldsymbol{U},\boldsymbol{V}) = (\boldsymbol{O},\boldsymbol{O})$.

Proof. Since $\boldsymbol{X}^* \in \mathcal{S}_+$ and $\boldsymbol{Y}^* \in \mathcal{S}_+$ are commutative, we can take an orthogonal matrix $\boldsymbol{P}$ that diagonalizes $\boldsymbol{X}^*$ and $\boldsymbol{Y}^*$, simultaneously:

$$\boldsymbol{P}^T\boldsymbol{X}^*\boldsymbol{P} = \boldsymbol{\Gamma} \text{ and } \boldsymbol{P}^T\boldsymbol{Y}^*\boldsymbol{P} = \boldsymbol{\Delta}$$

for some $n \times n$ diagonal matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Delta}$. Since $0 = \boldsymbol{X}^* \bullet \boldsymbol{Y}^* = \boldsymbol{\Gamma} \bullet \boldsymbol{\Delta}$ and $\boldsymbol{\Gamma} + \boldsymbol{\Delta} = \boldsymbol{P}^T(\boldsymbol{X}^* + \boldsymbol{Y}^*)\boldsymbol{P} \succ \boldsymbol{O}$, we may assume without loss of generality that the diagonal matrix $\boldsymbol{\Gamma}$ and $\boldsymbol{\Delta}$ have the forms

$$\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{\Gamma}_{11} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{pmatrix} \text{ and } \boldsymbol{\Delta} = \begin{pmatrix} \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{\Delta}_{22} \end{pmatrix},$$

respectively. Here $\boldsymbol{\Gamma}_{11}$ is an $m \times m$ positive diagonal matrix, $\boldsymbol{\Delta}_{22}$ an $(n-m)\times(n-m)$ positive diagonal matrix, and $0 \le m \le n$. Let

$$\boldsymbol{U}' = \boldsymbol{P}^T\boldsymbol{U}\boldsymbol{P} = \begin{pmatrix} \boldsymbol{U}'_{11} & \boldsymbol{U}'_{12} \\ (\boldsymbol{U}'_{12})^T & \boldsymbol{U}'_{22} \end{pmatrix} \text{ and } \boldsymbol{V}' = \boldsymbol{P}^T\boldsymbol{V}\boldsymbol{P} = \begin{pmatrix} \boldsymbol{V}'_{11} & \boldsymbol{V}'_{12} \\ (\boldsymbol{V}'_{12})^T & \boldsymbol{V}'_{22} \end{pmatrix}.$$

Then we have from the assumption that

$$(6.2) \qquad \begin{cases} \boldsymbol{\Gamma}_{11}\boldsymbol{V}'_{11} + \boldsymbol{V}'_{11}\boldsymbol{\Gamma}_{11} = \boldsymbol{O}, \\ \boldsymbol{\Delta}_{22}\boldsymbol{U}'_{22} + \boldsymbol{U}'_{22}\boldsymbol{\Delta}_{22} = \boldsymbol{O}, \\ \boldsymbol{\Gamma}_{11}\boldsymbol{V}'_{12} + \boldsymbol{U}'_{12}\boldsymbol{\Delta}_{22} = \boldsymbol{O}. \end{cases}$$

Using the Kronecker product of matrices, we can rewrite the first and second equalities as

$$(\boldsymbol{I} \otimes \boldsymbol{\Gamma}_{11} + \boldsymbol{\Gamma}_{11} \otimes \boldsymbol{I})(\mathbf{vec}\ \boldsymbol{V}'_{11}) = \mathbf{0} \ \text{ and } (\boldsymbol{I} \otimes \boldsymbol{\Delta}_{22} + \boldsymbol{\Delta}_{22} \otimes \boldsymbol{I})(\mathbf{vec}\ \boldsymbol{U}'_{22}) = \mathbf{0}.$$

Since $\boldsymbol{\Gamma}_{11}$ and $\boldsymbol{\Delta}_{22}$ are positive definite, so are the matrices $(\boldsymbol{I} \otimes \boldsymbol{\Gamma}_{11} + \boldsymbol{\Gamma}_{11} \otimes \boldsymbol{I})$ and $(\boldsymbol{I} \otimes \boldsymbol{\Delta}_{22} + \boldsymbol{\Delta}_{22} \otimes \boldsymbol{I})$. Hence we have that $\boldsymbol{V}'_{11} = \boldsymbol{O}$ and $\boldsymbol{U}'_{22} = \boldsymbol{O}$. It follows from $\boldsymbol{V}'_{11} = \boldsymbol{O}$, $\boldsymbol{U}'_{22} = \boldsymbol{O}$, and the last equality of (6.2) that

$$\begin{aligned} \boldsymbol{X}^*\boldsymbol{V} + \boldsymbol{U}\boldsymbol{Y}^* &= \boldsymbol{P}\left(\boldsymbol{\Gamma}\boldsymbol{V}' + \boldsymbol{U}'\boldsymbol{\Delta}\right)\boldsymbol{P}^T \\ &= \boldsymbol{P}\begin{pmatrix} \boldsymbol{\Gamma}_{11}\boldsymbol{V}'_{11} & \boldsymbol{\Gamma}_{11}\boldsymbol{V}'_{12} + \boldsymbol{U}'_{12}\boldsymbol{\Delta}_{22} \\ \boldsymbol{O} & \boldsymbol{U}'_{22}\boldsymbol{\Delta}_{22} \end{pmatrix}\boldsymbol{P}^T \\ &= \boldsymbol{O}. \end{aligned}$$

Therefore we obtain by Condition 6.1 that $(\boldsymbol{U}, \boldsymbol{V}) = (\boldsymbol{O}, \boldsymbol{O})$. ∎

LEMMA 6.3. *There exists a positive number* η_c *such that*

$$\|(d\boldsymbol{X}^k_c, d\boldsymbol{Y}^k_c)\|_F \le \eta_c\theta^{k+1} \ \text{ for every } k = 0, 1, 2, \dots\ .$$

Proof. Since $(\boldsymbol{X}^k_c, \boldsymbol{Y}^k_c) \in \widetilde{\mathcal{N}}(\gamma, \theta^{k+1}\mu^0)$, we see by (i) of Lemma 3.1 that the right-hand side $2\theta^{k+1}\mu^0\boldsymbol{I} - \boldsymbol{X}^k\boldsymbol{Y}^k - \boldsymbol{Y}^k\boldsymbol{X}^k$ of the system (2.11) of equations satisfies

$$\begin{aligned} \frac{\|2\theta^{k+1}\mu^0\boldsymbol{I} - \boldsymbol{X}^k\boldsymbol{Y}^k - \boldsymbol{Y}^k\boldsymbol{X}^k\|_F}{\theta^{k+1}} &\le \frac{\theta^{k+1}\mu^0\sqrt{n} + \|\boldsymbol{X}^k\boldsymbol{Y}^k + \boldsymbol{Y}^k\boldsymbol{X}^k\|_F}{\theta^{k+1}} \\ &\le 2\sqrt{n}(\mu^0 + n + n\zeta\gamma). \end{aligned}$$

On the other hand, we see from (2.11) that

$$(6.3) \qquad \begin{cases} \boldsymbol{X}^k_c\dfrac{d\boldsymbol{Y}^k_c}{\theta^{k+1}} + \dfrac{d\boldsymbol{Y}^k_c}{\theta^{k+1}}\boldsymbol{X}^k_c + \dfrac{d\boldsymbol{X}^k_c}{\theta^{k+1}}\boldsymbol{Y}^k_c + \boldsymbol{Y}^k_c\dfrac{d\boldsymbol{X}^k_c}{\theta^{k+1}} \\ = \dfrac{2\theta^{k+1}\mu^0\boldsymbol{I} - \boldsymbol{X}^k\boldsymbol{Y}^k - \boldsymbol{Y}^k\boldsymbol{X}^k}{\theta^{k+1}}, \\ \dfrac{(d\boldsymbol{X}^k_c, d\boldsymbol{Y}^k_c)}{\theta^{k+1}} \in \mathcal{F}_0. \end{cases}$$

Assume on the contrary that the sequence $\left\{\frac{(d\boldsymbol{X}^k_c, d\boldsymbol{Y}^k_c)}{\theta^{k+1}}\right\}$ is unbounded. Along a subsequence, we then have that

$$\frac{\|(d\boldsymbol{X}^k_c, d\boldsymbol{Y}^k_c)\|_F}{\theta^{k+1}} \to \infty \ \text{ and } \frac{(d\boldsymbol{X}^k_c, d\boldsymbol{Y}^k_c)}{\|(d\boldsymbol{X}^k_c, d\boldsymbol{Y}^k_c)\|_F} \to (\boldsymbol{U}, \boldsymbol{V}) \ne (\boldsymbol{O}, \boldsymbol{O})$$

for some $(\boldsymbol{U}, \boldsymbol{V}) \in \mathcal{F}_0$. Now, dividing the identity (6.3) by $\|(d\boldsymbol{X}^k_c, d\boldsymbol{Y}^k_c)\|_F/\theta^{k+1}$ and taking its limit along the subsequence, we obtain that

$$\boldsymbol{X}^*\boldsymbol{V} + \boldsymbol{V}\boldsymbol{X}^* + \boldsymbol{U}\boldsymbol{Y}^* + \boldsymbol{Y}^*\boldsymbol{U} = \boldsymbol{O} \ \text{ and } (\boldsymbol{O}, \boldsymbol{O}) \ne (\boldsymbol{U}, \boldsymbol{V}) \in \mathcal{F}_0.$$

This contradicts Lemma 6.2. ∎

Proof of Theorem 6.1. In view of Lemma 6.3, we know that $\delta^k_c \to 0$ as $k \to \infty$, which implies (i) and (ii) of the theorem.

7. Concluding remarks.

(A) The admissible region $\{(\boldsymbol{X},\boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma,\tau) \ : \ \tau > 0\}$ in which we confine iterates $(\boldsymbol{X}^k,\boldsymbol{Y}^k)$ $(k = 0, 1, 2, \dots)$ becomes larger as we take larger $\gamma < 1$. Taking the limit as $\gamma \to 1$, we have the largest admissible region

$$\bigcup_{0<\gamma<1} \{(\boldsymbol{X},\boldsymbol{Y}) \in \widetilde{\mathcal{N}}(\gamma,\tau) \ : \ \tau > 0\} = \{(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S}_+ \times \mathcal{S}_+ \ : \ \boldsymbol{XY} + \boldsymbol{YX} \succ \boldsymbol{O}\}.$$

It is easily seen that this set is contained in $\mathcal{S}_{++} \times \mathcal{S}_{++}$. But the converse relation

$$\mathcal{S}_{++} \times \mathcal{S}_{++} \subset \{(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S}_+ \times \mathcal{S}_+ \ : \ \boldsymbol{XY} + \boldsymbol{YX} \succ \boldsymbol{O}\}$$

is not true. For a counterexample to this relation, see the paper [29].

(B) We can use a different admissible region. For every $\gamma \in [0, 1]$ and $\tau \geq 0$, define

$$\widehat{\mathcal{N}}(\gamma,\tau) = \left\{(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S}_+ \times \mathcal{S}_+ \ : \ \left\| \frac{\boldsymbol{XY} + \boldsymbol{YX}}{2} - \tau \boldsymbol{I} \right\|_F \leq \gamma\tau \right\}. \tag{7.1}$$

We can easily verify that

$$\{(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S}_+ \times \mathcal{S}_+ \ : \ \boldsymbol{XY} = \tau \boldsymbol{I}\} \subset \widehat{\mathcal{N}}(\gamma,\tau) \subset \widetilde{\mathcal{N}}(\gamma,\tau) \text{ for every } \gamma \in (0,1) \text{ and } \tau > 0.$$

But we need some modification in Algorithm 2.1 to prove global and local convergence.

(C) It is interesting to compare our (modified) algorithm with the predictor-corrector infeasible-interior-point algorithm given by Potra and Sheng [28]. (See also [27].)

- Our modified algorithm uses the combination of the AHO search direction [2] and the neighborhood $\widehat{\mathcal{N}}(\gamma,\tau)$ (given in (7.1)) of $\{(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S}_+ \times \mathcal{S}_+ : \boldsymbol{XY} = \tau \boldsymbol{I}\}$, while Potra and Sheng's algorithm uses the combination of the HRVW/KSH/M direction [8, 15, 20] and the neighborhood

  $$\mathcal{N}(\gamma,\tau) = \{(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S}_+ \times \mathcal{S}_+ \ : \ \|\boldsymbol{X}^{1/2}\boldsymbol{Y}\boldsymbol{X}^{1/2} - \tau \boldsymbol{I}\|_F \leq \gamma\tau\}$$

  of $\{(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S}_+ \times \mathcal{S}_+ : \boldsymbol{XY} = \tau \boldsymbol{I}\}$. Using Lemma 3.3 of Monteiro [20], we can prove that $\widehat{\mathcal{N}}(\gamma,\tau) \subset \mathcal{N}(\gamma,\tau)$ for every $\gamma \in (0,1)$ and $\tau > 0$.
- Our global convergence theorem, Theorem 2.1, has not guaranteed the global convergence at any linear rate (see also Remark 4.1) while Potra and Sheng proved the global convergence at a linear rate for their algorithm (Theorems 3.6, 3.7, and 3.8 of [26]).
- We have proved the quadratic local convergence under the strict complementarity condition while their algorithm requires different assumptions (section 4 of [26]) or additional restrictions (sections 3, 4, 5, and 6 of Kojima, Shida, and Shindoh [14]) to attain the superlinear convergence.

(D) We briefly mention some recent results on interior-point methods based on the AHO direction after the first version of this paper was written in January 1996. In May 1996, Alizadeh, Haeberly, and Overton [4] studied several variants of primal-dual interior-point algorithms using the AHO direction (or the XZ+ZX direction in their terminology), the HRVW/KSH/M direction [8, 15, 20] (or the XZ direction in their terminology), and the Nesterov–Todd direction [23, 24]. They reported through numerical results that the Mehrotra-type predictor-corrector algorithm using the AHO direction is more stable, converges faster, and achieves higher accuracy than the other variants. In July 1996, Monteiro [21] presented the polynomial iteration complexity for a short-step primal-dual path-following interior-point algorithm and a Mizuno–Todd–Ye-type predictor-corrector interior-point algorithm for the SDP based on the Monteiro–Zhang family of directions which covers the AHO direction as a special case. In general, polynomial iteration complexity is stronger than global convergence. We should mention, however, that our global convergence result (Theorem 2.1) does not follow from his polynomial iteration complexity result. A critical difference lies in the neighborhoods which Monteiro and we use. To ensure polynomial iteration complexity, his predictor-corrector algorithm (Algorithm II in section 4 of [21]) needs to generate a sequence in a very narrow neighborhood of the central trajectory, which is apparently of theoretical interest. On the other hand, our algorithm (Algorithm 2.1 in section 2) uses a fairly large neighborhood of the central trajectory. In the second version of the paper (August 1996), we proved local quadratic convergence (Theorem 5.1) without the nondegeneracy condition (Condition 6.1); the first version used both Conditions 5.1 and 6.1 to prove quadratic convergence. Tseng [30] also presented polynomial iteration complexity for a Mizuno–Todd–Ye-type predictor-corrector interior-point algorithm for the SDP using the AHO direction in August 1996.

REFERENCES

[1] F. Alizadeh, *Interior point methods in semidefinite programming with application to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.

[2] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Primal-dual interior-point methods for semidefinite programming*, manuscript presented at the Math. Programming Symposium, Ann Arbor, MI, 1994.

[3] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Complementarity and nondegeneracy in semidefinite programming*, Math. Programming, 77 (1997), pp.111–128.

[4] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results*, SIAM J. Optim., 8 (1998), pp. 746–768.

[5] R. M. Freund, *Complexity of an Algorithm for Finding an Approximate Solution of a Semidefinite Program with No Regularity Assumption*, Technical report OR 302-94, Operations Research Center, MIT, Cambridge, MA, 1994.

[6] A. Graham, *Kronecker Products and Matrix Calculus with Applications*, Ellis Horwood Ltd., West Sussex, England, 1981.

[7] J.-P. A. Haeberly, private communication, 1996.

[8] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. Optim., 6 (1996), pp. 342–361.

[9] M. Kojima, S. Kojima, and S. Hara, *Linear algebra for semidefinite programming*, RIMS Kokyuroku, 1004 (1997), pp. 1–23, Research Institute of Mathematical Sciences, Kyoto University, Kyoto, Japan.

[10] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point algorithm for linear programming*, in Progress in Mathematical Programming: Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 29–47.

[11] M. Kojima, S. Mizuno, and A. Yoshise, *A polynomial-time algorithm for a class of linear complementary problems*, Math. Programming, 44 (1989), pp. 1–26.

[12] M. Kojima, S. Mizuno, and A. Yoshise, *An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems*, Math. Programming, 50 (1991), pp. 331–342.

[13] M. Kojima, M. Shida, and S. Shindoh, *Reduction of monotone linear complementarity problems over cones to linear programs over cones*, Acta Math. Vietnam., 22 (1997), pp. 147–157.

[14] M. Kojima, M. Shida, and S. Shindoh, *Local convergence of predictor-corrector infeasible-interior-point algorithms for SDPs and SDLCPs*, Math. Programming, 80 (1998), pp. 129–160.

[15] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone semidefinite linear complementarity problems*, SIAM J. Optim., 7 (1997), pp. 86–125.

[16] C.-J. Lin and R. Saigal, *A Predictor-Corrector Method for Semi-definite Linear Programming*, Working paper, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI, 1995.

[17] S. Mizuno, M. Kojima, and M. J. Todd, *Infeasible-interior-point primal-dual potential-reduction algorithms for linear programming*, SIAM J. Optim., 5 (1995), pp. 52–67.

[18] S. Mizuno, M. J. Todd and Y. Ye, *On Adaptive-step primal-dual interior-point algorithms for linear programming*, Math. Oper. Res., 18 (1993), pp. 964–981.

[19] R. D. C. Monteiro and I. Adler, *Interior path-following primal-dual algorithm. Part* I: *Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[20] R. D. C. Monteiro, *Primal-dual path following algorithms for semidefinite programming*, SIAM J. Optim., 7 (1997), pp. 663–678.

[21] R. D. C. Monteiro, *Polynomial convergence of primal-dual algorithms for semidefinite programming based on the Monteiro and Zhang family of directions*, SIAM J. Optim., 8 (1998), pp. 797–812.

[22] Yu. E. Nesterov and A. S. Nemirovskii, *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, SIAM, Philadelphia, PA, 1994.

[23] Yu. E. Nesterov and M. J. Todd, *Self-scaled barriers and interior-point methods for convex programming*, Math. Oper. Res., 22 (1997), pp. 1–42.

[24] Yu. E. Nesterov and M. J. Todd, *Primal-dual interior-point methods for self-scaled cones*, SIAM J. Optim., 8 (1998), pp. 324–364.

[25] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[26] F. A. Potra, *An $O(nL)$ infeasible-interior-point algorithm for LCP with quadratic convergence*, *Interior point methods in mathematical programming*, Ann. Oper. Res., 62 (1996), pp. 81–102.

[27] F. A. Potra and R. Sheng, *A superlinearly convergent primal-dual infeasible-interior-point algorithm for semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 1007–1028.

[28] F. A. Potra and R. Sheng, *Superlinear Convergence of Infeasible-Interior-Point Algorithms for Semidefinite Programming*, Department of Mathematics, University of Iowa, Iowa City, IA, 1996.

[29] M. Shida, S. Shindoh, and M. Kojima, *Existence and uniqueness of search directions in interior-point-algorithms for the SDP and the monotone SDLCP*, SIAM J. Optim., 8 (1998), pp. 387–396.

[30] P. Tseng, *Analysis of Infeasible Path-Following Methods Using the Alizadeh-Haeberly-Overton Directions for the Monotone Semi-Definite LCP*, Technical report, Department of Mathematics, University of Washington, Seattle, WA, 1996.

[31] L. Vandenberghe and S. Boyd, *A primal-dual potential reduction method for problems involving matrix inequalities*, Math. Programming, 69 (1995), pp. 205–236.

[32] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

[33] Y. Zhang, *On the convergence of a class of infeasible interior-point algorithms for the horizontal linear complementarity problem*, SIAM J. Optim., 4 (1994), pp. 208–227.

[34] Y. Zhang, *On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 365–386.

SIAM J. OPTIM.
Vol. 9, No. 2, pp. 466–492

SECOND ORDER OPTIMALITY CONDITIONS BASED ON PARABOLIC SECOND ORDER TANGENT SETS*

J. FRÉDÉRIC BONNANS[†], ROBERTO COMINETTI[‡], AND ALEXANDER SHAPIRO[§]

Abstract. In this paper we discuss second order optimality conditions in optimization problems subject to abstract constraints. Our analysis is based on various concepts of second order tangent sets and parametric duality. We introduce a condition, called second order regularity, under which there is no gap between the corresponding second order necessary and second order sufficient conditions. We show that the second order regularity condition always holds in the case of semidefinite programming.

Key words. second order optimality conditions, semidefinite programming, semi-infinite programming, tangent sets, Lagrange multipliers, cone constraints, duality

AMS subject classifications. 49K27, 90C30, 90C34

PII. S1052623496306760

1. Introduction.

In this paper we investigate *necessary* as well as *sufficient* second order optimality conditions for an optimization problem in the form

$$\operatorname*{Min}_{x\in X} f(x) \ \text{ subject to } \ G(x) \in K, \tag{1.1}$$
(P)

where X is a finite dimensional space, Y is a Banach space, K is a closed convex subset of Y, and the objective function $f : X \to \mathbb{R}$ as well as the constraint mapping $G : X \to Y$ are assumed to be twice continuously differentiable. By $\Phi := G^{-1}(K)$ we denote the feasible set of (P).

A number of optimization problems can be formulated in the form (1.1) in a natural way. When $Y = \mathbb{R}^p$ and $K = \{0\} \times \mathbb{R}^{p-q}_+$, the feasible set of (P) is defined by a finite number of equality and inequality constraints and (P) becomes a nonlinear programming problem. As another example, consider the space $Y = C(\Omega)$ of continuous functions $\psi : \Omega \to \mathbb{R}$, defined on a compact metric space Ω and equipped with the sup-norm $\|\psi\| := \sup_{\omega\in\Omega} |\psi(\omega)|$. Let $K := C_+(\Omega)$ be the cone of nonnegative valued functions, i.e.,

$$C_+(\Omega) := \{\psi \in C(\Omega) : \psi(\omega) \geq 0 \text{ for all } \omega \in \Omega\}.$$

In that case the abstract constraint $G(x) \in K$ corresponds to $g(x, \omega) \geq 0$ for all $\omega \in \Omega$, where $g(x, \cdot) := G(x)(\cdot)$. If the set Ω is infinite, this leads to an infinite number of constraints and (P) becomes a semi-infinite programming problem (cf. [18] and references therein). Yet another example is provided by semidefinite programming (see, e.g., [43]). There $Y = \mathcal{S}^n$ is the space of $n \times n$ symmetric matrices and $K = \mathcal{S}^n_+$

[*] Received by the editors July 16, 1996; accepted for publication (in revised form) February 4, 1998; published electronically March 17, 1999. This research was supported by the French-Chilean ECOS Program and by the European Community under contract 931091CL.

http://www.siam.org/journals/siopt/9-2/30676.html

[†] INRIA, B.P. 105, 78153 Rocquencourt, France (Frederic.Bonnans@inria.fr).

[‡] Universidad de Chile, Casilla 170/3 Correo 3, Santiago, Chile (rcominet@dim.uchile.cl). The work of this author was supported in part by Fondecyt 1961131.

[§] School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205 (ashapiro@isye.gatech.edu). The work of this author was supported in part by National Science Foundation grant DMI-9713878.

is the cone of positive semidefinite matrices. Note that $\mathcal{S}^n_+$ can be represented in the form

$$\mathcal{S}^n_+ = \left\{Z \in \mathcal{S}^n : \omega^T Z \omega \geq 0,\ \omega \in \mathbb{R}^n, \|\omega\| = 1\right\}$$

so that semidefinite programming can be considered in the framework of semi-infinite programming.

An alternative approach for studying abstract optimality conditions is to consider optimization problems of the form

$$\operatorname*{Min}_{x \in X} g(F(x)), \tag{1.2}$$

where $g : Y \to \mathbb{R} \cup \{+\infty\}$ is a lower semicontinuous proper convex function and $F : X \to Y$. This problem, known as a *composite optimization* problem, is equivalent to (e.g., [27])

$$\operatorname*{Min}_{(x,c) \in X \times \mathbb{R}} c \quad \text{subject to} \quad (F(x), c) \in \operatorname{epi}(g), \tag{1.3}$$

where $\operatorname{epi}(g) := \{(y, c) \in Y \times \mathbb{R} : g(y) \leq c\}$ is the epigraph of g and hence it can be considered as a particular case of the problem (1.1). The converse is also true, that is, problem (1.1) can be represented in the form (1.2) by taking $g(r, y) = r + I_K(y)$ and $F(x) = (f(x), G(x))$, where $I_K(y) = 0$ for $y \in K$ and $+\infty$ elsewhere (see [20, 27]), so that both approaches are essentially equivalent.

Second order optimality conditions have been studied in numerous publications. In order to give a general idea of that type of result, consider for the moment the simplest case when problem (P) is unconstrained. Let x_0 be a stationary point, i.e., it satisfies the first order optimality condition $\nabla f(x_0) = 0$. Then it is well known that the second order necessary condition for x_0 to be locally optimal is that the Hessian matrix $\nabla^2 f(x_0)$ should be positive semidefinite, i.e., $h^T \nabla^2 f(x_0) h \geq 0$ for all $h \in X$. The corresponding second order sufficient condition is that there exists $\alpha > 0$ such that $h^T \nabla^2 f(x_0) h > \alpha \|h\|^2$ for all $h \in X \setminus \{0\}$. Since X is finite dimensional, this is equivalent to $h^T \nabla^2 f(x_0) h > 0$ for all $h \in X \setminus \{0\}$, i.e., $\nabla^2 f(x_0)$ is positive definite. This condition is in fact necessary and sufficient for quadratic growth (3.13). The only difference between the second order necessary condition and the sufficient condition is the term $\alpha \|h\|^2$ in the right-hand side of the former. In such a case we say that there is *no gap* between the necessary and the sufficient second order conditions.

In the case of nonlinear programming (i.e., when the space Y is finite dimensional and the set K is polyhedral), "no gap" second order optimality conditions were already given, under somewhat restrictive assumptions, in [15]. In a sense, a complete description of no gap second order conditions for nonlinear programming was given in Ioffe [19], Ben-Tal [2], and Ben-Tal and Zowe [3].

In semi-infinite programming second order optimality conditions were first derived (under quite restrictive assumptions) by the so-called reduction method, e.g., [1, 16, 17, 37, 44] (see [18] for additional references). It was already clear in those papers that an additional term, representing the curvature of the set K, should appear in second order optimality conditions in order to obtain no gap second order conditions. An attempt to describe this additional term in an abstract way (in the case of semi-infinite programming) was made in Kawasaki [23]. This sparked an intensive investigation aimed at closing the gap between necessary and sufficient second order conditions [11, 12, 20, 21, 25, 26, 27, 34].

Second order optimality conditions for problem (P) may also be obtained by formulating it as a composite optimization problem in the form (1.2) and using the so-called second order (epi)subderivatives. That approach was investigated in Rockafellar [34] for twice epidifferentiable functions and further explored by Ioffe [21] and Cominetti [13]. (See also [36] for a detailed account of that approach.) In particular, in the case of the composition of a piecewise linear-quadratic convex function with a twice continuously differentiable mapping, no gap second order optimality conditions can be explicitly stated in terms of second order (epi)subderivatives.

An alternative approach developed in this paper, which goes back to Ben-Tal [2] and was later refined in Cominetti [12], is based on verification of optimality along curves that have a second order expansion (in that case we speak of a parabolic curve). This approach leads to more explicit second order optimality conditions involving the Hessian of the Lagrangian and the support function of a second order tangent approximation of the set K. Explicit expressions of this support function are known in various situations (see Cominetti and Penot [14]), including semidefinite programming (see Shapiro [41]). This approach is also convenient for sensitivity analysis of parameterized optimization problems [5, 9].

It is clear, however, that there is no reason a priori why optimality should be verified along parabolic curves only. Therefore, one may expect a gap between such necessary and corresponding sufficient second order optimality conditions. Nevertheless, one may search for classes of problems for which the "parabolic" estimates coincide with the estimates based on the second order lower epiderivatives approach. This was done, in the context of infinite dimensional sensitivity analysis, in [5] under an assumption of generalized polyhedricity (although second order lower epiderivatives are not explicitly mentioned in [5], all lower estimates in that paper, in fact, are lower epiderivative estimates).

The main purpose of this paper is to identify a wide class of sets K for which there is no gap between necessary and sufficient second order optimality conditions obtained via the parabolic curve approach. We argue that such sets, which we call *second order regular*, are natural for purposes of second order analysis. In particular we show that cones of positive semidefinite matrices are always second order regular. This complements results in [40, 41] and gives quite a complete description of no gap second order optimality conditions in semidefinite programming. It is possible to show that the epigraph of a piecewise linear-quadratic convex function is second order regular, and hence the suggested approach can be shown to cover the second order optimality conditions obtained for composite optimization in [34]. In the follow-up paper [6], we also show that the concept of second order regularity is useful in sensitivity analysis of parameterized optimization problems.

The organization of this paper is as follows. In the next section we introduce and discuss some concepts of second order tangent sets. Second order necessary and second order sufficient optimality conditions, for the problem (P) in the form (1.1), are given in section 3. Those conditions become no gap second order conditions under the assumption of second order regularity of the set K, which is discussed in section 4. In section 5 we translate the obtained results into the framework of composite optimization. Finally in section 6 some extensions to the case of nonisolated minima are presented.

Throughout this paper we use the following notation and terminology. Let $h : Y \to \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$ be an extended real valued function. Assuming that $h(\cdot)$ is finite at a point $y \in Y$, we denote by $h'(y,d)$ its directional derivative

$$h'(y,d) := \lim_{t\downarrow 0} \frac{h(y+td)-h(y)}{t}$$

at the point y in the direction $d \in Y$. Recall that if $h(\cdot)$ is convex, finite valued, and continuous at y, then $h'(y,d)$ exists and is finite valued [31]. In order to deal with possibly discontinuous convex functions in composite optimization (see section 5), we also use the lower directional subderivative $h^{\downarrow}(y,d)$ (see [35])

$$h^{\downarrow}(y,d) := \liminf_{t\downarrow 0, d'\to d} \frac{h(y+td')-h(y)}{t}.$$

It is not difficult to show from the definitions that, provided $h(y)$ is finite, the epigraph of $\psi(\cdot) := h^{\downarrow}(y,\cdot)$ coincides with the contingent (Bouligand) cone (see (2.3) below) to the epigraph of h at the point $(y, h(y))$ (cf. [35]). Therefore the epigraph of $h^{\downarrow}(y,\cdot)$ is closed and hence $h^{\downarrow}(y,\cdot)$ is lower semicontinuous. Note that if $h(\cdot)$ is convex, finite valued, and continuous at y, and hence is Lipschitz continuous in a neighborhood of y, then $h^{\downarrow}(y,\cdot) \equiv h'(y,\cdot)$. In general, if h is a convex, possibly discontinuous function, then the topological closure of the epigraph of $h'(y,\cdot)$ coincides with the epigraph of $h^{\downarrow}(y,\cdot)$.

When $h'(y,d)$ exists and is finite, we denote by $h''_{-}(y;d,w)$ and $h''_{+}(y;d,w)$ its lower and upper second order parabolic derivatives [3], respectively, i.e.,

$$h''_{-}(y;d,w) := \liminf_{t\downarrow 0} \frac{h(y+td+\frac{1}{2}t^2 w)-h(y)-th'(y,d)}{\frac{1}{2}t^2},$$

$$h''_{+}(y;d,w) := \limsup_{t\downarrow 0} \frac{h(y+td+\frac{1}{2}t^2 w)-h(y)-th'(y,d)}{\frac{1}{2}t^2}.$$

We say that $h(\cdot)$ is second order directionally differentiable, at y in the direction d, if $h''_{-}(y;d,w)$ is equal to $h''_{+}(y;d,w)$ and is finite for all $w \in Y$. In that case the common value is denoted $h''(y;d,w)$. We also use, when $h(y)$ and $h^{\downarrow}(y,d)$ are finite, the following lower second order parabolic derivative:

$$h^{\downarrow\downarrow}_{-}(y;d,w) := \liminf_{t\downarrow 0,\, w'\to w} \frac{h(y+td+\frac{1}{2}t^2 w')-h(y)-th^{\downarrow}(y,d)}{\frac{1}{2}t^2}.$$

Note that if $h(\cdot)$ is Lipschitz continuous near y, then $h^{\downarrow\downarrow}_{-}(y;d,w) \equiv h''_{-}(y;d,w)$. This holds, in particular, if $h(\cdot)$ is convex, finite, and continuous, and hence is Lipschitz continuous, at y.

By Y^* we denote the dual space of Y and by $\langle y^*, y\rangle$ the value $y^*(y)$ of the linear functional $y^* \in Y^*$ at $y \in Y$. For a linear continuous mapping $A : X \to Y$ we denote by $A^* : Y^* \to X^*$ its adjoint mapping, i.e., $\langle A^* y^*, x\rangle = \langle y^*, Ax\rangle$ for all $x \in X$ and $y^* \in Y^*$. For a set $T \subset Y$ we denote by $\sigma(\cdot, T)$ its support function, i.e., $\sigma(y^*, T) := \sup_{y\in T} \langle y^*, y\rangle$, and by $\operatorname{dist}(\cdot, T)$ the distance $\operatorname{dist}(y,T) := \inf_{z\in T} \|y - z\|$. By $Df(x)$ and $D^2 f(x)$ we denote the first and second order derivatives, respectively, of a function $f(x)$. We denote by $B_Y := \{y \in Y : \|y\| \le 1\}$ the unit ball in Y. By $[\![y]\!] := \{ty : t \in \mathbb{R}\}$ we denote the linear space (one dimensional if $y \neq 0$) generated by vector y.

2. Tangent sets.

In this section we discuss the notions of first and second order tangent sets on which our second order optimality conditions are based.

Let us first recall the notion of limits in the sense of Painlevé–Kuratowski for a multifunction $\Psi : X \to 2^Y$ from a normed space X into the set 2^Y of subsets of Y. The upper limit $\limsup_{x \to x_0} \Psi(x)$ is the set of points $y \in Y$ for which *there exists* a sequence $x_n \to x_0$ such that $y_n \to y$ for some $y_n \in \Psi(x_n)$. The lower limit $\liminf_{x \to x_0} \Psi(x)$ is the set of points $y \in Y$ such that for *every* sequence $x_n \to x_0$ it is possible to find $y_n \in \Psi(x_n)$ such that $y_n \to y$.

Let K be a closed subset of a Banach space Y. The (first order) tangent set (cone) to K at a point $y \in K$ can be defined as follows:

$$T_K(y) := \{h \in Y : \operatorname{dist}(y + th, K) = o(t),\ t \geq 0\}. \tag{2.1}$$

By the definition of lower set limits this can be written in the form

$$T_K(y) = \liminf_{t \downarrow 0} \frac{K - y}{t}. \tag{2.2}$$

It is well known that whenever K is convex, it is also true that

$$T_K(y) = \limsup_{t \downarrow 0} \frac{K - y}{t}. \tag{2.3}$$

Note that if K is a convex cone and $y \in K$, then $T_K(y) = \operatorname{cl}(K + [\![y]\!])$, where $[\![y]\!]$ denotes the linear space generated by vector y and "cl" stands for the topological closure in the norm topology of Y.

Similarly to (2.2) and (2.3) we consider second order variations of the set K at a point $y \in K$ in a direction d. That is,

$$T_K^2(y, d) := \liminf_{t \downarrow 0} \frac{K - y - td}{\frac{1}{2}t^2}, \tag{2.4}$$

$$O_K^2(y, d) := \limsup_{t \downarrow 0} \frac{K - y - td}{\frac{1}{2}t^2}. \tag{2.5}$$

We call $T_K^2(y, d)$ and $O_K^2(y, d)$ the *inner* and *outer* second order tangent sets, respectively. Alternatively these tangent sets can be written in the form

$$T_K^2(y, d) = \left\{w \in Y : \operatorname{dist}(y + td + \tfrac{1}{2}t^2 w, K) = o(t^2),\ t \geq 0\right\},$$

$$O_K^2(y, d) = \left\{w : \exists\, t_n \downarrow 0 \text{ such that } \operatorname{dist}(y + t_n d + \tfrac{1}{2}t_n^2 w, K) = o(t_n^2)\right\}.$$

It is clear from the above definitions that $T_K^2(y, d) \subset O_K^2(y, d)$ and that these second order tangent sets can be nonempty only if $d \in T_K(y)$. Also, both sets $T_K^2(y, d)$ and $O_K^2(y, d)$ are closed. If K is convex, then the set $T_K^2(y, d)$ is convex. On the other hand the outer second order tangent set $O_K^2(y, d)$ can be nonconvex. An example of a convex set K (in $\mathbb{R}^4$) for which $O_K^2(y, d)$ is nonconvex is constructed in the forthcoming book [10]. (That example is not trivial and will be not repeated here.)

The following example demonstrates that unlike the first order tangent variations, the second order inner and outer tangent sets can be different. (Other examples have been given in [14, 27].) It also shows that lower and upper second order directional derivatives can be different even for a convex continuous function of one variable.

Example 2.1. *Let us first construct a convex piecewise linear function $y = \eta(x)$, $x \in \mathbb{R}$, oscillating between two parabolas $y = x^2$ and $y = 2x^2$. That is, we construct $\eta(x)$ in such a way that $\eta(x) = \eta(-x)$, $\eta(0) = 0$ and for some monotonically decreasing to zero sequence x_k, the function $\eta(x)$ is linear on every interval $[x_{k+1}, x_k]$, $\eta(x_k) = x_k^2$ and the straight line passing through the points $(x_k, \eta(x_k))$ and $(x_{k+1}, \eta(x_{k+1}))$ is tangent to the curve $y = 2x^2$. It is quite clear how such a function can be constructed. For a given point $x_k > 0$ consider the straight line passing through the point (x_k, x_k^2) and tangent to the curve $y = 2x^2$. This straight line intersects the curve $y = x^2$ at a point x_{k+1}. Clearly $x_k > x_{k+1} > 0$. One can proceed with the construction in an iterative way. It is easily proved that $x_k \to 0$.*

Define $K := \{(x, y) \in \mathbb{R}^2 : y \geq \eta(x)\}$. We have then that for the direction $d := (1, 0)$, $T_K^2(0, d) = \{(x, y) : y \geq 4\}$ and $O_K^2(0, d) = \{(x, y) : y \geq 2\}$. It also can be seen that for any $w \in \mathbb{R}$, $\eta''_-(0; 1, w) = 2$ and $\eta''_+(0; 1, w) = 4$ and hence $\eta(\cdot)$ is not second order directionally differentiable at zero.

We say that the set K is *second order directionally differentiable*, at $y \in K$ in a direction d, if $T_K^2(y, d) = O_K^2(y, d)$ (for various related concepts see [35]). This terminology is justified by the following result, which is an extension of [12, Proposition 4.1].

Proposition 2.1. *Suppose that the set K is defined in the form $K = \{y \in Y : h(y) \leq 0\}$, where $h : Y \to \mathbb{R} \cup \{+\infty\}$ is a proper convex function. Let $h(y) = 0$ and $h^{\downarrow}(y, d) = 0$, and suppose that there exists $\bar{y}$ such that $h(\bar{y}) < 0$ (Slater condition). Then*

$$O_K^2(y, d) = \left\{w : h_-^{\downarrow\downarrow}(y; d, w) \leq 0\right\}. \tag{2.6}$$

If, in addition, $h(\cdot)$ is continuous at y, then

$$T_K^2(y, d) = \{w : h''_+(y; d, w) \leq 0\}. \tag{2.7}$$

Proof. We show only that (2.6) holds since the proof of (2.7) is similar. Consider $w \in O_K^2(y, d)$, and choose sequences $t_n \to 0^+$ and $w_n \to w$ such that $y + t_n d + \frac{1}{2}t_n^2 w_n \in K$, and hence $h(y + t_n d + \frac{1}{2}t_n^2 w_n) \leq 0$. Then

$$h_-^{\downarrow\downarrow}(y; d, w) \leq \liminf_{n\to\infty} \frac{h(y + t_n d + \frac{1}{2}t_n^2 w_n)}{\frac{1}{2}t_n^2} \leq 0.$$

Conversely, suppose first that $h_-^{\downarrow\downarrow}(y; d, w) < 0$. Then for some $t_n \to 0^+$ and $w_n \to w$, we have that

$$h(y + t_n d + \tfrac{1}{2}t_n^2 w_n) = \tfrac{1}{2}t_n^2 h_-^{\downarrow\downarrow}(y; d, w) + o(t_n^2),$$

and hence $h(y + t_n d + \frac{1}{2}t_n^2 w_n) < 0$ for n large enough. Consequently

$$y + t_n d + \tfrac{1}{2}t_n^2 w_n \in K,$$

which implies that $w \in O_K^2(y, d)$.

Suppose now that $h_-^{\downarrow\downarrow}(y; d, w) = 0$, and hence for some $t_n \to 0^+$ and $w_n \to w$, $h(y + t_n d + \frac{1}{2}t_n^2 w_n) = o(t_n^2)$. Given $\alpha > 0$ and $w' \in Y$, set $w'_\alpha := w' + \alpha(\bar{y} - y)$. Then, by convexity of h, we have that for $t' \geq 0$ small enough such that $1 - \frac{1}{2}\alpha {t'}^2 > 0$,

$$h(y + t'd + \tfrac{1}{2}{t'}^2 w'_\alpha) \leq (1 - \tfrac{1}{2}\alpha {t'}^2)\gamma(t', w') + \tfrac{1}{2}\alpha {t'}^2 h(\bar{y}), \tag{2.8}$$

where

$$\gamma(t', w') := h\left(y + t'(1 - \tfrac{1}{2}\alpha {t'}^2)^{-1} d + \tfrac{1}{2}{t'}^2(1 - \tfrac{1}{2}\alpha {t'}^2)^{-1} w\right).$$

Define t'_n and w'_n by the relations $t'_n(1 - \frac{1}{2}\alpha {t'_n}^2)^{-1} = t_n$, i.e., $t'_n = 2t_n/(1 + \sqrt{1 + 2\alpha t_n^2})$, and $(1 - \frac{1}{2}\alpha {t'_n}^2) w'_n = w_n$. Then

$$\gamma(t'_n, w'_n) = h(y + t_n d + \tfrac{1}{2}t_n^2 w_n) = o(t_n^2).$$

Since $t'_n \to 0^+$, $w'_n + \alpha(\bar{y} - y) \to w_\alpha$, and $h(\bar{y}) < 0$, it follows then by (2.8) that for any $\alpha > 0$,

$$h_-^{\downarrow\downarrow}(y; d, w_\alpha) \le \alpha h(\bar{y}) < 0$$

and hence $w_\alpha \in O_K^2(y, d)$. Since $O_K^2(y, d)$ is closed, letting $\alpha \to 0^+$ we obtain that $w \in O_K^2(y, d)$, which completes the proof of (2.6). □

If $h(\cdot)$ is convex and continuous at y, then second order derivatives $h_-^{\downarrow\downarrow}(y; d, \cdot)$ and $h''_-(y; d, \cdot)$ are the same. Then it follows from the above proposition, provided the Slater condition holds, that K is second order directionally differentiable, at the point y in the direction d, if and only if the level sets $\{w : h''_-(y; d, w) \le 0\}$ and $\{w : h''_+(y; d, w) \le 0\}$ coincide. In particular, K is second order directionally differentiable if $h(\cdot)$ is second order directionally differentiable.

To close this section we state two results, which extend Proposition 3.1 and Theorem 3.1 in [12] to the case of outer second order tangent sets. We omit the proofs, which are simple modifications of those in the cited reference.

PROPOSITION 2.2. *For all $y \in K, d \in T_K(y)$ one has*

$$T_K^2(y, d) + T_{T_K(y)}(d) \subset T_K^2(y, d) \subset T_{T_K(y)}(d), \tag{2.9}$$

$$O_K^2(y, d) + T_{T_K(y)}(d) \subset O_K^2(y, d) \subset T_{T_K(y)}(d). \tag{2.10}$$

In particular, it follows from the above proposition that $T_{T_K(y)}(d)$ is the recession cone of $T_K^2(y, d)$ and $O_K^2(y, d)$ whenever these sets are nonempty. Moreover, if $0 \in O_K^2(y, d)$, then $O_K^2(y, d) = T_{T_K(y)}(d)$ and when $0 \in T_K^2(y, d)$ all three sets coincide:

$$T_K^2(y, d) = O_K^2(y, d) = T_{T_K(y)}(d).$$

Note also that $T_{T_K(y)}(d) = \operatorname{cl}\{T_K(y) + [\![d]\!]\}$, provided $d \in T_K(y)$; $T_{T_K(y)}(d)$ is empty otherwise.

The following formulas (2.12) and (2.13) provide a rule for computing the second order tangent approximations of the feasible set $\Phi := G^{-1}(K)$ of (P) in terms of the second order tangent approximations of K. These formulas are valid under Robinson's constraint qualification [29]

$$0 \in \operatorname{int}\{G(x_0) + DG(x_0)X - K\} \tag{2.11}$$

and can be proved by using the Robinson–Ursescu [30, 42] stability theorem (see [12]).

PROPOSITION 2.3. *Let $x_0 \in \Phi := G^{-1}(K)$, and suppose that Robinson's constraint qualification* (2.11) *holds. Then, for all $h \in X$,*

$$T_\Phi^2(x_0, h) = DG(x_0)^{-1}\left[T_K^2(G(x_0), DG(x_0)h) - D^2G(x_0)(h, h)\right], \tag{2.12}$$

$$O_\Phi^2(x_0, h) = DG(x_0)^{-1}\left[O_K^2(G(x_0), DG(x_0)h) - D^2G(x_0)(h, h)\right]. \tag{2.13}$$

3. Second order optimality conditions. In this section we derive second order necessary and sufficient optimality conditions for a problem (P) given in the form (1.1). With problem (P) are associated the Lagrangian

$$L(x, \lambda) := f(x) + \langle \lambda, G(x) \rangle, \quad \lambda \in Y^*,$$

and the generalized Lagrangian

$$L^*(x, \alpha, \lambda) := \alpha f(x) + \langle \lambda, G(x) \rangle, \quad (\alpha, \lambda) \in \mathbb{R} \times Y^*.$$

Let x_0 be a locally optimal solution of problem (P). Then F. John–type (first order) optimality conditions can be written in the following form: there exists $(\alpha, \lambda) \in \mathbb{R} \times Y^*$, $(\alpha, \lambda) \neq (0,0)$, such that

$$D_x L^*(x_0, \alpha, \lambda) = 0, \ \alpha \geq 0, \ \lambda \in N_K(G(x_0)). \tag{3.1}$$

Here $N_K(y) := \{y^* \in Y^* : \langle y^*, z - y \rangle \leq 0 \text{ for all } z \in K\}$ is the normal cone to K at y. We denote by $\Lambda^*(x_0)$ the set of generalized Lagrange multipliers $(\alpha, \lambda) \neq (0,0)$ satisfying condition (3.1). It should be noted that for a general Banach space Y the set $\Lambda^*(x_0)$ can be empty. The above F. John optimality condition is necessary for local optimality, i.e., $\Lambda^*(x_0) \neq \emptyset$, in two important cases, namely, when the space Y is finite dimensional or when the set K has a nonempty interior [24, 45].

If the multiplier α in (3.1) is nonzero, then we can take $\alpha = 1$ and hence the corresponding first order necessary condition becomes

$$D_x L(x_0, \lambda) = 0, \ \lambda \in N_K(G(x_0)). \tag{3.2}$$

Under Robinson's constraint qualification (2.11) the set $\Lambda(x_0)$ of Lagrange multipliers satisfying (3.2) is nonempty and bounded [28, 45]. When the set K is a convex *cone* and $y \in K$, the normal cone $N_K(y)$ can be written in the form $N_K(y) = \{y^* \in K^- : \langle y^*, y \rangle = 0\}$, where

$$K^- := \{y^* \in Y^* : \langle y^*, y \rangle \leq 0 \text{ for all } y \in K\}$$

is the polar (negative dual) cone of the cone K. In that case condition $\lambda \in N_K(G(x_0))$ becomes $\lambda \in K^-$ and $\langle \lambda, G(x_0) \rangle = 0$.

Let us finally recall that the cone

$$C(x_0) := \{h \in X : DG(x_0)h \in T_K(G(x_0)), Df(x_0)h \leq 0\} \tag{3.3}$$

is called the *critical cone* of the problem (P) at the point x_0. It represents those directions for which a first order linearization of (P) does not provide information about the optimality of x_0. It may be noted that when the set $\Lambda(x_0)$ of Lagrange multipliers is nonempty, then $Df(x_0)h \geq 0$ for any $h \in X$ satisfying $DG(x_0)h \in T_K(G(x_0))$. In such a case the inequality $Df(x_0)h \leq 0$ in the definition of the critical cone can be replaced by the equation $Df(x_0)h = 0$, which in turn is equivalent to $\langle \lambda, DG(x_0)h \rangle = 0$ for any $\lambda \in \Lambda(x_0)$.

With these preliminaries we may now state a second order necessary condition for optimality, which is based on the analysis of feasible parabolic paths of the form

$$x(t) = x_0 + th + \tfrac{1}{2}t^2 w + o(t^2), \tag{3.4}$$

where $t \geq 0$. This necessary condition, combined with the sufficient condition given in Theorem 3.2, will lead to the notion of second order regularity (studied in the next section) under which they become no gap second order optimality conditions.

The following result improves [12, Theorem 4.2], where a similar theorem is stated based on the *inner* second order tangent set. We should mention here an alternative approach suggested by Penot [27] based on the notion of second order compound tangent set, which is a variant of the concept of outer second order tangent set specifically tailored to derive no gap optimality conditions. In this sense we observe that the following result is contained in [27, Corollary 3.6]. However, we will show that under second order regularity, a condition covering many interesting situations, there is no need to resort to the more complicated (and less explicit) concept of compound tangent set, and therefore the following result will suffice for our purpose of stating no gap second order optimality conditions. For the sake of completeness we provide a direct proof which follows the lines of [12, Theorem 4.2].

THEOREM 3.1. *Let x_0 be a locally optimal solution of the problem* (P). *Suppose that Robinson's constraint qualification* (2.11) *holds. Then for all $h \in C(x_0)$ and any convex set $\mathcal{T}(h) \subset O^2_K(G(x_0), DG(x_0)h)$,*

$$\sup_{\lambda \in \Lambda(x_0)} \left\{ D^2_{xx} L(x_0, \lambda)(h,h) - \sigma(\lambda, \mathcal{T}(h)) \right\} \geq 0. \tag{3.5}$$

Proof. Note that if $\mathcal{T}(h) = \emptyset$, then $\sigma(\cdot, \mathcal{T}(h)) = -\infty$ and (3.5) trivially holds. Therefore we assume that the set $\mathcal{T}(h)$, and hence the set $O^2_K(G(x_0), DG(x_0)h)$, is nonempty.

We claim that the optimal value of the optimization problem

$$\begin{array}{ll} \text{Min}_{w \in X} & Df(x_0)w + D^2 f(x_0)(h,h) \\ \text{subject to} & DG(x_0)w + D^2 G(x_0)(h,h) \in O^2_K(G(x_0), DG(x_0)h) \end{array} \tag{3.6}$$

is nonnegative. Indeed if w is feasible for this problem, then using Proposition 2.3 we obtain $w \in O^2_\Phi(x_0, h)$, where $\Phi := G^{-1}(K)$. Therefore we can find a sequence $t_k \downarrow 0$ such that $x_k := x_0 + t_k h + \frac{1}{2} t_k^2 w + o(t_k^2) \in \Phi$. The sequence x_k is feasible for (P) and converges to the local minimum x_0, consequently $f(x_k) \geq f(x_0)$ for all k sufficiently large. By using the second order Taylor expansion we have

$$f(x_0) \leq f(x_k) = f(x_0) + t_k Df(x_0)h + \tfrac{1}{2} t_k^2 [Df(x_0)w + D^2 f(x_0)(h,h)] + o(t_k^2),$$

and since $Df(x_0)h = 0$ for any $h \in C(x_0)$, we obtain

$$Df(x_0)w + D^2 f(x_0)(h,h) \geq 0,$$

establishing our claim.

Consider now the following set $T(h) := \text{cl}\{\mathcal{T}(h) + T_K(G(x_0))\}$. This set is the topological closure of the sum of two convex sets and hence is convex. Moreover, it follows from the first inclusion of (2.10), and the fact that second order outer tangent sets are closed, that $T(h) \subset O^2_K(G(x_0), DG(x_0)h)$. Clearly if we replace the outer second order tangent set in (3.6) by its subset $T(h)$, the optimal value of the obtained optimization problem will be greater than or equal to the optimal value of (3.6), and hence the optimal value of the problem

$$\begin{array}{ll} \text{Min}_{w \in X} & Df(x_0)w + D^2 f(x_0)(h,h) \\ \text{subject to} & DG(x_0)w + D^2 G(x_0)(h,h) \in T(h) \end{array} \tag{3.7}$$

is nonnegative as well.

The optimization problem (3.7) is convex and its (parametric) dual (cf. [32], [5]) is

$$\operatorname*{Max}_{\lambda \in \Lambda(x_0)} \left\{ D^2_{xx} L(x_0, \lambda)(h, h) - \sigma(\lambda, T(h)) \right\}. \tag{3.8}$$

Indeed, the Lagrangian of (3.7) is

$$\mathcal{L}(w, \lambda) = D_x L(x_0, \lambda) w + D^2_{xx} L(x_0, \lambda)(h, h).$$

Since for any $z \in T(h)$ we have that $z + T_K(G(x_0)) \subset T(h)$, it follows that $\sigma(\lambda, T(h)) = +\infty$ for any $\lambda \notin [T_K(G(x_0))]^- = N_K(G(x_0))$. Therefore the effective domain of the parametric dual of (3.7) is contained in $\Lambda(x_0)$. The duality then follows. Moreover, Robinson's constraint qualification (2.11) implies that

$$DG(x_0)X - T_K(G(x_0)) = Y.$$

Since for any $z \in T(h)$ it follows that $z + T_K(G(x_0)) \subset T(h)$, we have that

$$z + DG(x_0)X - T(h) = Y.$$

Therefore (3.7) has a feasible solution and Robinson's constraint qualification for the problem (3.7) holds as well. Consequently there is no duality gap between (3.7) and its dual (3.8) (cf. [5]).

We obtain that the optimal value of (3.8) is nonnegative. Since $\mathcal{T}(h) \subset T(h)$, we have that $\sigma(\lambda, \mathcal{T}(h)) \leq \sigma(\lambda, T(h))$ and hence (3.5) follows, which completes the proof. □

Remarks. (i) As we mentioned earlier, the outer second order tangent set

$$O^2_K(G(x_0), DG(x_0)h)$$

can be nonconvex. However, when it is convex, one can use this set in the second order condition (3.5), providing a sharper necessary condition. In any case one can take $\mathcal{T}(h)$ to be the *inner* second order tangent set $T^2_K(G(x_0), DG(x_0)h)$. For such a choice of $\mathcal{T}(h)$, (3.5) coincides with the second order necessary condition obtained in [12, Theorem 4.2]. In general, however, the set $\mathcal{T}(h)$ could be taken larger than $T^2_K(G(x_0), DG(x_0)d)$ and therefore Theorem 3.1 is stronger.

(ii) Note that in the second order necessary condition the optimal value of (3.6) is nonnegative, irrespective of whether $O^2_K(G(x_0), DG(x_0)h)$ is convex.

(iii) If

$$0 \in O^2_K(G(x_0), DG(x_0)h)$$

for every $h \in C(x_0)$, in particular if the set K is *polyhedral*, then

$$O^2_K(G(x_0), DG(x_0)h) = T_{T_K(G(x_0))}(DG(x_0)h)$$

and $\sigma(\lambda, \mathcal{T}(h)) = 0$ for every $\lambda \in \Lambda(x_0)$ and $\mathcal{T}(h) := O^2_K(G(x_0), DG(x_0)h)$. Therefore in that case the "sigma term" in (3.5) vanishes. This happens in the case of nonlinear programming.

(iv) Let Σ be the set of sequences $\{t_n\}$ of positive numbers converging to zero. With any $s = \{t_n\} \in \Sigma$, $y \in K$, and $d \in T_K(y)$ we can associate the following second order tangent set:

$$T^{2,s}_K(y, d) := \left\{ w : \operatorname{dist}(y + t_n d + \tfrac{1}{2} t_n^2 w, K) = o(t_n^2) \right\}.$$

For any $s \in \Sigma$ the set $T_K^{2,s}(y,d)$ is convex and closed. It is clear that the intersection of $T_K^{2,s}(y,d)$ over all $s \in \Sigma$ is $T_K^2(y,d)$ and the union of $T_K^{2,s}(y,d)$ over all $s \in \Sigma$ is $O_K^2(y,d)$. A possible choice for $\mathcal{T}(h)$ is then $T_K^{2,s}(G(x_0), DG(x_0)h)$ for any $s \in \Sigma$.

(v) We can formulate the second order necessary condition (3.5) in the form

$$\inf_{\mathcal{T}(h)\in\mathcal{O}(h)} \sup_{\lambda\in\Lambda(x_0)} \left\{ D_{xx}^2 L(x_0,\lambda)(h,h) - \sigma(\lambda, \mathcal{T}(h)) \right\} \geq 0, \tag{3.9}$$

where $\mathcal{O}(h)$ denotes the set of all convex subsets of $O_K^2(G(x_0), DG(x_0)h)$. In particular, if we take all singleton subsets of $O_K^2(G(x_0), DG(x_0)h)$ (i.e., consisting from one point), then condition (3.9) implies the following necessary condition:

$$\inf_{y\in O_K^2(G(x_0),DG(x_0)h)} \sup_{\lambda\in\Lambda(x_0)} \left\{ D_{xx}^2 L(x_0,\lambda)(h,h) - \langle \lambda, y\rangle \right\} \geq 0. \tag{3.10}$$

If $\Lambda(x_0)$ is a singleton, say, $\Lambda(x_0) = \{\lambda_0\}$, then condition (3.10) becomes

$$D_{xx}^2 L(x_0,\lambda_0)(h,h) - \sigma(\lambda_0, O_K^2(G(x_0), DG(x_0)h)) \geq 0, \tag{3.11}$$

irrespective of whether $O_K^2(G(x_0), DG(x_0)h)$ is convex.

Definition 1. *Let $S \subset \Phi$ be a set of feasible points of the problem* (P) *such that $f(x) = f_0$ for all $x \in S$. It is said that the second order growth condition holds at S if there exist a constant $c > 0$ and a neighborhood N of S such that*

$$f(x) \geq f_0 + c\,[\operatorname{dist}(x,S)]^2 \text{ for all } x \in \Phi \cap N. \tag{3.12}$$

In particular, if $S = \{x_0\}$ is a singleton, the second order growth condition (3.12) takes the form

$$f(x) \geq f(x_0) + c\|x - x_0\|^2 \text{ for all } x \in \Phi \cap N, \tag{3.13}$$

which clearly implies that x_0 is a locally optimal solution of (P). Moreover, in this case (assuming always that Robinson's condition (2.11) holds) it follows easily that for any $h \in C(x_0)$ the optimal value of (3.6) is greater than or equal to $2c\|h\|^2$, so that the second order necessary condition (3.5) can be strengthened to strict inequality for all nonzero $h \in C(x_0)$.

The second order necessary condition (3.5) is based on *upper* estimates of the objective function along feasible parabolic curves of the form (3.4). In order to derive lower estimates, and hence to obtain second order *sufficient* conditions, we need an additional concept.

Definition 2. *Let $y \in K$, $d \in T_K(y)$, and consider a continuous linear mapping $M : X \to Y$. We say that a closed set $\mathcal{A}_{K,M}(y,d) \subset Y$ is an upper second order approximation set for K at the point y in the direction d and with respect to M, if for any sequence $y_k \in K$ of the form $y_k := y + t_k d + \frac{1}{2}t_k^2 r_k$, where $t_k \downarrow 0$ and $r_k = Mw_k + a_k$ with $\{a_k\}$ being a convergent sequence in Y and $\{w_k\} \subset X$ satisfying $t_k w_k \to 0$, the following condition holds:*

$$\lim_{k\to\infty} \operatorname{dist}(r_k, \mathcal{A}_{K,M}(y,d)) = 0. \tag{3.14}$$

If the above holds for any X and M, i.e., (3.14) *is satisfied for any sequence*

$$y + t_k d + \tfrac{1}{2}t_k^2 r_k \in K$$

such that $t_k r_k \to 0$, we omit M and say that the set $\mathcal{A}_K(y,d)$ is an upper second order approximation set for K at the point y in the direction d.

Let us make the following observations. The above definition is aimed at providing a sufficiently large set $\mathcal{A}_K(y,d)$ such that if $y+td+\varepsilon(t)$ is a curve in K tangential to d with $\varepsilon(t)=o(t)$, then the second order remainder $r(t):=\varepsilon(t)/(\frac{1}{2}t^2)$ tends to $\mathcal{A}_K(y,d)$ as $t\downarrow 0$. Note that this remainder $r(t)$ and its sequential analogue $r_k = r(t_k)$ can be unbounded. The additional complication of considering the linear mapping M, etc. is needed for technical reasons, as is typically encountered in infinite dimensional functional spaces.

The upper second order approximation set $\mathcal{A}_K(y,d)$ is not unique. Clearly, if $\mathcal{A}_K(y,d)\subset B$, then B is also an upper second order approximation set. Since if $y\in K$, $d\in T_K(y)$ and $y+d+w\in K$ imply $d+w\in T_K(y)$ and hence $w\in T_{T_K(y)}(d)$, it follows that the set $T_{T_K(y)}(d)$ is always an upper second order approximation set. It is also not difficult to see from the definitions that the outer second order tangent set $O_K^2(y,d)$ is included in any upper second order approximation set $\mathcal{A}_K(y,d)$.

THEOREM 3.2. *Let x_0 be a feasible point of the problem* (P) *satisfying the first order (F. John–type) optimality condition* (3.1)*. Let every $h\in C(x_0)$ correspond to an upper second order approximation set $\mathcal{A}(h):=\mathcal{A}_{K,M}(y_0,d)$ for the set K at the point $y_0:=G(x_0)$ in the direction $d:=DG(x_0)h$ and with respect to the linear mapping $M:=DG(x_0)$, and suppose that the following second order condition is satisfied:*

$$\sup_{(\alpha,\lambda)\in\Lambda^*(x_0)}\left\{D_{xx}^2 L^*(x_0,\alpha,\lambda)(h,h)-\sigma\left(\lambda,\mathcal{A}(h)\right)\right\}>0 \tag{3.15}$$

for all $h\in C(x_0)\setminus\{0\}$. Then the second order growth condition (3.13) *holds at x_0, and hence x_0 is a strict locally optimal solution of* (P)*.*

Proof. We argue by contradiction. Suppose that the second order growth condition does not hold at x_0. Then there exists a sequence of feasible points $x_k\in\Phi$, $x_k\neq x_0$, converging to x_0 and such that

$$f(x_k)\le f(x_0)+o(t_k^2), \tag{3.16}$$

where $t_k:=\|x_k-x_0\|$. Since the space X is finite dimensional, and hence bounded closed sets in X are compact, we can assume that $h_k:=(x_k-x_0)/t_k$ converges to a vector $h\in X$. Clearly $\|h\|=1$ and hence $h\neq 0$. By using first order Taylor expansions, we obtain from $G(x_k)\in K$ that $DG(x_0)h\in T_K(G(x_0))$ and from (3.16) that $Df(x_0)h\le 0$. Therefore it follows that $h\in C(x_0)$.

By a second order Taylor expansion of $G(x_k)$ at x_0, we have that

$$G(x_k)=y_0+t_k d+\tfrac{1}{2}t_k^2\left(DG(x_0)w_k+D^2G(x_0)(h,h)\right)+o(t_k^2),$$

where $y_0:=G(x_0)$, $d:=DG(x_0)h$, and $w_k:=2t_k^{-2}(x_k-x_0-t_k h)$. Note that $x_k-x_0-t_k h=o(t_k)$ and hence $t_k w_k\to 0$. Together with the definition of upper second order approximation set this implies that

$$DG(x_0)w_k+D^2G(x_0)(h,h)\in\mathcal{A}(h)+o(1)B_Y. \tag{3.17}$$

We also have that

$$f(x_k)=f(x_0)+t_k Df(x_0)h+\tfrac{1}{2}t_k^2\left(Df(x_0)w_k+D^2f(x_0)(h,h)\right)+o(t_k^2)$$

so that using (3.16) and (3.17) one can find a sequence $\varepsilon_k\to 0$ such that

$$\begin{cases} 2t_k^{-1}Df(x_0)h+(Df(x_0)w_k+D^2f(x_0)(h,h))\le\varepsilon_k, \\ DG(x_0)w_k+D^2G(x_0)(h,h)\in\mathcal{A}(h)+\varepsilon_k B_Y. \end{cases} \tag{3.18}$$

By (3.15) there exists $(\alpha, \lambda) \in \Lambda^*(x_0)$ such that

$$D^2_{xx}L^*(x_0, \alpha, \lambda)(h, h) - \sigma(\lambda, \mathcal{A}(h)) \geq \kappa \tag{3.19}$$

for some $\kappa > 0$. It follows from the second condition in (3.18) that

$$\langle \lambda, DG(x_0)w_k + D^2G(x_0)(h, h)\rangle \leq \sigma(\lambda, \mathcal{A}(h) + \varepsilon_k B_Y) = \sigma(\lambda, \mathcal{A}(h)) + \varepsilon_k \|\lambda\|.$$

Also $\alpha \geq 0$, and if $\alpha \neq 0$, then there exists a Lagrange multiplier and hence $Df(x_0)h = 0$. In any case $\alpha Df(x_0)h = 0$, and hence we obtain from (3.18) and (3.19) that

$$\begin{array}{rcl} 0 & \geq & \alpha(2t_k^{-1}Df(x_0)h + Df(x_0)w_k + D^2f(x_0)(h, h) - \varepsilon_k) \\ & & +\langle \lambda, DG(x_0)w_k + D^2G(x_0)(h, h)\rangle - \sigma(\lambda, \mathcal{A}(h)) - \varepsilon_k\|\lambda\| \\ & = & D^2_{xx}L^*(x_0, \alpha, \lambda)(h, h) - \sigma(\lambda, \mathcal{A}(h)) - \varepsilon_k(\alpha + \|\lambda\|) \\ & \geq & \kappa - \varepsilon_k(\alpha + \|\lambda\|). \end{array}$$

Since $\varepsilon_k \to 0$ we obtain a contradiction which completes the proof. □

Let us first observe that *finite* dimensionality of the space X was used in the derivation of the above second order *sufficient* condition, while the corresponding second order necessary condition did not require that assumption.

If the set $\Lambda(x_0)$ of Lagrange multipliers is nonempty, then the second order sufficient condition (3.15) is equivalent to

$$\sup_{\lambda \in \Lambda(x_0)} \left\{ D^2_{xx}L(x_0, \lambda)(h, h) - \sigma(\lambda, \mathcal{A}(h)) \right\} > 0 \text{ for all } h \in C(x_0) \setminus \{0\}. \tag{3.20}$$

Also, as was mentioned earlier, the set $\mathcal{Z}(h) := T_{T_K(G(x_0))}(DG(x_0)h)$ is always an upper second order approximation set. Furthermore,

$$\sigma(\lambda, \mathcal{Z}(h)) = \begin{cases} 0 & \text{if } \lambda \in T_K(G(x_0)) \text{ and } \langle \lambda, DG(x_0)h\rangle = 0, \\ +\infty & \text{otherwise.} \end{cases}$$

Therefore for that choice of upper second order approximation set, the second order sufficient condition (3.15) takes the form

$$\sup_{(\alpha,\lambda) \in \Lambda^*(x_0)} D^2_{xx}L^*(x_0, \alpha, \lambda)(h, h) > 0 \text{ for all } h \in C(x_0) \setminus \{0\}. \tag{3.21}$$

We obtain the following result.

COROLLARY 3.3. *Let x_0 be a feasible point of the problem* (P) *satisfying the first order (F. John–type) optimality condition* (3.1). *Suppose that the second order sufficient condition* (3.21) *is satisfied. Then the second order growth condition* (3.13) *holds at x_0.*

If the set $\Lambda(x_0)$ of Lagrange multipliers is nonempty, then one can replace $\Lambda^*(x_0)$ in (3.21) by $\Lambda(x_0)$. In that form the second order sufficient condition (3.21) is well known [3, 19]. Moreover, if the set K is polyhedral, i.e., in the case of nonlinear programming, as we mentioned earlier the sigma term vanishes in the corresponding second order necessary condition, which leads to a pair of no gap second order conditions in that case.

4. Second order regularity. Comparing the necessary and sufficient conditions given in (3.5) and (3.20), respectively, one may observe that besides the change from weak to strict inequality, the set $\mathcal{T}(h) \subset O_K^2(G(x_0), DG(x_0)h)$ in the former was replaced by a possibly larger set $\mathcal{A}(h)$. Now, conditions (3.15) and (3.20) become stronger if one can take a smaller second order approximation set $\mathcal{A}(h)$. In particular, if $O_K^2(G(x_0), DG(x_0)h)$ is an upper second order approximation set, condition (3.20) becomes the strongest possible by taking $\mathcal{A}(h) = O_K^2(G(x_0), DG(x_0)h)$. In that case, provided $O_K^2(G(x_0), DG(x_0)h)$ is convex, the gap between (3.5) and (3.20) reduces to the difference between weak and strict inequality, and hence we obtain a pair of no gap second order conditions. This motivates the following definition.

DEFINITION 3. *We say that the set K is outer second order regular at a point $y \in K$ in a direction $d \in T_K(y)$ and with respect to a linear mapping $M : X \to Y$ if for any sequence $y_n \in K$ of the form $y_n := y + t_n d + \frac{1}{2}t_n^2 r_n$, where $t_n \downarrow 0$ and $r_n = Mw_n + a_n$ with $\{a_n\}$ being a convergent sequence in Y and $\{w_n\}$ being a sequence in X satisfying $t_n w_n \to 0$, the following condition holds:*

$$\lim_{n\to\infty} \operatorname{dist}\left(r_n, O_K^2(y,d)\right) = 0. \tag{4.1}$$

If K is outer second order regular at $y \in K$ in every direction $d \in T_K(y)$ and with respect to any X and M, we say that K is outer second order regular at y. If, in addition, $O_K^2(y,d) = T_K^2(y,d)$ for every $d \in T_K(y)$, we say that K is second order regular at y.

Outer second order regularity means that the outer second order tangent set $O_K^2(y,d)$ provides an upper second order approximation for K at y in direction d. If in addition the outer and inner second order tangent sets coincide, we simply talk about second order regularity. Second order regularity means that if $y + td + \varepsilon(t)$ is a curve in K tangential to d with $\varepsilon(t) = o(t)$, then $r(t) := \varepsilon(t)/(\frac{1}{2}t^2)$ is arbitrarily close to $T_K^2(y,d)$ as $t \downarrow 0$. Loosely speaking, second order regular sets are the appropriate ones for second order optimality conditions in the sense that there is no gap between the corresponding second order necessary and sufficient conditions; see the following theorem.

THEOREM 4.1. *Let x_0 be a feasible point of* (P) *satisfying the first order necessary condition* (3.2). *Suppose that Robinson's constraint qualification* (2.11) *holds, that for every $h \in C(x_0)$ the set K is outer second order regular at $G(x_0)$ in direction $DG(x_0)h$ and with respect to $M := DG(x_0)$, and that the outer second order tangent set $O_K^2(G(x_0), DG(x_0)h)$ is convex. Then the second order growth condition* (3.13) *holds if and only if the second order sufficient condition* (3.20) *is satisfied with $\mathcal{A}(h) = O_K^2(G(x_0), DG(x_0)h)$.*

Proof. The implication (3.20)⇒(3.13) follows from Theorem 3.2, while the converse is a consequence of Theorem 3.1 and the discussion following the statement of equation (3.13). □

Recall that the inner second order tangent sets are always convex, and hence in the case $O_K^2(G(x_0), DG(x_0)h) = T_K^2(G(x_0), DG(x_0)h)$ the assumed convexity of the outer second order tangent set automatically holds.

At first glance the second order regularity concept, introduced in Definition 3, may seem to be rather technical. Nevertheless it is possible to verify the second order regularity in a number of particular situations. It holds, for example, when $0 \in T_K^2(y,d)$ for every $d \in T_K(y)$, since then $T_K^2(y,d) = T_{T_K(y)}(d)$. This occurs, for instance, when K is a polyhedral set. We discuss in the next subsections several other situations where the second order regularity holds. In particular, we show that the cone $\mathcal{S}^n_+$ of $n \times n$ positive semidefinite matrices is second order regular (at every point $y \in \mathcal{S}^n_+$).

4.1. Sets defined by smooth and convex constraints. Second order regularity is preserved when taking inverse images through twice continuously differentiable mappings satisfying Robinson's constraint qualification.

PROPOSITION 4.2. *Let K be a closed convex subset of Y and $G : X \to Y$ be a twice continuously differentiable mapping. If Robinson's constraint qualification* (2.11) *holds and K is (outer) second order regular at $G(x_0)$ in the direction $DG(x_0)h$ with respect to the linear mapping $M := DG(x_0)$, then the set $G^{-1}(K)$ is (outer) second order regular at x_0 in the direction h.*

Proof. Let $x_k := x_0 + t_k h + \frac{1}{2}t_k^2 r_k \in G^{-1}(K)$ be such that $t_k \downarrow 0$ and $t_k r_k \to 0$. By Proposition 2.3 and the Robinson–Ursescu stability theorem, we obtain for some constant L and all k large enough

$$\begin{aligned}&\operatorname{dist}\Big(r_k, O^2_{G^{-1}(K)}(x_0,h)\Big)\\ &\quad = \operatorname{dist}\big(r_k, DG(x_0)^{-1}\big[O^2_K(G(x_0), DG(x_0)h) - D^2G(x_0)(h,h)\big]\big)\\ &\quad \le L \operatorname{dist}\big(DG(x_0)r_k + D^2G(x_0)(h,h), O^2_K(G(x_0), DG(x_0)h)\big).\end{aligned}$$

Now, a second order expansion of $G(x_k)$ gives

$$G(x_k) = G(x_0) + t_k DG(x_0)h + \tfrac{1}{2}t_k^2\big(DG(x_0)r_k + D^2G(x_0)(h,h)\big) + o(t_k^2).$$

Since $G(x_k) \in K$, the assumed (outer) second order regularity of K implies

$$\operatorname{dist}\big(DG(x_0)r_k + D^2G(x_0)(h,h), O^2_K(G(x_0), DG(x_0)h)\big) \to 0$$

and therefore $\operatorname{dist}(r_k, O^2_{G^{-1}(K)}(x_0,h)) \to 0$, as had to be proved. □

Consider the set

$$F := \{x \in X : g_i(x) \le 0,\ i = 1,\dots,p;\ \ h_j(x) = 0,\ j = 1,\dots,q\},$$

defined by a finite number of constraints. Suppose that the functions g_i and h_j are twice continuously differentiable. As a straightforward consequence of Proposition 4.2 and the fact that polyhedral sets are second order regular, we obtain that the set F is second order regular at every point $x_0 \in F$ satisfying the Mangasarian–Fromovitz constraint qualification. Another direct consequence of Proposition 4.2 is the following result.

COROLLARY 4.3. *Let $K_1, \dots, K_n$ be closed convex sets which are second order regular at a point $y_0 \in K_1 \cap \cdots \cap K_n$ in a direction $d \in T_{K_1}(y_0) \cap \cdots \cap T_{K_n}(y_0)$. If there exists a point in K_n which belongs to the interior of the remaining K_i's, $i = 1, \dots, n-1$, then the intersection $K_1 \cap \cdots \cap K_n$ is second order regular at y_0 in the direction d.*

Proof. It suffices to apply Proposition 4.2 with $G : Y \to Y^n$ given by $G(y) = (y, \dots, y)$ and $K = K_1 \times \cdots \times K_n$. It is easily seen that K is second order regular at $(y_0, \dots, y_0)$ in the direction $(d, \dots, d)$.

In order to check Robinson's constraint qualification we take $\bar{y} \in Y$ and $\varepsilon > 0$ such that $\bar{y} \in K_n$ and $\bar{y} + 2\varepsilon B_Y \subset K_1 \cap \cdots \cap K_{n-1}$. If $u_1, \dots, u_n \in \varepsilon B_Y$, letting $y = \bar{y} + u_n$ we have $k_i := y - u_i \in \bar{y} + 2\varepsilon B_Y \subset K_i$ for all $i = 1, \dots, n-1$. Therefore, if we set $k_n := \bar{y} \in K_n$ we have $u_i = y - k_i \in y - K_i$ for all $i = 1, \dots, n$ and then $[\varepsilon B_Y]^n \subset G(Y) - K$, which proves Robinson's constraint qualification. □

Returning to the case of sets defined by inequality constraints, we observe that when the constraint functions are convex one may relax the differentiability assumptions.

PROPOSITION 4.4. *Let $K := \{y : h(y) \le 0\}$, where $h(\cdot)$ is a convex function which is continuous at a point y_0. Suppose that the Slater condition holds and that $h(y_0) = 0$. Then K is outer second-order regular at y_0 if and only if, for any $d \in T_K(y_0)$ satisfying $h'(y_0, d) = 0$ and any path $y(t) \in K$ of the form $y(t) = y_0 + td + \frac{1}{2}t^2 r(t)$, $t \ge 0$, with $tr(t) \to 0$ as $t \downarrow 0$, the inequality*

$$\limsup_{t \downarrow 0} h''_-(y_0; d, r(t)) \le 0 \tag{4.2}$$

holds.

Proof. Since h is convex and continuous at y_0, it is directionally differentiable at y_0. Consider a direction $d \in T_K(y_0)$ and a sequence $y_k := y_0 + t_k d + \frac{1}{2}t_k^2 r_k \in K$ with $t_k \downarrow 0$ and $t_k r_k \to 0$. It follows from $d \in T_K(y_0)$ that $h'(y_0, d) \le 0$. Since $h'(y_0, d) < 0$ implies that $O_K^2(y_0, d) = Y$, we only need to consider the case $h'(y_0, d) = 0$.

Because of the Slater condition, there is a point $\bar{y} \in Y$ such that $h(\bar{y}) < 0$. By convexity of $h(\cdot)$ we then have that $h(y_0 + t(\bar{y} - y_0)) < 0$ for any $t \in (0, 1)$ and hence a point $\bar{y}$ where $h(\bar{y}) < 0$ can be chosen arbitrarily close to y_0. Therefore we can assume that $h(\cdot)$ is continuous at $\bar{y}$.

Assume that (4.2) holds. For $\alpha > 0$ let $w_\alpha := r_k + \alpha(\bar{y} - y_0)$. By convexity we get for all $t > 0$ small enough

$$h(y_0 + td + \tfrac{1}{2}t^2 w_\alpha) \le (1 - \tfrac{1}{2}\alpha t^2)h(y_0 + td + \tfrac{1}{2}t^2 r_k) + \tfrac{1}{2}\alpha t^2 h(\bar{y} + td + \tfrac{1}{2}t^2 r_k).$$

Since $h(y_0) = 0$ and $h'(y_0, d) = 0$, dividing by $\frac{1}{2}t^2$ and letting $t \to 0^+$ we deduce

$$h''_-(y_0; d, w_\alpha) \le h''_-(y_0; d, r_k) + \alpha h(\bar{y}),$$

and by virtue of (4.2) we deduce $h''_-(y_0; d, r_k + \alpha(\bar{y} - y_0)) < 0$ for all k sufficiently large. Proposition 2.1 implies that $r_k + \alpha(\bar{y} - y_0) \in O_K^2(y_0, d)$ so that

$$\limsup_{k \to \infty} \ \operatorname{dist}(r_k, O_K^2(y_0, d)) \le \alpha \|\bar{y} - y_0\|.$$

Since α can be made arbitrarily small, we obtain that K is second order regular.

Conversely, assume that K is second order regular. Let $t_k \downarrow 0$ be a sequence through which the upper limit (4.2) is attained as a limit, and let $r_k := r(t_k)$. Set $\varepsilon_k := \operatorname{dist}(r_k, O_K^2(y_0, d)) + 1/k$, so that $\varepsilon_k \to 0$, and choose $\tilde{r}_k \in O_K^2(y_0, d)$ such that $\|r_k - \tilde{r}_k\| < \varepsilon_k$. Since ε_k tends to 0, with no loss of generality we may assume that for all k we have $\bar{y} + 2\varepsilon_k \alpha^{-1} B_Y \subset K$. Choose a sequence $\tau_\ell \downarrow 0$ such that $y_0 + \tau_\ell d + \frac{1}{2}\tau_\ell^2 \tilde{r}_k + o(\tau_l^2) \in K$ and therefore $y_0 + \tau_\ell d + \frac{1}{2}\tau_\ell^2 r_k \in K + \frac{1}{2}\varepsilon_k \tau_\ell^2 B_Y$. Then, for all $\alpha > 0$ and $w_\alpha := r_k + \alpha(\bar{y} - y_0)$ we get

$$\begin{aligned} y_0 + \tau_\ell d + \tfrac{1}{2}\tau_\ell^2 w_\alpha &= (1 - \tfrac{1}{2}\alpha\tau_\ell^2)(y_0 + \tau_\ell d + \tfrac{1}{2}\tau_\ell^2 r_k) + \tfrac{1}{2}\alpha\tau_\ell^2(\bar{y} + \tau_\ell d + \tfrac{1}{2}\tau_\ell^2 r_k) \\ &\subset (1 - \tfrac{1}{2}\alpha\tau_\ell^2)(K + \tfrac{1}{2}\varepsilon_k \tau_\ell^2 B_Y) + \tfrac{1}{2}\alpha\tau_\ell^2(\bar{y} + \tau_\ell d + \tfrac{1}{2}\tau_\ell^2 r_k) \\ &= (1 - \tfrac{1}{2}\alpha\tau_\ell^2)K + \tfrac{1}{2}\alpha\tau_\ell^2 \left[\bar{y} + \tau_\ell d + \tfrac{1}{2}\tau_\ell^2 r_k + (1 - \tfrac{1}{2}\alpha\tau_\ell^2)\frac{\varepsilon_k}{\alpha} B_Y\right]. \end{aligned}$$

Since $\bar{y} + 2\varepsilon_k \alpha^{-1} B_Y \subset K$ we deduce $y_0 + \tau_\ell d + \frac{1}{2}\tau_\ell^2 w_\alpha \in K$. By Proposition 2.1, $h''_-(y_0, d, w_\alpha) \le 0$. Since h is continuous at y_0, it is locally Lipschitz continuous.

Therefore, $h''_-(y_0, d, \cdot)$ is globally Lipschitz continuous with the same constant, say, L, and $h''_-(y_0, d, r_k) \le L\|w_\alpha - r_k\| = \alpha L\|\bar{y} - y_0\|$, from which

$$\limsup_{t\downarrow 0} h''_-(y_0, d, r(t)) = \lim_k h''_-(y_0, d, r_k) \le \alpha L\|\bar{y} - y_0\|.$$

Since α may be taken arbitrarily small, the conclusion follows. □

Let us derive now some criteria which allow us to check condition (4.2), assuming that h is convex and continuous at y_0. We first observe that this condition is satisfied whenever

$$h(y_0 + td + \tfrac{1}{2}t^2 r(t)) \ge h(y_0) + th'(y_0, d) + \tfrac{1}{2}t^2 h''_-(y_0; d, r(t)) + o(t^2) \tag{4.3}$$

for all $r(t)$ such that $\mathrm{tr}(t) \to 0$ as $t \downarrow 0$. This holds, for instance, when h is twice continuously differentiable.

A nondifferentiable (at zero) function satisfying (4.3) is the Euclidean norm $h(y) := \|y\|$. Many problems of robust optimization boil down to the minimization of a sum of Euclidean norms subject to linear constraints (see, e.g., [4]), say, $\sum_{i=1}^m \|A_i x\|$, where A_i are $q_i \times n$ matrices. Let us consider for simplicity the unconstrained problem. Introducing slack variables z_i, the problem reduces to the minimization of $\sum_{i=1}^m z_i$, subject to the constraints

$$\|A_i x\| - z_i \le 0, \quad 1 \le i \le m.$$

Set $h(y) = \|y\|$. Note that this is a twice continuously differentiable function at $y_0 \ne 0$. If $y_0 = 0$, we obtain $h'(0, d) = \|d\|$. If $d = 0$, then $h''(0; d, w) = \|w\|$, otherwise $h''(0; d, w) = \langle d, w\rangle / \|d\|$. In both cases (4.3) is easily checked. Therefore $h_i(x, z) = \|A_i x\| - z_i$ also satisfies (4.3) and the Slater condition is trivially satisfied.

Note also that if the functions $\{h_i : i = 1, ..., m\}$ are convex and second order directionally differentiable and satisfy (4.2), then $h := \sum_{i=1}^n h_i$ satisfies (4.2) as well. On the other hand it follows from Corollary 4.3 that the set

$$K = \{y \in Y : h_i(y) \le 0,\ i = 1, ..., m\}$$

is also second order regular, provided the Slater condition holds for the function $h(y) := \max\{h_i(y) : 1 \le i \le m\}$. This result can also be derived directly, as follows. Let y_0 be such that $h(y_0) = 0$. It is not difficult to show that $h(\cdot)$ is second order directionally differentiable with

$$h'(y_0, d) = \max\{h'_i(y_0, d) : i \in I_1(y_0)\},$$
$$h''(y_0; d, w) = \max\{h''_i(y_0; d, w) : i \in I_2(y_0, d)\},$$

where $I_1(y) := \{i : h_i(y) = h(y)\}$ and $I_2(y) := \{i \in I_1(y) : h'_i(y, d) = h'(y, d)\}$. Since $h_i(\cdot)$ satisfy (4.2), we have that for $y(t) := y_0 + td + \frac{1}{2}t^2 r(t)$, such that $h(y(t)) \le 0$, $\mathrm{tr}(t) \to 0$ and for d satisfying $h'(y_0, d) = 0$,

$$h''(y_0; d, r(t)) = \max_{i \in I_2(y_0, d)} h''_i(y_0; d, r(t)) \le o(t^2).$$

It then follows, assuming the Slater condition holds (i.e., there exists $\bar{y}$ such that $h_i(\bar{y}) < 0$, $i = 1, ..., m$), that the set K is second order regular with

$$T_K^2(y_0, d) = O_K^2(y_0, d) = \{w \in Y : h''_i(y_0, d, w) \le 0,\ i \in I_2(y_0, d)\}.$$

4.2. Semi-infinite and semidefinite programming. Let us consider now the case of semi-infinite programming with $Y := C(\Omega)$ and $K := C_+(\Omega)$ and with Ω being a compact metric space. For a function $y \in C_+(\Omega)$ its *contact set* is defined as

$$\Delta(y) := \{\omega \in \Omega : y(\omega) = 0\}. \tag{4.4}$$

It is well known that $d \in T_K(y)$ if and only if $d(\omega) \geq 0$ for all $\omega \in \Delta(y)$ (e.g., [39]). Denote

$$\Delta^*(y, d) := \{\omega \in \Delta(y) : d(\omega) = 0\}. \tag{4.5}$$

Note that if the set $\Delta^*(y, d)$ is empty, then d belongs to the interior of $T_K(y)$, and hence in that case $T_K^2(y, d) = Y$.

Suppose that Ω is a smooth compact manifold of finite dimension n. Consider a twice continuously differentiable function $y \in K$ with a nonempty contact set and a function $d \in T_K(y)$. A general formula for $T_K^2(y, d)$ is given in [14]. We derive now a particular case of that formula by direct arguments in the case where $\Delta(y)$ is a smooth submanifold of Ω. Moreover, we show that in such a case the second order regularity condition holds. These derivations are similar to the analyses in [38] and [5, Part III].

Since Ω is a smooth manifold, by using a local system of coordinates we identify an open neighborhood of a point $\bar{\omega} \in \Omega$ with an open subset of $\mathbb{R}^n$. Such an identification will not effect our local analysis and will simplify the presentation. Moreover, since $\Delta(y)$ is a smooth submanifold of Ω, for each $\bar{\omega} \in \Delta(y)$ we can choose such a local system of coordinates that $\Delta(y)$ is locally represented by a linear subspace of $\mathbb{R}^n$ in that system of coordinates. We denote by $T_{\Delta(y)}(\omega) \subset \mathbb{R}^n$ the tangent space to $\Delta(y)$ at $\omega \in \Delta(y)$ and by $N(\omega)$ its normal complement in $\mathbb{R}^n$, i.e., $N(\omega)$ is a linear space orthogonal to $T_{\Delta(y)}(\omega)$ and such that $T_{\Delta(y)}(\omega) + N(\omega) = \mathbb{R}^n$. Due to the above choice of local coordinates, these sets $T_{\Delta(y)}(\omega)$ and $N(\omega)$ are constant in the chosen system of local coordinates.

For a point $\omega \in \Omega$ we define its projection onto $\Delta(y)$ to be a point $\hat{\omega} \in \Delta(y)$ closest to ω with respect to the Euclidean distance in the chosen system of coordinates of Ω. This operation is well defined in the vicinity of $\bar{\omega}$ and of course depends on the choice of a local system of coordinates. Let $V(\omega)$ be a matrix whose columns form a basis of the linear space $N(\omega)$. Consider the following second order growth condition: for any $\bar{\omega} \in \Delta(y)$, there exists a local system of coordinates of the type described above such that

$$y(\omega) \geq c \operatorname{dist}(\omega, \Delta(y))^2 \text{ for all } \omega \in \Omega \cap \mathcal{N} \tag{4.6}$$

for some $c > 0$ and a neighborhood $\mathcal{N}$ of $\bar{\omega}$. Note that this condition does not depend on the system of coordinates (although the value of the constant c does of course) and is satisfied if and only if the matrix

$$U(\omega) := V(\omega)^T \nabla^2 y(\omega) V(\omega) \tag{4.7}$$

is positive definite for every $\omega \in \Delta(y)$ (see [38]).

THEOREM 4.5. *Let $y \in K := C_+(\Omega)$ be a twice continuously differentiable function, and let $d \in T_K(y)$ be continuously differentiable. Suppose that the set Ω is a smooth compact manifold, that $\Delta(y)$ is a smooth submanifold of Ω, and that the second order growth condition* (4.6) *holds for some $c > 0$ and with $\mathcal{N}$ being a neighborhood of*

$\Delta(y)$. Then the set K is second order directionally differentiable at y in the direction d with

$$T_K^2(y,d) = \left\{h \in C(\Omega) : h(\omega) \geq A(\omega)^T [U(\omega)]^{-1} A(\omega) \text{ for all } \omega \in \Delta^*(y,d)\right\}, \tag{4.8}$$

where $A(\omega) := V(\omega)^T \nabla d(\omega)$ and $U(\omega)$ is given in (4.7).

Moreover, let $M(x) := \sum_{i=1}^m x_i \psi_i(\cdot)$ be a linear mapping from $\mathbb{R}^m$ into $C(\Omega)$ such that the functions $\psi_i(\cdot)$, $i = 1, ..., m$, are Lipschitz continuous on Ω. Then the set K is second order regular at y in the direction d and with respect to M.

Proof. We already observed that when $\Delta^*(y,d)$ is empty we have $O_K^2(y,d) = T_K^2(y,d) = Y$ and the result holds trivially, so we may also assume that $\Delta^*(y,d) \neq \emptyset$.

Consider a path $\bar{y}_t(\cdot) := y(\cdot) + td(\cdot) + \frac{1}{2}t^2 h(\cdot)$ and the corresponding min-function $\nu(t) := \min_{\omega \in \Omega} \bar{y}_t(\omega)$. Since $\operatorname{dist}(\bar{y}_t, K) = \max\{0, -\nu(t)\}$, we have $h \in T_K^2(y,d)$ if and only if $\liminf_{t \downarrow 0} \nu(t)/t^2 \geq 0$ and $h \in O_K^2(y,d)$ if and only if $\limsup_{t\downarrow 0} \nu(t)/t^2 \geq 0$. We shall prove that in fact the limit $\lim_{t \downarrow 0} \nu(t)/t^2$ exists so that both second order tangent sets coincide.

Let $\bar{\omega}_t$ be a minimizer of $\bar{y}_t(\omega)$ over Ω, and let the sequence $t_n \to 0^+$ be such that $t^{-2}\nu(t)$ attains its lower limit. Let us denote $\bar{\omega}^n := \bar{\omega}_{t_n}$. Extracting if necessary a subsequence, we can assume that $\bar{\omega}^n \to \bar{\omega}^0 \in \Omega$. Since $\Delta^*(y,d) \neq \emptyset$, we have $\nu(t_n) \leq O(t_n^2)$, from which it follows that $\bar{\omega}^0 \in \Delta^*(y,d)$ (see [38]).

For n large enough, $\bar{\omega}^n$ can be described in terms of a local system of coordinates containing $\bar{\omega}^0$ in which the submanifold $\Delta(y)$ coincides with an affine space. To avoid heavy notation we will identify elements of Ω close to $\bar{\omega}^0$ with the corresponding vector of coordinates. Denote by $\hat{\omega}^n$ the projection of ω^n onto $\Delta(y)$ (in the given local system). Then $\delta^n := t_n^{-1}(\bar{\omega}^n - \hat{\omega}^n)$ is orthogonal to $\Delta(y)$ at the point $\hat{\omega}^n$, i.e., $\delta^n \in N(\hat{\omega}^n)$. Because of the second order growth condition (4.6) we get

$$\|\bar{\omega}^n - \hat{\omega}^n\| = \operatorname{dist}(\bar{\omega}^n, \Delta(y)) = O(t_n). \tag{4.9}$$

By expanding $\bar{y}^n(\bar{\omega}^n)$ at $\hat{\omega}^n$, and since $y(\hat{\omega}^n) = 0$ and $\nabla y(\hat{\omega}^n) = 0$, we obtain

$$\nu(t_n) = \bar{y}^n(\bar{\omega}^n) = \bar{y}^n(\hat{\omega}^n) + \tfrac{1}{2}t_n^2 \nabla^2 y(\hat{\omega}^n)(\delta^n, \delta^n) + t_n^2 \nabla d(\hat{\omega}^n)\delta^n + o(t_n^2).$$

Since $y(\hat{\omega}^n) = 0$ and $d(\hat{\omega}^n) \geq 0$, it follows that

$$\nu(t_n) \geq \tfrac{1}{2}t_n^2 h(\hat{\omega}^n) + \tfrac{1}{2}t_n^2 \nabla^2 y(\hat{\omega}^n)(\delta^n, \delta^n) + t_n^2 \nabla d(\hat{\omega}^n)\delta^n + o(t_n^2). \tag{4.10}$$

Since $\hat{\omega}^n \to \hat{\omega}^0$, the continuity of the mapping

$$\omega \mapsto \min_{\delta \in N(\omega)} \{h(\omega) + \nabla^2 y(\omega)(\delta, \delta) + 2\nabla d(\omega)\delta\}$$

leads to

$$\liminf_{t \downarrow 0} \frac{\nu(t)}{t^2/2} \geq \min_{\omega \in \Delta^*(y,d)} \min_{\delta \in N(\omega)} \{h(\omega) + \nabla^2 y(\omega)(\delta, \delta) + 2\nabla d(\omega)\delta\}. \tag{4.11}$$

On the other hand, for any $\omega \in \Delta^*(y,d)$ and $\delta \in N(\omega)$ we have that $\nu(t) \leq \bar{y}_t(\omega + t\delta)$. Again using local coordinates, by expanding the right-hand side of this inequality, and since $y(\omega) = 0$, $\nabla y(\omega) = 0$, $d(\omega) = 0$, and $h(\omega + t\delta) = h(\omega) + o(1)$, we obtain that

$$\nu(t) \leq \tfrac{1}{2}t^2 \{h(\omega) + \nabla^2 y(\omega)(\delta, \delta) + 2\nabla d(\omega)\delta\} + o(t^2), \tag{4.12}$$

which combined with (4.11) leads to

$$\lim_{t\downarrow 0}\frac{\nu(t)}{t^2/2}=\min_{\omega\in\Delta^*(y,d)}\min_{\delta\in N(\omega)}\{h(\omega)+\nabla^2 y(\omega)(\delta,\delta)+2\nabla d(\omega)\delta\}. \tag{4.13}$$

It follows that $O^2_K(y,d)=T^2_K(y,d)$ and $h\in T^2_K(y,d)$ if and only if for every $\omega\in\Delta^*(y,d)$,

$$h(\omega)+\min_{\delta\in N(\omega)}\{\nabla^2 y(\omega)(\delta,\delta)+2\nabla d(\omega)\delta\}\geq 0. \tag{4.14}$$

By calculating the minimum in (4.14) we obtain (4.8). We point out that, because of the second order growth condition (4.6), the minimum on $\delta\in N(\omega)$ is attained for $\|\delta\|\leq\|\nabla d\|_\infty/c$.

The proof of second order regularity involves similar arguments. Let $\Psi(\omega):=(\psi_1(\omega),...,\psi_m(\omega))^T$, and consider $t_k\downarrow 0$ and $y_k(\cdot):=y(\cdot)+t_k d(\cdot)+\frac{1}{2}t_k^2 h_k(\cdot)\in K$, where $h_k\in C(\Omega)$ are such that $h_k(\cdot)=x_k^T\Psi(\cdot)+a_k(\cdot)$ with $C(\Omega)\ni a_k\to a$ and $t_k x_k\to 0$. Consider also $\nu_k:=\min_{\omega\in\Omega}y_k(\omega)$. Similarly to (4.12) we have that, given a system of local coordinates, for every $\omega\in\Delta^*(y,d)$ and $\delta\in N(\omega)$,

$$\nu_k\leq\tfrac{1}{2}t_k^2\left[h_k(\omega+t_k\delta)+\nabla^2 y(\omega)(\delta,\delta)+2\nabla d(\omega)\delta\right]+o(t_k^2). \tag{4.15}$$

Moreover, since $t_k x_k\to 0$ and $\Psi(\cdot)$ is Lipschitz continuous on Ω we have that

$$x_k^T\Psi(\omega+t_k\delta)=x_k^T\Psi(\omega)+o(1).$$

Also $a_k(\omega+t_k\delta)=a_k(\omega)+o(1)$ and hence

$$t_k^2 h_k(\omega+t_k\delta)=t_k^2 h_k(\omega)+o(t_k^2). \tag{4.16}$$

Since $y_k\in K$ and hence $\nu_k\geq 0$, we then obtain from (4.15) and (4.16) that

$$h_k(\omega)+\nabla^2 y(\omega)(\delta,\delta)+2\nabla d(\omega)\delta+o(1)\geq 0, \tag{4.17}$$

where (due to compactness of Ω) the term $o(1)$ can be taken uniformly in $\omega\in\Delta^*(y,d)$ and $\|\delta\|\leq\|\nabla d\|_\infty/c$. By using formula (4.14), we obtain from (4.17) that $h_k+o(1)\in O^2_K(y,d)$, which completes the proof. □

It follows from the above theorem that for semi-infinite programs with constraints of the form $g(x,\omega)\geq 0$, $\omega\in\Omega$, there is no gap between the corresponding second order necessary and sufficient conditions under the following conditions:

(i) $g(\cdot,\omega)$ is twice differentiable with $\nabla^2_{xx}g(x,\omega)$ being continuous on $X\times\Omega$,

(ii) Robinson's constraint qualification holds,

(iii) $g(x_0,\cdot)$ satisfies the second order growth condition (4.6),

(iv) Ω is a smooth compact manifold and $\Delta(g(x_0,\cdot))$ is a smooth submanifold of Ω,

(v) $g(x_0,\cdot)$ is twice continuously differentiable and the functions $\psi_i(\cdot)=\frac{\partial g}{\partial x_i}(x_0,\cdot)$ are continuously differentiable.

Note that since Ω is compact, the last assumption (v) implies that the functions $\psi_i(\cdot)$ are Lipschitz continuous on Ω. Also in the case of semi-infinite programming, Robinson's constraint qualification (postulated in the above condition (ii)) is equivalent to the extended Mangasarian–Fromovitz condition, that is, there exists $h\in X$ such that $h^T\nabla_x g(x_0,\omega)>0$ for all $\omega\in\Delta_0:=\Delta(g(x_0,\cdot))$ (e.g., [39]). We also observe that when the function $g(\cdot,\omega)$ is concave for every fixed $\omega\in\Omega$, the feasible set

$$\Phi:=\{x\in X:g(x,\omega)\geq 0\text{ for all }\omega\in\Omega\}$$

is convex and Robinson's constraint qualification is equivalent to the Slater condition: there exists $\bar{x} \in X$ such that $g(\bar{x}, \omega) > 0$ for all $\omega \in \Omega$.

Combining Theorem 4.5 with Propositions 2.3 and 4.2 we deduce that, under assumptions (i)–(v) above, the set Φ is second order regular at x_0 and also second order directionally differentiable with

$$T^2_\Phi(x_0, h) = \{u \in X : \nabla_x g(x_0, \omega)u + \gamma(h, \omega) \geq 0 \text{ for all } \omega \in \Delta_1(h)\}\,, \tag{4.18}$$

where

$$\gamma(h, \omega) := \min_{\delta \in N(\omega)} \nabla^2 g(x_0, \omega)((h, \delta), (h, \delta)),$$

$\Delta_0 := \Delta(g(x_0, \cdot))$, and $\Delta_1(h) := \{\omega \in \Delta_0 : \nabla_x g(x_0, \omega)h = 0\}$. This formula may also be derived from Proposition 2.1 by using the characterization of second order directional derivatives of the min-function $\varphi(x) := \min_{\omega \in \Omega} g(x, \omega)$ given in [38, Theorem 4.1].

As an application, consider the example of semidefinite programming where

$$K = \mathcal{S}^n_+ := \{Z \in \mathcal{S}^n : g(Z, \omega) \geq 0 \text{ for all } \omega \in \Omega\}$$

with $g(Z, \omega) := \omega^T Z \omega$ and $\Omega := \{\omega \in \mathbb{R}^n : \|\omega\| = 1\}$. In this example the set Ω is a sphere, hence a smooth compact manifold. For a positive semidefinite matrix Z the corresponding contact set $\Delta(Z) := \{\omega \in \Omega : \omega^T Z \omega = 0\}$ is given by $\{\omega \in \Omega : Z\omega = 0\}$, which is a smooth submanifold of Ω. It is also not difficult to show that the corresponding second order growth condition holds (cf. [38]) and that the Lipschitz condition on functions ψ_i is automatically satisfied. Combining Theorem 4.5 and Proposition 2.3, we obtain the following result.

COROLLARY 4.6. *For any $n = 1, 2, ...,$ the cone $\mathcal{S}^n_+$ of symmetric positive semidefinite $n \times n$ matrices is second order directionally differentiable and second order regular at every point $Z \in \mathcal{S}^n_+$.*

An expression for the second order tangent sets and the corresponding second order optimality conditions for semidefinite optimization problems are given explicitly in [41].

5. Composite optimization.

As we mentioned in the introduction, an alternative approach to derivation of second order optimality conditions is to consider composite functions as in the problem (1.2) and that such problems can be investigated in the form (1.3). In this transformation the corresponding convex function is replaced by its epigraph. In this section we translate results obtained in the previous sections into the framework of composite optimization and compare them with results discussed in some recent publications. We assume throughout this section that the function $g(\cdot)$ in (1.2) is convex, proper, and lower semicontinuous and the mapping $F : X \to Y$ is continuously differentiable.

Let $K := \text{epi}(g)$ and $G(x, c) := (F(x), c)$. Consider a point $(x_0, c_0) \in X \times \mathbb{R}$ such that $F(x_0) \in \text{dom}(g)$ and $c_0 = g(F(x_0))$, where $\text{dom}(g) := \{y \in Y : g(y) < +\infty\}$ is the domain of g. Note that $DG(x_0, c_0)(h, c) = (DF(x_0)h, c)$. Therefore Robinson's constraint qualification (2.11) at (x_0, c_0) becomes

$$0 \in \text{int}\{F(x_0) + DF(x_0)X - \text{dom}(g)\}. \tag{5.1}$$

Note that if $g(\cdot)$ is continuous at the point $y_0 := F(x_0)$, then $\text{dom}(g)$ contains a neighborhood of y_0, and hence in that case constraint qualification (5.1) holds. Robinson's constraint qualification (5.1) can be also written in the following equivalent form [45]:

$$Y = DF(x_0)X - \mathcal{R}_{\mathrm{dom}(g)}(F(x_0)), \tag{5.2}$$

where $\mathcal{R}_A(y) := \cup\{t(A - y) : t \geq 0\}$ denotes the radial cone to the convex set A at $y \in A$. By taking the polar cone of both sides of (5.2), and since the polar of $\mathcal{R}_A(y)$ is $N_A(y)$, we obtain that (5.2) implies the following condition:

$$\{0\} = [DF(x_0)X]^{\perp} \cap N_{\mathrm{dom}(g)}(F(x_0)). \tag{5.3}$$

If the space Y is finite dimensional, then (5.2) and (5.3) are equivalent. Constraint qualification (5.3) was used in [33] (in the finite dimensional case) and in [11], while (5.2) has been used, for instance, in [13, 27].

The Lagrangian of (1.3) is

$$L(x, c, \lambda, \gamma) := c + \langle \lambda, F(x) \rangle + \gamma c. \tag{5.4}$$

The tangent cone to epi(g) at the point $(F(x_0), c_0)$ is given by

$$T_{\mathrm{epi}(g)}(F(x_0), c_0) = \{(d, c) : g^{\downarrow}(F(x_0), d) \leq c\}. \tag{5.5}$$

Consequently the first order necessary condition (3.2) can be written in the form

$$[DF(x_0)]^* \lambda = 0,\ \gamma = -1,\ \langle \lambda, d \rangle \leq g^{\downarrow}(F(x_0), d) \ \text{ for all } d \in Y.$$

Since the epigraph of $g^{\downarrow}(F(x_0), \cdot)$ coincides with the topological closure of the epigraph of $g'(F(x_0), \cdot)$, we have that the condition $\langle \lambda, d \rangle \leq g^{\downarrow}(F(x_0), d)$ for all $d \in Y$ is equivalent to $\lambda \in \partial g(F(x_0))$, where $\partial g(F(x_0))$ is the subdifferential of $g(\cdot)$ at $F(x_0)$. Therefore the above first order necessary condition can be written in the following form: there exists $\lambda \in Y^*$ such that

$$[DF(x_0)]^* \lambda = 0, \ \ \lambda \in \partial g(F(x_0)). \tag{5.6}$$

We obtain that if x_0 is a locally optimal solution of (1.2), then under constraint qualification (5.1) the set $\Lambda(x_0)$ of Lagrange multipliers satisfying (5.6) is nonempty and bounded. In the above form (5.6), first order necessary conditions in composite optimization were used in a number of publications [11, 13, 21, 26, 33].

Definition 4. *Let $g(y)$ be a proper lower semicontinuous convex function with a finite value at a point $y_0 \in Y$. We say that $g(\cdot)$ is (outer) second order regular at y_0 if the set $K :=$ epi(g) is (outer) second order regular at the point $(y_0, g(y_0))$.*

The set epi(g) is defined by the constraint $h(y, c) \leq 0$, where $h(y, c) := g(y) - c$. Since g is proper, and hence its domain dom(g) is nonempty, we can find $\bar{y}$ and $\bar{c}$ such that $h(\bar{y}, \bar{c}) < 0$, i.e., the Slater condition always holds in the present situation. Now Proposition 4.4 implies the following result.

Proposition 5.1. *Let $g(y)$ be a proper lower semicontinuous convex function. If g is finite and continuous at a point $y_0 \in Y$, then g is outer second order regular at y_0 if and only if, for every $d \in Y$ and every path $r : \mathbb{R}_+ \to Y$ satisfying* $\mathrm{t}r(t) \to 0$ *as $t \downarrow 0$, the inequality*

$$g\left(y_0 + td + \tfrac{1}{2}t^2 r(t)\right) \geq g(y_0) + tg'(y_0, d) + \tfrac{1}{2}t^2 g''_{-}(y_0; d, r(t)) + o(t^2) \tag{5.7}$$

holds.

Let $y \in \mathrm{dom}(g)$ be such that $g^{\downarrow}(y,d)$ is finite. Then it follows from Proposition 2.1 that

$$O^2_{\mathrm{epi}(g)}((y,g(y)),(d,g^{\downarrow}(y,d))) = \left\{(w,c) : g^{\downarrow\downarrow}_{-}(y;d,w) \le c\right\}. \tag{5.8}$$

Denote $\mathcal{T} := O^2_{\mathrm{epi}(g)}((y,g(y)),(d,g^{\downarrow}(y,d)))$, and $\psi(\cdot) := g^{\downarrow\downarrow}_{-}(y;d,\cdot)$. Then for $\lambda \in \Lambda(x_0)$ the corresponding sigma term becomes

$$\begin{array}{rcl} \sigma((\lambda,-1),\mathcal{T}) & = & \sup_{c,w}\{\langle\lambda,w\rangle - c : \psi(w) \le c\} \\ & = & \sup_w\{\langle\lambda,w\rangle - \psi(w)\} = \psi^*(\lambda), \end{array} \tag{5.9}$$

where ψ^* denotes the conjugate function of ψ.

Let us also note that the critical cone here can be written in the form

$$C(x_0,c_0) = \{(h,c) : g^{\downarrow}(F(x_0),DF(x_0)h) \le c,\ c = 0\},$$

provided constraint qualification (5.1) holds. Moreover, by the first order necessary conditions, $g^{\downarrow}(F(x_0),DF(x_0)h) \ge 0$ for any $h \in X$. Therefore this motivates us to consider the cone

$$\mathcal{C}(x_0) := \{h : g^{\downarrow}(F(x_0),DF(x_0)h) = 0\}. \tag{5.10}$$

Since $g^{\downarrow}(F(x_0),\cdot)$ is lower semicontinuous, this cone is closed. Combining Theorems 3.1 and 3.2 we get the following result.

THEOREM 5.2. *Suppose that $g(y)$ is a proper lower semicontinuous convex function, that $F : X \to Y$ is a twice continuously differentiable mapping, that $F(x_0) \in$* dom(g), *and that constraint qualification* (5.1) *holds. Then,*

(i) *(second order necessary condition) let x_0 be a locally optimal solution of* (1.2), *then for any $h \in \mathcal{C}(x_0)$ and any convex function $\phi(\cdot) \ge g^{\downarrow\downarrow}_{-}(F(x_0);DF(x_0)h,\cdot)$ the following inequality holds:*

$$\sup_{\lambda\in\Lambda(x_0)} \{\langle\lambda, D^2_{xx}F(x_0)(h,h)\rangle - \phi^*(\lambda)\} \ge 0; \tag{5.11}$$

(ii) *(second order sufficient condition) let x_0 be a stationary point of* (1.2), *i.e., it satisfies the first order necessary condition* (5.6), *and suppose that g is outer second order regular at $y_0 := F(x_0)$ and that*

$$\sup_{\lambda\in\Lambda(x_0)} \{\langle\lambda, D^2_{xx}F(x_0)(h,h)\rangle - \psi^*(\lambda)\} > 0 \text{ for all } h \in \mathcal{C}(x_0)\setminus\{0\}, \tag{5.12}$$

where $\psi(\cdot) := g^{\downarrow\downarrow}_{-}(F(x_0);DF(x_0)h,\cdot)$. Then for some $\alpha > 0$ and all x in a neighborhood of x_0,

$$g(F(x)) \ge g(F(x_0)) + \alpha\|x - x_0\|^2, \tag{5.13}$$

and hence x_0 is a locally optimal solution of (1.2).

It follows that if $g^{\downarrow\downarrow}_{-}(F(x_0);DF(x_0)h,\cdot)$ is convex and g is outer second order regular at $F(x_0)$, then there is no gap between second order necessary and sufficient conditions in the above theorem.

The second order optimality conditions of Theorem 5.2 are essentially equivalent to those obtained via second order (epi)subderivatives in [34, 13, 36], but they apply under different conditions.

For instance, in [34] (and subsequent work by the author) the function g is assumed to be piecewise linear-quadratic convex, a situation covered by Theorem 5.2 since such functions are second order regular. In order to check this we observe that any twice continuously differentiable convex function, in particular a quadratic convex function $g(y)$, is second order regular (see, e.g., Proposition 4.2 or 5.1). Now, if K is a polyhedral convex subset of Y and since the epigraph of the function $g(y)+I_K(y)$ is given by the intersection of the epigraph of g and $K \times \mathbb{R}$, it follows from Corollary 4.3 that $g(y)+I_K(y)$ is also second order regular. Finally, the epigraph of a piecewise linear-quadratic convex function is given by the union of a finite number of epigraphs of functions of the form $g(y)+I_K(y)$, with g being quadratic and K being polyhedral. It can be easily verified that union of a finite number of second order regular sets is also second order regular, from which the conclusion follows.

The results in [34] were extended in [13] beyond the class of piecewise linear-quadratic functions. The regularity condition used in that extension does not allow us to cover the second order regular case as in Theorem 5.2. (Among other things it requires some kind of local radiality of the domain of g, which is certainly not needed in our analysis.) However, it is also not clear whether second order regularity is weak enough to recover the results in [13].

The comparison with the results in [21] is more involved, since the optimality conditions are expressed in terms of lower second order epiderivatives instead of the parabolic ones as we express them. Of course, under suitable regularity assumptions both types of conditions can be shown to be equivalent thanks to the duality relation existing between both types of derivatives. However, in the general settings of [21] such duality relation cannot be ensured and the results are not comparable. The only exception concerns [21, Corollary 4], which is in fact a slight extension of results in [13], but, as in that paper, the regularity condition on which it is based is not comparable with second order regularity.

It is also possible to show that the second order regularity of g is a sufficient (but not necessary) condition for g to be parabolically regular. (See [36] for a discussion of the concept of parabolic regularity.) A detailed study of the relation between the concepts of second order regularity and parabolic regularity is given in the forthcoming book [10].

6. Extensions to nonisolated minima. Little is known about second order optimality conditions for nonisolated minima. A characterization of the second order growth condition is given in [8], under a constraint qualification, for smooth convex optimization problems with finitely many constraints. In [7] some sufficient conditions are stated for nonlinear programming problems. It is relatively easy to formulate a second order necessary condition that generalizes a result in [7].

Let $S \subset G^{-1}(K)$ be a set of optimal solutions (minimizers) of the problem (P), and let $\mathcal{T}_S(x) := \limsup_{t\downarrow 0} t^{-1}(S-x)$ be the contingent cone to S at x. It is easily checked that if $x \in S$ and $h \in X$, then $\operatorname{dist}(x+th, S) \geq t \operatorname{dist}(h, \mathcal{T}_S(x)) + o(t)$ for $t > 0$. Suppose that Robinson's constraint qualification holds at every point $x \in S$ and that S is compact. We have that if the second order growth condition holds at S, then for any feasible path $x(t)$ of the form (3.4) with $x_0 \in S$, $f(x(t)) \geq f(x_0) + ct^2 \operatorname{dist}(h, \mathcal{T}_S(x)) + o(t^2)$ for some $c > 0$ and $t > 0$ small enough. It then follows by the arguments used in the proof of Theorem 3.1 that a necessary condition for the second order growth (at S) is that there exists $c > 0$ such that for all $x \in S$ and $h \in C(x)$,

$$\sup_{\lambda\in\Lambda(x)} \{D^2_{xx}L(x,\lambda)(h,h) - \sigma(\lambda, \mathcal{T}(x,h))\} \geq 2ct^2 \operatorname{dist}(h, \mathcal{T}_S(x)), \tag{6.1}$$

where $\mathcal{T}(x,h)$ is a convex subset of $O^2_K(G(x), DG(x)h)$. Recall that the set of proximal normals to S at $x \in S$ is defined as

$$\mathcal{N}_S(x) := \{h \in X;\ \operatorname{dist}(x + th, S) = t\|h\| \ \text{ for some } \ t > 0\},$$

and set $\mathcal{N}^\varepsilon_S(x) := \{h \in X;\ \operatorname{dist}(h, \mathcal{N}_S(x)) \leq \varepsilon\}$. As $\operatorname{dist}(h, \mathcal{T}_S(x)) = \|h\|$ whenever $h \in \mathcal{N}_S(x)$, a consequence of (6.1), and therefore a necessary condition for quadratic growth (see [7]), is that for $\varepsilon > 0$ small enough

$$\sup_{\lambda\in\Lambda(x)} \{D^2_{xx}L(x,\lambda)(h,h) - \sigma(\lambda, \mathcal{T}(x,h))\} \geq c\|h\|^2 \text{ for all } h \in C(x) \cap \mathcal{N}^\varepsilon_S(x). \tag{6.2}$$

Definition 5. *We say that the set S satisfies a property of uniform approximation of critical cones if for every $\varepsilon > 0$ there exists $\alpha > 0$ such that for all $x \in S$ and $h \in X$ satisfying $Df(x)h \leq \alpha\|h\|$ and $DG(x)h \in T_K(G(x)) + \alpha\|h\|B_Y$, we have $\operatorname{dist}(h, C(x)) \leq \varepsilon\|h\|$.*

Definition 6. *We say that K is uniformly regular with respect to the set S and the mapping $G(x)$ if for $x \in S$ and $h \in C(x)$, $O^2_K(G(x), DG(x)h)$ is an upper second order approximation set for K at the point $G(x)$ in the direction $DG(x)h$ with respect to $DG(x)$ uniformly over S. That is, if $x_k \in S$, $h_k \in C(x_k)$, $t_k \downarrow 0$, and $r_k = DG(x_k)z_k + a_k$ are sequences such that $\{a_k\}$ is convergent, $t_kz_k \to 0$ and $G(x_k) + t_kDG(x_k)h_k + \frac{1}{2}t_k^2 r_k \in K$, then*

$$\lim_{k\to\infty} \operatorname{dist}(r_k, O^2_K(G(x_k), DG(x_k)h_k)) = 0. \tag{6.3}$$

Theorem 6.1. *Let $S \subset G^{-1}(K)$ satisfy the property of uniform approximation of critical cones, and suppose that Robinson's constraint qualification holds at every point $x \in S$, that S is compact, that K is uniformly regular with respect to the set S and the mapping $G(x)$, and that $O^2_K(G(x), DG(x)h) = T^2_K(G(x), DG(x)h)$ for all $x \in S$ and $h \in C(x) \setminus \mathcal{T}_S(x)$. Then condition* (6.2) *is necessary and sufficient for the second order growth at S.*

Proof. We already observed that the condition is necessary. It suffices therefore to prove that it is sufficient. Let x_k be a sequence of feasible points $x_k \in G^{-1}(K)$ converging to a point $x_0 \in S$ and such that (3.16) holds. Let $\hat{x}_k$ be a projection of x_k onto S, i.e., $\hat{x}_k \in S$ and $\|x_k - \hat{x}_k\| = \operatorname{dist}(x_k, S)$. Set $t_k := \|x_k - \hat{x}_k\|$ and $\hat{h}_k := (x_k - \hat{x}_k)/t_k$. Then $\hat{h}_k \in \mathcal{N}_S(\hat{x}_k)$. From the property of uniform approximation of critical cones, there exists h_k such that $h_k \in C(\hat{x}_k)$ and $\|h_k - \hat{h}_k\| \to 0$, hence for all $\varepsilon > 0$, $h_k \in \mathcal{N}^\varepsilon_S(\hat{x}_k)$ for large enough k. Then $x_k = \hat{x}_k + t_kh_k + o(t_k)$. The remainder of the proof is similar to the one of Theorem 3.2. □

It was proved in [8] that the property of uniform approximation of critical cones is satisfied for finitely constrained convex optimization problems. Whether this property holds in more general settings is an open problem.

Acknowledgment. The authors thank the two referees for their useful remarks.

REFERENCES

[1] A. Ben-Tal, M. Teboulle, and J. Zowe, *Second order necessary optimality conditions for semi-infinite programming problems*, in Semi-infinite Programming, Lecture Notes in Control and Inform. Sci. 15, R. Hettich, ed., Springer-Verlag, Berlin, 1979, pp. 17–30.

[2] A. Ben-Tal, *Second order and related extremality conditions in nonlinear programming*, J. Optim. Theory Appl., 31 (1980), pp. 143–165.

[3] A. Ben-Tal and J. Zowe, *A unified theory of first and second order conditions for extremum problems in topological vector spaces*, Math. Programming Study, 19 (1982), pp. 39–76.

[4] A. Ben-Tal and A. Nemirovski, *Robust solutions of uncertain linear programs*, Oper. Res. Lett., to appear.

[5] J. F. Bonnans and R. Cominetti, *Perturbed optimization in Banach spaces* I: *A general theory based on a weak directional constraint qualification,* II: *A theory based on a strong directional qualification,* III: *Semi-infinite optimization*, SIAM J. Control Optim., 34 (1996), pp. 1151–1171, 1172–1189, and 1555–1567.

[6] J. F. Bonnans, R. Cominetti, and A. Shapiro, *Sensitivity analysis of optimization problems under second order regular constraints*, Math. Oper. Res., to appear.

[7] J. F. Bonnans and A. D. Ioffe, *Second-order sufficiency and quadratic growth for non isolated minima*, Math. Oper. Res., 20 (1995), pp. 801–817.

[8] J. F. Bonnans and A. D. Ioffe, *Quadratic growth and stability in convex programming problems with multiple solutions*, J. Convex Anal., 2 (1995), pp. 41–57.

[9] J. F. Bonnans and A. Shapiro, *Optimization problems with perturbations: A guided tour*, SIAM Rev., 40 (1998), pp. 228–264.

[10] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer-Verlag, New York, to appear.

[11] J. V. Burke and R. A. Poliquin, *Optimality conditions for non-finite valued convex composite functions*, Math. Programming, 57 (1992), pp. 103–120.

[12] R. Cominetti, *Metric regularity, tangent sets and second order optimality conditions*, Appl. Math. Optim., 21 (1990), pp. 265–287.

[13] R. Cominetti, *On pseudo-differentiability*, Trans. Amer. Math. Soc., 324 (1991), pp. 843–865.

[14] R. Cominetti and J-P. Penot, *Tangent sets to unilateral convex sets*, C. R. Acad. Sci. Sér. I Math., 321 (1995), pp. 1631–1636.

[15] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York, 1968.

[16] R. P. Hettich and H. Th. Jongen, *Semi-infinite programming: Conditions of optimality and applications*, in Optimization Techniques, Proc. 8th IFIP Conf. on Optimization Techniques, Würzburg, Part 2, J. Stoer, ed., Springer-Verlag, New York, 1977.

[17] R. Hettich and P. Zencke, *Numerische Methoden der Approximation und Semi-infiniten Optimierung*, Teubner, Stuttgart, 1982.

[18] R. Hettich and K. O. Kortanek, *Semi-infinite programming: Theory, methods and applications*, SIAM Rev., 35 (1993), pp. 380–429.

[19] A. D. Ioffe, *Necessary and sufficient conditions for a local minimum.* 3: *Second order conditions and augmented duality*, SIAM J. Control Optim., 17 (1979), pp. 266–288.

[20] A. D. Ioffe, *On some recent developments in the theory of second order optimality conditions*, in Optimization, Lecture Notes in Math. 1405, S. Dolecki, ed., Springer-Verlag, Berlin, 1989, pp. 55–68.

[21] A. D. Ioffe, *Variational analysis of a composite function: A formula for the lower second order epi-derivative*, J. Math. Anal. Appl., 160 (1991), pp. 379–405.

[22] A. D. Ioffe, *On sensitivity analysis of nonlinear programs in Banach spaces: The approach via composite unconstrained optimization*, SIAM J. Optim., 4 (1994), pp. 1–43.

[23] H. Kawasaki, *An envelope-like effect of infinitely many inequality constraints on second order necessary conditions for minimization problems*, Math. Programming, 41 (1988), pp. 73–96.

[24] S. Kurcyusz, *On the existence and nonexistence of Lagrange multipliers in Banach spaces*, J. Optim. Theory Appl., 20 (1976), pp. 81–110.

[25] Zs. Pales and V. M. Zeidan, *Nonsmooth optimum problems with constraints*, SIAM J. Control Optim., 32 (1994), pp. 1476–1502.

[26] J. P. Penot, *Optimality Conditions for Minimax Problems, Semi-infinite Programming Problems and Their Relatives*, Report 92/16, UPRA, Laboratoire de Math. Appl., Av. de l'Université, 64000 Pau, France, 1992.

[27] J. P. Penot, *Optimality conditions in mathematical programming and composite optimization*, Math. Programming, 67 (1994), pp. 225–245.

[28] S. M. Robinson, *First order conditions for general nonlinear optimization*, SIAM J. Appl. Math., 30 (1976), pp. 597–607.

[29] S. M. Robinson, *Stability theory for systems of inequalities, Part* II: *Differentiable nonlinear systems*, SIAM J. Numer. Anal., 13 (1976), pp. 497–513.

[30] S. M. Robinson, *Regularity and stability for convex multivalued functions*, Math. Oper. Res.,

1 (1976), pp. 130–143.

[31] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[32] R. T. Rockafellar, *Conjugate Duality and Optimization*, CBMS-NSF Regional Conference Series in Applied Mathematics 16, SIAM, Philadelphia, PA, 1974.

[33] R. T. Rockafellar, *First and second-order epi-differentiability in nonlinear programming*, Trans. Amer. Math. Soc., 307 (1988), pp. 75–108.

[34] R. T. Rockafellar, *Second-order optimality conditions in nonlinear programming obtained by way of epi-derivatives*, Math. Oper. Res., 14 (1989), pp. 462–484.

[35] R. T. Rockafellar, *Nonsmooth analysis and parametric optimization*, in Methods of Nonconvex Analysis, Lecture Notes in Math. 1446, A. Cellina, ed., Springer-Verlag, Berlin, 1990, pp. 137–151.

[36] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer-Verlag, Berlin, 1997.

[37] A. Shapiro, *Second-order derivatives of extremal-value functions and optimality conditions for semi-infinite programs*, Math. Oper. Res., 10 (1985), pp. 207–219.

[38] A. Shapiro, *Perturbation theory of nonlinear programs when the set of optimal solutions is not a singleton*, Appl. Math. Optim., 18 (1988), pp. 215–229.

[39] A. Shapiro, *On Lipschitzian stability of optimal solutions of parametrized semi-infinite programs*, Math. Oper. Res., 19 (1994), pp. 743–752.

[40] A. Shapiro and M. K. H. Fan, *On eigenvalue optimization*, SIAM J. Optim., 5 (1995), pp. 552–569.

[41] A. Shapiro, *First and second order analysis of nonlinear semidefinite programs*, Math. Programming, Series B, 77 (1997), pp. 301–320.

[42] C. Ursescu, *Multifunctions with convex closed graph*, Czechoslovak Math. J., 25 (1975), pp. 438–441.

[43] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

[44] W. Wetterling, *Definitheitsbedingungen fürrelative Extrema bei Optimieungs- und Approximation-saufgaben*, Numer. Math., 15 (1879), pp. 122–136.

[45] J. Zowe and S. Kurcyusz, *Regularity and stability for the mathematical programming problem in Banach spaces*, Appl. Math. Optim., 5 (1979), pp. 49–62.

SIAM J. Optim.
Vol. 9, No. 2, pp. 493–503

STABLE SET POLYTOPES FOR A CLASS OF CIRCULANT GRAPHS*

GEIR DAHL†

Abstract. We study the stable set polytope $P(G_n)$ for the graph G_n with n nodes and edges $[i,j]$ with $j \in \{i+1, i+2\}$, $i = 1,\dots,n$ and where nodes $n+1$ and 1 (resp., $n+2$ and 2) are identified. This graph coincides with the antiweb $\bar{W}(n,3)$. A minimal linear system defining $P(G_n)$ is determined. The system consists of certain rank inequalities with some number theoretic flavor. A characterization of the vertices of a natural relaxation of $P(G_n)$ is also given.

Key words. polyhedral combinatorics, stable sets, circulant graphs

AMS subject classifications. 90C27, 05C85, 52B11

PII. S1052623497321973

1. Introduction.

Let $n \geq 3$ be a positive integer, and let $\mathbf{C}_n = (c_{i,j}) \in \mathbb{R}^{n,n}$ be the $(3,n)$-circulant matrix; i.e., for $i = 1,\dots,n$ we have $c_{i,j} = 1$ if $i \leq j \leq i+2$, where $n+1$ and 1 (resp., $n+2$ and 2) are identified (modulo n calculation of indices). We let $\mathbf{0}$, $\mathbf{1}$, and $\mathbf{2}$ denote a vector of suitable dimension with all components being 0, 1, and 2, respectively. In this paper we are concerned with the polytope

$$P_n = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{C}_n\mathbf{x} \leq \mathbf{1}, \ \ \mathbf{x} \geq \mathbf{0}\} \tag{1.1}$$

and its integer hull, i.e., the convex hull of the integral points in P_n. These objects relate to stable sets. Let $V = \{1,\dots,n\}$ and consider the *circulant graph* $G_n = (V,E)$ with node set V and edge set E consisting of the edges $[i,i+1]$ and $[i,i+2]$ for $i \in V$. It is useful to imagine the nodes of V placed consecutively along a circle so that node 1 and n are adjacent; see Fig. 1.1. The graph G_n coincides with the antiweb $\bar{W}(n,3)$. We recall from [10] that the *web* $W(n,k)$ is defined as the graph on n nodes with edges $[i,j]$ for $j = i+k,\dots,i+n-k$. The *antiweb* $\bar{W}(n,k)$ is the complement of $W(n,k)$, i.e., the graph on n nodes containing precisely those edges that are not in $W(n,k)$. A stable set in a graph is a subset S of nodes such that no pair of nodes in S are adjacent. A stable set S in G_n is a set of nodes such that the distance between consecutive nodes is at least 3. Let $\mathcal{S}_n$ denote the set of all stable sets in G_n. Then the integral points in P_n coincide with the incidence vectors of sets in $\mathcal{S}_n$, so the integer hull of P_n equals the *stable set polytope* $P(G_n)$ associated with the graph G_n. P_n may be viewed as the relaxation of $P(G_n)$ consisting of nonnegativity constraints and clique constraints. Note that $P(G_n)$ is full dimensional because it contains the origin and the coordinate vectors. Moreover, each nonnegativity constraint $x_j \geq 0$ defines a facet of $P(G_n)$ which is called a *trivial facet.* Each clique inequality $x_i + x_{i+1} + x_{i+2} \leq 1$ also defines a facet of $P(G_n)$, and we call it a *clique facet.*

The purpose of this paper is to study the polytopes P_n and $P(G_n)$. We determine all the vertices of P_n and a minimal linear system of inequalities defining $P(G_n)$. This system contains, apart from the inequalities defining P_n, certain inequalities with $(0,1)$-coefficients called the 1-*interval inequalities.* These inequalities are of interest

*Received by the editors May 28, 1997; accepted for publication (in revised form) March 12, 1998; published electronically March 17, 1999.

http://www.siam.org/journals/siopt/9-2/32197.html

†University of Oslo, Department of Informatics, P.O. Box 1080, Blindern, 0316 Oslo, Norway (geird@ifi.uio.no).

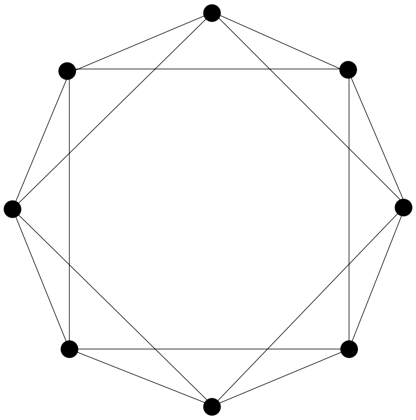


FIG. 1.1. *The graph G_8.*

for stable sets in general graphs; they also produce facets via the procedure of lifting. The work was motivated by a study of spanning trees (in a given graph H) satisfying a "2-hop constraint"; see [2]. This constraint says that each node or one of its neighbors is adjacent to a given root node. An interesting special case is when H is a wheel, i.e., a cycle with an additional root node joined to all the cycle nodes. Then a certain integer linear programming model for finding a minimum-cost 2-hop spanning tree has as its linear relaxation the polytope P_n, and the vertices of $P(G_n)$ correspond to incidence vectors of 2-hop spanning trees. In particular, the results of the present paper lead to completeness results for polytopes associated with 2-hop spanning trees (see [2]).

Consider a weighted stable set problem in G_n: for given numbers $w_j, j \in V$ find a stable set S in G_n with $\sum_{j \in S} w_j$ largest possible. This problem may be solved in polynomial time as follows. Choose $j \in V$. Any stable set S satisfies (i) $j \in S$, (ii) $j+1 \in S$, or (iii) $j, j+1 \notin S$. Thus the weighted stable set problem may be solved by finding an optimal stable set for each of these three cases and comparing the solutions. Each of the three subproblems may be solved by linear programming since deleting proper columns results in a totally unimodular coefficient matrix. The existence of an efficient algorithm for solving the weighted stable set problem in G_n is a motivation for seeking a complete linear description of $P(G_n)$.

A survey of the stable set problem and stable set polytopes is given in [4, Chapter 9]. Complete linear descriptions of stable set polytopes are known for certain classes of graphs, such as bipartite graphs, interval graphs, and chordal graphs; all these classes are perfect graphs so nonnegativity constraints and clique constraints suffice to describe the corresponding stable set polytopes. Furthermore, for series-parallel graphs the stable set polytopes are described by nonnegativity constraints, edge constraints, and odd circuit constraints; for a proof see [7]. For graph theory and polyhedral theory used in this paper, see [8] and [9]. A $(0,1)$-matrix is called an *interval matrix* provided that in each row the 1's occur consecutively. A well-known fact is that every interval matrix is totally unimodular (see [8]). If $\mathbf{a}^T\mathbf{x} \leq \alpha$ is a valid inequality for a polytope P, we say that each point in $P \cap \{\mathbf{x} : \mathbf{a}^T\mathbf{x} = \alpha\}$ is a *root* of the inequality $\mathbf{a}^T\mathbf{x} \leq \alpha$ (or the corresponding face of P). The incidence vector $\chi^T \in \mathbb{R}^n$ of a subset T of $\{1, \dots, n\}$ is the vector where χ^T_j equals 1 if $j \in T$ and 0 otherwise.

2. The polytope P_n. It is clear that the incidence vector of each stable set in G_n is a vertex of P_n. In this section we determine the remaining vertices.

Certain subsets of the node set V are of interest in what follows. We shall call a subset of consecutive nodes in V an *interval* and note that, e.g., $\{n-1, n, 1\}$ is an interval (the modulo n calculation). Any *strict* subset T of V corresponds to a partition of V into nonempty, consecutive, disjoint intervals $I_1, J_1, I_2, J_2, \dots, I_t, J_t$, where $T = \cup_{s=1}^{t} I_s$. Note that the intervals J_s are determined by the intervals I_s. We then write $T = I_1 + \cdots + I_t$. A 1*-interval set* is a subset T being the union of intervals $I_1, \dots, I_t$ separated by just one node, i.e., $|J_s| = 1$ for all $s \le t$.

Consider a 1-interval set $T = I_1 + \cdots + I_t$ satisfying $|I_s| \in \{1,2\}$ for $s \le t$. Associated with T is the point $\mathbf{x}^T \in \mathbb{R}^n$ given by $x_j^T = 1/2$ for $j \in T$ and $x_j^T = 0$ otherwise, i.e., $\mathbf{x}^T = (1/2)\chi^T$. A point $\mathbf{x}^T$ for which t (also equal to the number of zeros in $\mathbf{x}^T$) is odd will be called an *odd* 1/2*-string.*

Proposition 2.1. *The vertices of P_n are the incidence vectors of stable sets in G_n, all odd* 1/2*-strings, and, provided that n is not a multiple of* 3*, the vector with all components equal to* 1/3*.*

Proof. Let $\mathbf{x}$ be a *nonintegral* vertex of $P(G_n)$. We establish several properties of $\mathbf{x}$, eventually showing that $\mathbf{x}$ must be an odd 1/2-string or have all components equal to 1/3.

Property 1*: For each $i \le n$ we have that $x_i < 1$ and that either x_i or x_{i+1} is positive.* If $x_i = x_{i+1} = 0$, the $(n-2)$-dimensional vector $\mathbf{x}'$ with the remaining components of $\mathbf{x}$ must be a vertex of the polytope defined by $\mathbf{C}'\mathbf{x}' \le \mathbf{1}$, $\mathbf{x}' \ge \mathbf{0}$, where $\mathbf{C}'$ is the matrix obtained from $\mathbf{C}$ by deleting columns i and $i+1$. But $\mathbf{C}'$ is totally unimodular because it is obtained from an interval matrix by column permutations. (The property of total unimodularity is preserved under permutations of columns or rows.) This implies that $\mathbf{x}'$ is integral (in fact $(0,1)$) and so is $\mathbf{x}$, a contradiction. Therefore, either x_i or x_{i+1} is positive. Similarly, if $x_i = 1$, then $x_{i-2} = x_{i-1} = x_{i+1} = x_{i+2} = 0$, the remaining components are determined from an interval matrix (deleting the columns $i-2, \dots, i+2$ and one row in $\mathbf{C}$), and we again arrive at the desired contradiction.

By Property 1 the support T of $\mathbf{x}$ (i.e., the indices of nonzero components) is either V or a 1-interval set $T = I_1 + \cdots + I_t$. Consider first the case when $T = V$. Then all variables in $\mathbf{x}$ are positive and therefore $\mathbf{C}_n\mathbf{x} = \mathbf{1}$ (as there must be n active inequalities). But $\mathbf{C}_n$ is nonsingular if and only if n is not a multiple of 3, and in that case we see that $\mathbf{x} = (1/3, \dots, 1/3)$. In the remaining part of the proof we may assume that $T \ne V$.

Property 2*: If $i \notin T$, then the equation $x_{i-1} + x_i + x_{i+1} = 1$ holds.* Otherwise we would again have that all components in $\mathbf{x}$ except the ith were determined by a totally unimodular matrix (deleting the assumed nonactive constraint). A contradiction arises.

Property 3*: $|I_s| \le 2$ for all $s \le t$.* To prove this we determine an upper bound on the number of nonredundant, active clique inequalities in $\mathbf{x}$. If $|I_s| = 1$, say $I_s = \{i\}$, then the clique inequality $x_{i-1} + x_i + x_{i+1} \le 1$ is not active because that would give $x_i = 1$ (as $x_{i-1} = x_{i+1} = 0$); this is a contradiction due to Property 1. If $|I_s| = 2$, say $I_s = \{i, i+1\}$, then $x_{i-1} + x_i + x_{i+1} \le 1$ and $x_i + x_{i+1} + x_{i+2} \le 1$ are equivalent, so one of them is redundant. Finally, if $|I_s| > 2$, say $I_s = \{i, i+1, \dots, j\}$, then neither inequality $x_{i-1} + x_i + x_{i+1} \le 1$ nor $x_{j-1} + x_j + x_{j+1} \le 1$ is active, for that would give that either x_{i+2} or x_{j-2} was 0. Note that all the mentioned inactive or redundant inequalities are distinct. Let m_1 and m_2 be the number of intervals I_s with $|I_s|$ equal to 1 and 2, respectively. The number m_3 of intervals I_s with $|I_s| \geq 3$ clearly satisfies $m_3 = t - m_1 - m_2$. Our discussion shows that an upper bound on the number of active, nonredundant clique inequalities in $\mathbf{x}$ is $n-(m_1+m_2+2m_3) = n-t-m_3$. But $\mathbf{x}$ has $n-t$ positive components to be determined by the active clique constraints, so $n-t-m_3 \geq n-t$, which implies that $m_3 = 0$ and Property 3 follows.

The counting argument just given also shows that, except for those special constraints mentioned in the paragraph above, *all other* clique constraints are active in $\mathbf{x}$. From this we deduce that the nonzero components of $\mathbf{x}$ must alternate between α and $1-\alpha$, where $0 < \alpha < 1$. The number of nonzeros $n-t$ must be odd, otherwise we could write $\mathbf{x}$ as the midpoint of two different solutions in P_n similar to $\mathbf{x}$ (with α replaced by $\alpha - \epsilon$ and $\alpha + \epsilon$, respectively, for suitably small ϵ). Finally, as $n-t$ is odd, one of the active clique inequalities gives that $\alpha = 1-\alpha$, i.e., $\alpha = 1/2$. This means that $\mathbf{x}$ is an odd 1/2-string and the proof is complete. □

3. Rank facets of $P(G_n)$. In this section we study the stable set polytope $P(G_n)$ and valid inequalities for $P(G_n)$ of the form $x(T) \leq \alpha$ for $T \subseteq V$; such inequalities are called *rank* (or *canonical*) *inequalities*. Clearly, we may restrict the attention to $\alpha = \alpha(T) := \max\{|S \cap T| : S$ is a stable set in $G_n\}$, which is the stability number in the subgraph $G_n[T]$ of G_n induced by T.

First we consider how to compute $\alpha(T)$ for a given subset T of V. In [6] a polynomial algorithm is given for computing the stability number of a *claw-free graph*, i.e., a graph with no induced subgraph isomorphic to the star $K_{1,3}$. The algorithm is based on a reduction to a matching problem. Since G_n is claw free, the subgraph $G_n[T]$ is also claw free and the algorithm of [6] could be used to determine $\alpha(T)$. However, the special structure of G_n makes it possible to determine $\alpha(T)$ by a simple greedy algorithm which is discussed in the following.

Let $\mathbf{A} \in \mathbb{R}^{m,n}$ be a $(0,1)$-matrix. Following [3] we say that $\mathbf{A}$ is *greedy* if the greedy algorithm correctly solves the linear program

$$\max\{\mathbf{c}^T\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b},\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}\} \tag{3.1}$$

for all $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{c}, \mathbf{u} \in \mathbb{R}^n$ with $c_1 \geq c_2 \geq \cdots \geq c_n$. The greedy algorithm for (3.1) determines a solution $\mathbf{x}'$ as follows: for $j = 1, \ldots, n$ let x'_j be the maximum real number r such that $(x'_1, \ldots, x'_{j-1}, r, 0, \ldots, 0)$ is feasible in (3.1). Note that $\mathbf{x}'$ is integral whenever both $\mathbf{u}$ and $\mathbf{b}$ are integral, so the greedy algorithm also solves the integer LP corresponding to (3.1). It was shown in [5] that $\mathbf{A}$ is greedy if and only if neither of the following two submatrices is a submatrix of $\mathbf{A}$:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

An immediate consequence of this result is that every interval matrix is greedy. Now, consider the circulant matrix $\mathbf{C}_n$ defined in the introduction. The matrix $\mathbf{C}'$ obtained from $\mathbf{C}_n$ by deleting columns j and $j+1$ and permuting the columns suitably is an interval matrix and therefore greedy. In particular, for each $\mathbf{u} \in \{0,1\}^{n-2}$ we can solve the integer program

$$\max\{\mathbf{1}^T\mathbf{x}' : \mathbf{C}'\mathbf{x}' \leq \mathbf{1},\ \mathbf{0} \leq \mathbf{x}' \leq \mathbf{u},\ \mathbf{x}' \text{ is integral}\} \tag{3.2}$$

by the greedy algorithm. We see that (3.2) is the stable set problem in the subgraph of G_n induced by the nodes $\{k : u_k = 1 \text{ and } k \neq j, j+1\}$.

For $T \subseteq V$ and $j \in T$ define

$$\alpha_j(T) = \max\{|S \cap T| : S \in \mathcal{S}_n,\ j \in S\}. \tag{3.3}$$

The following greedy algorithm determines $\alpha_j(T)$: initially let $S = \{j\}$, $s := j$, and choose $k \in \{s+3, \dots, j-3\}$ "smallest possible" with $k \in T$ and add k to S. Repeat this process for $s := k$ until no more k can be found. The correctness of this algorithm follows from the discussion above.

We can calculate $\alpha(T)$ as follows. If $T = V$, we obtain $\alpha(T) = \alpha(G_n) = \lfloor n/3 \rfloor$. Assume next that $T \neq V$, say $j \in T$ but $j+1 \notin T$. We determine the number $\alpha' := \max\{|S \cap T| : S \in \mathcal{S}_n,\ j \notin S\} = \max\{|S \cap T| : S \in \mathcal{S}_n,\ j, j+1 \notin S\}$ by removing nodes j and $j+1$ from the graph and using the greedy algorithm in the interval graph we then obtain starting in $j+2$. We then calculate $\alpha_j(T)$ using the greedy algorithm above and conclude that $\alpha(T) = \max\{\alpha', \alpha_j(T)\}$. We call this procedure for finding $\alpha(T)$ the α-GREEDY algorithm (with start in node j). It is used in some proofs later.

We now calculate $\alpha(T)$ for certain interesting 1-interval sets.

Lemma 3.1. *Let $T = I_1 + \cdots + I_t$ be a 1-interval set satisfying, for $s = 1, \dots, t$, $|I_s| \equiv 1 (\mathrm{mod}\ 3)$, say $|I_s| = 3k_s + 1$, where k_s is a nonnegative integer. Then $\alpha(T) = \sum_{s=1}^{t} k_s + \lfloor t/2 \rfloor$ or, equivalently, $\alpha(T) = n/3 - t/6$ when t is even and $\alpha(T) = n/3 - t/6 - 1/2$ when t is odd.*

Proof. The result may be found by the α-GREEDY algorithm, but we give an alternative proof here. Let $s \in \{1, \dots, t\}$. We note that $\alpha(I_s) = k_s + 1$ and that there is a unique maximum stable set S in I_s and, moreover, S contains both of the end points of the interval I_s. The next observation is that $\alpha(I_s \cup I_{s+1}) = k_s + k_{s+1} + 1$ and every maximum stable set in $I_s \cup I_{s+1}$ must contain both of the end points of (exactly) one of the two intervals. One such stable set, say S_s, contains $k_s + 1$ nodes in I_s (and therefore the two end points) but it does not contain the "right-hand end node" of I_{s+1}. Assume now that t is even, and let S be the union of such sets $S_1, S_3, \dots, S_{t-1}$. From the construction we see that S is a stable set in G_n. Furthermore, $|S| = \sum_{s=1}^{t} k_s + t/2$ so we conclude that $\alpha(T) \geq \sum_{s=1}^{t} k_s + t/2$. Moreover, this is an equality; otherwise, for some s, I_s and I_{s+1} would contain $k_s + 1$ and $k_{s+1} + 1$ nodes, respectively. This is a contradiction as explained above. Using $\sum_{s=1}^{t}(3k_s + 2) = n$ we conclude that $\alpha(T) = n/3 - t/6$ for t even. Finally, if t is odd, similar arguments lead to $\alpha(T) = \sum_{s=1}^{t} k_s + (t-1)/2 = n/3 - t/6 - 1/2$ and the proof is complete. □

Recall that when $T = V$ we have $\alpha(T) = \alpha(G_n) = \lfloor n/3 \rfloor$. Therefore the inequality

$$\mathbf{x}(V) \leq \lfloor n/3 \rfloor \tag{3.4}$$

is valid for $P(G_n)$; this is the antiweb inequality introduced in [10]. It is easy to see that this inequality is nonredundant if and only if n is not a multiple of 3. In the remaining discussion we consider rank inequalities $x(T) \leq \alpha(T)$ for which $T \neq V$.

Lemma 3.2. *Let $T = I_1 + \cdots + I_t$ be a strict subset of V (where I_s are disjoint intervals) such that $x(T) \leq \alpha(T)$ is a facet of $P(G_n)$ different from each trivial and clique facet. Then the following holds:*

$$\begin{array}{ll} \text{(i)} & T \text{ is a 1-interval set;} \\ \text{(ii)} & |I_s| \equiv 1 \ (\mathrm{mod}\ 3) \ \text{ for } s = 1, \dots, t; \\ \text{(iii)} & t \text{ is odd and } t \geq 3. \end{array} \tag{3.5}$$

Proof. (i) Let F be the facet of $P(G_n)$ defined by $x(T) \leq \alpha(T)$. Assume that $i, i+1 \notin T$ for some $i \leq n$. We may assume that $i-1 \in T$ (otherwise another i could be chosen). Consider the clique $K = \{i-3, i-2, i-1\}$. Since F is not a clique facet, F has a root S with $S \cap K = \emptyset$. Note that $S \setminus \{i, i+1\}$ is also a root of F as $i, i+1 \notin T$, so we may assume that $i, i+1 \notin S$. Let $S' = S \cup \{i-1\}$, and observe that S' is a stable set in G_n. But $|S' \cap T| = |S \cap T| + 1 = \alpha + 1$, which contradicts the validity of $x(T) \leq \alpha(T)$. Thus, for each $i \leq n$, T contains either i or $i+1$ and therefore T is a 1-interval set.

(ii) Assume that $|I_s| \equiv 2 \pmod 3$, and let $I_s = \{l, \ldots, r\}$. Using the α-GREEDY algorithm starting in node $l-2$ it is easy to see that $\alpha(T) = \alpha(T \cup \{r+1\})$ (as we find an optimal stable set not containing the node $r+1$). But then $x(T) \leq \alpha(T)$ is the sum of the two valid inequalities $x(T \cup \{r+1\}) \leq \alpha(T \cup \{r+1\})$ and $-x_{r+1} \leq 0$, which contradicts that $x(T) \leq \alpha(T)$ defines a facet of $P(G_n)$. This proves that $|I_s| \not\equiv 2 \pmod 3$.

Assume next that $|I_s| \equiv 0 \pmod 3$, say $|I_s| = 3k$ for some $k \geq 1$. Let $I_s = \{l, \ldots, r\}$. There is a root S of $x(T) \leq \alpha(T)$ such that $S \cap I_s$ consists of the k nodes $l+1, l+4, \ldots, r-1$. Note that $S \cap \{l-1, l, r, r+1\} = \emptyset$. Thus the incidence vector of $S \setminus I_s$ must maximize $\mathbf{x}(T \setminus I_s)$ over the set of stable sets in G_n. Therefore $\mathbf{x}(T \setminus I_s) \leq |(S \setminus I_s) \cap T|$ is a valid inequality for $P(G_n)$. But $\mathbf{x}(I_s) \leq k$ is clearly a valid inequality as well, and if we add these two inequalities we obtain $x(T) \leq |S| = \alpha(T)$. This contradicts that $x(T) \leq \alpha(T)$ is nonredundant, and (ii) follows.

(iii) Assume that t is even. From the proof of Lemma 3.1 (and (ii)) it is clear that $\alpha(T)$ equals the sum of the stability numbers $\alpha(I_s \cup I_{s+1})$ for all $s \leq t$ being odd. As above, this means that the rank inequality $x(T) \leq \alpha(T)$ is redundant, a contradiction. Therefore, t must be odd. Furthermore, one can check that $\alpha(V \setminus \{i\}) = \alpha(V)$ for each $i \in V$. This implies that the rank inequality $\mathbf{x}(V \setminus \{i\}) \leq \alpha(V \setminus \{i\})$ is redundant as it is the sum of the rank inequality $\mathbf{x}(V) \leq \alpha(V)$ and the inequality $-x_i \leq 0$. Therefore t cannot be 1, so $t \geq 3$. □

Our next result characterizes *all* the nonredundant rank inequalities in an explicit way.

THEOREM 3.3. *Let $T = I_1 + \cdots + I_t \subseteq V$ be a strict subset of V. Then the rank inequality $x(T) \leq \alpha(T)$ defines a facet of $P(G_n)$ if and only if* (3.5) *holds.*

Proof. Due to Lemma 3.2 we need to show only the sufficiency of the conditions. Therefore, assume that (3.5) holds. Let $|I_s| = 3k_s + 1$ for $s \leq t$. If S is a stable set in G_n and $|S \cap I_s| = k_s + 1$ ($|S \cap I_s| = k_s$), we say that I_s is *closed* (*open*). From the proof of Lemma 3.1 and (3.5) it follows that a stable set is a root of $x(T) \leq \alpha(T)$ if and only if the intervals I_s alternate between being closed and open, except for one s where both I_s and I_{s+1} are open. For instance, for $t = 5$, we could have I_1, I_3, and I_5 open while I_2 and I_4 are both closed.

The face F of $P(G_n)$ induced by $x(T) \leq \alpha(T)$ is contained in some facet of $P(G_n)$; consider that such a facet is induced by the valid inequality $\sum_{j=1}^{n} b_j x_j \leq \beta$. We shall prove that $(b_1, \ldots, b_m)$ is a positive multiple of χ^T. This is done by exploiting symmetries of the stable sets of cardinality k_s on I_s. Let $s \leq t$, and let $I_s = \{l, l+1, \ldots, r\}$. Let i satisfy (if any) $i, i+1 \in I_s$. Consider the two (possibly empty) intervals $T_1 = \{l, l+1, \ldots, i-3\}$ and $T_2 = \{i+4, i+5, \ldots, r\}$. One can check that $\alpha(T_1) + \alpha(T_2) = k_s - 1$. Furthermore, there is a stable set S in $T_1 \cup T_2$ with $|S| = k_s - 1$ such that $|S \cap \{l, r\}| \leq 1$, say $r \notin S$. Therefore we can augment S into a stable set in G_n with $|S \cap T| = \alpha(T) - 1$ by adding nodes to S such that suitable intervals become open and closed. This means that the incidence vectors of both $S \cup \{i\}$ and $S \cup \{i+1\}$ are roots of $x(T) \leq \alpha(T)$ and therefore $\sum_j b_j \chi_j^{S\cup\{i\}} = \sum_j b_j \chi_j^{S\cup\{i+1\}}$, which gives $b_i = b_{i+1}$. This implies that b_j has the same value, say β_s, for all $j \in I_s$.

Assume that $k \in I_s$ and $k+2 \in I_{s+1}$ (so $k+1 \notin T$). Choose a root S of $x(T) \leq \alpha(T)$ for which both I_s and I_{s+1} are open. Using the α-GREEDY algorithm we find that $\alpha(T \setminus \{k-2, k-1, \ldots, k+4\}) = \alpha(T) - 1$. Arguments similar to those given above then give that $b_k = b_{k+2}$, so $\beta_s = \beta_{s+1}$. Since s is arbitrary, we have shown that $(b_1, \ldots, b_n)$ is a multiple of χ^T, and therefore $\mathbf{x}(T) \leq \alpha(T)$ induces a facet of $P(G_n)$. □

When T is a 1-interval set we call the rank inequality $\mathbf{x}(T) \leq \alpha(T)$ a 1-*interval inequality*. Note that for the nonredundant 1-interval inequalities the value of $\alpha(T)$ is known; see Lemma 3.1.

4. Completeness.

In this section we determine a complete and nonredundant linear description of the stable set polytope $P(G_n)$. Recall that each facet defining inequality $\mathbf{a}^T\mathbf{x} \leq \alpha$ which does not define a trivial facet must have nonnegative coefficients. The following result generalizes Lemma 3.2.

LEMMA 4.1. *Let $\mathbf{a}^T\mathbf{x} \leq \alpha$ define a facet F_a of $P(G_n)$ which is not a trivial, clique, or antiweb facet. Define $M = \max_j a_j$, and let $T = \{j \leq n : a_j = M\}$. Then the following statements hold:*

(i) *T is a 1-interval set, say $T = I_1 + \cdots + I_t$, where $t \geq 2$, and*

(ii) $|I_s| \equiv 1 \pmod 3$ *for $s = 1, \ldots, t$.*

Proof. (i) We first note that $T \neq V$ (otherwise $\mathbf{a}^T\mathbf{x} \leq \alpha$ would be equivalent to the antiweb inequality). We may then choose $i \in T$ such that $i-1 \notin T$ and therefore $a_{i-1} < M$. Since F_a is not a clique facet there is a root S of F_a with $S \cap K = \emptyset$, where $K = \{i, i+1, i+2\}$ (otherwise F_a would be contained in the facet induced by the clique inequality $\mathbf{x}(K) \leq 1$). Thus there is an interval $I = \{l+1, \ldots, r-1\}$ satisfying $l, r \in S$, $K \subseteq I$, and $S \cap I = \emptyset$. We may assume that S is chosen such that $|I|$ is minimal.

We observe that $l \in T$, i.e., $a_l = M$. Otherwise, $S' = (S \setminus \{l\}) \cup \{i\}$ would be a stable set whose incidence vector violates $\mathbf{a}^T\mathbf{x} \leq \alpha$. Therefore $l \neq i-1$ as $i-1 \notin T$. In fact we must have $l = i-2$; otherwise we could add the node i to S and violate the inequality $\mathbf{a}^T\mathbf{x} \leq \alpha$.

Thus we have shown that if $i \in T$ and $i-1 \notin T$, then $i-2 \in T$. This clearly implies that T is a 1-interval set $T = I_1 + \cdots + I_t$. If $t = 1$, then $T = V \setminus \{i\}$ for some i and it is easy to see that $\mathbf{a}^T\mathbf{x} \leq \alpha$ is implied by the antiweb inequality and the trivial inequality $-x_i \leq 0$. It follows that $t \geq 2$ and (i) holds.

(ii) Assume that $|I_s| \equiv 2 \pmod 3$ for some $s \leq t$, and let $I_s = \{l, \ldots, r\}$. Let S be a root of F_a with $l-1 \in S$ (such a root exists, otherwise F_a would be contained in the hyperplane given by $x_{l-1} = 0$). This implies that S also contains the set $\{j \in I_s : j \equiv l-1 \bmod (3)\} \cup \{r+1\}$ (i.e., nodes $l+2, l+5, \ldots$ lying in T plus $r+1$). Otherwise the distance between two consecutive nodes in S would be at least 4, and we could modify S by replacing $l-1$ by l, $l+2$ by $l+3$, etc. This produces a stable set violating $\mathbf{a}^T\mathbf{x} \leq \alpha$ because $a_{l-1} < a_l$. Thus each root containing $l-1$ also contains $r+1$. Due to symmetry, we conclude that a root of F_a contains $l-1$ if and only if it contains $r+1$. But this means that F_a is contained in the hyperplane given by $x_{l-1} - x_{r+1} = 0$, contradicting that F_a is a facet. This proves that $|I_s| \not\equiv 2 \pmod 3$ for all $s \leq t$.

Assume that $|I_s| \equiv 0 \pmod 3$ for some $s \leq t$, say $|I_s| = 3k$. Let $I_s = \{l, \ldots, r\}$. We observe, using similar arguments to those of the previous paragraph, that for each root S of F_a we have (a) S contains at most one of the nodes $l-1$ and $r+1$ and (b) if S contains either $l-1$ or $r+1$, then $|S \cap I_s| = k$.

Furthermore, because $|I_s| = 3k$, $\mathbf{x}(I_s) \leq k$ is a valid inequality for $P(G_n)$ obtained by adding k clique inequalities for nodes in I_s ($x_i+x_{i+1}+x_{i+2} \leq 1$, $x_{i+3}+x_{i+4}+x_{i+5} \leq 1$, etc.). Therefore there must be a root S of F_a satisfying $\mathbf{x}(I_s) \leq k$ with strict inequality, i.e., $|S \cap I_s| \leq k-1$. This implies, due to the observation above, that $|S \cap \{l-1, \dots, r+1\}| = k$. Let S' be the set obtained from S by replacing the (at most) $k-1$ nodes in $S \cap I_s$ by the k nodes $l+1, \dots, r-1$. Then S' is a stable set which violates $\mathbf{a}^T\mathbf{x} \leq \alpha$ (by an amount which is not smaller than M). This proves that $|I_s| \not\equiv 0 \pmod 3$ for all $s \leq t$ and the proof is complete. □

The next result concerns projection of facets. It gives a simple procedure for producing facets for $P(G_{n-3})$ from those of $P(G_n)$. The technique has some resemblance to a shrinking result given in [1].

Consider an inequality

$$\sum_{j=1}^{n} a_j x_j \leq \alpha \tag{4.1}$$

which defines a facet F_a of $P(G_n)$ which is different from each trivial, clique, or antiweb facet. (The procedure also works for the antiweb facet, but this is not of importance here.) As before, let $M = \max_j a_j$ and $T = \{j \leq n : a_j = M\}$. From Lemma 4.1 we have that T is a 1-interval set $T = I_1 + \cdots + I_t$ with $t \geq 2$ and $|I_s| \equiv 1 \pmod 3$ for each s. Consider the interval I_s and let $|I_s| = 3k+1$. Our procedure may be applied whenever $k \geq 1$. Assume, for notational simplicity, that $I_s = \{n-3k, \dots, n\}$ so that, in particular, $1 \notin T$. We then have the following result.

LEMMA 4.2. *The inequality*

$$\sum_{j=1}^{n-3} a_j x_j \leq \alpha - M \tag{4.2}$$

is valid for $P(G_{n-3})$. Moreover it defines a facet of $P(G_{n-3})$.

Proof. Assume that there is a stable set S in G_{n-3} with $\sum_{j=1}^{n-3} a_j \chi_j^S > \alpha - M$. Consider first the case when $n-3 \notin S$. Then S cannot contain both the nodes 1 and $n-4$ (because S is stable). Consider $1 \notin S$; the other case is treated similarly. Then $S' = S \cup \{n-1\}$ is a stable set in G_n and $\sum_{j=1}^{n} a_j \chi_j^{S'} > \alpha - M + M = \alpha$, which contradicts that (4.1) is valid for $P(G_n)$. Consider the remaining case when $n-3 \in S$. Then $1, 2 \notin S$ and therefore $S \cup \{n\}$ is a stable set in G_n and its incidence vector violates (4.1). It follows, by contradiction, that (4.2) is valid for $P(G_{n-3})$.

Assume that there is a root S of (4.1) with $S \cap \{n-3, n-2, n-1, n\} = \emptyset$. If $1 \in S$, we could violate (4.1) by replacing 1 by n, so we conclude that $1 \notin S$. But this is also impossible, for then we could add the node $n-1$ and violate (4.1). Thus every root of (4.1) contains either one or two nodes in the set $\{n-3, n-2, n-1, n\}$.

Since (4.1) defines a facet of $P(G_n)$ there is a nonsingular matrix $\mathbf{B} \in \mathbb{R}^{n,n}$ with rows being the incidence vectors of stable sets that are roots of (4.1). (Because $\mathbf{0}$ is not a root of the inequality, the affine rank and the linear rank of the roots coincide.) The columns of $\mathbf{B} = (b_{i,j})$ correspond to the nodes $1, \dots, n$. As shown above, each row of $\mathbf{B}$ contains one or two 1's in positions $n-3, n-2, n-1, n$ and, after a reordering of rows, we may assume that all the rows with two 1's in the mentioned positions are the last rows of $\mathbf{B}$. Let $\mathbf{B}' = (b'_{i,j}) \in \mathbb{R}^{n,n-3}$ be the matrix obtained by replacing the last four columns of $\mathbf{B}$ by a single column with ith entry being 1 if $\sum_{j=n-3}^{n} b_{i,j}$ equals 2 and 0 otherwise. By this construction it is clear that each row of $\mathbf{B}'$ is the incidence vector of a stable set in G_{n-3}.

We claim that $\text{rank}(\mathbf{B}') = n-3$. To prove this, let $\mathbf{b}_1, \dots, \mathbf{b}_{n-4} \in \mathbb{R}^n$ be the first $n-4$ columns of $\mathbf{B}'$ and $\mathbf{B}$ (these columns are equal in the two matrices). These vectors are linearly independent as $\mathbf{B}$ is nonsingular. Assume that the last column $\mathbf{b}^T = \left[\begin{array}{c} \mathbf{0}^T, \mathbf{1}^T \end{array}\right]$ of $\mathbf{B}'$ lies in the span L of the vectors $\mathbf{b}_1, \dots, \mathbf{b}_{n-4}$. Then $-\mathbf{b}$ also lies in L and there is an $\mathbf{x}' \in \mathbb{R}^{n-4}$ such that

$$\left[\begin{array}{c} \mathbf{b}_1, \dots, \mathbf{b}_{n-4} \end{array}\right] \mathbf{x}' = \left[\begin{array}{c} \mathbf{0} \\ -\mathbf{1} \end{array}\right].$$

Thus, with $\mathbf{x}^T = \left[\begin{array}{c} (\mathbf{x}')^T, 1, 1, 1, 1 \end{array}\right]$, we obtain that

$$\mathbf{B}\mathbf{x} = \left[\begin{array}{c} \mathbf{0} \\ -\mathbf{1} \end{array}\right] + \left[\begin{array}{c} \mathbf{1} \\ \mathbf{2} \end{array}\right] = \mathbf{1}$$

due to the structure of the last four columns of $\mathbf{B}$. On the other hand, the rows of $\mathbf{B}$ are the incidence vectors of roots of $\mathbf{a}^T\mathbf{x} \le \alpha$, so $\mathbf{B}\mathbf{a} = \alpha\mathbf{1}$. Since $\mathbf{B}$ is nonsingular we conclude that $\alpha\mathbf{x} = \mathbf{a}$ and therefore $a_j = \alpha$ for $n-3 \le j \le n$. Thus the inequality $\mathbf{a}^T\mathbf{x} \le \alpha$ has the form

$$\sum_{j=1}^{n-4} a_j x_j + \alpha \sum_{j=n-3}^{n} x_j \le \alpha.$$

Inserting the stable set χ^{n-1} we conclude that $a_j = 0$ for $2 \le j \le n-4$. This contradicts the form of $\mathbf{a}^T\mathbf{x} \le \alpha$, where $T = I_1 + \cdots + I_t$ with $t \ge 2$ and a_j positive (and maximal) for each $j \in T$. This proves that the columns of $\mathbf{B}'$ are linearly independent and it follows that $\mathbf{B}'$ has rank $n-3$. Thus there are $n-3$ linearly independent rows of $\mathbf{B}'$, and since all these are roots of (4.2) we have shown that (4.2) defines a facet of $P(G_{n-3})$. □

THEOREM 4.3. *For each n the stable set polytope $P(G_n)$ is the solution set of the nonnegativity constraints, the clique inequalities, the antiweb inequality* (3.4)*, and the nonredundant* 1*-interval inequalities described in Theorem* 3.3.

Proof. Let $\mathbf{a}^T\mathbf{x} \le \alpha$ be a facet defining inequality for $P(G_n)$ which is neither a nonnegativity constraint, a clique inequality, nor the antiweb constraint. We shall prove that the inequality is a positive multiple of some 1-interval inequality.

Due to Lemma 4.1 the inequality $\mathbf{a}^T\mathbf{x} \le \alpha$ may be written as

$$M \sum_{j \in T} x_j + \sum_{j \notin T} a_j x_j \le \alpha, \tag{4.3}$$

where $T = I_1 + \ldots + I_t$ is a 1-interval set and $M = \max_j a_j$. Recall also from Lemma 4.1 that $|I_s| \equiv 1 \pmod 3$ for $s = 1, \dots, t$. By repeated application of the reduction procedure of Lemma 4.2, say p times, we get an inequality

$$M \sum_{j \in T'} x_j + \sum_{j \notin T'} a_j x_j \le \alpha - pM \tag{4.4}$$

which defines a facet of $P(G_{n-3p})$. Here T' is a 1-interval set for which each interval consists of exactly one node. Thus, the coefficients in this inequality alternate between

M and numbers strictly smaller than M. We show that all of the numbers different from M are equal to 0.

Let $i \in T'$. Since (4.4) is not a clique facet, there is a root S of this inequality with $S \cap \{i, i+1, i+2\} = \emptyset$. Note that $a_{i+1} < a_i = a_{i+2} = M$ as T' is a 1-interval set. S cannot contain $i-1$ or $i+3$ (both outside T'), for then we could modify S by replacing that node by i or $i+2$ and violate (4.4). Thus $S \cap \{i-1, i, i+1, i+2, i+3\} = \emptyset$, which implies that $S \cup \{i+1\}$ is a stable set. But since S is a root, we see that $a_{i+1} = 0$. This shows that $a_i = 0$ for each $i \notin T'$ and therefore (4.4) is a positive multiple of a 1-interval inequality with all intervals of length 1 (the right-hand side must have the proper value, otherwise the inequality would be redundant). This also proves that the original inequality (4.3) is M times a 1-interval inequality (in particular, $\alpha = M\alpha(T)$) and the theorem follows. □

Examples. When $n = 9$ the minimal linear system for $P(G_9)$ consists of nonnegativity and clique constraints as well as the following 1-interval inequalities:

$$\begin{array}{cccccccccc}
 & x_2 & & +x_4 & & +x_6 & +x_7 & +x_8 & +x_9 & \leq 2 \\
 & x_2 & & +x_4 & +x_5 & +x_6 & +x_7 & & +x_9 & \leq 2 \\
 & x_2 & +x_3 & +x_4 & +x_5 & & +x_7 & & +x_9 & \leq 2 \\
x_1 & & +x_3 & & +x_5 & & +x_7 & +x_8 & +x_9 & \leq 2 \\
x_1 & & +x_3 & & +x_5 & +x_6 & +x_7 & +x_8 & & \leq 2 \\
x_1 & & +x_3 & +x_4 & +x_5 & +x_6 & & +x_8 & & \leq 2 \\
x_1 & +x_2 & & +x_4 & & +x_6 & & +x_8 & +x_9 & \leq 2 \\
x_1 & +x_2 & +x_3 & & +x_5 & & +x_7 & & +x_9 & \leq 2 \\
x_1 & +x_2 & +x_3 & +x_4 & & +x_6 & & +x_8 & & \leq 2.
\end{array}$$

These inequalities correspond to 1-interval sets with three intervals of cardinalities 4, 1, and 1. Consider next $n = 16$. Then the antiweb inequality $\mathbf{x}(V) \leq 5$ defines a facet of $P(G_{16})$. The inequality

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \quad + x_9 \quad + x_{11} \quad + x_{13} \quad + x_{15} \leq 4$$

is the 1-interval inequality for $T = I_1 + \cdots + I_5$, where $I_1 = \{1, \ldots, 7\}$, $I_2 = \{9\}$, $I_3 = \{11\}$, $I_4 = \{13\}$, $I_5 = \{15\}$. It defines a facet of $P(G_{16})$. Similarly, the 1-interval $T = I_1 + \cdots + I_5$ with $I_1 = \{1, 2, 3, 4\}$, $I_2 = \{6, 7, 8, 9\}$, $I_3 = \{11\}$, $I_4 = \{13\}$, $I_5 = \{15\}$ gives the inequality

$$x_1 + x_2 + x_3 + x_4 \quad + x_6 + x_7 + x_8 + x_9 \quad + x_{11} \quad + x_{13} \quad + x_{15} \leq 4.$$

In fact, the 1-interval sets that correspond to facets of $P(G_{16})$ all consist of $t = 5$ intervals with cardinalities either $7, 1, 1, 1, 1$ or $4, 4, 1, 1, 1$. The minimal linear system for $P(G_{16})$ consists of 48 inequalities corresponding to 1-interval sets in addition to the antiweb inequality, 16 clique inequalities, and 16 nonnegativity constraints.

REFERENCES

[1] F. Barahona and A.R. Mahjoub, *Compositions of graphs and polyhedra* II*: Stable sets*, SIAM J. Discrete Math., 7 (1994), pp. 359–371.

[2] G. Dahl, *The* 2*-hop spanning tree problem*, Oper. Res. Lett., to appear.

[3] U. Faigle, A.J. Hoffman, and W. Kern, *A characterization of nonnegative box-greedy matrices*, SIAM J. Discrete Math., 9 (1996), pp. 1–6.

[4] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.

[5] A. Hoffman, A.W.J. Kolen, and M. Sakarovitch, *Totally-balanced and greedy matrices*, SIAM J. Alg. Discrete Methods, 6 (1985), pp. 721–730.

[6] L. Lovász and M.D. Plummer, *Matching Theory*, North–Holland, Amsterdam, 1986.

[7] A.R. Mahjoub, *On the stable set polytope of a series-parallel graph*, Math. Programming, 40 (1988), pp. 53–57.

[8] G. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley, New York, 1988.

[9] A. Schrijver, *Theory of linear and integer programming*, John Wiley, Chichester, 1986.

[10] L.E. Trotter, *A class of facet producing graphs for vertex packing polyhedra*, Discrete Math., 12 (1975), pp. 373–388.

SIAM J. OPTIM.
Vol. 9, No. 2, pp. 504–525

SOLVING THE TRUST-REGION SUBPROBLEM USING THE LANCZOS METHOD*

NICHOLAS I. M. GOULD[†], STEFANO LUCIDI[‡], MASSIMO ROMA[‡], AND PHILIPPE L. TOINT[§]

Abstract. The approximate minimization of a quadratic function within an ellipsoidal trust region is an important subproblem for many nonlinear programming methods. When the number of variables is large, the most widely used strategy is to trace the path of conjugate gradient iterates either to convergence or until it reaches the trust-region boundary. In this paper, we investigate ways of continuing the process once the boundary has been encountered. The key is to observe that the trust-region problem within the currently generated Krylov subspace has a very special structure which enables it to be solved very efficiently. We compare the new strategy with existing methods. The resulting software package is available as HSL_VF05 within the Harwell Subroutine Library.

Key words. trust-region subproblem, Lanczos method, conjugate gradients, preconditioning

AMS subject classifications. 90C20, 90C30, 65K05, 65F10

PII. S1052623497322735

1. Introduction.

Trust-region methods for unconstrained minimization are blessed with both strong theoretical convergence properties and a good reputation in practice. The main computational step in these methods is to find an approximate minimizer of some *model* of the true objective function within a "trust" region for which a suitable norm of the correction lies within a given bound. This restriction is known as the *trust-region constraint*, and the bound on the norm is its *radius.* The radius is adjusted so that successive model problems mimic the true objective within the trust region.

The most widely used models are quadratic approximations to the objective function, as these are simple to manipulate and may lead to rapid convergence of the underlying method. From a theoretical point of view, the norm which defines the trust region is irrelevant so long as it is "uniformly" related to the ℓ_2-norm. From a practical perspective, this choice certainly affects the subproblem and thus the methods one can consider when solving it. The most popular practical choices are the ℓ_2- and ℓ_∞-norms and weighted variants thereof. In our opinion, it is important that the choice of norm reflects the underlying geometry of the problem; simply picking the ℓ_2-norm may not be adequate when the problem is large and the eigenvalues of the Hessian of the model widely spread. We believe that weighting the norm is essential for many large-scale problems.

In this paper, we consider the solution of the quadratic-model trust-region subproblem in a weighted ℓ_2-norm. We are interested in solving large problems and thus cannot rely solely on factorizations of the matrices involved. We thus concentrate on

[*] Received by the editors June 11, 1997; accepted for publication (in revised form) March 27, 1998; published electronically March 17, 1999. This research was supported in part by British Council–MURST grant ROM/889/95/53.

http://www.siam.org/journals/siopt/9-2/32273.html

[†] Department for Computation and Information, Rutherford Appleton Laboratory, Chilton, Oxfordshire, OX11 0QX, England (n.gould@rl.ac.uk).

[‡] Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza," via Buonarroti 12-00185, Roma, Italy (lucidi@dis.uniroma1.it, roma@dis.uniroma1.it).

[§] Department of Mathematics, Facultés Universitaires ND de la Paix, 61, rue de Bruxelles, B-5000 Namur, Belgium (pht@math.fundp.ac.be).

iterative methods. If the model of the Hessian is known to be positive definite and the trust-region radius sufficiently large that the trust region constraint is inactive at the unconstrained minimizer of the model, the obvious way to solve the problem is to use the preconditioned conjugate-gradient method. Note that the role of the preconditioner here is the same as the role of the norm used for the trust-region, namely, to change the underlying geometry so that the Hessian in the rescaled space is better conditioned. Thus, it will come as no surprise that the two should be intimately connected. Formally, we shall require that the weighting in the ℓ_2-norm and the preconditioning be performed by the *same* matrix.

When the radius is smaller than a critical value, the unconstrained minimizer of the model will no longer lie within the trust region and thus the required solution will lie on the trust-region *boundary.* The simplest strategy in this case is to consider the piecewise linear path connecting the conjugate-gradient iterates and to stop at the point where this path leaves the trust region. Such a strategy was first proposed independently by Steihaug [22] and Toint [23], and we shall refer to the terminating point as the *Steihaug–Toint* point. Remarkably, it is easy to establish the global convergence of a trust-region method based on such a simple strategy. The key is that global convergence may be proved provided that the accepted estimate of the solution has a model value no larger than at the Cauchy point (see [14]). The *Cauchy point* is simply the minimizer of the model within the trust region along the preconditioned steepest-descent direction. As the first segment on the piecewise-linear conjugate-gradient path gives precisely this point, and as the model value is monotonically decreasing along the entire path, the Steihaug–Toint strategy ensures convergence.

If the model Hessian is indefinite, the solution must also lie on the trust-region boundary. This case may also be simply handled using preconditioned conjugate gradients. Once again the piecewise linear path is followed until either it leaves the trust region or a segment with negative curvature is found. (A vector p is a direction of *negative curvature* if the inner product $\langle p, Hp \rangle < 0$, where H is the model Hessian.) In the latter case, the path is continued downhill along this direction of negative curvature as far as the constraint boundary. This variant was proposed in [22], while [23] suggests simply returning to the Cauchy point. As before, global convergence is ensured at either of these terminating points, as the objective function values there are no larger than at the Cauchy point. For consistency with the previous paragraph, we shall continue to refer to the terminating point in Steihaug's algorithm as the Steihaug–Toint point, although strictly Toint's point in this case may be different.

The Steihaug–Toint method is basically unconcerned with the trust region until it blunders into its boundary and stops. This is rather unfortunate, particularly as considerable experience has shown that this frequently happens during the first few, and often the first, iteration(s) when negative curvature is present. The resulting step is then barely, if at all, better than the Cauchy direction, and this may lead to a slow but globally convergent algorithm in theory and a barely convergent method in practice. In this paper, we consider an alternative which aims to avoid this drawback by trying harder to solve the subproblem when the boundary is encountered, while maintaining the efficiencies of the conjugate-gradient method so long as the iterates lie interior. The mechanism we use is the Lanczos method.

The paper is organized as follows. In section 2 we formally define the problem and any notation that we will use. The basis of our new method is given in section 3, while in section 4, we will review basic properties of the preconditioned conjugate-gradient and Lanczos methods. Our new method is given in detail in section 5. Some numerical experiments demonstrating the effectiveness of the approach are given in section 6, and a number of conclusions and perspectives are drawn in the final section.

2. The trust-region subproblem and its solution.

Let M be a symmetric positive-definite easily invertible approximation to the symmetric matrix H. Furthermore, define the M-norm of a vector as

$$\|s\|_M^2 = \langle s, Ms \rangle,$$

where $\langle \cdot, \cdot \rangle$ is the usual Euclidean inner product. In this paper, we consider the M-norm trust-region problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad q(s) \equiv \langle g, s \rangle + \tfrac{1}{2}\langle s, Hs \rangle \quad \text{subject to} \quad \|s\|_M \le \Delta \tag{2.1}$$

for some vector g and radius $\Delta > 0$.

A global solution to the problem is characterized by the following result.

THEOREM 2.1 (see [4], [20]). *Any global minimizer s^M of $q(s)$ subject to $\|s\|_M \le \Delta$ satisfies the equation*

$$H(\lambda^M) s^M = -g, \tag{2.2}$$

where $H(\lambda^M) \equiv H + \lambda M$ is positive semidefinite, $\lambda^M \ge 0$, and $\lambda^M(\|s^M\|_M - \Delta) = 0$. If $H(\lambda^M)$ is positive definite, s^M is unique.

This result is the basis of a series of related methods for solving the problem which are appropriate when forming factorizations of $H(\lambda) \equiv H + \lambda M$ for a number of different values of λ is realistic. Then either the solution lies interior, and hence $\lambda^{\mathrm{M}} = 0$ and $s^{\mathrm{M}} = -H^+ g$, or the solution lies on the boundary and λ^{M} satisfies the nonlinear equation

$$\|H(\lambda)^+ g\|_M = \Delta, \tag{2.3}$$

where H^+ denotes the pseudoinverse of H. Equation (2.3) is straightforward to solve using a safeguarded Newton iteration, except in the so-called hard case for which g lies in the null-space of $H(\lambda^{\mathrm{M}})$. In this case, an additional vector in the range-space of $H(\lambda^{\mathrm{M}})$ may be required if a solution on the trust-region boundary is sought. Goldfeldt, Quandt, and Trotter [5], Hebden [7], and Gay [4] all proposed algorithms of this form. The most sophisticated algorithm to date, that by Moré and Sorensen [9], is available as subroutine `GQTPAR` in the `MINPACK-2` package and guarantees that a nearly optimal solution will be obtained after a finite number of factorizations.

While such algorithms are appropriate for large problems with special Hessian structure—such as for band matrices—the demands of a factorization at each iteration limit their applicability for general large problems. It is for this reason that methods which do not require factorizations are of interest.

Throughout this paper, we shall denote the k by k identity matrix by I_k and its jth column by e_j. A set of vectors $\{q_i\}$ are said to be M-orthonormal if $\langle q_i, Mq_j \rangle = \delta_{ij}$, the Kronecker delta, and the matrix $Q_k = (q_0 \cdots q_k)$ formed from these vectors is an M-orthonormal matrix. The set of vectors $\{p_i\}$ are H-conjugate (or H-orthogonal) if $\langle p_i, Hp_j \rangle = 0$ for $i \ne j$.

3. A new algorithm for large-scale trust-region subproblems.

To set the scene for this paper, we recall that the Cauchy point may be defined as the solution to the problem

$$\underset{s \,\in\, \text{span}\left\{M^{-1}g\right\}}{\text{minimize}} \quad q(s) \equiv \langle g, s\rangle + \tfrac{1}{2}\langle s, Hs\rangle \quad \text{subject to} \quad \|s\|_M \le \Delta, \tag{3.1}$$

that is, as the minimizer of q within the trust region where s is restricted to the one-dimensional subspace span $\left\{M^{-1}g\right\}$. The dogleg methods (see [3], [13]) aim to solve the same problem over a particular two-dimensional subspace (a one-dimensional arc), while [19] does the same over a general two-dimensional subspace. In each of these cases the solution is easy to find as the search space is small. The difficulty with the general problem (2.1) is that the search space $\Re^n$ is large. This leads immediately to the possibility of solving a compromise problem

$$\underset{s \,\in\, \mathcal{S}}{\text{minimize}} \quad q(s) \quad \text{subject to} \quad \|s\|_M \le \Delta, \tag{3.2}$$

where $\mathcal{S}$ is a specially chosen subspace of $\Re^n$.

Now consider the Steihaug–Toint algorithm at an iteration k before the trust-region boundary is encountered. In this case, the point s_{k+1} is the solution to (3.2) with the set

$$\mathcal{S} = \mathcal{K}_k \overset{\text{def}}{=} \text{span}\left\{M^{-1}g, (M^{-1}H)M^{-1}g, (M^{-1}H)^2M^{-1}g, \ldots, (M^{-1}H)^kM^{-1}g\right\}, \tag{3.3}$$

the Krylov space generated by the starting vector $M^{-1}g$, and matrix $M^{-1}H$. That is, the Steihaug–Toint algorithm gradually widens the search space using the very efficient preconditioned conjugate-gradient method. However, as soon as the Steihaug–Toint algorithm moves across the trust-region boundary, the terminating point s_{k+1} no longer necessarily solves the problem (3.2) over the set (3.3); indeed it is very unlikely to do so when $k > 0$. (As the iterates generated by the method increase in M-norm, once an iterate leaves the trust region, the solution to (3.2)–(3.3), and thus (2.1), must lie on the boundary. See [22, Theorem 2.1] for details.) Can we do better? Yes, by recalling that the preconditioned conjugate-gradient and Lanczos methods generate different bases for the same Krylov space.

4. The preconditioned conjugate-gradient and Lanczos methods.

The preconditioned conjugate-gradient and Lanczos methods may be viewed as efficient techniques for constructing different bases for the same Krylov space, $\mathcal{K}_k$. The conjugate-gradient method aims for an H-conjugate basis, while the Lanczos method obtains an M-orthonormal basis.

ALGORITHM 4.1 (the preconditioned conjugate-gradient method). *Set $g_0 = g$, and let $v_0 = M^{-1}g_0$ and $p_0 = -v_0$. For $j = 0, 1, \ldots, k-1$, perform the iteration*

$$\alpha_j = \langle g_j, v_j\rangle / \langle p_j, Hp_j\rangle, \tag{4.1}$$

$$g_{j+1} = g_j + \alpha_j H p_j, \tag{4.2}$$

$$v_{j+1} = M^{-1}g_{j+1}, \tag{4.3}$$

$$\beta_j = \langle g_{j+1}, v_{j+1}\rangle / \langle g_j, v_j\rangle, \tag{4.4}$$

$$p_{j+1} = -v_{j+1} + \beta_j p_j. \tag{4.5}$$

Algorithm 4.2 (preconditioned Lanczos method). *Set $t_0 = g$, $w_{-1} = 0$, and, for $j = 0, 1, \ldots, k$, perform the iteration*

$$y_j = M^{-1} t_j, \tag{4.6}$$

$$\gamma_j = \sqrt{\langle t_j, y_j \rangle}, \tag{4.7}$$

$$w_j = t_j / \gamma_j, \tag{4.8}$$

$$q_j = y_j / \gamma_j, \tag{4.9}$$

$$\delta_j = \langle q_j, H q_j \rangle, \tag{4.10}$$

$$t_{j+1} = H q_j - \delta_j w_j - \gamma_j w_{j-1}. \tag{4.11}$$

The conjugate-gradient method generates the basis

$$\mathcal{K}_k = \text{span}\,\{p_0, p_1, \ldots, p_k\} \tag{4.12}$$

from Algorithm 4.1, while the Lanczos method generates the basis

$$\mathcal{K}_k = \text{span}\,\{q_0, q_1, \ldots, q_k\} \tag{4.13}$$

from Algorithm 4.2. The Lanczos iteration is often written in the more compact form

$$H Q_k - M Q_k T_k = \gamma_{k+1} w_{k+1} e_{k+1}^T \quad \text{and} \tag{4.14}$$

$$Q_k^T M Q_k = I_{k+1}, \tag{4.15}$$

where Q_k is the matrix $(q_0 \cdots q_k)$ and the matrix

$$T_k = \begin{pmatrix} \delta_0 & \gamma_1 & & & \\ \gamma_1 & \delta_1 & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \delta_{k-1} & \gamma_k \\ & & & \gamma_k & \delta_k \end{pmatrix} \tag{4.16}$$

is tridiagonal. It then follows directly that

$$Q_k^T H Q_k = T_k, \tag{4.17}$$

$$Q_k^T g = \gamma_0 e_1, \quad \text{and} \tag{4.18}$$

$$g = M y_0 = \gamma_0 M q_0. \tag{4.19}$$

The two methods are intimately related. In particular, so long as the conjugate-gradient iteration does not break down, the Lanczos vectors may be recovered from the conjugate-gradient iterates as

$$q_k = \sigma_k v_k / \sqrt{\langle g_k, v_k \rangle}, \text{ where } \sigma_k = -\text{sign}(\alpha_{k-1}) \sigma_{k-1} \text{ and } \sigma_0 = 1,$$

while the Lanczos tridiagonal may be expressed as

$$T_k = \begin{pmatrix} \frac{1}{\alpha_0} & \frac{\sqrt{\beta_0}}{|\alpha_0|} & & & & \\ \frac{\sqrt{\beta_0}}{|\alpha_0|} & \frac{1}{\alpha_1} + \frac{\beta_0}{\alpha_0} & \frac{\sqrt{\beta_1}}{|\alpha_1|} & & & \\ & \frac{\sqrt{\beta_1}}{|\alpha_1|} & \frac{1}{\alpha_2} + \frac{\beta_1}{\alpha_1} & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \frac{1}{\alpha_{k-1}} + \frac{\beta_{k-2}}{\alpha_{k-2}} & \frac{\sqrt{\beta_{k-1}}}{|\alpha_{k-1}|} \\ & & & & \frac{\sqrt{\beta_{k-1}}}{|\alpha_{k-1}|} & \frac{1}{\alpha_k} + \frac{\beta_{k-1}}{\alpha_{k-1}} \end{pmatrix}. \tag{4.20}$$

The conjugate-gradient iteration may break down if $\langle p_j, Hp_j \rangle = 0$, which can occur only if H is not positive definite, and will stop if $\langle g_j, v_j \rangle = 0$. On the other hand, the Lanczos iteration can fail only if $\mathcal{K}_j$ is an invariant subspace for $M^{-1}H$.

If $q(s)$ is convex in the manifold $\mathcal{K}_{j+1}$, the minimizer s_{j+1} of q in this manifold satisfies

$$s_{j+1} = s_j + \alpha_j p_j \tag{4.21}$$

so long as the initial value $s_0 = 0$ is chosen. Thus this estimate easily recurs from the conjugate-gradient iteration. The minimizers in successive manifolds may also be easily obtained using the Lanczos process, although the conjugate-gradient iteration is slightly less expensive and thus preferred.

The vector g_{j+1} in the conjugate-gradient method gives the gradient of $q(s)$ at s_{j+1}. It is quite common to stop the method as soon as this gradient is sufficiently small, and the method naturally records the M^{-1}-norm of the gradient, $\|g_{k+1}\|_{M^{-1}} = \langle g_j, v_j \rangle$. This norm is also available in the Lanczos method as

$$g_{k+1} = \gamma_{k+1} \langle e_{k+1}, h_k \rangle w_{k+1} \quad \text{and} \quad \|g_{k+1}\|_{M^{-1}} = \gamma_{k+1} |\langle e_{k+1}, h_k \rangle|, \tag{4.22}$$

where h_k solves the tridiagonal linear system $T_k h_k + \gamma_0 e_1 = 0$. The last component, $\langle e_{k+1}, h_k \rangle$, of h_k is available as a further by-product.

5. The truncated Lanczos approach.

Rather than use the preconditioned conjugate-gradient basis $\{p_0, p_1, \ldots, p_k\}$ for $\mathcal{S}$, we shall use the equivalent Lanczos M-orthonormal basis $\{q_0, q_1, \ldots, q_k\}$. The Lanczos basis has previously been used by [10]—to convexify the quadratic model—and [8]—to compute good directions of negative curvature—within linesearch-based methods for unconstrained minimization. We shall consider vectors of the form

$$s \in \mathcal{S} = \{s \in \Re^n \mid s = Q_k h\}$$

and seek

$$s_k = Q_k h_k, \tag{5.1}$$

where s_k solves the problem

$$\underset{s\, \in \mathcal{S}}{\text{minimize}} \quad q(s) \equiv \langle g, s \rangle + \tfrac{1}{2} \langle s, Hs \rangle \quad \text{subject to} \quad \|s\|_M \leq \Delta. \tag{5.2}$$

It then follows directly from (4.15), (4.17), and (4.18) that h_k solves the problem

$$\underset{h\, \in \mathbb{R}^{k+1}}{\text{minimize}} \quad \langle h, \gamma_0 e_1 \rangle + \tfrac{1}{2} \langle h, T_k h \rangle \quad \text{subject to} \quad \|h\|_2 \leq \Delta. \tag{5.3}$$

There are a number of crucial observations to be made here. First, it is important to note that the resulting trust-region problem involves the two-norm rather than the M-norm. Second, as T_k is tridiagonal, it is feasible to use the Moré–Sorensen algorithm to compute the model minimizer *even* when n is large. Third, having found h_k, the matrix Q_k is needed to recover s_k, and thus the Lanczos vectors will either need to be saved on backing store or regenerated. As we shall see, we need only Q_k once we are satisfied that continuing the Lanczos process will give little extra benefit. Fourth, one would hope that as a sequence of such problems may be solved, and as T_k

only changes by the addition of an extra diagonal and superdiagonal entry, solution data from one subproblem may be useful for starting the next. We consider this issue in section 5.2.

The basic trust-region solution classification theorem, Theorem 2.1, shows that

$$(T_k + \lambda_k I_{k+1}) h_k = -\gamma_0 e_1, \tag{5.4}$$

where $T_k + \lambda_k I_{k+1}$ is positive semidefinite, $\lambda_k \geq 0$, and $\lambda_k(\|h_k\|_2 - \Delta) = 0$. What does this tell us about s_k? First, using (4.17), (4.18), and (5.4) we have

$$Q_k^T(H + \lambda_k M)s_k = Q_k^T(H + \lambda_k M)Q_k h_k = (T_k + \lambda_k I_{k+1})h_k = -\gamma_0 e_1 = -Q_k^T g$$

and additionally that

$$\lambda_k(\|s_k\|_M - \Delta) = 0 \quad \text{and} \quad \lambda_k \geq 0. \tag{5.5}$$

Comparing these with the trust-region classification theorem, we see that s_k is the Galerkin approximation to s^{M} from the space spanned by Q_k.

We may then ask how good the approximation is. In particular, what is the error $(H + \lambda_k M)s_k + g$? The simplest way of measuring this error would be to calculate h_k and λ_k by solving (5.3), then to recover s_k as $Q_k h_k$, and finally to substitute s_k and λ_k into $(H + \lambda M)s + g$. However, this is inconvenient as it requires that we have easy access to Q_k. Fortunately there is a far better way.

Theorem 5.1.

$$(H + \lambda_k M)s_k + g = \gamma_{k+1}\langle e_{k+1}, h_k\rangle w_{k+1} \tag{5.6}$$

and

$$\|(H + \lambda_k M)s_k + g\|_{M^{-1}} = \gamma_{k+1}|\langle e_{k+1}, h_k\rangle|. \tag{5.7}$$

Proof. We have that

$$\begin{aligned} Hs_k &= HQ_k h_k \\ &= MQ_k T_k h_k + \gamma_{k+1}\langle e_{k+1}, h_k\rangle w_{k+1} \quad \text{from (4.14)} \\ &= -MQ_k(\lambda_k h_k + \gamma_0 e_1) + \gamma_{k+1}\langle e_{k+1}, h_k\rangle w_{k+1} \quad \text{from (5.4)} \\ &= -\lambda_k MQ_k h_k - \gamma_0 MQ_k e_1 + \gamma_{k+1}\langle e_{k+1}, h_k\rangle w_{k+1} \\ &= -\lambda_k Ms_k - \gamma_0 Mq_0 + \gamma_{k+1}\langle e_{k+1}, h_k\rangle w_{k+1} \\ &= -\lambda_k Ms_k - g + \gamma_{k+1}\langle e_{k+1}, h_k\rangle w_{k+1} \quad \text{from (4.19).} \end{aligned}$$

This then directly gives (5.6). Equation (5.7) follows from the M^{-1}-orthonormality of w_{k+1}. □

Therefore we can indirectly measure the error (in the M^{-1}-norm) knowing simply γ_{k+1} and the last component of h_k, and we do not need s_k or Q_k at all. Observant readers will notice the strong similarity between this error estimate and the estimate (4.22) for the gradient of the model in the Lanczos method, but this is not at all surprising as the two methods are aiming for the same point if the trust-region radius is large enough. An interpretation of (5.7) is also identical to that of (4.22). The error will be small when either γ_{k+1} or the last component of h_k is small.

We now consider the problem (5.3) in more detail. We say that a symmetric tridiagonal matrix is *reducible* if one or more of its off-diagonal entries is zero; otherwise it is *irreducible*. We then have the following preliminary result.

Lemma 5.2 (see also [11, Theorem 7.9.5]). *Suppose that the tridiagonal matrix T is irreducible and that v is an eigenvector of T. Then the first component of v is nonzero.*

Proof. By definition

$$Tv = \theta v \tag{5.8}$$

for some eigenvalue θ. Suppose that the first component of v is zero. Considering the first component of (5.8), we have that the second component of v is zero as T is tridiagonal and irreducible. Repeating this argument for the ith component of (5.8), we deduce that the $(i+1)$st component of v is zero for all i and hence that $v = 0$. But this contradicts the assumption that v is an eigenvector, and so the first component of v cannot be zero. □

This immediately yields the following useful result.

Theorem 5.3. *Suppose that T_k is irreducible. Then the hard case cannot occur for the subproblem* (5.3).

Proof. Suppose the hard case occurs. Then, by definition, $\gamma_0 e_1$ is orthogonal to v_k, the eigenvector corresponding to the leftmost eigenvalue, $-\theta_k$, of T_k. Thus, the first component of v_k is zero, which, following Lemma 5.2, contradicts the assumption that v_k is an eigenvector. Thus the hard case cannot occur. □

This result is important as it suggests that the full power of the Moré–Sorensen [9] algorithm is not needed to solve (5.3). We shall return to this in section 5.2. We also have an immediate corollary.

Corollary 5.4. *Suppose that T_{n-1} is irreducible. Then the hard case cannot occur for the original problem* (2.1).

Proof. When $k = n - 1$, the columns of Q_{n-1} form a basis for $\Re^n$. Thus the problems (2.1) and (5.2) are identical and (5.2) and (5.3) are related through a nonsingular transformation. The result then follows directly from Theorem 5.3 in the case $k = n - 1$. □

Thus, if the hard case occurs for (2.1), the Lanczos tridiagonal must become reducible at some stage.

Theorem 5.5. *Suppose that T_k is irreducible, that h_k and λ_k satisfy* (5.4)*, and that $T_k + \lambda_k I_{k+1}$ is positive semidefinite. Then $T_k + \lambda_k I_{k+1}$ is positive definite.*

Proof. Suppose that $T_k + \lambda_k I_{k+1}$ is singular. Then there is a nonzero eigenvector v_k for which $(T_k + \lambda_k I_{k+1})v_k = 0$. Hence, combining this with (5.4) reveals that

$$0 = \langle h_k, (T_k + \lambda_k I_{k+1})v_k\rangle = \langle v_k, (T_k + \lambda_k I_{k+1})h_k\rangle = -\gamma_0\langle v_k, e_1\rangle$$

and hence that the first component of v_k is zero. But this contradicts Lemma 5.2. Hence $T_k + \lambda_k I_{k+1}$ is both positive semidefinite and nonsingular and thus positive definite. □

This result implies that (5.4) has a unique solution. We now consider this solution.

Theorem 5.6. *Suppose that $\langle e_{k+1}, h_k\rangle = 0$. Then T_k is reducible.*

Proof. Suppose that T_k is irreducible. As the $(k+1)$st component of h_k is zero, then from the irreducibility of T_k and the $(k+1)$st equation of (5.4), we deduce that the kth component of h_k is zero. Repeating this argument for the $(i+1)$st equation of (5.4), we deduce that the ith component of h_k is zero for $1 \le i \le k$ and hence that $h_k = 0$. But this contradicts the first equation of (5.4), and thus T_k must be reducible. □

Thus we see that of the two possibilities suggested by Theorem 5.1 for obtaining an s_k for which $(H + \lambda_k M)s_k + g = 0$, it will be the possibility $\gamma_{k+1} = 0$ that occurs before $\langle e_{k+1}, h_k \rangle = 0$.

THEOREM 5.7. *Suppose that the hard case does not occur for* (2.1)*, and that* $\gamma_{k+1} = 0$*. Then* s_k *solves* (2.1).

Proof. If $\gamma_{k+1} = 0$, the Krylov space $\mathcal{K}_k$ is an invariant subspace of $M^{-1}H$, and by construction the first basis element of this space is $M^{-1}g$. As the hard case does not occur for (2.1), this space must also contain at least one eigenvector corresponding to the leftmost eigenvalue, $-\theta$, of $M^{-1}H$. Thus one of the eigenvalues of T_k must be $-\theta$, and $\lambda_k \geq \theta$ as $T_k + \lambda_k I_{k+1}$ is positive semidefinite. But this implies that $H + \lambda_k M$ is positive semidefinite, which combines with (5.1), (5.5), and Theorem 5.1 with $\gamma_{k+1} = 0$ to show that s_k satisfies the optimality conditions shown in Theorem 2.1. □

Thus we see that in the easy case, the required solution will be obtained from the first irreducible block of the Lanczos tridiagonal. It remains for us to consider the hard case. In view of Corollary 5.4, this case can only occur when T_k is reducible. Suppose therefore that T_k reduces into ℓ blocks of the form

$$T_k = \begin{pmatrix} T_{k_1} & & & \\ & T_{k_2} & & \\ & & \cdot & \\ & & & T_{k_\ell} \end{pmatrix}, \tag{5.9}$$

where each of the T_{k_i} defines an invariant subspace for $M^{-1}H$ and where the last block T_{k_ℓ} is the first to yield the leftmost eigenvalue, $-\theta$, of $M^{-1}H$. Then there are two cases to consider.

THEOREM 5.8. *Suppose that the hard case occurs for* (2.1)*, that* T_k *is as described by* (5.9)*, and that the last block* T_{k_ℓ} *is the first to yield the leftmost eigenvalue,* $-\theta$*, of* $M^{-1}H$*. Then,*

1. *if* $\theta \leq \lambda_{k_1}$*, a solution to* (2.1) *is given by* $s_k = Q_{k_1} h_{k_1}$*, where* h_{k_1} *solves the positive-definite system*

$$(T_{k_1} + \lambda_{k_1} I_{k_1+1}) h_{k_1} = -\gamma_0 e_1;$$

2. *if* $\theta > \lambda_{k_1}$*, a solution to* (2.1) *is given by* $s_k = Q_k h_k$*, where*

$$h_k = \begin{pmatrix} h \\ 0 \\ \cdot \\ 0 \\ \alpha u \end{pmatrix}, \tag{5.10}$$

h is the solution of the nonsingular tridiagonal system

$$(T_{k_1} + \theta I_{k_1+1}) h = -\gamma_0 e_1,$$

u is an eigenvector of T_{k_ℓ} *corresponding to* $-\theta$*, and* α *is chosen so that*

$$\|h_{k_1}\|_2^2 + \alpha^2 \|u\|_2^2 = \Delta^2.$$

Proof. In case 1, $H + \lambda_{k_1} M$ is positive semidefinite as $\lambda_{k_1} \geq \theta$ and the remaining optimality conditions are satisfied as $\gamma_{k_1+1} = 0$ and h_{k_1} solves (5.2). That $T_{k_1} + \lambda_{k_1} I_{k_1+1}$ is positive definite follows from Theorem 5.5. In case 2, $H + \theta M$ is positive semidefinite. Furthermore, as $\theta > \lambda_{k_1}$, it is easy to show that $\|h\|_2 < \|h_{k_1}\|_2 \leq \Delta$ and hence that there is a root α for which $\|s_k\|_M = \|h_k\|_2 = \Delta$. Finally, as each Q_{k_i} defines an invariant subspace, $HQ_{k_i} = MQ_{k_i}T_{k_i}$. Writing $s = Q_{k_1}h$ and $v = Q_{k_\ell}u$, we therefore have

$$Hs = HQ_{k_1}h = MQ_{k_1}T_{k_1}h = MQ_{k_1}(-\theta h - \gamma_0 e_1) = -\theta Ms - g$$

and

$$Hv = HQ_{k_\ell}u = MQ_{k_\ell}T_{k_\ell}u = -\theta MQ_{k_\ell}u = -\theta Mv.$$

Thus $(H + \theta M)s_k = -g$ and s_k satisfies all optimality conditions for (5.2). □

Notice that to obtain s_k as described in this theorem, we require only the Lanczos vectors corresponding to blocks one and, perhaps, ℓ of T_k.

We do not claim that solving the problem as outlined in Theorem 5.8 is realistic, as it relies on our being sure that we have located the leftmost eigenvalue of $M^{-1}H$. With Lanczos-type methods, one cannot normally guarantee that all eigenvalues, including the leftmost, will be found unless one ensures that all invariant subspaces have been investigated, and this may prove to be very expensive for large problems. In particular, the Lanczos algorithm, Algorithm 4.2, terminates each time an invariant subspace has been determined and must be restarted using a vector q which is M-orthonormal to the previous Lanczos directions. Such a vector may be obtained from the Gram–Schmidt process by reorthonormalizing a suitable vector—a vector with some component M-orthogonal to the existing invariant subspaces, perhaps a random vector—with respect to the previous Lanczos directions, which means that these directions will have to be regenerated or reread from backing store. Thus, while this form of the solution is of theoretical interest, it is unlikely to be of practical interest if a cheap approximation to the solution is all that is required.

5.1. The algorithm. We may now outline our algorithm, Algorithm 5.1, the generalized Lanczos trust-region (GLTR) method. We stress that, as our goal is merely to improve upon the value delivered by the Steihaug–Toint method, we do not use the full power of Theorem 5.8 and are content just to investigate the first invariant subspace produced by the Lanczos algorithm. In almost all cases, this subspace contains the global solution to the problem, and the complications and costs required to implement a method based on Theorem 5.8 are, we believe, prohibitive in our context.

ALGORITHM 5.1 (the GLTR method). *Let* $s_0 = 0$, $g_0 = g$, $v_0 = M^{-1}g_0$, $\gamma_0 = \sqrt{\langle v_0, g_0\rangle}$, *and* $p_0 = -v_0$. *Set the flag* `INTERIOR` *as true. For* $k = 0, 1, \ldots$ *until convergence, perform the iteration*

```
$\alpha_k = \langle g_k, v_k\rangle / \langle p_k, Hp_k\rangle$
Obtain $T_k$ from $T_{k-1}$ using (4.20)
If INTERIOR is true, but $\alpha_k \leq 0$ or $\|s_k + \alpha_k p_k\|_M \geq \Delta$
    reset INTERIOR to false.
If INTERIOR is true
    $s_{k+1} = s_k + \alpha_k p_k$
else
    solve the tridiagonal trust-region subproblem (5.3) to obtain $h_k$
end if
$g_{k+1} = g_k + \alpha_k Hp_k$
$v_{k+1} = M^{-1}g_{k+1}$
```

```
    If INTERIOR is true
        test for convergence using the residual $\|g_{k+1}\|_{M^{-1}}$
    else
        test for convergence using the value $\gamma_{k+1}|\langle e_{k+1}, h_k\rangle|$
    end if
    $\beta_k = \langle g_{k+1}, v_{k+1}\rangle / \langle g_k, v_k\rangle$
    $p_{k+1} = -v_{k+1} + \beta_k p_k$
If INTERIOR is false, recover $s_k = Q_k h_k$ by rerunning the recurrences or obtaining $Q_k$
from backing store.
```

When recovering $s_k = Q_k h_k$ by rerunning the recurrences, economies can be made by saving the α_i and β_i during the first pass and reusing them during the second. A potentially bigger savings may be made if one is prepared to accept a slightly inferior value of the objective function. The idea is simply to save the value of q at each iteration. On convergence, one looks back through this list to find an iteration, ℓ, say, for which a required percentage of the best value was obtained, recompute h_ℓ, and then accept $s_\ell = Q_\ell h_\ell$ as the required estimate of the solution. If the required percentage occurs at an iteration before the boundary is encountered, both the final point before the boundary and the Steihaug–Toint point are suitable and available without the need for the second pass.

We note that we have used the conjugate-gradient method (Algorithm 4.1) to generate the Lanczos vectors. If the inner-product $\langle p_k, Hp_k\rangle$ proves to be tiny, it is easy to continue using the Lanczos method (Algorithm 4.2) itself; the vectors

$$q_j = v_j / \sqrt{\langle g_j, v_j\rangle} \quad \text{and} \quad w_j = g_j / \sqrt{\langle g_j, v_j\rangle}$$

required to continue the Lanczos recurrence (4.11) are directly calculable from the conjugate-gradient method.

At each stage of both the Steihaug–Toint algorithm and our GLTR method (Algorithm 5.1), we need to calculate $\|s_k + \alpha p_k\|_M$. This issue is not discussed by Steihaug as it is implicitly assumed that M is available. However, it may be the case that all that is actually available is a procedure which returns $M^{-1}v$ for a given input v, and thus M is unavailable. Fortunately this is not a significant drawback as it is possible to calculate $\|s_k + \alpha p_k\|_M$ from available information.

To see this, observe that

$$\|s_k + \alpha p_k\|_M^2 = \|s_k\|_M^2 + 2\alpha\langle s_k, Mp_k\rangle + \alpha^2\|p_k\|_M^2 \tag{5.11}$$

and thus that we can find $\|s_{k+1}\|_M^2$ from $\|s_k\|_M^2$ so long as we already know $\langle s_k, Mp_k\rangle$ and $\|p_k\|_M^2$. But it is straightforward to show that these quantities may be calculated from the pair of recurrences

$$\langle s_k, Mp_k\rangle = \beta_{k-1}\left(\langle s_{k-1}, Mp_{k-1}\rangle + \alpha_{k-1}\|p_{k-1}\|_M^2\right) \quad \text{and} \tag{5.12}$$

$$\|p_k\|_M^2 = \langle g_k, v_k\rangle + \beta_{k-1}^2\|p_{k-1}\|_M^2, \tag{5.13}$$

where, of course, $\langle g_k, v_k\rangle$ has already been calculated as part of the preconditioned conjugate-gradient method.

5.2. Solving the irreducible tridiagonal trust-region subproblem.

In view of Theorem 5.3, the irreducible tridiagonal trust-region subproblem (5.3) is, in theory, easier to solve than the general problem. This is so both because the Hessian is tridiagonal (and thus very inexpensive to factorize) and because the hard case cannot occur. We should be cautious here because the so-called almost hard case—which occurs when g only has a tiny component in the range-space of $H(\lambda^{\text{M}})$—may still happen, and the trust-region problem in this case is naturally ill conditioned and thus likely to be difficult to solve.

The Moré–Sorensen [9] algorithm is based on being able to form factorizations of the model Hessian (which is certainly the case here as $T_k + \lambda I_{k+1}$ is tridiagonal), but does not try to calculate the leftmost eigenvalue of the pencil $H + \lambda M$. In the tridiagonal case, computing the extreme eigenvalues is straightforward, particularly if a sequence of related problems is to be solved. Thus, rather than using the Moré–Sorensen algorithm, we prefer the following method.

We restrict ourselves to the case where the solution lies on the trust-region boundary—we will only switch to this approach when the conjugate-gradient iteration leaves the trust region. The basic iteration is identical to that proposed in [9], namely, to apply Newton's method to

$$\phi(\lambda) \overset{\text{def}}{=} \frac{1}{\|h_k(\lambda)\|_2} - \frac{1}{\Delta} = 0, \tag{5.14}$$

where

$$(T_k + \lambda I_{k+1})h_k(\lambda) = -\gamma_0 e_1, \tag{5.15}$$

to find the required root λ_k. Recalling that we denote the leftmost eigenvalue of T_k by $-\theta_k$, the main difference between our approach and Moré and Sorensen's is that we always start from some point in the interval $[\max(0, \theta_k), \lambda_k]$—this interval is characterized by both $T_k + \lambda I_{k+1}$ being positive definite and $\|h_k(\lambda)\|_2 \geq \Delta$—as then the resulting Newton iteration is globally linearly, and asymptotically quadratically, convergent without any further safeguards. The Newton iteration is performed using Algorithm 5.2.

Algorithm 5.2 (Newton's method to solve $\phi(\lambda) = 0$). *Let $\lambda > \theta_k$ and $\Delta > 0$ be given.*

1. *Factorize $T_k + \lambda I_{k+1} = BDB^T$, where B and D are unit bidiagonal and diagonal matrices, respectively.*
2. *Solve $BDB^T h = -\gamma_0 e_1$.*
3. *Solve $Bw = h$.*
4. *Replace λ by*

$$\lambda + \left(\frac{\|h\|_2 - \Delta}{\Delta}\right)\left(\frac{\|h\|_2^2}{\|w\|_{D^{-1}}^2}\right).$$

The Newton correction in step 4 of this algorithm is given by

$$\lambda - \frac{\phi(\lambda)}{\phi'(\lambda)} = \lambda + \left(\frac{\|h\|_2 - \Delta}{\Delta}\right)\left(\frac{\|h\|_2^2}{\langle h, (T_k + \lambda I_{k+1})^{-1}h\rangle}\right),$$

while the exact form given is obtained by using the identity

$$\langle h, (T_k + \lambda I_{k+1})^{-1}h\rangle = \langle h, B^{-T}D^{-1}B^{-1}h\rangle = \langle B^{-1}h, D^{-1}B^{-1}h\rangle = \|w\|_{D^{-1}}^2,$$

where w is as computed in step 3. It is slightly more efficient to pick B to be unit upper-bidiagonal rather than unit lower-bidiagonal, as then step 2 simplifies to $B^T h = -\gamma_0 D^{-1} e_1$ because of the structure of the right-hand side.

To obtain a suitable starting value, two possibilities are considered. First, we attempt to use the solution value λ_{k-1} from the previous subproblem. Recall that T_k is merely T_{k-1} with an appended row and column. As we already have a factorization of $T_{k-1} + \lambda_{k-1} I_k$, it is trivial to obtain that of $T_k + \lambda_{k-1} I_{k+1}$ and thus to determine if the latter is positive definite. If $T_k + \lambda_{k-1} I_{k+1}$ turns out to be positive definite, $h_k(\lambda_{k-1})$ is computed from (5.15), and if $\|h_k(\lambda_{k-1})\|_2 \geq \Delta$, λ_{k-1} is used to start the Newton iteration.

Second, if λ_{k-1} is unsuitable, we monitor T_k to see if it is indefinite. This is trivial, as, for instance, the matrix is positive definite so long as all of the α_i, $0 \leq i \leq k$, generated by the conjugate-gradient method are positive. If T_k is positive definite, we start the Newton iteration with the value $\lambda = 0$, which by assumption gives $\|h_k(0)\|_2 \geq \Delta$ as the unconstrained solution lying outside the trust region. Otherwise, we determine the leftmost eigenvalue, $-\theta_k$, of T_k and start with $\lambda = \theta + \epsilon$, where ϵ is a small positive number chosen to make $T_k + \lambda_{k-1} I_{k+1}$ numerically "just" positive definite. By this we mean that its BDB^T factorization should exist, but that ϵ should be as small as possible. We have found that a value $(1 + \theta_k)\epsilon_m^{0.5}$, where ϵ_m is the unit roundoff, is almost always suitable, but we have added the precaution of multiplying this value by increasing powers of 2 so long as the factorization fails.

If we need to compute the leftmost eigenvalue of T_k, we use an iteration based upon the last-pivot function proposed by Parlett and Reid [12]. The *last-pivot* function, $\delta_k(\theta)$, is simply the value of the last diagonal entry of the BDB^T factor $D_k(\lambda)$ of $T_k - \theta I_{k+1}$. This value will be zero, and the other diagonal entries positive, when $\theta = \theta_k$ and $\delta_k(\theta) > 0$ for $\theta > \theta_k$. An interval of uncertainty $[\theta_l, \theta_u]$ is placed around the required root. The initial interval is given by the Gersgorin bounds on the leftmost eigenvalue. When it is known, the leftmost eigenvalue, $-\theta_{k-1}$, of T_{k-1} may be used to improve the lower bound because of the Cauchy interlacing property of the eigenvalues of T_{k-1} and T_k (see, for instance, [11, Theorem 10.1.2]). Given an initial estimate of θ_k, an improvement may be sought by applying Newton's method to $\delta_k(\theta)$; the derivative of δ_k is easy to obtain by recurrence. However, as Parlett and Reid point out,

$$\delta_k(\theta) = \frac{\det(T_k - \theta I_{k+1})}{\det(T_{k-1} - \theta I_k)}$$

and thus has a pole at $\theta = \theta_{k-1}$. Hence it is better to choose the new point by fitting the model

$$\delta_k^{\mathrm{M}}(\theta) = \frac{(\theta - a)(\theta - b)}{\theta - \theta_{k-1}} \tag{5.16}$$

to the function and derivative value at the current θ and then to pick the new iterate as the larger root of $\delta_k^{\mathrm{M}}(\theta)$. If the new iterate lies outside the interval of uncertainty, it is replaced by the midpoint of the interval. The interval is then contracted by computing δ_k at the new iterate and replacing the appropriate endpoint by the iterate. The iteration is stopped if the length of the interval or the value of $\delta_k(\theta_k)$ is small.

If θ_{k-1} is known, the initial iterate chosen as $\theta_{k-1} + \epsilon$ for some small positive $\epsilon \leq \theta_k - \theta_{k-1}$, and successive iterates generated from (5.16), the iterates converge globally, and asymptotically superlinearly, from the left. If the Newton iteration is used, the required root is frequently obscured and the scheme resorts to interval bisection. Thus the Parlett–Reid scheme is preferred.

Other means of locating the required eigenvalue based on using the determinant $\det(T_k - \theta I_{k+1})$ instead of $\delta_k(\theta)$ were tried, but proved to be less reliable because of the huge numerical range (and thus potential overflow) of the determinant.

6. Numerical experiments. The algorithm sketched in sections 5.1 and 5.2 has been implemented as a Fortran-90 module, `HSL_VF05`, within the Harwell Subroutine Library [6].

As our main interest is in using the methods described in this paper within a trust-region algorithm, we are particularly concerned with two issues. First, can we obtain significantly better values of the model by finding better approximations to its solution than the Steihaug–Toint method? And second, do better approximations to the minimizer of the model necessarily translate into fewer iterations of the trust-region method? In this section, we address these outstanding questions.

Throughout, we will consider the basic problem of minimizing an objective $f(x)$ of n real variables x. We shall use the following standard trust-region method.

ALGORITHM 6.1 (standard trust-region algorithm).

0. *An initial point* x_0 *and an initial trust-region radius* Δ_0 *are given, as are constants* ϵ_g, η_1, η_2, γ_1, *and* γ_2, *which are required to satisfy the conditions*

$$0 < \eta_1 \leq \eta_2 < 1 \text{ and } 0 < \gamma_1 < 1 \leq \gamma_2. \tag{6.1}$$

Set $k = 0$.

1. *Stop if* $\|\nabla_x f(x_k)\|_2 \leq \epsilon_g$.
2. *Define a second-order Taylor series model* q_k *and a positive-definite preconditioner* M_k. *Compute a step* s_k *to "sufficiently reduce the model"* q_k *within the trust region* $\|s\|_{M_k} \leq \Delta_k$.
3. *Compute the ratio*

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{q_k(x_k) - q_k(x_k + s_k)}. \tag{6.2}$$

If $\rho_k \geq \eta_1$, *let* $x_{k+1} = x_k + s_k$; *otherwise let* $x_{k+1} = x_k$.

4. *Set*

$$\Delta_{k+1} = \begin{cases} \gamma_2 \Delta_k & \text{if } \rho_k \geq \eta_2, \\ \Delta_k & \text{if } \rho_k \in [\eta_1, \eta_2), \\ \gamma_1 \Delta_k & \text{if } \rho_k < \eta_1. \end{cases} \tag{6.3}$$

Increment k *by one, and go to step* 1.

We choose the specific values $\epsilon_g = 0.00001$, $\eta_1 = 0.01$, $\eta_2 = 0.95$, $\gamma_1 = 0.5$, and $\gamma_2 = 2$ and set an upper limit of n iterations. The step s_k in step 2 is computed using either Algorithm 5.1 or the Steihaug–Toint algorithm. Convergence in both algorithms for the subproblem occurs as soon as

$$\|g_{k+1}\|_{M^{-1}} \leq \min(0.1, \|g_0\|_{M^{-1}}^{0.1}) \|g_0\|_{M^{-1}} \tag{6.4}$$

or if more than n iterations have been performed. In addition, of course, the Steihaug–Toint algorithm terminates as soon as the boundary is crossed.

All our tests were performed on an IBM RISC System/6000 3BT workstation with 64 Megabytes of RAM; the codes are all double precision Fortran-90, compiled under xlf90 with -O optimization, and IBM library BLAS are used. The test examples we consider are the larger examples from the CUTE test set [1] for which negative curvature is frequently encountered. Tests were terminated if more than 30 CPU minutes elapsed.

6.1. Can we get much better model values than Steihaug–Toint? We first consider problems of the form (2.1). Our test examples are generated by running Algorithm 6.1 on the CUTE set for 10 iterations and taking the trust-region subproblem at iteration 10 as our example. The idea here is to simulate the kind of subproblems which occur in practice, not those which result at the starting point for the algorithm, as such points frequently have special (favorable) properties.

Our aim is to see whether there is any significant advantage in continuing the minimization of the trust-region subproblem once the boundary of the trust region has been encountered. We ran HSL_VF05 to convergence, stopping when $\|g_{k+1}\|_{M^{-1}} \leq \max(10^{-15}, 10^{-5}\|g_0\|_{M^{-1}})$ or more than n iterations had been performed.

In all of the experiments reported here, the best value found was in fact the optimum value—a factorization of $H + \lambda M$ was used to confirm that the matrix was positive semidefinite, while the algorithm ensured that the remaining optimality conditions held—although, of course, there is no guarantee that this will always be the case. We measured the iteration (ST) and the percentage (ratio) of the optimal value obtained at the point at which the Steihaug–Toint method left the trust region, as well as the number of iterations taken to achieve 10%, 90%, and 99% of the optimal reduction (10%, 90%, 99%, respectively).

The results of these experiments are summarized in Table 6.1. In this table we give the name of each example used, along with its dimension n, and the statistics "ratio"(expressed in the form $x(y)$ as a shorthand for $x \times 10^{y}$), "ST," "10%," "90%," and "99%" as just described. Some of the problems had interior solutions, in which case the "ratio" and "ST" statistics are absent (as indicated by a dash). We considered both the unpreconditioned method ($M = I_n$) and a variety of standard preconditioners—a band preconditioner with semibandwidth of 5, and modified incomplete and sparse Cholesky factorizations, with the modifications as proposed in [18]—used by the LANCELOT package (see [2, Chapter 3]). The Cholesky factorization methods both failed for the problem MSQRTALS for which the Hessian matrix required too much storage.

We make a number of observations.

1. On some problems, the Steihaug–Toint point gives a model value which is a good approximation to the optimal value.
2. On other problems, a few extra iterations beyond the Steihaug–Toint point pay handsome dividends.
3. Getting to within 90% or even 99% of the best value very rarely requires many more iterations than to find the Steihaug–Toint point.

In conclusion, based on these numbers, we suggest that a good strategy would be to perform a few (say, 5) iterations beyond the Steihaug–Toint point and accept the improved point only if its model value is significantly better (as this will cost a second pass to compute the Lanczos vectors). We shall consider this further in the next section.

6.2. Do better values than Steihaug–Toint imply a better trust-region method? We now consider how the methods we have described for approximately solving the trust-region subproblem perform within a trust-region algorithm. Of particular interest is the question whether solving the subproblem more accurately reduces the number of trust-region iterations or more particularly the cost of solving the problem—the number of iterations is of concern if the evaluation of the objective

Table 6.1

A comparison of the number of iterations required to achieve a given percentage of the optimal model value for a variety of preconditioners. See the text for a key to the data.

		No preconditioner					5 band				
Example	n	Ratio	ST	10%	90%	99%	Ratio	ST	10%	90%	99%
BROYDN7D	1000	9(-1)	1	1	2	2	8(-3)	3	4	8	19
BRYBND	1000	3(-5)	23	24	28	39	5(-5)	1	2	6	17
CHAINWOO	1000	4(-5)	15	16	20	31	8(-1)	1	1	2	2
COSINE	1000	8(-1)	1	1	2	2	2(-13)	1	2	6	17
CRAGGLVY	1000	-	-	1	2	3	-	-	1	1	1
DIXMAANA	1500	-	-	1	1	1	-	-	1	1	1
DQRTIC	1000	8(-1)	1	1	2	2	8(-1)	1	1	2	2
EIGENALS	930	8(-1)	1	1	2	2	8(-1)	1	1	2	2
FREUROTH	1000	-	-	1	4	5	8(-1)	1	1	2	2
GENROSE	1000	8(-3)	8	9	9	10	8(-1)	1	1	2	2
HYDC20LS	99	5(-6)	23	25	29	40	8(-1)	1	1	3	3
MANCINO	100	8(-1)	1	1	2	5	8(-1)	1	1	2	2
MSQRTALS	1024	1(-1)	12	11	23	49	1(-5)	1	2	6	17
NCB20B	1000	3(-5)	65	66	70	81	2(-4)	96	97	101	112
NONCVXUN	1000	8(-1)	1	1	2	2	8(-1)	1	1	2	2
NONCVXU2	1000	8(-1)	1	1	2	2	8(-1)	1	1	2	2
SENSORS	100	7(-1)	1	1	2	7	7(-6)	1	2	6	16
SINQUAD	5000	1	3	2	2	2	6(-3)	11	11	12	13
SPARSINE	1000	4(-1)	44	1	50	54	8(-1)	1	1	2	2
SPMSRTLS	1000	4(-2)	5	5	6	7	2(-7)	1	2	6	17
		Incomplete Cholesky					Modified Cholesky				
Example	n	Ratio	ST	10%	90%	99%	Ratio	ST	10%	90%	99%
BROYDN7D	1000	6(-6)	1	2	6	17	6(-3)	2	3	7	18
BRYBND	1000	8(-1)	1	1	2	2	-	-	1	1	1
CHAINWOO	1000	8(-1)	1	1	2	2	8(-1)	1	1	2	2
COSINE	1000	8(-1)	1	1	2	2	7(-20)	1	2	6	17
CRAGGLVY	1000	-	-	1	1	1	-	-	1	1	1
DIXMAANA	1500	5(-1)	1	1	4	11	3(-11)	1	2	6	17
DQRTIC	1000	8(-1)	1	1	2	2	8(-1)	1	1	2	2
EIGENALS	930	1(-4)	1	2	6	17	2(-10)	1	2	6	17
FREUROTH	1000	8(-1)	1	1	2	2	8(-1)	1	1	2	2
GENROSE	1000	8(-1)	1	1	2	2	2(-7)	1	2	6	17
HYDC20LS	99	8(-1)	1	1	2	2	4(-1)	1	1	5	14
MANCINO	100	8(-1)	1	1	2	2	8(-1)	1	1	2	2
MSQRTALS	1024	factorization failure					factorization failure				
NCB20B	1000	1(-6)	2	3	7	18	2(-4)	3	4	8	19
NONCVXUN	1000	8(-1)	1	1	2	2	3(-5)	1	2	6	17
NONCVXU2	1000	7(-2)	1	2	6	17	3(-6)	1	2	6	17
SENSORS	100	8(-1)	1	1	2	2	7(-14)	1	2	6	16
SINQUAD	5000	-	-	2	2	2	-	-	1	1	1
SPARSINE	1000	5(-4)	1	2	6	17	1E(-10)	1	2	6	17
SPMSRTLS	1000	3(-7)	1	2	6	17	8E(-14)	1	2	6	17

function and its derivatives is the dominant cost, as then there is a direct correlation between the number of iterations and the overall cost of solving the problem.

In Tables 6.2 and 6.3, we compare the Steihaug–Toint scheme with the GLTR algorithm (Algorithm 5.1) run to high accuracy. We exclude the problem HYDC20LS for our reported results, as no method succeeded in solving the problem in fewer than our limit of n iterations, and the problems BROYDN7D and SPMSRTLS, as a number of different local minima were found. In these tables, in addition to the name and dimension of each example, we give the number of objective function ("#f") and derivative ("#g") values computed, the total number of matrix-vector products ("#prod") required to solve the subproblems, and the total CPU time required in seconds. We compare the same preconditioners M as we used in the previous section. We indicate those cases where one or another method performs at least 10% better than its competitor by highlighting the relevant figure in bold.

Table 6.2

A comparison of the Steihaug–Toint and exact model minimization techniques within a trust-region method, using a variety of preconditioners, for unconstrained minimization (part 1*). See the text for a key to the data.*

No preconditioner		Steihaug–Toint				Model optimum			
Example	n	#f	#g	#prods	CPU	#f	#g	#prods	CPU
BRYBND	1000	13	13	80	0.9	13	13	80	1.0
CHAINWOO	1000	$> n$ iterations				**865**	577	34419	**145.02**
COSINE	1000	11	11	14	0.1	11	11	14	0.1
CRAGGLVY	1000	19	19	130	1.0	19	19	130	0.9
DIXMAANA	1500	13	13	12	0.3	13	13	17	0.3
DQRTIC	1000	43	43	83	0.3	43	43	91	0.3
EIGENALS	930	68	56	1303	68.2	**52**	45	1107	**57.3**
FREUROTH	1000	17	17	34	0.4	17	17	34	0.4
GENROSE	1000	859	777	6092	**28.8**	**773**	642	24466	82.2
MANCINO	100	25	24	29	21.0	26	24	67	21.6
MSQRTALS	1024	44	34	7795	486.0	**32**	27	6009	**373.6**
NCB20B	1000	40	25	2057	**92.3**	**27**	16	7533	327.8
NONCVXUN	1000	**492**	466	177942	**1017.9**	$>$ 1800 seconds			
NONCVXU2	1000	414	381	3582	**26.2**	**335**	283	6987	44.0
SENSORS	100	20	19	37	**6.4**	20	19	140	8.8
SINQUAD	5000	182	114	363	24.3	**161**	106	382	24.6
SPARSINE	1000	15	15	3790	31.5	15	15	4143	34.4
5 band		Steihaug–Toint				Model optimum			
Example	n	#f	#g	#prods	CPU	#f	#g	#prods	CPU
BRYBND	1000	29	25	42	2.1	29	25	44	2.1
CHAINWOO	1000	**146**	99	145	**4.8**	191	123	196	6.3
COSINE	1000	21	15	20	0.4	21	15	30	0.5
CRAGGLVY	1000	22	22	21	1.1	22	22	21	1.1
DIXMAANA	1500	13	13	14	0.5	13	13	16	0.6
DQRTIC	1000	54	54	53	0.9	54	54	53	1.0
EIGENALS	930	56	43	171	75.2	53	42	222	75.8
FREUROTH	1000	20	18	19	0.8	20	18	17	0.8
GENROSE	1000	$> n$ iterations				$> n$ iterations			
MANCINO	100	91	72	90	87.2	**52**	43	90	**52.2**
MSQRTALS	1024	88	62	9793	**700.2**	**73**	52	19416	1292.2
NCB20B	1000	28	18	827	**41.2**	**23**	14	4775	214.4
NONCVXUN	1000	$> n$ iterations				$> n$ iterations			
NONCVXU2	1000	$> n$ iterations				$> n$ iterations			
SENSORS	100	**33**	29	38	**12.2**	45	38	197	19.3
SINQUAD	5000	239	154	753	67.0	**203**	133	806	65.4
SPARSINE	1000	**46**	37	3289	**32.4**	64	50	3678	36.9

We observe the following:

1. The use of different M leads to radically different behavior. Different preconditioners appear to be particularly suited to different problems. Surprisingly, perhaps, the unpreconditioned algorithm often performs the best overall.
2. In the unpreconditioned case, the model-optimum variant frequently requires significantly fewer function evaluations than the Steihaug–Toint method. However, the extra algebraic costs per iteration often outweigh the reduction in the numbers of iterations. The advantage in function calls for the other preconditioners is less pronounced.

Ideally, one would like to retain the advantage in numbers of function calls while reducing the cost per iteration. As we noted in section 6.1, one normally gets a good approximation to the optimal model value after a modest number of iterations. Moreover, while the Steihaug–Toint point often gives a significantly suboptimal value, a few extra iterations usually suffice to give a large percentage of the optimum. Thus, we next investigate both of these issues in the context of an overall trust-region method.

In Tables 6.4 and 6.5, we compare the number of function evaluations (#f) and the CPU time taken to solve the problem for the Steihaug–Toint ("ST") method

Table 6.3

A comparison of the Steihaug–Toint and exact model minimization techniques within a trust-region method, using a variety of preconditioners, for unconstrained minimization (part 2). See the text for a key to the data.

Incomplete Cholesky		Steihaug–Toint				Model optimum			
Example	n	#f	#g	#prods	CPU	#f	#g	#prods	CPU
BRYBND	1000	55	18	54	**3.9**	59	37	61	7.7
CHAINWOO	1000	174	115	173	**8.1**	183	121	309	10.3
COSINE	1000	22	17	26	**0.8**	22	19	49	1.2
CRAGGLVY	1000	22	22	21	1.5	22	22	21	1.5
DIXMAANA	1500	**16**	14	15	**0.8**	32	23	37	1.8
DQRTIC	1000	54	54	53	0.9	54	54	53	1.1
EIGENALS	930	**76**	52	76	**94.6**	89	60	112	111.1
FREUROTH	1000	$> n$ iterations				$> n$ iterations			
GENROSE	1000	948	629	951	35.5	**496**	322	847	**23.5**
MANCINO	100	29	27	30	125.0	31	28	32	130.1
MSQRTALS	1024	factorization failure				factorization failure			
NCB20B	1000	**34**	18	48	**23.2**	54	28	150	41.2
NONCVXUN	1000	$> n$ iterations				$> n$ iterations			
NONCVXU2	1000	$> n$ iterations				$> n$ iterations			
SENSORS	100	49	41	48	24.8	**44**	37	136	24.8
SINQUAD	5000	77	52	89	542.6	78	50	121	526.7
SPARSINE	1000	**90**	75	3465	**89.1**	135	109	4974	130.3
Modified Cholesky		Steihaug–Toint				Model optimum			
Example	n	#f	#g	#prods	CPU	#f	#g	#prods	CPU
BRYBND	1000	**15**	15	14	**2.2**	59	37	61	7.7
CHAINWOO	1000	178	119	177	**7.6**	183	121	309	10.3
COSINE	1000	41	25	40	1.1	**22**	19	49	1.2
CRAGGLVY	1000	23	23	33	1.4	22	22	21	1.6
DIXMAANA	1500	35	23	34	1.3	32	23	37	1.8
DQRTIC	1000	54	54	53	1.2	54	54	53	1.1
EIGENALS	930	133	92	132	167.8	**89**	60	112	**111.0**
FREUROTH	1000	$> n$ iterations				$> n$ iterations			
GENROSE	1000	462	332	463	**16.5**	496	322	847	23.4
MANCINO	100	31	28	30	129.3	31	28	32	130.1
MSQRTALS	1024	factorization failure				factorization failure			
NCB20B	1000	**38**	23	81	**26.1**	54	28	150	41.2
NONCVXUN	1000	$> n$ iterations				$> n$ iterations			
NONCVXU2	1000	$> n$ iterations				$> n$ iterations			
SENSORS	100	97	67	97	40.6	**44**	37	136	**24.8**
SINQUAD	5000	**14**	14	13	**99.4**	78	50	121	527.1
SPARSINE	1000	324	176	796	852.6	**135**	109	4974	**130.4**

with a number of variations on our basic GLTR method (Algorithm 5.1). The basic requirement is that we compute a model value which is at least 90% of the best value found during the first pass of the GLTR method. If this value is obtained by an iterate before that which gives the Steihaug–Toint point, the Steihaug–Toint point is accepted. Otherwise, a second pass is performed to recover the first point at which 90% of the best value was observed. The other ingredient is the choice of the stopping rule for the first pass. One possibility is to stop this pass as soon as the test (6.4) is satisfied. We denote this strategy by "90%best." The other possibility is to stop when either (6.4) is satisfied or at most a fixed number of iterations beyond the Steihaug–Toint point have occurred. We refer to this as "90%(ST+k)," where k gives the number of additional iterations allowed. We investigate the cases $k = 1, 5$, and 10. Once again, we compare the same preconditioners M as we used in the previous section. We highlight in bold those entries which are at least 10% better than the competition.

The conclusions are as broad as before. Each method has its successes and failures, and there is no clear overall best method or preconditioner, although the unpreconditioned version performs surprisingly well. Restricting the number of iterations allowed

Table 6.4

A comparison of a variety of GLTR techniques within a trust-region method, using a variety of preconditioners, for unconstrained minimization (part 1). See the text for a key to the data.

No preconditioner		ST		90%(ST+1)		90%(ST+5)		90%(ST+10)		90%best	
Example	n	#f	CPU	#f	CPU	#f	CPU	#f	CPU	#f	CPU
BRYBND	1000	13	0.9	13	0.9	13	0.9	13	1.0	13	1.0
CHAINWOO	1000	$> n$ its.		902	**61.8**	915	81.8	**884**	87.9	887	112.5
COSINE	1000	11	0.1	11	0.2	11	0.1	11	0.1	11	0.1
CRAGGLVY	1000	19	1.0	19	0.9	19	0.9	19	0.9	19	1.0
DIXMAANA	1500	13	0.3	13	0.3	13	0.3	13	0.3	13	0.3
DQRTIC	1000	43	0.3	43	0.3	43	0.3	43	0.3	43	0.3
EIGENALS	930	68	68.2	**59**	**61.5**	66	71.0	61	71.4	62	69.7
FREUROTH	1000	17	0.4	17	0.4	17	0.4	17	0.4	17	0.4
GENROSE	1000	859	**28.8**	748	38.9	**721**	48.1	738	57.3	728	60.0
MANCINO	100	25	21.0	24	20.2	24	20.2	24	20.4	24	20.4
MSQRTALS	1024	44	486.0	45	558.8	**35**	**394.2**	45	569.8	62	824.4
NCB20B	1000	40	**92.3**	40	104.7	45	141.1	33	104.6	**30**	182.3
NONCVXUN	1000	492	1017.9	**368**	**861.3**	> 1800 secs.		> 1800 secs.		433	1198.6
NONCVXU2	1000	414	26.2	263	**24.4**	272	29.7	270	31.4	292	36.2
SENSORS	100	20	**6.4**	23	7.3	21	8.1	21	8.0	21	8.1
SINQUAD	5000	182	24.3	**152**	**20.8**	152	21.7	152	21.4	152	21.5
SPARSINE	1000	15	**31.5**	16	36.4	16	36.5	16	36.5	16	36.6
5 band		ST		90%(ST+1)		90%(ST+5)		90%(ST+10)		90%best	
Example	n	#f	CPU	#f	CPU	#f	CPU	#f	CPU	#f	CPU
BRYBND	1000	29	2.1	29	2.1	29	2.1	29	2.1	29	2.1
CHAINWOO	1000	**146**	4.8	159	5.1	159	5.1	159	5.2	159	5.1
COSINE	1000	21	0.4	21	0.5	21	0.4	21	0.4	21	0.5
CRAGGLVY	1000	22	1.1	22	1.0	22	1.1	22	1.1	22	1.1
DIXMAANA	1500	13	0.5	13	0.6	13	0.6	13	0.6	13	0.6
DQRTIC	1000	54	0.9	54	0.9	54	1.0	54	1.0	54	1.0
EIGENALS	930	**56**	**75.2**	79	97.9	80	98.7	80	98.6	80	98.4
FREUROTH	1000	20	0.8	20	0.8	20	0.8	20	0.9	20	0.8
GENROSE	1000	$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.	
MANCINO	100	91	87.2	**52**	**51.8**	**52**	**51.8**	**52**	**52.0**	**52**	**51.8**
MSQRTALS	1024	88	700.2	97	756.7	**73**	704.9	74	844.7	79	981.5
NCB20B	1000	28	41.2	28	43.0	28	53.7	29	58.6	25	88.3
NONCVXUN	1000	$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.	
NONCVXU2	1000	$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.	
SENSORS	100	**33**	**12.2**	41	15.7	44	18.2	44	18.3	44	18.0
SINQUAD	5000	239	67.0	221	61.4	232	67.0	232	66.8	232	66.6
SPARSINE	1000	**46**	32.4	62	37.6	78	38.4	65	30.9	65	31.0

after the Steihaug–Toint point has been found appears to curb the worst behavior of the unrestricted method.

7. Perspectives and conclusions.

We have considered a number of methods which aim to find a better approximation to the solution of the trust-region subproblem than that delivered by the Steihaug–Toint scheme. These methods are based on solving the subproblem within a subspace defined by the Krylov space generated by the conjugate-gradient and Lanczos methods. The Krylov subproblem has a number of useful properties which lead to its efficient solution. The resulting algorithm is available as a Fortran-90 module, HSL_VF05 [6].

We must admit to being slightly disappointed that the new method did not perform uniformly better than the Steihaug–Toint scheme, and we were genuinely surprised that a more accurate approximation does not appear to significantly reduce the number of function evaluations within a standard trust-region method, at least in the tests we performed. While this may limit the use of the methods developed here, it also calls into question a number of other recent eigensolution-based proposals for solving the trust-region subproblem (see [15], [16], [17], [21]). While these authors demonstrate that their methods provide an effective means of solving the subproblem,

Table 6.5

A comparison of a variety of GLTR techniques within a trust-region method, using a variety of preconditioners, for unconstrained minimization (part 2). See the text for a key to the data.

Incomplete Cholesky		ST		90%(ST+1)		90%(ST+5)		90%(ST+10)		90%best	
Example	n	#f	CPU	#f	CPU	#f	CPU	#f	CPU	#f	CPU
BRYBND	1000	55	3.9	56	4.2	56	4.3	56	4.3	56	5.0
CHAINWOO	1000	**174**	**8.1**	199	9.7	199	10.1	199	10.2	199	10.1
COSINE	1000	**22**	**0.8**	45	1.9	45	1.9	45	1.9	45	2.0
CRAGGLVY	1000	22	1.5	22	1.6	22	1.6	22	1.5	22	1.6
DIXMAANA	1500	**16**	**0.8**	32	1.7	32	1.7	32	1.7	32	1.7
DQRTIC	1000	54	0.9	54	1.0	54	1.1	54	1.1	54	1.1
EIGENALS	930	76	94.6	77	97.2	74	97.2	74	97.3	74	96.8
FREUROTH	1000	$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.	
GENROSE	1000	948	35.5	**500**	**22.4**	**499**	**23.0**	**499**	**23.0**	**499**	**23.0**
MANCINO	100	29	125.0	31	129.6	31	130.1	31	129.7	31	129.9
MSQRTALS	1024	fact.	failure	fact.	failure	fact.	failure	fact.	failure	fact.	failure
NCB20B	1000	**34**	**23.2**	40	27.2	40	27.7	40	27.6	40	27.4
NONCVXUN	1000	$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.	
NONCVXU2	1000	$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.	
SENSORS	100	49	24.8	45	25.6	55	28.8	55	29.0	55	28.8
SINQUAD	5000	77	542.6	**68**	**484.2**	**68**	**484.1**	**68**	485.4	**68**	489.0
SPARSINE	1000	90	89.1	144	117.8	177	143.1	177	138.7	177	138.9
Modified Cholesky		ST		90%(ST+1)		90%(ST+5)		90%(ST+10)		90%best	
Example	n	#f	CPU	#f	CPU	#f	CPU	#f	CPU	#f	CPU
BRYBND	1000	15	2.2	15	2.2	15	2.3	15	2.2	15	2.2
CHAINWOO	1000	178	7.6	176	7.9	176	7.9	176	7.8	176	8.0
COSINE	1000	41	1.1	41	1.3	41	1.3	41	1.3	41	1.3
CRAGGLVY	1000	23	1.4	23	1.4	23	1.4	23	1.5	23	1.5
DIXMAANA	1500	35	1.3	35	1.5	35	1.4	35	1.4	35	1.4
DQRTIC	1000	54	1.2	54	1.2	54	1.3	54	1.3	54	1.3
EIGENALS	930	133	167.8	113	123.5	**63**	**85.4**	**63**	**85.5**	**63**	**86.2**
FREUROTH	1000	$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.	
GENROSE	1000	462	**16.5**	434	18.8	434	19.3	434	19.1	434	19.1
MANCINO	100	**31**	**129.3**	64	232.3	77	275.9	77	275.5	77	275.6
MSQRTALS	1024	fact.	failure	fact.	failure	fact.	failure	fact.	failure	fact.	failure
NCB20B	1000	38	26.1	**33**	**22.8**	**33**	26.8	**33**	26.8	**33**	26.4
NONCVXUN	1000	$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.	
NONCVXU2	1000	$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.		$> n$ its.	
SENSORS	100	97	40.6	72	32.6	**66**	**32.0**	**66**	**31.9**	**66**	**31.9**
SINQUAD	5000	14	99.4	14	99.9	14	100.0	14	99.7	14	99.7
SPARSINE	1000	324	852.6	**254**	**738.1**	361	1047.5	363	1063.7	363	1063.6

they make no effort to evaluate whether this is actually useful within a trust-region method. The results given in this paper suggest that this may not in fact be the case. This also leads to the interesting question as to whether it is possible to obtain useful low-accuracy solutions with these methods. We believe that further testing is needed to confirm the trends we have observed here.

We should not pretend that the formulae given in this paper are exact or even accurate in floating-point arithmetic. Indeed, it is well known that the floating-point matrices Q_k from the Lanczos method quickly lose M-orthonormality (see, for instance, [11, Section 13.3]). Despite this, the method as given appears to be capable of producing usable approximate solutions to the trust-region subproblem. We are currently investigating why this should be so.

One further possibility, which we have not considered so far, is to find an estimate λ using the first pass of Algorithm 5.1 and then to compute the required s by minimizing the unconstrained model $\langle g, s\rangle + \frac{1}{2}\langle s, (H + \lambda M)s\rangle$ using the preconditioned conjugate-gradient method. The advantage of doing this is that any instability in the first pass does not necessarily reappear in this auxiliary calculation. The disadvantages are that it may require more work than simply using (5.1) and that λ must be computed sufficiently large to ensure that $H + \lambda M$ is positive semidefinite.

Acknowledgments. We would like to thank John Reid for his helpful advice on computing eigenvalues of tridiagonal matrices and Jorge Moré for his useful comments on the Moré–Sorensen [9] method. Many thanks are also due to Jorge Nocedal and two anonymous referees.

REFERENCES

[1] I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint, CUTE*: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[2] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, LANCELOT*: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer Series in Computational Mathematics 17, Springer-Verlag, Heidelberg, Berlin, New York, 1992.

[3] J. E. Dennis and H. H. W. Mei, *Two new unconstrained optimization algorithms which use function and gradient values*, J. Optim. Theory Appl., 28 (1979), pp. 453–482.

[4] D. M. Gay, *Computing optimal locally constrained steps*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 186–197.

[5] S. M. Goldfeldt, R. E. Quandt, and H. F. Trotter, *Maximization by quadratic hill-climbing*, Econometrica, 34 (1966), pp. 541–551.

[6] Harwell Subroutine Library, *A Catalogue of Subroutines (Release* 13*)*, AEA Technology, Harwell, Oxfordshire, England, 1998, to appear.

[7] M. D. Hebden, *An Algorithm for Minimization Using Exact Second Derivatives*, Tech. Rep. T. P. 515, AERE Harwell Laboratory, Harwell, UK, 1973.

[8] S. Lucidi and M. Roma, *Numerical experience with new truncated Newton methods in large scale unconstrained optimization*, Comput. Optim. Appl., 7 (1997), pp. 71–87.

[9] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[10] S. G. Nash, *Newton-type minimization via the Lanczos method*, SIAM J. Numer. Anal., 21 (1984), pp. 770–788.

[11] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice–Hall, Englewood Cliffs, NJ, 1980; reprinted by SIAM, Philadelphia, PA, 1998.

[12] B. N. Parlett and J. K. Reid, *Tracking the progress of the Lanczos algorithm for large symmetric eigenproblems*, J. Inst. Math. Appl., 1 (1981), pp. 135–155.

[13] M. J. D. Powell, *A new algorithm for unconstrained optimization*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, London, New York, 1970, pp. 31–65.

[14] M. J. D. Powell, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, London, New York, 1975, pp. 1–27.

[15] F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *Max-min eigenvalue problems, primal-dual interior point algorithms, and trust region subproblems*, Optim. Methods Softw., 5 (1995), pp. 1–16.

[16] F. Rendl and H. Wolkowicz, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Math. Programming, Series B, 77 (1997), pp. 273–299.

[17] S. A. Santos and D. C. Sorensen, *A New Matrix-Free Algorithm for the Large-Scale Trust-Region Subproblem*, Tech. Rep. TR95-20, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1995.

[18] R. B. Schnabel and E. Eskow, *A new modified Cholesky factorization*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 1136–1158.

[19] G. A. Shultz, R. B. Schnabel, and R. H. Byrd, *A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties*, SIAM J. Numer. Anal., 22 (1985), pp. 47–67.

[20] D. C. Sorensen, *Newton's method with a model trust modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[21] D. C. Sorensen, *Minimization of a large-scale quadratic function subject to a spherical constraint*, SIAM J. Optim., 7 (1997), pp. 141–161.

[22] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[23] Ph. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., Academic Press, London, New York, 1981, pp. 57–88.

SIAM J. Optim.
Vol. 9, No. 2, pp. 526–549

THE $\mathcal{U}$-LAGRANGIAN OF THE MAXIMUM EIGENVALUE FUNCTION*

FRANÇOIS OUSTRY[†]

Abstract. In this paper we apply the $\mathcal{U}$-Lagrangian theory to the maximum eigenvalue function λ_1 and to its precomposition with affine matrix-valued mappings. We first give geometrical interpretations of the $\mathcal{U}$-objects that we introduce. We also show that the $\mathcal{U}$-Lagrangian of λ_1 has a Hessian which can be explicitly computed; the second-order development of the $\mathcal{U}$-Lagrangian provides a second-order development of λ_1 along a characteristic smooth manifold: the set of symmetric matrices whose maximal eigenvalues have a fixed multiplicity. The same results can be obtained when we precompose λ_1 with an affine matrix-valued mapping A, provided that this mapping satisfies a regularity condition (*transversality condition*). We show that the Hessian of the $\mathcal{U}$-Lagrangian of $\lambda_1 \circ A$ coincides with the reduced Hessian encountered in sequential quadratic programming. Finally, we use the $\mathcal{U}$-Lagrangian to derive second-order algorithms for minimizing $\lambda_1 \circ A$.

Key words. eigenvalue optimization, convex optimization, generalized derivative, second-order derivative

AMS subject classifications. Primary, 15A18, 52A41; Secondary, 65K10, 49J52

PII. S1052623496311776

1. Introduction.

1.1. Motivation. Optimization problems involving eigenvalues of symmetric matrices arise in many applications. (For a tutorial survey and numerous references, see [26].) Here, we focus our attention on the particular model problem

$$\text{(P)} \qquad \inf_{x\in\mathbb{R}^m} \lambda_1(A(x)),$$

where $\lambda_1(\cdot)$ is the maximum eigenvalue function and the mapping

$$A : \mathbb{R}^m \ni x \mapsto A_0 + \mathcal{A}\cdot x \tag{1.1}$$

is affine: A_0 is a given real $n \times n$ symmetric matrix and $\mathcal{A}$ is a linear operator from $\mathbb{R}^m$ to the space of $n \times n$ symmetric matrices.

Affine mappings $A(\cdot)$ cover a large class of engineering applications: control theory (see, e.g., [9] and the references therein), combinatorial optimization (e.g., [3], [37]), and structural design (see, e.g., [7], [10]).

The function $f := \lambda_1 \circ A$ is usually not differentiable when the maximum eigenvalue of $A(x)$ is multiple; yet f has a strong structure which can be exploited for algorithmic perspectives:

(s_1) f is the composite function of λ_1 and an affine mapping;

(s_2) λ_1 is a max-function over a compact set: from *Rayleigh's variational formulation* we have, for all $A \in \mathcal{S}_n$,

$$\lambda_1(A) = \max_{q\in\mathbb{R}^m, \|q\|=1} q^T A q\,;$$

(s_3) f is convex.

[*] Received by the editors November 8, 1996; accepted for publication (in revised form) March 23, 1998; published electronically March 17, 1999. This work was done at ENSTA-INRIA and was partially supported by a scholarship from DRET.

http://www.siam.org/journals/siopt/9-2/31177.html

[†] INRIA Rhone-Alpes, ZIRST, 655 avenue de l'Europe, 38330 Montbonnot Saint-Martin, France (Francois.Oustry@inria.fr).

The structural information (s_3), obviously implied by (s_1) and (s_2), needs to be emphasized since it is the minimal hypothesis required to get a *global complexity theory* [30]. Without any additional structure, convex analysis and nonsmooth optimization techniques [19] are the appropriate tools to minimize f: simple *ε-descent methods* as in [11] and [40] or more advanced *bundle methods* as in [23], [42], and [16] can be implemented.

Apart from this approach, which will be referred to as the classical one, a distinguished methodology consists of transforming the initial problem into a more structured one. This is the aim of interior-point methods. As Nesterov explains [31], path-following and potential reduction methods can be seen as a process that transforms the initial problem into an equivalent one which can be solved "easily" thanks to an addition of *structure*. *Self-concordance* is used to obtain the *polynomiality* of interior-point schemes. This is proved in a general framework in [32], [33] and more specifically in the framework of semidefinite programming (which includes (P)) in [2], [3], [20], [29], [45], [9, Chapter II, Notes and References]. Yet it seems that the initial transformation has a price: some local information is lost. Intuitively we understand that the "smoothing effect" of interior-point methods slightly modifies the local (second-order) information of the problem. In order to speed up the convergence of these schemes, *long-step* strategies [28], [34] or some geometrical information [41], [5] are then needed.

In this paper we address the following question: for a given structure $\{s_1, s_2(, s_3)\}$, what is the local information that is available to increase the speed of convergence of classical nonsmooth methods? This question was partially answered in the 1980s. Using sensitivity analysis results for eigenvalues and eigenprojections of [21], [24], as well as the pioneering work of [14], Overton introduced in [36] a local metric which enables him to obtain a quadratically converging algorithm for (P). This approach was then developed further in [37], [39], [38], [44], and [13]. Roughly speaking, assume that the multiplicity r of $\lambda_1(A(x^*))$ at an optimal point x^* is known; then the approach consists of minimizing the maximum eigenvalue subject to the constraint that its multiplicity is r. A local C^2-parametrization of (P) is then used to develop a successive quadratic programming method. In [39, page 102], the authors explain that two subspaces are playing a crucial role at x: the subspace parallel to the affine hull of $\partial f(x)$ and its orthogonal complement. In fact, they show that while second-order information is needed in the first subspace, a first-order approximation is enough in the second one. Our motivation here, therefore, is to prove that the recent theory of $\mathcal{U}$-Lagrangians [25] formalizes these observations and enables us to insert the geometrical approach proposed by Overton in a convex analysis framework. This will give us a better understanding of the method. Since our point of view remains simultaneously local and conceptual, the comparison with interior-point methods cannot be completely achieved here. The next step, which is the subject of another paper [35], consists of showing that the $\mathcal{U}$-Lagrangian theory can be used to derive second-order bundle methods which enjoy the following properties:

1. a minimizing sequence is always generated (global);
2. the rate of convergence (of the sequence of iterates) is asymptotically quadratic when some nondegeneracy assumptions hold (local).

Note that similar work has been under investigation in the discrete minimax context [27].

Our paper is organized as follows. After recalling some useful results from the $\mathcal{U}$-Lagrangian theory in section 2 and from differential geometry in section 3, we give our main result in section 4: the $\mathcal{U}$-Lagrangian of the maximum eigenvalue is a C^∞ convex function. For practical (i.e., numerical) application, we also show how its first- and second-order derivatives can be explicitly computed. Then in section 5 we come back to the space of decision variables $\mathbb{R}^m$ and establish chain rules to obtain the first- and second-order derivatives of the $\mathcal{U}$-Lagrangian of the composite function $\lambda_1 \circ \mathcal{A}$. To go further than first order, an additional condition is needed: the *transversality condition* from differential geometry. Finally, we present a conceptual scheme which coincides, in a more implementable version, with the superlinear algorithm described in [38, Iteration 4].

1.2. Basic notation and terminology. Our notation closely follows that of [19] and [3].

- $\mathbb{R}^m$, m-dimensional Euclidean space;
- $\langle x, y\rangle$, scalar product of $x, y \in \mathbb{R}^m$;
- $\|x\| := \sqrt{\langle x, x\rangle}$, Euclidean norm of $x \in \mathbb{R}^m$;
- $\mathcal{U}^\perp$, orthogonal subspace of the subspace $\mathcal{U}$;
- $u \oplus v$, direct sum of $u \in \mathcal{U}$ and $v \in \mathcal{U}^\perp$;
- $\operatorname{proj}_{\mathcal{U}} : \mathbb{R}^m \to \mathcal{U}$, projection operator onto the subspace $\mathcal{U}$ of $\mathbb{R}^m$;
- $\operatorname{proj}_{\mathcal{U}}^* : \mathcal{U} \to \mathbb{R}^m$, canonical injection $\mathcal{U} \ni u \mapsto u \oplus 0 \in \mathbb{R}^m$;
- $\langle \cdot, \cdot \rangle_{\mathcal{U}}$, scalar product induced by $\langle \cdot, \cdot \rangle$ in $\mathcal{U}$;
- $\| \cdot \|_{\mathcal{U}}$, norm induced by $\| \cdot \|$ in $\mathcal{U}$;
- $\operatorname{aff} C$, affine hull of the nonempty set $C \subset \mathbb{R}^m$;
- $\operatorname{ri} C$, relative interior of the convex set C;
- $\operatorname{span} C$, linear subspace generated by the nonempty set $C \subset \mathbb{R}^m$;
- $x + C$, sum of the singleton $\{x\} \subset \mathbb{R}^m$ and the set $C \subset \mathbb{R}^m$;
- $B(x, \delta)$, open ball centered at $x \in \mathbb{R}^m$ with radius $\delta > 0$;
- $\mathcal{S}_n$, space of $n \times n$ symmetric matrices;
- $\mathcal{S}_n^+$, cone of positive semidefinite matrices;
- $A \succ 0$, the matrix $A \in \mathcal{S}_n$ is a positive definite matrix;
- $\operatorname{tr} A := \sum_{i=1}^n A_{ii}$, trace of the matrix $A \in \mathcal{S}_n$;
- $A \bullet B := \operatorname{tr} AB$, Fröbenius scalar product of $A, B \in \mathcal{S}_n$;
- $\|A\| := \sqrt{A \bullet A}$, Fröbenius norm of $A \in \mathcal{S}_n$;
- $A^\dagger$, Moore–Penrose inverse of A;
- $\lambda_1(A) \geq \cdots \geq \lambda_n(A)$, eigenvalues of $A \in \mathcal{S}_n$ in decreasing order;
- $E_1(A)$, first eigenspace of $A \in \mathcal{S}_n$, i.e., the eigenspace associated with $\lambda_1(A)$;
- $Q_1(A)$, orthonormal basis of $E_1(A)$, i.e., a matrix whose columns are orthonormal and generate $E_1(A)$;
- $\mathcal{A}^* : \mathcal{S}_n \to \mathbb{R}^m$, adjoint operator of the linear operator $\mathcal{A} : \mathbb{R}^m \to \mathcal{S}_n$;
- $\mathcal{M}_r := \{A \in \mathcal{S}_n : \lambda_1(A) = \cdots = \lambda_r(A) > \lambda_{r+1}(A)\}$, set of symmetric matrices whose maximum eigenvalue has a given multiplicity r; this is a C^∞-submanifold of $\mathcal{S}_n$ (see [6]).

Additional notation from differential geometry will be given in section 3.

2. The $\mathcal{U}$-Lagrangian of a convex function.

We briefly recall here the theory developed in [25]. Let $f : \mathbb{R}^m \to \mathbb{R}$ be a finite-valued convex function. For a given $x \in \mathbb{R}^m$, we start by defining a decomposition of the space $\mathbb{R}^m = \mathcal{U}(x) \oplus \mathcal{V}(x)$. The subspaces $\mathcal{U}(x)$ and $\mathcal{V}(x)$ are equivalently defined as follows.

Definition 2.1.

(i) *$\mathcal{V}(x)$ is the subspace parallel to* $\operatorname{aff} \partial f(x)$ *and* $\mathcal{U}(x) = \mathcal{V}(x)^\perp$.

(ii) *For any* $g \in \operatorname{ri} \partial f(x)$, *$\mathcal{U}(x)$ and $\mathcal{V}(x)$ are, respectively, the normal and tangent cones to $\partial f(x)$ at g.*

Given $g \in \partial f(x)$, define the *$\mathcal{U}$-Lagrangian* of f at x by

$$\mathcal{U}(x) \ni u \mapsto L_{\mathcal{U}}(u) := \min_{v \in \mathcal{V}(x)} \{ f(x + u \oplus v) - \langle \operatorname{proj}_{\mathcal{V}(x)} g, v \rangle_{\mathcal{V}(x)} \}. \tag{2.1}$$

Associated with (2.1) we have the set of minimizers (possibly empty)

$$\mathcal{U}(x) \ni u \hookrightarrow w(u) := \operatorname{Argmin}_{v \in \mathcal{V}(x)} \{ f(x + u \oplus v) - \langle \operatorname{proj}_{\mathcal{V}(x)} g, v \rangle_{\mathcal{V}(x)} \} \subset \mathcal{V}(x). \tag{2.2}$$

These definitions make sense (recall [25, Lemma 3.1]), and we have the following result.

THEOREM 2.2 (see [25, Theorems 3.1–3.2]). *The function $L_{\mathcal{U}}$ of* (2.1) *is convex. In addition, if $g \in \operatorname{ri} \partial f(x)$, the set $w(u)$ defined in* (2.2) *is not empty and the following properties hold:*

(i) *Its subdifferential is*

$$\partial L_{\mathcal{U}}(u) = \operatorname{proj}_{\mathcal{U}(x)} [\partial f(x + u \oplus v) \cap (g + \mathcal{U}(x))] \,, \tag{2.3}$$

where v is taken arbitrarily in $w(u)$.

(ii) *When $u = 0$, we have $w(0) = \{0\}$ and $L_{\mathcal{U}}(0) = f(x)$. Moreover, $L_{\mathcal{U}}$ is differentiable at* 0 *and*

$$\nabla L_{\mathcal{U}}(0) = \operatorname{proj}_{\mathcal{U}(x)} g \,. \qquad \Box \tag{2.4}$$

We now prove another result in the framework of multifunctions.

COROLLARY 2.3. *With the notation above, the following hold:*

(i) *The multifunction $u \hookrightarrow \partial L_{\mathcal{U}}(u)$ is continuous at $u = 0$:*

$$\lim_{u \to 0} \partial L_{\mathcal{U}}(u) = \{ \nabla L_{\mathcal{U}}(0) \} \,. \tag{2.5}$$

(ii) *For all $u \in \mathcal{U}(x)$, we have*

$$\partial f(x + u \oplus v) \cap (g + \mathcal{U}(x)) = \partial L_{\mathcal{U}}(u) \oplus \{ \operatorname{proj}_{\mathcal{V}(x)} g \} \text{ for all } v \in w(u) \,. \tag{2.6}$$

(iii) *Denoting by $\partial f(x + u \oplus w(u)) \cap (g + \mathcal{U}(x))$ the right-hand side of* (2.6)*, the multifunction $u \hookrightarrow \partial f(x + u \oplus w(u)) \cap (g + \mathcal{U}(x))$ is continuous at* 0*:*

$$\lim_{u \to 0} \partial f(x + u \oplus w(u)) \cap (g + \mathcal{U}(x)) = \{ g \} \,. \tag{2.7}$$

Proof. (i) From Theorem 2.2, $L_{\mathcal{U}}$ is convex; then from [19, Theorem VI.6.2.4], $\partial L_{\mathcal{U}}$ is outer semicontinuous. This, with the differentiability of $L_{\mathcal{U}}$ at $u = 0$, implies that all the subgradients at u tend to $\nabla L_{\mathcal{U}}(0)$ when u tends to 0. The inner semicontinuity then follows: $\partial L_{\mathcal{U}}$ is actually continuous at 0.

(ii) This is a direct consequence of (2.3).

(iii) This is straightforward from (i) and (ii). $\Box$

With no additional assumptions, we can go beyond first-order analysis.

THEOREM 2.4 (see [25, Corollary 3.3]).

$$\sup_{v \in w(u)} \|v\| = o(\|u\|) \,. \tag{2.8}$$

We will often use a weaker (first-order) version of Theorem 2.4.

Corollary 2.5. *The multifunction $u \hookrightarrow w(u)$ is continuous at $u = 0$:*

$$\lim_{u \to 0} w(u) = \{0\}\,.$$

Proof. From Theorem 2.4, $w(u) \subset w(0) + B(0, o(\|u\|))$. This proves the outer semicontinuity of w at 0. Now use the fact that $w(0)$ is a singleton to get the inner semicontinuity. □

Note that the function $u \mapsto L_{\mathcal{U}}(u)$ depends not only on x but also on g. In what follows, we use the notation $L_{\mathcal{U}}(x, g; u)$ and $\nabla L_{\mathcal{U}}(x, g; u)$, and since there is no confusion, ∇ will always mean derivation with respect to u. In this sense, as is done for the classical Lagrangian, we will call (x, g) a *primal-dual* pair.

3. Some differential geometry. For the convenience of the reader we recall some basic concepts from differential geometry. We assume only that the definitions of C^∞-manifolds and C^∞-submanifolds are known. (To get a solid understanding of the essentials, refer to Hicks [17].) Sometimes we will omit the prefix C^∞ (C^2 would be enough for our purposes). We give here some more notation.

- $\mathcal{S}$ and $\mathcal{T}$ are two Euclidean spaces;
- $\mathrm{T}_{\mathcal{M}}(A)$ and $\mathrm{N}_{\mathcal{M}}(A)$ are, respectively, the tangent and normal spaces to the submanifold $\mathcal{M}$ at $A \in \mathcal{M}$;
- $\ker \mathcal{D}$ and $\operatorname{range} \mathcal{D}$ are, respectively, the kernel and the range of the linear operator $\mathcal{D} : \mathcal{S} \to \mathcal{T}$;
- $\phi : B(\hat{A}, \delta_0) \to \mathcal{T}$ with $\delta_0 > 0$ is a C^∞-map;
- $\mathrm{D}\phi(A)$ is the differential of ϕ at $A \in B(\hat{A}, \delta_0)$.

Definition 3.1 (regular value). *We say that $Z \in \mathcal{T}$ is a* regular value *of ϕ if for each $A \in \phi^{-1}(Z) := \{\Omega \in B(\hat{A}, \delta_0) : \phi(\Omega) = Z\}$, $\mathrm{D}\phi(A)$ is surjective.* □

Theorem 3.2 (submersion theorem). *Let Z be a regular value of ϕ. Then the level set $\phi^{-1}(Z)$ is a submanifold of $\mathcal{S}$ whose tangent space is $\mathrm{T}_{\phi^{-1}(Z)}(A) = \ker \mathrm{D}\phi(A)$.*

Proof. The proof can be found in a more general framework in Abraham, Marsden, and Ratiu [1, Theorem 3.5.4]. □

Definition 3.3 (local equation of a submanifold). *Assume that 0 is a regular value of ϕ and $\mathcal{M} \cap B(\hat{A}, \delta_0) = \phi^{-1}(0)$. Then we say that $\phi(A) = 0$ is a* local equation *of $\mathcal{M}$ in $B(\hat{A}, \delta_0)$.* □

Observe that, via Theorem 3.2,

$$\mathrm{T}_{\mathcal{M}}(A) = \ker \mathrm{D}\phi(A) \text{ for all } A \in \mathcal{M} \cap B(\hat{A}, \delta_0)\,. \tag{3.1}$$

The next theorem and its corollary introduce the idea of *tangential parametrization* of a manifold. This concept is not standard in differential geometry, yet it will be interesting in our $\mathcal{U}$-context.

Theorem 3.4. *Let $\phi(A) = 0$ be a local equation of $\mathcal{M}$ in $B(\hat{A}, \delta_0)$. Then, there exists a scalar δ such that $0 < \delta \leq \delta_0$ and a unique map*

$$v : \mathrm{T}_{\mathcal{M}}(\hat{A}) \cap B(0, \delta) \to \mathrm{N}_{\mathcal{M}}(\hat{A})$$

such that, for all $(u, v) \in (\mathrm{T}_{\mathcal{M}}(\hat{A}), \mathrm{N}_{\mathcal{M}}(\hat{A}))$,

$$\Big(\ \|u\| \leq \delta,\ \|v\| \leq \delta \ \text{ and } \ \phi(\hat{A} + u \oplus v) = 0\ \Big) \Rightarrow v = v(u)\,. \tag{3.2}$$

The map v is C^∞, and at $u = 0$ we have

$$\mathrm{D}v(0) = 0\,. \tag{3.3}$$

Proof. For $\hat{A} \in \mathcal{M} \cap B(\hat{A}, \delta_0)$, consider the C^∞-map

$$\psi : (\mathrm{T}_{\mathcal{M}}(\hat{A}), \mathrm{N}_{\mathcal{M}}(\hat{A})) \ni (u, v) \mapsto \phi(\hat{A} + u \oplus v).$$

Then the partial differential $\mathrm{D}_v\psi(0, 0)$ is given by

$$\mathrm{D}_v\psi(0, 0) = \mathrm{D}\phi(\hat{A}) \circ \mathrm{proj}^*_{\mathrm{N}_{\mathcal{M}}(\hat{A})}.$$

Now, 0 is a regular value of ϕ and $\hat{A} \in \phi^{-1}(0)$. Hence the implicit function theorem (see, e.g., [1, Theorem 2.5.7]) applies: there exists a map $v(\cdot)$ satisfying (3.2) and such that, for all $u \in B(0, \delta)$,

$$\mathrm{D}v(0) = -[\mathrm{D}_v\psi(0, 0)]^{-1}[\mathrm{D}_u\psi(0, 0)],$$

where

$$\mathrm{D}_u\psi(0, 0) = \mathrm{D}\phi(\hat{A}) \circ \mathrm{proj}^*_{\mathrm{T}_{\mathcal{M}}(\hat{A})}.$$

To derive (3.3), recall that $\mathrm{T}_{\mathcal{M}}(\hat{A}) = \ker \mathrm{D}\phi(\hat{A})$. □

We isolate the following consequence of Theorem 3.4 for the purpose of our $\mathcal{U}$-analysis in section 4.

COROLLARY 3.5. *Let $\hat{A} \in \mathcal{M}$; then there exists $\delta > 0$ such that*

$$\mathrm{proj}_{\mathrm{N}_{\mathcal{M}}(\hat{A})} d = v(\mathrm{proj}_{\mathrm{T}_{\mathcal{M}}(\hat{A})} d) \tag{3.4}$$

for all $d \in B(0, \delta)$ satisfying $\hat{A} + d \in \mathcal{M}$.

Proof. Take $\delta < \delta_0$ given by Theorem 3.4 and $d \in B(0, \delta)$ to obtain

$$\max\left\{\|\mathrm{proj}_{\mathrm{T}_{\mathcal{M}}(\hat{A})} d\|_{\mathrm{T}_{\mathcal{M}}(\hat{A})}, \|\mathrm{proj}_{\mathrm{N}_{\mathcal{M}}(\hat{A})} d\|_{\mathrm{N}_{\mathcal{M}}(\hat{A})}\right\} \leq \|d\| \leq \delta.$$

Since $A + d \in \mathcal{M}$ can be written

$$\phi\left(\hat{A} + \mathrm{proj}_{\mathrm{T}_{\mathcal{M}}(\hat{A})} d \oplus \mathrm{proj}_{\mathrm{N}_{\mathcal{M}}(\hat{A})} d\right) = 0,$$

the conclusion follows using Theorem 3.4. □

In other words, Corollary 3.5 says that the map

$$\pi_{\hat{A}} : \mathrm{T}_{\mathcal{M}}(\hat{A}) \cap B(0, \delta) \ni u \mapsto \hat{A} + u \oplus v(u)$$

covers a whole neighborhood of $\hat{A}$ in $\mathcal{M}$. This enables us to call $\pi_{\hat{A}}$ a *tangential parametrization* of the submanifold $\mathcal{M}$ at $\hat{A}$.

We now consider a C^∞-map $A : \mathbb{R}^m \to \mathcal{S}$, and we address the following question: when is the set $A^{-1}(\mathcal{M})$ a C^∞-submanifold of $\mathbb{R}^m$?

DEFINITION 3.6 (transversal map). *Let $\hat{x} \in \mathbb{R}^m$; the C^∞ map $A(\cdot)$ is said to be transversal to the submanifold $\mathcal{M}$ at $\hat{x}$ if $A(\hat{x}) \in \mathcal{M}$ and the range of $\mathrm{D}A(\hat{x})$ is transversal to the subspace $\mathrm{T}_{\mathcal{M}}(A(\hat{x}))$, i.e.,*

$$\mathrm{range}\, \mathrm{D}A(\hat{x}) + \mathrm{T}_{\mathcal{M}}(A(\hat{x})) = \mathcal{S}. \qquad \square \tag{3.5}$$

THEOREM 3.7. *Let $\hat{x} \in A^{-1}(\mathcal{M}) \subset \mathbb{R}^m$. If $A(\cdot)$ is transversal to $\mathcal{M}$ at $\hat{x}$, then $A^{-1}(\mathcal{M})$ is a C^∞-submanifold in a neighborhood of $\hat{x}$, i.e., there exists $\rho > 0$*

such that $B(\hat{x},\rho)\cap A^{-1}(\mathcal{M})$ is a C^∞-submanifold of $\mathbb{R}^m$. Moreover, for all $x \in B(\hat{x},\rho)\cap A^{-1}(\mathcal{M})$, we have

$$T_{A^{-1}(\mathcal{M})}(x) = [\mathrm{D}A(x)]^{-1}\mathrm{T}_{\mathcal{M}}(A(x))\,. \tag{3.6}$$

Proof. Apply a local version of the *transversal mapping theorem* (see, e.g., [1, Theorem III.5.12]). □

When $A(\cdot)$ is transversal to $\mathcal{M}$ at $\hat{x}$, we derive a local equation of $A^{-1}(\mathcal{M})$.

THEOREM 3.8. *Let $\hat{x} \in A^{-1}(\mathcal{M}) \subset \mathbb{R}^m$ be such that $A(\cdot)$ is transversal to $\mathcal{M}$ at $\hat{x}$, and let $\phi(A) = 0$ be a local equation of $\mathcal{M}$ in a neighborhood of $A(\hat{x})$. Then there exists $\rho > 0$ such that*

(i) *the map $A(\cdot)$ is transversal to $\mathcal{M}$ at $x \in B(\hat{x},\rho)$,*

(ii) *the equation $\phi(A(x)) = 0$ is a local equation of $A^{-1}(\mathcal{M}) \cap B(\hat{x},\rho)$.*

Proof. (i) By a composition rule, we have

$$\mathrm{D}(\phi\circ A)(\hat{x}) = \mathrm{D}\phi(A(\hat{x}))\circ \mathrm{D}A(\hat{x})\,. \tag{3.7}$$

The transversality of $A(\cdot)$ to $\mathcal{M}$ at $\hat{x}$ is then equivalent to the surjectivity of $\mathrm{D}(\phi\circ A)(\hat{x})$; by continuity, this holds in a whole neighborhood of $\hat{x}$. Hence (i) follows.

(ii) Now define $\varphi : B(\hat{x},\rho) \ni x \mapsto \phi(A(x))$; from (i), 0 is a regular value of φ. It is now obvious that

$$A^{-1}(\mathcal{M}) \cap B(\hat{x},\rho) = \varphi^{-1}(0),$$

hence, according to Definition 3.3, $\varphi(x) = 0$ is a local equation of $A^{-1}(\mathcal{M})\cap B(\hat{x},\rho)$. □

4. $\mathcal{U}$-Lagrangian of the maximum eigenvalue function.

4.1. The subspaces $\mathcal{U}$ and $\mathcal{V}$. We now study the maximum eigenvalue function $\mathcal{S}_n \ni A \mapsto \lambda_1(A) \in \mathbb{R}$. To apply the results of section 2, we identify the space $\mathcal{S}_n$ with $\mathbb{R}^{\frac{n(n+1)}{2}}$. As far as notation is concerned, we replace lowercase by capital letters to stress the fact that we work now with spaces of matrices. At a point $A \in \mathcal{S}_n$, we consider the subspaces $\mathcal{U}(A)$ and $\mathcal{V}(A)$ of Definition 2.1. Let $r \geq 1$ be the multiplicity of $\lambda_1(A)$, i.e., A lies on the submanifold $\mathcal{M}_r$ (see section 1.2). Let $Q_1(A)$ be an orthonormal basis of $E_1(A)$ (see section 1.2); then a well-known description of $\partial\lambda_1(A)$ can be obtained.

THEOREM 4.1 (see [37, Theorem 3], [18, Theorem 3.1]).

$$\partial\lambda_1(A) = \{Q_1(A)ZQ_1(A)^T : Z \in \mathcal{S}_r^+,\ \operatorname{tr} Z = 1\}. \qquad \square \tag{4.1}$$

We also have an explicit formulation for the relative interior of $\partial\lambda_1(A)$.

PROPOSITION 4.2. *The relative interior of $\partial\lambda_1(A)$ has the expression*

$$\operatorname{ri}\partial\lambda_1(A) = \{Q_1(A)ZQ_1(A)^T : Z \in \mathcal{S}_r,\ Z \succ 0,\ \operatorname{tr} Z = 1\}. \tag{4.2}$$

Proof. Use (4.1) to write

$$\begin{aligned}\operatorname{ri}\partial\lambda_1(A) &= \operatorname{ri}\{Q_1(A)ZQ_1(A)^T : Z \in \mathcal{S}_r^+,\ \operatorname{tr} Z = 1\}\\ &= Q_1(A)\operatorname{ri}\{Z \in \mathcal{S}_r^+,\ \operatorname{tr} Z = 1\}Q_1(A)^T \qquad \text{(by [19, Proposition III.2.1.12])}\\ &= Q_1(A)\{Z \in \mathcal{S}_r,\ Z \succ 0,\ \operatorname{tr} Z = 1\}Q_1(A)^T \quad \text{(by [19, Proposition III.2.1.10]).}\end{aligned}$$

This completes the proof. □

We are now in a position to give a characterization of the subspaces $\mathcal{U}(A)$ and $\mathcal{V}(A)$.

THEOREM 4.3. *The subspaces $\mathcal{U}(A)$ and $\mathcal{V}(A)$ are, respectively, characterized by*

$$\mathcal{U}(A) = \left\{U \in \mathcal{S}_n : Q_1(A)^T U Q_1(A) - \frac{1}{r}\operatorname{tr}\,(Q_1(A)^T U Q_1(A)) I_r = 0\right\} \tag{4.3}$$

and

$$\mathcal{V}(A) = \{Q_1(A) Z Q_1(A)^T : Z \in \mathcal{S}_r,\ \operatorname{tr} Z = 0\}. \tag{4.4}$$

Proof. Take an element of $\operatorname{ri} \partial\lambda_1(A)$, for instance, its center $C_r := \frac{1}{r} Q_1(A) Q_1(A)^T$. By Definition 2.1 (ii), $\mathcal{U}(A)$ is the normal cone to $\partial\lambda_1(A)$ at C_r; then $U \in \mathcal{U}(A)$ means for all $Z \in \mathcal{Z} := \{Z \in \mathcal{S}_r^+, \operatorname{tr} Z = 1\}$, it holds that

$$0 \geq U \bullet (Q_1(A) Z Q_1(A)^T - C_r) = U \bullet Q_1(A)[Z - I_r/r] Q_1(A)^T,$$

which is in turn equivalent to

$$\max_{Z \in \mathcal{Z}} Q_1(A)^T U Q_1(A) \bullet Z \leq \frac{1}{r}\operatorname{tr} Q_1(A)^T U Q_1(A).$$

Given that the support function of $\mathcal{Z}$ is the maximum eigenvalue function (see, for instance, [19, Section VI.5.1] and [18, Section 2.1]), the inequality above is equivalent to

$$r\lambda_1(Q_1(A)^T U Q_1(A)) \leq \operatorname{tr} Q_1(A)^T U Q_1(A).$$

Therefore $Q_1(A)^T U Q_1(A)$ is a homothety:

$$Q_1(A)^T U Q_1(A) - \frac{1}{r}\operatorname{tr}\,(Q_1(A)^T U Q_1(A)) I_r = 0;$$

this is the right-hand side of (4.3).

On the other hand, $\mathcal{V}(A)$ is the subspace parallel to $\operatorname{aff}(\partial\lambda_1(A))$ (see Definition 2.1 (i)). Then, relaxing the positivity constraint in (4.1), we see that $\mathcal{V}(A)$ is contained in the right-hand side of (4.4). The converse is straightforward:

$$Q_1(A)\{Z \in \mathcal{S}_r : \operatorname{tr} Z = 0\} Q_1(A)^T \subset \mathcal{U}(A)^\perp = \mathcal{V}(A). \qquad \square$$

4.2. Tangential parametrization of $\mathcal{M}_r$. Several parametrizations of the submanifold $\mathcal{M}_r$ can be found in recent independent works; for instance, a matrix exponential formulation is used in [38]. Here we follow the approach of [44] and complete it by defining a *tangential parametrization* (see section 3) of $\mathcal{M}_r$. This approach is quite natural and relies on perturbation theory applied to linear operators in finite-dimensional spaces (see [22, Chapter II]): let $\hat{A} \in \mathcal{M}_r$; thus by continuity, for any $A \in \mathcal{S}_n$ close enough to $\hat{A}$ the eigenvalues $(\lambda_1(A))_{i=1,\dots,r}$ remain close to $\lambda_1(\hat{A})$ and greater than $\lambda_{r+1}(\hat{A})$. In other words, in a neighborhood of $\hat{A}$, we have a separation between the first r eigenvalues and the others. For a matrix A in such a neighborhood, we can then define the following objects.

DEFINITION 4.4.

(i) *The set of r first eigenvalues of A is called the* $\lambda_1(\hat{A})$-group *at A.*

(ii) *The subspace spanned by the r first eigenvectors of A is called the* total eigenspace *for the* $\lambda_1(\hat{A})$-group *at $\hat{A}$; we denote it by $E_{tot}(A)$.*

(iii) *The value* $\hat{\lambda}(A) := \frac{1}{r}\sum_{i=1}^{r}\lambda_i(A)$ *is the* weighted mean *of the* $\lambda_1(\hat{A})$-group *at* A. □

One usually says that "eigenvectors are not continuous," which is true if we consider $E_1(A)$. Yet this difficulty can be overcome when dealing with the stable subspace $E_{tot}(A)$. In fact we can build a C^∞-map to track an orthonormal basis of $E_{tot}(A)$ in a neighborhood of $\hat{A}$. This is stated in the following theorem (see section 1.2 and section 3 for the notation).

THEOREM 4.5. *Take* $\hat{A} \in \mathcal{M}_r$ *and choose an orthonormal basis* $Q_1(\hat{A})$ *of* $E_1(\hat{A}) = E_{tot}(\hat{A})$. *Then there exist* $\delta > 0$ *and a map* $Q_{tot} : B(\hat{A}, \delta) \to \mathbb{R}^{n\times r}$ *such that*

(i) *for all* $A \in B(\hat{A}, \delta)$, *the columns of* $Q_{tot}(A)$ *form an orthonormal basis of* $E_{tot}(A)$ *and* $Q_{tot}(\hat{A}) = Q_1(\hat{A})$,

(ii) Q_{tot} *is* C^∞ *and, in particular,*

$$\mathrm{D}Q_{tot}(\hat{A}) \cdot H = (\lambda_1(\hat{A})I_n - \hat{A})^\dagger H Q_{tot}(\hat{A}) \text{ for all } H \in \mathcal{S}_n. \tag{4.5}$$

Proof. See [44, page 6]. □

We derive the following technical result.

COROLLARY 4.6. *The functions* $\Lambda_{tot} : B(\hat{A}, \delta) \ni A \mapsto Q_{tot}(A)^T A Q_{tot}(A)$ *and* $\hat{\lambda} : B(\hat{A}, \delta) \ni A \mapsto \frac{1}{r}\sum_{i=1}^{r}\lambda_i(A)$ *are* C^∞. *In particular, for* $A \in \mathcal{M}_r \cap B(\hat{A}, \delta)$ *and for all* $H \in \mathcal{S}_n$,

$$\mathrm{D}\Lambda_{tot}(A) \cdot H = Q_{tot}(A)^T H Q_{tot}(A) \tag{4.6}$$

and

$$\mathrm{D}\hat{\lambda}(A) \cdot H = \frac{1}{r}\mathrm{tr}\, Q_{tot}(A)^T H Q_{tot}(A). \tag{4.7}$$

Proof. The fact that Λ_{tot} is C^∞ is a consequence of Theorem 4.5 (i); $\hat{\lambda}$ is C^∞ as well, since $\hat{\lambda}(A) = \frac{1}{r}\mathrm{tr}\,\Lambda_{tot}(A)$. Now, for $A \in \mathcal{M}_r \cap B(\hat{A}, \delta)$ and for all $H \in \mathcal{S}_n$, we have

$$\mathrm{D}\Lambda_{tot}(A)\cdot H = Q_{tot}(A)^T H Q_{tot}(A) + [\mathrm{D}Q_{tot}(A)\cdot H]^T A Q_{tot}(A) + Q_{tot}(A)^T A[\mathrm{D}Q_{tot}(A)\cdot H].$$

From Theorem 4.5 (i), $A \in \mathcal{M}_r$ implies $E_{tot}(A) = E_1(A)$ and $AQ_{tot}(A) = \lambda_1(A)Q_{tot}(A)$. Then we obtain for all $A \in \mathcal{M}_r \cap B(\hat{A}, \delta)$,

$$\begin{aligned} \mathrm{D}\Lambda_{tot}(A) \cdot H - Q_{tot}(A)^T H Q_{tot}(A) &= \Delta^T A Q_{tot}(A) + Q_{tot}(A)^T A \Delta \\ &= \lambda_1(A)(\Delta^T Q_{tot}(A) + Q_{tot}(A)^T \Delta) \\ &= 0, \end{aligned}$$

where we have set $\Delta := \mathrm{D}Q_{tot}(A)\cdot H$ and used the normality of $Q_{tot}(A)$ (i.e., differentiate $Q_{tot}(A)^T Q_{tot}(A) = I_r$). Thus (4.6) is proved. The differential of $\hat{\lambda}$ is obtained by composition with the linear operator tr . □

Now let us introduce the subspace

$$\mathcal{H} := \{Z \in \mathcal{S}_r : \mathrm{tr}\, Z = 0\} \tag{4.8}$$

equipped with the induced Fröbenius product and consider the map

$$\phi : B(\hat{A}, \delta_0) \ni A \mapsto Q_{tot}(A)^T A Q_{tot}(A) - \frac{1}{r}\mathrm{tr}\,(Q_{tot}(A)^T A Q_{tot}(A))I_r \in \mathcal{H}. \tag{4.9}$$

The following theorem gives a local equation of $\mathcal{M}_r$ (see section 3 with $\mathcal{S} = \mathcal{S}_n$, $\mathcal{M} = \mathcal{M}_r$, and $\mathcal{T} = \mathcal{H}$).

THEOREM 4.7.

(i) *The map* ϕ *of* (4.9) *is* C^∞*; in particular, for all* $A \in \mathcal{M}_r \cap B(\hat{A}, \delta_0)$*, we have*

$$\mathrm{D}\phi(A) \cdot H = Q_{tot}(A)^T H Q_{tot}(A) - \frac{1}{r}\operatorname{tr}(Q_{tot}(A)^T H Q_{tot}(A)) I_r \tag{4.10}$$

for all $H \in \mathcal{S}_n$*.*

(ii) *The equation* $\phi(A) = 0$ *is a local equation (see Definition* 3.3) *of the submanifold* $\mathcal{M}_r$ *on* $B(\hat{A}, \delta_0)$*, and for all* $A \in \mathcal{M}_r \cap B(\hat{A}, \delta_0)$*, we have*

$$\mathrm{T}_{\mathcal{M}_r}(A) = \ker \mathrm{D}\phi(A).$$

Proof. (i) Since $\phi(A) = \Lambda_{tot}(A) - \hat{\lambda}(A) I_r$ for all $A \in \mathcal{M}_r \cap B(\hat{A}, \delta_0)$, apply Corollary 4.6 to prove.

(ii) For all $A \in \mathcal{M}_r \cap B(\hat{A}, \delta_0)$, clearly $\phi(A) = 0$. Conversely, if $A \in B(\hat{A}, \delta_0)$ and $\phi(A) = 0$, this means that $\Lambda_{tot}(A)$ is a homothety. In other words, the r first eigenvalues of A are equal to $\lambda_1(A)$ and, since the $\lambda_1(\hat{A})$-group (see Definition 4.4 (iii)) is well separated from the other eigenvalues of $A \in B(\hat{A}, \delta_0)$, the multiplicity of $\lambda_1(A)$ is exactly r. According to Definition 3.3, it remains to show that 0 is a regular value of ϕ, which means, via Definition 3.1, to show that $\mathrm{D}\phi(A)$ is surjective for all $A \in \mathcal{M}_r \cap B(\hat{A}, \delta_0)$. Let $Z \in \mathcal{H}$; then, using (4.10) we have $\mathrm{D}\phi(A) \cdot H = Z$, where $H = Q_{tot}(A) Z Q_{tot}(A)^T$, and we are done. □

We are now in a position to establish the first link between convex analysis and differential geometry.

COROLLARY 4.8. *At* $\hat{A} \in \mathcal{M}_r$*, the subspaces* $\mathcal{U}(\hat{A})$ *and* $\mathcal{V}(\hat{A})$ *of Definition* 2.1 *are, respectively, the tangent and normal spaces to the submanifold* $\mathcal{M}_r$ *at* $\hat{A}$*.*

Proof. By construction, $Q_{tot}(\hat{A}) = Q_1(\hat{A})$. Now, from Theorem 4.7 (ii), $\mathrm{T}_{\mathcal{M}_r}(\hat{A}) = \ker \mathrm{D}\phi(\hat{A})$, i.e., together with (4.10),

$$\mathrm{T}_{\mathcal{M}_r}(\hat{A}) = \left\{ H \in \mathcal{S}_n : Q_1(\hat{A})^T H Q_1(\hat{A}) - \frac{1}{r}\operatorname{tr}(Q_1(\hat{A})^T H Q_1(\hat{A})) I_r = 0 \right\},$$

which is exactly the right-hand side of (4.3). It follows that $\mathrm{T}_{\mathcal{M}_r}(\hat{A}) = \mathcal{U}(\hat{A})$ and $\mathrm{N}_{\mathcal{M}_r}(\hat{A}) = \mathcal{V}(\hat{A})$. □

From Theorem 4.7, we also deduce the following corollary.

COROLLARY 4.9. *There exists* $\delta > 0$ *and a unique* C^∞ *map*

$$V : \mathcal{U}(\hat{A}) \cap B(0, \delta) \to \mathcal{V}(\hat{A})$$

such that the map

$$\pi_{\hat{A}} : \mathcal{U}(\hat{A}) \cap B(0, \delta) \ni U \mapsto \hat{A} + U \oplus V(U) \tag{4.11}$$

is a tangential parametrization of the submanifold $\mathcal{M}_r$*.*

Proof. From Corollary 4.8, we know that the subspaces $\mathcal{U}(\hat{A})$ and $\mathcal{V}(\hat{A})$ are, respectively, the tangent and normal spaces to the submanifold $\mathcal{M}_r$ at $\hat{A}$. Now apply Theorem 3.4 to get the C^∞ map $V : B(0, \delta) \subset \mathcal{U}(\hat{A}) \to \mathcal{V}(\hat{A})$ and Corollary 3.5 to obtain the tangential parametrization $\pi_{\hat{A}}$. □

4.3. The $\boldsymbol{\mathcal{U}}$-Lagrangian of $\boldsymbol{\lambda_1}$. For $\hat{A} \in \mathcal{M}_r$, take $\hat{G} \in \operatorname{ri} \partial\lambda_1(\hat{A})$. Then, recalling (2.1) and (2.2), we define the $\mathcal{U}$-Lagrangian $L_{\mathcal{U}}(\hat{A}, \hat{G}; U)$ and the corresponding set of minimizers $W(U)$. We start with an easy result.

Proposition 4.10. *The convex function $L_{\mathcal{U}}(\hat{A},\hat{G};\cdot)$ is differentiable at $U=0$ and its gradient is given by*

$$\nabla L_{\mathcal{U}}(\hat{A},\hat{G};0)=\operatorname{proj}_{\mathcal{U}(\hat{A})}\hat{G}. \tag{4.12}$$

Proof. Rewrite Theorem 2.2 (i) in this matrix context. □

To go further, we need another strong link between convex analysis and differential geometry.

Theorem 4.11. *There exists $\eta>0$ such that for all $U\in B(0,\eta)\subset\mathcal{U}(\hat{A})$ the set $W(U)$ of* (2.2) *is a singleton:*

$$W(U)=\{V(U)\}\ \textit{ for all }\ U\in B(0,\eta)\,, \tag{4.13}$$

where $V(\cdot)$ is the map defined in Corollary 4.9.

Proof. Let $U\in\mathcal{U}(\hat{A})$, $V\in W(U)$, and $G\in\partial\lambda_1(\hat{A}+U\oplus V)\cap[\hat{G}+\mathcal{U}(\hat{A})]$. From (4.1), $G\in\partial\lambda_1(\hat{A}+U\oplus V)$ implies the *complementarity condition*

$$(\lambda_1(A+U\oplus V)I_n-[A+U\oplus V])G=0,$$

which in turn implies the *rank condition*

$$\operatorname{rank}(\lambda_1(A+U\oplus V)I_n-[A+U\oplus V])+\operatorname{rank}G\le n\,. \tag{4.14}$$

Furthermore, at $U=0$, $G=\hat{G}\in\operatorname{ri}\partial\lambda_1(\hat{A})$; then use (4.2) to see that the following *strict complementarity condition* holds (see Remark 6.6):

$$\operatorname{rank}(\lambda_1(\hat{A})I_n-\hat{A})=(n-r)\ \text{ and }\ \operatorname{rank}\hat{G}=r.$$

Then, by continuity of eigenvalues, together with Corollary 2.3 (iii) and Corollary 2.5, there exists $\eta>0$ such that

$$\begin{aligned}&\operatorname{rank}(\lambda_1(A+U\oplus V)I_n-[A+U\oplus V])\ge n-r\ \text{ and }\ \operatorname{rank}G\ge r\\ &\text{for all }\ U\in B(0,\eta)\ \text{ and all }\ (V,G)\in W(U)\times\partial\lambda_1(\hat{A}+U\oplus W(U))\cap[\hat{G}+\mathcal{U}(\hat{A})]\,.\end{aligned}$$

Together with inequality (4.14), we obtain

$$\begin{aligned}&\operatorname{rank}(\lambda_1(A+U\oplus V)I_n-[A+U\oplus V])= n-r\ \text{ and }\ \operatorname{rank}G= r\\ &\text{for all }\ U\in B(0,\eta)\ \text{ and all }\ (V,G)\in W(U)\times\partial\lambda_1(\hat{A}+U\oplus W(U))\cap[\hat{G}+\mathcal{U}(\hat{A})]\,.\end{aligned}$$

Then, taking η small enough, we have

$$A+U\oplus W(U)\subset B(\hat{A},\delta)\cap\mathcal{M}_r\,,$$

where δ is the radius introduced in Corollary 3.5. This enables us to apply Corollary 3.5 and to derive (4.13). □

From now on, we follow the path $\pi_{\hat{A}}(U)\in\mathcal{M}_r$ of (4.11). On the manifold $\mathcal{M}_r$ and close enough to $\hat{A}$, the first and total eigenspaces coincide. Hence, a natural choice for an orthonormal basis mapping in a neighborhood of $U=0$ is

$$\mathcal{U}(\hat{A})\ni U\to Q_1(\pi_{\hat{A}}(U)):=Q_{tot}(\pi_{\hat{A}}(U)).$$

This leads us to our main theoretical result.

Theorem 4.12. *There exists $\rho>0$ such that the $\mathcal{U}$-Lagrangian $L_{\mathcal{U}}(\hat{A},\hat{G};\cdot)$ is C^∞ on $B(0,\rho)\subset\mathcal{U}(\hat{A})$. In particular,*

(i) *the gradient at* $U \in B(0,\rho)$ *is*

$$\nabla L_{\mathcal{U}}(\hat{A},\hat{G};U) = \text{proj}_{\mathcal{U}(x)} Q_1(\pi_{\hat{A}}(U)) Z(U) Q_1(\pi_{\hat{A}}(U))^T\,, \tag{4.15}$$

where $Z(U)$ *is characterized by*

$$\begin{cases} Z(U) \in \{Z \in \mathcal{S}_r,\ \operatorname{tr} Z = 1\}\,, \\ Q_1(\pi_{\hat{A}}(U)) Z(U) Q_1(\pi_{\hat{A}}(U))^T - \hat{G} \in \mathcal{U}(\hat{A})\,; \end{cases} \tag{4.16}$$

(ii) *the Hessian at* $U = 0$ *is*

$$\nabla^2 L_{\mathcal{U}}(\hat{A},\hat{G};0) = \text{proj}_{\mathcal{U}(\hat{A})} \circ H(\hat{A},\hat{G}) \circ \text{proj}_{\mathcal{U}(\hat{A})}{}^*\,, \tag{4.17}$$

where $H(\hat{A},\hat{G})$ *is the symmetric operator defined by*

$$H(\hat{A},\hat{G}) \cdot Y = \hat{G} Y [\lambda_1(\hat{A}) I_n - \hat{A}]^\dagger + [\lambda_1(\hat{A}) I_n - \hat{A}]^\dagger Y \hat{G} \tag{4.18}$$

for all $Y \in \mathcal{S}_n$.

Proof. Because $\lambda_1(A) = \hat{\lambda}(A)$ for all $A \in \mathcal{M}_r$ close enough to $\hat{A}$, Theorem 4.11, together with (2.1), gives

$$L_{\mathcal{U}}(\hat{A},\hat{G};U) = \hat{\lambda}(\pi_{\hat{A}}(U)) - \langle \text{proj}_{\mathcal{V}(\hat{A})} \hat{G}, V(U) \rangle_{\mathcal{V}(\hat{A})} \quad \text{for all} \quad U \in B(0,\rho), \tag{4.19}$$

where $\rho := \min\{\eta_{\text{Th. 4.11}}, \delta_{\text{Cor 4.9}}\}$. Then $L_{\mathcal{U}}(\hat{A},\hat{G};\cdot)$ is C^∞ on $B(0,\rho)$.

(i) Now recall (4.1) and (2.3) to obtain

$$\begin{aligned} \partial L_{\mathcal{U}}(U) \quad = \quad & \text{proj}_{\mathcal{U}(\hat{A})}\{Q_1(\pi_{\hat{A}}(U)) Z Q_1(\pi_{\hat{A}}(U))^T : \\ & Q_1(\pi_{\hat{A}}(U)) Z Q_1(\pi_{\hat{A}}(U))^T - \hat{G} \in \mathcal{U}(\hat{A}),\ Z \in \mathcal{S}_r^+,\ \operatorname{tr} Z = 1\}\,. \end{aligned}$$

From Corollary 4.8, $\mathcal{U}(\hat{A}) = \mathrm{T}_{\mathcal{M}_r}(\hat{A})$, i.e., $\mathcal{U}(\hat{A}) = \ker \mathrm{D}\phi(\hat{A})$. Then the conditions $Q_1(\pi_{\hat{A}}(U)) Z Q_1(\pi_{\hat{A}}(U))^T - \hat{G} \in \mathcal{U}$ and $\operatorname{tr} Z = 1$ become

$$\mathrm{D}\phi(\hat{A}) \cdot (Q_1(\pi_{\hat{A}}(U)) Z Q_1(\pi_{\hat{A}}(U))^T - \hat{G}) = 0 \text{ and } \operatorname{tr} Z = 1\,. \tag{4.20}$$

Now let us consider the change of variable $Z = \frac{1}{r} I_r + \Omega$ with $\Omega \in \mathcal{H}$ of (4.8). Then, introducing

$$\mathrm{D}\phi(\pi_{\hat{A}}(U))^* : \mathcal{H} \ni \Omega \mapsto Q_1(\pi_{\hat{A}}(U)) \Omega Q_1(\pi_{\hat{A}}(U))^T \in \mathcal{S}_n,$$

we obtain together with (4.20)

$$\mathrm{D}\phi(\hat{A}) \circ \mathrm{D}\phi(\pi_{\hat{A}}(U))^* \cdot \Omega = \mathrm{D}\phi(\hat{A}) \cdot \hat{G} + \frac{1}{r} \mathrm{D}\phi(\hat{A}) \cdot (Q_1(\pi_{\hat{A}}(U)) Q_1(\pi_{\hat{A}}(U))^T). \tag{4.21}$$

Since $\mathrm{D}\phi(\hat{A})$ is surjective (Theorem 4.7 (ii)), $\mathrm{D}\phi(\hat{A}) \circ \mathrm{D}\phi(\hat{A})^*$ is invertible. By continuity, $\mathrm{D}\phi(\hat{A}) \circ \mathrm{D}\phi(\pi_{\hat{A}}(U))^*$ is also invertible for U small enough. Then invert (4.21) to obtain the solution $\Omega(U)$ and

$$\mathcal{U}(\hat{A}) \cap B(0,\rho) \ni U \mapsto Z(U) := \frac{1}{r} I_r + \Omega(U) \tag{4.22}$$

as C^∞-maps.

(ii) To calculate the second-order term, we need to differentiate (4.15) at $U = 0$; since we apply a fixed linear operator ($\mathrm{proj}_{\mathcal{U}(\hat{A})}$) to a product of three matrices, we obtain the sum of three terms. One of these terms is

$$\mathrm{proj}_{\mathcal{U}(\hat{A})} Q_1(\hat{A})[\mathrm{D}Z(0) \cdot H] Q_1(\hat{A})^T,$$

which is 0: indeed $\mathrm{D}Z(0) \cdot H = \mathrm{D}\Omega(0) \cdot H \in \mathcal{H}$ from (4.22) and therefore

$$Q_1(\hat{A})[\mathrm{D}Z(0) \cdot H] Q_1(\hat{A})^T \in \mathcal{V}(\hat{A}).$$

To calculate the other two terms (which are adjoint to each other), first note, together with (3.3), that

$$\mathrm{D}\pi_{\hat{A}}(0) \cdot H = H + \mathrm{D}V(0) \cdot H = H.$$

Using (4.5), we finally obtain (4.17). □

Then a second-order development of λ_1 along the manifold $\mathcal{M}_r$ can be derived.

COROLLARY 4.13. *Let* $D \in \mathcal{S}_n$ *be such that* $\hat{A} + D \in \mathcal{M}_r$ *and* $\|D\| \to 0$*; then*

$$\lambda_1(\hat{A}+D) = \lambda_1(\hat{A}) + \hat{G} \bullet D + \frac{1}{2} \mathrm{proj}_{\mathcal{U}(\hat{A})} D \bullet \nabla^2 L_{\mathcal{U}}(\hat{A}, \hat{G}; 0) \cdot (\mathrm{proj}_{\mathcal{U}(\hat{A})} D) + o(\|D\|^2).$$

Proof. For D small enough such that $\hat{A} + D \in \mathcal{M}_r$, apply Corollary 3.5 and set $U = \mathrm{proj}_{\mathcal{U}(\hat{A})} D$, $V = V(U) = \mathrm{proj}_{\mathcal{V}(\hat{A})} D$. To obtain the second-order development, apply Theorem 4.12:

$$\begin{aligned} L_{\mathcal{U}}(\hat{A}, \hat{G}; U) &= \lambda_1(\hat{A}) + \nabla L_{\mathcal{U}}(\hat{A}, \hat{G}; 0) \bullet U \\ &\quad + \tfrac{1}{2} U \bullet \nabla^2 L_{\mathcal{U}}(\hat{A}, \hat{G}; 0) \cdot U + o(\|U\|^2) \\ &= \lambda_1(\hat{A} + U \oplus V(U)) - \mathrm{proj}_{\mathcal{V}(\hat{A})} \hat{G} \bullet V(U). \end{aligned}$$

Finally, remember (3.3) (or (2.8)): $V = O(\|U\|^2) = O(\|D\|^2)$, and the proof is complete. □

Note here that, along the lines of [38] and [12], the operator $H(\hat{A}, \hat{G})$ has a precise geometrical interpretation: it is the *second covariant derivative* in the Euclidean metric of the function

$$\hat{\lambda_1}(M) := \frac{1}{r} \sum_{i=1}^{r} \lambda_i(M),$$

which is smooth near $\hat{A} \in \mathcal{M}_r$ and coincides with λ_1 on $\mathcal{M}_r$.

5. Composition with an affine operator. The function we want to minimize in problem (P) is in fact the convex function

$$f : \mathbb{R}^m \ni x \mapsto \lambda_1(A(x)),$$

where $A : \mathbb{R}^m \ni x \mapsto A(x) \in \mathcal{S}_n$ has the form of (1.1). In this section we obtain results for f similar to those in section 4 for λ_1. Yet we will see that, in order to obtain the existence of the $\mathcal{U}$-Hessian, we will need an additional assumption, which is not surprising after we realize that composing with operators amounts to intersecting submanifolds.

Let us start by recalling the following *chain rule*.

THEOREM 5.1. *Let $\hat{x} \in \mathbb{R}^m$; then*

$$\partial f(\hat{x}) = \mathcal{A}^* \partial \lambda_1(A(\hat{x})) \tag{5.1}$$

and

$$\operatorname{ri} \partial f(\hat{x}) = \mathcal{A}^* \operatorname{ri} \partial \lambda_1(A(\hat{x})). \tag{5.2}$$

Proof. Apply, for example, the chain rule given in [19, Theorem VI.4.2.1] to obtain (5.1) and the calculus rule [19, Proposition III.2.1.12] to obtain (5.2). □

Now we want to obtain chain rules for the subspaces $\mathcal{U}(\hat{x})$ and $\mathcal{V}(\hat{x})$ of Definition 2.1. To stress the dependence of these subspaces on f, we use the notation $\mathcal{U}_f(\hat{x})$ and $\mathcal{V}_f(\hat{x})$.

THEOREM 5.2. *Let $\hat{x} \in \mathbb{R}^m$; then*

$$\mathcal{V}_f(\hat{x}) = \mathcal{A}^* \mathcal{V}_{\lambda_1}(A(\hat{x})) \tag{5.3}$$

and

$$\mathcal{U}_f(\hat{x}) = \mathcal{A}^{-1} \mathcal{U}_{\lambda_1}(A(\hat{x})). \tag{5.4}$$

Proof. Taking the affine hull of the right- and left-hand sides in (5.1), we obtain, because $\mathcal{A}$ is linear,

$$\operatorname{aff} \partial f(\hat{x}) = \mathcal{A}^* \operatorname{aff} \partial \lambda_1(A(\hat{x})).$$

With Definition 2.1 (i), this gives (5.3). Write $\mathcal{U}_f(\hat{x}) = \mathcal{V}_f(\hat{x})^\perp$ and deduce

$$\mathcal{U}_f(\hat{x}) = \{u \in \mathbb{R}^m : \mathcal{A}(u) \in \mathcal{V}_{\lambda_1}(A(\hat{x}))^\perp\},$$

which is exactly (5.4). □

Now, take $\hat{g} \in \operatorname{ri} \partial f(\hat{x})$ and define the $\mathcal{U}$-Lagrangian of f at $(\hat{x}, \hat{g})$ according to (2.1); in what follows, we denote it by $L_{\mathcal{U},f}(x, g; \cdot)$. From Theorem 2.2, $L_{\mathcal{U},f}(\hat{x}, \hat{g}; \cdot)$ is differentiable at $u = 0$. We can prove the following composition rule.

THEOREM 5.3. *Let $\hat{G} \in \operatorname{ri} \partial \lambda_1(A(\hat{x}))$ be such that $\hat{g} = \mathcal{A}^* \cdot \hat{G}$. Then,*

$$\nabla L_{\mathcal{U},f}(\hat{x}, \hat{g}; 0) = [\operatorname{proj}_{\mathcal{U}_f(\hat{x})} \circ \mathcal{A}^* \circ \operatorname{proj}^*_{\mathcal{U}_{\lambda_1}(A(\hat{x}))}] \cdot \nabla L_{\mathcal{U},\lambda_1}(A(\hat{x}), \hat{G}; 0), \tag{5.5}$$

where $\mathcal{U}_f(\hat{x})$ is given by (5.4).

Proof. First the existence of $\hat{G}$ is assured by (5.2). This $\hat{G}$ is not unique, yet the composition rule we are going to prove does not depend on the choice of $\hat{G}$, given that all candidates have the same projection onto $\mathcal{U}_{\lambda_1}(A(\hat{x}))$. Using (2.4),

$$\begin{aligned}
\nabla L_{\mathcal{U},f}(\hat{x}, \hat{g}; 0) &= \operatorname{proj}_{\mathcal{U}_f(\hat{x})} \hat{g} \\
&= \operatorname{proj}_{\mathcal{U}_f(\hat{x})} (\mathcal{A}^* \cdot \hat{G}) \\
&= \operatorname{proj}_{\mathcal{U}_f(\hat{x})} \circ \mathcal{A}^* \cdot (\operatorname{proj}_{\mathcal{U}_{\lambda_1}(A(\hat{x}))} \hat{G} \oplus \operatorname{proj}_{\mathcal{V}_{\lambda_1}(A(\hat{x}))} \hat{G}) \\
&= \operatorname{proj}_{\mathcal{U}_f(\hat{x})} \circ \mathcal{A}^* \cdot (\operatorname{proj}_{\mathcal{U}_{\lambda_1}(A(\hat{x}))} \hat{G} \oplus 0),
\end{aligned}$$

since, from (5.3),

$$\mathcal{A}^* \cdot (0 \oplus \operatorname{proj}_{\mathcal{V}_{\lambda_1}(A(\hat{x}))} \hat{G}) \in \mathcal{V}_f(\hat{x}) = \mathcal{U}_f(\hat{x})^\perp.$$

Finally, from (4.12),

$$\operatorname{proj}_{\mathcal{U}_{\lambda_1}(A(\hat{x}))}\hat{G}\oplus 0=\operatorname{proj}^*_{\mathcal{U}_{\lambda_1}(A(\hat{x}))}\nabla L_{\mathcal{U},\lambda_1}(A(\hat{x}),\hat{G};0),$$

and (5.5) follows. □

Here as in section 4, we would like to identify a characteristic C^∞-manifold. A natural idea (in this composition framework) is to examine the set of vectors $x\in\mathbb{R}^m$ such that $\lambda_1(A(x))$ has a fixed multiplicity r, namely, to consider $A^{-1}(\mathcal{M}_r)$. The difficulty is that, even in the affine case, some catastrophe may appear. To ensure that $A^{-1}(\mathcal{M}_r)$ is a smooth manifold in a neighborhood of $\hat{x}$, we need to assume that $A(\cdot)$ is *transversal* to $\mathcal{M}_r$ at $\hat{x}$. In view of Corollary 4.8, Definition 3.6 becomes as follows.

DEFINITION 5.4. *We say that the* transversality condition (T) *holds at* $\hat{x}$ *if*

$$\operatorname{range}\mathcal{A}+\mathcal{U}_{\lambda_1}(A(\hat{x}))=\mathcal{S}_n. \tag{5.6}$$

This condition allows us to obtain a local equation of $\mathcal{W}_r:=A^{-1}\mathcal{M}_r$ via a simple composition rule.

THEOREM 5.5. *If* (T) *is satisfied at* $\hat{x}$*, then there exists* $\rho>0$ *such that* $\varphi(x)=0$*, where* $\varphi: B(\hat{x},\rho)\ni x\mapsto\phi(A(x))\in\mathcal{S}_r$ *and* ϕ *is given by* (4.9)*, is a local equation of* $\mathcal{W}_r\cap B(\hat{x},\rho)$*. Moreover, for all* $x\in B(\hat{x},\rho)$*, we have*

$$\mathrm{T}_{\mathcal{W}_r}(x)=\ker \mathrm{D}\varphi(x).$$

Proof. From Theorem 4.7, the map ϕ of (4.9) defines a local equation of $\mathcal{M}_r$. Then, from Theorem 3.8, there exists $\rho>0$ such that $\phi\circ A$ defines a local equation of $\mathcal{W}_r\cap B(\hat{x},\rho)$. Furthermore, from (3.1), we have that for all $x\in B(\hat{x},\rho)$, $\mathrm{T}_{\mathcal{W}_r}(x)=\ker\mathrm{D}\varphi(x)$. □

Now Corollaries 4.8 and 4.9 can be easily extended to the space of decision variables.

THEOREM 5.6. *Assume* (T) *is satisfied at* $\hat{x}$ *and take* $\hat{g}\in\operatorname{ri}\partial f(\hat{x})$*. Then*

(i) *the subspaces* $\mathcal{U}_f(\hat{x})$ *and* $\mathcal{V}_f(\hat{x})$ *are, respectively, the tangent and normal spaces to* $\mathcal{W}_r$ *at* $\hat{x}$*,*

(ii) *there exist* $\rho>0$ *and a* C^∞*-map* $v:\mathcal{U}_f(\hat{x})\cap B(0,\rho)\to\mathcal{V}_f(\hat{x})$ *such that the map*

$$p_{\hat{x}}:\mathcal{U}_f(\hat{x})\cap B(0,\rho)\ni u\mapsto\hat{x}+u\oplus v(u) \tag{5.7}$$

is a tangential parametrization of $\mathcal{W}_r$*.*

Proof. (i) From (3.6),

$$\mathrm{T}_{\mathcal{W}_r}(\hat{x})=\mathcal{A}^{-1}\mathrm{T}_{\mathcal{M}_r}(A(\hat{x})),$$

which is exactly the right-hand side of (5.4). Then $\mathcal{U}_f(\hat{x})=\mathrm{T}_{\mathcal{W}_r}(\hat{x})$ and $\mathcal{V}_f(\hat{x})=\mathrm{N}_{\mathcal{W}_r}(\hat{x})$.

(ii) As was done in the proof of Corollary 4.9, just apply Theorem 3.4 and Corollary 3.5. □

The following result is relevant when coming to algorithmic considerations (see section 6).

COROLLARY 5.7. *Assume* (T) *is satisfied at* $\hat{x}$*. Then, the map* $\mathcal{W}_r\ni x\mapsto\operatorname{proj}_{\mathcal{U}_f(x)}$ *is* C^∞ *in a neighborhood of* $\hat{x}$*.*

Proof. From Theorem 3.8, there exists $\rho > 0$ such that the transversality condition (T) holds in $\mathcal{W}_r \cap B(\hat{x}, \rho)$. Then, together with Theorems 5.6 (i) and 5.5, we have

$$\mathcal{U}_f(x) = \ker \mathrm{D}\varphi(x)$$

and then

$$\mathrm{proj}_{\mathcal{U}_f(x)} = \mathcal{I} - \mathrm{D}\varphi(x)^*[\mathrm{D}\varphi(x)\mathrm{D}\varphi(x)^*]^{-1}\mathrm{D}\varphi(x),$$

where $\mathcal{I}$ is the identity operator on $\mathbb{R}^m$. This proves that the map $\mathcal{W}_r \ni x \mapsto \mathrm{proj}_{\mathcal{U}_f(x)}$ is C^∞ on $B(\hat{x}, \rho) \cap \mathcal{W}_r$. □

To obtain the analogue of Theorem 4.11 in the space of decision variables, the following lemma will be useful.

LEMMA 5.8. *Assume (T) is satisfied at $\hat{x}$. Then,*

(i) *the set $[\mathcal{A}^*]^{-1}\hat{g} \cap \partial\lambda_1(A(\hat{x}))$ is a singleton,*

(ii) *the multifunction $\mathcal{U}_f(\hat{x}) \ni u \hookrightarrow \Gamma(u)$ defined by*

$$\Gamma(u) := \bigcup_{v \in w(u)} \left\{ \; [\mathcal{A}^*]^{-1}[\partial f(\hat{x} + u \oplus v) \cap (\hat{g} + \mathcal{U}(\hat{x}))] \cap \partial\lambda_1(A(\hat{x} + u \oplus v)) \; \right\}$$

is continuous at 0*:*

$$\lim_{u \to 0} \Gamma(u) = \{\hat{G}\},$$

where $\hat{G}$ is the unique element of $[\mathcal{A}^*]^{-1}\hat{g} \cap \partial\lambda_1(A(\hat{x}))$.

Proof. (i) Take two subgradients G and G' in $\partial\lambda_1(A(\hat{x}))$ such that $\mathcal{A}^* \cdot G = \mathcal{A}^* \cdot G'$; then observe in $\mathcal{S}_n$ that

$$\begin{aligned} G - G' \in \mathcal{V}_{\lambda_1}(A(\hat{x})) \cap \ker \mathcal{A}^* \quad &= \quad [\mathcal{U}_{\lambda_1}(A(\hat{x})) + \mathrm{range}\, \mathcal{A}]^\perp \\ &= \quad \{0\} \quad \text{(by transversality condition (5.6)).} \end{aligned}$$

(ii) Let us consider $G \in \lim\mathrm{ext}_{u \to 0} \Gamma(u)$: there exists a sequence $\{u_k, G_k\}_k$ (see, e.g., [19, A.5]) such that

$$G_k \in \Gamma(u_k)\,, u_k \to 0 \text{ and } G_k \to G \;\text{ when } k \to +\infty.$$

Hence, there exists a sequence $\{v_k\}_k$ such that for all k

$$v_k \in w(u_k)\,, \;\; G_k \in \partial\lambda_1(A(\hat{x} + u_k \oplus v_k)), \;\text{ and }\; \mathcal{A}^* \cdot G_k \in \partial f(\hat{x} + u_k \oplus v_k) \cap (\hat{g} + \mathcal{U}(\hat{x})).$$

By the continuity at 0 of the set-valued maps $w(\cdot)$ (Corollary 2.5) and of $\partial f(\hat{x} + \cdot \oplus w(\cdot)) \cap (\hat{g} + \mathcal{U}(\hat{x}))$ (Corollary 2.3) at 0 and the closedness of the graph of $\partial\lambda_1(\cdot)$ (Proposition VI.6.2.1 in [19]), we obtain

$$\lim_{k \to +\infty} v_k = 0\,, \;\; G \in \partial\lambda_1(A(\hat{x})), \;\text{ and }\; \mathcal{A}^* \cdot G \in \partial f(\hat{x}) \cap (\hat{g} + \mathcal{U}(\hat{x})).$$

The unicity implies $G = \hat{G}$ and then $\lim\mathrm{ext}_{u \to 0} \Gamma(u) = \{\hat{G}\} = \Gamma(0)$. This proves the outer semicontinuity of Γ at 0. Because $\Gamma(0)$ is a singleton, the continuity follows. □

We are now in a position to give the analogues of Theorem 4.11 in the space of decision variables $\mathbb{R}^m$.

Theorem 5.9. *Assume (T) is satisfied at $\hat{x}$ and take $\hat{g} \in \operatorname{ri} \partial f(\hat{x})$. Then there exists $\eta > 0$ such that for all $u \in B(0,\eta) \subset \mathcal{U}_f(\hat{x})$, the set $w(u)$ of* (2.2) *is a singleton:*

$$w(u) = \{v(u)\} \text{ for all } u \in B(0,\eta),$$

where $v(\cdot)$ is the C^∞-map defined in Theorem 5.6 (ii).

Proof. Let $u \in \mathcal{U}_f(\hat{x})$, $v \in w(u)$, and

$$G \in [\mathcal{A}^*]^{-1}[\partial f(\hat{x} + u \oplus v) \cap (\hat{g} + \mathcal{U}(\hat{x}))] \cap \partial\lambda_1(A(\hat{x} + u \oplus v)).$$

From Corollary 2.5 and Lemma 5.8 (ii), we show (as we did for Theorem 4.11) that the *strict complementarity condition* holds:

$$\begin{cases} (\lambda_1(A(\hat{x}+u\oplus v))I_n - A(\hat{x}+u\oplus v))G = 0\,, \\ \operatorname{rank}(\lambda_1(A(\hat{x}+u\oplus v))I_n - A(\hat{x}+u\oplus v)) = n-r \text{ and } \operatorname{rank} G = r \end{cases}$$
$$\text{for all } v \in w(u) \text{ and all } G \in [\mathcal{A}^*]^{-1}[\partial f(\hat{x}+u\oplus v) \cap \hat{g} + \mathcal{U}(\hat{x})] \cap \partial\lambda_1(A(\hat{x}+u\oplus v))$$

provided u is small enough. Hence, for all $u \in B(0,\eta)$ and η small enough, we have the inclusion

$$\hat{x} + u \oplus w(u) \subset \mathcal{W}_r \cap B(\hat{x},\delta),$$

where δ is the radius that was introduced in Corollary 3.5. We conclude using Corollary 3.5. □

There is no longer an obstacle to getting the desired second-order chain rule for $L_{\mathcal{U},f}$.

Theorem 5.10. *Assume (T) is satisfied at $\hat{x}$ and take $\hat{g} \in \operatorname{ri}\partial f(\hat{x})$. Then the $\mathcal{U}$-Lagrangian $L_{\mathcal{U},f}(\hat{x},\hat{g};\cdot)$ is C^∞. Moreover, at $u=0$,*

$$\nabla^2 L_{\mathcal{U},f}(\hat{x},\hat{g};0) = \operatorname{proj}_{\mathcal{U}_f(\hat{x})} \circ \mathcal{A}^* \circ H(A(\hat{x}),\hat{G}) \circ \mathcal{A} \circ \operatorname{proj}^*_{\mathcal{U}_f(\hat{x})}, \tag{5.8}$$

where $\hat{G}$ is the unique subgradient of $\partial\lambda_1(A(\hat{x}))$ such that $\hat{g} = \mathcal{A}^*\hat{G}$ and the operator $H(A(\hat{x}),\hat{G})$ is given by (4.18) *(with $\hat{A} = A(\hat{x})$). This can also be written*

$$\nabla^2 L_{\mathcal{U},f}(\hat{x},\hat{g};0) = B(\hat{x})^* \circ \nabla^2 L_{\mathcal{U},\lambda_1}(A(\hat{x}),\hat{G};0) \circ B(\hat{x}), \tag{5.9}$$

where $B(\hat{x}) = \operatorname{proj}_{\mathcal{U}_{\lambda_1}(A(\hat{x}))} \circ \mathcal{A} \circ \operatorname{proj}^*_{\mathcal{U}_f(\hat{x})}$ and $\mathcal{U}_f(\hat{x})$ is given by (5.4).

Proof. Similarly to the proof of Theorem 4.12, use Theorem 5.9 to get, in a neighborhood of $u = 0$,

$$L_{\mathcal{U},f}(\hat{x},\hat{g};u) = \hat{\lambda}(p_{\hat{x}}(u)) - \langle \hat{g}, v(u)\rangle_{\mathcal{V}_f(\hat{x})},$$

where $(p_{\hat{x}})$ is defined by (5.7). Then we use the transversality condition at $u = 0$, together with continuity arguments, to prove that the operator $\mathrm{D}\varphi(\hat{x}) \circ \mathrm{D}\varphi(p_{\hat{x}}(u))^*$ is invertible for u small enough. There exists a C^∞ map $\mathcal{U}_f(\hat{x}) \ni u \mapsto Z(u)$ defined in a neighborhood of $u = 0$ such that

$$\nabla L_{\mathcal{U},f}(\hat{x},\hat{g};u) = \operatorname{proj}_{\mathcal{U}_f(\hat{x})} \circ \mathcal{A}^* \cdot \big(\; Q_1(A(p_{\hat{x}}(u)))Z(u)Q_1(A(p_{\hat{x}}(u)))^T \;\big)\,. \tag{5.10}$$

To obtain the differential at $u = 0$ of the right-hand side of (5.10), we use exactly the same idea as in the proof of Theorem 4.12: for all $h \in \mathcal{U}_f(\hat{x})$,

$$\operatorname{proj}_{\mathcal{U}_f(\hat{x})} \circ \mathcal{A}^* \cdot \big(\; Q_1(A(p_{\hat{x}}(u)))[\mathrm{D}Z(0)\cdot h]Q_1(A(p_{\hat{x}}(u)))^T \;\big) = 0.$$

Then (5.8) follows similarly to (4.17).

Finally, the composition rule (5.9) is obtained after observing that (5.4) implies

$$\operatorname{range}(\mathcal{A} \circ \operatorname{proj}^*_{\mathcal{U}_f(\hat{x})}) = \operatorname{proj}^*_{\mathcal{U}_{\lambda_1}(A(\hat{x}))} \mathcal{U}_{\lambda_1}(A(\hat{x})).$$

Due to (4.17) the operator $H(A(\hat{x}), \hat{G})$ can be replaced in (5.8) by

$$\operatorname{proj}^*_{\mathcal{U}_{\lambda_1}(A(\hat{x}))} \circ \nabla^2 L_{\mathcal{U},\lambda_1}(A(\hat{x}), \hat{G}; 0) \circ \operatorname{proj}_{\mathcal{U}_{\lambda_1}(A(\hat{x}))},$$

and we are done. □

Finally, we give a second-order development of $f = \lambda_1 \circ A$ along the manifold $\mathcal{W}_r$.

COROLLARY 5.11. *With the assumptions of Theorem* 5.10*, we have for $\hat{x}+d \in \mathcal{W}_r$ and $d \to 0$ that*

$$f(\hat{x}+d) = f(\hat{x}) + \langle \hat{g}, d\rangle + \frac{1}{2}\langle \operatorname{proj}_{\mathcal{U}_f(\hat{x})} d, \nabla^2 L_{\mathcal{U},f}(\hat{x}, \hat{g}; 0) \cdot (\operatorname{proj}_{\mathcal{U}_f(\hat{x})} d)\rangle_{\mathcal{U}_f(\hat{x})} + o(\|d\|^2).$$

Proof. Use the same proof as for Corollary 4.13. □

6. Link with the SQP approach.

6.1. The Hessians. Let us consider a primal-dual pair $(\hat{A}, \hat{G}) \in \mathcal{S}_n \times \operatorname{ri} \partial\lambda_1(\hat{A})$. We have seen in Theorem 4.12 that the Hessian of the $\mathcal{U}$-Lagrangian $L_{\mathcal{U}}(\hat{A}, \hat{G}; \cdot)$ at $U = 0$ is the operator induced in $\mathcal{U}_{\lambda_1}(\hat{A})$ by $H(\hat{A}, \hat{G})$ of (4.18). Actually, more can be said about this operator.

THEOREM 6.1. *For all $\hat{A} \in \mathcal{M}_r$ and $G \in \partial\lambda_1(\hat{A})$, the symmetric operator $H(\hat{A}, \hat{G})$ defined by* (4.18) *satisfies the following properties:*

(i) $\mathcal{V}_{\lambda_1}(\hat{A}) \subset \ker H(\hat{A}, \hat{G})$*,*

(ii) $\operatorname{span} \partial\lambda_1(\hat{A}) \subset \ker H(\hat{A}, \hat{G})$*,*

(iii) $H(\hat{A}, \hat{G})$ *is positive semidefinite.*

Proof. (i) Consider a spectral decomposition of $\hat{A}$:

$$\hat{A} = \lambda_1(\hat{A}) Q_1(\hat{A}) Q_1(\hat{A})^T + Q_2 \Lambda_2 Q_2^T,$$

where $Q_1(\hat{A})$ is an orthonormal basis of $E_1(\hat{A})$, Q_2 is an orthonormal basis of $E_1(\hat{A})^\perp$, and $\Lambda_2 = \operatorname{diag}(\lambda_{r+1}(\hat{A}), \ldots, \lambda_n(\hat{A}))$. The Moore–Penrose inverse of $\lambda_1(\hat{A}) I_n - \hat{A}$ (see, e.g., [15, Section 5.5.4]) can then be written

$$[\lambda_1(\hat{A}) I_n - \hat{A}]^\dagger = Q_2 [\lambda_1(\hat{A}) I_{n-r} - \Lambda_2]^{-1} Q_2^T.$$

We have therefore

$$[\lambda_1(\hat{A}) I_n - \hat{A}]^\dagger Q_1(\hat{A}) = 0. \tag{6.1}$$

Take now $Y \in \mathcal{V}_{\lambda_1}(\hat{A})$; from (4.4), it has the form $Y = Q_1(\hat{A}) Z Q_1(\hat{A})^T$ for some $Z \in \mathcal{H}$ of (4.8). Use then (4.18) and (6.1) to obtain $H(\hat{A}, \hat{G}) \cdot Y = 0$ and conclude.

(ii) Use

$$\operatorname{span} \partial\lambda_1(\hat{A}) = Q_1(\hat{A}) \mathcal{S}_r Q_1(\hat{A})^T,$$

together with (4.18) and (6.1) again, to obtain the desired result.

(iii) The operator induced in $\mathcal{U}(\hat{A})$ by $H(\hat{A}, \hat{G})$ is the Hessian of a convex function (the $\mathcal{U}$-Lagrangian); then it is positive semidefinite. Together with (i) we obtain the positive semidefiniteness of $H(\hat{A}, \hat{G})$. □

COROLLARY 6.2. $\mathcal{A}^* \circ H(\hat{A}, \hat{G}) \circ \mathcal{A}$ *is positive semidefinite.*

Proof. This is straightforward from Theorem 6.1 (iii). □

It is then legitimate to ask the following question: Is $\mathcal{A}^* \circ H(\hat{A}, \hat{G}) \circ \mathcal{A}$ the Hessian of a convex function? The answer is yes provided the transversality condition holds: it is the Hessian introduced in SQP.

THEOREM 6.3. *Suppose* (T) *is satisfied at* $\hat{x}$*; take* $\hat{g} \in \operatorname{ri} \partial f(\hat{x})$ *and* $\hat{G} \in [\mathcal{A}^*]^{-1}\hat{g} \cap \partial\lambda_1(\hat{A})$*, where* $\hat{A} = A(\hat{x})$*. Then* $\mathcal{A}^* \circ H(\hat{A}, \hat{G}) \circ \mathcal{A}$ *coincides with the matrix* (3.11) *of* [44].

Proof. With the notation of Theorem 4.12, once $Q_1(\hat{A})$ is chosen there exists a unique $\hat{Z} = Z(0) \in \mathcal{S}_r^+$, $\operatorname{tr} \hat{Z} = 1$, such that $Q_1(\hat{A})\hat{Z}Q_1(\hat{A})^T = \hat{G}$. Then, taking Z as the new dual variable, we see that $H(\hat{A}, \hat{G})$ is the Hessian of the Lagrangian introduced in [44] at the primal-dual pair $(\hat{A}, \hat{Z})$. □

6.2. $\mathcal{U}$-Newton algorithm. The theory developed thus far strongly suggests the following algorithmic application: near a solution of (P), minimize the second-order development of the $\mathcal{U}$-Lagrangian of f. Here we present first a conceptual algorithm which relies on this simple idea. In section 6.3, we will show how it can lead to a more implementable scheme in a SQP context.

Let us consider a minimum point x^* and call r the multiplicity of $\lambda_1(A(x^*))$. Given $x \in B(x^*, \rho)$ for some $\rho > 0$, we need to compute some x_+ superlinearly closer to x^*. We consider the following conceptual algorithm.

ALGORITHM 6.4.

$\mathcal{V}$-step. Compute $\hat{x} \in \mathcal{W}_r$, a solution of

$$\min\{\|\hat{x} - x\| : \hat{x} \in \mathcal{W}_r\}. \tag{6.2}$$

Dual-step. Compute

$$g(\hat{x}) := \operatorname{proj}_{\partial f(\hat{x})}(0). \tag{6.3}$$

$\mathcal{U}$-step. Solve

$$\min_{u \in \mathcal{U}(\hat{x})} \langle u, \nabla L_{\mathcal{U},f}(\hat{x}, g(\hat{x}); 0)\rangle_{\mathcal{U}_f(\hat{x})} + \frac{1}{2}\langle u, \nabla^2 L_{\mathcal{U},f}(\hat{x}, g(\hat{x}); 0)u\rangle_{\mathcal{U}_f(\hat{x})}. \tag{6.4}$$

Update. Set $x_+ = \hat{x} + u$.

To make sure that x_+ is well defined, we introduce some additional conditions.

DEFINITION 6.5 (strict complementarity (SC)). *We say that the strict complementarity holds at* x^* *if* $0 \in \operatorname{ri} \partial f(x^*)$.

Remark 6.6. In view of Theorem 5.1, (SC) at x^* is equivalent to the existence of $G^* \in \operatorname{ri} \partial\lambda_1(A(x^*))$ such that $\mathcal{A}^*(G^*) = 0$. From Proposition 4.2, $G^* \in \mathcal{S}_n^+$ satisfies $[f(x^*)I_n - A(x^*)]G^* = 0$ and $\operatorname{rank} G^* = r$, which can be written

$$[f(x^*)I_n - A(x^*)]G^* = 0 \ \text{ and } \ [f(x^*)I_n - A(x^*)] + G^* \succ 0.$$

This is what is called the strict complementarity condition (see [4]) in semidefinite programming.

DEFINITION 6.7 (strict second-order condition (SSOC)). *We say that the strict second-order condition holds at* x^* *if* (T) *and* (SC) *are satisfied at* x^* *and the Hessian of* $L_{\mathcal{U},f}(x^*, 0; \cdot)$ *is positive definite at* $u = 0$.

Remark 6.8. The (SSOC) is natural in a Newton context. It plays a paramount role in the sensivity analysis of semidefinite programs; see the recent works [43] and [8].

The following proposition will be used as a preliminary result to get Theorem 6.10 and for a reformulation of the Dual-step of Algorithm 6.4 to get Algorithm 6.12.

PROPOSITION 6.9. *Let $\hat{x} \in \mathcal{W}_r \cap B(x^*, \rho)$. Then $g(\hat{x})$ of* (6.3) *satisfies the following.*

(i)

$$g(\hat{x}) = \mathcal{A}^* \cdot (Q_1(A(\hat{x})) Z Q_1(A(\hat{x}))^T),$$

where Z is a solution of

$$\min_{Z \in \mathcal{S}_r^+,\, \operatorname{tr} Z = 1} \|\mathcal{A}^* \cdot (Q_1(A(\hat{x})) Z Q_1(A(\hat{x}))^T)\|^2. \tag{6.5}$$

(ii) *If (T) holds at x^*, then the following minimization program*

$$\min_{Z \in \mathcal{S}_r,\, \operatorname{tr} Z = 1} \|\mathcal{A}^* \cdot (Q_1(A(\hat{x})) Z Q_1(A(\hat{x}))^T)\|^2 \tag{6.6}$$

has a unique solution $Z(\hat{x})$ for ρ small enough; moreover, the map

$$\mathcal{W}_r \cap B(x^*, \rho) \ni \hat{x} \mapsto Z(\hat{x}) \in \mathcal{S}_r$$

is C^∞.

(iii) *If (T) and* (SC) *hold, then $Z(\hat{x}) \succ 0$ for ρ small enough. Consequently $Z(\hat{x})$ is also the unique solution of* (6.5) *and*

$$g(\hat{x}) \in \operatorname{ri} \partial f(\hat{x})\,. \tag{6.7}$$

Proof. (i) Use (4.1) and (5.1) to rewrite the projection problem (6.3) in the form (6.5).

(ii) Write the optimality condition of (6.6) and use the transversality condition at x^* to obtain, via the implicit function theorem, the uniqueness and desired regularity.

(iii) Recall Definition 6.5: $0 \in \operatorname{ri} \partial f(x^*)$. This, together with (4.2) and (5.2), implies

$$\exists Z^* \in \mathcal{S}_r \text{ such that } \operatorname{tr} Z^* = 1,\ Z^* \succ 0, \text{ and } 0 = \mathcal{A}^* \cdot (Q_1(A(x^*)) Z^* Q_1(x^*)^T).$$

Furthermore, Z^* is a feasible point for (6.6) and $\|\mathcal{A}^* \cdot (Q_1(A(x^*)) Z^* Q_1(x^*)^T)\| = 0$. Hence, Z^* is optimal for (6.6) and $Z(x^*) = Z^* \succ 0$. Because the map $Z(\cdot)$ is continuous, we have $Z(\hat{x}) \succ 0$ for $\hat{x}$ in a neighborhood of x^*. Now, in this neighborhood, since $Z(\hat{x})$ is optimal for (6.6) and feasible for (6.5), it is optimal for (6.5). Finally, using again (4.2) and (5.2), we derive (6.7) easily. □

This enables us to show that Algorithm 6.4 is well defined when (SSOC) holds.

THEOREM 6.10. *Suppose* (SSOC) *holds at x^*. Then there exists $\rho > 0$ such that for all $x \in B(x^*, \rho)$ the point x_+ constructed by Algorithm* 6.4 *is well defined.*

Proof. Several ambiguities appear when following Algorithm 6.4.

First, we need to show that (6.2) has a nonempty solution set. Since (T) is satisfied at x^*, we can define from Theorem 5.5 a local equation $\varphi(x) = 0$ in $B(x^*, \rho)$ for some $\rho > 0$. Then, for ρ small enough,

$$\mathcal{W}_r \cap \bar{B}(x^*, \rho) = \varphi^{-1}(0) \cap \bar{B}(x^*, \rho)$$

is compact in $\mathbb{R}^m$. Then (6.2) has at least one solution.

Next, to define the $\mathcal{U}$-Lagrangian at the primal-dual pair $(\hat{x}, g(\hat{x}))$, we need to ensure that $g(\hat{x}) \in \operatorname{ri} \partial f(\hat{x})$ for ρ small enough. This is Proposition 6.9 (iii).

The last point consists of showing that the quadratic program (6.4) is well posed. For that purpose, it is sufficient to realize that

$$B(x^*, \rho) \cap \mathcal{W}_r \ni \hat{x} \mapsto \nabla^2 L_{\mathcal{U},f}(\hat{x}, g(\hat{x}); 0) = \operatorname{proj}_{\mathcal{U}_f(\hat{x})} \circ \mathcal{A}^* \circ H(A(\hat{x}), G(\hat{x})) \circ \mathcal{A} \circ \operatorname{proj}^*_{\mathcal{U}_f(\hat{x})}$$

with $G(\hat{x}) := Q_{tot}(A(\hat{x}))Z(\hat{x})Q_{tot}(A(\hat{x}))^T$ and $H(A(\hat{x}), G(\hat{x}))$ given by (4.18) is continuous: indeed the map $\hat{x} \mapsto H(A(\hat{x}), G(\hat{x}))$ is continuous in a neighborhood of x^*, as well as (see Corollary 5.7) the map $\hat{x} \mapsto \operatorname{proj}_{\mathcal{U}_f(\hat{x})}$. Therefore, $\nabla^2 L_{\mathcal{U},f}(\hat{x}, g(\hat{x}); 0)$ is positive definite for all $\hat{x} \in B(x^*, \rho) \cap \mathcal{W}_r$ and ρ small enough. □

6.3. Practical considerations. One practical difficulty of Algorithm 6.4 is the computation of $\hat{x}$. To overcome this, we proceed in two steps.

First observe the following.

PROPOSITION 6.11. *Assume* (SSOC) *holds at x^*. Then, for ρ small enough, the Dual-step, the $\mathcal{U}$-step, and the Update of Algorithm* 6.4 *are equivalent to the following steps.*

Dual-step. *Compute a solution $Z \in \mathcal{S}_r$ of* (6.6) *and set*

$$G(\hat{x}) := Q_1(A(\hat{x}))ZQ_1(A(\hat{x}))^T .$$

$\mathcal{U}$-step. *Compute a solution $d \in \mathbb{R}^m$ of*

$$\begin{aligned}&\min \mathcal{A}(d) \bullet G(\hat{x}) + \mathcal{A}(d) \bullet H(A(\hat{x}), G(\hat{x})) \cdot V(x) + \tfrac{1}{2}\mathcal{A}(d) \bullet H(A(\hat{x}), G(\hat{x})) \cdot \mathcal{A}(d)\\ &V(x) + \mathcal{A}(d) \in \mathcal{U}_{\lambda_1}(A(\hat{x})) ,\end{aligned}$$

where $V(x) := A(x) - A(\hat{x})$.

Update. *Set $x_+ = x + d$.*

Proof. The Dual-step is simply reformulated in the space of symmetric matrices: in view of Proposition 6.9 we have

$$g(\hat{x}) = \mathcal{A}^* \cdot G(\hat{x}) \text{ for } \hat{x} \text{ in a neighborhood of } x^*.$$

Then, apply the chain rules (5.5), (5.9), and (4.17) and make the change of variable $d := \hat{x} - x + u$ in (6.4) to obtain the desired $\mathcal{U}$-step and Update. □

Second, we consider the simplest approximation of $A(\hat{x})$ by setting the first r eigenvalues of $A(x)$ equal to $\lambda_1(A(x))$ without affecting eigenvectors. More formally, for $x \in B(x^*, \rho)$ and ρ small enough, let $Q_{tot}(A(x))$ be an orthonormal basis of $E_{tot}(A(x))$ and $\Lambda_{tot}(A(x)) = Q_{tot}(A(x))^T A(x) Q_{tot}(A(x))$; then we take as an approximation of $A(\hat{x})$ the following matrix $\hat{A}(x) \in \mathcal{M}_r$:

$$\hat{A}(x) = \lambda_1(A(x))Q_{tot}(A(x))Q_{tot}(A(x))^T + A(x) - Q_{tot}(A(x))\Lambda_{tot}(A(x))Q_{tot}(A(x))^T.$$

According to Theorem 6.1 (ii), this approximation satisfies

$$A(x) - \hat{A}(x) \in \operatorname{span} \partial \lambda_1(\hat{A}(x)) \subset \ker H(\hat{A}(x), \hat{G}(x)).$$

Then, in view of Proposition 6.11, we replace Algorithm 6.4 by the following approximation.

ALGORITHM 6.12. Let $x \in B(x^*, \rho)$.

$\mathcal{V}$-step. Compute $Q_{tot}(A(x))$, $\Lambda_{tot}(A(x))$, $\hat{A}(x)$, and

$$\hat{V}(x) := A(x) - \hat{A}(x).$$

Dual-step. Compute a solution $Z \in \mathcal{S}_r$ of

$$\min_{Z \in \mathcal{S}_r,\, \operatorname{tr} Z = 1} \|\mathcal{A}^* \cdot (Q_1(\hat{A}(x)) Z Q_1(\hat{A}(x))^T)\|^2,$$

and set

$$\hat{G}(x) := Q_1(\hat{A}(x)) Z Q_1(\hat{A}(x))^T.$$

$\mathcal{U}$-step. Compute the solution $d \in \mathbb{R}^m$ of

$$\begin{array}{l} \min \mathcal{A}(d) \bullet \hat{G}(x) + \frac{1}{2}\mathcal{A}(d) \bullet H(\hat{A}(x), \hat{G}(x)) \cdot \mathcal{A}(d) \\ \hat{V}(x) + \mathcal{A}(d) \in \mathcal{U}_{\lambda_1}(\hat{A}(x)). \end{array}$$

Update. Set $x_+ = x + d$.

Algorithm 6.12 is exactly the algorithm described in [44, Section 4] or, with some slight differences in the Hessian matrix $H(\hat{A}(x), \hat{G}(x))$, the one described in [38, Iteration 4]. It is a more implementable version of Algorithm 6.4 and enjoys the following property of quadratic rate of convergence.

THEOREM 6.13. *If* (SSOC) *holds at x^*, then there exists $\rho > 0$ and $C > 0$ such that for all $x \in B(x^*, \rho)$, x_+ defined by Algorithm* 6.12 *satisfies*

$$\|x_+ - x^*\| \leq C\|x - x^*\|^2.$$

Proof. See, e.g., [44, Section 6]. □

Acknowledgments. I wish to thank Claude Lemaréchal, my advisor, for the numerous and fruitful discussions we had together. I am also indebted to Claudia Sagastizábal, Jean-Charles Gilbert, and Laurent El Ghaoui for their careful reading and suggestions for improving the paper.

REFERENCES

[1] R. Abraham, J. E. Marsden, and T. Ratiu, *Manifolds, Tensor Analysis, and Applications*, Applied Mathematical Sciences 75, 2nd ed., Springer-Verlag, Berlin, 1988.

[2] F. Alizadeh, *Combinatorial Optimization with Interior Point Methods and Semi-Definite Matrices*, Ph.D. thesis, University of Minnesota, Minneapolis, MN, 1991.

[3] F. Alizadeh, *Optimization over the positive-definite cone: Interior-point methods and combinatorial applications*, in Advances in Optimization and Parallel Computing, Panos Pardalos, ed., North–Holland, Amsterdam, 1992.

[4] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Complementarity and nondegeneracy in semidefinite programming*, Math. Programming Ser. B, 77 (1997), pp. 111–128.

[5] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results*, SIAM J. Optim., 8 (1998), pp. 746–768.

[6] V. I. Arnold, *On matrices depending on parameters*, Russian Math. Surveys, 26 (1971), pp. 29–43.

[7] M. P. Bendsøe, A. Ben-Tal, and J. Zowe, *Optimization methods truss geometry and topology design*, Structural Optimization, 7 (1994), pp. 141–159.

[8] J. F. Bonnans, R. Cominetti, and A. Shapiro, *Sensivity Analysis of Optimization Problems Under Second Order Regular Constraints*, Technical Report 2989, INRIA, Le Chesnay, France, 1996.

[9] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, Studies in Applied Mathematics 15, SIAM, Philadelphia, PA, 1994.

[10] S. J. Cox and M. L. Overton, *The optimal design of columns against buckling*, SIAM J. Math. Anal., 23 (1992), pp. 287–325.

[11] J. Cullum, W. E. Donath, and P. Wolfe, *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices*, Math. Programming Study, 3 (1975), pp. 35–55.

[12] A. Edelman, T. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.

[13] M. K. H. Fan, *A quadratically convergent local algorithm on minimizing the largest eigenvalue of a symmetric matrix*, Linear Algebra Appl., (1993), pp. 231–253.

[14] R. Fletcher, *Semi-definite matrix constraints in optimization*, SIAM J. Control Optim., 23 (1985), pp. 493–522.

[15] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1989.

[16] C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM J. Optim., submitted.

[17] N. J. Hicks, *Notes on Differential Geometry*, Van Nostrand, Princeton, NJ, 1965.

[18] J.-B. Hiriart-Urruty and D. Ye, *Sensivity analysis of all eigenvalues of a symmetric matrix*, Numer. Math., 70 (1995), pp. 45–72.

[19] J. B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.

[20] F. Jarre, *An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices*, SIAM J. Control Optim., 31 (1993), pp. 1360–1377.

[21] T. Kato, *Perturbation theory of matrices*, J. Assoc. Comput. Mach., 5 (1958), p. 104.

[22] T. Kato, *Perturbation Theory for Linear Operators*, Springer-Verlag, New York, 1980.

[23] K. C. Kiwiel, *A linearization algorithm for optimizing control systems subject to singular value inequalities*, IEEE Trans. Automat. Control, AC-31 (1986), pp. 595–602.

[24] P. Lancaster, *On eigenvalues of matrices dependent on a parameter*, Numer. Math., 6 (1964), pp. 377–387.

[25] C. Lemaréchal, F. Oustry, and C. Sagastizábal, *The $\mathcal{U}$-Lagrangian of a convex function*, Trans. Amer. Math. Soc. (1996), to appear.

[26] A. S. Lewis and M. L. Overton, *Eigenvalue optimization*, Acta Numerica, 5 (1996), pp. 149–190.

[27] R. Mifflin and C. Sagastizábal, *$\mathcal{VU}$-derivatives for convex max-functions*, Math. Oper. Res. (1997), submitted.

[28] A. Nemirovsky, *On Normal Self-concordant Barriers and Long-step Interior Point Methods*, Technical report, Optimization Laboratory, Faculty of Industrial Engineering and Management, Technion, Israel Institute of Technology, Technion City, Haifa, Israel, 1997.

[29] A. Nemirovski and P. Gahinet, *The projective method for solving linear matrix inequalities*, Math. Programming Ser. B, 77 (1997), pp. 163–190.

[30] A. Nemirovsky and D. Yudin, *Problem Complexity and Method Efficiency in Optimization*, John Wiley, New York, 1983.

[31] Yu. Nesterov, *Interior-point methods: An old and new approach to nonlinear programming*, Math. Programming, 79 (1997), pp. 285–297.

[32] Yu. Nesterov and A. Nemirovsky, *A General Approach to Polynomial-time Algorithms Design for Convex Programming*, Technical report, Centr. Econ. & Math. Inst., USSR Academy of Sciences, Moscow, USSR, 1988.

[33] Yu. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Stud. Appl. Math. 13, SIAM, Philadelphia, PA, 1994.

[34] Yu. Nesterov and M. J. Todd, *Self-scaled cones and interior-points methods in nonlinear programming*, Math. Oper. Res., 22 (1997), pp. 1–42.

[35] F. Oustry, *A second-order bundle method to minimize the maximum eigenvalue function*, Math. Programming (1997), submitted.

[36] M. L. Overton, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 256–268.

[37] M. L. Overton, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.

[38] M. L. Overton and R. S. Womersley, *Second derivatives for optimizing eigenvalues of symmetric matrices*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 667–718.

[39] M. L. Overton and X. Ye, *Toward second-order methods for structured nonsmooth optimization*, in Advances in Optimization and Numerical Analysis, S. Gomez and J.-P. Hennart, eds., Kluwer Academic Publishers, Norwell, MA, 1994, pp. 97–109.

[40] E. Polak and Y. Wardi, *Nondifferentiable optimization algorithm for designing control systems having singular value inequalities*, Automatica, 18 (1982), pp. 267–283.

[41] F. A. Potra and R. Sheng, *A superlinearly convergent primal-dual infeasible-interior-point algorithm for semidefinite programming*, SIAM J. Optim. 8 (1998), pp. 1007–1028.

[42] H. Schramm and J. Zowe, *A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results*, SIAM J. Optim., 2 (1992), pp. 121–152.

[43] A. Shapiro, *First and second order analysis of nonlinear semidefinite programs*, Math. Programming Ser. B, 77 (1997), pp. 301–320.

[44] A. Shapiro and M. K. H. Fan, *On eigenvalue optimization*, SIAM J. Optim., 5 (1995), pp. 552–568.

[45] L. Vandenberghe and S. Boyd, *Primal-dual potential reduction method for problems involving matrix inequalities*, Math. Programming, Series B, 69 (1995), pp. 205–236.

SIAM J. OPTIM.
Vol. 9, No. 3, pp. 551–577

POLYNOMIAL CONVERGENCE OF A NEW FAMILY OF PRIMAL-DUAL ALGORITHMS FOR SEMIDEFINITE PROGRAMMING*

RENATO D. C. MONTEIRO[†] AND TAKASHI TSUCHIYA[‡]

Abstract. This paper establishes the polynomial convergence of a new class of primal-dual interior-point path-following feasible algorithms for semidefinite programming (SDP) whose search directions are obtained by applying Newton's method to the symmetric central path equation

$$(PXP^T)^{1/2}(P^{-T}SP^{-1})(PXP^T)^{1/2} - \mu I = 0,$$

where P is a nonsingular matrix. Specifically, we show that the short-step path-following algorithm based on the Frobenius norm neighborhood and the semilong-step path-following algorithm based on the operator 2-norm neighborhood have $O(\sqrt{n}L)$ and $O(nL)$ iteration-complexity bounds, respectively. When $P = I$, this yields the first polynomially convergent semilong-step algorithm based on a pure Newton direction. Restricting the scaling matrix P at each iteration to a certain subset of nonsingular matrices, we are able to establish an $O(n^{3/2}L)$ iteration complexity for the long-step path-following method. The resulting subclass of search directions contains both the Nesterov–Todd direction and the Helmberg–Rendl–Vanderbei–Wolkowicz/Kojima–Shindoh–Hara/Monteiro direction.

Key words. semidefinite programming, interior-point methods, polynomial complexity, path-following methods, primal-dual methods

AMS subject classifications. 65K05, 90C25, 90C30

PII. S1052623496312836

1. Introduction.

Several authors have discussed generalizations of interior-point algorithms for linear programming (LP) to the context of semidefinite programming (SDP). The landmark work in this direction is due to Nesterov and Nemirovskii [22, 23], where a general approach for using interior-point methods for solving convex programs is proposed, based on the notion of self-concordant functions. (See their book [25] for a comprehensive treatment of this subject.) They show that the problem of minimizing a linear function over a convex set can be solved in "polynomial time" as long as a self-concordant barrier function for the convex set is known. In particular, Nesterov and Nemirovskii show that linear programs, convex quadratic programs with convex quadratic constraints, and semidefinite programs all have explicit and easily computable self-concordant barrier functions, and hence can be solved in "polynomial time." On the other hand, Alizadeh [1] extends Ye's projective potential reduction algorithm [37] for LP to SDP and argues that many known interior-point algorithms for LP can also be transformed into algorithms for SDP in a mechanical way. Since then many authors have proposed interior-point algorithms for solving SDP problems, including Alizadeh, Haeberly, and Overton [2], Freund [3], Helmberg et al. [4],

*Received by the editors December 1, 1996; accepted for publication (in revised form) February 17, 1998; published electronically April 19, 1999.
http://www.siam.org/journals/siopt/9-3/31283.html

[†]School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332 (monteiro@isye.gatech.edu). The work of this author was based on research supported by the Office of Naval Research under grant N00014-94-1-0340 and the National Science Foundation under grants INT-9600343 and CCR-9700448.

[‡]The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-Ku, Tokyo, 106, Japan (tsuchiya@sun312.ism.ac.jp). This author's research was supported in part by the Grant-in-Aid for Encouragement of Young Scientists (A) 08780227 (1996) of the Ministry of Education, Science and Culture of Japan.

Jarre [5], Kojima, Shida, and Shindoh [10], Kojima, Shindoh, and Hara [11], Lin and Saigal [12], Luo, Sturm, and Zhang [13], Monteiro [15, 16], Monteiro and Tsuchiya [19], Monteiro and Zhang [21], Nesterov and Nemirovskii [24], Nesterov and Todd [26, 27], Potra and Sheng [29], Sturm and Zhang [30], Tseng [35], Vandenberghe and Boyd [36], and Zhang [38]. Most of these more recent works are concentrated on primal-dual methods.

The first algorithms for SDP which are extensions of well-known primal-dual LP algorithms, such as the long-step path-following algorithm of Kojima, Mizuno, and Yoshise [7] and Tanabe [31, 32], the short-step path-following algorithm of Kojima, Mizuno, and Yoshise [6] and Monteiro and Adler [17, 18], and the predictor-corrector algorithm of Mizuno, Todd, and Ye [14], use one of the following three search directions: (i) the Alizadeh–Haeberly–Overton (AHO) direction proposed in [2]; (ii) a direction independently proposed by Helmberg et al. [4] and Kojima, Shindoh, and Hara [11], and later rediscovered by Monteiro [15], which we refer to as the HRVW/KSH/M direction; and (iii) the Nesterov–Todd (NT) direction introduced in [26, 27]. Application of Newton's method to the central path equation $XS = \sigma\mu I$ results in an equation of the form

$$X\Delta S + \Delta X S = \sigma\mu I - XS, \tag{1.1}$$

which in general yields nonsymmetric directions. The AHO direction corresponds to the symmetric equation obtained by symmetrizing both sides of (1.1).

Another way of symmetrizing (1.1) is first to apply a similarity transformation $P(\cdot)P^{-1}$ to both sides of (1.1) and then to symmetrize it. Such an approach was first introduced by Monteiro [15] for the cases $P = X^{-1/2}$ and $P = S^{1/2}$. The resulting directions were found to be equivalent to two special directions of the KSH family of directions introduced earlier by Kojima, Shindoh, and Hara [11] using a different approach. The second direction (with $P = S^{1/2}$), which is the HRVW/KSH/M direction, was also proposed by Helmberg et al. [4] independently from [11]. (For simplicity, we refer to the first direction with $P = X^{-1/2}$ as the HRVW/KSH/M dual direction. We use the term HRVW/KSH/M directions to refer to both of them.) To unify the NT direction and the HRVW/KSH/M directions, Zhang [38] formally introduced the above scaling and symmetrization scheme for a general nonsingular scaling matrix P, which leads to a class of search directions parametrized by P, usually referred to as the Monteiro and Zhang (MZ) family. Subsequently, Todd, Toh, and Tütüncü [34] and Kojima, Shida, and Shindoh [8] showed that the NT direction is a member of the MZ family and the KSH family, respectively. In contrast, it is known that the AHO direction does not belong to the KSH family.

Unified convergence analyses for the MZ family have been given by Monteiro and Zhang [21] and Monteiro [16]. In the paper [21], iteration-complexity bounds are derived for the long-step primal-dual path-following method based on a subclass of the MZ family of search directions, which contains the NT and HRVW/KSH/M directions but not the AHO direction. In particular, it is shown that the corresponding algorithms based on the NT and the HRVW/KSH/M directions perform $\mathcal{O}(nL)$ and $\mathcal{O}(n^{3/2}L)$ iterations, respectively, to reduce the duality gap by a factor of at least $2^{-\mathcal{O}(L)}$. (The $\mathcal{O}(n^{3/2}L)$ iteration-complexity bound for the HRVW/KSH/M directions was in fact obtained earlier by Monteiro [15].) More recently, Monteiro [16] proves the polynomiality of the short-step primal-dual path-following algorithm and the Mizuno–Todd–Ye predictor-corrector–type algorithm based on any member of the MZ family, thus obtaining as a by-product the important result that Frobenius-norm–type algorithms based on the AHO direction are polynomially convergent.

Unified analyses for the KSH family of directions are provided in Kojima, Shindoh, and Hara [11] and Monteiro and Tsuchiya [19]. The paper [11] introduces the KSH family and establishes: (1) the polynomiality of the short-step path-following (feasible) method based on the two KSH/HRVW/M directions (both members of the KSH family); and (2) the polynomiality of a potential reduction (feasible and infeasible) algorithm based on *any* direction of the KSH family. Using techniques developed in Monteiro [16], the paper [19] extends the result (1) above to any direction of the KSH family. It also proves polynomial convergence of a Mizuno–Todd–Ye predictor-corrector-type algorithm for semidefinite linear complementarity problems based on the whole KSH family.

This paper considers primal-dual path-following methods for SDP based on the Newton direction for the symmetric central path equation

$$X^{1/2}SX^{1/2} - \mu I = 0. \tag{1.2}$$

This pure Newton direction is quite natural in view of the fact that the neighborhoods of the central path used to develop polynomially convergent algorithms are all based on the eigenvalues of the left-hand side of (1.2). (We use the qualifier "pure Newton" for those directions that are Newton directions with respect to a central path equation of the form $\Phi(X,S) = \mu I$, where the map $\Phi(\cdot,\cdot)$ is independent of the current iterate or any parameter.) In contrast, these neighborhoods have no connection with the eigenvalues of the left-hand side of the central path equation $XS + SX - \mu I = 0$ used to derive the AHO direction. Even though it is possible to define central path neighborhoods based on the eigenvalues of $XS + SX$, primal-dual path-following methods based on these neighborhoods are not known to be polynomially convergent. The polynomial convergence result obtained in [16] for the short-step path-following method using the AHO direction is based on the Frobenius norm neighborhood defined in terms of the left-hand side of (1.2).

We consider two primal-dual SDP algorithms based on the above Newton direction: (1) a short-step path-following method based on the Frobenius norm neighborhood; and (2) a semilong-step path-following method based on the operator norm neighborhood, which in terms of the eigenvalues of $X^{1/2}SX^{1/2}$ is equivalent to the infinity norm neighborhood for LP. We establish that algorithms (1) and (2) have iteration-complexity bounds of $\mathcal{O}(\sqrt{n}L)$ and $\mathcal{O}(nL)$, respectively, to reduce the duality gap by a factor of $2^{-\mathcal{O}(L)}$. It should be noted that nothing is known regarding the polynomial convergence of the semilong-step path-following algorithm using the AHO direction.

We also introduce a family of search directions which consists of the Newton directions applied to all the central path equations of the form

$$(PXP^T)^{1/2}(P^{-T}SP^{-1})(PXP^T)^{1/2} - \mu I = 0,$$

where P is a nonsingular matrix. We argue that this new family, referred to as the MT family, is related to the above Newton direction in the same way as the MZ family is related to the AHO direction, and we show that the iteration-complexity bounds of algorithms (1) and (2) above extend to any member of the MT family. Finally, we show that the long-step path-following method based on a subclass of the MT family, called the MT* subclass, has $O(n^{3/2}L)$ iteration-complexity bound, and hence does not depend on the choice of the sequence of scaling matrices $\{P^k\}$. In contrast, the iteration-complexity bound obtained in Monteiro and Zhang [21] for the long-step path-following algorithm based on the MZ* subclass of the MZ family depends on a certain condition number determined by the choice of $\{P^k\}$. Like the MZ* subclass, the MT* subclass also contains both the NT direction and the HRVW/KSH/M directions.

This paper is organized as follows. In section 2, we introduce the SDP problem and the associated assumptions and derive the Newton direction for the central path equation (1.2). We also give some existence results for this Newton direction and state a generic primal-dual algorithm based on it. In section 3, we state and prove technical results which are used in the polynomial convergence analysis of section 4. In section 4, we establish the polynomiality of the short-step and the semilong-step path-following algorithms based on the Newton direction for (1.2). In section 5, we introduce the MT family of search directions and generalize the convergence analysis of the short-step and semilong-step algorithms of section 4 to any member of this family. In section 6, we introduce the MT* subclass of directions and give the convergence analysis of the long-step path-following algorithm based on these directions. Finally, we end the paper with some concluding remarks in section 7.

1.1. Notation and terminology.

The following notation is used throughout the paper. The superscript ^{T} denotes transpose. $\Re^p$ denotes the p-dimensional Euclidean space. The set of all $p \times q$ matrices with real entries is denoted by $\Re^{p\times q}$. The set of all symmetric $p \times p$ matrices is denoted by $\mathcal{S}^p$. For $Q \in \mathcal{S}^p$, $Q \succeq 0$ means Q is positive semidefinite and $Q \succ 0$ means Q is positive definite. The trace of a matrix $Q \in \Re^{p\times p}$ is denoted by $\mathrm{Tr}\ Q \equiv \sum_{i=1}^n Q_{ii}$. For a matrix $Q \in \Re^{p\times p}$ with all real eigenvalues, we denote its eigenvalues by $\lambda_i[Q]$, $i = 1, \ldots, p$, and its largest and smallest eigenvalue by $\lambda_{\max}[Q]$ and $\lambda_{\min}[Q]$, respectively. Given P and Q in $\Re^{p\times q}$, the inner product between them in the vector space $\Re^{p\times q}$ is defined as $P \bullet Q \equiv \mathrm{Tr}\ P^TQ$. The Euclidean norm and its associated operator norm are both denoted by $\|\cdot\|$; hence, $\|Q\| \equiv \max_{\|u\|=1} \|Qu\|$ for any $Q \in \Re^{p\times p}$. The Frobenius norm of $Q \in \Re^{p\times p}$ is $\|Q\|_F \equiv (Q \bullet Q)^{1/2}$. $\mathcal{S}^p_+$ and $\mathcal{S}^p_{++}$ denote the set of all matrices in $\mathcal{S}^p$ which are positive semidefinite and positive definite, respectively.

2. The SDP problem and preliminary discussion.

In this section, we describe the SDP problem considered in this paper, state our assumptions, and derive the Newton direction for the central path equation (1.2). We also give some existence results for this Newton direction and state a generic primal-dual algorithm based on it.

2.1. The SDP problem.

This subsection describes the SDP problem and the corresponding assumptions. It also contains some notation and terminology that are used throughout our presentation.

We consider the SDP problem

$$(P) \qquad \min\{C \bullet X : A_i \bullet X = b_i,\ i = 1, \ldots, m,\ X \succeq 0\} \tag{2.1}$$

and its associated dual SDP problem

$$(D) \qquad \max\left\{b^Ty : \sum_{i=1}^m y_iA_i + S = C,\ S \succeq 0\right\}, \tag{2.2}$$

where $C \in \mathcal{S}^n$, $A_i \in \mathcal{S}^n$, $i = 1, \ldots, m$, and $b = (b_1, \ldots, b_m)^T \in \Re^m$ are the data, and $X \in \mathcal{S}^n_+$ and $(S, y) \in \mathcal{S}^n_+ \times \Re^m$ are the primal and dual variables, respectively.

The set of *interior feasible solutions* of (2.1) and (2.2) is

$$F^0(P) \equiv \{X \in \mathcal{S}^n : A_i \bullet X = b_i,\ i = 1, \ldots, m,\ X \succ 0\},$$
$$F^0(D) \equiv \left\{(S, y) \in \mathcal{S}^n \times \Re^m : \sum_{i=1}^m y_i A_i + S = C,\ S \succ 0\right\},$$

respectively. Throughout this paper, we assume that $F^0(P) \times F^0(D) \neq \emptyset$ and that the matrices A_i, $i = 1, \ldots, m$, are linearly independent. Under the first assumption, it is well known that both (2.1) and (2.2) have optimal solutions X^* and (S^*, y^*) such that $C \bullet X^* = b^T y^*$; i.e., the optimal values of (2.1) and (2.2) coincide. This last condition, called the strong duality, can be alternatively expressed as $X^* \bullet S^* = 0$ or $X^* S^* = 0$. Hence, the set of primal and dual optimal solutions consists of all the solutions $(X, S, y) \in \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^m$ to the following optimality system:

$$XS = 0, \tag{2.3a}$$
$$\sum_{i=1}^m y_i A_i + S - C = 0, \tag{2.3b}$$
$$A_i \bullet X - b_i = 0, \qquad i = 1, \ldots, m, \tag{2.3c}$$

where (2.3a) is called the complementarity equation. It is well known that for every $\nu > 0$, the perturbed system

$$XS = \nu I, \tag{2.4a}$$
$$\sum_{i=1}^m y_i A_i + S - C = 0, \tag{2.4b}$$
$$A_i \bullet X - b_i = 0, \qquad i = 1, \ldots, m, \tag{2.4c}$$

has a unique solution, denoted (X_ν, S_ν, y_ν), and that the limit $\lim_{\nu \to 0}(X_\nu, S_\nu, y_\nu)$ exists and is a solution of (2.3) (e.g., see Kojima, Shindoh, and Hara [11]). The set of all solutions (X_ν, S_ν, y_ν) with $\nu > 0$ is known as the *central path.*

It is known that for each $V \in \mathcal{S}^n_+$, there exists a unique $U \in \mathcal{S}^n_+$ such that $U^2 = V$. The matrix U is called the square root of V and is denoted by $V^{1/2}$. Using the square root $X^{1/2}$, (2.4a) can be alternatively expressed in the following symmetric form:

$$X^{1/2} S X^{1/2} = \nu I, \quad (X, S, y) \in \mathcal{F}^0(P) \times \mathcal{F}^0(D). \tag{2.5}$$

The path-following algorithms studied in this paper are all based on one of the following three centrality measures of a point $(X, S) \in \mathcal{S}^n_+ \times \mathcal{S}^n_+$:

$$d_F(X, S) \equiv \left\| X^{1/2} S X^{1/2} - \mu I \right\|_F = \left[\sum_{i=1}^n (\lambda_i[XS] - \mu)^2 \right]^{1/2},$$
$$d_\infty(X, S) \equiv \left\| X^{1/2} S X^{1/2} - \mu I \right\| = \max_{i=1,\ldots,n} |\lambda_i[XS] - \mu|,$$
$$d_{-\infty}(X, S) \equiv \left\| X^{1/2} S X^{1/2} - \mu I \right\|_{-\infty} = \max(0, \mu - \lambda_{\min}[XS]),$$

where $\mu \equiv (X \bullet S)/n = (\sum_{i=1}^n \lambda_i[XS])/n$, and $\|\cdot\|_{-\infty}$ is defined as

$$\|Q\|_{-\infty} \equiv \max(0, \lambda_{\max}[-Q]) \quad \text{for } Q \in \mathcal{S}^n.$$

Note that $\|\cdot\|_{-\infty}$ is a seminorm in the sense that it satisfies

$$\|\alpha Q\|_{-\infty} = \alpha\|Q\|_{-\infty}, \quad \|Q+R\|_{-\infty} \le \|Q\|_{-\infty} + \|R\|_{-\infty} \tag{2.6}$$

for every $Q, R \in \mathcal{S}^n$ and $\alpha > 0$. Clearly, we have

$$\|Q\|_{-\infty} \le \|Q\| \le \|Q\|_F, \tag{2.7}$$

and for $\gamma > 0$,

$$\lambda_{\min}[XS] \ge (1-\gamma)\mu \iff d_{-\infty}(X,S) \le \gamma\mu.$$

The short-step, semilong-step, and long-step path-following methods are based on the following central path neighborhoods, respectively:

$$\mathcal{N}_F(\gamma) \equiv \{(X,S,y) \in \mathcal{F}^0(P) \times \mathcal{F}^0(D) : d_F(X,S) \le \gamma\mu\}, \tag{2.8a}$$

$$\mathcal{N}_\infty(\gamma) \equiv \{(X,S,y) \in \mathcal{F}^0(P) \times \mathcal{F}^0(D) : d_\infty(X,S) \le \gamma\mu\}, \tag{2.8b}$$

$$\mathcal{N}_{-\infty}(\gamma) \equiv \{(X,S,y) \in \mathcal{F}^0(P) \times \mathcal{F}^0(D) : d_{-\infty}(X,S) \le \gamma\mu\}, \tag{2.8c}$$

where $\gamma > 0$ is a given constant.

2.2. The Newton direction and the generic algorithm. In this subsection, we derive the Newton direction for system (2.4b), (2.4c), and (2.5) and state a generic primal-dual method based on it. We end the subsection by giving some existence results for this Newton direction.

We start with the following technical result.

LEMMA 2.1. *For every $A \in \mathcal{S}^n_{++}$ and $H \in \mathcal{S}^n$, the equation*

$$AU + UA = H \tag{2.9}$$

has a unique solution $U \in \mathcal{S}^n$. Moreover, this solution satisfies

$$\|AU\|_F \le \|H\|_F/\sqrt{2}. \tag{2.10}$$

Proof. The first part of the lemma follows from the fact that the linear map $\Phi_A : \mathcal{S}^n \to \mathcal{S}^n$ defined by $\Phi_A(U) = AU + UA$ is an isomorphism. Indeed, since Φ_A has the same domain and codomain, it suffices to show that Φ_A is one-to-one, or equivalently that $AU + UA = 0$ implies $U = 0$. In turn, this last implication follows from the fact that any solution U of (2.9) satisfies (2.10) (simply set $H = 0$ in (2.10) to conclude that $U = 0$). To show the last claim, we square both sides of (2.9) to obtain

$$2\|AU\|_F^2 + 2\mathrm{Tr}\ [UAUA] = \|H\|_F^2.$$

Since Tr $[UAUA] = \|A^{1/2}UA^{1/2}\|_F^2 \ge 0$, (2.10) follows. □

Throughout this paper, we denote the unique solution U of (2.9) by $\langle\langle H \rangle\rangle_A$.

LEMMA 2.2. *Let $\theta : \mathcal{S}^n_{++} \to \mathcal{S}^n_{++}$ denote the square root function $\theta(X) = X^{1/2}$. Then, θ is an analytic function, and*

$$\theta'(X)H = \langle\langle H \rangle\rangle_{X^{1/2}} \quad \textit{for every } X \in \mathcal{S}^n_{++} \textit{ and } H \in \mathcal{S}^n,$$

where $\theta'(X)$ is the derivative of θ at X and $\theta'(X)H$ is the linear map $\theta'(X)$ evaluated at H.

Proof. Observe that the inverse function of θ is the analytic function given by $\theta^{-1}(A) = A^2$ for $A \in \mathcal{S}^n_{++}$. Clearly, the derivative $(\theta^{-1})'(A)$ of θ^{-1} is equal to the function Φ_A defined in the proof of Lemma 2.1. Since Φ_A is an isomorphism for every $A \in \mathcal{S}^n_{++}$, it follows from the inverse function theorem that θ is analytic and

$$\theta'(X) = \left[(\theta^{-1})'(X^{1/2})\right]^{-1} = \Phi^{-1}_{X^{1/2}}.$$

Hence, $\theta'(X)H = \Phi^{-1}_{X^{1/2}}(H) = \langle\langle H \rangle\rangle_{X^{1/2}}$. □

Using Lemma 2.2, it is now easy to see that the Newton direction $(\Delta X, \Delta S, \Delta y)$ for system (2.5) is the solution of the following system of linear equations:

(2.11a)
$$\langle\langle \Delta X \rangle\rangle_{X^{1/2}} S X^{1/2} + X^{1/2} S \langle\langle \Delta X \rangle\rangle_{X^{1/2}} + X^{1/2} \Delta S X^{1/2} = H,$$

(2.11b)
$$\sum_{i=1}^m \Delta y_i A_i + \Delta S = R,$$

(2.11c)
$$A_i \bullet \Delta X = r_i, \quad i = 1, \ldots, m,$$

where

(2.12a)
$$H \equiv \nu I - X^{1/2} S X^{1/2},$$

(2.12b)
$$R \equiv C - \sum_{i=1}^m y_i A_i - S,$$

(2.12c)
$$r_i \equiv b_i - A_i \bullet X, \quad i = 1, \ldots, m.$$

Let $U \equiv \langle\langle \Delta X \rangle\rangle_{X^{1/2}}$. Then, in terms of U, we can write (2.11a) as two equivalent equations:

(2.13)
$$U S X^{1/2} + X^{1/2} S U + X^{1/2} \Delta S X^{1/2} = \nu I - X^{1/2} S X^{1/2},$$

(2.14)
$$U X^{1/2} + X^{1/2} U = \Delta X.$$

We next state the generic primal-dual feasible algorithm that will be studied in this paper.

```
ALGORITHM I.
    Let $(X^0, S^0, y^0) \in \mathcal{F}^0(P) \times \mathcal{F}^0(D)$, $\mu_0 \equiv (X^0 \bullet S^0)/n$ and set $k = 0$.
    **Repeat until** $\mu_k \leq 2^{-L}\mu_0$ **do**
        (1) Let $(X, S, y) = (X^k, S^k, y^k)$ and $\mu \equiv (X \bullet S)/n$;
        (2) Choose a centrality parameter $\sigma = \sigma_k \in [0, 1]$;
        (3) Compute the solution $(\Delta X^k, \Delta S^k, \Delta y^k)$ of system (2.11) with
            $H \equiv \sigma\mu I - X^{1/2} S X^{1/2}$ and $(R, r) = (0, 0)$;
        (4) Choose a stepsize $\alpha_k > 0$ such that
            $(X^{k+1}, S^{k+1}, y^{k+1}) = (X^k, S^k, y^k) + \alpha_k(\Delta X^k, \Delta X^k, \Delta y^k) \in \mathcal{S}^n_{++}$;
        (5) Set $\mu_{k+1} \equiv (X^{k+1} \bullet S^{k+1})/n$ and increment $k$ by 1.
    **End**
```

The complete specification of Algorithm I depends on the choices of the initial point (X^0, S^0, y^0) and the sequences $\{\sigma_k\}$ and $\{\alpha_k\}$. These elements will be specified later when we discuss specific instances of the above algorithm. In general, the initial iterate (X^0, S^0, y^0) is chosen within one of the neighborhoods (2.8a)–(2.8b), and the sequences $\{\sigma_k\}$ and $\{\alpha_k\}$ are chosen so that the subsequent iterates lie in the same neighborhood and converge to an optimal solution of (2.1) and (2.2).

The following lemma establishes some important bounds on the Newton direction (2.11) and yields as a consequence Theorem 2.4, which establishes the nonsingularity of system (2.11) for any $(X, S, y) \in \mathcal{N}_\infty(\gamma)$ for $\gamma \in (0, 1/\sqrt{2})$.

LEMMA 2.3. *Suppose that $\gamma \in [0, 1/\sqrt{2})$ and that $(X, S, y) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \times \Re^m$ is such that $d_\infty(X, S) \le \gamma\mu$. If $(\Delta X, \Delta S, \Delta y)$ is a solution of* (2.11) *with* $(R, r) = (0, 0)$ *and $H \in \mathcal{S}^n$, then*

$$\max\left\{\mu \left\|X^{-1/2}\Delta X X^{-1/2}\right\|_F, \left\|X^{1/2}\Delta S X^{1/2}\right\|_F\right\} \le \frac{\|H\|_F}{(1-\sqrt{2}\gamma)}. \tag{2.15}$$

Proof. Multiplying (2.14) on the left and on the right by $X^{-1/2}$ and using inequality (2.10) of Lemma 2.1, we conclude that

$$\left\|UX^{-1/2}\right\|_F \le \frac{\|X^{-1/2}\Delta X X^{-1/2}\|_F}{\sqrt{2}}. \tag{2.16}$$

Since $(R, r) = (0, 0)$, it follows from (2.11b) and (2.11c) that

$$\Delta X \bullet \Delta S = 0. \tag{2.17}$$

By (2.11a) and (2.14), we have

$$\begin{aligned}\mu X^{-1/2}\Delta X X^{-1/2} + X^{1/2}\Delta S X^{1/2} = H &- UX^{-1/2}(X^{1/2}SX^{1/2} - \mu I)\\ &- (X^{1/2}SX^{1/2} - \mu I)X^{-1/2}U.\end{aligned}$$

Taking the Frobenius norm of both sides of this equality and using (2.16) and (2.17), we obtain

$$\begin{aligned}&\max\left\{\mu \left\|X^{-1/2}\Delta X X^{-1/2}\right\|_F, \left\|X^{1/2}\Delta S X^{1/2}\right\|_F\right\}\\ &\le \left(\mu^2 \left\|X^{-1/2}\Delta X X^{-1/2}\right\|_F^2 + \left\|X^{1/2}\Delta S X^{1/2}\right\|_F^2\right)^{1/2}\\ &= \left\|H - UX^{-1/2}(X^{1/2}SX^{1/2} - \mu I) - (X^{1/2}SX^{1/2} - \mu I)X^{-1/2}U\right\|_F\\ &\le \|H\|_F + 2\left\|UX^{-1/2}\right\|_F \left\|X^{1/2}SX^{1/2} - \mu I\right\|\\ &\le \|H\|_F + \sqrt{2}\gamma\mu \left\|X^{-1/2}\Delta X X^{-1/2}\right\|_F, \end{aligned} \tag{2.18}$$

which clearly implies that

$$\mu \left\|X^{-1/2}\Delta X X^{-1/2}\right\|_F \le \frac{\|H\|_F}{(1-\sqrt{2}\gamma)}.$$

Using this last inequality to bound the right-hand side of (2.18), we obtain (2.15). □

THEOREM 2.4. *If $(X, S, y) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \times \Re^m$ is such that $d_\infty(X, S) < \mu/\sqrt{2}$ then, for every $(H, R, r) \in \mathcal{S}^n \times \mathcal{S}^n \times \Re^m$, system* (2.11) *has a unique solution.*

Proof. In terms of $(\Delta X, \Delta S, \Delta y)$, the left-hand side of system (2.11) is a linear function from the space $\mathcal{S}^n \times \mathcal{S}^n \times \Re^m$ into itself. The lemma easily follows from the fact that this linear map is an isomorphism. To prove this fact, it is sufficient to show that this map is one-to-one, or equivalently that $(\Delta X, \Delta S, \Delta y) = (0,0,0)$ is the only solution of system (2.11) with $(H, R, r) = (0,0,0)$. Indeed, it follows from Lemma 2.3 that $(\Delta X, \Delta S) = (0,0)$. Using the linear independence of the matrices A_i, $i = 1, \ldots, m$, we conclude that $\Delta y = 0$. □

Note that the above result holds for both feasible and infeasible points. In particular, it implies the well-definedness of the Newton direction (2.11) for any point in $\mathcal{N}_\infty(\gamma)$, where $\gamma < 1/\sqrt{2}$. By slightly modifying Lemma 2.3, it is possible to establish the nonsingularity of system (2.11) for any point $(X, S, y) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \times \Re^m$ such that $\|X^{1/2}SX^{1/2} - \nu I\| \le \gamma\nu$ for some $\nu \in \Re$ and $\gamma < 1/\sqrt{2}$. This yields a larger region of points since ν is not constrained to be equal to μ.

3. Technical results.

In this section, we develop technical results which will be used in section 4 to establish the polynomial convergence of two specific instances of Algorithm I, namely, the short-step and the semilong-step path-following algorithms. The main novelty of the analysis of this paper is the use of second- or third-order Taylor expansions to analyze the behavior of the centrality measure when a Newton step is taken (see Lemma 3.3).

Let $(X, S, y) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \times \Re^m$ denote the current iterate and let $(\Delta X, \Delta S, \Delta y)$ denote the Newton direction for system (2.5) at the point (X, S, y), that is, the solution of (2.11) with $(R, r) = (0, 0)$ and $H = \sigma\mu I - X^{1/2}SX^{1/2}$, where $\mu \equiv (X \bullet S)/n$ and $\sigma \in [0, 1]$. Define

$$(X_\alpha, S_\alpha, y_\alpha) \equiv (X, S, y) + \alpha(\Delta X, \Delta S, \Delta y), \tag{3.1}$$

$$\mu(\alpha) \equiv \frac{X_\alpha \bullet S_\alpha}{n}, \tag{3.2}$$

$$\phi(\alpha) \equiv X_\alpha^{1/2} S_\alpha X_\alpha^{1/2} - \mu(\alpha) I. \tag{3.3}$$

Lemma 3.1. *We have*

$$\mu(\alpha) = (1 - \alpha + \alpha\sigma)\mu.$$

Proof. By (3.1) and the fact that $\Delta X \bullet \Delta S = 0$, we have

$$X_\alpha \bullet S_\alpha = X \bullet S + \alpha\,(S \bullet \Delta X + X \bullet \Delta S). \tag{3.4}$$

Using (2.13), (2.14), and the fact that $\nu = \sigma\mu$, we obtain

$$\begin{aligned} S \bullet \Delta X + X \bullet \Delta S &= \mathrm{Tr}[S\Delta X + X\Delta S] \\ &= \mathrm{Tr}[S(UX^{1/2} + X^{1/2}U) + X\Delta S] \\ &= \mathrm{Tr}[X^{1/2}SU + USX^{1/2} + X^{1/2}\Delta S X^{1/2}] \\ &= \mathrm{Tr}[\sigma\mu I - X^{1/2}SX^{1/2}] \\ &= n\sigma\mu - X \bullet S. \end{aligned} \tag{3.5}$$

The lemma now follows by substituting this equality into (3.4) and using the relations (3.2) and $X \bullet S = n\mu$. □

To study how the centrality measures for the points $(X_\alpha, S_\alpha, y_\alpha)$ vary, we will use either the second- or the third-order Taylor expansions of the function $\phi(\alpha)$. The following lemma gives expressions for the derivatives of this function.

LEMMA 3.2. *For every $\alpha \in \Re$ such that $(X_\alpha, S_\alpha) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++}$, we have*

$$\phi'(\alpha) = U^{(1)}_\alpha S_\alpha X^{1/2}_\alpha + X^{1/2}_\alpha S_\alpha U^{(1)}_\alpha + X^{1/2}_\alpha \Delta S X^{1/2}_\alpha + (1-\sigma)\mu I, \tag{3.6}$$

$$\begin{aligned}\phi''(\alpha) = {} & U^{(2)}_\alpha S_\alpha X^{1/2}_\alpha + X^{1/2}_\alpha S_\alpha U^{(2)}_\alpha + 2U^{(1)}_\alpha \Delta S X^{1/2}_\alpha \\ & + 2X^{1/2}_\alpha \Delta S U^{(1)}_\alpha + 2U^{(1)}_\alpha S_\alpha U^{(1)}_\alpha,\end{aligned} \tag{3.7}$$

$$\begin{aligned}\phi'''(\alpha) = {} & U^{(3)}_\alpha S_\alpha X^{1/2}_\alpha + X^{1/2}_\alpha S_\alpha U^{(3)}_\alpha + 3U^{(2)}_\alpha \Delta S X^{1/2}_\alpha + 3X^{1/2}_\alpha \Delta S U^{(2)}_\alpha \\ & + 3U^{(2)}_\alpha S_\alpha U^{(1)}_\alpha + 3U^{(1)}_\alpha S_\alpha U^{(2)}_\alpha + 6U^{(1)}_\alpha \Delta S U^{(1)}_\alpha,\end{aligned} \tag{3.8}$$

where

$$U^{(1)}_\alpha \equiv \frac{d}{d\alpha}[X^{1/2}_\alpha], \quad U^{(2)}_\alpha \equiv \frac{d^2}{d\alpha^2}[X^{1/2}_\alpha], \quad U^{(3)}_\alpha \equiv \frac{d^3}{d\alpha^3}[X^{1/2}_\alpha],$$

and $U^{(1)}_\alpha$, $U^{(2)}_\alpha$, and $U^{(3)}_\alpha$ satisfy

$$X^{1/2}_\alpha U^{(1)}_\alpha + U^{(1)}_\alpha X^{1/2}_\alpha = \Delta X, \tag{3.9}$$

$$X^{1/2}_\alpha U^{(2)}_\alpha + U^{(2)}_\alpha X^{1/2}_\alpha = -2U^{(1)}_\alpha U^{(1)}_\alpha, \tag{3.10}$$

$$X^{1/2}_\alpha U^{(3)}_\alpha + U^{(3)}_\alpha X^{1/2}_\alpha = -3\left(U^{(1)}_\alpha U^{(2)}_\alpha + U^{(2)}_\alpha U^{(1)}_\alpha\right). \tag{3.11}$$

Also,

$$U^{(1)}_0 = U. \tag{3.12}$$

Proof. Expressions (3.6), (3.8), and (3.8) follow immediately from (3.3) and Lemma 3.1. Observe that $X^{1/2}_\alpha = \theta(X + \alpha\Delta X)$, where θ is the function defined in Lemma 2.2. It follows from this lemma that $U^{(1)}_\alpha = \theta'(X_\alpha)\Delta X = \langle\langle \Delta X \rangle\rangle_{X^{1/2}_\alpha}$, or equivalently that (3.9) holds. Expressions (3.10) and (3.11) now follow by differentiating (3.9) once and twice, respectively. Since, by Lemma 2.1, U is uniquely determined by (2.14), it follows from (3.9) with $\alpha = 0$ that $U = U^{(1)}_0$. □

The analysis of this paper strongly relies on the following simple result.

LEMMA 3.3. *For every $\alpha \in [0,1]$, we have*

$$\|\phi(\alpha)\|_. \le (1-\alpha)\|\phi(0)\|_. + \frac{1}{2}\alpha^2 \sup_{\xi\in[0,\alpha]} \|\phi''(\xi)\|_F, \tag{3.13}$$

$$\|\phi(\alpha)\|_. \le (1-\alpha)\|\phi(0)\|_. + \frac{1}{2}\alpha^2 \|\phi''(0)\|_. + \frac{1}{6}\alpha^3 \sup_{\xi\in[0,\alpha]} \|\phi'''(\xi)\|_F, \tag{3.14}$$

where $\|\cdot\|_.$ represents one of the norms $\|\cdot\|_F$ or $\|\cdot\|$ or the seminorm $\|\cdot\|_{-\infty}$.

Proof. By (3.6) with $\alpha = 0$, (3.12), (2.13), and (3.3), we have

$$\begin{aligned}\phi'(0) &= USX^{1/2} + X^{1/2}SU + X^{1/2}\Delta S X^{1/2} + (1-\sigma)\mu I \\ &= \sigma\mu I - X^{1/2}SX^{1/2} + (1-\sigma)\mu I \;=\; -\phi(0).\end{aligned} \tag{3.15}$$

The lemma now follows from this last equality, relations (2.6) and (2.7), and the two higher-order Taylor integral formulae:

$$\phi(\alpha) = \phi(0) + \alpha\phi'(0) + \alpha^2 \int_0^1 (1-t)\phi''(t\alpha)dt,$$

$$\phi(\alpha) = \phi(0) + \alpha\phi'(0) + \frac{1}{2}\alpha^2\phi''(0) + \alpha^3 \int_0^1 \frac{(1-t)^2}{2}\phi'''(t\alpha)dt. \qquad □$$

The analysis of section 4 is based on the inequality (3.13). Hence, in the remaining part of the section, we derive bounds for the second derivative $\phi''(\alpha)$. The other inequality (3.14) will be used in the analysis of section 6 to establish the polynomiality of the long-step path-following method based on the new family of directions introduced in section 5.

To simplify notation, we let

$$\widehat{\Delta X} \equiv X^{-1/2}\Delta X X^{-1/2}, \quad \widehat{\Delta S} \equiv S^{-1/2}\Delta S S^{-1/2}, \tag{3.16}$$

$$D_\alpha^X \equiv X^{1/2}X_\alpha^{-1/2}, \quad D_\alpha^S \equiv S^{1/2}S_\alpha^{-1/2}. \tag{3.17}$$

Observe that D_α^X and D_α^S are only well defined when $X_\alpha \in \mathcal{S}_{++}^n$ and $S_\alpha \in \mathcal{S}_{++}^n$, respectively.

LEMMA 3.4. *Let $\tau \in (0,1)$ be given. The following statements hold:*

(a) *If $\alpha > 0$ is such that $\alpha\|\widehat{\Delta X}\| \le \tau$, then $X_\alpha \in \mathcal{S}_{++}^n$, D_α^X is well defined and*

$$\max\left\{\|D_\alpha^X\|, \|(D_\alpha^X)^{-1}\|\right\} \le \frac{1}{\sqrt{1-\tau}}. \tag{3.18}$$

(b) *If $\alpha > 0$ is such that $\alpha\|\widehat{\Delta S}\| \le \tau$, then $S_\alpha \in \mathcal{S}_{++}^n$, D_α^S is well defined and*

$$\max\left\{\|D_\alpha^S\|, \|(D_\alpha^S)^{-1}\|\right\} \le \frac{1}{\sqrt{1-\tau}}. \tag{3.19}$$

Proof. We prove only (a), since the proof of (b) is similar. Let $\alpha > 0$ satisfying $\alpha\|\widehat{\Delta X}\| \le \tau$ be given. Clearly, $I + \alpha\widehat{\Delta X} \in \mathcal{S}_{++}^n$, and hence $X_\alpha = X^{1/2}(I + \alpha\widehat{\Delta X})X^{1/2} \in \mathcal{S}_{++}^n$ and D_α^X is well defined. By (3.17), we have

$$\left[D_\alpha^X (D_\alpha^X)^T\right]^{-1} = X^{-1/2}X_\alpha X^{-1/2} = X^{-1/2}(X + \alpha\Delta X)X^{-1/2} = I + \alpha\widehat{\Delta X}.$$

Hence,

$$\|(D_\alpha^X)^{-1}\|^2 = \lambda_{\max}\left[I + \alpha\widehat{\Delta X}\right] \le 1 + \alpha\|\widehat{\Delta X}\| \le \frac{1}{1-\alpha\|\widehat{\Delta X}\|} \le \frac{1}{1-\tau}$$

and

$$\|D_\alpha^X\|^2 = \lambda_{\max}\left[\left(I + \alpha\widehat{\Delta X}\right)^{-1}\right] = \frac{1}{\lambda_{\min}\left[I + \alpha\widehat{\Delta X}\right]} \le \frac{1}{1-\alpha\|\widehat{\Delta X}\|} \le \frac{1}{1-\tau};$$

that is, (3.18) holds. □

LEMMA 3.5. *Let $\tau \in (0,1)$ be given. If $\alpha > 0$ is such that $\alpha \max\{\|\widehat{\Delta X}\|, \|\widehat{\Delta S}\|\} \le \tau$, then*

$$\left\|X_\alpha^{1/2}S_\alpha^{1/2}\right\| \le \frac{\|X^{1/2}S^{1/2}\|}{1-\tau}. \tag{3.20}$$

Proof. By (3.17), (3.18), and (3.19), we have

$$\begin{aligned}\left\|X_\alpha^{1/2}S_\alpha^{1/2}\right\| = \left\|(D_\alpha^X)^{-1}X^{1/2}S^{1/2}(D_\alpha^S)^{-T}\right\| &\le \|(D_\alpha^X)^{-1}\|\ \|(D_\alpha^S)^{-1}\|\ \left\|X^{1/2}S^{1/2}\right\| \\ &\le \frac{\|X^{1/2}S^{1/2}\|}{1-\tau}. \quad \square\end{aligned}$$

LEMMA 3.6. *Let $\tau \in (0,1)$ be given. If $\alpha > 0$ is such that $\alpha \|\widehat{\Delta X}\| \le \tau$, then*

$$\left\| U^{(1)}_\alpha X_\alpha^{-1/2} \right\|_F \le \frac{\|\widehat{\Delta X}\|_F}{\sqrt{2}(1-\tau)}, \tag{3.21}$$

$$\left\| U^{(2)}_\alpha X_\alpha^{-1/2} \right\|_F \le \frac{\|\widehat{\Delta X}\|_F^2}{\sqrt{2}(1-\tau)^2}, \tag{3.22}$$

$$\left\| U^{(3)}_\alpha X_\alpha^{-1/2} \right\|_F \le \frac{3}{\sqrt{2}} \frac{\|\widehat{\Delta X}\|_F^3}{(1-\tau)^3}. \tag{3.23}$$

Proof. Multiplying (3.9) on the left and on the right by $X_\alpha^{-1/2}$ and using inequality (2.10) of Lemma 2.1 and relation (3.18), we obtain (3.21) as follows:

$$\left\| U^{(1)}_\alpha X_\alpha^{-1/2} \right\|_F \le \frac{1}{\sqrt{2}} \left\| X_\alpha^{-1/2} \Delta X X_\alpha^{-1/2} \right\|_F = \frac{1}{\sqrt{2}} \left\| (D^X_\alpha)^T \widehat{\Delta X} D^X_\alpha \right\|_F \le \frac{\|\widehat{\Delta X}\|_F}{\sqrt{2}(1-\tau)}.$$

Multiplying (3.10) on the left and on the right by $X_\alpha^{-1/2}$ and using inequality (2.10) of Lemma 2.1 and relation (3.21), we obtain (3.22) as follows:

$$\left\| U^{(2)}_\alpha X_\alpha^{-1/2} \right\|_F \le \sqrt{2} \left\| X_\alpha^{-1/2} U^{(1)}_\alpha U^{(1)}_\alpha X_\alpha^{-1/2} \right\|_F \le \sqrt{2} \left\| (U^{(1)}_\alpha) X_\alpha^{-1/2} \right\|_F^2 \le \frac{\|\widehat{\Delta X}\|_F^2}{\sqrt{2}(1-\tau)^2}.$$

Finally, multiplying (3.11) on the left and on the right by $X_\alpha^{-1/2}$ and using inequality (2.10) of Lemma 2.1 and relations (3.21) and (3.22), we obtain (3.23) as follows:

$$\begin{aligned}\left\| U^{(3)}_\alpha X_\alpha^{-1/2} \right\|_F &\le \frac{3}{\sqrt{2}} \left\| X_\alpha^{-1/2} U^{(1)}_\alpha U^{(2)}_\alpha X_\alpha^{-1/2} + X_\alpha^{-1/2} U^{(2)}_\alpha U^{(1)}_\alpha X_\alpha^{-1/2} \right\|_F \\ &\le \frac{6}{\sqrt{2}} \left\| U^{(2)}_\alpha X_\alpha^{-1/2} \right\|_F \left\| U^{(1)}_\alpha X_\alpha^{-1/2} \right\|_F \le \frac{3}{\sqrt{2}} \frac{\|\widehat{\Delta X}\|_F^3}{(1-\tau)^3}. \qquad \square\end{aligned}$$

LEMMA 3.7. *Let constants $\tau \in (0,1)$ and $\gamma \in (0, 1/\sqrt{2})$ be given. Suppose that $(X,S,y) \in \mathcal{N}_\infty(\gamma)$ and that $\alpha > 0$ satisfies*

$$\alpha \max \left\{ \|\widehat{\Delta X}\|, \|\widehat{\Delta S}\| \right\} \le \tau.$$

Then,

$$\left\| U^{(1)}_\alpha \Delta S X_\alpha^{1/2} \right\|_F \le \frac{\|H\|_F^2}{\sqrt{2}(1-\tau)^2(1-\sqrt{2}\gamma)^2\mu}, \tag{3.24}$$

$$\left\| U^{(1)}_\alpha S_\alpha U^{(1)}_\alpha \right\|_F \le \frac{(1+\gamma)\|H\|_F^2}{2(1-\tau)^4(1-\sqrt{2}\gamma)^2\mu}, \tag{3.25}$$

$$\left\| U^{(2)}_\alpha S_\alpha X_\alpha^{1/2} \right\|_F \le \frac{(1+\gamma)\|H\|_F^2}{\sqrt{2}(1-\tau)^4(1-\sqrt{2}\gamma)^2\mu}. \tag{3.26}$$

Proof. Using Lemma 2.3, (3.18), and (3.21), we obtain

$$\begin{aligned}\left\| U^{(1)}_\alpha \Delta S X_\alpha^{1/2} \right\|_F &\le \left\| U^{(1)}_\alpha X_\alpha^{-1/2} \right\|_F \left\| X_\alpha^{1/2} \Delta S X_\alpha^{1/2} \right\| \\ &\le \left\| U^{(1)}_\alpha X_\alpha^{-1/2} \right\|_F \|(D^X_\alpha)^{-1}\|^2 \left\| X^{1/2} \Delta S X^{1/2} \right\| \\ &\le \frac{\|\widehat{\Delta X}\|_F \|H\|_F}{\sqrt{2}(1-\tau)^2(1-\sqrt{2}\gamma)} \le \frac{\|H\|_F^2}{\sqrt{2}(1-\tau)^2(1-\sqrt{2}\gamma)^2\mu}.\end{aligned}$$

In addition, using Lemma 2.3, relations (3.20), (3.21), and (3.22), and the fact that $\left\|X^{1/2}S^{1/2}\right\|^2 \le (1+\gamma)\mu$ whenever $(X,S,y) \in \mathcal{N}_\infty(\gamma)$, we obtain

$$\begin{aligned}\left\|U_\alpha^{(1)} S_\alpha U_\alpha^{(1)}\right\|_F \le \left\|U_\alpha^{(1)} X_\alpha^{-1/2}\right\|_F^2 \left\|X_\alpha^{1/2} S_\alpha^{1/2}\right\|^2 &\le \frac{\|\widehat{\Delta X}\|_F^2 \left\|X^{1/2}S^{1/2}\right\|^2}{2(1-\tau)^4} \\ &\le \frac{(1+\gamma)\|H\|_F^2}{2(1-\tau)^4(1-\sqrt{2}\gamma)^2\mu},\end{aligned}$$

and

$$\begin{aligned}\left\|U_\alpha^{(2)} S_\alpha X_\alpha^{1/2}\right\|_F \le \left\|U_\alpha^{(2)} X_\alpha^{-1/2}\right\|_F \left\|X_\alpha^{1/2} S_\alpha^{1/2}\right\|^2 &\le \frac{\|\widehat{\Delta X}\|_F^2 \left\|X^{1/2}S^{1/2}\right\|^2}{\sqrt{2}(1-\tau)^4} \\ &\le \frac{(1+\gamma)\|H\|_F^2}{\sqrt{2}(1-\tau)^4(1-\sqrt{2}\gamma)^2\mu}. \qquad \square\end{aligned}$$

The following result gives the desired bound on the second derivative $\phi''(\alpha)$.

LEMMA 3.8. *Let a constant* $\gamma \in (0, 1/\sqrt{2})$ *be given. Suppose that* $(X,S,y) \in \mathcal{N}_\infty(\gamma)$ *and* $\alpha > 0$ *is such that*

$$\alpha \max\left\{\|\widehat{\Delta X}\|, \|\widehat{\Delta S}\|\right\} \le \frac{1}{2}. \tag{3.27}$$

Then, $(X_\alpha, S_\alpha, y_\alpha) \in \mathcal{F}^0(P) \times \mathcal{F}^0(D)$ *and*

$$\|\phi''(\alpha)\|_F \le 80\frac{\|H\|_F^2}{(1-\sqrt{2}\gamma)^2\mu}.$$

Proof. It is easy to see that (3.27) and the fact that $(X,S,y) \in \mathcal{F}^0(P) \times \mathcal{F}^0(D)$ imply that $(X_\alpha, S_\alpha) \in \mathcal{F}^0(P) \times \mathcal{F}^0(D)$. It follows from (3.8) and Lemma 3.7 with $\tau = 1/2$ that

$$\begin{aligned}\|\phi''(\alpha)\|_F &\le 2\left\|U_\alpha^{(2)} S_\alpha X_\alpha^{1/2}\right\|_F + 2\left\|U_\alpha^{(1)} S_\alpha U_\alpha^{(1)}\right\|_F + 4\left\|U_\alpha^{(1)} \Delta S X_\alpha^{1/2}\right\|_F \\ &\le \left(16\sqrt{2}(1+\gamma) + 16(1+\gamma) + 8\sqrt{2}\right)\frac{\|H\|_F^2}{(1-\sqrt{2}\gamma)^2\mu} \\ &\le 80\frac{\|H\|_F^2}{(1-\sqrt{2}\gamma)^2\mu},\end{aligned}$$

where the last inequality follows from the fact that $\gamma < 1/\sqrt{2}$. $\square$

We end this section by stating without proof the following well-known result.

LEMMA 3.9. *The following statements hold:*

(a) *if* $(X,S,y) \in \mathcal{N}_F(\gamma)$, *then*

$$\|H\|_F \le \left[\gamma^2 + (1-\sigma)^2 n\right]^{1/2} \mu; \tag{3.28}$$

(b) *if* $(X,S,y) \in \mathcal{N}_\infty(\gamma)$, *then*

$$\|H\|_F \le \left[\gamma^2 + (1-\sigma)^2\right]^{1/2} \sqrt{n}\,\mu; \tag{3.29}$$

(c) *if* $(X,S,y) \in \mathcal{N}_{-\infty}(\gamma)$, *then*

$$\|\widehat{H}\|_F \le \left(1 - 2\sigma + \frac{\sigma^2}{1-\gamma}\right)^{1/2} \sqrt{n\mu}, \tag{3.30}$$

where $\widehat{H} \equiv H X^{-1/2} S^{-1/2}$.

4. Path-following algorithms based on the pure Newton direction. Based on the results developed in section 3, we now prove polynomiality of the short-step and the semilong-step path-following algorithms based on the pure Newton direction (2.11).

THEOREM 4.1. *Let $\gamma \in (0, 1/\sqrt{2})$ and $\delta \in (0,1)$ be constants satisfying*

$$\frac{40\,(\gamma^2+\delta^2)}{(1-\sqrt{2}\gamma)^2} \le \left(1-\frac{\delta}{\sqrt{n}}\right)\gamma. \tag{4.1}$$

Suppose that $(X,S,y) \in \mathcal{N}_F(\gamma)$ and let $(\Delta X, \Delta S, \Delta y)$ denote the solution of system (2.11) *with (H,R,r) given by* (2.12)*, $\nu \equiv \sigma\mu$, $\sigma \equiv 1-\delta/\sqrt{n}$, and $\mu \equiv (X \bullet S)/n$. Then,*

(a) $(X_1, S_1, y_1) \equiv (X+\Delta X, S+\Delta S, y+\Delta y) \in \mathcal{N}_F(\gamma)$*;*

(b) $X_1 \bullet S_1 = (1-\delta/\sqrt{n})(X \bullet S)$*.*

Proof. Statement (b) is an immediate consequence of Lemma 3.1 and the definition of σ. By Lemma 3.9(a) and the definition of σ, we have

$$\|H\|_F \le (\gamma^2+\delta^2)^{1/2}\mu. \tag{4.2}$$

Using Lemma 2.3, relations (4.1) and (4.2), and the fact that $\gamma < 1/\sqrt{2}$, we obtain

$$\begin{aligned}
\max\{\|\widehat{\Delta X}\|, \|\widehat{\Delta S}\|\} &\le \max\left\{\left\|X^{-1/2}\Delta X X^{-1/2}\right\|_F, \left\|X^{-1/2}S^{-1/2}\right\|^2 \left\|X^{1/2}\Delta S X^{1/2}\right\|_F\right\} \\
&\le \frac{1}{(1-\gamma)\mu}\max\left\{\mu\left\|X^{-1/2}\Delta X X^{-1/2}\right\|_F, \left\|X^{1/2}\Delta S X^{1/2}\right\|_F\right\} \\
&\le \frac{\|H\|_F}{(1-\gamma)(1-\sqrt{2}\gamma)\mu} \;\le\; \frac{(\gamma^2+\delta^2)^{1/2}}{(1-\gamma)(1-\sqrt{2}\gamma)} \\
&\le \frac{1}{1-\gamma}\left(\frac{\gamma}{40}\right)^{1/2} \;\le\; \frac{1}{2}.
\end{aligned}$$

Hence, it follows from Lemma 3.8 and relations (4.1) and (4.2) that $(X_1, S_1, y_1) \in \mathcal{F}^0(P) \times \mathcal{F}^0(D)$ and

$$\sup_{\xi\in[0,1]} \|\phi''(\xi)\|_F \le 80\frac{\gamma^2+\delta^2}{(1-\sqrt{2}\gamma)^2}\,\mu \le 2\gamma\left(1-\frac{\delta}{\sqrt{n}}\right)\mu.$$

This inequality together with relation (3.13) and Lemma 3.1 (both with $\alpha = 1$) imply that

$$\left\|X_1^{1/2}S_1X_1^{1/2} - \mu(1)I\right\|_F = \|\phi(1)\| \le \frac{1}{2}\sup_{\xi\in[0,1]}\|\phi''(\xi)\|_F \le \gamma\left(1-\frac{\delta}{\sqrt{n}}\right)\mu = \gamma\mu(1).$$

Hence, $(X_1, S_1, y_1) \in \mathcal{N}_F(\gamma)$. □

As an immediate consequence of Theorem 4.1, we have the following polynomial convergence result for the short-step path-following algorithm obtained from Algorithm I by letting $(X^0, S^0, y^0) \in \mathcal{N}_F(\gamma)$, $\sigma_k = 1-\delta/\sqrt{n}$, and $\alpha_k = 1$ for every $k \ge 0$.

COROLLARY 4.2 (polynomiality of short-step path-following algorithm). *Suppose that $\gamma \in (0, 1/\sqrt{2})$ and $\delta \in (0,1)$ are constants satisfying* (4.1)*. For Algorithm* I*, assume that $(X^0, S^0, y^0) \in \mathcal{N}_F(\gamma)$, $\sigma_k = 1-\delta/\sqrt{n}$, and $\alpha_k = 1$ for every $k \ge 0$. Then, every iterate (X^k, S^k, y^k) generated by Algorithm* I *is in the neighborhood $\mathcal{N}_F(\gamma)$ and* *satisfies* $X^k \bullet S^k = (1-\delta/\sqrt{n})^k (X^0 \bullet S^0)$. *Moreover, Algorithm* I *terminates in at most* $\mathcal{O}(\sqrt{n}L)$ *iterations.*

We now consider the semilong-step path-following algorithm based on the neighborhood $\mathcal{N}_\infty(\gamma)$. It is the special case of Algorithm I for which (X^0, S^0, y^0) is selected in $\mathcal{N}_\infty(\gamma)$, and the sequences $\{\sigma_k\}$ and $\{\alpha_k\}$ are defined as

$$\text{(4.3a)} \quad \sigma_k \equiv \bar{\sigma},$$

$$\text{(4.3b)} \quad \alpha_k \equiv \max\left\{\alpha \in [0,1] : (X^k, S^k, y^k) + \alpha(\Delta X^k, \Delta S^k, \Delta y^k) \in \mathcal{N}_\infty(\gamma)\right\}$$

for every $k \geq 0$, where $\bar{\sigma}$ is a prespecified constant in $(0,1)$.

THEOREM 4.3. *Suppose that* $(X,S,y) \in \mathcal{N}_\infty(\gamma)$ *for some given constant* $\gamma \in (0, 1/\sqrt{2})$, *and that* $(\Delta X, \Delta S, \Delta y)$ *denote the solution of* (2.11) *with* (H,R,r) *given by* (2.12), $\nu = \sigma\mu$, $\sigma \in (0,1)$, *and* $\mu \equiv (X \bullet S)/n$. *Let*

$$\tilde{\alpha} \equiv \frac{\sigma\gamma(1-\sqrt{2}\gamma)^2}{40n\left[\gamma^2 + (1-\sigma)^2\right]}. \tag{4.4}$$

Then for any $\alpha \in [0, \tilde{\alpha}]$, *we have*

(a) $(X_\alpha, S_\alpha, y_\alpha) \in \mathcal{N}_\infty(\gamma)$,

(b) $X_\alpha \bullet S_\alpha = (1-\alpha+\alpha\sigma)(X \bullet S)$.

Proof. Statement (b) is an immediate consequence of Lemma 3.1. Using Lemma 2.3, relations (3.29) and (4.4), and the fact that $\gamma < 1/\sqrt{2}$, we obtain

$$\begin{aligned}
\tilde{\alpha} \max\left\{\|\widehat{\Delta X}\|, \|\widehat{\Delta S}\|\right\} &\leq \tilde{\alpha} \max\left\{\left\|X^{-1/2}\Delta X X^{-1/2}\right\|_F,\right. \\
&\qquad\qquad \left.\left\|X^{-1/2}S^{-1/2}\right\|^2 \left\|X^{1/2}\Delta S X^{1/2}\right\|_F\right\} \\
&\leq \frac{\tilde{\alpha}}{(1-\gamma)\mu} \max\left\{\mu\left\|X^{-1/2}\Delta X X^{-1/2}\right\|_F, \left\|X^{1/2}\Delta S X^{1/2}\right\|_F\right\} \\
&\leq \frac{\tilde{\alpha}\,\|H\|_F}{(1-\gamma)(1-\sqrt{2}\gamma)\mu} \;\leq\; \frac{\tilde{\alpha}\left[\gamma^2+(1-\sigma)^2\right]^{1/2}\sqrt{n}}{(1-\gamma)(1-\sqrt{2}\gamma)} \\
&\leq \frac{\sigma\gamma(1-\sqrt{2}\gamma)}{40\sqrt{n}(1-\gamma)\left[\gamma^2+(1-\sigma)^2\right]^{1/2}} \;\leq\; \frac{\sigma}{40\sqrt{n}} \;\leq\; \frac{1}{2}.
\end{aligned}$$

Hence, it follows from Lemma 3.8 and Lemma 3.9(b) and relation (4.4) that $(X_\alpha, S_\alpha, y_\alpha) \in \mathcal{F}^0(P) \times \mathcal{F}^0(D)$ for any $\alpha \in [0, \tilde{\alpha}]$, and

$$\tilde{\alpha} \sup_{\xi \in [0,\tilde{\alpha}]} \|\phi''(\xi)\|_F \leq 80\tilde{\alpha}\frac{\gamma^2+(1-\sigma)^2}{(1-\sqrt{2}\gamma)^2}\, n\mu = 2\sigma\gamma\mu.$$

This inequality together with (3.14) and Lemma 3.1 imply that for every $\alpha \in [0, \tilde{\alpha}]$,

$$\begin{aligned}
\left\|X_\alpha^{1/2} S_\alpha X_\alpha^{1/2} - \mu(\alpha) I\right\| = \|\phi(\alpha)\| \;&\leq\; (1-\alpha)\|\phi(0)\| + \frac{1}{2}\alpha^2 \sup_{\xi\in[0,\alpha]} \|\phi''(\xi)\|_F \\
&\leq (1-\alpha)\gamma\mu + \frac{1}{2}\alpha\tilde{\alpha} \sup_{\xi\in[0,\tilde{\alpha}]} \|\phi''(\xi)\|_F \\
&\leq (1-\alpha)\gamma\mu + \alpha\sigma\gamma\mu \;=\; \gamma\mu(\alpha).
\end{aligned}$$

Hence, $(X_\alpha, S_\alpha, y_\alpha) \in \mathcal{N}_\infty(\gamma)$ for every $\alpha \in [0, \tilde{\alpha}]$; that is, (a) holds. $\square$

As an immediate consequence of Theorem 4.3, we have the following polynomial convergence result for the semilong-step path-following algorithm based on the pure Newton direction (2.11).

COROLLARY 4.4 (polynomiality of semilong-step path-following algorithm). *Let constants $\gamma \in (0, 1/\sqrt{2})$ and $\bar\sigma \in (0,1)$ be given. For Algorithm* I*, assume that $(X^0, S^0, y^0) \in \mathcal{N}_\infty(\gamma)$ and that the sequences $\{\sigma_k\}$ and $\{\alpha_k\}$ are chosen according to* (4.3)*. Then, the sequence of iterates $\{(X^k, S^k, y^k)\} \subset \mathcal{N}_\infty(\gamma)$ generated by Algorithm* I *satisfies $X^k \bullet S^k \le (1-\bar\eta)^k (X^0 \bullet S^0)$ for all $k \ge 0$, where*

$$\bar\eta \equiv \frac{\bar\sigma(1-\bar\sigma)\gamma(1-\sqrt{2}\gamma)^2}{40n\left[\gamma^2 + (1-\bar\sigma)^2\right]}.$$

Moreover, if the quantity $\max\{\gamma^{-1}, (1-\sqrt{2}\gamma)^{-1}, \bar\sigma^{-1}, (1-\bar\sigma)^{-1}\}$ *is independent of n, then the method terminates in at most $\mathcal{O}(nL)$ iterations.*

5. A family of "scaled" Newton directions.

In this section we introduce a new family of search directions which arises by computing the Newton direction (2.11) with respect to a scaled problem and mapping the direction back to the original space. Each direction of the family is then associated with the scaling matrix chosen to construct the scaled problem.

For the purpose of simplifying the notation in this and the next section, we assume that the variables for the original primal and dual problems are now $\widetilde{X}$ and $(\widetilde{S}, \widetilde{y})$ and that their associated data are $\widetilde{C} \in \mathcal{S}^n$, $\widetilde{A}_i \in \mathcal{S}^n$, $i = 1, \ldots, m$, and $\widetilde{b} = (\widetilde{b}_1, \ldots, \widetilde{b}_m) \in \Re^m$; that is, we assume that these problems are

$$(\widetilde{P}) \qquad \min\{\widetilde{C} \bullet \widetilde{X} : \widetilde{A}_i \bullet \widetilde{X} = \widetilde{b}_i,\ i = 1, \ldots, m,\ \widetilde{X} \succeq 0\},$$

$$(\widetilde{D}) \qquad \max\left\{\widetilde{b}^T\widetilde{y} : \sum_{i=1}^m \widetilde{y}_i \widetilde{A}_i + \widetilde{S} = \widetilde{C},\ \widetilde{S} \succeq 0\right\}.$$

Given a nonsingular matrix $\widetilde{P}$, consider the following change of variables:

$$X \equiv \widetilde{P}\widetilde{X}\widetilde{P}^T, \qquad (S, y) \equiv (\widetilde{P}^{-T}\widetilde{S}\widetilde{P}^{-1}, \widetilde{y}). \tag{5.1}$$

Letting

$$C \equiv \widetilde{P}^{-T}\widetilde{C}\widetilde{P}^{-1}, \qquad (A_i, b_i) \equiv (\widetilde{P}^{-T}\widetilde{A}_i\widetilde{P}^{-1}, \widetilde{b}_i) \ \text{ for } i = 1, \ldots, m,$$

problems $(\widetilde{P})$ and $(\widetilde{D})$ can be written in terms of these new variables as problems (P) and (D) of section 2. It can be easily verified that if (X, S, y) and $(\widetilde{X}, \widetilde{S}, \widetilde{y})$ in $\mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \times \Re^m$ are related according to (5.1), then $d_F(\widetilde{X}, \widetilde{S}) = d_F(X, S)$, $d_\infty(\widetilde{X}, \widetilde{S}) = d_\infty(X, S)$, $d_{-\infty}(\widetilde{X}, \widetilde{S}) = d_{-\infty}(X, S)$. Letting $\widetilde{\mathcal{N}}_F(\gamma)$, $\widetilde{\mathcal{N}}_\infty(\gamma)$, $\widetilde{\mathcal{N}}_{-\infty}(\gamma)$ denote the neighborhoods associated with the pair of problems $(\widetilde{P}, \widetilde{D})$, the above observation immediately implies that

$$(\widetilde{X}, \widetilde{S}, \widetilde{y}) \in \widetilde{\mathcal{N}}_F(\gamma) \iff (X, S, y) \in \mathcal{N}_F(\gamma), \tag{5.2a}$$

$$(\widetilde{X}, \widetilde{S}, \widetilde{y}) \in \widetilde{\mathcal{N}}_\infty(\gamma) \iff (X, S, y) \in \mathcal{N}_\infty(\gamma), \tag{5.2b}$$

$$(\widetilde{X}, \widetilde{S}, \widetilde{y}) \in \widetilde{\mathcal{N}}_{-\infty}(\gamma) \iff (X, S, y) \in \mathcal{N}_{-\infty}(\gamma). \tag{5.2c}$$

Moreover, if $(\widetilde{X}_\nu, \widetilde{S}_\nu, \widetilde{y}_\nu)$ denote the point on the central path with parameter $\nu > 0$ for the pair $(\widetilde{P}, \widetilde{D})$, then $(X_\nu, S_\nu, y_\nu) = (\widetilde{P}\widetilde{X}_\nu\widetilde{P}^T, \widetilde{P}^{-T}\widetilde{S}_\nu\widetilde{P}^{-1}, \widetilde{y}_\nu)$.

The matrix $\widetilde{P}$ also determines a scaled Newton direction (with parameter $\sigma > 0$) as follows. An interior feasible point $(\widetilde{X}, \widetilde{S}, \widetilde{y})$ for $(\widetilde{P}, \widetilde{D})$ determines an interior feasible point (X, S, y) for (P, D) as in (5.1). At the scaled point (X, S, y), the pure Newton direction (2.11) is computed and the resulting direction $(\Delta X, \Delta S, \Delta y)$ is mapped back into the original space to yield the scaled Newton direction $(\Delta\widetilde{X}, \Delta\widetilde{S}, \Delta\widetilde{y})$ as follows:

$$(\Delta\widetilde{X}, \Delta\widetilde{S}, \Delta\widetilde{y}) \equiv (\widetilde{P}^{-1}\Delta X\widetilde{P}^{-T}, \widetilde{P}^{T}\Delta S\widetilde{P}, \Delta y). \tag{5.3}$$

Hence, $(\Delta\widetilde{X}, \Delta\widetilde{S}, \Delta\widetilde{y})$ is a solution of

$$\begin{aligned} \nu I - X^{1/2}SX^{1/2} &= \langle\langle \widetilde{P}\Delta\widetilde{X}\widetilde{P}^{T}\rangle\rangle_{X^{1/2}} SX^{1/2} + X^{1/2}S\langle\langle \widetilde{P}\Delta\widetilde{X}\widetilde{P}^{T}\rangle\rangle_{X^{1/2}} \\ &\qquad + X^{1/2}\widetilde{P}^{-T}\Delta\widetilde{S}\widetilde{P}^{-1}X^{1/2}, \\ \widetilde{C} - \sum_{i=1}^{m} \widetilde{y}_i\widetilde{A}_i - \widetilde{S} &= \sum_{i=1}^{m} \Delta\widetilde{y}_i\widetilde{A}_i + \Delta\widetilde{S}, \\ \widetilde{b}_i - \widetilde{A}_i \bullet \widetilde{X} &= \widetilde{A}_i \bullet \Delta\widetilde{X}, \quad i = 1, \ldots, m, \end{aligned} \tag{5.4}$$

where $X \equiv \widetilde{P}\widetilde{X}\widetilde{P}^{T}$ and $S \equiv \widetilde{P}^{-T}\widetilde{S}\widetilde{P}^{-1}$.

Observe that the scaled Newton direction at the point $(\widetilde{X}, \widetilde{S}, \widetilde{y})$ depends on $\widetilde{P}$, and as $\widetilde{P}$ varies over the set of nonsingular matrices, we obtain a family of search directions, which we refer to as the MT family. Several observations are in order with respect to this family. It includes both the NT direction and the two HRVW/KSH/M directions. Indeed, if $\widetilde{P} = \widetilde{X}^{-1/2}$ then $X = I$, $S = \widetilde{X}^{1/2}\widetilde{S}\widetilde{X}^{1/2}$, $\langle\langle \widetilde{P}\Delta\widetilde{X}\widetilde{P}^{T}\rangle\rangle_{X^{1/2}} = (\widetilde{X}^{-1/2}\Delta\widetilde{X}\widetilde{X}^{-1/2})/2$, and the first equation of system (5.4) reduces to

$$\nu I - \widetilde{X}^{1/2}\widetilde{S}\widetilde{X}^{1/2} = \frac{1}{2}\left(\widetilde{X}^{-1/2}\Delta\widetilde{X}\widetilde{S}\widetilde{X}^{1/2} + \widetilde{X}^{1/2}\widetilde{S}\Delta\widetilde{X}\widetilde{X}^{-1/2}\right) + \widetilde{X}^{1/2}\Delta\widetilde{S}\widetilde{X}^{1/2},$$

which corresponds to the HRVW/KSH/M dual direction. If $\widetilde{P} = \widetilde{S}^{1/2}$, then $X = \widetilde{S}^{1/2}\widetilde{X}\widetilde{S}^{1/2}$, $S = I$,

$$\begin{aligned} &\langle\langle \widetilde{P}\Delta\widetilde{X}\widetilde{P}^{T}\rangle\rangle_{X^{1/2}} SX^{1/2} + X^{1/2}S\langle\langle \widetilde{P}\Delta\widetilde{X}\widetilde{P}^{T}\rangle\rangle_{X^{1/2}} \\ &\quad = \langle\langle \widetilde{S}^{1/2}\Delta\widetilde{X}\widetilde{S}^{1/2}\rangle\rangle_{X^{1/2}} X^{1/2} + X^{1/2}\langle\langle \widetilde{S}^{1/2}\Delta\widetilde{X}\widetilde{S}^{1/2}\rangle\rangle_{X^{1/2}} = \widetilde{S}^{1/2}\Delta\widetilde{X}\widetilde{S}^{1/2}, \end{aligned}$$

and the first equation of system (5.4) becomes

$$\nu I - \widetilde{S}^{1/2}\widetilde{X}\widetilde{S}^{1/2} = \widetilde{S}^{1/2}\Delta\widetilde{X}\widetilde{S}^{1/2} + (\widetilde{S}^{1/2}\widetilde{X}\widetilde{S}^{1/2})^{1/2}\widetilde{S}^{-1/2}\Delta\widetilde{S}\widetilde{S}^{-1/2}(\widetilde{S}^{1/2}\widetilde{X}\widetilde{S}^{1/2})^{1/2},$$

which is the equation corresponding to the NT direction. After the release of the first version of this paper, Todd [33] showed that the HRVW/KSH/M direction is also in the MT family and can be obtained by taking $\widetilde{P} = (\widetilde{S}\widetilde{X}\widetilde{S})^{1/2}$ so that $SXS = I$. Needless to say, we observe that if $\widetilde{P} = I$ then system (5.4) reduces to system (2.11), and hence it corresponds to the (pure) Newton direction considered in section 2.

Another possible choice is to take $\widetilde{P}$ to be the NT scaling matrix satisfying $\widetilde{P}\widetilde{X}\widetilde{P}^{T} = \widetilde{P}^{-T}\widetilde{S}\widetilde{P}^{-1}$, so that $X = S$ holds. Like the NT and HRVW/KSH/M directions, the resulting direction can be shown to have the scaling invariance property discussed in [34]. This direction is referred to as the MTW direction in [33].

The results obtained in section 2 for the pure Newton direction (2.11)–(2.12) can be extended to the whole MT family due to the fact that any member of this family reduces to the Newton direction (2.11)–(2.12) in the scaled space and the fact that the duality gap and the centrality measures remain invariant. In what follows, we summarize these results.

COROLLARY 5.1. *If* $(\widetilde{X}, \widetilde{S}, \widetilde{y}) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \times \Re^m$ *is such that* $d_\infty(\widetilde{X}, \widetilde{S}) < \widetilde{\mu}/\sqrt{2}$ *where* $\widetilde{\mu} \equiv (\widetilde{X} \bullet \widetilde{S})/n$*, then system* (5.4) *has a unique solution.*

Proof. Due to the invariance of the duality gap and the centrality measure $d_\infty(\cdot,\cdot)$, the assumption implies that $d_\infty(X, S) < \mu/\sqrt{2}$. Since the direction $(\Delta X, \Delta S, \Delta y) \equiv (\widetilde{P}\Delta\widetilde{X}\widetilde{P}^T, \widetilde{P}^{-T}\Delta\widetilde{S}\widetilde{P}^{-1}, \Delta\widetilde{y})$ is a solution of (2.11), the corollary follows immediately from Theorem 2.4. □

The generic primal-dual feasible algorithm based on the MT family of directions is stated next.

ALGORITHM II.

```
Let $(\widetilde{X}^0, \widetilde{S}^0, \widetilde{y}^0) \in \mathcal{F}^0(\widetilde{P}) \times \mathcal{F}^0(\widetilde{D})$, $\widetilde{\mu}_0 \equiv (\widetilde{X}^0 \bullet \widetilde{S}^0)/n$ and set $k = 0$.
**Repeat until** $\widetilde{\mu}_k \le 2^{-L}\widetilde{\mu}_0$, **do**
        (1) Let $(\widetilde{X}, \widetilde{S}, \widetilde{y}) = (\widetilde{X}^k, \widetilde{S}^k, \widetilde{y}^k)$ and $\widetilde{\mu} \equiv (\widetilde{X} \bullet \widetilde{S})/n$;
        (2) Choose a centrality parameter $\sigma = \sigma_k \in [0, 1]$ and a nonsingular
            matrix $\widetilde{P} = P^k$;
        (3) Compute the solution $(\Delta\widetilde{X}^k, \Delta\widetilde{S}^k, \Delta\widetilde{y}^k)$ of system (5.4) with
            $X \equiv \widetilde{P}\widetilde{X}\widetilde{P}^T$, $S \equiv \widetilde{P}^{-T}\widetilde{S}\widetilde{P}^{-1}$, and $\nu \equiv \sigma\widetilde{\mu}$;
        (4) Choose a stepsize $\alpha_k > 0$ such that
            $(\widetilde{X}^{k+1}, \widetilde{S}^{k+1}, \widetilde{y}^{k+1}) = (\widetilde{X}^k, \widetilde{S}^k, \widetilde{y}^k) + \alpha_k(\Delta\widetilde{X}^k, \Delta\widetilde{X}^k, \Delta\widetilde{y}^k) \in \mathcal{S}^n_{++}$;
        (5) Set $\widetilde{\mu}_{k+1} \equiv (\widetilde{X}^{k+1} \bullet \widetilde{S}^{k+1})/n$ and increment $k$ by 1.
**End**
```

The following two results follow immediately from Theorems 4.1 and 4.3, the equivalences in (5.2), and the invariance of the duality gap and the centrality measures.

COROLLARY 5.2 (polynomiality of short-step path-following algorithm for the MT family). *Suppose that* $\gamma \in (0, 1/\sqrt{2})$ *and* $\delta \in (0, 1)$ *are constants satisfying* (4.1). *For Algorithm* II, *assume that* $(\widetilde{X}^0, \widetilde{S}^0, \widetilde{y}^0) \in \widetilde{\mathcal{N}}_F(\gamma)$, $\sigma_k = 1 - \delta/\sqrt{n}$, *and* $\alpha_k = 1$ *for every* $k \ge 0$. *Then, every iterate* $(\widetilde{X}^k, \widetilde{S}^k, \widetilde{y}^k)$ *generated by Algorithm* II *is in the neighborhood* $\widetilde{\mathcal{N}}_F(\gamma)$ *and satisfies* $\widetilde{X}^k \bullet \widetilde{S}^k = (1 - \delta/\sqrt{n})^k(\widetilde{X}^0 \bullet \widetilde{S}^0)$. *Moreover, Algorithm* II *terminates in at most* $\mathcal{O}(\sqrt{n}L)$ *iterations.*

COROLLARY 5.3 (polynomiality of semilong-step path-following algorithm for the MT family). *Let constants* $\gamma \in (0, 1/\sqrt{2})$ *and* $\bar{\sigma} \in (0, 1)$ *be given. For Algorithm* II, *assume that* $(\widetilde{X}^0, \widetilde{S}^0, \widetilde{y}^0) \in \mathcal{N}_\infty(\gamma)$ *and that the sequences* $\{\sigma_k\}$ *and* $\{\alpha_k\}$ *are chosen according to*

$$\begin{aligned} \sigma_k &= \bar{\sigma}, \\ \alpha_k &= \max\left\{\alpha \in [0, 1] : (\widetilde{X}^k, \widetilde{S}^k, \widetilde{y}^k) + \alpha(\Delta\widetilde{X}^k, \Delta\widetilde{S}^k, \Delta\widetilde{y}^k) \in \widetilde{\mathcal{N}}_\infty(\gamma)\right\}. \end{aligned}$$

Then, the sequence of iterates $\{(\widetilde{X}^k, \widetilde{S}^k, \widetilde{y}^k)\} \subset \mathcal{N}_\infty(\gamma)$ *generated by Algorithm* II *satisfies* $\widetilde{X}^k \bullet \widetilde{S}^k \le (1 - \bar{\eta})^k(\widetilde{X}^0 \bullet \widetilde{S}^0)$ *for all* $k \ge 0$*, where*

$$\bar{\eta} \equiv \frac{\bar{\sigma}(1 - \bar{\sigma})\gamma(1 - \sqrt{2}\gamma)^2}{40n\left[\gamma^2 + (1 - \bar{\sigma})^2\right]}.$$

Moreover, if the quantity $\max\{\gamma^{-1}, (1 - \sqrt{2}\gamma)^{-1}, \bar{\sigma}^{-1}, (1 - \bar{\sigma})^{-1}\}$ *is independent of* n, *then the method terminates in at most* $\mathcal{O}(nL)$ *iterations.*

6. Long-step method based on a subclass of the MT family. In this section we consider a subclass of the MT family whose members are well defined at every point $(\widetilde{X}, \widetilde{S}, \widetilde{y}) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \times \Re^m$. Moreover, we establish an $\mathcal{O}(n^{3/2}L)$ iteration-complexity bound for a long-step path-following feasible algorithm based on this subclass of the MT family. The analysis of this section is based on the third-order derivative inequality (3.14) and hence is more involved than the one presented in sections 3 and 4. It is possible to derive polynomial convergence for the long-step path-following algorithm using second-order derivative inequality (3.13), but the iteration-complexity bound obtained is worse than the $\mathcal{O}(n^{3/2}L)$ bound obtained using (3.14).

We first describe the subclass of the MT family, which we refer to as the MT* family. The members of the MT* family at a point $(\widetilde{X}, \widetilde{S}, \widetilde{y}) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \times \Re^m$ consists of all the members of the MT family corresponding to those scaling matrices $\widetilde{P}$ satisfying

$$X^{1/2}S + SX^{1/2} = (\widetilde{P}\widetilde{X}\widetilde{P}^T)^{1/2}(\widetilde{P}^{-T}\widetilde{S}\widetilde{P}^{-1}) + (\widetilde{P}^{-T}\widetilde{S}\widetilde{P}^{-1})(\widetilde{P}\widetilde{X}\widetilde{P}^T)^{1/2} \succ 0. \tag{6.1}$$

The next two results imply that any member of the MT* family is well defined for any point $(\widetilde{X}, \widetilde{S}, \widetilde{y}) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \times \Re^m$.

LEMMA 6.1. *Suppose that $(X, S, y) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \times \Re^m$ is such that $X^{1/2}S + SX^{1/2} \succ 0$. If $(\Delta X, \Delta S, \Delta y)$ is a solution of system* (2.11) *with $(R, r) = (0, 0)$ and $H \in \mathcal{S}^n$, then*

$$\left\| S^{1/2}U \right\|_F \leq \|\widehat{H}\|_F, \tag{6.2}$$

$$\left\| X^{-1/2}\Delta X X^{-1/2} \right\|_F \leq \frac{2}{\sqrt{\lambda_{\min}}}\|\widehat{H}\|_F, \tag{6.3}$$

$$\left\| X^{1/2}\Delta S S^{-1/2} \right\|_F \leq 3\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{1/2}\|\widehat{H}\|_F, \tag{6.4}$$

where $\lambda_{\min} \equiv \lambda_{\min}[XS]$, $\lambda_{\max} \equiv \lambda_{\max}[XS]$, $U \equiv \langle\langle \Delta X \rangle\rangle_{X^{1/2}}$, and $\widehat{H} \equiv HX^{-1/2}S^{-1/2}$.

Proof. It follows from $(R, r) = (0, 0)$, (2.11b), and (2.11c) that $\Delta X \bullet \Delta S = \mathrm{Tr}\ (\Delta X \Delta S) = 0$, which together with (2.14) imply that

$$\mathrm{Tr}\ (UX^{1/2}\Delta S) = 0. \tag{6.5}$$

Multiplying (2.13) on the left by U and on the right by $X^{-1/2}$, taking the trace of both sides of the equality, and using (6.5), we obtain

$$\mathrm{Tr}\ (U^2S) + \mathrm{Tr}\ \left(UX^{1/2}SUX^{-1/2}\right) = \mathrm{Tr}\ \left(UHX^{-1/2}\right). \tag{6.6}$$

Since $UX^{-1/2}U \succeq 0$ and, by assumption, $X^{1/2}S + SX^{1/2} \succ 0$, we have

$$\begin{aligned} \mathrm{Tr}\ \left(UX^{1/2}SUX^{-1/2}\right) &= \mathrm{Tr}\ \left(UX^{-1/2}UX^{1/2}S\right) \\ &= \frac{1}{2}\mathrm{Tr}\ \left[UX^{-1/2}U\left(X^{1/2}S + SX^{1/2}\right)\right] \geq 0. \end{aligned}$$

Relation (6.6) together with the last inequality and the fact that $\mathrm{Tr}\ U^2S = \|S^{1/2}U\|_F^2$ imply that

$$\|S^{1/2}U\|_F^2 \leq \mathrm{Tr}\ \left(UHX^{-1/2}\right) = \mathrm{Tr}\ \left(S^{1/2}U\widehat{H}\right) \leq \|S^{1/2}U\|_F\|\widehat{H}\|_F,$$

from which (6.2) immediately follows. To show (6.3), observe that by (2.14) we have

$$X^{-1/2}U + UX^{-1/2} = X^{-1/2}\Delta X X^{-1/2},$$

which together with (6.2) and the fact that $\|X^{-1/2}S^{-1/2}\|^2 = 1/\lambda_{\min}[XS]$ imply

$$\left\|X^{-1/2}\Delta X X^{-1/2}\right\|_F \le 2\|X^{-1/2}U\|_F \le 2\|X^{-1/2}S^{-1/2}\|\,\|S^{1/2}U\|_F \le \frac{2}{\sqrt{\lambda_{\min}}}\|\widehat{H}\|_F;$$

that is, (6.3) holds. To show (6.4), we multiply (2.13) on the right by $X^{-1/2}S^{-1/2}$ and rearrange to obtain

$$X^{1/2}\Delta S S^{-1/2} = \widehat{H} - US^{1/2} - X^{1/2}SUX^{-1/2}S^{-1/2}.$$

Taking the Frobenius norm of both sides of the last equality and using the triangle inequality, relation (6.2) and the fact that $\|X^{1/2}S^{1/2}\|^2 = \lambda_{\max}$ and $\|X^{-1/2}S^{-1/2}\|^2 = 1/\lambda_{\min}$, we obtain

$$\begin{aligned}\left\|X^{1/2}\Delta S S^{-1/2}\right\|_F &\le \|\widehat{H}\|_F + \|S^{1/2}U\|_F + \|X^{1/2}SUX^{-1/2}S^{-1/2}\|_F \\ &\le 2\|\widehat{H}\|_F + \|X^{1/2}S^{1/2}\|\,\|S^{1/2}U\|_F\|X^{-1/2}S^{-1/2}\| \\ &\le \left[2 + \left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{1/2}\right]\|\widehat{H}\|_F \;\le\; 3\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{1/2}\|\widehat{H}\|_F. \qquad \square\end{aligned}$$

THEOREM 6.2. *If $(X,S,y) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \times \Re^m$ is such that $X^{1/2}S + SX^{1/2} \succ 0$ then, for every $(H,R,r) \in \mathcal{S}^n \times \mathcal{S}^n \times \Re^m$, system* (2.11) *has exactly one solution. In particular, for any $(\widetilde{X},\widetilde{S},\widetilde{y}) \in \mathcal{S}^n_{++}\times\mathcal{S}^n_{++}\times\Re^m$ and any nonsingular matrix $\widetilde{P} \in \Re^{n\times n}$ satisfying* (6.1)*, system* (5.4) *has exactly one solution.*

Proof. The proof of the first part is analogous to that of Theorem 2.4. The only difference is that Lemma 6.1 should be invoked in place of Lemma 2.3. The second part follows from the fact that $(\Delta\widetilde{X}, \Delta\widetilde{S}, \Delta\tilde{y})$ is a solution of (5.4) if and only if $(\Delta X, \Delta S, \Delta y) \equiv (\widetilde{P}\Delta\widetilde{X}\widetilde{P}^T, \widetilde{P}^{-T}\Delta\widetilde{S}\widetilde{P}^{-1}, \Delta\tilde{y})$ is a solution of (2.11) with (H,R,r) given by (2.12). $\square$

LEMMA 6.3. *If $(X,S,y) \in \mathcal{F}^0(P) \times \mathcal{F}^0(D)$ is such that $X^{1/2}S + SX^{1/2} \succ 0$, then*

$$\|S^{1/2}U_0^{(2)}\| \le \frac{2}{\sqrt{\lambda_{\min}}}\|\widehat{H}\|_F^2.$$

Proof. Multiplying (3.10) on the left by $X^{-1/2}$ and on the right by $S^{1/2}$, setting $\alpha = 0$ and using (3.12), we obtain

$$U_0^{(2)}S^{1/2} + X^{-1/2}U_0^{(2)}X^{1/2}S^{1/2} = -2X^{-1/2}UUS^{1/2}. \tag{6.7}$$

Using the assumption that $X^{1/2}S + SX^{1/2} \succ 0$, we have

$$\begin{aligned}\left(U_0^{(2)}S^{1/2}\right) \bullet \left(X^{-1/2}U_0^{(2)}X^{1/2}S^{1/2}\right) &= \mathrm{Tr}\left(U_0^{(2)}X^{-1/2}U_0^{(2)}X^{1/2}S\right) \\ &= \frac{1}{2}\mathrm{Tr}\left[U_0^{(2)}X^{-1/2}U_0^{(2)}\left(X^{1/2}S + SX^{1/2}\right)\right] \;\ge\; 0.\end{aligned}$$

Taking the Frobenius norm of both sides of (6.7) and using the last inequality together with (6.2) and the fact that $\|X^{-1/2}S^{-1/2}\|^2 = 1/\lambda_{\min}$, we obtain

$$\begin{aligned}\|S^{1/2}U_0^{(2)}\|_F &\le \left(\|S^{1/2}U_0^{(2)}\|_F^2 + \|X^{-1/2}U_0^{(2)}X^{1/2}S^{1/2}\|_F^2\right)^{1/2}\\ &\le \|U_0^{(2)}S^{1/2} + X^{-1/2}U_0^{(2)}X^{1/2}S^{1/2}\|_F \;=\; 2\|X^{-1/2}UUS^{1/2}\|_F\\ &\le 2\|X^{-1/2}S^{-1/2}\|\,\|S^{1/2}U\|_F^2 \;\le\; \frac{2}{\sqrt{\lambda_{\min}}}\|\widehat{H}\|_F^2. \qquad \square\end{aligned}$$

LEMMA 6.4. *Let $\tau \in (0,1)$ be given. If $\alpha > 0$ is such that*

$$\alpha \max\left\{\|\widehat{\Delta X}\|, \|\widehat{\Delta S}\|\right\} \le \tau,$$

then

$$\left\|U_\alpha^{(2)}\Delta S X_\alpha^{1/2}\right\|_F \le \frac{6\sqrt{2}}{(1-\tau)^3}\frac{\lambda_{\max}}{\lambda_{\min}^{3/2}}\|\widehat{H}\|_F^3, \tag{6.8}$$

$$\left\|U_\alpha^{(1)}\Delta S U_\alpha^{(1)}\right\|_F \le \frac{6}{(1-\tau)^3}\frac{\lambda_{\max}}{\lambda_{\min}^{3/2}}\|\widehat{H}\|_F^3, \tag{6.9}$$

$$\left\|U_\alpha^{(2)}S_\alpha U_\alpha^{(1)}\right\|_F \le \frac{4}{(1-\tau)^5}\frac{\lambda_{\max}}{\lambda_{\min}^{3/2}}\|\widehat{H}\|_F^3, \tag{6.10}$$

$$\left\|U_\alpha^{(3)}S_\alpha X_\alpha^{1/2}\right\|_F \le \frac{12\sqrt{2}}{(1-\tau)^5}\frac{\lambda_{\max}}{\lambda_{\min}^{3/2}}\|\widehat{H}\|_F^3. \tag{6.11}$$

Proof. Using (3.18), (3.21), (3.22), Lemma 6.1, and the fact that $\|X^{1/2}S^{1/2}\| = \sqrt{\lambda_{\max}}$, we obtain

$$\begin{aligned}\left\|U_\alpha^{(2)}\Delta S X_\alpha^{1/2}\right\|_F &\le \left\|U_\alpha^{(2)}X_\alpha^{-1/2}\right\|_F \left\|X_\alpha^{1/2}\Delta S X_\alpha^{1/2}\right\|\\ &\le \frac{\|\widehat{\Delta X}\|_F^2}{\sqrt{2}(1-\tau)^2}\left\|(D_\alpha^X)^{-1}\right\|^2 \left\|X^{1/2}\Delta S X^{1/2}\right\|_F\\ &\le \frac{\|\widehat{\Delta X}\|_F^2}{\sqrt{2}(1-\tau)^3}\left\|X^{1/2}\Delta S S^{-1/2}\right\|_F \left\|X^{1/2}S^{1/2}\right\| \;\le\; \frac{6\sqrt{2}}{(1-\tau)^3}\frac{\lambda_{\max}}{\lambda_{\min}^{3/2}}\|\widehat{H}\|_F^3\end{aligned}$$

and

$$\begin{aligned}\left\|U_\alpha^{(1)}\Delta S U_\alpha^{(1)}\right\|_F &\le \left\|U_\alpha^{(1)}X_\alpha^{-1/2}\right\|_F^2 \left\|X_\alpha^{1/2}\Delta S X_\alpha^{1/2}\right\|\\ &\le \frac{\|\widehat{\Delta X}\|_F^2}{2(1-\tau)^2}\left\|(D_\alpha^X)^{-1}\right\|^2 \left\|X^{1/2}\Delta S X^{1/2}\right\|_F\\ &\le \frac{\|\widehat{\Delta X}\|_F^2}{2(1-\tau)^3}\left\|X^{1/2}\Delta S S^{-1/2}\right\|_F \left\|X^{1/2}S^{1/2}\right\| \;\le\; \frac{6}{(1-\tau)^3}\frac{\lambda_{\max}}{\lambda_{\min}^{3/2}}\|\widehat{H}\|_F^3.\end{aligned}$$

Also, using (3.20), (3.21), (3.22), (3.23), and Lemma 6.1, we obtain

$$\begin{aligned}\left\|U_\alpha^{(2)}S_\alpha U_\alpha^{(1)}\right\|_F &\le \left\|U_\alpha^{(2)}X_\alpha^{-1/2}\right\|_F \left\|U_\alpha^{(1)}X_\alpha^{-1/2}\right\|_F \left\|X_\alpha^{1/2}S_\alpha^{1/2}\right\|^2\\ &\le \frac{\|\widehat{\Delta X}\|_F^3 \left\|X^{1/2}S^{1/2}\right\|^2}{2(1-\tau)^5} \;\le\; \frac{4}{(1-\tau)^5}\frac{\lambda_{\max}}{\lambda_{\min}^{3/2}}\|\widehat{H}\|_F^3\end{aligned}$$

and

$$\begin{aligned}\left\|U_\alpha^{(3)} S_\alpha X_\alpha^{1/2}\right\|_F &\le \left\|U_\alpha^{(3)} X_\alpha^{-1/2}\right\|_F \left\|X_\alpha^{1/2} S_\alpha^{1/2}\right\|^2 \\ &\le \frac{3\|\widehat{\Delta X}\|_F^3 \left\|X^{1/2}S^{1/2}\right\|^2}{\sqrt{2}(1-\tau)^5} \ \le\ \frac{12\sqrt{2}}{(1-\tau)^5}\frac{\lambda_{\max}}{\lambda_{\min}^{3/2}}\|\widehat{H}\|_F^3. \qquad \square\end{aligned}$$

LEMMA 6.5. *If $(X,S,y) \in \mathcal{F}^0(P)\times\mathcal{F}^0(D)$ is such that $X^{1/2}S+SX^{1/2} \succ 0$, then*

$$\|\phi''(0)\|_F \le 18\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{1/2}\|\widehat{H}\|_F^2.$$

In addition, if $\alpha > 0$ is such that $\alpha\max\{\|\widehat{\Delta X}\|, \|\widehat{\Delta S}\|\} \le 1/2$, then $(X_\alpha, S_\alpha, y_\alpha) \in \mathcal{F}^0(P)\times\mathcal{F}^0(D)$ and

$$\|\phi'''(\alpha)\|_F \le 3000\,\frac{\lambda_{\max}}{\lambda_{\min}^{3/2}}\|\widehat{H}\|_F^3.$$

Proof. Using (3.8) with $\alpha = 0$, (3.12), (6.2), (6.4), and Lemma 6.3, we obtain

$$\begin{aligned}\|\phi''(0)\|_F &\le 2\|X^{1/2}SU_0^{(2)}\|_F + 2\|USU\|_F + 4\|X^{1/2}\Delta SU\|_F \\ &\le 2\|X^{1/2}S^{1/2}\|\,\|S^{1/2}U_0^{(2)}\|_F + 2\|S^{1/2}U\|_F^2 + 4\|X^{1/2}\Delta SS^{-1/2}\|_F\|S^{1/2}U\|_F \\ &\le 4\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{1/2}\|\widehat{H}\|_F^2 + 2\|\widehat{H}\|_F^2 + 12\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{1/2}\|\widehat{H}\|_F^2 \\ &\le 18\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{1/2}\|\widehat{H}\|_F^2.\end{aligned}$$

Since $\alpha\max\{\|\widehat{\Delta X}\|, \|\widehat{\Delta S}\|\} \le 1/2$ and $(X,S,y) \in \mathcal{F}^0(P)\times\mathcal{F}^0(D)$, we have $(X_\alpha, S_\alpha) \in \mathcal{F}^0(P)\times\mathcal{F}^0(D)$. It follows from (3.8) and Lemma 6.4 with $\tau = 1/2$ that

$$\begin{aligned}\|\phi'''(\alpha)\|_F &\le 2\left\|U_\alpha^{(3)}S_\alpha X_\alpha^{1/2}\right\|_F + 6\left\|U_\alpha^{(2)}S_\alpha U_\alpha^{(1)}\right\|_F \\ &\quad +6\left\|U_\alpha^{(2)}\Delta SX_\alpha^{1/2}\right\|_F + 6\left\|U_\alpha^{(1)}\Delta SU_\alpha^{(1)}\right\|_F \\ &\le \left(768\sqrt{2}+768+288\sqrt{2}+288\right)\frac{\lambda_{\max}}{\lambda_{\min}^{3/2}}\|\widehat{H}\|_F^3 \ \le\ 3000\,\frac{\lambda_{\max}}{\lambda_{\min}^{3/2}}\|\widehat{H}\|_F^3. \qquad \square\end{aligned}$$

THEOREM 6.6. *Let $\gamma, \sigma \in (0,1)$ be given. Suppose that $(X,S,y) \in \mathcal{N}_{-\infty}(\gamma)$ satisfies $X^{1/2}S+SX^{1/2} \succ 0$ and $(\Delta X, \Delta S, \Delta y)$ is the solution of (2.11) with (H,R,r) given by (2.12) and $\mu \equiv (X\bullet S)/n$. Let*

$$\hat{\alpha} \equiv \frac{\sigma\gamma(1-\gamma)^{1/2}}{30n^{3/2}\zeta}, \tag{6.12}$$

where $\zeta \equiv 1-2\sigma+\sigma^2/(1-\gamma) = (1-\sigma)^2+\gamma\sigma^2/(1-\gamma)$. Then for any $\alpha \in [0,\hat{\alpha}]$, we have:

(a) $(X_\alpha, S_\alpha, y_\alpha) \in \mathcal{N}_{-\infty}(\gamma)$;

(b) $X_\alpha \bullet S_\alpha = (1-\alpha+\alpha\sigma)(X\bullet S)$.

Proof. Statement (b) is an immediate consequence of Lemma 3.1. Using Lemma 6.1, relations (3.30) and (6.12), and the fact that $\gamma \le 1$ and $\zeta \ge \gamma\sigma^2/(1-\gamma)$, we obtain

$$\begin{aligned}
\hat{\alpha}\ \max\left\{\|\widehat{\Delta X}\|, \|\widehat{\Delta S}\|\right\} &\le \hat{\alpha}\ \max\left\{\left\|X^{-1/2}\Delta X X^{-1/2}\right\|_F,\right. \\
&\qquad\qquad \left.\left\|X^{-1/2}S^{-1/2}\right\|\left\|X^{1/2}\Delta S S^{-1/2}\right\|_F\right\} \\
&\le \hat{\alpha}\max\left\{\frac{2}{\sqrt{\lambda_{\min}}}\|\widehat{H}\|_F, 3\frac{\lambda_{\max}^{1/2}}{\lambda_{\min}}\|\widehat{H}\|_F\right\} \\
&\le \hat{\alpha}\max\left\{\frac{2}{\sqrt{(1-\gamma)\mu}}(\zeta n\mu)^{1/2}, 3\frac{(n\mu)^{1/2}}{(1-\gamma)\mu}(\zeta n\mu)^{1/2}\right\} \\
&\le \max\left\{\frac{\sigma\gamma}{15n\sqrt{\zeta}}, \frac{\sigma\gamma}{10(\zeta(1-\gamma)n)^{1/2}}\right\} \ \le\ \frac{1}{2}.
\end{aligned}$$

Hence, it follows from Lemma 6.5, Lemma 3.9(c), and the fact that $\lambda_{\min} \ge (1-\gamma)\mu$ and $\lambda_{\max} \le n\mu$ that $(X_\alpha, S_\alpha, y_\alpha) \in \mathcal{F}^0(P) \times \mathcal{F}^0(D)$ for any $\alpha \in [0, \hat{\alpha}]$, and

$$\begin{aligned}
\|\phi''(0)\|_F &\le 18\frac{n^{1/2}}{(1-\gamma)^{1/2}}\zeta n\mu, \\
\sup_{\xi\in[0,\tilde{\alpha}]}\|\phi'''(\xi)\|_F &\le 3000\frac{n}{(1-\gamma)^{3/2}}(\zeta n)^{3/2}\mu.
\end{aligned}$$

These two inequalities together with (3.14), (6.12), Lemma 3.1, and the fact that $\gamma \le 1$ and $\zeta \ge \gamma\sigma^2/(1-\gamma)$ imply that for every $\alpha \in [0, \tilde{\alpha}]$,

$$\begin{aligned}
\left\|X_\alpha^{1/2}S_\alpha X_\alpha^{1/2} - \mu(\alpha)I\right\|_{-\infty} &= \|\phi(\alpha)\|_{-\infty} \\
&\le (1-\alpha)\|\phi(0)\|_{-\infty} \\
&\qquad +\alpha\left[\frac{1}{2}\hat{\alpha}\|\phi''(0)\|_F + \frac{1}{6}\hat{\alpha}^2\sup_{\xi\in[0,\alpha]}\|\phi'''(\xi)\|_F\right] \\
&\le (1-\alpha)\gamma\mu \\
&\qquad +\alpha\left[9\hat{\alpha}\frac{n^{1/2}}{(1-\gamma)^{1/2}}\zeta n\mu + 500\hat{\alpha}^2\frac{n}{(1-\gamma)^{3/2}}(\zeta n)^{3/2}\mu\right] \\
&\le (1-\alpha)\gamma\mu + \alpha\left[\frac{3}{10}\sigma\gamma\mu + \frac{5\sigma^2\gamma^2\mu}{9(1-\gamma)^{1/2}n^{1/2}\zeta^{1/2}}\right] \\
&\le (1-\alpha)\gamma\mu + \alpha\sigma\gamma\mu\left[\frac{3}{10} + \frac{5\gamma^{1/2}}{9n^{1/2}}\right] \\
&\le (1-\alpha)\gamma\mu + \alpha\sigma\gamma\mu \ =\ \gamma\mu(\alpha).
\end{aligned}$$

Hence, $(X_\alpha, S_\alpha, y_\alpha) \in \mathcal{N}_{-\infty}(\gamma)$ for every $\alpha \in [0, \tilde{\alpha}]$; that is, (a) holds. □

COROLLARY 6.7 (polynomiality of long-step path-following algorithm for the MT* family). *Let constants $\gamma, \bar{\sigma} \in (0,1)$ be given. For Algorithm* II, *assume that $(\widetilde{X}^0, \widetilde{S}^0, \widetilde{y}^0) \in \mathcal{N}_{-\infty}(\gamma)$ and that the sequences $\{\sigma_k\}$ and $\{\alpha_k\}$ are chosen according to*

$$\begin{aligned}
&\sigma_k = \bar{\sigma}, \\
&\alpha_k = \max\left\{\alpha \in [0,1] : (\widetilde{X}^k, \widetilde{S}^k, \widetilde{y}^k) + \alpha(\Delta\widetilde{X}^k, \Delta\widetilde{S}^k, \Delta\widetilde{y}^k) \in \widetilde{\mathcal{N}}_{-\infty}(\gamma)\right\}.
\end{aligned}$$

Then, the sequence of iterates $\{(\widetilde{X}^k, \widetilde{S}^k, \widetilde{y}^k)\} \subset \mathcal{N}_{-\infty}(\gamma)$ *generated by Algorithm* II *satisfies* $\widetilde{X}^k \bullet \widetilde{S}^k \le (1-\bar{\eta})^k(\widetilde{X}^0 \bullet \widetilde{S}^0)$ *for all* $k \ge 0$*, where*

$$\bar{\eta} \equiv \frac{\bar{\sigma}(1-\bar{\sigma})\gamma(1-\gamma)^{1/2}}{30n^{3/2}\bar{\zeta}},$$

and $\bar{\zeta} \equiv 1 - 2\bar{\sigma} + \bar{\sigma}^2/(1-\gamma)$*. Moreover, if the quantity* $\max\{\gamma^{-1}, (1-\gamma)^{-1}, \bar{\sigma}^{-1}, (1-\bar{\sigma})^{-1}\}$ *is independent of* n*, then the method terminates in at most* $\mathcal{O}(n^{3/2}L)$ *iterations.*

Proof. One step of the algorithm in the scaled space is analyzed by Theorem 6.6. By translating the result into the terms of the original space using the invariance of μ and $d_{-\infty}$ and (5.2c), the result readily follows. □

We have thus established an $O(n^{3/2}L)$ iteration complexity for the long-step path-following feasible algorithm based on any member of the MT* family. A natural question is whether our approach yields better iteration complexities for the special cases in which $X = I$ (the HRVW/KSH/M dual direction), $SXS = I$ (the HRVW/KSH/M direction), $S = I$ (the NT direction), and $X = S$ (the MTW direction). Unfortunately, our approach does not seem to yield the $O(nL)$ iteration-complexity bound that has been obtained in Monteiro and Zhang [21] for the NT direction nor to improve the $O(n^{3/2}L)$ iteration-complexity bound for the HRVW/KSH/M dual direction obtained in Monteiro [15]. For the MTW direction, we can show that the long-step algorithm has an $O(n^{11/8}L)$ iteration-complexity bound, slightly improving the general $O(n^{3/2}L)$ bound. We omit the proof of this claim here.

7. Concluding remarks. We proposed a new family of primal-dual interior-point methods for SDP. The method is based on the application of Newton's method to the equation

$$(\widetilde{P}\widetilde{X}\widetilde{P}^T)^{1/2}(\widetilde{P}^{-T}\widetilde{S}\widetilde{P}^{-1})(\widetilde{P}\widetilde{X}\widetilde{P}^T)^{1/2} - \nu I = 0$$

for some $\nu > 0$ and scaling nonsingular matrix $\widetilde{P}$. We proved existence of the Newton direction for any $(\widetilde{X}, \widetilde{S}, \widetilde{y}) \in \widetilde{\mathcal{N}}_\infty(\gamma)$ with $\gamma \in (0, 1/\sqrt{2})$, and established an $O(\sqrt{n}L)$ iteration-complexity bound for the short-step path-following algorithm and an $O(nL)$ iteration-complexity bound for the semilong-step path-following algorithm. Furthermore, we showed that for any interior feasible point $(\widetilde{X}, \widetilde{S}, \widetilde{y})$, the Newton direction corresponding to those scaling matrices $\widetilde{P}$ satisfying

$$(\widetilde{P}^{-T}\widetilde{S}\widetilde{P}^{-1})(\widetilde{P}\widetilde{X}\widetilde{P}^T)^{1/2} + (\widetilde{P}\widetilde{X}\widetilde{P}^T)^{1/2}(\widetilde{P}^{-T}\widetilde{S}\widetilde{P}^{-1}) \succ 0$$

always exists, and we established an $O(n^{3/2}L)$ iteration-complexity bound for the long-step path-following algorithm based on this subclass of scaling matrices. This subclass yields two well-known search directions, namely, the HRVW/KSH/M dual direction when $\widetilde{P} = \widetilde{X}^{-1/2}$, the HRVW/KSH/M direction when $\widetilde{P} = (\widetilde{S}\widetilde{X}\widetilde{S})^{1/2}$, and the NT direction when $\widetilde{P} = \widetilde{S}^{1/2}$.

It is possible to derive a symmetric MT family based on the central path equation $S^{1/2}XS^{1/2} - \nu I = 0$, obtained from the one in section 2 by interchanging the role of X and S. It is easy to see that the symmetric MT family obtained by applying Newton's method to this equation (in the scaled space) has similar properties to the one studied in this paper and that it contains the NT direction and the two HRVW/KSH/M directions.

It is interesting to compare the MZ family and the MT family in light of the motivation used in section 5 to derive the MT family. We know that search directions of the MT family correspond to the Newton direction for the central path equation $X^{1/2}SX^{1/2} - \nu I = 0$ in the scaled space for an appropriate choice of $\widetilde{P}$. In a similar vein, it is easy to see that search directions of the MZ family correspond to the Newton direction for the central path equation $XS + SX - \nu I = 0$ in the scaled space for an appropriate choice of $\widetilde{P}$. This observation indicates the existence of a natural association of the MT family with the (pure) Newton direction of section 2 and of the MZ family with the (pure) Newton AHO direction.

Based on the theoretical results obtained so far, the pure Newton direction of section 2 has clear advantages over the AHO direction in the sense that polynomial convergence of the semilong-step path-following algorithm is only known for the former direction. So far this is the only pure Newton path-following algorithm which is polynomially convergent and is based on a wide neighborhood of the central path.

The MT* family also has theoretical advantages over the MZ* family based on the results so far. While for the MZ* family, the iteration-complexity bound depends on a certain condition number associated with the sequence $\{P^k\}$ of scaling matrices, the corresponding bound for the MT* family does not depend on this sequence.

After the release of this paper, Monteiro and Zanjácomo [20] have reported promising computational results for algorithms based on the pure Newton direction (2.11) and two other pure Newton directions based on the central path equations:

$$\begin{aligned} S^{1/2}XS^{1/2} &= \nu I, \\ L_S^T X L_S &= \nu I, \end{aligned}$$

respectively, where L_S denotes the Cholesky lower triangular factor of S, that is, $S = L_S L_S^T$ with L_S lower triangular.

Finally, we mention that an interesting topic for future study would be to develop algorithms based on the pure Newton direction (2.11) that are superlinearly or quadratically convergent. We refer the reader to [9, 13, 29, 28], where quadratically convergent SDP algorithms based on other primal-dual directions are developed under the presence of strict complementarity and/or nondegeneracy assumptions.

REFERENCES

[1] F. Alizadeh, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.

[2] F. Alizadeh, J.-P. Haeberly, and M. Overton, *Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results*, SIAM J. Optim., 8 (1998), pp. 746–768.

[3] R. M. Freund, *Complexity of an Algorithm for Finding an Approximate Solution of a Semidefinite Program with No Regularity Condition*, Working paper OR 302-94, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, December 1994.

[4] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. Optim, 6 (1996), pp. 342–361.

[5] F. Jarre, *An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices*, SIAM J. Control. Optim., 31 (1993), pp. 1360–1377.

[6] M. Kojima, S. Mizuno, and A. Yoshise, *A polynomial-time algorithm for a class of linear complementarity problems*, Math. Programming, 44 (1989), pp. 1–26.

[7] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point algorithm for linear programming*, in Progress in Mathematical Programming: Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 29–47.

[8] M. Kojima, M. Shida, and S. Shindoh, *A Note on the Nesterov-Todd and the Kojima-Shindoh-Hara Search Directions in Semidefinite Programming*, Optim. Methods Softw., to appear.

[9] M. Kojima, M. Shida, and S. Shindoh, *A predictor-corrector interior-point algorithm for the semidefinite linear complementarity problem using the Alizadeh–Haeberly–Overton search direction*, SIAM J. Optim., 9 (1999), pp. 444–465.

[10] M. Kojima, M. Shida, and S. Shindoh, *Local convergence of predictor-corrector infeasible-interior-point algorithms for SDPs and SDLCPs*, Math. Programming, 80 (1998), pp. 129–160.

[11] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices*, SIAM J. Optim., 7 (1997), pp. 86–125.

[12] C.-J. Lin and R. Saigal, *A Predictor-Corrector Method for Semi-Definite Programming*, Working paper, Dept. of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1995.

[13] Z.-Q. Luo, J. F. Sturm, and S. Zhang, *Superlinear convergence of a symmetric primal-dual path-following algorithm for semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 59–81.

[14] S. Mizuno, M. J. Todd, and Y. Ye, *On adaptive step primal-dual interior–point algorithms for linear programming*, Math. Oper. Res., 18 (1993), pp. 945–981.

[15] R. D. C. Monteiro, *Primal-dual path-following algorithms for semidefinite programming*, SIAM J. Optim., 7 (1997), pp. 663–678.

[16] R. D. C. Monteiro, *Polynomial convergence of primal-dual algorithms for semidefinite programming based on Monteiro and Zhang family of directions*, SIAM J. Optim., 8 (1998), pp. 59–81.

[17] R. D. C. Monteiro and I. Adler, *Interior path-following primal-dual algorithms. Part* I: *Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[18] R. D. C. Monteiro and I. Adler, *Interior path-following primal-dual algorithms. Part* II: *Convex quadratic programming*, Math. Programming, 44 (1989), pp. 43–66.

[19] R. D. C. Monteiro and T. Tsuchiya, *Polynomiality of primal-dual algorithms for semidefinite linear complementarity problems based on the Kojima-Shindoh-Hara family of directions*, Manuscript, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, 1996; Math. Programming, to appear.

[20] R. D. C. Monteiro and P. R. Zanjácomo, *Implementation of Primal-Dual Methods for Semidefinite Programming Based on Monteiro and Tsuchiya Newton Directions and Their Variants*, Tech. Report, School of ISyE, Georgia Institute of Technology, Atlanta, 1997; Optim. Math. Softw., to appear.

[21] R. D. C. Monteiro and Y. Zhang, *A unified analysis for a class of path-following primal-dual interior-point algorithms for semidefinite programming*, Math. Programming, 81 (1998), pp. 281–299.

[22] Y. E. Nesterov and A. S. Nemirovskii, *A general approach to the design of optimal methods for smooth convex functions minimization*, Ekonomika i Matem. Metody, 24 (1988), pp. 509–517 (in Russian). (English transl.: Matekon: Translations of Russian and East European Math. Economics.)

[23] Y. E. Nesterov and A. S. Nemirovskii, *Self-Concordant Functions and Polynomial Time Methods in Convex Programming*, Preprint, Central Economic & Mathematical Institute, USSR Acad. Sci. Moscow, USSR, 1989.

[24] Y. E. Nesterov and A. S. Nemirovskii, *Optimization over Positive Semidefinite Matrices: Mathematical Background and User's Manual*, Tech. report, Central Economic & Mathematical Institute, USSR Acad. Sci. Moscow, USSR, 1990.

[25] Y. E. Nesterov and A. S. Nemirovskii, *Interior Point Methods in Convex Programming: Theory and Applications*, SIAM, Philadelphia, 1994.

[26] Y. E. Nesterov and M. Todd, *Primal-dual interior-point methods for self-scaled cones*, SIAM J. Optim., 8 (1998), pp. 324–364.

[27] Y. E. Nesterov and M. Todd, *Self-scaled barriers and interior-point methods for convex programming*, Math. Oper. Res., 22 (1997), pp. 1–42.

[28] F. A. Potra and R. Sheng, *On the Local Convergence of a Predictor-Corrector Method for Semidefinite Programming*, Reports on Computational Mathematics 98, Dept. of Mathematics, The University of Iowa, Iowa City, 1997.

[29] F. A. Potra and R. Sheng, *A superlinearly convergent primal-dual infeasible-interior-point algorithm for semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 1007–1028.

[30] J. F. Sturm and S. Zhang, *Symmetric Primal-Dual Path-Following Algorithms for Semidefinite Programming*, Report 9554/A, Econometric Institute, Erasmus University, Rotterdam, the Netherlands, November 1995.

[31] K. Tanabe, *Complementarity-enforcing centered Newton method for mathematical program-*

ming: Global methods, in New Method for Linear Programming, K. Tone, ed., Cooperative Research Report 5, Institute of Statistical Mathematics, Tokyo, Japan, 1987, pp. 118–144.

[32] K. Tanabe, *Centered Newton method for mathematical programming*, in System Modelling and Optimization, M. Iri and K. Yajima, eds., Springer-Verlag, Tokyo, Japan, 1988, pp. 197–206.

[33] M. Todd, *On the Search Directions in Interior-Point Methods for Semidefinite Programming*, Tech. Report 1205, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, October, 1997.

[34] M. J. Todd, K. C. Toh, and R. H. Tütüncü, *On the Nesterov-Todd direction in semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 769–796.

[35] P. Tseng, *Search directions and convergence analysis of some infeasible path-following methods for the monotone semi-definite LCP*, Optim. Methods Softw., 9 (1998), pp. 245–268.

[36] L. Vandenberghe and S. Boyd, *A primal-dual potential reduction method for problems involving matrix inequalities*, Math. Programming, 69 (1995), pp. 205–236.

[37] Y. Ye, *A class of projective transformations for linear programming*, SIAM J. Comput., 19 (1990), pp. 457–466.

[38] Y. Zhang, *On extending some primal-dual interior–point algorithms from linear programming to semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 365–386.

SIAM J. OPTIM.
Vol. 9, No. 3, pp. 578–604

PRACTICAL UPDATE CRITERIA FOR REDUCED HESSIAN SQP: GLOBAL ANALYSIS[*]

Y. F. XIE[†] AND R. H. BYRD[‡]

Abstract. In this paper, a new update criterion is proposed to improve the Nocedal–Overton update criterion for reduced Hessian successive quadratic programming (SQP). Global and R-linear convergence is proved for the new criterion and the Nocedal–Overton criterion using nonorthogonal basis matrices, which allow efficient implementations of the reduced Hessian SQP for solving large-scale equality constrained problems.

Key words. constrained optimization, nonlinear programming, quasi-Newton methods, reduced Hessian algorithms, successive quadratic programming

AMS subject classifications. 49, 65

PII. S105262349427861X

1. Introduction.

In this paper, we consider some critical issues in efficiently solving nonlinearly constrained optimization problems by reduced Hessian successive quadratic programming (SQP).

SQP algorithms have proven to be very efficient for solving small and medium size equality constrained optimization problems,

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & c(x) = 0, \end{array} \tag{1.1}$$

where $f : R^n \to R^1$ and $c : R^n \to R^t$ for positive integers n and t, with $n > t$ (see Han [8] and Powell [14]). The reduced Hessian approach allows us to use SQP for a significant class of very large problems, especially when implemented with generalized basis matrices.

Given an approximate solution x_k, an SQP algorithm computes a search direction d_k from the quadratic programming problem:

$$\begin{array}{ll} \min & g_k^T d + \frac{1}{2} d^T M_k d \\ \text{s.t.} & c(x_k) + A_k^T d = 0, \end{array}$$

where $g_k = \nabla f(x_k)$, $A_k = \nabla c(x_k) \equiv (\nabla c_1(x_k), \ldots, \nabla c_t(x_k))$, and the matrix M_k approximates the Hessian $G_k = \nabla_{xx}^2 L(x_k, \lambda_k)$ of the Lagrangian function of (1.1), $L(x, \lambda)$, which has the form

$$L(x, \lambda) = f(x) + \lambda^T c(x), \tag{1.2}$$

where λ is a Lagrange multiplier. The Lagrange multiplier can be estimated at x_k by

$$\lambda_k = -(A_k)_L^{-1} g_k, \tag{1.3}$$

[*] Received by the editors December 13, 1994; accepted for publication (in revised form) March 10, 1998; published electronically April 19, 1999.
http://www.siam.org/journals/siopt/9-3/27861.html

[†] NOAA, ERL/FSL, Forecast Systems Laboratory, 325 Broadway, Boulder, CO 80303 (xiey@fsl.noaa.gov). This author was supported by Air Force Office of Scientific Research grant AFOSR-90-0109 and National Science Foundation grant CCR-8920519.

[‡] Department of Computer Science, Campus Box 430, University of Colorado, Boulder, CO 80309. This author was supported in part by Air Force Office of Scientific Research grant AFOSR-90-0109, Army Research Office grant DAAL 03-91-G-0151, and National Science Foundation grant CCR-9101795.

where $(A_k)_L^{-1}$ is a left inverse of A_k, giving an approximate solution to $g_k + A_k\lambda = 0$. Then a new approximation to the solution x^* is given by

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.4}$$

where some line search strategy is used to determine step length α_k and ensure convergence. (Alternatively, a trust region can be used instead of a line search, but this paper will focus on the more widely used line search approach.) Although SQP methods are among the best approaches for small and medium size problems, the applicability of this approach for very large problems is limited because of the need to store and manipulate an $n \times n$ matrix M_k, which cannot be expected to be sparse if a quasi-Newton update is used.

However, reduced Hessian SQP (RHSQP) algorithms, which use a matrix B_k to approximate $Z_k^T G_k Z_k$ where Z_k is a null space basis, can potentially be very efficient for solving large-scale constrained optimization problems (i.e., n is very large), especially for the problems where $n - t$ is small relative to n.

RHSQP algorithms have two advantages compared with the other SQP algorithms:

- It is natural to use quasi-Newton methods to approximate the reduced Hessian matrix $Z_k^T G_k Z_k$ because this matrix is positive definite when x_k is close to the solution of (1.1) and the second-order sufficient optimality condition holds at the solution.
- It is more efficient to store an $(n-t) \times (n-t)$ matrix B_k than to store an $n \times n$ matrix M_k. Thus for a given n, a larger t requires less space for storing B_k. This is an important advantage for solving large-scale problems.

How to update the matrix B_k by quasi-Newton methods is an important issue, and many update strategies have been proposed; see, for example, [2], [3], [6], [7], [10], and [11]. Among these update strategies, there are two principal approaches, and others are slight variations of these two. One uses exact null space information [3] and the other uses full step information [11]. We call them the null space secant update strategy and the step secant update strategy, respectively.

To ensure the accuracy of the step secant update strategy, Nocedal and Overton suggest an update criterion [11], under which B_k is updated. In order to improve the numerical performance of the step secant update strategy using the Nocedal–Overton criterion, a new update criterion is proposed in this paper.

For the methods using these two update strategies, several convergence results have been established. For the RHSQP algorithms using the null space secant update strategy, Coleman and Conn [3] have proved two-step Q-superlinear convergence assuming x_1 and B_1 are sufficiently close to x^* and $Z_*^T \nabla_{xx}^2 L(x^*, \lambda^*) Z_*$, respectively, and Byrd and Nocedal [2] have shown its global convergence and R-linear and two-step Q-superlinear convergence with the l_1 and Fletcher merit functions. For the step secant update strategy with the Nocedal–Overton update criterion (2.12), Nocedal and Overton [11] established local two-step Q-superlinear convergence for x_1 and B_1 sufficiently close to x^* and $Z_*^T \nabla_{xx}^2 L(x^*, \lambda^*) Z_*$, respectively; however, no global and R-linear convergence was proved. All of these analyses assume Z_k is an orthonormal basis of $null(A_k^T)$.

A general basis Z_k of $null(A_k)$ has been used by Fletcher [5], Gabay [6], and Gilbert [7]. Fletcher has discussed his successive linear programming algorithm using any basis of $null(A_k)$. Gabay and Gilbert use general bases to discuss RHSQP. Gabay's update strategy is equivalent to the step secant update strategy, but he used

Powell's damping technique to ensure positive definiteness. It is difficult to prove superlinear convergence without assuming that $\{B_k\}$ and $\{B_k^{-1}\}$ are bounded for the Powell damped technique. Although Gilbert [7] has discussed general Z_k in his global analysis, his longitudinal path strategy may cost more gradient evaluations.

Because of the complexity of the analysis, superlinear convergence will be discussed in a second paper subsequent to this one. This paper is devoted to proposing a new update criterion and to proving the global and R-linear convergence for the step secant update strategy with two commonly used merit functions. All of these results are proved without requiring orthogonality of the basis matrix Z_k and without assuming that $\{B_k\}$ and $\{B_k^{-1}\}$ are bounded. In the next section, the new update criterion used in the step secant update strategy is introduced, and the general RHSQP algorithms and the merit functions used to force global convergence are described. The global convergence analysis will be presented in section 3. The R-linear convergence will be established in section 4. The numerical experiments are presented in section 5.

In the rest of the paper, the following notation is used:

$$\begin{aligned} S_1 &= \{j \mid B_{j+1} = B^{BFGS}(B_j, s_j, y_j)\}, \\ S_2 &= \{j \mid B_{j+1} = B_j\}, \\ S_1^k &= [1, 2, \ldots, k] \cap S_1, \\ S_2^k &= [1, 2, \ldots, k] \cap S_2, \end{aligned}$$

where

$$B^{BFGS}(B_j, s_j, y_j) = B_j - \frac{B_j s_j s_j^T B_j}{s_j^T B_j s_j} + \frac{y_j y_j^T}{s_j^T y_j} \tag{1.5}$$

is the BFGS update. Furthermore, $\|\cdot\|$ stands for the l_2 norm, $\|\cdot\|_1$ for the l_1 norm, and $\|\cdot\|_\infty$ for the infinity norm.

2. A new update criterion and general RHSQP with merit functions. The reduced Hessian technique can be derived from SQP methods by considering general basis matrices and their pseudoinverses. Suppose Z_k is any basis matrix of the null space of A_k^T (i.e., $A_k^T Z_k = 0$ and Z_k is full rank). Let $(Z_k)_L^{-1}$ and $(A_k)_L^{-1}$ be left inverse matrices of Z_k and A_k, respectively, that satisfy

$$(A_k)_L^{-1}(Z_k)_L^{-T} = (Z_k)_L^{-1}(A_k)_L^{-T} = 0. \tag{2.1}$$

Then, we have

$$\begin{pmatrix} (Z_k)_L^{-1} \\ A_k^T \end{pmatrix} (Z_k \quad (A_k)_L^{-T}) = (Z_k \quad (A_k)_L^{-T}) \begin{pmatrix} (Z_k)_L^{-1} \\ A_k^T \end{pmatrix} = I, \tag{2.2}$$

and G_k can be written as the following:

$$\begin{aligned} G_k &= ((Z_k)_L^{-T} \quad A_k) \begin{pmatrix} Z_k^T \\ (A_k)_L^{-1} \end{pmatrix} G_k (Z_k \quad (A_k)_L^{-T}) \begin{pmatrix} (Z_k)_L^{-1} \\ A_k^T \end{pmatrix} \\ &= ((Z_k)_L^{-T} \quad A_k) \begin{pmatrix} Z_k^T G_k Z_k & Z_k^T G_k (A_k)_L^{-T} \\ (A_k)_L^{-1} G_k Z_k & (A_k)_L^{-1} G_k (A_k)_L^{-T} \end{pmatrix} \begin{pmatrix} (Z_k)_L^{-1} \\ A_k^T \end{pmatrix}. \end{aligned}$$

As is well known, the reduced Hessian approach is to neglect the cross terms, i.e.,

$$\begin{aligned} G_k &\simeq ((Z_k)_L^{-T} \quad A_k) \begin{pmatrix} Z_k^T G_k Z_k & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} (Z_k)_L^{-1} \\ A_k^T \end{pmatrix} \\ &= (Z_k)_L^{-T} (Z_k^T G_k Z_k)(Z_k)_L^{-1} = M_k. \end{aligned}$$

It chooses such a matrix M_k in the SQP method and uses B_k, an $(n-t)\times(n-t)$ matrix, to approximate $(Z_k^T G_k Z_k)$. Thus, RHSQP algorithms generate a search direction d_k at x_k by solving

$$\begin{array}{ll} \min & g_k^T d + \frac{1}{2} d^T (Z_k)_L^{-T} B_k (Z_k)_L^{-1} d \\ \text{s.t.} & c(x_k) + A_k^T d = 0. \end{array}$$

Note that (2.1) implies that (1.3) is also the Lagrange multiplier of the above quadratic program. The solution d_k may be expressed as

$$d_k = h_k + v_k, \tag{2.3}$$

where

$$h_k = -Z_k B_k^{-1} Z_k^T g_k \tag{2.4}$$

and

$$v_k = -(A_k)_L^{-T} c(x_k). \tag{2.5}$$

There are two widely used instantiations of the generalized inverses. One is based on a QR factorization and widely used in discussions of RHSQP methods, for example, [11]. In this instantiation, the inverse matrices are given, using Nocedal and Overton's notation, by

$$(A_k)_L^{-1} = (R_k^{-1} \quad 0) \begin{pmatrix} Y_k^T \\ Z_k^T \end{pmatrix}, \qquad (Z_k)_L^{-1} = Z_k^T, \tag{2.6}$$

where Y_k and Z_k are orthonormal matrices derived from a QR factorization of A_k,

$$A_k = (Y_k \quad Z_k) \begin{pmatrix} R_k \\ 0 \end{pmatrix}. \tag{2.7}$$

The other instantiation, which is more appropriate for large-scale problems, is based on an LU decomposition of A_k. Suppose $A_k^T = (A_B \quad A_N)$, where A_B is nonsingular. This instantiation in effect chooses

$$\begin{aligned} (A_k)_L^{-1} &= (A_B^{-T} \quad 0), \\ Z_k &= \begin{pmatrix} -A_B^{-1} A_N \\ I \end{pmatrix}, \\ (Z_k)_L^{-1} &= (0 \quad I), \end{aligned}$$

where an LU decomposition of A_B is actually computed rather than its inverse, and may take advantage of the sparsity of A_k for a large-scale problem. For some very large scale problems where LU is not applicable, iterative methods could be used to invert A_B and its transpose. The generalized basis matrices give the RHSQP methods great flexibility in dealing with large-scale problems.

The matrix B_k is to be updated using gradient difference information. Two ways of obtaining this information have been proposed. Consider the gradient difference of the Lagrangian function $\nabla_x L(x, \lambda_k) - \nabla_x L(x_k, \lambda_k)$. Its projection on the null space using a given basis matrix Z_k can be approximated by

$$\begin{aligned} & Z_k^T (\nabla_x L(x, \lambda_k) - \nabla_x L(x_k, \lambda_k)) \\ & \simeq Z_k^T G_k (Z_k \quad (A_k)_L^{-T}) \begin{pmatrix} (Z_k)_L^{-1} \\ A_k^T \end{pmatrix} (x - x_k) \\ & = Z_k^T G_k Z_k (Z_k)_L^{-1} (x - x_k) + Z_k^T G_k (A_k)_L^{-T} A_k^T (x - x_k). \end{aligned} \tag{2.8}$$

To apply a quasi-Newton method such that $B_k \simeq Z_k^T G_k Z_k$, it is ideal to choose x such that the second term of the last equation in (2.8) disappears. By using x_{k+1}, one would choose the component of $x_{k+1} - x_k$ along the null space of A_k from x_k; i.e., $x = x_k + \alpha_k h_k$ and

$$y_k = Z_k^T(\nabla_x L(x_k + \alpha_k h_k, \lambda_k) - \nabla_x L(x_k, \lambda_k)), \tag{2.9}$$

$$s_k = (Z_k)_L^{-1} \alpha_k h_k = (Z_k)_L^{-1}(x_{k+1} - x_k), \tag{2.10}$$

and then the quasi-Newton equation $y_k = B_{k+1}s_k \simeq Z_k^T G_k Z_k s_k$ is satisfied. We call this first update strategy the null space secant update strategy because it uses the exact reduced Hessian information along the null space of A_k. The drawback of this strategy is that it imposes a significant extra cost to evaluate y_k when gradient evaluations of f and c are expensive.

The step secant update strategy of the second update category uses

$$y_k = Z_k^T(\nabla_x L(x_{k+1}, \lambda_k) - \nabla_x L(x_k, \lambda_k)) \tag{2.11}$$

to update B_{k+1} and saves the extra computation of the gradients of the Lagrangian. However, such y_k may not provide accurate information on the derivatives of $L(x, \lambda)$ along the null space of the constraints because of the presence of the second term in (2.8). Thus, updates of B_{k+1} must be skipped at some iterations where the second terms are large. If we replace x by x_{k+1} in (2.8), the second term becomes $Z_k^T G_k v_k$. Because B_{k+1} is expected to approximate $Z_k^T \nabla_{xx}^2 L(x_k, \lambda_k) Z_k$, the update could in fact result in great loss in accuracy of B_{k+1} if the vertical component v_k is not small when B_{k+1} is updated. Nocedal and Overton [11] proposed a criterion which we refer to as the Nocedal–Overton update criterion and under which B_{k+1} is updated if and only if

$$\|v_k\|_2 \le \frac{\eta}{(k+1)^{1+\epsilon}} \|h_k\|_2, \tag{2.12}$$

where η and ϵ are positive constants; otherwise $B_{k+1} = B_k$. Actually, they use $\|s_k\|$ instead of $\|h_k\|$ in their criterion, but under their orthogonality assumption on Z_k, $\|s_k\| = \|h_k\|$. It can be seen that the larger k is, the more accurate the information on the reduced Hessian (2.12) must be when B_{k+1} is updated. In section 4, we show a set of similar criteria with milder conditions on update steps. To globalize the algorithm, $s_k^T y_k > 0$ has to be tested to ensure that the positive definiteness of B_k is inherited.

In this section, a new update criterion is introduced to improve the numerical performance of the Nocedal–Overton update criterion, and general RHSQP algorithms are described using two merit functions.

Numerical experiments with the step secant update strategy show that the Nocedal–Overton update criterion often skips the updates in a large proportion of the cases. It appears that the criterion (2.12), which depends on the iteration number, is too strong, forcing updates to be skipped and sometimes slowing down the convergence. The criterion (2.12) may be relaxed by allowing updates whenever the horizontal component $\|h_k\|$ is not small compared to the vertical component $\|v_k\|$. A new update criterion is thus proposed, which not only allows more updates but also automatically guarantees the positive definiteness of $\{B_k\}$.

Positive Curvature Criterion. For constants $\zeta_1 \ge \zeta_2 > 0$, the update criterion requires

$$s_k^T y_k > \zeta_1 \|\alpha_k v_k\|^2 \quad \forall k \in S_1, \tag{2.13}$$

$$s_k^T y_k \le \zeta_2 \|\alpha_k v_k\|^2 \quad \forall k \in S_2. \tag{2.14}$$

If $\zeta_2 < \zeta_1$, these conditions leave an intermediate case where neither equation is imposed, giving the algorithm flexibility in deciding whether to update. This new criterion is referred to as the "positive curvature update criterion" for simplicity because (2.13) implies that the Lagrangian has a significantly positive curvature, which makes B_{k+1} automatically inherit the positive definiteness from B_k. Lemma 4.7, proved later, shows that this criterion satisfies $\|v_k\| \le \gamma_8 \|h_k\|$ whenever B_{k+1} is updated, and $\|h_k\| \le \gamma_8 \|v_k\|$ whenever an update is skipped, where $\gamma_8 > 0$ is a constant. Intuitively, it allows more updates than the Nocedal–Overton update criterion, and our numerical experiments in section 5 support this. Its numerical performance is so good that the step secant update strategy with the positive curvature criterion is very competitive with the null space secant update strategy.

Because the global and R-linear convergence of the null space secant update strategy has been proved by Byrd and Nocedal [2], we consider only the step secant update strategy in this paper. In the following description of the algorithm, $\varphi(x)$ stands for the merit function and $D\varphi(x; d)$ denotes the directional derivative of φ along d at x.

ALGORITHM 2.1.

```
The constants $\eta \in (0, \frac{1}{2})$ and $\tau$, $\tau'$ with $0 < \tau < \tau' < 1$ are given.
Let $x_1$ and $B_1$ be an initial point and an initial positive definite
matrix.
  1. Compute $d_k = h_k + v_k$ by solving (2.4) and (2.5).
  2. Adjust the merit function $\varphi$ according to $x_k$ if it is necessary.
  3. Set $\alpha_k = 1$ and check the line search condition
         $\varphi(x_k + \alpha_k d_k) \le \varphi(x_k) + \alpha_k \eta D\varphi(x_k; d_k).$
     If it is violated, choose a new $\alpha_k \in [\tau\alpha_k, \tau'\alpha_k]$ and check it
     again.
  4. Set $x_{k+1} = x_k + \alpha_k d_k$.
  5. Compute $s_k$ by (2.10) and $y_k$ by (2.11). Update $B_{k+1}$ by
         $B_{k+1} = \begin{cases} B^{BFGS}(B_k, s_k, y_k) & \text{if a criterion holds,} \\ B_k & \text{otherwise.} \end{cases}$
  6. If a stopping condition is not satisfied, set $k = k + 1$ and go to
     step 1; otherwise, stop. □
```

In analyzing this algorithm, in order to study the entire sequence, we do not impose a stopping condition. In our numerical tests, we use the stopping condition, $\|Z_k^T g_k\| + \|c_k\| < \varepsilon$, where $\varepsilon > 0$ is a given constant.

There are two widely used merit functions, the l_1 and Fletcher merit functions. Han [8] first introduced the l_1 merit function as

$$\phi_\mu(x) = f(x) + \mu \|c(x)\|_1,$$

where μ is called a penalty parameter. The l_1 merit function allows a strong global analysis. However, its penalty term is nondifferentiable and this nondifferentiability may affect the speed of convergence (the Maratos effect). Still, a directional derivative exists and, as is shown in Lemma 3.3 of [2],

$$D\phi_\mu(x_k; d_k) \le g_k^T h_k - (\mu - \|\lambda_k\|)\|c_k\|_1, \tag{2.15}$$

using the fact that $g_k^T v_k = -\lambda_k^T c_k$. The Fletcher merit function [4] is a differentiable merit function:

$$\Phi_\nu(x) = f(x) + \lambda(x)^T c(x) + \frac{1}{2}\nu \|c(x)\|_2^2,$$

where $\lambda(x)$ is a Lagrange multiplier estimate at x having the form of (1.3) and ν is a penalty parameter. Note that at x_k, the directional derivative is

$$\nabla\Phi_\nu(x_k)^T d_k = g_k^T h_k + g_k^T v_k + \lambda_k^T A_k^T d_k + c_k^T \nabla\lambda(x_k)^T d_k + \nu c_k^T A_k^T d_k.$$

From (1.3) and (2.5), $A_k^T d_k = -c_k$ and $g_k^T v_k = -g_k^T (A_k)_L^{-T} c_k = -\lambda_k^T c_k$. Thus

$$\nabla\Phi_\nu(x_k)^T d_k = g_k^T h_k + c_k^T \nabla\lambda(x_k)^T d_k - \nu c_k^T c_k. \tag{2.16}$$

With these merit functions, we can explicitly define step 2 of Algorithm 2.1, i.e., how to choose the penalty parameters μ and ν. In our global and R-linear convergence analysis, it is assumed that the penalty parameters μ_k and ν_k are monotonically increasing. The following adjusting procedure of these penalty parameters is simply called step $2'$ of Algorithm 2.1: for the l_1 merit function, the penalty parameter μ_k is chosen by

$$\mu_{k+1} = \begin{cases} \|\lambda_k\|_\infty + 2\rho & \text{if} \quad \mu_k < \|\lambda_k\|_\infty + \rho, \\ \mu_k & \text{otherwise,} \end{cases} \tag{2.17}$$

which, by (2.15), clearly implies that d_k is a descent direction. For the Fletcher merit function, the penalty parameter ν_k is chosen by

$$\nu_{k+1} = \begin{cases} \bar{\nu}_k + 2\rho & \text{if} \quad \nu_k < \bar{\nu}_k + \rho, \\ \nu_k & \text{otherwise,} \end{cases} \tag{2.18}$$

where $\bar{\nu}_k$ is defined by

$$\bar{\nu}_k = \frac{d_k^T \nabla\lambda(x_k) c_k + \frac{1}{2} g_k^T h_k}{\|c_k\|^2}, \tag{2.19}$$

where $\rho > 0$ is a constant. It is shown in Lemma 3.6 of [2] that $\bar{\nu}_k$ is bounded above by $O(\frac{s_k^T s_k}{s_k^T B_k s_k})$ and thus $\{\bar{\nu}_k\}$ is bounded if $\{\|B_k^{-1}\|\}$ is. Although we cannot bound $\{\|B_k^{-1}\|\}$ prior to our global convergence analysis, this fact indicates that boundedness of $\{\bar{\nu}_k\}$ is at least a reasonable assumption. If (2.18) is imposed, it follows easily from (2.16) that, as shown in [2],

$$\nabla\Phi_{\nu_k}(x_k)^T d_k \le -\frac{1}{2} g_k^T h_k - \rho\|c_k\|^2. \tag{2.20}$$

Since by (2.15) and (2.20), d_k is a descent direction for either merit function, it follows that step 3 of Algorithm 2.1 will terminate in a finite number of iterations. There are also some nonmonotonically increasing strategies which are widely used, e.g., $\mu_{k+1} = \|\lambda_k\| + 2\rho$. Although nonmonotonically increasing procedures numerically perform better than the monotonically increasing strategies in many numerical tests, there is no global and R-linear analysis established.

3. Global convergence. Our global convergence analysis of RHSQP algorithms using the step secant update strategy is based on the following assumptions.

ASSUMPTION 3.1.

1. *$f : R^n \to R^1$ and $c : R^n \to R^t$ and their first- and second-order derivatives are uniformly bounded in a closed set $D \subset R^n$, which contains $\{x_k\}$.*

2. *The matrix* $A(x)$ *is full rank for all* $x \in D$ *and there are constants* $\gamma_A > 0$ *and* $\gamma_Z > 0$ *such that for all* $x \in D$,

$$\begin{aligned} \|A(x)\| &\le \gamma_A, &\qquad \|A(x)_L^{-1}\| &\le \gamma_A, \\ \|Z(x)\| &\le \gamma_Z, &\qquad \|Z(x)_L^{-1}\| &\le \gamma_Z, \\ A(x)^T Z(x) &= 0, &\qquad Z(x)_L^{-1} A(x)_L^{-T} &= 0. \end{aligned}$$

3. *For a given* μ *or* ν, $\phi_\mu(x)$ *and* $\Phi_\nu(x)$ *are bounded below.*

We also assume that there are $m > 0$ and $M > 0$ such that

$$\frac{s_k^T y_k}{s_k^T s_k} \ge m, \tag{3.1}$$

$$\frac{y_k^T y_k}{s_k^T y_k} \le M \tag{3.2}$$

for the global analysis. For unconstrained problems, a proper line search strategy and the convexity of the objective function imply (3.1) and (3.2) [13]. However, for constrained problems, (3.1) and (3.2) are harder to guarantee. The following lemma shows that uniform positive definiteness of the reduced Hessian on the null space of $A(x)^T$, which holds locally around a solution x^* satisfying second-order sufficiency conditions, implies (3.1) and (3.2) for constrained problems, provided the step secant update strategy satisfies certain conditions. These conditions are satisfied by the positive curvature criterion as well as by the Nocedal–Overton update criterion.

LEMMA 3.1. *Suppose an RHSQP algorithm uses the step secant update strategy in such a way that* (2.13) *is satisfied or* $\|v_k\|/\|h_k\|$ *is sufficiently small for* $k \in S_1$ *large enough. Let* D *be a closed convex set containing* $\{x_k\}_{k=K_0}^{\infty}$ *for some* K_0. *Assume the following:*

1. *The second-order sufficient conditions hold on* D,

$$m_0\|u\|^2 \le u^T \nabla_{xx}^2 L(x,\lambda_k) u \qquad \forall u \in R^n : \quad A_k^T u = 0$$

for all $x \in D$, *any integer* $k \ge K_0$, *and some constant* $m_0 > 0$.

2. *For some constant* $M_0 > 0$ *and any* $x \in D$ *and* $k \ge K_0$,

$$\|\nabla_{xx}^2 L(x,\lambda_k)\| \le M_0.$$

Then there are constants $m > 0$ *and* $M > 0$ *such that* (3.1) *and* (3.2) *hold whenever* B_{k+1} *is updated with* $k > K_0$.

Proof. First consider a criterion satisfying (2.13). Consider two cases.

Case 1. $2\gamma_Z M_0\|\alpha_k v_k\| \le m_0\|s_k\|$: By (2.10), the Taylor expansion of y_k, and the inequalities $\|Z_k s_k\| \le \gamma_Z\|s_k\|$ and $\|s_k\| = \|(Z_k)_L^{-1} Z_k s_k\| \le \gamma_Z\|Z_k s_k\|$, the hypotheses of this lemma imply

$$\begin{aligned} s_k^T y_k &= s_k^T Z_k^T \nabla_{xx}^2 L(x_k + \xi d_k, \lambda_k)(\alpha_k h_k + \alpha_k v_k) \\ &= s_k^T Z_k^T \nabla_{xx}^2 L(x_k + \xi d_k, \lambda_k) Z_k s_k + s_k^T Z_k^T \nabla_{xx}^2 L(x_k + \xi d_k, \lambda_k)(\alpha_k v_k) \\ &\ge m_0\|Z_k s_k\|^2 - M_0\|Z_k s_k\|\|\alpha_k v_k\| \ge \frac{1}{2}\frac{m_0}{\gamma_Z^2}\|s_k\|^2. \end{aligned}$$

Since $\alpha_k h_k = Z_k s_k$, we have

$$\begin{aligned} \frac{y_k^T y_k}{s_k^T y_k} &= \|Z_k^T \nabla_{xx}^2 L(x_k + \xi d_k, \lambda_k)(Z_k s_k + \alpha_k v_k)\|^2/(s_k^T y_k) \\ &\le (\gamma_Z M_0)^2 \frac{(\gamma_Z\|s_k\| + \|\alpha_k v_k\|)^2}{\frac{m_0}{2\gamma_Z^2}\|s_k\|^2} \le 2\frac{\gamma_Z^4 M_0^2}{m_0}\left(\gamma_Z + \frac{m_0}{2\gamma_Z M_0}\right)^2. \end{aligned}$$

Case 2. $2\gamma_Z M_0\|\alpha_k v_k\| > m_0\|s_k\|$: Since the update criterion satisfies (2.13),

$$s_k^T y_k \geq \zeta_1\|\alpha_k v_k\|^2 \geq \zeta_1\left[\frac{m_0}{2\gamma_Z M_0}\right]^2\|s_k\|^2,$$

and

$$\begin{aligned}\frac{y_k^T y_k}{s_k^T y_k} &\leq (\gamma_Z M_0)^2\frac{(\gamma_Z\|s_k\| + \|\alpha_k v_k\|)^2}{\zeta_1\|\alpha_k v_k\|^2}\\ &\leq \frac{(\gamma_Z M_0)^2}{\zeta_1}\left(2\frac{\gamma_Z M_0}{m_0} + 1\right)^2.\end{aligned}$$

Therefore, there exist $m > 0$ and $M > 0$ such that (3.1) and (3.2) hold.

If we consider the criterion that $\|v_k\|/\|h_k\|$ is sufficiently small, the analysis is identical to Case 1 above. □

Under the assumption that (3.1) and (3.2) hold, global convergence is proved in the following. First we define the quantities

$$\cos\hat{\theta}_k = \frac{s_k^T B_k s_k}{\|s_k\|\,\|B_k s_k\|} \qquad \text{and} \qquad \hat{q}_k = \frac{s_k^T B_k s_k}{s_k^T s_k}.$$

For these quantities, the following theorem holds if Assumption 3.1 is satisfied.

THEOREM 3.2. *Let $\{B_k\}_{k\in S_1}$ be generated by the BFGS method. Suppose* (3.1) *and* (3.2) *hold for any $s_k \neq 0$. Then for any $p \in (0, 1]$, there exist constants $\beta_1 > 0$, $\beta_2 > 0$, and $\beta_3 > 0$ such that for any k, the relations*

$$\begin{aligned}\cos\hat{\theta}_i &\geq \beta_1 > 0,\\ 0 < \beta_2 &\leq \hat{q}_i \leq \beta_3,\\ \beta_2 &\leq \frac{\|B_i s_i\|}{\|s_i\|} \leq \frac{\beta_3}{\beta_1}\end{aligned}$$

hold for at least $\lceil p|S_1^k|\rceil$ values of $i \in S_1^k$. In other words, the index set J_k in which for any i the above three inequalities hold has at least $\lceil p|S_1^k|\rceil$ elements, i.e., $|J_k| \geq \lceil p|S_1^k|\rceil$.

Theorem 3.2 can be proved by applying the analysis of Theorem 3.1 of Byrd and Nocedal [1] to the subsequence $\{S_1^k\}$; the proof is omitted. The following two theorems describe the behaviors of the two merit functions, the l_1 and Fletcher merit functions.

THEOREM 3.3. *Suppose $\{x_k\}$ is generated by an RHSQP algorithm using the l_1 merit function with its penalty parameter chosen so that*

$$\mu_k \geq \|\lambda_k\|_\infty + \rho \tag{3.3}$$

for all k, where ρ is a positive constant. Then for all k,

$$D\phi_{\mu_k}(x_k; d_k) \leq -\frac{1}{\gamma_Z}\|Z_k^T g_k\|\|h_k\|\cos\hat{\theta}_k - \rho\|c_k\|_1. \tag{3.4}$$

In addition, for given constants, $\beta_1 > 0$, $\beta_2 > 0$, and $\beta_3 > 0$, there is a constant $\hat{\gamma} > 0$ such that if the conditions

$$\cos\hat{\theta}_k \geq \beta_1 > 0, \tag{3.5}$$

$$0 < \beta_2 \leq \hat{q}_k \leq \beta_3 \tag{3.6}$$

hold for some k, the directional derivative at x_k satisfies

$$D\phi_{\mu_k}(x_k; d_k) \le -\gamma_D[\|Z_k^T g_k\|^2 + \|c_k\|_1]. \tag{3.7}$$

Moreover, for any value μ, there is a positive constant γ_μ such that if $\mu_k = \mu$ satisfies (3.3) *and if* (3.5) *and* (3.6) *hold, then*

$$\phi_{\mu_k}(x_k) - \phi_{\mu_k}(x_{k+1}) \ge \gamma_\mu[\|Z_k^T g_k\|^2 + \|c_k\|_1]. \tag{3.8}$$

Proof. The proof of this theorem is similar to that for Lemma 3.3 in [2], but it handles the more general basis $Z(x)$ satisfying Assumption 3.1. The main difference is in (3.4), and we prove it as follows.

By the definition of ϕ_{μ_k},

$$D\phi_{\mu_k}(x_k; d_k) = g_k^T d_k - \mu_k\|c_k\|_1,$$

as shown in [2]. Since $v_k^T g_k = c_k^T \lambda_k$, (2.3) and (2.4) imply

$$\begin{aligned} D\phi_{\mu_k}(x_k; d_k) &\le g_k^T h_k - (\mu_k - \|\lambda_k\|_\infty)\|c_k\|_1 \le -g_k^T Z_k B_k^{-1} Z_k^T g_k - \rho\|c_k\|_1 \\ &= -\cos\hat{\theta}_k \|B_k^{-1} Z_k^T g_k\| \|Z_k^T g_k\| - \rho\|c_k\|_1. \end{aligned}$$

By (2.4) and Assumption 3.1, $\|h_k\| \le \gamma_Z\|B_k^{-1} Z_k^T g_k\|$, and then (3.4) holds. The remainder of this theorem can be proved using the same analysis as [2] and considering the factor γ_Z. □

A corresponding result can be proved for the Fletcher merit function.

THEOREM 3.4. *Suppose $\{x_k\}$ are generated by an RHSQP algorithm using the Fletcher merit function with the penalty parameter chosen so that*

$$\nu_k \ge \frac{d_k^T \nabla\lambda(x_k)c_k + \frac{1}{2}g_k^T h_k}{\|c_k\|^2} + \rho = \bar{\nu}_k + \rho \tag{3.9}$$

for $k > 0$ and some positive constant $\rho > 0$. Then for all $k \ge 0$,

$$D\Phi_{\nu_k}(x_k; d_k) \le -\frac{1}{\gamma_Z}\|Z_k^T g_k\|\|h_k\|\cos\hat{\theta}_k - \rho\|c_k\|^2. \tag{3.10}$$

In addition, for given constants, $\beta_1 > 0$, $\beta_2 > 0$, and $\beta_3 > 0$, there is a constant $\hat{\gamma} > 0$ such that if the conditions (3.5) *and* (3.6) *hold for some k, the directional derivative at x_k satisfies*

$$D\Phi_{\nu_k}(x_k; d_k) \le -\gamma_D[\|Z_k^T g_k\|^2 + \|c_k\|^2]. \tag{3.11}$$

Moreover, for any value ν, there is a positive constant γ_ν such that if $\nu_k = \nu$ satisfies (3.9) *and if* (3.5) *and* (3.6) *hold, then*

$$\Phi_{\nu_k}(x_k) - \Phi_{\nu_k}(x_{k+1}) \ge \gamma_\nu[\|Z_k^T g_k\|^2 + \|c_k\|^2]. \tag{3.12}$$

Proof. The proof is analogous to the previous analysis by considering the general basis matrices and using the directional derivative

$$\nabla\Phi_{\nu_k}(x_k)^T d_k = g_k^T h_k + d_k^T \nabla\lambda_k c_k + \nu_k\|c_k\|^2. \qquad \square$$

Based on the above two theorems about the two merit functions, the global convergence of RHSQP algorithms using the step secant update strategy is proved.

THEOREM 3.5. *Suppose $\{x_k\}$ is generated by an RHSQP algorithm using the step secant update strategy with any update criteria and using the l_1 and Fletcher merit functions with step 2 in Algorithm 2.1 replaced by step 2$'$. Suppose Assumption 3.1 and (3.1) and (3.2) are satisfied for all k sufficiently large. For the Fletcher merit function, $\bar{\nu}_k$ is assumed to be bounded above. Then*

$$\lim_{k\to\infty} \inf_{i\le k} \{\|Z_i^T g_i\| + \|c_i\|\} = 0.$$

Proof. If the l_1 merit function is used, it follows that $\mu_k \equiv \mu$ for some constant $\mu > 0$ and for sufficiently large k because μ_k is chosen by (2.17) and $\|\lambda(x)\|$ is bounded above. Similarly, if the Fletcher merit function is used, $\nu_k \equiv \nu$ for some constant ν and for k sufficiently large because $\bar{\nu}_k$ is assumed to be bounded above. Without loss of generality, we assume for any k, $\mu_k = \mu$ and $\nu_k = \nu$.

Suppose $|S_1| = \infty$. Since (3.1) and (3.2) hold for large k, by Theorem 3.2, for any $p \in (0,1)$, there are constants β_1, β_2, β_3, and index sets J_k with $|J_k| \ge p|S_1^k|$ for all k, such that for any $j \in J_k$, (3.5) and (3.6) hold. Theorems 3.3 and 3.4 imply

$$\phi_\mu(x_0) - \phi_\mu(x_k) \ge \gamma_\mu \sum_{j\in J_k} [\|Z_j^T g_j\|^2 + \|c_j\|_1],$$
$$\Phi_\nu(x_0) - \Phi_\nu(x_k) \ge \gamma_\nu \sum_{j\in J_k} [\|Z_j^T g_j\|^2 + \|c_j\|^2],$$

as both $\{\phi_\mu(x_k)\}$ and $\{\Phi_\nu(x_k)\}$ are decreasing sequences. Then

$$\sum_{j\in J_k} [\|Z_j^T g_j\|^2 + \|c_j\|_1] \le \phi_\mu(x_0) - \min_x \phi_\mu(x) < \infty,$$
$$\sum_{j\in J_k} [\|Z_j^T g_j\|^2 + \|c_j\|^2] \le \Phi_\nu(x_0) - \min_x \Phi_\nu(x) < \infty,$$

since the merit functions are bounded below for fixed penalty parameters by Assumption 3.1. Because $|J_k| \ge p|S_1^k| \to \infty$ as $k \to \infty$,

$$\lim_{j\in J_k\to\infty} \|Z_j^T g_j\|^2 + \|c_j\|_1 = 0,$$
$$\lim_{j\in J_k\to\infty} \|Z_j^T g_j\|^2 + \|c_j\|^2 = 0.$$

Since the l_1 and l_2 norms are equivalent, the conclusion of the theorem follows for this case.

If $|S_1|$ is finite, there is a K_1 large enough so that for any $k > K_1$, $B_k \equiv B_{K_1}$ and thus for all $k \ge K_1$, (3.5) and (3.6) hold for some constants $\beta_1 > 0$, $\beta_2 > 0$, and $\beta_3 > 0$. Similarly, by Theorems 3.3 and 3.4, we know that there are constants $\gamma_\mu > 0$ and $\gamma_\nu > 0$ such that for any $k > K_1$

$$\phi_\mu(x_{K_1}) - \phi_\mu(x_k) \ge \gamma_\mu \sum_{j=K_1}^{k} [\|Z_j^T g_j\|^2 + \|c_j\|_1],$$
$$\Phi_\nu(x_{K_1}) - \Phi_\nu(x_k) \ge \gamma_\nu \sum_{j=K_1}^{k} [\|Z_j^T g_j\|^2 + \|c_j\|^2].$$

These two inequalities imply that

$$\lim_{k\to\infty} [\|Z_k^T g_k\| + \|c_k\|] = 0$$

when S_1 is finite. □

Note that the convergence result for the Fletcher merit function is somewhat weaker than for the l_1 merit function because of the plausible but optimistic assumption on $\{\bar{\nu}_k\}$.

With global convergence now established, in the next section we discuss the R-linear convergence of the step secant update strategy.

4. Local and R-linear convergence.

In this section R-linear convergence is proved for the RHSQP algorithms using the step secant update with the Nocedal–Overton update strategy and the positive curvature criterion, along with either the l_1 merit function or the Fletcher merit function. Although the positive curvature criterion allows more updates than the Nocedal–Overton update criterion and we prove that they are both R-linear convergent, we cannot establish a single unified analysis for them. In this section, we present the analysis of R-linear convergence of the two criteria separately because of their different update characters.

4.1. Properties of the local minimizer.

Before the analysis of the R-linear convergence, some characteristics of the solution of (1.1) are shown under the following assumption.

ASSUMPTION 4.1. *Let x^* be a local minimizer of* (1.1).

1. *Assumption* 3.1 *holds on a set D containing x^* in its interior.*
2. *The matrix $A(x^*)$ is full rank. This implies that x^* is a Kuhn–Tucker point. That is, there is a Lagrange multiplier vector, $\lambda^* \in R^t$, such that*

$$\nabla_x L(x^*, \lambda^*) = g(x^*) + A(x^*)\lambda^* = 0. \tag{4.1}$$

3. *The matrix $Z(x^*)^T \nabla^2_{xx} L(x^*, \lambda^*) Z(x^*)$ is positive definite.*
4. *In a neighborhood of x^* the functions $\lambda(x)$ and $Z(x)$ are Lipschitz continuous; i.e.,*

$$\|\lambda(x) - \lambda(z)\| \le \gamma_\lambda \|x - z\|, \tag{4.2}$$

$$\|Z(x) - Z(z)\| \le \gamma_z \|x - z\|, \tag{4.3}$$

where γ_λ and γ_z are constants. Locally, there exist constants γ_Z and gamma$_A$ such that inequalities in Assumption 3.1 *hold.*

Assumption 4.1 implies that for any (x, λ) sufficiently close to (x^*, λ^*) and $\delta > 0$ sufficiently small,

$$m_0 \|u\|^2 \le u^T Z(x)^T \nabla^2_{xx} L(x + \Delta x, \lambda) Z(x) u \le M_0 \|u\|^2 \tag{4.4}$$

for some constants $m_0 > 0$ and $M_0 > 0$ with $\|\Delta x\| \le \delta$. That means that the assumptions of Lemma 3.1 are satisfied, and thus (3.1) and (3.2) hold near (x^*, λ^*).

Under Assumption 4.1, the following lemma similar to Lemmas 4.1 and 4.2 given by Byrd and Nocedal [2] can be proved under mild conditions.

LEMMA 4.1. *If Assumption* 4.1 *holds, then for x sufficiently close to x^*,*

$$\gamma_1 \|x - x^*\| \le \|c(x)\| + \|Z(x)^T g(x)\| \le \gamma_2 \|x - x^*\| \tag{4.5}$$

for some constants $\gamma_1 > 0$ and $\gamma_2 > 0$. In addition, for any $\mu > \|\lambda^*\|_\infty$ and for any ν sufficiently large, there are constants $\gamma_3 > 0$, $\gamma_4 > 0$, and $\gamma_5 > 0$, $\gamma_6 > 0$ such that

$$\gamma_3 \|x - x^*\|^2 \le \phi_\mu(x) - \phi_\mu(x^*) \le \gamma_4 [\|Z(x)^T g(x)\|^2 + \|c(x)\|_1], \tag{4.6}$$

$$\gamma_5 \|x - x^*\|^2 \le \Phi_\nu(x) - \Phi_\nu(x^*) \le \gamma_6 [\|Z(x)^T g(x)\|^2 + \|c(x)\|^2]. \tag{4.7}$$

Proof. By using (2.10) and (3.1) for the general matrix functions, $Z(x)$, $(Z(x))_L^{-1}$, and $A(x)_L^{-1}$, the inequality (4.5) follows from the analysis of Lemma 4.1 in [2] because there are no derivatives higher than second order involved. If (4.7) holds, (4.6) follows, using the same technique as in Lemma 4.2 in [2]. The analysis in [2] involves the third-order derivatives only in the proof of (4.7) itself.

Let us consider (4.7). Since (2.2) also holds on x^*, we can express $x - x^* = h + v$, where $h = Z^*(Z^*)_L^{-1}(x - x^*)$ and $v = A^*{}_L^{-T}A^{*T}(x - x^*)$. Because $\Phi_\nu(x^*) = L(x^*, \lambda^*)$ and $\nabla_x L(x^*, \lambda^*) = 0$, it follows from Taylor's theorem applied to L and from (4.4), (4.2), and (4.3) that

$$\begin{aligned}
\Phi_\nu(x) - \Phi_\nu(x^*) &= L(x, \lambda(x)) - L(x^*, \lambda^*) + \frac{\nu}{2}\|c(x)\|^2 \\
&\geq \frac{1}{2}(x - x^*)^T \nabla^2_{xx} L(x^*, \lambda^*)(x - x^*) + (\lambda(x) - \lambda^*)^T c(x) \\
&\quad + o(\|x - x^*\|^2) + \frac{\nu}{2}\|c(x)\|^2 \\
&= \frac{1}{2}(h^T \nabla^2_{xx} L(x^*, \lambda^*)h + 2h^T \nabla^2_{xx} L(x^*, \lambda^*)v + v^T \nabla^2_{xx} L(x^*, \lambda^*)v) \\
&\quad + (\lambda(x) - \lambda^*)^T c(x) + o(\|x - x^*\|^2) + \frac{\nu}{2}\|c(x)\|^2 \\
&\geq \frac{1}{2} m_0 \|h\|^2 - M_0\|h\|\|v\| - \frac{1}{2}M_0\|v\|^2 \\
&\quad - \gamma_\lambda \|x - x^*\|\|c(x)\| + \frac{\nu}{2}\|c(x)\|^2 + o(\|x - x^*\|^2).
\end{aligned}$$

Since $c(x) - c(x^*) = A^{*T}(x - x^*) + O(\|x - x^*\|^2)$ and $A^*{}_L^{-1}$ is bounded, it follows that $\|v\| \leq \gamma_A \|c(x)\| + O(\|x - x^*\|^2)$. Thus,

$$\begin{aligned}
\Phi_\nu(x) - \Phi_\nu(x^*) &\geq -\gamma_\lambda(\|h\| + \gamma_A\|c(x)\|)\|c(x)\| + \frac{1}{2}m_0\|h\|^2 - M_0\gamma_A\|h\|\|c(x)\| \\
&\quad - \frac{1}{2}M_0\gamma_A\|c(x)\|^2 + \frac{\nu}{2}\|c(x)\|^2 + o(\|x - x^*\|^2) \\
&= \frac{1}{2}m_0\|h\|^2 + \left(-\gamma_\lambda\gamma_A - \frac{1}{2}M_0\gamma_A + \frac{\nu}{2}\right)\|c\|^2 \\
&\quad + (-\gamma_\lambda - M_0\gamma_A)\|h\|\|c\| + o(\|x - x^*\|^2).
\end{aligned}$$

Consider the above equation as a quadratic polynomial in $\|h\|$ and $\|c\|$. There are positive constants $\bar{\nu}$, γ', and γ_5 such that if $\nu > \bar{\nu}$,

$$\Phi_\nu(x) - \Phi_\nu(x^*) \geq \gamma'(\|h\|^2 + \|v\|^2) + o(\|x - x^*\|^2) \geq \gamma_5\|x - x^*\|^2.$$

Similarly, using the Lipschitz continuity of $\lambda(x)$, $\nabla_x L(x^*, \lambda^*) = 0$ and (4.5),

$$\begin{aligned}
\Phi_\nu(x) - \Phi_\nu(x^*) &= L(x, \lambda(x)) - L(x^*, \lambda^*) + \frac{\nu}{2}\|c(x)\|^2 \\
&\leq \gamma_\lambda\|x - x^*\|\|c(x)\| + \frac{M_0}{2}\|x - x^*\|^2 \\
&\quad + o(\|x - x^*\|^2) + \frac{\nu}{2}\|c(x)\|^2 \\
&\leq O(\|x - x^*\|^2) + \frac{\nu}{2}\|c(x)\|^2 \\
&\leq O(\|Z(x)^T g(x)\| + \|c(x)\|)^2 + \frac{\nu}{2}\|c(x)\|^2 \\
&\leq \gamma_6(\|Z(x)^T g(x)\|^2 + \|c(x)\|^2). \qquad \square
\end{aligned}$$

In order to guarantee that $\{x_k\}_{k=1}^{\infty}$ converges to x^*, another assumption is made for the constrained problem,

ASSUMPTION 4.2. *The line search procedure has the property that if x_k is sufficiently close to x^*, then $\forall\theta \in [0,1]$,*

$$\varphi((1-\theta)x_k + \theta x_{k+1}) \leq \varphi(x_k),$$

where φ is the merit function used in RHSQP algorithms.

Actually, there is no practical line search strategy that can absolutely guarantee Assumption 4.2 to be satisfied, but it seems unlikely that it is violated when x_k is close to x^*. It is clearly satisfied when φ is quasi-convex. The following theorem shows that Assumption 4.2 implies $\{x_k\} \to x^*$.

THEOREM 4.2. *Let $\{x_k\}$ be generated by an RHSQP algorithm using the l_1 merit function with μ_k chosen by* (2.17) *and using either the Nocedal–Overton criterion or the positive curvature criterion. Suppose Assumptions* 4.1 *and* 4.2 *hold and $\{\lambda_k\}$ is bounded above. Then for sufficiently large K, μ_k is fixed for $k > K$ and there is a neighborhood of x^* such that if an iterate x_{k_0} with $k_0 > K$ falls in the neighborhood, then $x_k \to x^*$ and* (3.1) *and* (3.2) *hold for all k sufficiently large. If the Fletcher merit function with ν_k chosen by* (2.18) *is used, the same conclusion holds under the additional assumption that $\bar{\nu}_k$ is bounded and ν_k is large enough.*

Proof. By Assumption 4.1, there exists $\delta_1 > 0$ such that, for all x in the neighborhood $N_1 = \{x : \|x - x^*\| < \delta_1\}$ of x^*,

$$\|\lambda(x)\|_\infty + \rho > \|\lambda^*\|_\infty, \tag{4.8}$$

and the conditions of Assumption 3.1 hold for $D = N_1$.

Now, since $\{\|\lambda(x_k)\|_\infty\}$ and $\{\bar{\nu}_k\}$ are bounded, the procedure (2.17) or (2.18) implies that for all k greater than some value $\bar{k}$, μ_k or ν_k are fixed at some values μ and ν. Suppose ν is sufficiently large that (4.7) holds. By (2.17), (2.18), and (4.8), if an iterate x_k, with $k > \bar{k}$, occurs in N_1 then it must be that $\mu > \|\lambda^*\|_\infty$. In other words, Lemma 4.1 holds on N_1, and ϕ_μ and Φ_ν have a strict local minimizer x^*. Suppose K is an integer such that $\mu_k = \mu$ or $\nu_k = \nu$ for any $k > K$. For such μ and ν, it follows from Lemma 4.1 that there exists $\delta_2 \in (0, \delta_1]$ such that if $\|x_{k_0} - x^*\| < \delta_2$ for $k_0 > K$, the connected component of the level set $\{z : \phi_\mu(z) < \phi_\mu(x_{k_0})\}$ or $\{z : \Phi_\nu(z) < \Phi_\nu(x_{k_0})\}$ containing x^* is a subset N_2 of N_1. Since N_2 is connected, by Assumption 4.2, all iterates x_k for $k > k_0$ remain in N_2. If δ_2 is chosen sufficiently small, then by Assumption 4.1 the hypotheses of Lemma 3.1 hold for $D = N_2$ and therefore (3.1) and (3.2) hold at all update steps for $k > k_0$. Then the assumptions of Theorem 3.5 are satisfied for $k > \hat{k}$ and thus there is a subsequence $\{x_{k_i}\}$ of $\{x_k\}_{k=k_0}^{\infty}$ such that

$$\lim_{i\to\infty} [\|Z_{k_i}^T g_{k_i}\| + \|c_{k_i}\|] = 0.$$

By (4.5), (4.6), and (4.7), we have

$$\lim_{i\to\infty} \phi_\mu(x_{k_i}) - \phi_\mu(x^*) = 0,$$
$$\lim_{i\to\infty} \Phi_\nu(x_{k_i}) - \Phi_\nu(x^*) = 0,$$

and by the monotone decreasing property of $\{\phi_\mu(x_k)\}$ or $\{\Phi_\nu(x_k)\}$,

$$\lim_{k\to\infty} \phi_\mu(x_k) - \phi_\mu(x^*) = 0,$$
$$\lim_{k\to\infty} \Phi_\nu(x_k) - \Phi_\nu(x^*) = 0.$$

By Lemma 4.1, x^* is a local minimizer of either merit function. So Assumption 4.1 implies $x_k \to x^*$. The neighborhood guaranteed by this theorem is thus N_2. □

Based on this theorem, we can show the R-linear convergence under the hypothesis $\{x_k\} \to x^*$ for the Nocedal–Overton criterion and the positive curvature criterion, respectively. First, we show that both criteria have an R-linear convergent subsequence, and the remaining subsequence is discussed separately for these two criteria.

4.2. A subsequential R-linear convergence. We now define a subsequential R-linear convergence. Given a subset $S \subset [1, \ldots, \infty)$, we refer to $\{x_k\}$ as S R-linear convergent if there exists $r < 1$ such that for all k, $\|x_k - x^*\| \le r^{|S^k|}$, where $S^k = S \bigcap [1, \ldots, k]$. We now show that both criteria generate an S_1 R-linear convergent sequence.

Lemma 4.3. *Suppose $\{x_k\}$ is generated by an RHSQP algorithm using the step secant update strategy with either the Nocedal–Overton criterion or the positive curvature criterion and using the l_1 merit function or the Fletcher merit function. Then $\{x_k\}$ converges S_1 R-linearly if the hypotheses of Theorem* 4.2 *are satisfied.*

Proof. Since $x_k \to x^*$ by Theorem 4.2, (3.1) and (3.2) hold, and Lemma 4.1 also holds for $x = x_k$, if k is large enough. Without loss of generality, assume that these lemmas hold for any k. Choose $p = \frac{1}{2}$ and apply Theorems 3.2, 3.3, and 3.4 to the RHSQP algorithm. Then for the index set J_k defined in Theorem 3.2,

$$\begin{aligned} \phi_\mu(x_i) - \phi_\mu(x_{i+1}) &\ge \gamma_\mu[\|Z_i^T g_i\|^2 + \|c_i\|_1] \qquad \forall i \in J_k, \\ \Phi_\nu(x_i) - \Phi_\nu(x_{i+1}) &\ge \gamma_\nu[\|Z_i^T g_i\|^2 + \|c_i\|^2]. \end{aligned}$$

By (4.6) or (4.7), the above inequalities imply

$$\begin{aligned} \phi_\mu(x_i) - \phi_\mu(x_{i+1}) &\ge \frac{\gamma_\mu}{\gamma_4}(\phi_\mu(x_i) - \phi_\mu(x^*)) \quad \forall i \in J_k, \\ \Phi_\nu(x_i) - \Phi_\nu(x_{i+1}) &\ge \frac{\gamma_\nu}{\gamma_6}(\Phi_\nu(x_i) - \Phi_\nu(x^*)) \quad \forall i \in J_k. \end{aligned}$$

Then

$$\begin{aligned} \phi_\mu(x_{i+1}) - \phi_\mu(x^*) &\le \left(1 - \frac{\gamma_\mu}{\gamma_4}\right)(\phi_\mu(x_i) - \phi_\mu(x^*)), \\ \Phi_\nu(x_{i+1}) - \Phi_\nu(x^*) &\le \left(1 - \frac{\gamma_\nu}{\gamma_6}\right)(\Phi_\nu(x_i) - \Phi_\nu(x^*)). \end{aligned}$$

Let $r' = (1 - \frac{\gamma_\mu}{\gamma_4})^{\frac{1}{4}} < 1$ for the l_1 merit function or $r' = (1 - \frac{\gamma_\nu}{\gamma_6})^{\frac{1}{4}} < 1$ for the Fletcher merit function and choose $r'' = \frac{1}{\gamma_3}(\phi_\mu(x_0) - \phi_\mu(x^*))^{\frac{1}{2}} > 0$ for the l_1 merit function and $r'' = \frac{1}{\gamma_5}(\Phi_\nu(x_0) - \Phi_\nu(x^*))^{\frac{1}{2}} > 0$ for the Fletcher merit function. Then for any $i \in J_k$,

$$\begin{aligned} \phi_\mu(x_{i+1}) - \phi_\mu(x^*) &\le {r'}^4(\phi_\mu(x_i) - \phi_\mu(x^*)), \\ \Phi_\nu(x_{i+1}) - \Phi_\nu(x^*) &\le {r'}^4(\Phi_\nu(x_i) - \Phi_\nu(x^*)), \end{aligned}$$

and by the decreasing properties of $\{\phi_\mu(x_i)\}$ along with (4.6),

$$\begin{aligned} \|x_k - x^*\| &\le \frac{1}{\gamma_3}(\phi_\mu(x_k) - \phi_\mu(x^*))^{\frac{1}{2}} \\ &\le \frac{1}{\gamma_3}({r'}^{4|J_k|}(\phi_\mu(x_0) - \phi_\mu(x^*)))^{\frac{1}{2}} \end{aligned}$$

$$\begin{aligned}&\leq \frac{1}{\gamma_3}(r'^{2|S_1^k|}(\phi_\mu(x_0) - \phi_\mu(x^*)))^{\frac{1}{2}}\\ &= r''r'^{|S_1^k|},\end{aligned}$$

because $p = \frac{1}{2}$ and $|J_k| \geq p|S_1^k|$. For the Fletcher merit function, this conclusion follows similarly. This implies that there is a constant $r \in (0, 1)$ such that

$$\|x_k - x^*\| \leq r^{|S_1^k|}$$

for both merit functions. □

Note that this analysis can be applied to any update that guarantees that (3.1) and (3.2) are satisfied. By the S_1 R-linear convergence, it follows that the entire sequence is R-linearly convergent if

- there is a constant $p > 0$ such that $|S_1^k| \geq pk$ for any k; or
- $|S_1|$ is finite (because B_k will be a fixed matrix for large k and the proof of Lemma 4.3 can be applied to S_2).

The difficult case is when neither of these holds and S_2 R-linear convergence has to be proved. Because of the differences in S_2 between the Nocedal–Overton update criterion and the positive curvature criterion, we prove their R-linear convergence separately.

4.3. The Nocedal–Overton criterion. For the Nocedal–Overton update criterion, one can prove that the matrices $\{B_k\}$ and $\{B_k^{-1}\}$ are bounded, and then it is not difficult to prove R-linear convergence. We need only show the boundedness of $\{B_k\}$ and $\{B_k^{-1}\}$ for the Nocedal–Overton update criterion.

Consider the scaled version of the matrix function $\psi(\cdot)$ developed by Byrd and Nocedal [1] for the quasi-Newton methods. The ψ function is defined as

$$\psi(B) = \mathrm{Tr}(H^{*-\frac{1}{2}}BH^{*-\frac{1}{2}}) - \ln\det(H^{*-\frac{1}{2}}BH^{*-\frac{1}{2}}), \tag{4.9}$$

where $H^* = Z^{*T}\nabla^2_{xx}L(x^*, \lambda^*)Z^* > 0$. In order to discuss the boundedness of B_k and B_k^{-1} using ψ, we define the quantities $\cos\theta_k$ and q_k, which are scaled versions of the quantities $\cos\hat{\theta}_k$ and $\hat{q}_k$, used for the global convergence analysis:

$$\cos\theta_k = \frac{s_k^T B_k s_k}{\|H^{*\frac{1}{2}}s_k\|\|H^{*-\frac{1}{2}}B_k s_k\|}, \qquad q_k = \frac{s_k^T B_k s_k}{s_k^T H^* s_k}. \tag{4.10}$$

Now we estimate $\psi(B_{k+1})$ by the following lemma.

LEMMA 4.4. *When x_k and x_{k+1} are close to x^* and $k \in S_1$,*

$$\psi(B_{k+1}) \leq \psi(B_k) - \frac{q_k}{\cos^2\theta_k} + \ln q_k + 1 + \tilde{\gamma} \quad \textit{and} \tag{4.11}$$

$$\psi(B_{k+1}) \leq \psi(B_k) - \frac{q_k}{\cos^2\theta_k} + \ln q_k + 1 + L_0\sigma_k + \tilde{\gamma}\omega_k, \tag{4.12}$$

where L_0 and $\tilde{\gamma}$ are constants, $\omega_k = \|\alpha_k c_k\|/\|s_k\|$, and $\sigma_k = \max\{\|e_{k+1}\|, \|e_k\|\}$ with $e_k = x_k - x^*$.

Proof. By a result of Pearson [12] for the BFGS update,

$$\det(H^{*-\frac{1}{2}}B_{k+1}H^{*-\frac{1}{2}}) = \det(H^{*-\frac{1}{2}}B_kH^{*-\frac{1}{2}})\frac{s_k^T y_k}{s_k^T B_k s_k}. \tag{4.13}$$

By the definition of ψ, (4.13), and (4.10),

$$\begin{aligned}(4.14)\qquad \psi(B_{k+1}) &= \mathrm{Tr}(B_k) - \mathrm{Tr}\left(H^{*-\frac{1}{2}}\left(\frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}\right) H^{*-\frac{1}{2}}\right)\\ &\quad - \ln\det(H^{*-\frac{1}{2}} B_k H^{*-\frac{1}{2}}) + \ln\frac{s_k^T B_k s_k}{s_k^T y_k}\\ &= \psi(B_k) - \frac{\|H^{*-\frac{1}{2}} B_k s_k\|^2}{s_k^T B_k s_k} + \frac{y_k^T H^{*-1} y_k}{s_k^T y_k} + \ln\frac{s_k^T B_k s_k}{s_k^T y_k}\\ &= \psi(B_k) - \frac{q_k}{\cos^2\theta_k} + \frac{y_k^T H^{*-1} y_k}{s_k^T y_k} - \ln\frac{s_k^T y_k}{s_k^T H^* s_k} + \ln q_k.\end{aligned}$$

Then by Assumption 4.1, the conditions (3.1) and (3.2) hold on S_1. Thus

$$\begin{aligned}\psi(B_{k+1}) &\le \psi(B_k) - \frac{q_k}{\cos^2\theta_k} + \ln q_k + 1\\ &\qquad + \left(\|H^{*-1}\|M - 1 - \ln\frac{m}{\|H^*\|}\right).\end{aligned}$$

That is, (4.11) holds. To prove (4.12), we need to estimate the third and fourth terms in the last equation of (4.14). Let

$$H_k = Z_k^T G_k Z_k, \quad \tilde{H}_k = Z_k^T G_k (A_k)_L^{-T},$$

and then by Taylor's theorem and (2.2),

$$\begin{aligned}(4.15)\qquad y_k &= Z_k^T[\nabla_x L(x_{k+1},\lambda_k) - \nabla_x L(x_k,\lambda_k)]\\ &= Z_k^T[G_k\alpha_k d_k] + O(\|\alpha_k d_k\|^2)\\ &= Z_k^T\left[G_k(Z_k(Z_k)_L^{-1} + (A_k)_L^{-T}A_k^T)\alpha_k d_k\right] + O(\|\alpha_k d_k\|^2)\\ &= H_k s_k - \tilde{H}_k\alpha_k c_k + O(\|\alpha_k d_k\|^2)\\ &= H^* s_k - \tilde{H}_k(\alpha_k c_k) + O(\|e_k\|\|s_k\| + \|\alpha_k d_k\|^2).\end{aligned}$$

To estimate the third term of (4.12), we multiply both sides of (4.15) by $y_k^T H^{*-1}$ and yield

$$\begin{aligned}y_k^T H^{*-1} y_k &= s_k^T y_k - y_k^T H^{*-1}\tilde{H}_k(\alpha_k c_k) + O(\|e_k\|\|s_k\| + \|\alpha_k d_k\|^2)\|y_k\|\\ &\le s_k^T y_k + O(\|\alpha_k c_k\|)\|y_k\| + O(\|e_k\|\|s_k\| + \|\alpha_k d_k\|^2)\|y_k\|.\end{aligned}$$

By (3.1) and (3.2), we have $s_k^T y_k \ge mM\|s_k\|\|y_k\|$ and then

$$\frac{y_k^T H^{*-1} y_k}{s_k^T y_k} \le 1 + O\left(\frac{\|\alpha_k c_k\|}{\|s_k\|}\right) + O(\|e_k\|) + O\left(\frac{\|\alpha_k d_k\|^2}{\|s_k\|}\right).$$

Since $\|\alpha_k d_k\| \le O(\sigma_k)$ and

$$\begin{aligned}\frac{\|\alpha_k d_k\|^2}{\|s_k\|} &= O(\sigma_k)\frac{\|\alpha_k d_k\|}{\|s_k\|} \le O(\sigma_k)\frac{\|\alpha_k h_k\| + \|\alpha_k v_k\|}{\sqrt{s_k^T y_k}}\\ &= O(\sigma_k) + O\left(\sigma_k\frac{\|\alpha_k c_k\|}{\|s_k\|}\right),\end{aligned}$$

it follows that

$$\frac{y_k^T H^{*-1} y_k}{s_k^T y_k} \le 1 + O\left(\frac{\|\alpha_k c_k\|}{\|s_k\|}\right) + O(\sigma_k). \tag{4.16}$$

Similarly, to estimate the fourth term, we multiply both sides of (4.15) by s_k^T,

$$\begin{aligned} s_k^T y_k &= s_k^T H^* s_k - s_k^T \tilde{H}_k(\alpha_k c_k) + O(\|e_k\|\|s_k\| + \|\alpha_k d_k\|^2)\|s_k\|, \\ \frac{s_k^T y_k}{s_k^T H^* s_k} &= 1 + \frac{1}{s_k^T H^* s_k}[-s_k^T \tilde{H}_k(\alpha_k c_k) + O(\|e_k\|\|s_k\| + \|\alpha_k d_k\|^2)\|s_k\|] \\ &= 1 + O\left(\frac{\|\alpha_k c_k\|}{\|s_k\|}\right) + O(\sigma_k). \end{aligned}$$

Since $\frac{s_k^T H^* s_k}{s_k^T y_k} \le \frac{\|H^*\|}{m}$, we have

$$\begin{aligned} -\ln \frac{s_k^T y_k}{s_k^T H^* s_k} &= \ln \frac{s_k^T H^* s_k}{s_k^T y_k} \\ &\le \frac{s_k^T H^* s_k}{s_k^T y_k} - 1 \\ &= \frac{s_k^T H^* s_k}{s_k^T y_k}\left(1 - \frac{s_k^T y_k}{s_k^T H^* s_k}\right) \\ &\le \frac{\|H^*\|}{m}\left|\frac{s_k^T y_k}{s_k^T H^* s_k} - 1\right| \\ &\le O\left(\frac{\|\alpha_k c_k\|}{\|s_k\|}\right) + O(\sigma_k). \end{aligned} \tag{4.17}$$

Using (4.16) and (4.17) and the definition of ω_k, we know that there exist constants L_0 and $\tilde{\gamma}$ satisfying (4.11) and (4.12). □

From Lemmas 4.3 and 4.4, it follows that for any update criterion satisfying

$$\sum_{j \in S_1} \frac{\|v_j\|}{\|h_j\|} < \infty, \tag{4.18}$$

the quasi-Newton matrices and their inverses are bounded below and above. This is because $\|\alpha_k c_k\|/\|s_k\| \le (\gamma_A/\gamma_Z)(\|v_k\|/\|h_k\|)$, and

$$\begin{aligned} \psi(B_k) &\le \psi(B_0) + L_0 \sum_{j \in S_1^{k-1}} \sigma_j + \tilde{\gamma} \sum_{j \in S_1^{k-1}} \left(\frac{\gamma_A}{\gamma_Z}\frac{\|v_j\|}{\|h_j\|}\right) \\ &\le \psi(B_0) + L_0 \sum_{j \in S_1^{k-1}} r^{|S_1^j|} + \tilde{\gamma} \sum_{j \in S_1^{k-1}} \left(\frac{\gamma_A}{\gamma_Z}\frac{\|v_j\|}{\|h_j\|}\right) < \infty, \end{aligned}$$

since $-q_k/\cos^2\theta_k + \ln q_k + 1 \le 0$. That is, there is a $\bar{\psi} > 0$ such that $\psi(B_k) \le \bar{\psi} < \infty$, which implies that $\|B_k\|$ and $\|B_k^{-1}\|$ are bounded, and we have the following theorem.

THEOREM 4.5. *Under the assumptions of Theorem* 4.2 *with either merit function, if an iterate lands close enough to x^*, then $x_k \to x^*$. In addition, if* (4.18) *holds, $\{\|B_k\|\}$ and $\{\|B_k^{-1}\|\}$ are bounded above and $\{x_k\}$ converges to x^* R-linearly.*

Proof. As shown above, $\{\|B_k\|\}_{k=1}^{\infty}$ and $\{\|B_k^{-1}\|\}_{k=1}^{\infty}$ are bounded. Therefore, (3.7) and (3.10) and thus (3.8) and (3.12) hold for all k. The R-linear convergence follows from the same argument as in the proof of Lemma 4.3, applied to the entire sequence. □

The Nocedal–Overton update criterion satisfies (4.18). Actually, there are other criteria satisfying (4.18); for example, (4.18) will hold if

$$\|v_k\| \leq \frac{\zeta}{|S_1^k|^{1+\epsilon}}\|h_k\| \tag{4.19}$$

whenever B_{k+1} is updated, where ζ and ϵ are positive constants.

COROLLARY 4.6. *Suppose the assumptions of Theorem* 4.5 *are satisfied, and either the Nocedal–Overton update criterion or the criterion given by* (4.19) *are used. If Assumptions* 4.1 *and* 4.2 *hold, then* $\{x_k\}_{k=1}^{\infty}$ *converges to* x^* *R-linearly.* □

4.4. R-linear convergence using the positive curvature criterion. Unlike the Nocedal–Overton update criterion, the positive curvature criterion allows updates even when $\|v_k\|/\|h_k\|$ is not small. The analysis of Lemma 4.4 cannot be applied. Without assuming that $\{B_k\}$ and $\{B_k^{-1}\}$ are bounded, the sufficient reductions of the merit functions in (3.4) and (3.10) cannot rely on the terms involving $\cos\theta_k$, because based on the current analysis, $\cos\theta_k$ cannot be proved to be bounded away from zero even though numerically it rarely happens that $\cos\theta_k$ tends to be unbounded. Fortunately, the positive curvature criterion guarantees that $\|c_k\|$ is relatively large compared to $\|h_k\|$ for any $k \in S_2$.

LEMMA 4.7. *If Algorithm* 2.1 *is used with the positive curvature criterion and the conditions of Theorem* 4.2 *are satisfied, then there are constants* $\gamma_8 > 0$ *and* $\gamma_9 > 0$ *such that for sufficiently large* k,

$$\begin{aligned}\|v_k\| &\leq \gamma_8\|h_k\|, \qquad k \in S_1,\\ \|h_k\| &\leq \gamma_9\|v_k\|, \qquad k \in S_2.\end{aligned}$$

Proof. By (2.2) and (4.15),

$$s_k^T y_k \leq O(\|s_k\|^2 + \|s_k\|\|\alpha_k v_k\|) + \|s_k\|O(\|s_k\| + \|\alpha_k v_k\|)^2.$$

Thus, based on (2.13), for $k \in S_1$,

$$\zeta_1\|\alpha_k v_k\|^2 \leq \|\alpha_k h_k\|O(\|s_k\| + \|\alpha_k v_k\|) + \|\alpha_k h_k\|O(\|s_k\| + \|\alpha_k v_k\|)^2.$$

Either $\|h_k\| \leq \|v_k\|$, which implies

$$\|v_k\| \leq \|h_k\|O(\|v_k\|) \leq \|h_k\|O(\gamma_A \sup_D \|c(x)\|) = O(\|h_k\|),$$

or $\|v_k\| \leq \|h_k\|$ shows the existence of the constant $\gamma_8 > 0$.

For the second part of this lemma, the existence of $\gamma_9 > 0$ can be proved as follows. For any $k \in S_2$, Lemma 3.1 and (2.14) imply

$$m\|s_k\|^2 \leq s_k^T y_k \leq \zeta_2\|\alpha_k v_k\|^2.$$

If Assumption 4.1 holds, using the second inequality from the bottom in (4.15), we have

$$\begin{aligned}s_k^T y_k &\geq s_k^T H_k s_k - |s_k^T \tilde{H}_k(\alpha_k c_k)| - O(\|d_k\|^3)\\ &\geq m_0\|s_k\|^2 - M_0\gamma_Z\gamma_A\|\alpha_k A_k^T v_k\|\|s_k\| - O(\|d_k\|^3)\\ &\geq m_0\|s_k\|^2 - M_0\gamma_Z\gamma_A^2\|\alpha_k v_k\|\|s_k\| - O(\|d_k\|^3),\end{aligned}$$

by the definition of $\tilde{H}_k$, since $\|s_k\| = O(\|d_k\|)$. Then for $k \in S_2$,

$$\zeta_2 \|\alpha_k v_k\|^2 \geq s_k^T y_k \geq m_0 \|s_k\|^2 - M_0 \gamma_Z \gamma_A^2 \|\alpha_k v_k\| \|s_k\| - O(\|d_k\|^3).$$

Since $x_k \to x^*$ and $v_k \to 0$, $\|s_k\|$ is small for large k. Then either

$$M_0 \gamma_Z \gamma_A^2 \frac{\|\alpha_k v_k\|}{\|s_k\|} \leq \frac{m_0}{3},$$

which shows

$$\zeta_2 \|\alpha_k v_k\|^2 \geq \frac{2m_0}{3} \|s_k\|^2 - O(\|s_k\|^3) \geq \frac{m_0}{2} \|s_k\|^2,$$

$$\sqrt{\zeta_2} \|\alpha_k v_k\| \geq \sqrt{\frac{m_0}{2}} \|s_k\| \geq \sqrt{\frac{m_0}{2}} \frac{\|\alpha_k h_k\|}{\gamma_Z},$$

or $M_0 \gamma_Z \gamma_A^2 \frac{\|\alpha_k v_k\|}{\|s_k\|} > \frac{m_0}{3}$, which implies

$$(M_0 \gamma_Z \gamma_A^2) \|v_k\| \geq \frac{m_0}{3} \|B_k^{-1} Z_k^T g_k\| \geq \frac{m_0}{3\gamma_Z} \|h_k\|.$$

In either case γ_9 exists. □

To prove R-linear convergence for the positive curvature criterion, we concentrate on the reductions in the vertical direction. We show that $\alpha_k = 1$ for $k \in S_2$ sufficiently large if an RHSQP algorithm uses the positive curvature criterion and either the l_1 or Fletcher merit function in the following lemmas.

LEMMA 4.8. *If the conditions of Theorem* 4.2 *are satisfied and the l_1 merit function is used in Algorithm* 2.1 *with the positive curvature criterion to generate a sequence, $\{x_k\}$, then for any sufficiently large $k \in S_2$, $\alpha_k = 1$.*

Proof. For $\alpha_k < 1$ for $k \in S_2$ in the backtracking line search, we show that the reduction in the l_1 merit function is greater than a positive constant. Since the merit function is bounded below, this implies that there is only a finite number $k \in S_k$ such that $\alpha_k < 1$.

Suppose $\alpha_k < 1$ for some $k \in S_2$. That means the descent condition fails for a step length $\tilde{\alpha}$, and $\alpha_k \geq \tau \tilde{\alpha}$. This means

$$\phi_\mu(x_k + \tilde{\alpha} d_k) - \phi_\mu(x_k) > \eta \tilde{\alpha} D\phi_\mu(x_k; d_k).$$

On the other hand, by the Taylor expansion,

$$\phi_\mu(x_k + \tilde{\alpha} d_k) - \phi_\mu(x_k) \leq \tilde{\alpha} D\phi_\mu(x_k; d_k) + O(\tilde{\alpha}^2 \|d_k\|^2).$$

Thus

$$-(1-\eta) D\phi_\mu(x_k; d_k) < \tilde{\alpha} O(\|d_k\|^2) \leq \tilde{\alpha} \gamma_{10} \|d_k\|^2, \tag{4.20}$$

and furthermore, we have a lower bound for α_k:

$$\alpha_k \geq \tau \tilde{\alpha} > -\tau(1-\eta) D\phi_\mu(x_k; d_k) / (\gamma_{10} \|d_k\|^2).$$

Using (3.4),

$$\begin{aligned}\alpha_k &\geq \tau(1-\eta)\left(\frac{1}{\gamma_Z}\|Z_k^T g_k\|\|h_k\|\cos\hat{\theta}_k + \rho\|c_k\|_1\right)/(\gamma_{10}\|d_k\|^2)\\ &\geq \tau(1-\eta)\rho\|c_k\|_1/(\gamma_{10}\|d_k\|^2)\\ &\geq \tau(1-\eta)\rho\|c_k\|_1/(\gamma_{10}(1+\gamma_9)^2\|v_k\|^2)\\ &\geq \frac{\gamma_{11}}{\|c_k\|}.\end{aligned}$$

Since $c_k \to 0$, this contradicts the assumption that $\alpha_k < 1$. Thus, for large k, $\alpha_k = 1$. □

For the Fletcher merit function, a stronger condition on the penalty parameter must be added to guarantee $\alpha_k = 1$.

LEMMA 4.9. *Suppose the conditions of Theorem* 4.2 *are satisfied and the Fletcher merit function is used in Algorithm* 2.1 *with the positive curvature criterion. Then there is a constant* $\tilde{\nu} > 0$ *such that if the penalty parameter* ν_k *is greater than* $\tilde{\nu}$, $\alpha_k = 1$ *for any* $k \in S_2$ *large enough.*

Proof. Since the Fletcher merit function is differentiable,

$$\nabla\Phi_\nu(x) = g(x) + \nabla\lambda(x)c(x) + A(x)\lambda(x) + \nu A(x)c(x),$$

and by using the relation $\lambda(x_k)^T c(x_k) = g_k^T v_k$,

$$\nabla\Phi_\nu(x_k)^T d_k = g_k^T h_k + d_k^T\nabla\lambda(x_k)c_k - \nu c_k^T c_k.$$

By Lemma 4.7, $d_k \to 0$ for $k \in S_2$ as $x_k \to 0$. Thus, using (2.16), and noticing that $\lambda(x_k+d_k) - \lambda_k \to 0$ and $c(x_k+d_k) = O(\|d_k\|^2)$, the Taylor expansion for the Lagrangian function gives

$$\begin{aligned}&\Phi_\nu(x_k+d_k) - \Phi_\nu(x_k) - \eta\nabla\Phi_\nu(x_k)^T d_k\\ &= f(x_k+d_k) + \lambda_k^T c(x_k+d_k) + \frac{\nu}{2}c(x_k+d_k)^T c(x_k+d_k) - \Big(f(x_k) + \lambda_k^T c_k + \frac{\nu}{2}c_k^T c_k\Big)\\ &\quad + (\lambda(x_k+d_k)-\lambda_k)^T c(x_k+d_k) - \eta(g_k^T h_k + d_k^T\nabla\lambda(x_k)c_k - \nu c_k^T c_k)\\ &\leq g_k^T d_k + \frac{1}{2}d_k^T\nabla^2 f_k d_k + \lambda_k^T A_k^T d_k + \frac{1}{2}d_k^T\sum_i(\lambda_k)_i\nabla^2 c_i(x_k)d_k + o(\|d_k\|^2)\\ &\quad - \frac{\nu}{2}c_k^T c_k - \eta(g_k^T h_k + d_k^T\nabla\lambda(x_k)c_k - \nu c_k^T c_k)\\ &= (1-\eta)g_k^T h_k - \eta d_k^T\nabla\lambda(x_k)c_k\\ &\quad + \frac{1}{2}d_k^T\nabla_{xx}^2 L(x_k,\lambda(x_k))d_k - \left(\frac{1}{2}-\eta\right)\nu c_k^T c_k + o(\|d_k\|^2)\\ &\leq -\eta d_k^T\nabla\lambda(x_k)c_k + M_0\|d_k\|^2 - \left(\frac{1}{2}-\eta\right)\nu c_k^T c_k + o(\|d_k\|^2),\end{aligned}$$

by (2.19) and the fact that $g_k^T h_k < 0$. Because Lemma 4.7 implies $\|h_k\| \leq \gamma_9\gamma_A\|c_k\|$ for any $k \in S_2$ and $(d_k^T\nabla\lambda(x_k)c_k)/\|c_k\|^2 \leq \sup\|\nabla\lambda(x)\|(1+\gamma_9\gamma_A)$,

$$
\begin{aligned}
&\Phi_\nu(x_k+d_k)-\Phi_\nu(x_k)-\eta\nabla\Phi_\nu(x_k)^T d_k\\
&\le -\eta d_k^T\nabla\lambda(x_k)c_k + M_0(1+\gamma_9)^2\gamma_A^2\|c_k\|^2-\left(\frac{1}{2}-\eta\right)\nu c_k^T c_k + o(\|c_k\|^2)\\
&\le -\left(\nu\left(\frac{1}{2}-\eta\right)-\eta\sup\|\nabla\lambda(x)\|(1+\gamma_9\gamma_A)-M_0(1+\gamma_9)^2\gamma_A^2-\frac{1}{2}\right)\|c_k\|^2\\
&\quad -\frac{1}{2}\|c_k\|^2+o(\|c_k\|^2)\\
&\le 0
\end{aligned}
$$

for k large enough and $\nu \ge \tilde{\nu} > 0$, where $\tilde{\nu}$ is a constant satisfying

$$
\tilde{\nu} \ge \frac{-\eta\sup\|\nabla\lambda(x)\|(1+\gamma_9\gamma_A)-M_0(1+\gamma_9)^2\gamma_A^2-\frac{1}{2}}{\frac{1}{2}-\eta} \tag{4.21}
$$

for any k. That is, for $k \in S_2$ large enough, $\alpha_k = 1$ is accepted by the line search for the Fletcher merit function. □

Given these results, we can show the R-linear convergence of the RHSQP algorithms using the positive curvature criterion. First, we have an estimate for $\|e_k\|$ as follows.

LEMMA 4.10. *If the conditions of Theorem* 4.2 *hold and Algorithm* 2.1 *is used with the positive curvature criterion and either the l_1 merit function with* (2.17) *or the Fletcher merit function with* (2.18) *and ν_k eventually sufficiently large, then for any index set $\mathcal{S} \subset [1, 2, \ldots, k-1]$,*

$$
\|e_k\| \le \gamma_{12}^{|\mathcal{S}|}\prod_{j\in\mathcal{S}}\frac{\|e_{j+1}\|}{\|e_j\|}.
$$

Proof. Without loss of generality, assume $\|e_1\| = 1$. By Lemma 4.1 and the monotonicity of $\{\Phi_\nu\}$,

$$
\|e_k\|^2 \le \frac{1}{\gamma_5}(\Phi_\nu(x_k)-\Phi_\nu(x^*)) \le \frac{1}{\gamma_5}(\Phi_\nu(x_{k'+1})-\Phi_\nu(x^*)),
$$

where k' is the largest index in $\mathcal{S}$. By (4.5) and (4.7),

$$
\begin{aligned}
\|e_k\|^2 &\le \frac{\gamma_2\gamma_6}{\gamma_5}\|e_{k'+1}\|^2,\\
\|e_k\| &\le \left(\frac{\gamma_2\gamma_6}{\gamma_5}\right)^{\frac{1}{2}}\frac{\|e_{k'+1}\|}{\|e_{k'}\|}\|e_{k'}\|.
\end{aligned}
$$

Therefore, applying the same procedure to the second largest index k'' in $\mathcal{S}$ and so on, we have

$$
\|e_k\| \le \left(\frac{\gamma_2\gamma_6}{\gamma_5}\right)^{\frac{1}{2}|\mathcal{S}|}\prod_{j\in\mathcal{S}}\frac{\|e_{j+1}\|}{\|e_j\|},
$$

and the lemma is proved with $\gamma_{12} = (\frac{\gamma_2\gamma_6}{\gamma_5})^{\frac{1}{2}}$. □

To show R-linear convergence, we need only consider the situation where $|S_2^k| \ge \frac{3}{4}k$. If $|S_2^k| < \frac{3}{4}k$, the S_1 R-linear convergence implies that the sequence $\{x_k\}$ is R-linear convergent. If $|S_2^k| \ge \frac{3}{4}k$, the index set

$$
\bar{S}^k = \{j \mid j, j+1 \in S_2^k\}
$$

contains at least $\frac{1}{4}k$ elements; otherwise $|S_2^k| < 2 \times \frac{1}{4}k + |S_1^k| \le \frac{1}{2}k + \frac{1}{4}k = \frac{3}{4}k$. Moreover, for any $j \in \bar{S}^k$, the following bound holds.

Lemma 4.11. *Under the conditions of Lemma* 4.10, *for any positive constant* $\epsilon > 0$ *and any sufficiently large* k *and* $j \in \bar{S}^k$ *such that* $\alpha_k = 1$,

$$\frac{\|e_{j+1}\|}{\|e_j\|} \le \epsilon.$$

Proof. Using the fact that

$$-\frac{q_k}{\cos^2\theta_k} + \ln q_k + 1 \le 0$$

in the inequality (4.11), we can obtain the growth bounds

$$\|B_k\| \le \gamma|S_1^k|, \qquad \|B_k^{-1}\| \le \gamma|S_1^k|$$

for some constant γ. For any constant τ with $1 > \tau > 0$, Lemma 4.3 implies

$$\|Z_k^T g_k\|^\tau \gamma_Z^{1-\tau}\|B_k\|^{1-\tau} \le \|e_k\|^\tau \gamma_Z^{1-\tau}\|B_k\|^{1-\tau} \le r^{\tau|S_1^k|}\gamma_Z^{1-\tau}|S_1^k|^{1-\tau} \le 1$$

for k sufficiently large. Since $\alpha_j = 1$ and $j \in S_2$ is large, $c_{j+1} = O(\|d_j\|^2)$ and by Lemma 4.1,

$$\begin{aligned}
\|e_{j+1}\| &\le \frac{1}{\gamma_1}(\|Z_{j+1}^T g_{j+1}\| + \|c_{j+1}\|)\\
&\le \frac{1}{\gamma_1}(\|Z_{j+1}^T g_{j+1}\|^\tau \|Z_{j+1}^T g_{j+1}\|^{1-\tau} + \|c_{j+1}\|)\\
&\le \frac{1}{\gamma_1}(\|Z_{j+1}^T g_{j+1}\|^\tau \gamma_Z^{1-\tau}\|B_{j+1}\|^{1-\tau}\|h_{j+1}\|^{1-\tau} + \|c_{j+1}\|)\\
&\le \frac{1}{\gamma_1}(\|h_{j+1}\|^{1-\tau} + \|c_{j+1}\|)\\
&\le \frac{1}{\gamma_1}(\gamma_8^{1-\tau}\|c_{j+1}\|^{1-\tau} + \|c_{j+1}\|)\\
&\le O(\|c_{j+1}\|^{1-\tau}) \le O(\|d_j\|^{2(1-\tau)})\\
&\le O(\|h_j\|^{2(1-\tau)} + \|c_j\|^{2(1-\tau)})\\
&\le O(\|c_j\|^{2(1-\tau)}) \le O(\|e_j\|^{2(1-\tau)}).
\end{aligned}$$

Since the assumptions of Theorem 4.2 are satisfied, $\|e_k\| \to 0$. Therefore, as long as $\tau < \frac{1}{2}$ is chosen, for any given constant $\epsilon > 0$,

$$\frac{\|e_{j+1}\|}{\|e_j\|} \le \epsilon$$

for k large enough. □

Using Lemmas 4.10 and 4.11,

$$\begin{aligned}
\|e_k\| &\le \gamma_{12}^{|\bar{S}^k|} \prod_{j\in\bar{S}^k} \frac{\|e_{j+1}\|}{\|e_j\|}\\
&\le \gamma_{12}^{|\bar{S}^k|} \prod_{j\in\bar{S}^k} \epsilon
\end{aligned}$$

$$\begin{aligned}
&\leq \gamma_{12}^{|\bar{S}^k|}\epsilon^{|\bar{S}^k|}\\
&\leq (\gamma_{12}\epsilon)^{|\bar{S}^k|}\\
&\leq (\gamma_{12}\epsilon)^{\frac{1}{4}k}\\
&\leq ([\gamma_{12}\epsilon]^{\frac{1}{4}})^k;
\end{aligned}$$

i.e., $\|e_k\| \leq r^k$ for some $r \in (0,1)$. Thus $\{x_k\}_{k=1}^{\infty}$ converges R-linearly, and we have the following theorem.

THEOREM 4.12. *If Assumptions* 4.1 *and* 4.2 *hold and an RHSQP algorithm uses the step secant update strategy with the positive curvature criterion and either the l_1 or the Fletcher merit function with the penalty parameter with ν_k sufficiently large and $\bar{\nu}_k$ bounded, then the sequence $\{x_k\}_{k=1}^{\infty}$ produced by the algorithm converges to the solution x^* R-linearly.* □

Now global and R-linear convergence has been established for the step secant update used with the l_1 and Fletcher merit functions. In the next section, we present some results of numerical experiments comparing the step secant update strategy using the positive curvature criterion and the Nocedal–Overton criterion with the null space secant update strategy.

5. Numerical experiments. Although the null space secant strategy and the step secant update strategies with the Nocedal–Overton update criterion and the positive curvature criterion are proved R-linearly convergent, their numerical performances differ. We present here numerical experiments with the step secant update strategy using these two update criteria and compare them with the null space secant update strategy, although it is known that the null space secant update strategy is expensive because of the extra gradient evaluations. We used a single FORTRAN code that allowed us to vary update strategies. In these numerical experiments, the simpler l_1 merit function is used. We used the QR factorization (2.7) to compute the null space basis matrix Z_k as well as the inverse matrices, as we described in (2.6). For the null space secant update, we used the BFGS update with y_k and s_k given by (2.9) and (2.10). We skipped the updates if $s_k^T y_k \leq 0$.

The algorithm parameters used are

- the general parameters, $\rho = 1$, $B_0 = I$, $\eta = 10^{-4}$, and $\tau = \tau' = 0.5$;
- parameters for the Nocedal–Overton criterion: $\zeta = 1.0$ and $\epsilon = 0.01$ (the same values used in [11]);
- parameters for the positive curvature criterion: $\zeta_1 = \zeta_2 = 0.01$.

The problems tested are chosen from Hock and Schittkowski's test problems [9]. For example, "hs10" stands for problem 10 from [9]. The following notation is used in Tables 5.1 and 5.2, which show our numerical results:

- "upd" is the number of updates;
- "ite" is the number of iterations;
- "rsd" is the residual, $\|Z_k^T g_k\| + \|c_k\|$; and
- "**F**" indicates the algorithm's failure on that problem.

From the tables, we can see that there are two kinds of failure cases: In some cases the number of iterations hit the maximum iteration allowance which was set to 100. In failure cases with a number of iterations less than the maximum iteration allowance, the algorithm stopped because of failure of the line search. That is, at the current approximation, the line search cannot find a positive step length greater than $\alpha_{\min}$ (which was chosen as $\alpha_{\min} = 10^{-30}$) such that (2.15) holds. At the bottoms of the two tables, there are rows to show the total numbers of updates/iterations

Table 5.1
Numerical tests with monotonically increasing parameter.

	Positive curvature		Nocedal–Overton		Null space secant	
Pro #	upd/ite	rsd	upd/ite	rsd	upd/ite	rsd
hs6	8/11	0.8d-08	28/36	0.7d-08	8/10	0.2d-08
hs7	3/8	0.2d-11	3/10	0.5d-09	5/7	0.1d-11
hs10	12/18	0.2d-10	3/15	0.9d-08	13/14	0.3d-10
hs11	5/9	0.6d-10	5/9	0.6d-10	7/8	0.9d-11
hs12	10/11	0.3d-08	6/13	0.4d-08	7/8	0.2d-08
hs26	12/100	0.2d-02**F**	23/26	0.3d-02**F**	33/35	0.7d-08
hs27	15/100	0.2d-01**F**	25/32	0.1d-01**F**	47/49	0.1d-08
hs29	10/12	0.1d-08	13/17	0.1d-08	7/9	0.5d-08
hs39	13/15	0.2d-09	9/28	0.8d-09	21/22	0.1d-09
hs40	4/6	0.7d-08	3/6	0.1d-09	4/5	0.1d-09
hs43	12/13	0.1d-08	34/41	0.5d-01**F**	9/10	0.8d-08
hs46	17/100	0.8d-02**F**	21/22	0.2d-01**F**	99/100	0.2d-05**F**
hs47	20/100	0.1d+01**F**	22/32	0.1d-01**F**	31/32	0.1d-08
hs56	16/17	0.6d-12	14/18	0.1d-09	10/11	0.6d-09
hs60	11/12	0.1d-09	9/12	0.6d-10	10/11	0.4d-08
hs61	36/80	0.3d-10	42/45	0.1d+02**F**	7/9	0.2d-08
hs63	5/7	0.5d-09	4/8	0.4d-10	5/6	0.6d-08
hs65	6/100	0.7d+01**F**	19/36	0.8d+01**F**	10/13	0.5d-09
hs66	10/11	0.6d-13	34/36	0.1d-02**F**	11/12	0.1d-13
hs71	10/11	0.4d-08	6/9	0.1d-09	6/7	0.3d-08
hs72	13/23	0.2d-08	13/27	0.6d-08	19/20	0.1d-09
hs77	14/15	0.2d-08	9/14	0.5d-09	18/19	0.2d-08
hs78	5/7	0.5d-08	4/7	0.9d-08	6/7	0.1d-08
hs79	26/28	0.1d-08	11/14	0.9d-09	11/12	0.1d-08
hs80	6/9	0.1d-10	6/10	0.2d-09	4/5	0.1d-09
hs81	14/22	0.5d-05**F**	8/19	0.2d+02**F**	4/5	0.1d-09
hs93	39/41	0.2d-09	26/32	0.1d+03**F**	22/26	0.1d-08
hs100	27/28	0.1d-08	36/43	0.1d-08	99/100	0.2d-06**F**
Total[1]	171/219	6**F**	146/253	10**F**	161/181	2**F**
Upd. ratio	0.781		0.577		0.890	

[1] Note that the totals are for problem solved by all three strategies.

and the update ratios. These totals are obtained by counting only the cases where all three algorithms successfully reached the solution. The stopping criterion was rsd $\leq \varepsilon = 10^{-8}$ and the starting points were the standard points given in [9]. The problems were tested on a Sun SPARC2 workstation.

First, we present the results obtained using the monotonically increasing strategy given in step $2'$, for which the global and R-linear convergence is established. These results are presented in Table 5.1. It shows that the positive curvature criterion improves the Nocedal–Overton criterion, as it not only has fewer failed cases but also uses fewer function evaluations and iterations. We believe that this improvement is due to the higher update rate of the positive curvature criterion.

Even though no convergence analysis has been established for the nonmonotonically increasing strategy, numerical experiments in Table 5.2 show it works much better than the monotonically increasing strategy. For these numerical experiments, the following nonmonotonic strategy is used:

$$\mu_k = \|\lambda_k\| + \rho,$$

where $\rho > 0$ is a constant. For this nonmonotonic increasing strategy, the step secant update strategy with the positive curvature criterion works almost as well as the null space update strategy, and in addition it saves the extra gradient evaluations.

Table 5.2
Numerical tests with nonmonotonically increasing parameter.

	Positive curvature		N–O criterion		Null space secant	
Pro #	upd/ite	rsd	upd/ite	rsd	upd/ite	rsd
hs6	11/13	0.7d-09	12/20	0.7d-09	8/10	0.2d-08
hs7	4/8	0.4d-15	3/10	0.3d-09	5/7	0.1d-11
hs10	26/34	0.5d-11	6/30	0.3d-08	21/22	0.4d-09
hs11	10/17	0.3d-09	3/16	0.4d-08	13/14	0.1d-12
hs12	9/10	0.1d-08	6/14	0.1d-10	7/8	0.2d-08
hs26	28/29	0.7d-08	27/31	0.7d-08	33/35	0.7d-08
hs27	34/36	0.3d-08	30/56	0.4d-08	27/28	0.1d-10
hs29	13/15	0.6d-09	8/30	0.6d-09	13/15	0.9d-08
hs39	16/18	0.1d-08	6/21	0.2d-15	16/17	0.5d-08
hs40	4/6	0.7d-08	3/6	0.1d-09	4/5	0.1d-09
hs43	14/15	0.1d-09	9/18	0.1d-10	10/11	0.1d-08
hs46	33/34	0.2d-08	33/34	0.2d-08	40/41	0.6d-08
hs47	19/23	0.2d-08	17/26	0.5d-08	29/30	0.2d-08
hs56	15/16	0.1d-09	0/100	**F**	16/17	0.2d-11
hs60	11/12	0.5d-10	10/13	0.3d-08	11/12	0.3d-08
hs61	10/13	0.6d-12	16/26	**F**	7/9	0.1d-11
hs63	8/9	0.9d-10	5/11	0.1d-11	7/8	0.1d-09
hs65	20/22	0.4d-10	8/15	0.4d-09	9/11	0.2d-09
hs66	8/9	0.1d-08	7/9	0.2d-08	8/9	0.2d-12
hs71	10/11	0.4d-08	6/9	0.1d-09	6/7	0.3d-08
hs72	13/23	0.2d-08	13/27	0.6d-08	19/20	0.1d-09
hs77	13/14	0.8d-10	9/14	0.3d-09	18/19	0.2d-08
hs78	5/7	0.5d-08	4/7	0.9d-08	6/7	0.1d-08
hs79	21/22	0.1d-09	8/12	0.7d-09	11/12	0.1d-08
hs80	4/21	0.3d-10	2/20	0.6d-08	17/18	0.1d-11
hs81	25/39	0.4d-08	8/100	**F**	4/5	0.1d-09
hs93	72/75	0.8d-08	2/100	**F**	21/24	0.1d-09
hs100	28/29	0.4d-08	32/39	0.1d-09	35/36	0.5d-13
Total[1]	362/437	0**F**	267/488	4**F**	373/402	0**F**
Upd. ratio	0.830		0.550		0.928	

[1] Note that the totals are for problem solved by all three strategies.

6. Conclusions.

The purpose of this paper is to present a more realistic analysis of reduced Hessian SQP and to present a new practical update criterion. It presents the first analysis of the step secant update for RHSQP in the context of a line search, and without assumptions on the accuracy of the initial Hessian approximation. We have done this for both the well-known Nocedal–Overton criterion and for the positive curvature criterion proposed here.

The positive curvature update criterion was proposed to allow more updates than the Nocedal–Overton update criterion, and based on numerical experiments, this seems to occur. It seems plausible that the superior performance using the positive curvature criterion is due to this greater update frequency.

From the numerical experiments made in this paper and the global and R-linear convergence results for the step secant update strategies, the positive curvature criterion may be a competitive candidate for solving very large scale constrained optimization problems, especially when it is combined with a nonmonotonic penalty parameter strategy, because it saves the extra gradient evaluation required by the null space secant update strategy.

REFERENCES

[1] R. H. Byrd and J. Nocedal, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM J. Numer. Anal., 26 (1989), pp. 727–739.

[2] R. H. Byrd and J. Nocedal, *An analysis of reduced Hessian methods for constrained optimization*, Math. Programming, 49 (1990), pp. 285–323.

[3] T. F. Coleman and A. R. Conn, *On the local convergence of a quasi-Newton method for the nonlinear programming problem*, SIAM J. Numer. Anal., 21 (1984), pp. 755–769.

[4] R. Fletcher, *A class of methods for nonlinear programming with termination and convergence properties*, in Integer and Nonlinear Programming, North–Holland, Amsterdam, 1970, pp. 157–175.

[5] R. Fletcher, *A First Derivative Method for Nonlinear Programming Based on Successive l_1 LP*, Numerical Analysis Report NA/114, Department of Mathematics and Computer Science, University of Dundee, Scotland, 1988.

[6] D. Gabay, *Reduced quasi-Newton methods with feasibility improvement for nonlinear constrained optimization*, Math. Programming Stud., 16 (1982), pp. 18–44.

[7] J. C. Gilbert, *Maintaining the positive definiteness of the matrices in reduced Hessian methods for equality constrained optimization*, Math. Programming, 50 (1991), pp. 1–28.

[8] S. P. Han, *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Math. Programming, 11 (1976), pp. 263–282.

[9] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Econom. and Math. Systems, Springer-Verlag, New York, 1981.

[10] W. Murray and M. H. Wright, *Projected Lagrangian Methods Based on the Trajectories of Penalty and Barrier functions*, Systems Optimization Laboratory Report 78-23, Stanford University, Stanford, CA, 1978.

[11] J. Nocedal and M. L. Overton, *Projected Hessian updating algorithms for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 22 (1985), pp. 821–850.

[12] J. D. Pearson, *Variable metric methods of minimization*, Computer J., 12 (1969), pp. 171–178.

[13] M. J. D. Powell, *Some global convergence properties of a variable metric algorithm for minimization without exact line searches*, in Nonlinear Programming, SIAM–AMS Proceedings, Vol. IX, American Mathematical Society, Providence, RI, 1976, pp. 53–72.

[14] M. J. D. Powell, *Variable metric methods for constrained optimization*, in Mathematical Programming: The State of the Art, Springer-Verlag, New York, 1983, pp. 288–311.

SIAM J. OPTIM.
Vol. 9, No. 3, pp. 605–623

A GLOBAL LINEAR AND LOCAL QUADRATIC NONINTERIOR CONTINUATION METHOD FOR NONLINEAR COMPLEMENTARITY PROBLEMS BASED ON CHEN–MANGASARIAN SMOOTHING FUNCTIONS[*]

BINTONG CHEN[†] AND NAIHUA XIU[‡]

Abstract. A noninterior continuation method is proposed for nonlinear complementarity problems. It improves the noninterior continuation methods recently studied by Burke and Xu [*Math. Oper. Res.*, 23 (1998), pp. 719–734] and Xu [*The Global Linear Convergence of an Infeasible Non-Interior Path-following Algorithm for Complementarity Problems with Uniform P-functions*, Preprint, Department of Mathematics, University of Washington, Seattle, 1996]; the interior point neighborhood technique is extended to a broader class of smoothing functions introduced by Chen and Mangasarian [*Comput. Optim. Appl.*, 5 (1996), pp. 97–138]. The method is shown to be globally linearly convergent following the methodology established by Burke and Xu. In addition, a local acceleration step is added to the method so that it is also locally quadratically convergent under suitable assumptions.

Key words. nonlinear complementarity problem, continuation method, smoothing function, global linear convergence, local quadratic convergence

AMS subject classification. 90C33

PII. S1052623497316191

1. Introduction.

Let $F : R^n \to R^n$ be a continuously differentiable function. The nonlinear complementarity problem (NCP) is to find $(x, y) \in R^n \times R^n$ such that

$$F(x) - y = 0, \tag{1}$$

$$x \geq 0, \quad y \geq 0, \quad x^T y = 0. \tag{2}$$

Numerous methods have been developed to solve the NCP; for a comprehensive survey, see [12, 18]. In this paper, we are interested in developing a noninterior continuation method for the NCP and analyzing its rate of convergence.

Like interior point algorithms, noninterior continuation methods approximate the complementarity condition (2) with a parameterized system of smooth equations. At each iteration, Newton's method is applied to solve the smooth equations. The smoothing parameter is then adjusted to refine the approximation. However, the smooth approximation of condition (2) used in noninterior continuation methods is different from that used in interior point methods. As a result, the intermediate iterates are not required to stay in positive orthant. The first noninterior method was introduced by Chen and Harker [3], where the authors concentrated on establishing the structural properties of the central path for linear complementarity problems (LCPs) with P_0- and R_0-matrices. The method was later improved by Kanzow [13], where the author refined the smoothing function and established the convergence for the continuation method under similar assumptions. However, both methods lack a systematic

[*] Received by the editors February 4, 1997; accepted for publication (in revised form) May 4, 1998; published electronically April 19, 1999.

http://www.siam.org/journals/siopt/9-3/31619.html

[†] Department of Management and Systems, Washington State University, Pullman, WA 99164-4736 (chenbi@wsu.edu).

[‡] Institute of Applied Mathematics, Academia Sinica, Beijing 100080, China (nhxiu@center.njtu.edu.cn).

procedure to reduce the smoothing parameter to zero, even though they have shown impressive numerical performance [3, 13] compared with interior point algorithms. As a result, no rate-of-convergence results were obtained. This gap was closed recently by Burke and Xu [1]. Inspired by many path-following interior point algorithms, the authors introduced a notion of neighborhood around the central path for their noninterior continuation methods. All intermediate iterates are required to stay within the neighborhood, and this provides a systematic way of reducing the smoothing parameter. This important addition to the continuation methods allowed them to establish the global linear convergence for both LCPs with P_0- and R_0-matrices [1] and NCPs with uniform P-functions [24]. In addition, their computational experiments have shown further improvement over previous noninterior continuation methods. Besides the above-mentioned literature, similar noninterior continuation methods have been developed to solve linear and quadratic programs [4], complementarity problems [14], and variational inequalities [5, 16].

All noninterior continuation methods mentioned above are based on smoothing functions derived from $x_i y_i = \mu$, the deformed complementarity condition used for interior point algorithms. Many other smoothing functions exist. Indeed, Chen and Mangasarian [8] have proposed a broad class of smoothing functions for the plus function $z_+ = \max\{z, 0\}$. Roughly speaking, their smoothing functions are derived from double integrals of parameterized probability density functions. Many smoothing functions proposed earlier, including the interior point related smoothing functions mentioned above, turned out to be special cases of the Chen–Mangasarian smoothing function family. They differ only in the choice of probability density functions. Since these smoothing functions can be derived through the same mechanism, one would expect that the continuation methods based on these functions would share similar properties and perform similarly. In fact, this conjecture has been partially confirmed by the extensive numerical experiments conducted by Chen and Mangasarian [7, 8], where they reported similar impressive performance for continuation (smoothing) methods based on other smoothing functions. Chen and Harker [6] later studied the structural properties of the continuation methods based on the Chen–Mangasarian smoothing function family.

The current paper improves and generalizes the noninterior continuation methods by Burke and Xu [1] and Xu [24]. The new method is based on the Chen–Mangasarian smoothing function family and is shown to have a global linear convergence, following the methodology established in [1]. In addition, we incorporate a local acceleration step into the method so that it also achieves local quadratic convergence.

The paper is organized as follows. Section 2 studies a subclass of the Chen–Mangasarian smoothing function family to be used for the continuation method. Some new properties of the smoothing functions are explored. Section 3 introduces a new definition of the neighborhood of the central path and describes the continuation method based on the Chen–Mangasarian smoothing functions. Section 4 shows that the continuation method converges globally linearly to a solution of the NCP under certain assumptions. Section 5 proves the local quadratic convergence of the continuation method under the strict complementarity assumption at the limit point. Finally, conditions on function F to guarantee the global linear convergence are discussed in section 6.

A brief note on the notation to be used in this paper: $\|\cdot\|$ denotes a 1, 2, or ∞ norm as well as its induced matrix norm. All vectors are column vectors. x, y, z are scalars in section 2 but vectors in all other sections. For simplicity, we sometimes use

(x, y) for the column vector $(x\ y)^T$. In addition, $\text{vec}\{x_i\}$ stands for a vector whose ith element is x_i. In this case, $\text{vec}\{x_i\} = x$. If x is a vector, then $x_+ = \text{vec}\{(x_i)_+\}$. We use $\text{diag}\{x_i\}$ for a diagonal matrix with iith entry equal to x_i. Finally, $\text{dist}\{S, T\}$ represents the minimum distance between set S and set T, measured in 1, 2, or ∞ norm.

2. Chen–Mangasarian smoothing functions and properties.

Chen and Mangasarian [8] introduced a class of smoothing function $p_\mu(z)$ that approximates the fundamental plus function z_+ by twice integrating a parameterized probability density function. More specifically, their smoothing function is defined as

$$p_\mu(z) = \int_{-\infty}^{z} \int_{-\infty}^{t} \frac{1}{\mu} d\left(\frac{x}{\mu}\right) dx dt, \tag{3}$$

where $0 \leq \mu < \infty$ is a smoothing parameter and $d(x)$ is a probability density function that satisfies certain technical assumptions. Clearly, as μ approaches zero, the probability density function $(1/\mu)d(x/\mu)$ approaches the delta function with all mass concentrated at the origin, and the smoothing function $p_\mu(z)$ approaches the plus function z_+. In this regard, $p_\mu(z)$ can be considered as a natural approximation of the plus function z_+. Indeed, it has been shown by Chen and Harker [6] that any "well"-behaved smooth approximation of the plus function must be a double integral of a probability density function. For convenience of presentation, we denote $p_0(z) = \lim_{\mu \to 0} p_\mu(z) = z_+$. In addition, $\partial p_0(z)$ stands for the generalized derivative of p_0 at z in the sense of Clarke [11].

In this paper, we consider a subset of the Chen–Mangasarian smoothing function family, whose probability density function d satisfies the following properties.

Assumption 1. *The probability density function d satisfies the following conditions:*

(A1) *$d(x)$ is a continuous function such that $0 < d(x) \leq A < \infty$ and $d(x) = d(-x)$ for all $x \in (-\infty, +\infty)$.*

(A2) $\int_0^{+\infty} x d(x) dx = B < \infty$.

Under Assumption 1, the smoothing function $p_\mu(z)$ has the following properties.

Proposition 1. *Assume that the probability density function d satisfies Assumption 1. The following properties hold for the smoothing function $p_\mu(z)$ defined in (3) with $\mu > 0$:*

1. *$p_\mu(z)$ is continuously differentiable, increasing, and strictly convex with respect to z.*
2. *$0 < p'_\mu(z) < 1$ and $p'_\mu(-z) = 1 - p'_\mu(z)$ for all z.*
3. *$0 < p''_\mu(z) \leq A/\mu$ for all z.*
4. *$|p_{\mu_2}(z) - p_{\mu_1}(z)| \leq B|\mu_2 - \mu_1|$ for all z and $\mu_1, \mu_2 \geq 0$.*
5. *If $z \neq 0$, then p_0 is continuously differentiable at z. In addition, $|p'_\mu(z) - p'_0(z)| \leq B\mu/|z|$ for all z.*
6. *Let $\lim_k z^k = z$ and $\lim_k \mu_k = 0$. Then $\lim_k \text{dist}\{p'_{\mu_k}(z^k), \partial p_0(z)\} = 0$.*

Proof. Result 1 has been shown in [8]. By definition,

$$p'_\mu(z) = \int_{-\infty}^{z/\mu} d(x)dx \quad \text{and} \quad p''_\mu(z) = \frac{1}{\mu} d\left(\frac{z}{\mu}\right).$$

Results 2 and 3 then follow from Assumption 1(A1). Result 4 with either $\mu_1 = 0$ or $\mu_2 = 0$ has been shown in [8] and clearly holds if $\mu_1 = \mu_2$. Without loss of generality, assume $\mu_2 > \mu_1 > 0$. Under Assumption 1(A2), it has been shown that [7]

$$p_\mu(z) = z\int_{-\infty}^{z/\mu} d(x)dx - \mu\int_{-\infty}^{z/\mu} xd(x)dx. \tag{4}$$

Thus,

$$\frac{\partial p_\mu(z)}{\partial \mu} = -\int_{-\infty}^{z/\mu} xd(x)dx > 0,$$

where the inequality is true because $d(x)$ is symmetric by assumption. Therefore, $p_{\mu_2}(z) > p_{\mu_1}(z)$. Let

$$D(z) = |p_{\mu_2}(z) - p_{\mu_1}(z)| = p_{\mu_2}(z) - p_{\mu_1}(z).$$

Then

$$D'(z) = p'_{\mu_2}(z) - p'_{\mu_1}(z) = \int_{z/\mu_1}^{z/\mu_2} d(x)dx.$$

Since $d(x) > 0$ for all x, $D'(z) = 0$ if and only if $z = 0$. Since

$$D''(0) = d(0)\left(\frac{1}{\mu_2} - \frac{1}{\mu_1}\right) < 0,$$

$z = 0$ must be the unique maximizer of $D(z)$ and $D(z) \le D(0)$ for all z. Using equality (4) again, we have

$$D(0) = -(\mu_2 - \mu_1)\int_{-\infty}^{0} xd(x)dx = B(\mu_2 - \mu_1),$$

and result 4 follows immediately. For result 5, notice that $p_0(z) = z_+$ and the latter is continuously differentiable at all $z \neq 0$. If $z > 0$ then $p'_0(z) = 1$. Therefore,

$$|p'_\mu(z) - p'_0(z)| = \int_{z/\mu}^{\infty} d(x)dx \le \frac{\int_0^\infty xd(x)dx}{z/\mu} = \frac{B\mu}{|z|},$$

where the inequality follows from the assumption that $d(x)$ is symmetric and the well-known Markov inequality in the probability theory. The proof for the case with $z < 0$ is similar and is omitted. For result 6, observe that $\partial p_0(z)$ has the following explicit expression, due to the special structure of function p_0:

$$\partial p_0(z) = \begin{cases} 1 & \text{if } z > 0, \\ \text{all } w \in [0,1] & \text{if } z = 0, \\ 0 & \text{if } z < 0. \end{cases}$$

If $z \neq 0$, then p_0 is continuously differentiable at z. Hence,

$$\begin{aligned} \operatorname{dist}\{p'_{\mu_k}(z^k), \partial p_0(z)\} &= |p'_{\mu_k}(z^k) - p'_0(z)| \\ &\le |p'_{\mu_k}(z^k) - p'_0(z^k)| + |p'_0(z^k) - p'_0(z)| \\ &\le (B\mu_k)/|z^k| + |p'_0(z^k) - p'_0(z)|. \end{aligned}$$

The result follows from passing to the limit on both sides. If $z = 0$, the result is clearly true since $p'_{\mu_k}(z^k) \in [0, 1]$ for all k. □

Notice that result 6 of the above proposition holds even if the limit of $p'_{\mu_k}(z^k)$ does not exist in general.

In addition to the above properties of p_μ, we also need the following Taylor expansion of p_μ with $\mu > 0$, which will be used in section 4 for the global linear convergence analysis:

$$p_\mu(z + \theta\Delta z) = p_\mu(z) + \theta p'_\mu(z)\Delta z + \frac{\theta^2}{2} p''_\mu(\bar{z})(\Delta z)^2, \tag{5}$$

where $\bar{z}$ is some number between z and $z + \theta\Delta z$.

We next provide several examples of the smoothing functions that satisfy Assumption 1. All of them have been used in the literature to design smoothing or continuation methods.

Example 1. Neural network smoothing function (Chen and Mangasarian [7]):

$$d(x) = e^{-x}/(1 + e^{-x})^2, \quad p_\mu(z) = z + \mu \ln(1 + e^{-\frac{z}{\mu}}), \quad A = \frac{1}{4}, \quad B = \ln 2.$$

Example 2. Interior point smoothing function (Chen and Harker [3], Kanzow [13], Smale [21]):

$$d(x) = 2/(x^2 + 4)^{1.5}, \quad p_\mu(z) = (z + \sqrt{z^2 + 4\mu^2})/2, \quad A = \frac{1}{4}, \quad B = 1.$$

This smoothing function is closely related to both the deformation used in interior point algorithms (see [21]) and the smoothing functions for several noninterior continuation methods [3, 13, 1] mentioned in the introduction.

Example 3. Normal smoothing function:

$$d(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \quad \text{no closed form expression for } p_\mu(z), \quad A = \frac{1}{\sqrt{2\pi}}, \quad B = \sqrt{\frac{2}{\pi}}.$$

3. A continuation method for NCP.

It is well known that the complementarity condition (2) in the NCP can be rewritten as

$$\min\{x, y\} = 0 \quad \text{or} \quad x - (x - y)_+ = 0,$$

where the plus function is taken componentwise. By deforming the plus function with the Chen–Mangasarian smoothing function p_μ, we obtain the following smoothed complementarity condition:

$$\Psi_\mu(x, y) \equiv x - P_\mu(x - y) = 0,$$

where $\mu \geq 0$ is a smoothing parameter, $P_\mu(x - y) = \text{vec}\{p_\mu(x_i - y_i)\}$, and $\Psi_\mu(x, y) = \text{vec}\{\psi_\mu(x_i, y_i)\}$. The smoothed NCP then becomes

$$H_\mu(x, y) = \left[\begin{array}{c} F(x) - y \\ \Psi_\mu(x, y) \end{array} \right] = 0, \tag{6}$$

and the NCP conditions (1) and (2) can be written as $H_0(x, y) = 0$. The idea behind continuation methods is to solve the smoothed NCP $H_\mu(x, y) = 0$ "approximately" for each given smoothing parameter $\mu > 0$ and gradually reduce μ to zero. Hopefully, as μ approaches zero, the solution of the smoothed NCP approaches a solution of the NCP.

Since the Jacobian matrix $\nabla H_\mu(z)$ plays an important role for the convergence analysis, we next take a look at its structure. Denote $P'_\mu(z) = \text{diag}\{p'_\mu(z_i)\}$. By definition,

$$\nabla_x \Psi_\mu(x, y) = I - P'_\mu(x - y) \quad \text{and} \quad \nabla_y \Psi_\mu(x, y) = P'_\mu(x - y).$$

By result 2 of Proposition 1, both $P'_\mu(x - y)$ and $P'_\mu(y - x)$ are positive diagonal matrices such that

$$I - P'_\mu(x - y) = P'_\mu(y - x), \quad 0 < P'_\mu(x - y) < I, \quad 0 < P'_\mu(y - x) < I.$$

Thus, the Jacobian matrix can be written as

$$\nabla H_\mu(x, y) = \begin{bmatrix} \nabla F(x) & -I \\ P'_\mu(y - x) & P'_\mu(x - y) \end{bmatrix}.$$

It is well known that the Jacobian $\nabla H_\mu(x, y)$, due to its special structure, is nonsingular if and only if matrix $P'_\mu(y - x) + P'_\mu(x - y)\nabla F(x)$ is nonsingular.

Define the following merit function for (6):

$$\rho_\mu(x, y) = \|F(x) - y\| + \|\Psi_\mu(x, y)\|.$$

Let the central path(s) of the NCP be the set of solutions of (6) for all $\mu > 0$. To construct an implementable continuation method for the NCP, we introduce a neighborhood around the central path:

$$\mathcal{N}(\beta) = \{(x, y) : \rho_\mu(x, y) \le \beta\mu, \mu > 0\},$$

where parameter $\beta > 0$ is called the width of the neighborhood. In addition, we define a slice of the neighborhood with $\mu \in U$ as

$$\mathcal{N}(\beta, U) = \{(x, y) : \rho_\mu(x, y) \le \beta\mu, \mu \in U\}.$$

For simplicity, if $U = \mu$, we will write the slice as $\mathcal{N}(\beta, \mu)$. We now describe the continuation method.

ALGORITHM 1. Given $\sigma \in (0, 1)$, and $\alpha_i \in (0, 1)$ for $i = 1, 2, 3$.

Step 0 (Initialization)

Set $k = 0$. Choose $\mu_0 > 0$, $(x^0, y^0) \in R^{2n}$, and $\beta > nB$ such that $(x^0, y^0) \in \mathcal{N}(\beta, \mu_0)$.

Step 1 (Calculate Centering Step)

If $H_0(x^k, y^k) = 0$, stop. (x^k, y^k) is a solution of the NCP; otherwise, if $\rho_{\mu_k}(x^k, y^k) = 0$, set $(\tilde{x}^{k+1}, \tilde{y}^{k+1}) = (x^k, y^k)$ and go to Step 3; otherwise, if $\nabla H_{\mu_k}(x^k, y^k)$ is singular, stop. The continuation method fails; otherwise, let $(\Delta\tilde{x}^k, \Delta\tilde{y}^k)$ solve the equation

$$H_{\mu_k}(x^k, y^k) + \nabla H_{\mu_k}(x^k, y^k)(\Delta x, \Delta y)^T = 0. \tag{7}$$

Step 2 (Line Search for Centering Step)

Let λ_k be the maximum of the values $1, \alpha_1, \alpha_1^2, \ldots$ such that

$$\rho_{\mu_k}(x^k + \lambda_k \Delta\tilde{x}^k, y^k + \lambda_k \Delta\tilde{y}^k) \le (1 - \sigma\lambda_k)\rho_{\mu_k}(x^k, y^k). \tag{8}$$

Set $(\tilde{x}^{k+1}, \tilde{y}^{k+1}) = (x^k, y^k) + \lambda_k(\Delta\tilde{x}^k, \Delta\tilde{y}^k)$.

Step 3 (μ Reduction Based on Centering Step)
Let γ_k be the maximum of the values $1, \alpha_2, \alpha_2^2, \ldots$ such that

$$(\tilde{x}^{k+1}, \tilde{y}^{k+1}) \in \mathcal{N}(\beta, (1 - \gamma_k)\mu_k). \tag{9}$$

Set $\tilde{\mu}_{k+1} = (1 - \gamma_k)\mu_k$.

Step 4 (Calculate Approximate Newton Step)
Let $(\Delta\hat{x}^k, \Delta\hat{y}^k)$ solve the equation

$$H_0(x^k, y^k) + \nabla H_{\mu_k}(x^k, y^k)(\Delta x, \Delta y)^T = 0. \tag{10}$$

Set $(\hat{x}^{k+1}, \hat{y}^{k+1}) = (x^k, y^k) + (\Delta\hat{x}^k, \Delta\hat{y}^k)$.

Step 5 (μ Reduction Based on Approximate Newton Step)
If $(\hat{x}^{k+1}, \hat{y}^{k+1}) \notin \mathcal{N}(\beta, \tilde{\mu}_{k+1})$, set

$$\mu_{k+1} = \tilde{\mu}_{k+1}, \quad (x^{k+1}, y^{k+1}) = (\tilde{x}^{k+1}, \tilde{y}^{k+1}),$$

and $k = k + 1$, Return to Step 1; otherwise, if $H_0(\hat{x}^{k+1}, \hat{y}^{k+1}) = 0$, stop. $(\hat{x}^{k+1}, \hat{y}^{k+1})$ is a solution of the NCP; otherwise, let η_k be the minimum of the values $1, \alpha_3, \alpha_3^2, \ldots$ such that

$$(\hat{x}^{k+1}, \hat{y}^{k+1}) \in \mathcal{N}(\beta, \eta_k \tilde{\mu}_{k+1}). \tag{11}$$

Set

$$\mu_{k+1} = \eta_k \tilde{\mu}_{k+1}, \quad (x^{k+1}, y^{k+1}) = (\hat{x}^{k+1}, \hat{y}^{k+1}),$$

and $k = k + 1$. Return to Step 1.

We make the following remarks about the continuation method:

- It is very easy to initialize the above continuation method. One may simply choose any $\mu_0 > 0$, $(x^0, y^0) \in R^{2n}$, and $\beta > \max\{\rho_{\mu_0}(x^0, y^0)/\mu_0, nB\}$.
- For global linear convergence, only the centering steps 1–3 are needed. The approximate Newton steps 4 and 5 are added to ensure local quadratic convergence. Notice the switch between the two steps is quite natural: the continuation method simply chooses the step that reduces μ faster. One may also use some other switching rules to achieve local quadratic convergence, such as that proposed by Wright and Ralph [23]; the approximate Newton (fast) step is chosen only if μ is reduced by more than a prespecified factor.
- Since the same matrix $\nabla H_{\mu_k}(x^k, y^k)$ is inverted in both (7) and (10), the additional computational time for calculating both the centering step and the approximate Newton step is very minimal.
- The centering steps 1–3 in our continuation method are very similar to the continuation method studied by Burke and Xu [1] and Xu [24]. However, our definition of merit function $\rho_\mu(x, y)$ seems to be simpler. Burke and Xu [1] used $\rho_\mu(x, y) = \|\bar{\Psi}_\mu(x, y)\|^2$ for the LCP and Xu [24] used $\rho_\mu(x, y) = \|F(x) - y\| + \|\bar{\Psi}_\mu(x, y)\|^2$ for the NCP, where $\bar{\Psi}_\mu(x, y) = \mathrm{vec}\{\bar{\psi}_\mu(x_i, y_i)\}$ and

$$\bar{\psi}_\mu(x_i, y_i) = x_i + y_i - \sqrt{x_i^2 + y_i^2 + 2\mu}.$$

Moreover, our choice of the merit function also leads to subsequent simplifications of the continuation method in terms of the neighborhood definition, the line search procedure, and the updating rule for μ.

- Step 4 is motivated by the recent work by Chen, Qi, and Sun [9] and Chen and Ye [10], where the approximate Newton step is shown to have good local convergence properties.

We end this section with a technical result that follows directly from the properties of our merit function and the Chen–Mangasarian smoothing function family.

LEMMA 1. *Let $\mu \geq 0$ and $\mu_i \geq 0$ for $i = 1, 2$. Then*

1. $\|\Psi_{\mu_1}(x, y) - \Psi_{\mu_2}(x, y)\| \leq nB|\mu_1 - \mu_2|$,
2. $|\rho_{\mu_1}(x, y) - \rho_{\mu_2}(x, y)| \leq nB|\mu_1 - \mu_2|$,
3. $\|H_\mu(x, y)\| \leq \rho_\mu(x, y) \leq 2\|H_\mu(x, y)\|$,
4. $\|H_0(x, y)\| \leq \rho_\mu(x, y) + nB\mu$,
5. $\rho_\mu(x, y) \leq 2\|H_0(x, y)\| + nB\mu$

for all $(x, y) \in R^{2n}$.

Proof. For each i, we have

$$|\psi_{\mu_1}(x_i, y_i) - \psi_{\mu_2}(x_i, y_i)| = |p_{\mu_1}(x_i, y_i) - p_{\mu_2}(x_i, y_i)| \leq B|\mu_1 - \mu_2|,$$

where the inequality follows from result 4 of Proposition 1. Result 1 then follows from the properties of the 1, 2, and ∞ norms. Result 2 is true because

$$\begin{aligned} |\rho_{\mu_1}(x, y) - \rho_{\mu_2}(x, y)| &= |\|\Psi_{\mu_1}(x, y)\| - \|\Psi_{\mu_2}(x, y)\|| \\ &\leq \|\Psi_{\mu_1}(x, y) - \Psi_{\mu_2}(x, y)\| \\ &\leq nB|\mu_1 - \mu_2|. \end{aligned}$$

Result 3 follows from the definitions of $H_\mu(x, y)$ and $\rho_\mu(x, y)$. Result 4 is an immediate consequence of results 2 and 3:

$$\|H_0(x, y)\| \leq \rho_0(x, y) \leq \rho_\mu(x, y) + nB\mu.$$

Result 5 follows from results 1 and 3:

$$\begin{aligned} \rho_\mu(x, y) &= \|F(x) - y\| + \|\Psi_\mu(x, y)\| \\ &\leq \|F(x) - y\| + \|\Psi_0(x, y)\| + nB\mu \\ &= \rho_0(x, y) + nB\mu \\ &\leq 2\|H_0(x, y)\| + nB\mu. \qquad \square \end{aligned}$$

The results in the above lemma will be used repeatedly in the remaining paper.

4. Global linear convergence. In this section, we show that the centering steps in the continuation method are well defined under certain assumptions. By following the centering steps, the smoothing parameter $\tilde{\mu}_k$ converges globally and linearly to zero. This result also implies the global linear convergence for the whole continuation method, since the approximate Newton step is taken only if it reduces the smoothing parameter μ_k faster. We obtain the above global linear convergence by following the pattern of proof established in Burke and Xu [1] and Xu [24]. In particular, it is established through two intermediate results: both the line search step length λ_k for the centering step and the step length γ_k for reducing μ are shown to be uniformly bounded below by a positive constant. Moreover, we show that the sequence (x^k, y^k) generated by the continuation method converges to a solution of the NCP, which may have multiple solutions.

We assume in the remaining paper that function F is Lipschitz continuous.

Assumption 2. *There exists a Lipschitz constant $L > 0$ such that*

$$\|F(z) - F(x) - \nabla F(x)(z - x)\| \leq L\|z - x\|^2$$

for all $x, z \in R^n$.

We start by studying the properties of the solution to (7) in the centering step.

Lemma 2. *Let $0 \leq \lambda \leq 1$. Suppose $\nabla H_\mu(x, y)$ is nonsingular at (x, y) for some $\mu > 0$ and $(\Delta\tilde{x}, \Delta\tilde{y})$ is the solution of (7) at (x, y).*

1. *If F satisfies Assumption 2, then*

$$\|F(x + \lambda\Delta\tilde{x}) - (y + \lambda\Delta\tilde{y})\| \leq (1 - \lambda)\|F(x) - y\| + L\lambda^2\|(\Delta\tilde{x}, \Delta\tilde{y})\|^2.$$

2. *Let A be the constant defined in Proposition 1. Then*

$$\|\Psi_\mu(x + \lambda\Delta\tilde{x}, y + \lambda\Delta\tilde{y})\| \leq (1 - \lambda)\|\Psi_\mu(x, y)\| + \frac{2A}{\mu}\lambda^2\|(\Delta\tilde{x}, \Delta\tilde{y})\|^2.$$

Proof. For result 1, notice that

$$\begin{aligned}
&\|F(x + \lambda\Delta\tilde{x}) - (y + \lambda\Delta\tilde{y})\| \\
&= \|F(x) - y + \lambda(\nabla F(x)\Delta\tilde{x} - \Delta\tilde{y}) + F(x + \lambda\Delta\tilde{x}) - F(x) - \lambda\nabla F(x)\Delta\tilde{x}\| \\
&= \|(1 - \lambda)(F(x) - y) + F(x + \lambda\Delta\tilde{x}) - F(x) - \lambda\nabla F(x)\Delta\tilde{x}\| \\
&\leq (1 - \lambda)\|F(x) - y\| + L\lambda^2\|\Delta\tilde{x}\|^2 \\
&\leq (1 - \lambda)\|F(x) - y\| + L\lambda^2\|(\Delta\tilde{x}, \Delta\tilde{y})\|^2,
\end{aligned}$$

where the second equality is true because $(\Delta\tilde{x}, \Delta\tilde{y})$ is the solution of (7) and the third inequality follows from Assumption 2. For result 2, we have

$$\begin{aligned}
&\Psi_\mu(x + \lambda\Delta\tilde{x}, y + \lambda\Delta\tilde{y}) \\
&= x + \lambda\Delta\tilde{x} - P_\mu(x - y + \lambda(\Delta\tilde{x} - \Delta\tilde{y})) \\
&= x - P_\mu(x - y) + \lambda(\Delta\tilde{x} - P'_\mu(x - y)(\Delta\tilde{x} - \Delta\tilde{y})) - \frac{\lambda^2}{2}\mathrm{vec}\{p''_\mu(\bar{x}_i - \bar{y}_i)(\Delta\tilde{x}_i - \Delta\tilde{y}_i)^2\} \\
&= \Psi_\mu(x, y) + \lambda(P'_\mu(y - x)\Delta\tilde{x} + P'_\mu(x - y)\Delta\tilde{y}) - \frac{\lambda^2}{2}\mathrm{vec}\{p''_\mu(\bar{x}_i - \bar{y}_i)(\Delta\tilde{x}_i - \Delta\tilde{y}_i)^2\} \\
&= (1 - \lambda)\Psi_\mu(x, y) - \frac{\lambda^2}{2}\mathrm{vec}\{p''_\mu(\bar{x}_i - \bar{y}_i)(\Delta\tilde{x}_i - \Delta\tilde{y}_i)^2\}.
\end{aligned}$$

In the above derivation, the second equality follows from the Taylor expansion of p_μ given by (5), where $\bar{x}_i - \bar{y}_i$ is between $x_i - y_i$ and $(x_i + \lambda\Delta\tilde{x}_i) - (y_i + \lambda\Delta\tilde{y}_i)$ for all i; the third equality follows from result 2 of Proposition 1; the fourth equality is true since $(\Delta\tilde{x}, \Delta\tilde{y})$ is the solution of (7). Result 2 then follows from the fact that

$$\begin{aligned}
\|\mathrm{vec}\{p''_\mu(\bar{x}_i - \bar{y}_i)(\Delta\tilde{x}_i - \Delta\tilde{y}_i)^2\}\| &\leq \frac{A}{\mu}\|\mathrm{vec}\{(\Delta\tilde{x}_i - \Delta\tilde{y}_i)^2\}\| \\
&\leq \frac{2A}{\mu}\|\mathrm{vec}\{(\Delta\tilde{x}_i)^2 + (\Delta\tilde{y}_i)^2\}\| \\
&\leq \frac{4A}{\mu}\|(\Delta\tilde{x}, \Delta\tilde{y})\|^2,
\end{aligned}$$

where the first inequality follows from result 3 of Proposition 1. □

We next bound the norm of each centering step and each approximate Newton step.

Lemma 3. *Let (x^k, y^k, μ_k) be the kth iterate of the continuation method. If $\|\nabla H_{\mu_k}(x^k, y^k)^{-1}\| \leq C$, then*

1. *For the centering step, we have*

$$\|(\Delta\tilde{x}^k, \Delta\tilde{y}^k)\| \le C\rho_{\mu_k}(x^k, y^k) \le \beta C\mu_k.$$

2. *For the approximate Newton step, we have*

$$\|(\Delta\hat{x}^k, \Delta\hat{y}^k)\| \le C(\beta + nB)\mu_k.$$

Proof. Since $(\Delta\tilde{x}^k, \Delta\tilde{y}^k)$ is a solution of (7), we have

$$\begin{aligned}\|(\Delta\tilde{x}^k, \Delta\tilde{y}^k)\| &\le \|\nabla H_{\mu_k}(x^k, y^k)^{-1}\|\|H_{\mu_k}(x^k, y^k)\| \\ &\le C\rho_{\mu_k}(x^k, y^k) \\ &\le \beta C\mu_k,\end{aligned}$$

where the second inequality follows from result 3 of Lemma 1. Similarly, for the approximate Newton step, we have

$$\begin{aligned}\|(\Delta\hat{x}^k, \Delta\hat{y}^k)\| &\le \|\nabla H_{\mu_k}(x^k, y^k)^{-1}\|\|H_0(x^k, y^k)\| \\ &\le C(\rho_{\mu_k}(x^k, y^k) + nB\mu_k) \\ &\le C(\beta + nB)\mu_k,\end{aligned}$$

where the second inequality follows from result 4 of Lemma 1. □

We are now ready to show that the line search step length λ_k of the centering step is bounded below by a positive constant.

Proposition 2. *Let (x^k, y^k, μ_k) be the kth iterate of the continuation method. If $\|\nabla H_{\mu_k}(x^k, y^k)^{-1}\| \le C$ and $\rho_{\mu_k}(x^k, y^k) \ne 0$, then $\lambda_k \ge \bar{\lambda}$, where*

$$\bar{\lambda} = \alpha_1\tilde{\lambda} \quad \textit{and} \quad \tilde{\lambda} = \min\left\{1, \frac{1-\sigma}{\beta(\mu_0 L + 2A)C^2}\right\} > 0.$$

Proof. In view of the line search procedure, it suffices to show that inequality (8) holds for all $\lambda \in [0, \tilde{\lambda}] \subseteq [0, 1]$. Indeed,

$$\begin{aligned}&\rho_{\mu_k}(x^k + \lambda\Delta\tilde{x}^k, y^k + \lambda\Delta\tilde{y}^k) \\ &= \|F(x^k + \lambda\Delta\tilde{x}^k - (y^k + \lambda\Delta\tilde{y}^k)\| + \|\Psi_{\mu_k}(x^k + \lambda\Delta\tilde{x}^k, y^k + \lambda\Delta\tilde{y}^k)\| \\ &\le (1-\lambda)\rho_{\mu_k}(x^k, y^k) + (L + 2A/\mu_k)\lambda^2\|(\Delta\tilde{x}^k, \Delta\tilde{y}^k)\|^2 \\ &\le (1-\lambda)\rho_{\mu_k}(x^k, y^k) + (L\rho_{\mu_k}(x^k, y^k) + 2A\rho_{\mu_k}(x^k, y^k)/\mu_k)C^2\lambda^2\rho_{\mu_k}(x^k, y^k) \\ &\le (1-\lambda)\rho_{\mu_k}(x^k, y^k) + (\beta\mu_k L + 2\beta A)C^2\lambda^2\rho_{\mu_k}(x^k, y^k) \\ &\le (1-\lambda)\rho_{\mu_k}(x^k, y^k) + \beta(\mu_0 L + 2A)C^2\lambda^2\rho_{\mu_k}(x^k, y^k) \\ &\le (1-\sigma\lambda)\rho_{\mu_k}(x^k, y^k) \quad \text{for all } \lambda \in [0, \tilde{\lambda}],\end{aligned}$$

where the first inequality follows from Lemma 2 and the second inequality follows from result 1 of Lemma 3. □

We next show that the step length γ_k for reducing μ based on the centering step is also bounded below by a positive constant.

Proposition 3. *Let (x^k, y^k, μ_k) be the kth iterate of the continuation method. If $\|\nabla H_{\mu_k}(x^k, y^k)^{-1}\| \le C$, then $\gamma_k \ge \bar{\gamma}$, where*

$$\bar{\gamma} = \alpha_2\tilde{\gamma} \quad \textit{and} \quad \tilde{\gamma} = \min\left\{1, \frac{\beta\sigma\bar{\lambda}}{\beta + nB}\right\} > 0,$$

where $\bar{\lambda}$ has been defined in Proposition 2.

Proof. In view of the updating rule for μ_k, it suffices to show that

$$(\tilde{x}^{k+1}, \tilde{y}^{k+1}) \in \mathcal{N}(\beta, (1-\gamma)\mu_k) \quad \text{or} \quad \rho_{(1-\gamma)\mu_k}(\tilde{x}^{k+1}, \tilde{y}^{k+1}) \leq \beta(1-\gamma)\mu_k \tag{12}$$

holds for all $\gamma \in [0, \tilde{\gamma}] \subseteq [0, 1]$. Consider the case $\rho_{\mu_k}(x^k, y^k) \neq 0$ first:

$$\begin{aligned} \rho_{(1-\gamma)\mu_k}(\tilde{x}^{k+1}, \tilde{y}^{k+1}) &\leq \rho_{\mu_k}(\tilde{x}^{k+1}, \tilde{y}^{k+1}) + nB\gamma\mu_k \\ &\leq (1-\sigma\bar{\lambda})\rho_{\mu_k}(x^k, y^k) + nB\gamma\mu_k \\ &\leq (\beta(1-\sigma\bar{\lambda}) + nB\gamma)\mu_k \\ &\leq \beta(1-\gamma)\mu_k \quad \text{for all } \gamma \in [0, \tilde{\gamma}], \end{aligned}$$

where the first inequality follows from result 2 of Lemma 1 and the second inequality follows from Proposition 2. As a by-product of the above proof, one can see that the inequality (12) also holds for the case $\rho_{\mu_k}(x^k, y^k) = 0$. This can be verified by setting $\rho_{\mu_k}(\tilde{x}^{k+1}, \tilde{y}^{k+1}) = \rho_{\mu_k}(x^k, y^k) = 0$ in the right-hand side of the first inequality. □

We are now in the position to show the global linear convergence for the continuation method. We assume the algorithm does not terminate finitely. In addition, we make the following blanket assumption on the infinite sequence $\{(x^k, y^k, \mu_k)\}$ generated by the continuation method.

ASSUMPTION 3. *There is a constant $C > 0$ such that $\|\nabla H_{\mu_k}(x^k, y^k)^{-1}\| \leq C$ for all k.*

Conditions under which the assumption is satisfied will be discussed in section 6.

THEOREM 1. *Suppose Assumption* 3 *holds for the infinite sequence (x^k, y^k, μ_k) generated by the continuation method. Then*

1. *For all $k = 0, 1, \ldots$, we have*

$$\mu_k \leq \mu_0(1-\bar{\gamma})^k. \tag{13}$$

Thus, the sequence $\{\mu_k\}$ converges to 0 *globally and at least Q-linearly.*

2. *The sequence $\{\|\min\{x^k, F(x^k)\}\|\}$ converges to* 0 *globally and R-linearly.*
3. *The sequence $\{(x^k, y^k)\}$ is bounded and converges to a solution of the NCP.*

Proof. At each iteration k, the continuation method takes the approximate Newton step only if it reduces μ_k faster. Therefore,

$$\mu_{k+1} \leq \tilde{\mu}_{k+1} = (1-\gamma_k)\mu_k \leq (1-\bar{\gamma})\mu_k \text{ for all } k,$$

where the second inequality follows from Proposition 3 and Assumption 3. Result 1 then follows immediately.

For result 2, notice that

$$\|\min\{x^k, F(x^k)\} - \min\{x^k, y^k\}\| \leq \|F(x^k) - y^k\|.$$

Therefore,

$$\begin{aligned} \|\min\{x^k, F(x^k)\}\| &\leq \|\min\{x^k, y^k\}\| + \|F(x^k) - y^k\| \\ &= \rho_0(x^k, y^k) \\ &\leq \rho_{\mu_k}(x^k, y^k) + nB\mu_k \\ &\leq (\beta + nB)\mu_k, \end{aligned} \tag{14}$$

where the second inequality follows from result 2 of Lemma 1. Result 2 then follows from result 1 of this theorem.

For result 3, we first show that the sequence $\{(x^k, y^k)\}$ is bounded. Since at each iteration the continuation method takes either a centering step or an approximate Newton step, we have

$$\begin{aligned} \|(x^{k+1}, y^{k+1}) - (x^k, y^k)\| &\le \max\{\lambda_k \|(\Delta\tilde{x}^k, \Delta\tilde{y}^k)\|, \|(\Delta\hat{x}^k, \Delta\hat{y}^k)\|\} \\ &\le C(\beta + nB)\mu_k \\ &\le \mu_0 C(\beta + nB)(1-\bar{\gamma})^k, \end{aligned} \tag{15}$$

where the second inequality follows from Lemma 3 and the fact that $\lambda_k \le 1$. It follows that $\{(x^k, y^k)\}$ is a Cauchy sequence and therefore must be bounded and has a unique limit point. Let (x^*, y^*) be the limit point. Since $H_0(x, y)$ is continuous at (x^*, y^*), by result 5 of Lemma 1, $\rho_\mu(x, y)$ is continuous at $(x, y, \mu) = (x^*, y^*, 0)$. Since $(x^k, y^k) \in \mathcal{N}(\beta, \mu_k)$ for all k, we have $\rho_0(x^*, y^*) = 0$ by result 1 of this theorem. This implies that (x^*, y^*) is a solution of the NCP. □

Notice that result 3 of the above theorem does not imply that the NCP has a unique solution. Instead, it shows that the sequence generated by the continuation method converges to one of the solutions of the NCP. This result is stronger than some of the existing global convergence results for both interior point algorithms (e.g., [22]) and noninterior continuation methods (e.g., [1]) under similar assumptions, which state that any accumulation point is a solution of the NCP.

5. Local quadratic convergence. In this section, we first establish the local superlinear convergence for the approximate Newton step. We then show that the continuation method eventually switches to the approximate Newton step for all k sufficiently large. As a result, it converges quadratically to a solution of the NCP.

For the local quadratic convergence, we need the following strict complementarity assumption at the limit point of the continuation method.

ASSUMPTION 4. *The limit point (x^*, y^*) of the sequence $\{(x^k, y^k)\}$ generated by the continuation method satisfies the following strict complementarity condition:*

$$x_i^* + y_i^* > 0 \quad \textit{for all } i = 1, \ldots, n.$$

LEMMA 4. *Let (x^*, y^*) be the limit point of the continuation method. Under Assumption 4, there exists a neighborhood $N(x^*, y^*)$ of (x^*, y^*) such that*

1. *$H_0(x, y)$ is continuously differentiable and*

$$\|\nabla H_\mu(x, y) - \nabla H_0(x, y)\| \le (4nB/\epsilon)\mu$$

for all $(x, y) \in N(x^*, y^*)$, where $\epsilon = \min_i\{x_i^* + y_i^*\}$.

2. *In addition,*

$$\|H_0(\hat{x}, \hat{y}) - H_0(x, y) - \nabla H_0(x, y)((\hat{x}, \hat{y}) - (x, y))\| \le L\|(\hat{x}, \hat{y}) - (x, y)\|^2$$

for all $(x, y), (\hat{x}, \hat{y}) \in N(x^*, y^*)$, where L is the Lipschitz constant defined in Assumption 2.

Proof. By result 3 of Theorem 1, (x^*, y^*) is a solution of the NCP and $\min\{x^*, y^*\} = 0$. Then, under Assumption 4, there exists a neighborhood $N(x^*, y^*)$ of (x^*, y^*) such that

$$\min_i |x_i - y_i| \ge \epsilon/2 > 0 \quad \text{for all } (x, y) \in N(x^*, y^*).$$

By result 5 of Proposition 1, $\Psi_0(x,y)$ is continuously differentiable in the neighborhood, and so is $H_0(x,y)$. Now, let $(x,y) \in N(x^*,y^*)$. By result 5 of Proposition 1, we have

$$|p'_\mu(x_i - y_i) - p'_0(x_i - y_i)| \le B\mu/|x_i - y_i| \le (2B/\epsilon)\mu.$$

Therefore, based on the structure of ∇H_μ and ∇H_0, we have

$$\begin{aligned}&\|\nabla H_\mu(x,y) - \nabla H_0(x,y)\|\\ &\le \|P'_\mu(y-x) - P'_0(y-x)\| + \|P'_\mu(y-x) - P'_0(y-x)\|\\ &\le (4nB/\epsilon)\mu.\end{aligned}$$

To prove result 2, let $(x,y), (\hat{x},\hat{y}) \in N(x^*,y^*)$ and $J = \{i | x_i^* > 0, y_i^* = 0\}$ be the active set. Then

$$\nabla H_0(x,y) = \begin{bmatrix} \nabla F(x) & -I \\ I - W & W \end{bmatrix},$$

where $W = \mathrm{diag}\{w_i\}$ and $w_i = 1$ for all $i \in J$ and $w_i = 0$ otherwise. In addition, we have

$$\Psi_0(x,y) = (I-W)x + Wy \quad \text{and} \quad \Psi_0(\hat{x},\hat{y}) = (I-W)\hat{x} + W\hat{y}.$$

It follows, after some algebraic calculations, that

$$\begin{aligned}&\|H_0(\hat{x},\hat{y}) - H_0(x,y) - \nabla H_0(x,y)((\hat{x},\hat{y}) - (x,y))\|\\ &= \|F(\hat{x}) - F(x) - \nabla F(x)(\hat{x} - x)\|\\ &\le L\|\hat{x} - x\|^2\\ &\le L\|(\hat{x},\hat{y}) - (x,y)\|^2,\end{aligned}$$

where the first inequality follows from Assumption 2. □

We next show the local superlinear convergence for the approximate Newton step of the continuation method.

LEMMA 5. *Suppose the sequence $\{(x^k,y^k)\}$ generated by the continuation method satisfies Assumption 3 and the limit point (x^*,y^*) satisfies Assumption 4. The following properties hold for the approximate Newton step:*

1. $\|(\hat{x}^{k+1},\hat{y}^{k+1}) - (x^*,y^*)\| = o(\|(x^k,y^k) - (x^*,y^*)\|)$,
2. $\|H_0(\hat{x}^{k+1},\hat{y}^{k+1})\| = o(\|H_0(x^k,y^k)\|)$

for all k sufficiently large.

Proof. By the definition of $(\hat{x}^{k+1},\hat{y}^{k+1})$, we have

$$\begin{aligned}&\|(\hat{x}^{k+1},\hat{y}^{k+1}) - (x^*,y^*)\|\\ &= \|(x^k,y^k) - (x^*,y^*) - \nabla H_{\mu_k}(x^k,y^k)^{-1}H_0(x^k,y^k)\|\\ &\le \|\nabla H_{\mu_k}(x^k,y^k)^{-1}[(\nabla H_{\mu_k}(x^k,y^k) - \nabla H_0(x^k,y^k))((x^k,y^k) - (x^*,y^*))\\ &\quad - (H_0(x^k,y^k) - H_0(x^*,y^*) - \nabla H_0(x^k,y^k)((x^k,y^k) - (x^*,y^*)))]\|\\ &\le \|\nabla H_{\mu_k}(x^k,y^k)^{-1}\|[\|\nabla H_{\mu_k}(x^k,y^k) - \nabla H_0(x^k,y^k)\|\|(x^k,y^k) - (x^*,y^*)\|\\ &\quad + \|H_0(x^k,y^k) - H_0(x^*,y^*) - \nabla H_0(x^k,y^k)((x^k,y^k) - (x^*,y^*))\|].\end{aligned}$$

We now bound each part of the right-hand side. By Assumption 3, $\|H_{\mu_k}(x^k,y^k)^{-1}\| \le C$. Since (x^k,y^k) converges to (x^*,y^*), we have

$$\|\nabla H_{\mu_k}(x^k,y^k) - \nabla H_0(x^k,y^k)\| \le (4nB/\epsilon)\mu_k = o(1)$$

for all k sufficiently large, where the inequality follows from result 1 of Lemma 4 and the equality is true since μ_k converges to 0. Finally, by result 2 of Lemma 4, we have

$$\|H_0(x^k,y^k)-H_0(x^*,y^*)-\nabla H_0(x^k,y^k)((x^k,y^k)-(x^*,y^*))\| = O(\|(x^k,y^k)-(x^*,y^*)\|^2)$$

for k sufficiently large. Result 1 then follows immediately.

We now prove result 2. Based on the proof of Lemma 4, $H_0(x,y)$ is continuously differentiable in a neighborhood of (x^*,y^*) and $\nabla H_0(x^*,y^*)^{-1}$ exists. Using the inverse function theorem [17, Theorem 5.2.1], we conclude that H_0^{-1} is continuously differentiable and, therefore, Lipschitz continuous in the neighborhood of $H_0(x^*,y^*)$. Let L_1 and L_2 be the Lipschitz constants of H_0 and H_0^{-1} in the respective neighborhoods mentioned above. Since the sequence $\{(x^k,y^k,\mu_k)\}$ converges to $(x^*,y^*,0)$ by Theorem 1, we have, by definition of Lipschitz continuity,

$$\|H_0(\hat{x}^{k+1},\hat{y}^{k+1})\| \le L_1\|(\hat{x}^{k+1},\hat{y}^{k+1})-(x^*,y^*)\|$$

and

$$\|(x^k,y^k)-(x^*,y^*)\| \le L_2\|H_0(x^k,y^k)\|$$

for all k sufficiently large. Result 2 then follows from result 1 of this lemma. □

Based on the above proof, if $\mu_k = O(\|(x^k,y^k)-(x^*,y^*)\|)$, then the approximate Newton step has a quadratic convergence rate.

The next result is key to the local quadratic convergence for the continuation method. It shows that the continuation method eventually takes only the approximate Newton step.

LEMMA 6. *Suppose the sequence $\{(x^k,y^k)\}$ generated by the continuation method satisfies Assumption* 3 *and the limit point (x^*,y^*) satisfies Assumption* 4. *Then $(x^k,y^k)=(\hat{x}^k,\hat{y}^k)$ for all k sufficiently large.*

Proof. It suffices to show that

$$(\hat{x}^{k+1},\hat{y}^{k+1}) \in \mathcal{N}(\beta,(1-\gamma_k)\mu_k)$$

for all k sufficiently large. Denote

$$D_1 = \frac{(1-\alpha_2)(\beta-nB)}{2(\beta+nB)} > 0.$$

By result 2 of Lemma 5,

$$\|H_0(\hat{x}^{k+1},\hat{y}^{k+1})\| \le D_1\|H_0(x^k,y^k)\|$$

holds for all k sufficiently large. Therefore,

$$\begin{aligned}
\rho_{(1-\gamma_k)\mu_k}(\hat{x}^{k+1},\hat{y}^{k+1}) &\le 2\|H_0(\hat{x}^{k+1},\hat{y}^{k+1})\| + nB(1-\gamma_k)\mu_k\\
&\le 2D_1\|H_0(x^k,y^k)\| + nB(1-\gamma_k)\mu_k\\
&\le 2D_1(\rho_{\mu_k}(x^k,y^k)+nB\mu_k) + nB(1-\gamma_k)\mu_k\\
&\le (2D_1(\beta+nB)+(1-\gamma_k)nB)\mu_k\\
&= ((1-\alpha_2)(\beta-nB)+(1-\gamma_k)nB)\mu_k\\
&\le ((1-\gamma_k)(\beta-nB)+(1-\gamma_k)nB)\mu_k\\
&= \beta(1-\gamma_k)\mu_k,
\end{aligned}$$

where the second equality is true by choice of D_1, the first and third inequalities follow from results 5 and 4 of Lemma 1, respectively, and the last inequality follows from the fact that the maximum reduction of μ based on the centering step is by a factor of $(1-\alpha_2)$ (otherwise, $\gamma_k = 1$, and the continuation method terminates finitely). □

The next result provides a bound for μ_k, whenever it is generated by the approximate Newton step. This result will be used to prove local quadratic convergence.

LEMMA 7. *If the kth iterate (x^k, y^k, μ_k) of the continuation method is generated by the approximate Newton step, then*

$$\mu_k \le \frac{2}{\alpha_3(\beta - nB)} \|H_0(x^k, y^k)\|.$$

Proof. In view of the μ reduction procedure of the approximate Newton step, we have

$$(x^k, y^k) \notin \mathcal{N}(\beta, \alpha_3\mu_k).$$

Therefore,

$$\beta\alpha_3\mu_k \le \rho_{\alpha_3\mu_k}(x^k, y^k) \le 2\|H_0(x^k, y^k)\| + nB\alpha_3\mu_k,$$

where the second inequality follows from result 5 of Lemma 1. The result then follows by reorganizing the above inequality. □

We are now ready to show the local quadratic convergence for the continuation method.

THEOREM 2. *Suppose that the sequence $\{(x^k, y^k)\}$ generated by the continuation method satisfies Assumption 3 and the limit point (x^*, y^*) satisfies Assumption 4. Then*

1. $\|(x^{k+1}, y^{k+1}) - (x^*, y^*)\| = O(\|(x^k, y^k) - (x^*, y^*)\|^2)$,
2. $\|H_0(x^{k+1}, y^{k+1})\| = O(\|H_0(x^k, y^k)\|^2)$,
3. $\mu_{k+1} = O(\mu_k^2)$

for all k sufficiently large.

Proof. By Lemma 6, $(x^k, y^k) = (\hat{x}^k, \hat{y}^k)$ for all k sufficiently large. From Lemma 7, we have

$$\mu_k = O(\|H_0(x^k, y^k)\|) = O(\|(x^k, y^k) - (x^*, y^*)\|)$$

for all k sufficiently large. The second equality is true since H_0 is continuously differentiable and therefore Lipschitz continuous in the neighborhood of (x^*, y^*). Results 1 and 2 then follow from the remark after the proof of Lemma 5. For result 3, it suffices to show that there exists a $0 < t < \infty$ such that

$$(\hat{x}^{k+1}, \hat{y}^{k+1}) \in \mathcal{N}(\beta, t\mu_k^2) \tag{16}$$

holds for all k sufficiently large. By result 2 of Lemma 5, there exists a $D_2 > 0$ such that

$$\|H_0(\hat{x}^{k+1}, \hat{y}^{k+1})\| \le D_2\|H_0(x^k, y^k)\|^2$$

for all k sufficiently large. By following the proof of Lemma 6, we can show that condition (16) holds for

$$t = \frac{2D_2(\beta + nB)^2}{\beta - nB}.$$

Clearly, $0 < t < \infty$ since $\beta > nB$ by construction of the continuation method. □

6. Conditions that guarantee Assumption 3. In this section, we introduce a set of sufficient conditions that guarantee Assumption 3. We start by defining several special functions for NCPs.

DEFINITION 1. *Let S be a nonempty subset of R^n. The mapping $F : R^n \to R^n$ is said to be*

1. *a P_0-function on set S if for any $x \neq y$, $x, y \in S$, there is an index i such that*

$$x_i - y_i \neq 0 \text{ and } (F_i(x) - F_i(y))(x_i - y_i) \geq 0;$$

2. *a uniform P-function on set S if for some $\gamma > 0$,*

$$\max_i (F_i(x) - F_i(y))(x_i - y_i) \geq \gamma \|x - y\|^2 \quad \text{for all } x, y \in S;$$

3. *an R_0^w-function (w for weak) on set S if for any sequence $\{x^k\} \in S$ satisfying*

$$\|x^k\| \to \infty, \quad \limsup_{k\to\infty} [-x^k]_+ < \infty, \quad \limsup_{k\to\infty} [-F(x^k)]_+ < \infty,$$

where the inequalities are interpreted as componentwise, there is an index i such that

$$x_i^k \to \infty, \quad F_i(x^k) \to \infty.$$

It is well known that any uniform P-function is a P_0-function. The definition of R_0^w-function was recently introduced in [2]. It is a natural generalization of the concept of R_0-matrix for LCPs. Indeed, if $F(x) = Mx + q$, F is an R_0^w-function if and only if M is an R_0-matrix. In addition, any uniform P-function is also an R_0^w-function. See [2, 6] for proofs of these results and definitions and properties of other R_0-type functions.

We first introduce a set of conditions that guarantee the global convergence of the continuation method.

THEOREM 3. *Suppose the following conditions are satisfied:*

(C1) *F is a P_0-function.*

(C2) *The slice of neighborhood $\mathcal{N}(\beta, 0 < \mu \leq \mu_0)$ is bounded.*

Then the sequence $\{(x^k, y^k, \mu_k)\}$ generated by the continuation method is well defined and has at least one accumulation point, and every accumulation point is a solution of the NCP.

Proof. Under assumption (C1), Jacobian $\nabla H_\mu(x, y)$ is nonsingular for all $\mu > 0$ and $(x, y) \in R^{2n}$. (See [8] for the proof of a similar result.) Hence, the sequence $\{(x^k, y^k, \mu_k)\}$ is well defined. In addition, it has an accumulation point by assumption (C2). Since μ_k decreases monotonically and $\mu_0 \geq \mu_k > 0$, it converges to some $\mu^* \geq 0$. In fact, μ^* must be 0. Otherwise, with $\mu_k > \mu^* > 0$ and $\{(x^k, y^k)\}$ being bounded, $\|\nabla H_{\mu_k}(x^k, y^k)^{-1}\|$ is uniformly bounded above by assumption (C1). However, this implies that Assumption 3 is satisfied and μ_k converges to 0 by Theorem 1, which leads to a contradiction. It remains to show that any accumulation point (x^*, y^*) is a solution of the NCP. Indeed, since $(x^k, y^k) \in \mathcal{N}(\beta, \mu_k)$ and μ_k converges to 0, we have $\rho_0(x^*, y^*) = 0$. That is, (x^*, y^*) is a solution of the NCP. □

Although conditions (C1) and (C2) are sufficient for the global convergence, additional conditions are needed to guarantee the global linear convergence rate. Specifically, we need some nonsingularity conditions at each possible accumulation point so that $\nabla H_{\mu_k}(x^k, y^k)$ is well conditioned for all k sufficiently large. Let $\partial H_0(x, y)$ be the generalized derivative of H_0 at (x, y) in the sense of Clarke [11]. The following nonsingularity condition has been introduced and studied in [19, 20]:

(C3) $\|V^{-1}\| \leq C' < \infty$ for all $V \in \partial H_0(x^*, y^*)$ and for every solution (x^*, y^*) of the NCP.

We next show that $\nabla H_\mu(x, y)$ satisfies certain Jacobian consistence properties.

LEMMA 8. *For any sequence* $\{(x^k, y^k, \mu_k)\}$ *that converges to* $(x^*, y^*, 0)$, *we have*

$$\lim_{k\to\infty} \operatorname{dist}\{\nabla H_{\mu_k}(x^k, y^k), \partial H_0(x^*, y^*)\} = 0.$$

Proof. Due to the special structure of H_0, $\partial H_0(x^*, y^*)$ has the following explicit expression:

$$\partial H_0(x^*, y^*) = \begin{bmatrix} \nabla F(x^*) & -I \\ \partial P_0(y^* - x^*) & \partial P_0(x^* - y^*) \end{bmatrix},$$

where $\partial P_0(y^* - x^*) = \operatorname{diag}\{\partial p_0(y_i^* - x_i^*)\}$ and $\partial P_0(x^* - y^*) = \operatorname{diag}\{\partial p_0(x_i^* - y_i^*)\}$. By result 6 of Proposition 1, we have

$$\lim_{k\to\infty} \operatorname{dist}\{p'_{\mu_k}(y_i^k - x_i^k), \partial p_0(y_i^* - x_i^*)\} = 0$$

and

$$\lim_{k\to\infty} \operatorname{dist}\{p'_{\mu_k}(x_i^k - y_i^k), \partial p_0(x_i^* - y_i^*)\} = 0$$

for all $i = 1, \ldots, n$. The result then follows by comparing $\nabla H_0(x^k, y^k)$ and $\partial H_0(x^*, y^*)$ element by element. □

Notice that the concept of Jacobian consistence in the above result is slightly different from that defined in [9], where the concept was first introduced for a class of semismooth functions.

We are now ready to provide conditions that guarantee Assumption 3.

THEOREM 4. *Assumption* 3 *holds if conditions* (C1)–(C3) *are satisfied.*

Proof. It suffices to show that $\|\nabla H_{\mu_k}(x^k, y^k)^{-1}\|$ is bounded for all k sufficiently large. Since $\{(x^k, y^k, \mu_k)\}$ generated by the continuation method is bounded by condition (C2), there is at least an accumulation point. Let $\{(x^k, y^k, \mu_k)\}$, $k \in K$, be a convergent subsequence with an accumulation point (x^*, y^*, μ^*). By conditions (C1) and (C2) and Theorem 1, $\mu^* = 0$, and (x^*, y^*) is a solution of the NCP. In addition, since each solution (x^*, y^*) is locally unique by assumption (C3) and Proposition 2.5 of [19], sequence $\{(x^k, y^k)\}$ has only a finite number of such accumulation points. On the other hand, we have by Lemma 8:

$$\lim_{k\to\infty, k\in K} \operatorname{dist}\{\nabla H_{\mu_k}(x^k, y^k), \partial H_0(x^*, y^*)\} = 0.$$

Assumption (C3) then implies that $\|\nabla H_{\mu_k}(x^k, y^k)^{-1}\|$ is bounded for all k sufficiently large and $k \in K$. Since the same argument applies for all convergent subsequences, we obtain the desired result. □

To conclude the section, we show that some special classes of function F satisfy conditions (C1)–(C3).

PROPOSITION 4. *If F is an R_0^w-function, then condition* (C2) *is satisfied.*

Proof. Since $(x^k, y^k) \in \mathcal{N}(\beta, 0 < \mu \leq \mu_0)$ for all k, we have, by (14),

$$\|\min\{x^k, F(x^k)\}\| \leq (\beta + nB)\mu_0. \tag{17}$$

By Proposition 4 of [2], sequence $\{x^k\}$ is bounded. The result then follows from

$$\|y^k - F(x^k)\| \le \rho_{\mu_k}(x^k, y^k) \le \beta\mu_k \le \beta\mu_0$$

and the fact that F is a continuous function. □

Proposition 5. *If F is a uniform P-function, then conditions* (C1)–(C3)*, and therefore Assumption* 3*, are satisfied. In addition, x^k converges to x^* globally and R-linearly, where (x^*, y^*) is the unique solution of the NCP and $\{(x^k, y^k)\}$ is the sequence generated by the continuation method.*

Proof. Notice that any uniform P-function is both a P_0- and an R_0^w-function. In addition, it is straightforward to show that $\partial H_0(x, y)$ is nonsingular for all (x, y) if F is a uniform P-function. As a result, conditions (C1)–(C3) are satisfied. The second part is based on the following global error bound (see [6, 15]) for the uniform P-function: there exists a constant $E > 0$ such that

$$\|x - x^*\| \le E\|\min\{x, F(x)\}\| \quad \text{for all } x \in R^n.$$

Thus, by inequality (14), we have

$$\|x^k - x^*\| \le (\beta + nB)E\mu_k$$

and the result follows from Theorem 1. □

As a final remark, we want to point out that the conditions in Theorem 4 are by no means necessary for Assumption 3, which assures the global linear convergence for the continuation method. While the nonsingularity of $\nabla H_{\mu_k}(x^k, y^k)$ is guaranteed for all bounded $\{(x^k, y^k)\}$ under condition (C1), it is needed only within the neighborhood $\mathcal{N}(\beta, 0 < \mu \le \mu_0)$. Indeed, as long as the whole sequence $\{(x^k, y^k)\}$ is well defined and all of its accumulation points have nonsingular generalized derivatives, the global linear convergence holds.

Acknowledgments. The first author would like to thank Pei-Lei Jiang for her careful reading of an early draft of this paper. In particular, her suggestions led to the improvement of result 3 of Theorem 1. We are also grateful to two anonymous referees for their careful reading of the paper and constructive comments.

REFERENCES

[1] J. Burke and S. Xu, *The global linear convergence of a noninterior path-following algorithm for linear complementarity problem*, Math. Oper. Res., 23 (1998), pp. 719–734.

[2] B. Chen, *Error Bounds for R_0-Type and Monotone Nonlinear Complementarity Problems*, Technical Report, Department of Management and Systems, Washington State University, Pullman, 1997.

[3] B. Chen and P. T. Harker, *A noninterior-point continuation method for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1168–1190.

[4] B. Chen and P. T. Harker, *A noninterior continuation method for quadratic and linear programming*, SIAM J. Optim., 3 (1993), pp. 503–515.

[5] B. Chen and P. T. Harker, *A continuation method for monotone variational inequalities*, Math. Programming, 69 (1995), pp. 237–253.

[6] B. Chen and P. T. Harker, *Smooth approximations to nonlinear complementarity problems*, SIAM J. Optim., 7 (1997), pp. 403–420.

[7] C. Chen and O. L. Mangasarian, *Smoothing methods for convex inequalities and linear complementarity problems*, Math. Programming, 71 (1995), pp. 51–69.

[8] C. Chen and O. L. Mangasarian, *A class of smoothing functions for nonlinear and mixed complementarity problems*, Comput. Optim. Appl., 5 (1996), pp. 97–138.

[9] X. Chen, L. Qi, and D. Sun, *Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities*, Math. Comput., 67 (1998), pp. 519–540.

[10] X. Chen and Y. Ye, *On Homotopy-Smoothing Methods for Variational Inequalities*, AMR 96/39, Applied Mathematics Report, School of Mathematics, The University of New South Wales, Sydney, Australia, 1996.

[11] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley, New York, 1983; reprinted by SIAM, Philadelphia, 1990.

[12] P. T. Harker and J. S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 161–120.

[13] C. Kanzow, *Some noninterior continuation methods for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 851–868.

[14] C. Kanzow, *A new approach to continuation methods for complementarity problems with uniform P-function*, Oper. Res. Lett., 20 (1997), pp. 85–92.

[15] C. Kanzow and M. Fukushima, *Equivalence of the generalized complementarity problem to differentiable unconstrained minimization*, J. Optim. Theory Appl., 90 (1996), pp. 581–603.

[16] C. Kanzow and H. Jiang, *A continuation method for (strongly) monotone variational inequalities*, Math. Programming, 81 (1998), pp. 103–125.

[17] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, San Diego, 1970.

[18] J. S. Pang, *Complementarity problems*, in Handbook of Global Optimization, R. Horst and P. Pardalos, eds., Kluwer, Boston, 1995, pp. 271–338.

[19] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.

[20] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–367.

[21] S. Smale, *Algorithms for solving equations*, in Proceedings of the International Congress of Mathematicians, Berkeley, CA, 1986, pp. 172–195.

[22] P. Tseng, *An infeasible path-following method for monotone complementarity problems*, SIAM J. Optim., 7 (1997), pp. 386–402.

[23] S. Wright and D. Ralph, *A superlinear infeasible-interior-point algorithm for monotone complementarity problems*, Math. Oper. Res., 21 (1996), pp. 815–838.

[24] S. Xu, *The Global Linear Convergence of an Infeasible Non-Interior Path-Following Algorithm for Complementarity Problems with Uniform P-Functions*, Preprint, Department of Mathematics, University of Washington, Seattle, 1996.

SIAM J. OPTIM.
Vol. 9, No. 3, pp. 624–645

A GLOBAL AND LOCAL SUPERLINEAR CONTINUATION-SMOOTHING METHOD FOR P_0 AND R_0 NCP OR MONOTONE NCP*

BINTONG CHEN† AND XIAOJUN CHEN‡

Abstract. We propose a continuation method for a class of nonlinear complementarity problems (NCPs), including the NCP with a P_0 and R_0 function and the monotone NCP with a feasible interior point. The continuation method is based on a class of Chen–Mangasarian smoothing functions. Unlike many existing continuation methods, the method follows noninterior smoothing paths, and, as a result, initial points can be easily constructed. In addition, we introduce a procedure to dynamically update the neighborhoods associated with the smoothing paths, so that the algorithm is both globally convergent and locally superlinearly convergent under suitable assumptions. Finally, a hybrid continuation-smoothing method is proposed and is shown to have the same convergence properties under weaker conditions.

Key words. smoothing method, global and superlinear convergence, P_0 and R_0 function, monotone function, complementarity problem

AMS subject classification. 90C33

PII. S1052623497321109

1. Introduction.

Let $F : R^n \to R^n$ be a continuously differentiable function. The nonlinear complementarity problem (NCP), denoted by NCP(F), is to find a vector $(x, y) \in R^n \times R^n$ such that

$$F(x) - y = 0,$$
$$x \geq 0, \quad y \geq 0, \quad x^T y = 0.$$

If F is an affine function of x, then NCP(F) reduces to a linear complementarity problem (LCP). The NCP is considered a fundamental problem for optimization theory since the optimality condition of many continuous optimization problems can be formulated as an NCP. In the last few decades, many algorithms have been developed to solve various NCPs. See [19, 27] for a comprehensive survey. In this paper, we are interested in developing a noninterior continuation method to solve NCP(F).

In general, a continuation method uses a smooth function to approximate NCP(F) as a family of parameterized smooth equations, solves the smooth equations "approximately" at each iteration, and refines the smooth approximation as the iterates progress towards (hopefully) a solution of the NCP. In many cases, the set of solutions of the smooth equations forms a path as the smoothing parameter is reduced to zero. We call the set of solutions the smoothing path of the continuation method. The continuation method is closely related to, and in many cases identical to, the homotopy methods in numerical analysis literature [26], the path-following algorithms in interior point algorithm literature [17, 24, 25, 42], and many smoothing methods developed recently [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] and [18, 20, 22, 30, 34, 40, 41]. The well-known interior point path-following algorithm can be considered a continuation

*Received by the editors May 6, 1997; accepted for publication (in revised form) April 7, 1998; published electronically April 19, 1999.

http://www.siam.org/journals/siopt/9-3/32110.html

†Department of Management and Systems, Washington State University, Pullman, WA 99164-4736 (chenbi@wsu.edu).

‡Department of Mathematics and Computer Science, Shimane University, Matsue 690-8504, Japan. This author's work was supported in part by the Australian Research Council.

method, where all iterates are confined within the positive orthant. In that case, the smoothing path reduces to the central path, the term often used in the interior point algorithm literature. Many interior point algorithms have been shown to solve a subclass of NCPs, including linear programs and quadratic programs, in polynomial time. See [17] for a survey of interior point algorithms. On the other hand, the so-called noninterior continuation methods require neither the smoothing path nor the initial and intermediate iterates to be in the positive orthant. As a result, they are more flexible than interior point algorithms and very convenient for numerical implementation. Indeed, it has been demonstrated that many noninterior continuation methods developed so far are competitive with, and in a number of cases superior to, interior point algorithms in terms of numerical performance. In particular, Billups, Dirkse, and Ferris [1] compared several numerical methods for mixed NCPs, and their numerical results showed that the *smooth code* was comparable to the best noninterior point codes available (e.g., the PATH solver [15, 33]). Furthermore, since noninterior smoothing methods are defined in a larger domain than the neighborhood of interior point methods, the sequence generated by these methods can have a superlinear or quadratic convergence rate without the strict complementarity condition. However, on the theoretical side, many open questions remain to be answered for the noninterior continuation methods.

We review some of the important progress made related to the noninterior continuation methods. Chen and Harker [5] introduced the first noninterior continuation method to solve the LCP with a P_0 and R_0 matrix. They concentrated on establishing the properties of the smoothing path. Later, Kanzow [22] improved the method by refining the smoothing function, sometimes referred to as the Chen–Harker–Kanzow–Smale (CHKS) smoothing function, and showing global convergence for the continuation method. Chen and Qi [10, 30] introduced some other smoothing functions and proposed a globally and locally linearly convergent smoothing method for a class of nonsmooth equations, which includes the NCP as a special case. Chen and Mangasarian [8] developed a smoothing method to solve LCPs and linear inequalities based on the neural network function, a double integration of the sigmoid function. Later, they [9] introduced a family of smoothing functions, which unified the smoothing functions studied in [5, 22, 10, 8]. Gabriel and Moré [18] further generalized the Chen–Mangasarian smooth function family to solve variational inequality problems with box constraints. Chen, Qi, and Sun [11] discovered the Jacobian consistency property for the Gabriel–Moré smooth function family. This property allowed us to design a globally and superlinearly convergent smoothing Newton method. Chen and Ye [12] introduced a hybrid switch technique that allows their smoothing method to converge finitely for LCPs. Burke and Xu [2] and Xu [40] introduced a notion of neighborhood for their continuation methods, which allowed them to establish the global linear convergence for the LCP with a P_0 and R_0 matrix and for the NCP with a uniform P function. Recently, Chen and Xiu [7] improved the Burke–Xu continuation method by simplifying the neighborhood definition and adding an approximate Newton step. They obtained both global linear convergence and local quadratic convergence for a class of NCPs. Chen and Chen [4] further simplified the neighborhood definition and the continuation method given in [7]. In particular, they introduced a nonhybrid continuation method that converges globally linearly and locally quadratically and a hybrid continuation method with the above convergence properties but without the strict complementarity assumption.

The smoothing paths in all of the above-mentioned continuation methods for

NCPs are confined in the positive orthant [6, 12]. As a result, they all require a relatively large bounded neighborhood or level set so that they can start from an arbitrary point. This is usually achieved by assuming F to be an R_0 function (R_0 matrix in the case of LCP). However, this assumption is not satisfied by many NCPs derived from convex optimization problems. In particular, the matrix associated with the LCP derived from a linear program is not an R_0 matrix. Therefore, the above-mentioned continuation methods are not guaranteed to work for convex optimization problems. On the other hand, the interior point algorithms often make the following assumption on the NCP: F is monotone and there exists a feasible interior point, i.e., an $x > 0$ such that $F(x) > 0$. In contrast, this assumption is often satisfied by NCPs derived from convex optimization problems. For the monotone problem with a feasible interior point, Chen and Ye [12] proved the existence of a bounded level set. Xu [41] showed the existence of a slice of bounded neighborhoods. Chen and Chen [4] improved Xu's results and unified the existence of the bounded level set and the bounded neighborhood. These results provide a theoretical tool for constructing a noninterior continuation-smoothing method for the monotone NCP with a feasible interior point. However, in practice, it is difficult to find an initial point within the bounded level set or the bounded neighborhood that depends on the information of the feasible interior point.

The gap between the interior and noninterior continuation methods has been observed by Hotta and Yoshise [20], and their recent work has laid down the theoretical foundation for continuation methods based on noninterior smoothing paths. Specifically, they showed the existence, uniqueness, and other structural properties of the noninterior smoothing paths for NCPs based on the CHKS smoothing function. They then developed a continuation method based on a noninterior smoothing path. Their algorithm was shown to converge globally for a class of NCPs including the monotone NCP with a feasible interior point. However, their neighborhood definition in the algorithm is quite restrictive, and no convergence rate was obtained.

The current paper attempts to solve some of the above-mentioned problems. In particular, we propose a continuation-smoothing method that follows noninterior smoothing paths based on a class of Chen–Mangasarian smoothing functions. As a result, we are able to easily construct a (noninterior) initial point and define the associated neighborhood. By adjusting the neighborhoods dynamically, we show that the continuation-smoothing method converges globally and locally superlinearly for a class of NCPs, including the NCP with a P_0 and R_0 function and the monotone NCP with a feasible interior point. In addition, we also propose a hybrid continuation method that possesses the above convergence properties under weaker assumptions. To the best of our knowledge, the continuation-smoothing method is the first noninterior continuation method that achieves global and local superlinear convergence for both the NCP with a P_0 and R_0 function and the monotone NCP with a feasible interior point.

The paper is organized as follows. Section 2 summarizes the properties of the Chen–Mangasarian smoothing function needed for this paper and studies the properties of the noninterior smoothing path and the associated neighborhood. Section 3 proposes a continuation-smoothing method based on the noninterior smoothing paths and proves the global and superlinear convergence for the continuation method. Section 4 proposes a hybrid continuation-smoothing method, which is shown to converge globally and locally superlinearly under weaker assumptions.

We now briefly describe the notation to be used in this paper. R^n_{++}, R^n_+, and R^n_-

stand for the positive, the nonnegative, and the nonpositive orthant of R^n, respectively. $\|\cdot\|$ denotes the 1-, 2-, or ∞-norm as well as its induced matrix norm. All vectors are column vectors. For simplicity, we sometimes use z for the column vector $(x^T, y^T)^T$. We use $\text{vec}\{x_i\}$ for a vector whose ith element is x_i and $\text{diag}\{x_i\}$ for a diagonal matrix with iith entry equal to x_i. We use e for a vector with all entries equal to 1 and I for a diagonal matrix with all diagonal entries equal to 1. Finally, unless noted otherwise, $f_1'(x, y)$ stands for the partial derivative of f with respect to the first argument x.

In addition, the following definitions related to function F will be used in the paper.

Definition 1.1. *The mapping $F : R^n \to R^n$ is said to be*

1. *a P_0 function if for all x^1, $x^2 \in R^n$ with $x^1 \neq x^2$, there is an index i such that*
$$x_i^1 \neq x_i^2 \quad \text{and} \quad (F_i(x^1) - F_i(x^2))(x_i^1 - x_i^2) \geq 0;$$
2. *a uniform P function if for some $\gamma > 0$,*
$$\max_{1 \leq i \leq n} (F_i(x^1) - F_i(x^2))(x_i^1 - x_i^2) \geq \gamma \|x^1 - x^2\|^2 \text{ for all } x^1, x^2 \in R^n;$$
3. *a monotone function if*
$$(F(x^1) - F(x^2))^T (x^1 - x^2) \geq 0 \text{ for all } x^1, x^2 \in R^n;$$
4. *an R_0 function if for any sequence $\{x^k\} \subset R^n$ satisfying $\{\|x^k\|\} \to \infty$ and*
$$\liminf_{k \to \infty} \frac{\min_i x_i^k}{\|x^k\|} \geq 0, \quad \liminf_{k \to \infty} \frac{\min_i F_i(x^k)}{\|x^k\|} \geq 0,$$
there exists an index j such that
$$\{x_j^k\} \to \infty \quad \text{and} \quad \{F_j(x^k)\} \to \infty.$$

It is well known that any monotone function or uniform P function is a P_0 function. The definition of the R_0 function was introduced in [37] and later modified in [6]. It can be viewed as a natural generalization of the concept of the R_0 matrix for LCPs. It has been shown in [6] that any uniform P function is also an R_0 function.

2. Smoothing functions, smoothing paths, and neighborhoods.

In this section, we first introduce a subclass of the Chen–Mangasarian smooth function family to be used in this paper and summarize its properties. We then approximate NCP(F) by a smoothing function and study the properties of the smoothed NCP. Based on these properties, we define a noninterior smoothing path and the associated neighborhood. Finally, we provide a simple procedure to construct an initial point within the neighborhood.

2.1. The Chen–Mangasarian smoothing function.

The Chen–Mangasarian smoothing function approximates the plus function s_+ by a double integration of a probability density function ρ. In this paper, we choose to use a subclass of the Chen–Mangasarian smoothing function, whose probability density function ρ satisfies the following assumption.

Assumption 2.1.

(A1) ρ is a continuous function such that $0 < \rho(t) \leq A < \infty$ and $\rho(t) = \rho(-t)$ for all $t \in (-\infty, +\infty)$,

(A2) $\sup_{\tau \geq 0} \int_{\tau}^{+\infty} \tau t \rho(t) dt = B_2^2 < \infty$.

Condition (A2) was introduced in [12]. Clearly, Assumption 2.1 implies that

$$0 < B_1 \equiv \int_0^{\infty} t \rho(t) dt < \infty. \tag{2.1}$$

In addition, Assumption 2.1 is satisfied by the density functions associated with the most commonly used smoothing functions, such as the neural network smoothing function, the CHKS smoothing function, and the normal smoothing function. Gabriel and Moré [18] proved that if condition (2.1) is satisfied, then the Chen–Mangasarian smoothing function is equivalent to a special case of the Gabriel–Moré smoothing function:

$$p(s, \mu) = \int_{-\infty}^{+\infty} (s - \mu t)_+ \rho(t) dt, \tag{2.2}$$

where $0 \leq \mu < \infty$ is a smoothing parameter. Clearly, as μ approaches zero, the smoothing function $p(s, \mu)$ approaches the plus function and $p_0(s) \equiv p(s, 0) = s_+$ for any $s \in R$. The properties of the above subclass of the Chen–Mangasarian smoothing function are summarized below.

Proposition 2.1. *Under assumption* (A1) *and condition* (2.1)*, the following properties hold for the smoothing function $p(s, \mu)$ defined in* (2.2).

1. *For any fixed $\mu > 0$, $p(s, \mu) > 0$ is continuously differentiable, strictly increasing, and convex with respect to s.*
2. *$p(s, \mu) - p(-s, \mu) = s$ for all $s \in R$ and $\mu \geq 0$. As a result,*

$$0 < p_1'(s, \mu) = \int_{-\infty}^{s/\mu} \rho(t) dt < 1 \quad \textit{and} \quad p_1'(-s, \mu) = 1 - p_1'(s, \mu)$$

for all $s \in R$ and $\mu > 0$.

3. *For any fixed $s \in R$, $p(s, \mu)$ is strictly increasing and continuous at $\mu = 0$. In particular, it is continuously differentiable, strictly increasing, and convex with respect to $\mu > 0$. Thus, $p(s, \mu) \to \infty$ if $\mu \to \infty$.*
4. *$p(s, \mu_2) - p(s, \mu_1) \leq B_1(\mu_2 - \mu_1)$ for all $s \in R$ and $\mu_2 \geq \mu_1 \geq 0$.*
5. *Let $p^0(s) \equiv \lim_{\mu \downarrow 0} p_1'(s, \mu)$. Then $p^0(s) \in \partial p_0(s)$, where $\partial p_0(s)$ stands for the generalized derivative of $p_0(\cdot)$ at s in the sense of Clark* [13]; *i.e.,*

$$\partial p_0(s) = \begin{cases} \{1\} & s > 0, \\ \{0\} & s < 0, \\ \{[0, 1]\} & s = 0. \end{cases}$$

If $s = 0$ then

$$p_1'(s, \mu) = p^0(s) = \frac{1}{2}.$$

Otherwise, $p_0(\cdot)$ is differentiable at s and

$$p^0(s) = p_0'(s) = \begin{cases} 0 & \textit{if } s < 0, \\ 1 & \textit{if } s > 0. \end{cases}$$

In addition, we have

$$|p_1'(s,\mu) - p^0(s)| \le B_1\mu/|s|.$$

6. *Let* $v(s+h) \in \partial p_0(s+h)$*; then*

$$|p_0(s+h) - p_0(s) - v(s+h)h| = O(|h|^r)$$

for all $r > 0$*; i.e., for all* h *sufficiently small, we have*

$$|p_0(s+h) - p_0(s) - v(s+h)h| = 0.$$

If, in addition, assumption (A2) *is satisfied, then*

$$p(s,\mu)p(-s,\mu) \le (B_1^2 + B_2^2)\mu^2 \tag{2.3}$$

for all $s \in R$ *and* $\mu \ge 0$.

Proof. Results 1 and 4–6 of the proposition have been shown in [4, 6, 7, 9].

The first part of result 2 clearly holds for $\mu = 0$ since $s_+ = p(s,0)$ by definition. We may assume without loss of generality that $s, \mu > 0$. By definition,

$$\begin{aligned} p(s,\mu) - p(-s,\mu) &= \int_{-\infty}^{s/\mu} (s-\mu t)\rho(t)dt - \int_{-\infty}^{-s/\mu} (-s-\mu t)\rho(t)dt \\ &= s\int_{-\infty}^{s/\mu} \rho(t)dt + s\int_{s/\mu}^{+\infty} \rho(t)dt - \mu\int_{-\infty}^{s/\mu} t\rho(t)dt + \mu\int_{-\infty}^{-s/\mu} t\rho(t)dt \\ &= s, \end{aligned}$$

where the third equality uses the fact that ρ is a symmetric function. The second part of the result 2 then follows immediately from result 1.

For result 3, $p(s,\mu)$ is continuous with respect to μ by definition. Suppose now that $\mu > 0$. If $s \le 0$, then

$$p(s,\mu) > p(s,0) = 0$$

by result 1 of this proposition. If $s > 0$, then

$$p(s,\mu) - p(s,0) = p(s,\mu) - s = p(-s,\mu) > 0$$

by results 1 and 2 of this proposition. Therefore, $p(s,u)$ is strictly increasing with respect to μ at $\mu = 0$. For the second part of result 3, notice that

$$p(s,\mu) = \int_{-\infty}^{s/\mu} (s-\mu t)\rho(t)dt$$

and p is continuously differentiable with respect to μ. Straightforward calculation shows that

$$\frac{\partial p(s,\mu)}{\partial \mu} = -\int_{-\infty}^{s/\mu} t\rho(t)dt = \int_{s/\mu}^{+\infty} t\rho(t)dt > 0$$

and

$$\frac{\partial^2 p(s,\mu)}{\partial^2 \mu} = \frac{s^2}{\mu^3}\rho\left(\frac{s}{\mu}\right) > 0.$$

Therefore, p is strictly increasing and convex with respect to $\mu > 0$, and the last part of result 3 follows immediately.

Finally, we show that inequality (2.3) holds under additional assumption (A2). The result is clearly true for $\mu = 0$, and we assume again that $s, \mu > 0$. Indeed,

$$\begin{aligned} p(s,\mu)p(-s,\mu) &= \left(\int_{-\infty}^{s/\mu}(s-\mu t)\rho(t)dt\right)\left(\int_{-\infty}^{-s/\mu}(-s-\mu t)\rho(t)dt\right) \\ &= \mu^2\left(\int_{-\infty}^{s/\mu} t\rho(t)dt\right)\left(\int_{-\infty}^{-s/\mu} t\rho(t)dt\right) - \mu s\left(\int_{-\infty}^{s/\mu}\rho(t)dt\right)\left(\int_{-\infty}^{-s/\mu} t\rho(t)dt\right) \\ &\quad - s^2\left(\int_{-\infty}^{s/\mu}\rho(t)dt\right)\left(\int_{-\infty}^{-s/\mu}\rho(t)dt\right) + \mu s\left(\int_{-\infty}^{s/\mu} t\rho(t)dt\right)\left(\int_{-\infty}^{-s/\mu}\rho(t)dt\right) \\ &\le \mu^2\left(\int_{-\infty}^{s/\mu} t\rho(t)dt\right)\left(\int_{-\infty}^{-s/\mu} t\rho(t)dt\right) - \mu s\left(\int_{-\infty}^{-s/\mu} t\rho(t)dt\right) \\ &\le \mu^2\left(\int_{s/\mu}^{+\infty} t\rho(t)dt\right)^2 + \mu^2\left(\int_{s/\mu}^{+\infty}\frac{s}{\mu}t\rho(t)dt\right) \\ &\le \mu^2(B_1^2 + B_2^2), \end{aligned}$$

where the first inequality uses the facts that ρ is a symmetric function and the last two terms after the second equality are nonpositive, and the last inequality follows from assumption (A2). ◻

We assume that the smooth function used in the remainder of the paper satisfies Assumption 2.1.

2.2. Smoothed NCP. It is well known that NCP(F) can be written as the following system of nonsmooth equations:

$$H_0(z) \equiv \begin{pmatrix} F(x) - y \\ x - (x-y)_+ \end{pmatrix} = 0. \tag{2.4}$$

Let $a \in R^n_{++}$ be a vector of smoothing parameters. By using the Chen–Mangasarian smoothing function, we may approximate $(x-y)_+$ by

$$P(x-y, a) \equiv \mathrm{vec}\{p(x_i - y_i, a_i)\}.$$

Denoting

$$\Psi(x, y, a) \equiv x - P(x-y, a),$$

we obtain the following smooth approximation of NCP(F):

$$H(z, a) \equiv \begin{pmatrix} F(x) - y \\ \Psi(x, y, a) \end{pmatrix} = 0. \tag{2.5}$$

Clearly, if $a = 0$, (2.5) reduces to (2.4). Thus, we can also write NCP(F) as $H_0(z) \equiv H(z, 0) = 0$. By result 1 of Proposition 2.1, $H(\cdot, a)$ is continuously differentiable in R^{2n} for any fixed $a \in R^n_{++}$. Let

$$P'(x-y, a) \equiv \mathrm{diag}\{p_1'(x_i - y_i, a_i)\}.$$

The Jacobian of $H(\cdot, a)$ is given by

$$H'(z, a) = \begin{pmatrix} F'(x) & -I \\ I - P'(x-y, a) & P'(x-y, a) \end{pmatrix}.$$

It is well known that $H'(\cdot, a)$ is nonsingular for any fixed $a \in R^n_{++}$ if F is a P_0 function.

Based on the properties of the subclass of the Chen–Mangasarian smoothing functions, we have the following results for Ψ and H. Similar results were obtained in [4, 6, 12]. However, some of our proofs below are much simpler.

LEMMA 2.1.

1. *If $\Psi(x, y, a) = 0$ for some $a \in R^n_{++}$, then*
$$(x, y) \in R^{2n}_{++} \quad \textit{and} \quad x^T y \le (B_1^2 + B_2^2) a^T a.$$
2. *For any $(x, y) \in R^{2n}_{+}$ ($(x, y) \in R^{2n}_{++}$), there is a unique vector $a \in R^n_+$ ($a \in R^n_{++}$) such that $\Psi(x, y, a) = 0$. In addition, for any $(x, y) \in R^{2n}$, and a closed set $A \subseteq R^n_+$, if $\{\|\Psi(x, y, a)\| \mid a \in A\}$ is bounded, then A is bounded.*
3. *If $\Psi(x, y, a) = v$ for some $(x, y) \in R^{2n}$, $v \in R^n$, and $a \in R^n_+$, then*
$$\Psi(x - v, y - v, a) = 0.$$
4. *Let $a^1, a^2 \in R^n_+$. Then for all $z \in R^{2n}$*
$$|\|H(z, a^1)\| - \|H(z, a^2)\|| \le B_1 \|a^1 - a^2\|.$$

Proof. By result 1 of Proposition 2.1, $\Psi(x, y, a) = 0$ implies $x = P(x - y, a) > 0$. By result 2 of Proposition 2.1, we also have $y = P(y - x, a) > 0$. Therefore,
$$x^T y = P(x - y, a)^T P(y - x, a),$$
and the second part of result 1 then follows from (2.3) in Proposition 2.1. By assumption of result 2, $P(x - y, 0) = x \in R^n_+$. Result 2 follows immediately from result 3 of Proposition 2.1. Indeed, the first part of result 2 is true since $p(s, \mu)$ is strictly increasing with respect to μ for $\mu \ge 0$, and the second part of result 2 is true since $\mu \to \infty$ implies $p(s, \mu) \to \infty$. By definition,
$$\Psi(x - v, y - v, a) = x - v - P((x - v) - (y - v), a) = \Psi(x, y, a) - v = 0,$$
and this proves result 3. For result 4, we have
$$\begin{aligned} |\|H(z, a^1)\| - \|H(z, a^2)\|| &\le \|H(z, a^1) - H(z, a^2)\| \\ &\le \|\Psi(z, a^1) - \Psi(z, a^2)\| \\ &\le B_1 \|a^1 - a^2\|, \end{aligned}$$
where the last inequality follows from result 4 of Proposition 2.1. □

We next characterize the range of H, denoted by
$$H(R^{2n}, R^n_{++}) = \{H(z, a) : \ z \in R^{2n}, a \in R^n_{++}\}.$$
Similar results have been obtained by Hotta and Yoshise [20] based on the CHKS smoothing function.

LEMMA 2.2.

1. *$H(R^{2n}, R^n_{++})$ is an open subset of R^{2n}.*
2. *If $(w, v) \in H(R^{2n}, R^n_{++})$, then*
$$(w, v) + R^{2n}_- \subset H(R^{2n}, R^n_{++}).$$
3. *In particular, if $(0, 0) \in H(R^{2n}, R^n_{++})$, then*
$$R^{2n}_- \subset H(R^{2n}, R^n_{++}).$$

Proof. The proof follows from result 3 of Lemma 2.1 and the same proof techniques used for Lemma 2.1 in [20]. □

2.3. Assumptions on NCP(F). Let us denote

$$Q(z,a) \equiv (H(z,a),a) \in R^{2n} \times R^n_+.$$

We make the following assumption on function F throughout the remainder of the paper.

ASSUMPTION 2.2.

1. F is a P_0 function.
2. NCP(F) has a feasible interior point, i.e., there is a vector $(x,y) > 0$ such that $y = F(x)$.
3. The set
$$Q^{-1}(D) \equiv \{(z,a) \in R^{2n} \times R^n_+ : \; Q(z,a) \in D\}$$
is bounded for every compact subsets $D \subset H(R^{2n}, R^n_{++}) \times R^n_+$.
4. There exists an $\epsilon > 0$ such that the level set
$$L(\epsilon) \equiv \{z \in R^{2n} : \; \|H_0(z)\| \leq \epsilon\}$$
is bounded.

Notice that conditions 1, 2, 3 of the above assumption are identical to condition 2.2 used in Hotta and Yoshise [20]. Using result 3 of Lemma 2.1, and the proof technique originally developed by Kojima, Megiddo, and Noma [25], and later used to show Corollary 3.5 in [6] and Theorem 2.10 in [20], we can obtain the following result.

THEOREM 2.1. *Under conditions* 1 *and* 3 *of Assumption* 2.2*, the mapping* Q *maps* $R^{2n} \times R^n_{++}$ *onto* $H(R^{2n} \times R^n_{++}) \times R^n_{++}$ *homeomorphically.*

The following two propositions show that Assumption 2.2 is implied by the two most popular assumptions used in the NCP literature.

PROPOSITION 2.2. *Assumption* 2.2 *holds if* F *is a* P_0 *and* R_0 *function.*

Proof. We show in sequence that conditions 4, 3, and 2 of Assumption 2.2 hold. By Proposition 3.12 in [6], if F is an R_0 function, then x is bounded if $\|\min\{x, F(x)\}\|$ is bounded. Since F is a continuous function and

$$\|\min\{x, F(x)\}\| \leq \|\min\{x,y\}\| + \|F(x) - y\| \leq 2\|H_0(z)\|, \tag{2.6}$$

condition 4 holds for all $0 \leq \epsilon < \infty$. Condition 3 then follows from the above result and the fact that

$$\|H_0(z)\| \leq \|H(z,a)\| + B_1\|a\|.$$

In addition, the above inequality implies that the level set

$$L'(\epsilon) \equiv \{z \in R^{2n} : \; \|H(z,a)\| \leq \epsilon\}$$

is bounded for all $0 \leq \epsilon < \infty$. It follows that $\frac{1}{2}\|H(z,a)\|^2$ has a bounded stationary point $\bar{z}$ such that

$$H'(\bar{z},a)^T H(\bar{z},a) = 0.$$

Since F is a P_0 function, $H'(\bar{z},a)$ is nonsingular. Hence, $H(\bar{z},a) = 0$ and $\bar{z} \in R^{2n}$ is a feasible interior point by definition of H. Therefore, condition 2 holds. □

PROPOSITION 2.3. *Assumption* 2.2 *holds if* F *is a monotone function and NCP*(F) *has a feasible interior point.*

Proof. Conditions 1 and 2 of Assumption 2.2 clearly hold. Condition 3 follows from results 1 and 3 of Proposition 2.1 and the same proof used for Lemma 2.3 in [20]. Condition 4 follows from result 2 of Corollary 1 in [4]. □

2.4. Smoothing paths, neighborhoods, and initial points. Given a smoothing vector $a \in R^n_{++}$, let $z(\mu, a)$ be the unique solution of equation

$$H(z, \mu a) + \mu e = 0. \tag{2.7}$$

The smoothing path associated with the vector a is defined as a set of solutions $z(\mu, a)$ for all $\mu > 0$; i.e.,

$$\mathcal{S}(a) = \{z : \ H(z, \mu a) + \mu e = 0, \mu > 0\}.$$

In addition, we choose the following neighborhood around the smoothing path:

$$\mathcal{N}(a, \beta) = \{z : \ \|H(z, \mu a) + \mu e\| \leq \beta\mu, \mu > 0\},$$

where $\beta \in (0, 1]$ is called the width of the neighborhood. The slice of the neighborhood with $\mu \in U \subset R_{++}$ is then given by

$$\mathcal{N}(a, \beta, U) = \{z : \ \|H(z, \mu a) + \mu e\| \leq \beta\mu, \mu \in U\}.$$

Notice that our neighborhood definition is simpler than those proposed in Hotta and Yoshise [20]. The next result shows the boundedness of certain slices of the neighborhood.

PROPOSITION 2.4. *Let $a \in R^n_{++}$ and $\beta \in (0, 1]$. Under conditions* 2 *and* 3 *of Assumption* 2.2*, the slice of neighborhood $\mathcal{N}(a, \beta, U)$ is bounded for all bounded $U \subset R_+$.*

Proof. By condition 2 of Assumption 2.2, there is a feasible interior point $\bar{z} = (\bar{x}, \bar{y}) > 0$ such that $\bar{y} = F(\bar{x})$. By result 2 of Lemma 2.1, there is an $\bar{a} \in R^n_{++}$ such that $H(\bar{z}, \bar{a}) = 0$. Therefore, $(0, 0) \in H(R^{2n}, R^n_{++})$. By result 3 of Lemma 2.2, $R^{2n}_- \subset H(R^{2n}, R^n_{++})$. Let

$$D = \{(w, v, \mu a) : \ -(1 + \beta)\mu e \leq (w, v) \leq -(1 - \beta)\mu e, \mu \in \text{closure}(U)\}.$$

Since $U \subset R_+$ is bounded, D is compact. In addition,

$$D \subset R^{2n}_- \times R^n_+ \subset H(R^{2n}, R^n_{++}) \times R^n_+.$$

By condition 3 of Assumption 2.2, $Q^{-1}(D)$ is bounded. On the other hand, based on the definition of D, we have

$$\mathcal{N}(a, \beta, U) \subset Q^{-1}(D).$$

Hence, the slice of the neighborhood is bounded. □

Moreover, Assumption 2.2 also ensures that the smoothing path $\mathcal{S}(a)$ is well defined for all $a \in R^n_{++}$ and leads to a solution of NCP(F) as μ approaches 0.

THEOREM 2.2. *Let $a \in R^n_{++}$ be arbitrary. Under conditions* 1, 2, *and* 3 *of Assumption* 2.2,

1. *$z(\mu, a)$ exists and is unique for all $\mu > 0$; in addition, $z(\mu, a)$ is continuous in μ and thus $\mathcal{S}(a)$ forms a trajectory;*
2. *$z(\mu, a)$ has a limiting point as $\mu \to 0$, and every limiting point is a solution of NCP(F).*

Proof. From the proof of Proposition 2.4, we have $R^{2n}_{-} \subset H(R^{2n}, R^n_{++})$ under condition 2 of Assumption 2.2. Since $\mu a \in R^n_{++}$ and $-\mu e \in R^{2n}_{-}$ for all $\mu > 0$, it follows from Theorem 2.1 that (2.7) has a unique solution under conditions 1 and 3 of Assumption 2.2. Result 1 then follows immediately. To show result 2, let $U = (0, \bar{\mu}]$ for some $0 < \bar{\mu} < \infty$. Then $z(\mu, a) \in \mathcal{N}(a, \beta, U)$ for all $\mu \in U$. Since the neighborhood $\mathcal{N}(a, \beta, U)$ is bounded by Proposition 2.4, $z(\mu, a)$ has an accumulation point z^* as $\mu \to 0$. By the continuity of H, $H(z^*, 0) = 0$ and z^* is a solution of NCP(F). □

To start a continuation method, it is required to have an initial point within a chosen neighborhood. Such an initial point is often assumed to be readily available by most existing continuation methods, interior or noninterior, that follow interior smoothing paths [25, 37, 39, 41]. This is especially true for those algorithms designed to solve the monotone NCP. It is important, however, for the purpose of both theoretical study and numerical implementation, to be able to construct an initial point within the neighborhood. We next provide a simple procedure to construct a noninterior initial point $z^0 \in R^{2n}$ such that $z^0 \in \mathcal{N}(a, \beta, (0, \mu_0])$ for some $a \in R^n_{++}$, $\mu_0 \in R_{++}$, and $\beta \in (0, 1]$.

Procedure 2.1.

1. *Choose any* $x^0 \in R^n$.
2. *Choose* $\mu_0 > \max_i \max\{0, -x^0_i, -F_i(x^0)/2\}$.
3. *Let* $y^0 = F(x^0) + \mu_0 e$.
4. *Find* $a \in R^n_{++}$ *such that*

$$p(x^0_i - y^0_i, \mu_0 a_i) = x^0_i + \mu_0 \quad \textit{for all } i = 1, 2, \ldots, n.$$

Notice that $a \in R^n_{++}$, if it exists, can be obtained by solving n one-dimensional equations. The next result guarantees the existence of the above initial point.

Proposition 2.5. *The initial point* (z^0, μ_0) *and the smoothing vector* $a \in R^n_{++}$ *constructed by Procedure* 2.1 *exist. In particular, they satisfy*

$$H(z^0, \mu_0 a) = -\mu_0 e.$$

Proof. It suffices to show that the vector a exists. Based on the choice of μ_0, we have

$$x^0_i + \mu_0 > 0 \quad \text{and} \quad x^0_i + \mu_0 > x^0_i - F_i(x^0) - \mu_0 = x^0_i - y^0_i$$

for all i. Therefore,

$$x^0_i + \mu_0 > (x^0_i - y^0_i)_+ = p(x^0_i - y^0_i, 0)$$

for all i. Moreover, by result 3 of Proposition 2.1, $p(x^0_i - y^0_i, \mu_0 a) \to \infty$ as $a \to \infty$. Hence the continuity of p implies that there exists a unique $a_i > 0$ such that

$$x^0_i + \mu_0 = p(x^0_i - y^0_i, \mu_0 a_i)$$

for each i. The result follows from the definition of H. □

Based on the above result, $z^0 \in \mathcal{N}(a, \beta, (0, \mu_0])$ holds as long as the smoothing vector a is a good approximate solution of the n one-dimensional equations in step 4 of Procedure 2.1.

3. The continuation method and its convergence. In this section, we propose a continuation method based on the noninterior smoothing path and the associated neighborhood defined in the previous section. By adjusting the smoothing vector a dynamically, we are able to show that the method converges globally under Assumption 2.2, and locally superlinearly under some regularity assumptions.

Algorithm 3.1. Given $\beta \in (0, 1]$, $\sigma, \eta \in (0, 1)$, $\tau \in (0, \eta)$, and $\alpha_i \in (0, 1)$ for $i = 1, 2$.

Step 0 (Initialization)

Choose any $x^0 \in R^n$ and $\mu_0 > \| \min\{x^0, F(x^0)\}\|$. Set $k = 0$.

Step 1 (Neighborhood Construction)

Let $y^k = F(x^k) + \mu_k e$. Choose $a^k \in R^n_{++}$ such that

$$\|H(z^k, \mu_k a^k) + \mu_k e\| \le \beta\mu_k. \tag{3.1}$$

Step 2 (Centering Step)

If $H(z^k, \mu_k a^k) + \mu_k e = 0$, set $\tilde{z}^{k+1} = z^k$ and go to step 4. Otherwise, Let $\Delta\tilde{z}^k$ solve the equation

$$H(z^k, \mu_k a^k) + \mu_k e + H'(z^k, \mu_k a^k)\Delta\tilde{z}^k = 0. \tag{3.2}$$

Step 3 (Line Search)

Let λ_k be the maximum of the values $1, \alpha_1, \alpha_1^2, \ldots$ such that

$$\|H(z^k + \lambda_k\Delta\tilde{z}^k, \mu_k a^k) + \mu_k e\| \le (1 - \sigma\lambda_k)\|H(z^k, \mu_k a^k) + \mu_k e\|. \tag{3.3}$$

Set $\tilde{z}^{k+1} = z^k + \lambda_k\Delta\tilde{z}^k$.

Step 4 (μ Reduction)

Let γ_k be the maximum of the values $\alpha_2, \alpha_2^2, \ldots$ such that

$$\|H(\tilde{z}^{k+1}, (1 - \gamma_k)\mu_k a^k) + (1 - \gamma_k)\mu_k e\| \le \beta(1 - \gamma_k)\mu_k. \tag{3.4}$$

Set $\tilde{\mu}_{k+1} = (1 - \gamma_k)\mu_k$.

Step 5 (Approximate Newton Step)

Let $\Delta\hat{z}^k$solve the equation

$$H_0(z^k) + H'(z^k, \mu_k a^k)\Delta\hat{z}^k = 0. \tag{3.5}$$

Set $\hat{z}^{k+1} = z^k + \Delta\hat{z}^k$.

Step 6 (Determining Next Iterate)

If $\|H_0(\hat{z}^{k+1})\| = 0$, stop. $\hat{z}^{k+1}$ is a solution of NCP(F). If $\|H_0(\hat{z}^{k+1})\| > \frac{1}{2}\tau\mu_k$, let $z^{k+1} = \tilde{z}^{k+1}$, $\mu_{k+1} = \tilde{\mu}_{k+1}$, and $a^{k+1} = a^k$. Set $k = k+1$ and go to Step 2. Otherwise, let $x^{k+1} = \hat{x}^{k+1}$ and $\mu_{k+1} = \frac{2}{\eta}\|H_0(\hat{z}^{k+1})\|$. Set $k = k+1$ and go to Step 1.

A few remarks about Algorithm 3.1:

- The algorithm may start from any initial point $x^0 \in R^n$.
- The algorithm differs from the existing interior or noninterior continuation methods in that it dynamically adjusts the smoothing vector $a \in R^n_{++}$ and the associated neighborhood. This unique feature allows us to show both global and local superlinear convergence for a broader class of NCPs, including the NCP with a P_0 and R_0 function and the monotone NCP with a feasible interior point.

- Similar to the continuation method developed in [7], the matrices inverted in the centering step and the approximate Newton step are identical. As a result, the additional computation time for the approximate Newton step is minimal.

PROPOSITION 3.1. *If F is a P_0 function, Algorithm* 3.1 *is well defined.*

Proof. We verify that each step of the algorithm is well defined. Step 0 is clearly well defined. For Step 1, if $k = 0$, we have

$$\mu_k > \|\min\{x^k, F(x^k)\}\| \geq \max_i \max\{0, -x_i^k, -F_i(x^k)/2\}.$$

If $k > 0$, then based on Step 6, we have

$$\mu_k = \frac{2}{\eta}\|H_0(\hat{z}^k)\| \geq \frac{1}{\eta}\|\min\{\hat{x}^k, F(\hat{x}^k)\}\| > \|\min\{x^k, F(x^k)\}\|,$$

where the first inequality follows from (2.6) and the second inequality holds since $\|H_0(\hat{z}^k)\| \neq 0$. Therefore, it follows from Proposition 2.5 that in either case we can find an $a^k \in R^n_{++}$ such that condition (3.1) holds. Step 2 is well defined since $\mu_k a^k \in R^n_{++}$ and F is a P_0 function and, therefore, $H'(z^k, \mu_k a^k)$ is nonsingular. Step 3 is well defined since $\Delta\tilde{z}^k \neq 0$ by construction and therefore is a strictly descent direction of $\|H(\cdot, \mu_k a^k) + \mu_k e\|$ at z^k. As a result, the line search procedure is finite by construction. Step 4 is well defined since if $H(z^k, \mu_k a^k) + \mu_k e = 0$ then $z^{k+1} = z^k$ and

$$\|H(z^{k+1}, \mu_k a^k) + \mu_k e\| = 0 < \beta\mu_k;$$

otherwise,

$$\|H(z^{k+1}, \mu_k a^k) + \mu_k e\| < \|H(z^k, \mu_k a^k) + \mu_k e\| \leq \beta\mu_k,$$

because $\lambda_k > 0$ from the line search step. Therefore, in either case, the μ reduction step terminates finitely by construction. □

We now show the global convergence of the continuation method.

THEOREM 3.1. *Let $\{(z^k, \mu_k)\}$ be a sequence generated by Algorithm* 3.1. *Then under Assumptions* 2.2

1. *the sequence $\{\mu_k\}$ decreases monotonically and converges to* 0 *as $k \to \infty$,*
2. *the sequence $\{z^k\}$ is bounded and any accumulation point of the sequence is a solution of NCP*(F).

Proof. By construction of the algorithm, μ_{k+1} is reduced by at least a factor of either $\tau/\eta \in (0, 1)$ or $(1 - \gamma_k) \in (0, 1)$. Thus, $\{\mu_k\}$ is a monotonically decreasing sequence. Since μ_k is nonnegative, it converges to some $\bar{\mu} \geq 0$.

If $\bar{\mu} = 0$, we have result 1. Suppose on the contrary that $\bar{\mu} > 0$. This implies $\gamma_k \to 0$ and the iteration index set

$$K \equiv \left\{k > 0 : \ \|H_0(\hat{z}^k)\| \leq \frac{1}{2}\tau\mu_{k-1}\right\} \tag{3.6}$$

is finite. As a result, there is a $k_0 > 0$ such that for all $k \geq k_0$, the sequence $\{(z^k, \mu_k)\}$ is essentially generated by Steps 2–4 with a fixed $\bar{a} \in R^n_{++}$; i.e., $z^k = \tilde{z}^k$, $\mu_k = \tilde{\mu}_k$, and $a^k = \bar{a}$ for all $k \geq k_0$. Now consider the sequence $\{(z^k, \mu_k)\}_{k \geq k_0}$. By Proposition 2.4, the sequence $z^k \in \mathcal{N}(\bar{a}, \beta, (0, \mu_0])$ is bounded. Taking a subsequence if necessary, we may assume that the sequence $\{z^k\}$ converges to some $\bar{z}$. Based on the μ reduction step, we have

$$\left\| H\left(z^k, \left(1-\frac{1}{\alpha_2}\gamma_{k-1}\right)\mu_{k-1}\bar{a}\right) + \left(1-\frac{1}{\alpha_2}\gamma_{k-1}\right)\mu_{k-1}e \right\| \geq \beta\left(1-\frac{1}{\alpha_2}\gamma_{k-1}\right)\mu_{k-1}.$$

Since $\gamma_k \to 0$, by passing limits on both sides, we have

$$\|H(\bar{z},\bar{\mu}\bar{a}) + \bar{\mu}e\| \geq \beta\bar{\mu} > 0. \tag{3.7}$$

Let $\Delta\bar{z}$ be the solution of (3.2) at $(\bar{z},\bar{\mu})$. Since $\bar{\mu} > 0$ and $\bar{z}$ is bounded, $H'(\bar{z},\bar{\mu}\bar{a})$ is nonsingular and $\Delta\bar{z}$ is well defined. In addition, in view of (3.7), it is a strictly descent direction for $\|H(\cdot,\bar{\mu}\bar{a})+\bar{\mu}e\|$ at $\bar{z}$. As a result, the corresponding linear search step length $\bar{\lambda}$ and μ reduction step length $\bar{\gamma}$ are both bounded below by a positive constant. On the other hand, the function H as well as its Jacobian H' are continuous in a neighborhood of $(\bar{z},\bar{\mu})$. It follows that Δz^k converges to $\Delta\bar{z}$ and therefore γ_k must be uniformly bounded below by some positive constant for all k sufficiently large. However, this contradicts the assumption that $\gamma_k \to 0$. Therefore, $\mu_k \to 0$ and this proves result 1.

For result 2, if the index set K is finite, then based on the above proof for result 1, the sequence $\{z^k\}$ is bounded, $a^k = \bar{a}$ for all $k \geq k_0$ and $\mu_k \to 0$. In view of (3.1), any accumulation point is a solution of the equation $H_0(z) = 0$ and therefore a solution of NCP(F). Now suppose that the index set K is infinite. We show first that result 2 holds for the subsequence $\{z^k\}_{k\in K}$ and then extend the proof to the whole sequence. By construction of Algorithm 3.1,

$$x^k = \hat{x}^k \quad \text{and} \quad y^k = F(\hat{x}^k) + \mu_k e \quad \text{for all } k \in K.$$

Since $\mu_k \to 0$ by result 1, $\|\hat{z}^k - z^k\| \to 0$ for all $k \in K$. Therefore,

$$\lim_{k\in K}\|H_0(z^k)\| = \lim_{k\in K}\|H_0(\hat{z}^k)\| = \frac{1}{2}\eta\lim_{k\in K}\mu_k = 0. \tag{3.8}$$

By condition 4 of Assumption 2.2, the sequence $\{z^k\}_{k\in K}$ is bounded. Clearly, any of its accumulation points is a solution of $H_0(z) = 0$.

To extend the result to the whole sequence, we need to show $\mu_k a^k \to 0$. Let K consist of $k_1 < k_2 < \cdots$. Since the sequence $\{z^{k_j}\}$ is bounded, $\mu_k \to 0$, and

$$\|H(z^{k_j}, \mu_{k_j}a^{k_j})\| \leq (\beta + \|e\|)\mu_{k_j},$$

the sequence $\{\mu_{k_j}a^{k_j}\}$ is bounded by result 2 of Lemma 2.1. Let $\bar{b} \in R^n_+$ be any accumulation point of $\{\mu_{k_j}a^{k_j}\}$ and $\bar{z}$ be the corresponding accumulation point of $\{z^{k_j}\}$. Then

$$H(\bar{z},\bar{b}) = 0.$$

Since it also holds that $H_0(\bar{z}) = 0$, we have $\bar{z} \in R^{2n}_+$. By result 2 of Lemma 2.1, $\bar{b} = 0$ is the unique vector that satisfies the above equation. It follows that $\mu_{k_j}a^{k_j} \to 0$ as $j \to \infty$. Now let k be any iteration index. Since K is infinite, there is a $k_j \in K$ such that

$$\mu_k a^k = \mu_k a^{k_j} \leq \mu_{k_j}a^{k_j}.$$

Thus, $\mu_k a^k \to 0$ as $k \to \infty$.

We are now ready to show result 2 for the whole sequence. By result 4 of Lemma 2.1, we have

$$\|H_0(z^k)\| \le \|H(z^k, \mu_k a^k)\| + B_1\|\mu_k a^k\| \le (\beta + \|e\|)\mu_k + B_1\|\mu_k a^k\| \to 0.$$

Therefore, $\{z^k\}$ is bounded by condition 4 of Assumption 2.2. In addition, any accumulation point is a solution of $H_0(z) = 0$ and therefore a solution of NCP(F). □

From Theorem 3.1, any accumulation point z^* of the sequence $\{z^k\}$ generated by Algorithm 3.1 is a solution of NCP(F). Additional assumptions are needed on the accumulation point z^* so that Algorithm 3.1 converges locally superlinearly.

ASSUMPTION 3.1. All elements in the generalized Jacobian $\partial H_0(z^*)$ are nonsingular.

ASSUMPTION 3.2. The strict complementarity condition holds at z^*; i.e., $x^* + y^* > 0$.

Notice that the above assumptions are identical to Condition 7.1 (i) in [25], the strong nondegeneracy assumption in [37], and Assumption 2 in [39].

PROPOSITION 3.2. *Let z^* be a solution of NCP*(F).

1. *Under Assumption* 3.1*, if F is a P_0 function, then z^* is the unique solution of NCP*(F).
2. *Under Assumption* 3.2*, $H_0(\cdot)$ is differentiable at z^* and*

$$\|H'(z, \mu a) - H_0'(z^*)\| = O(\mu\|a\|)$$

for all z in a neighborhood of z^*.

Proof. For result 1, consider function $H(z, \mu e)$, which maps R^{2n+1} to R^{2n}. By assumption, z^* is a solution of NCP(F) and thus $H(z^*, 0) = 0$. By Assumption 3.1, all elements in $\partial_z H(z^*, 0)$ are nonsingular. Since H is a Lipschitzian function, we can apply the generalized implicit function theorem [13, section 7.1] to H at $(z, \mu) = (z^*, 0)$. It follows that there exists a neighborhood U of μ at 0 and a Lipschitzian function $z : U \to R^{2n}$ such that $z(0) = z^*$ and $H(z(\mu), \mu e) = 0$ for all $\mu \in U$. Since F is a P_0 function, $z(\mu)$ is unique for all $\mu > 0$ by Theorem 2.1. Since $z(\mu)$ converges to z^* as $\mu \to 0$, z^* is unique.

Result 2 follows from results 5 and 6 of Proposition 1. Also see Lemma 4 in [7]. □

THEOREM 3.2. *Under Assumptions* 2.2, 3.1, *and* 3.2, *the sequence $\{(z^k, \mu_k)\}$ generated by Algorithm* 3.1 *converges to $(z^*, 0)$ locally superlinearly; i.e.,*

1. $\|z^{k+1} - z^*\| = o(\|z^k - z^*\|)$,
2. $\mu_{k+1} = o(\mu_k)$.

Proof. That $\{z^k\}$ converges to z^* follows from result 1 of Proposition 3.2 and Theorem 3.1. For superlinear convergence, we first show

$$\|\hat{z}^{k+1} - z^*\| = o(\|z^k - z^*\|).$$

From the proof of Theorem 3.1, $\mu_k a^k \to 0$. Hence, by Assumption 3.1 and result 2 of Proposition 3.2, there is a $C > 0$ such that

$$\|H'(z^k, \mu_k a^k)^{-1}\| \le C$$

holds for all k sufficiently large. Therefore,

$$\begin{aligned}\|\hat{z}^{k+1} - z^*\| &= \|z^k - z^* - H'(z^k, \mu_k a^k)^{-1} H_0(z^k)\| \\ &\le \|H'(z^k, \mu_k a^k)^{-1}\| \|H'(z^k, \mu_k a^k)(z^k - z^*) - H_0(z^k)\| \\ &\le C(\|H_0(z^k) - H_0(z^*) - H_0'(z^k)(z^k - z^*)\| \\ &\qquad + \|H'(z^k, \mu_k a^k) - H_0'(z^k)\| \|z^k - z^*\|) \\ &\le C(\|H_0(z^k) - H_0(z^*) - H_0'(z^k)(z^k - z^*)\| + O(\|\mu_k a^k\|)\|z^k - z^*\|) \\ &= o(\|z^k - z^*\|),\end{aligned}$$

where the last inequality follows from result 2 of Proposition 3.2. In addition, we have

$$\|H_0(\hat{z}^{k+1})\| = o(\|H_0(z^k)\|) \tag{3.9}$$

based on the proof of Theorem 3.1 in [29].

For result 1, it remains to show that Algorithm 3.1 eventually reduces to the approximate Newton step for all large k. To this end, we show first that the index set K defined by (3.6) is infinite. Suppose on the contrary that K is finite and $\bar{k} \in K$ is the largest iteration index. Then $a^k = \bar{a} \equiv a^{\bar{k}}$ for all $k \ge \bar{k}$. In addition, by result 4 of Lemma 2.1, we have

$$\begin{aligned}\|H_0(z^k)\| &\le \|H(z^k, \mu_k a^k) + \mu_k e\| + \mu_k(\|e\| + B_1\|a^k\|) \\ &\le \mu_k(\beta + \|e\| + B_1\|a^k\|) \\ &= \mu_k(\beta + \|e\| + B_1\|\bar{a}\|)\end{aligned}$$

for all $k \ge \bar{k}$. In view of (3.9),

$$\|H_0(\hat{z}^{k+1})\| \le \frac{1}{2}\tau\mu_k$$

holds for all $k \ge \bar{k}$ and k sufficiently large. However, this implies the expansion of set K, a contradiction to the assumption that K is finite.

Since both sequences $\{z^k\}$ and $\{\hat{z}^k\}$ converge to z^* and H_0 is Lipschitzian in a neighborhood of z^*, there exist a constant $L > 0$ and an integer $\bar{k}_1$ such that for all $k \ge \bar{k}_1$

$$\|H_0(\hat{z}^k) - H_0(z^k)\| \le L\|\hat{z}^k - z^k\|.$$

Let

$$\epsilon = \frac{\tau}{\eta + L(\eta + 2\|e\|)}.$$

By (3.9), there exists a large $\bar{k}_2$ such that for all $k \ge \bar{k}_2$,

$$\|H_0(\hat{z}^{k+1})\| \le \epsilon\|H_0(z^k)\|.$$

Let $\bar{k} = \max\{\bar{k}_1, \bar{k}_2\}$. Suppose now that z^k is generated by the approximate Newton step for some $k \ge \bar{k}$. Then

$$\mu_k = \frac{2}{\eta}\|H_0(\hat{z}^k)\|.$$

This implies

$$\begin{aligned}\|\hat{z}^k - z^k\| &= \|\hat{y}^k - y^k\|\\ &\le \|\hat{y}^k - F(\hat{x}^k)\| + \|e\|\mu_k\\ &\le \|H_0(\hat{z}^k)\| + \frac{2\|e\|}{\eta}\|H_0(\hat{z}^k)\|.\end{aligned}$$

Hence

$$\begin{aligned}\|H_0(\hat{z}^{k+1})\| &\le \epsilon\|H_0(z^k)\|\\ &\le \epsilon(\|H_0(\hat{z}^k)\| + L\|\hat{z}^k - z^k\|)\\ &\le \frac{\tau}{\eta}\|H_0(\hat{z}^k)\|\\ &\le \frac{1}{2}\tau\mu_k.\end{aligned}$$

Thus, the next iterate z^{k+1} will also be generated by the approximate Newton step. It follows that Algorithm 3.1 will eventually choose the approximate Newton step only for all large k. Result 1 then follows from

$$\begin{aligned}\|z^{k+1} - z^*\| &\le \|\hat{z}^{k+1} - z^*\| + \|\hat{z}^{k+1} - z^{k+1}\|\\ &\le o(\|z^k - z^*\|) + \left(1 + \frac{2\|e\|}{\eta}\right)\|H_0(\hat{z}^{k+1})\|\\ &\le o(\|z^k - z^*\|) + \left(1 + \frac{2\|e\|}{\eta}\right)(\|H_0(\hat{z}^{k+1})\| - \|H_0(z^*)\|)\\ &\le o(\|z^k - z^*\|) + L\left(1 + \frac{2\|e\|}{\eta}\right)\|\hat{z}^{k+1} - z^*\|\\ &\le o(\|z^k - z^*\|).\end{aligned}$$

Result 2 follows from (3.9) and the fact that $z^k = \hat{z}^k$ and $\mu_k = \frac{2}{\eta}\|H_0(z^k)\|$ for all large k. □

We conclude this section by commenting on the importance of updating the smoothing vector a dynamically. Based on the proof of Theorem 3.1, Steps 2–4 in Algorithm 3.1 with a fixed smoothing vector $a \in R^n_{++}$ would be sufficient for the algorithm to converge globally. In fact, the algorithm would have a global linear convergence rate under certain conditions, which can be shown by following the similar arguments used in [4, 7]. However, by restricting all iterates within a fixed neighborhood associated with a, the algorithm may be prevented from achieving the local superlinear convergence. To the best of our knowledge, Algorithm 3.1 is the first noninterior continuation method that dynamically updates the neighborhood and achieves both global and local superlinear convergence for the monotone NCP. Most of the existing continuation methods, such as those studied in [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] and [20, 22, 40], have fixed smoothing vectors and neighborhoods. As a result, they all have certain difficulties with the monotone NCP, either in finding a proper initial point or in establishing the local convergence rate.

4. A hybrid method. For many nonsmooth equation-based algorithms, Assumption 3.1 is sufficient for locally superlinear convergence. However, continuation-based algorithms often require in addition that the strict complementarity condition holds at the solution. For example, Assumption 3.2 is needed to show Theorem 3.2

for Algorithm 3.1 in this paper. This is because the Jacobian consistence property [11]

$$\mathrm{dist}\{H'(z^k, \mu_k a^k), \partial H_0(z^k)\} \to 0, \quad \text{as} \quad k \to \infty,$$

required for superlinear convergence, in general does not hold for the approximate Newton step without Assumption 3.2. Based on this observation, we present a hybrid continuation-smoothing method by replacing the approximate Newton step in Algorithm 3.1 by a smoothing Newton step that satisfies the Jacobian consistence property. We show that the hybrid method achieves the same convergence properties as Algorithm 3.1 but without the need of Assumption 3.2.

Let

$$P^0(x-y) \equiv \mathrm{diag}\{p^0(x_i - y_i)\}.$$

Then

$$H^0(z) \equiv \lim_{\epsilon \downarrow 0} H'(z, \epsilon e) = \begin{pmatrix} F'(x) & -I \\ I - P^0(x-y) & P^0(x-y) \end{pmatrix}.$$

By result 5 of Proposition 1,

$$H^0(z) \in \partial H_0(z)$$

for any $z \in R^{2n}$. Furthermore, from result 6 of Proposition 1 and the continuous differentiability of F, we have

$$\lim_{h \to 0} \frac{H_0(z+h) - H_0(z) - H^0(z+h)h}{\|h\|} = 0. \tag{4.1}$$

The following observation is instrumental for designing a smoothing Newton step in the hybrid method.

Proposition 4.1. *Let $z = (x, y) \in R^{2n}$ and $\epsilon > 0$. If $x = y$, then $H'(z, \epsilon e) = H^0(z)$. Otherwise,*

$$\|H'(z, \epsilon e) - H^0(z)\| \le \frac{2B_1 \epsilon}{\delta(z)},$$

where $\delta(z) = \min_i\{|x_i - y_i|,\ x_i - y_i \neq 0, i = 1, \ldots, n\}$.

Proof. From the definitions of $H'(z, \epsilon e)$ and $H^0(z)$, we have

$$H'(z, \epsilon e) - H^0(z) = \begin{pmatrix} 0 & 0 \\ -P'(x-y, \epsilon e) + P^0(x-y) & P'(x-y, \epsilon e) - P^0(x-y) \end{pmatrix}.$$

The first part then follows immediately from the fact that $p_1'(0, \epsilon) = p^0(0) = 1/2$. For the second part, we have

$$\begin{aligned} \|H'(z, \epsilon e) - H^0(z)\| &\le 2\|P'(x-y, \epsilon e) - P^0(x-y)\| \\ &= 2\|\mathrm{diag}\{p_1'(x_i - y_i, \epsilon) - p^0(x_i - y_i)\}\| \\ &\le 2\|I\| B_1 \epsilon / \delta(z) \\ &= 2B_1 \epsilon / \delta(z), \end{aligned}$$

where the last inequality follows from result 5 of Proposition 1. □

The hybrid continuation-smoothing method, called Algorithm 4.1, will replace the approximate Newton step (Step 5) with the following smoothing Newton step.

Step 5′ (Smoothing Newton Step)
If $x^k = y^k$ set $\epsilon_k = 1$. Otherwise, choose $\xi_k > 0$ and set $\epsilon_k = \delta(z^k)\xi_k$. Let $\Delta\hat{z}^k$ solve the equation

$$H_0(z^k) + H'(z^k, \epsilon_k e)\Delta\hat{z}^k = 0.$$

Set $\hat{z}^{k+1} = z^k + \Delta\hat{z}^k$.

A few remarks about the smoothing Newton step in the hybrid algorithm:

- Our definition of ϵ_k is simpler than that proposed in [11].
- To achieve superlinear convergence, the sequence $\{\xi_k\}$ has to approach 0. This can be achieved by setting $\xi_k = \|H_0(z^k)\|$. Indeed, for implementation purposes, we may choose

$$\epsilon_k = \min\{\mu_k \min_i\{a_i^k\}, \delta(z^k)\|H_0(z^k)\|\}.$$

- Unlike the approximate Newton step in Algorithm 3.1, the matrix inverted in Step 5′ is different from the matrix inverted in Step 2. Therefore, the removal of Assumption 3.2 is achieved at the cost of additional computations.

THEOREM 4.1. *Proposition* 3.1 *and Theorem* 3.1 *hold for Algorithm* 4.1. *If* $\xi_k \downarrow 0$, *then the conclusion of Theorem* 3.2 *holds for Algorithm* 4.1 *under Assumptions* 2.2 *and* 3.1.

Proof. Since the smoothing Newton step is well defined if F is a P_0 function, Algorithm 4.1 is well defined based on the proof of Proposition 3.1. In addition, with Step 5 replaced by Step 5′, the proof of Theorem 3.1 does not change.

Let $\{z^k\}$ be a sequence generated by Algorithm 4.1. By the global convergence result, every accumulation point z^* of $\{z^k\}$ is a solution of NCP(F). For local convergence, in view of the proof for Theorem 3.2, it suffices to show that

$$\|\hat{z}^{k+1} - z^*\| = o(\|z^k - z^*\|).$$

By Proposition 4.1 and the definition of ϵ_k, we have

$$\|H'(z^k, \epsilon_k e) - H^0(z^k)\| \le 2B_1\xi_k.$$

Since $\xi_k \downarrow 0$, $z^k \to z^*$, and $H^0(z) \in \partial H_0(z)$ for all $z \in R^{2n}$, Assumption 3.1 implies that there is a constant $C > 0$ such that

$$\|H'(z^k, \epsilon_k e)^{-1}\| \le C$$

holds for k sufficiently large. Therefore,

$$\begin{aligned}
\|\hat{z}^{k+1} - z^*\| &= \|z^k - z^* - H'(z^k, \epsilon_k e)^{-1}H_0(x^k)\| \\
&\le \|H'(z^k, \epsilon_k e)^{-1}\|\|H'(z^k, \epsilon_k e)(z^k - z^*) - H_0(z^k)\| \\
&\le C(\|H_0(z^k) - H_0(z^*) - H^0(z^k)(z^k - z^*)\| \\
&\qquad + \|H'(z^k, \epsilon_k e) - H^0(z^k)\|\|z^k - z^*\|) \\
&\le C(\|H_0(z^k) - H_0(z^*) - H^0(z^k)(z^k - z^*)\| + 2B_1\xi_k\|z^k - z^*\|) \\
&= o(\|z^k - z^*\|),
\end{aligned}$$

where the last equality follows from (4.1) and the assumption that $\xi_k \downarrow 0$. □

5. Final remarks. While this paper was under review, there had been active development in the smoothing methods for NCP related problems. Unlike the current paper, where the smoothing parameter is adjusted after each Newton iteration, the recent papers by Qi and Sun [31], Jiang [21], and Qi, Sun, and Zhou [32] treated the smoothing parameter as a variable in their expanded Newton equations. Tseng [38] improved the local convergence of the existing continuation methods by introducing an inexpensive active set strategy in computing the "fast" Newton direction. Burke and Xu [3] defined a new neighborhood for a noninterior smoothing path based on the CHKS smoothing function. Their continuation method for the monotone LCP based on the new neighborhood was shown to converge globally linearly and locally quadratically. More recently, Kanzow and Pieper [23] constructed a Jacobian smoothing method for general NCPs based on the concept of the Jacobian consistency property introduced in [11]. They reported good numerical results. We have also seen some new development in the regularization methods for NCPs, which are closely related to the smoothing methods. See [23, 28, 36]. The major advantage of the regularization methods is that they are able to solve the P_0 NCPs under weaker assumptions than those required by the existing smoothing methods.

REFERENCES

[1] S. C. Billups, S. P. Dirkse, and M. C. Ferris, *A comparison of algorithms for large-scale mixed complementarity problems*, Comput. Optim. Appl., 7 (1997), pp. 3–25.

[2] J. Burke and S. Xu, *The global linear convergence of a non-interior path-following algorithm for linear complementarity problem*, Math. Oper. Res., to appear.

[3] J. Burke and S. Xu, *A Non-Interior Predictor-Corrector Path Following Algorithm for the Monotone Linear Complementarity Problem*, Preprint, Department of Mathematics, University of Washington, Seattle, 1997.

[4] B. Chen and X. Chen, *A Global Linear and Local Quadratic Continuation Smoothing Method for Variational Inequalities with Box Constraints*, Department of Management and Systems, Washington State University, Pullman, 1997.

[5] B. Chen and P. T. Harker, *A non-interior-point continuation method for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1168–1190.

[6] B. Chen and P. T. Harker, *Smooth approximations to nonlinear complementarity problems*, SIAM J. Optim., 7 (1997), pp. 403–420.

[7] B. Chen and N. Xiu, *A global linear and local quadratic non-interior continuation method for nonlinear complementarity problems based on Chen–Mangasarian smoothing function*, SIAM J. Optim., 9 (1999), pp. 605–623.

[8] C. Chen and O. L. Mangasarian, *Smoothing methods for convex inequalities and linear complementarity problems*, Math. Programming, 71 (1995), pp. 51–69.

[9] C. Chen and O. L. Mangasarian, *A class of smoothing functions for nonlinear and mixed complementarity problems*, Comput. Optim. Appl., 5 (1996), pp. 97–138.

[10] X. Chen and L. Qi, *A parameterized Newton method and a Broyden-like method for solving nonsmooth equations*, Comput. Optim. Appl., 3 (1994), pp. 157–179.

[11] X. Chen, L. Qi, and D. Sun, *Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities*, Math. Comp., 67 (1998), pp. 519–540.

[12] X. Chen and Y. Ye, *On homotopy-smoothing methods for box-constrained variational inequalities*, SIAM J. Control Optim., 37 (1999), pp. 589–616.

[13] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.

[14] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Computer Science and Scientific Computing, Academic Press, San Diego, CA, 1990.

[15] S. P. Dirkse and M. C. Ferris, *The PATH solver: A non-monotone stabilization scheme for mixed complementarity problems*, Optim. Methods Softw., 5 (1995), pp. 123–156.

[16] F. Facchinei and C. Kanzow, *Beyond monotonicity in regularization methods for nonlinear complementarity problems*, SIAM J. Control Optim., to appear.

[17] R. M. Freund and S. Mizuno, *Interior point methods: Current status and future directions*, OPTIMA Newsletter, 51 (1996), pp. 1–9.

[18] S. A. Gabriel and J. J. Moré, *Smoothing of mixed complementarity problems*, in Complementarity and Variational Problems: State of the Art, M. C. Ferris and J. S. Pang, eds., SIAM, Philadelphia, 1996, pp. 105–116.

[19] P. T. Harker and J. S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problem: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 339–357.

[20] K. Hotta and A. Yoshise, *Global Convergence of a Class of Non-Interior-Point Algorithms Using Chen-Harker-Kanzow Functions for Nonlinear Complementarity Problems*, Discussion Paper Series 708, Institute of Policy and Planning Sciences, University of Tsukuba, Tsukuba, Japan, December, 1996.

[21] H. Jiang, *Smoothed Fischer-Burmeister Equation Methods for the Complementarity Problem*, Report, Department of Mathematics, University of Melbourne, Parkville, Australia, 1997.

[22] C. Kanzow, *Some noninterior continuation methods for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 851–868.

[23] C. Kanzow and H. Pieper, *Jacobian smoothing methods for nonlinear complementarity problems*, SIAM J. Optim., 9 (1999), pp. 342–373.

[24] M. Kojima, N. Megiddo, and S. Mizuno, *A general framework of continuation methods for complementarity problems*, Math. Oper. Res., 18 (1993), pp. 945–963.

[25] M. Kojima, N. Megiddo, and T. Noma, *Homotopy continuation methods for nonlinear complementarity problems*, Math. Oper. Res., 16 (1991), pp. 754–774.

[26] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, San Diego, 1970.

[27] J. S. Pang, *Complementarity problems*, in Handbook of Global Optimization, R. Horst and P. Pardalos, eds., Kluwer, Boston 1995, pp. 271–338.

[28] H. D. Qi, *A Regularized Smoothing Newton Method for Box Constrained Variational Inequality Problems with P_0-Functions*, Report, Chinese Academy of Sciences, Institute of Computational Mathematics and Scientific/Engineering Computing, Beijing, China, 1997.

[29] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.

[30] L. Qi and X. Chen, *A globally convergent successive approximation method for severely nonsmooth equations*, SIAM J. Control Optim., 33 (1995), pp. 402–418.

[31] L. Qi and D. Sun, *Globally Linearly, and Globally and Locally Superlinearly Convergent Versions of the Hotta-Yoshise Non-Interior Point Algorithm for Nonlinear Complementarity Problems*, Applied Mathematics Report, School of Mathematics, University of New South Wales, Sydney, Australia, 1997.

[32] L. Qi, D. Sun, and G. Zhou, *A New Look at Smoothing Newton Methods for Nonlinear Complementarity Problems and Box Constrained Variational Inequalities*, Applied Mathematics Report, AMR 97/13, School of Mathematics, University of New South Wales, Sydney, Australia, 1997.

[33] D. Ralph, *Global convergence of damped Newton's method for nonsmooth equations, via the path search*, Math. Oper. Res., 19 (1994), pp. 352–389.

[34] H. Sellami and S. M. Robinson, *Implementation of a continuation method for normal maps*, Math. Programming, 76 (1997), pp. 563–578.

[35] S. Smale, *Algorithms for solving equations*, in Proceedings of the International Congress of Mathematicians, Berkeley, CA, 1986, AMS, Providence, RI, 1987, pp. 172–195.

[36] D. Sun, *A regularization Newton method for solving nonlinear complementarity problems*, Appl. Math. Optim., to appear.

[37] P. Tseng, *An infeasible path-following method for monotone complementarity problems*, SIAM J. Optim., 7 (1997), pp. 386–402.

[38] P. Tseng, *Analysis of a non-interior continuation method based on Chen–Mangasarian smoothing functions for complementarity problems*, in Reformulation—Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, M. Fukushima and L. Qi, eds., Kluwer Academic Publishers, Norwell, MA, 1998, pp. 381–404.

[39] S. Wright and D. Ralph, *A superlinear infeasible-interior-point algorithm for monotone complementarity problems*, Math. Oper. Res., 21 (1996), pp. 815–838.

[40] S. Xu, *The Global Linear Convergence of an Infeasible Non-Interior Path-Following Algorithm for Complementarity Problems with Uniform P-Functions*, Preprint, Department of Mathematics, University of Washington, Seattle, 1996.

[41] S. Xu, *The Global Linear Convergence and Complexity of a Non-Interior Path-Following*

Algorithm for Monotone LCP Based on Chen-Harker-Kanzow-Smale Smooth Functions, Preprint, Department of Mathematics, University of Washington, Seattle, 1997.

[42] Y. Ye and K. Anstreicher, *On quadratic and $O(\sqrt{N}L)$ convergence of a predictor-corrector algorithm for LCP*, Math. Programming, 62 (1993), pp. 537–552.

[43] I. Zang, *A smoothing-out technique for min–max optimization*, Math. Programming, 19 (1980), pp. 61–71.

SIAM J. Optim.
Vol. 9, No. 3, pp. 646–667

A CLASS OF INDEFINITE DOGLEG PATH METHODS FOR UNCONSTRAINED MINIMIZATION[*]

JIANZHONG ZHANG[†] AND CHENGXIAN XU[‡]

Abstract. In this paper we propose a convenient curvilinear search method to solve trust region problems arising from unconstrained optimization problems. The curvilinear paths we set forth are dogleg paths, generated mainly by employing Bunch–Parlett factorization for general symmetric matrices that may be indefinite. This method is easy to implement and globally convergent. It is proved that the method satisfies the first- and second-order stationary point convergence properties and that the convergence rate is quadratic under commonly used conditions on functions. Numerical experiments are conducted to compare this method with some existing methods.

Key words. trust region method, curvilinear search, factorization of indefinite matrices, negative curvature, global convergence, superlinear and quadratic convergence rates

AMS subject classifications. 90C30, 65K05

PII. S105262349627523X

1. Introduction.

Trust region algorithms for solving the unconstrained minimization problem

$$\min \ f(x) \tag{1.1}$$

have strong convergence properties (see [12]). These algorithms are based on the following idea. Suppose $f(x)$ is twice continuously differentiable. For a given iterate $x^{(k)} \in R^n$, which is an estimate of a local solution x^* of problem (1.1), the solution, $\delta^{(k)}$ say, of the quadratic subproblem

$$\begin{aligned} \min q_k(\delta) &\overset{\text{def}}{=} f_k + {g^{(k)}}^T \delta + \frac{1}{2}\delta^T B_k \delta \\ \text{s.t. } &||\delta|| \le \Delta_k \end{aligned} \tag{1.2}$$

serves as a correction to $x^{(k)}$; i.e., $x^{(k)} + \delta^{(k)}$ is considered as a successor to $x^{(k)}$, where $f_k = f(x^{(k)})$, $g^{(k)}$ is the gradient $\nabla f(x^{(k)})$ of $f(x)$ at $x^{(k)}$, $\delta = x - x^{(k)}$, B_k is the Hessian matrix $\nabla^2 f(x^{(k)})$ of $f(x)$ at $x^{(k)}$ or its approximation, and Δ_k is a parameter called the trust region radius. The quadratic function $q_k(\delta)$ is a local approximation of the function $f(x)$ at point $x^{(k)}$ and the parameter Δ_k is adjusted at each iteration so that a reasonable agreement between $q_k(\delta)$ and $f(x)$ is maintained in a proper trust region $\{x| \ ||x - x^{(k)}|| \le \Delta_k\}$.

In implementing trust region algorithms, the basic issue is how to solve problem (1.2) efficiently. It is known that the solution $\delta^{(k)}$ of problem (1.2) generally satisfies the system

$$(B_k + \mu I)\delta(\mu) = -g^{(k)} \tag{1.3}$$

[*] Received by the editors August 26, 1996; accepted for publication (in revised form) April 17, 1998; published electronically April 19, 1999. This work was supported by the Hong Kong University Grants Committee (Competitive Earmarked Research Grant CITYU-9040159).

http://www.siam.org/journals/siopt/9-3/27523.html

[†] Department of Mathematics, City University of Hong Kong, Hong Kong (mazhang@cityu.edu.hk).

[‡] Department of Scientific Computing and Software, Xian Jiaotong University, Xian, People's Republic of China (mxxu@xjtu.edu.cn).

and $\|\delta^{(k)}\| = \Delta_k$, where $\mu \geq 0$ such that $B_k + \mu I$ is at least positive semidefinite. If B_k is positive definite and $\|B_k^{-1} g^{(k)}\| \leq \Delta_k$, then the solution is

$$\delta^{(k)} = -B_k^{-1} g^{(k)}. \tag{1.4}$$

In the hard case [16] when B_k is indefinite and $\|(B_k - \mu_1 I)^+ g^{(k)}\| < \Delta_k$, a solution of (1.2) is given by

$$\delta^{(k)} = -(B_k - \mu_1 I)^+ g^{(k)} + td, \quad t > 0, \quad \text{such that} \quad \|\delta^{(k)}\| = \Delta_k, \tag{1.5}$$

where $B_k d = \mu_1 d$, μ_1 is the smallest eigenvalue of B_k, and $(\cdot)^+$ denotes the generalized inverse of a matrix. Notice that $\delta^{(k)}$ satisfies (1.3) with $\mu = -\mu_1$ and that $B_k - \mu_1 I$ is positive semidefinite.

Newton's method is used to find an approximate solution of (1.2) on the full space. It can be seen that the solution of (1.2) is usually related to solving the nonlinear equation

$$\phi(\mu) - \Delta_k = 0, \qquad \phi(\mu) = \|\delta(\mu)\|, \quad \mu \ \geq \ \max\{-\mu_1, \ 0\}, \tag{1.6}$$

where $\delta(\mu)$ is the solution of (1.3). Based on a rational approximation to $\phi(\mu)$, Hebden [14] and Moré [16] employed Newton's iteration to (1.6) to find μ such that $\|\delta(\mu)\|$ is approximately equal to Δ_k. Moré and Sorensen [18] applied Newton's method to the equation $1/\phi(\mu) - 1/\Delta_k = 0$ to find $\mu > \max\{-\mu_1, \ 0\}$. Sorensen [23] applied Newton iteration to the system $(B_k + \mu I)\delta + g^{(k)} = 0, \ 1/\|\delta\| - 1/\Delta_k = 0$ to find μ and δ. In their implementation, these methods require the matrix $B_k + \mu I$ to be factorized by, for example, Cholesky factorization for every new value of μ.

An alternative to the full-dimensional solution of (1.2) is the curvilinear path method. Let Δ_k vary in subproblem (1.2); then the solution points form a curvilinear path with Δ_k as its parameter in the full-dimensional space, called the optimal path [4]. Then the subproblem can be interpreted as a problem of finding a point on the optimal curvilinear path, denoted by $\Gamma_{op}^{(k)}$, which minimizes $q_k(\delta)$ within the trust region, i.e.,

$$\delta^{(k)} = \arg\min \ \{ \ q_k(\delta) \mid \ \delta \in \Gamma_{op}^{(k)}, \ \ \|\delta\| \leq \Delta_k\}.$$

Besides the optimal path, there are other full-dimensional curvilinear paths, such as the gradient path [2] and the conjugate gradient path [4]. Forming these curvilinear paths requires the knowledge of the full eigensystem of the matrix B_k.

It is found that for the purpose of convergence, the full-dimensionality in the solution of (1.2) is not necessary (see [3] and [22]). Shultz, Schnabel, and Byrd [22] proposed a step computing function to form an algorithm that computes an approximate solution to (1.2) by performing a two-dimensional quadratic minimization

$$\min\{ \ q_k(\delta) \mid \delta \in \mathcal{S}, \ \ \|\delta\| \leq \Delta_k\}, \tag{1.7}$$

where $\mathcal{S}$ is a two-dimensional subspace spanned by $-g^{(k)}$ and $-B_k^{-1} g^{(k)}$ if B_k is positive definite and by two directions from $-g^{(k)}$, $-(B_k + \mu I)^{-1} g^{(k)}$, and a negative curvature direction d when B_k is indefinite. The algorithm maintains the strong convergence properties of the full-dimensional solution methods, and numerical results [7] show that this inexpensive algorithm performs almost as well as the expensive full-dimensional methods.

The most practical two-dimensional curvilinear paths are dogleg paths [20], [11], which are piecewise linear approximations to the optimal path in the subspace spanned by the steepest descent direction $-g^{(k)}$ and the Newton direction $-B_k^{-1}g^{(k)}$. Powell suggested in [20] a dogleg path which is called in this paper a single dogleg path, as it turns its direction once; Dennis and Mei, in a later paper [11], proposed a double dogleg path that makes two turns. These dogleg path methods work well when B_k is positive definite, but they are unable to deal with the indefinite case.

In this paper we describe a class of trust region indefinite dogleg path algorithms for unconstrained minimization. Dogleg paths are formulated over two-dimensional subspaces spanned by $-g^{(k)}$ and $-B_k^{-1}g^{(k)}$ if B_k is positive definite and by $-g^{(k)}$, $-(B_k + \mu I)^{-1}g^{(k)}$, and directions d of negative curvature when B_k is indefinite. Directions of negative curvature of B_k are obtained from the stable Bunch–Parlett [5] factorization of a symmetric matrix. This class of dogleg path algorithms is easy to implement and maintains the strong convergence properties of full-dimensional solution methods. That is, these algorithms are globally convergent and satisfy the first- and second-order stationary point convergence properties. The paper is organized as follows. In section 2 we describe the Bunch–Parlett factorization of the symmetric matrix and the properties of the factorization. It is also shown how to construct negative curvature directions of indefinite B_k from the factorization. In section 3 we present the indefinite dogleg path algorithms and describe formulations of indefinite dogleg paths. In section 4 we analyze the properties of the formulated indefinite dogleg paths. It is shown that these dogleg paths satisfy the required properties for curvilinear paths proposed in [22]. In section 5 it is proved that the proposed indefinite dogleg path algorithms are globally convergent and satisfy the first- and second-order stationary point convergence properties and that the quadratic convergence rate is preserved by these algorithms under reasonable conditions. Numerical results and comparison with the method in [22] are presented in section 6 and show that these indefinite dogleg path algorithms work as well as the two-dimensional minimization algorithm.

In the remainder of the paper we use the following notation:

- $||.||$ is the Euclidean norm.
- μ_1 and μ_n are, respectively, the smallest and largest eigenvalues of the matrix B_k.
- $[x^{(k)}, \ldots, y]$ or $[x^{(k)}, \ldots, y, w)$ denotes a dogleg path which starts from $x^{(k)}$ and changes direction at each point listed. The former is a finite dogleg path where y is the end point of the path, while the latter is an infinite dogleg path where the last piece of the path is a ray starting at point y along the direction w.

2. Bunch–Parlett factorization and directions of negative curvature. A real symmetric matrix B_k usually possesses a unique triangular factorization $B_k = LDL^T$, where L is a unit lower triangular matrix and D a diagonal matrix. When B_k is positive definite this factorization is stable. However, for an indefinite matrix, this factorization may not exist, and even if it exists, it may be unstable. A stable factorization method for real symmetric indefinite matrices was suggested by Bunch and Parlett [5] and Bunch, Kaufman, and Parlett [6]. The method factorizes matrix B_k into the form

$$PB_kP^T = LDL^T,$$

where P is a permutation matrix, L a unit lower triangular matrix, and D a block diagonal matrix with 1×1 and 2×2 diagonal blocks. If B_k is positive definite, D is diagonal. For convenience, without loss of generality, it is assumed in the following that $P = I$, i.e.,

$$B_k = LDL^T. \tag{2.1}$$

This factorization has the following properties (see [3]):

(i) D and B_k have the same inertia, that is, they have the same numbers of positive, zero, and negative eigenvalues.

(ii) The elements of the matrices L and L^{-1} are bounded by a fixed positive constant which is independent of B_k; i.e., there exist positive constants $c_1,\ c_2,\ c_3,$ and c_4 such that

$$c_1 \le \|L\| \le c_2, \qquad c_3 \le \|L^{-1}\| \le c_4. \tag{2.2}$$

Property (i) shows that the positive definiteness of the matrix B_k can be checked from that of the matrix D, whose eigenvalues are easy to calculate. Let $d_1 \le d_2 \le \cdots \le d_n$ be eigenvalues of the matrix D and $u^1,\ u^2, \dots,\ u^n$ be corresponding orthonormal eigenvectors. We partition the index set $\mathcal{N} = \{1, 2, \dots, n\}$ into $\mathcal{N}^+,\ \mathcal{N}^o$, and $\mathcal{N}^-$ corresponding to $d_i > 0,\ d_i = 0$, and $d_i < 0$. Clearly, the direction

$$d \overset{def}{=} -\mathrm{sgn}({g^{(k)}}^T L^{-T} v) L^{-T} v \tag{2.3}$$

with

$$v \in \mathcal{S} = \left\{ v \mid v = \sum_{i \in \mathcal{N}^- \cup \mathcal{N}^o} \ell_i u^i \ \ \forall\ \ell_i \in R \right\}$$

satisfies

$$d^T B_k d = v^T D v = \sum_{i \in \mathcal{N}^- \cup \mathcal{N}^o} d_i \ell_i^2 \le 0.$$

This shows that d is a direction of negative curvature of B_k. In practice, we are interested in the directions d with v in the set

$$\mathcal{C} = \left\{ v \mid v = \sum_{i \in \bar{\mathcal{N}}} r_i u^i, \ \ \bar{\mathcal{N}} \subset \mathcal{N}^- \cup \mathcal{N}^0 \right\} \subset \mathcal{S},$$

where $\bar{\mathcal{N}}$ is a selected index set, $r_i = {r^{(k)}}^T u^i,\ \ r^{(k)} \overset{\mathrm{def}}{=} L^T g^{(k)} = \sum_{i=1}^n r_i u^i$. For such directions d we have

$$ {g^{(k)}}^T B_k d = -\mathrm{sgn}\left({g^{(k)}}^T L^{-T} v\right) {g^{(k)}}^T L D v = -\mathrm{sgn}({g^{(k)}}^T L^{-T} v) d^T B_k d. \tag{2.4}$$

A particular choice for the vector $v \in \mathcal{C}$ is

$$v = r_1 u^1, \tag{2.5}$$

i.e., $\bar{\mathcal{N}} = \{1\}$. For such a vector v, the direction d of (2.3) has a desirable property given in Lemma 2.1 below, but we first show another property of Bunch–Parlett factorization.

(iii) Suppose B_k is not positive definite and let μ_1 and d_1 be the most negative eigenvalues of B_k and D, respectively. Then the following relations hold:

$$d_1\|L\|^2 \le \mu_1 \le d_1/\|L^{-1}\|^2. \tag{2.6}$$

This property can be proved as follows. Let y be the unit eigenvector of B_k corresponding to the eigenvalue μ_1 and set $z = L^T y$. Then

$$\|z\| \le \|L\| \cdot \|y\| = \|L\|$$

and thus

$$\mu_1 = y^T B_k y = y^T LDL^T y = z^T Dz \ge d_1\|z\|^2 \ge d_1\|L\|^2,$$

which gives the first inequality of (2.6). Let d be given by (2.3) with v of (2.5); then

$$\mu_1\|d\|^2 \le d^T B_k d = d_1 r_1^2. \tag{2.7}$$

Since

$$r_1^2 = \|v\|^2 = \|L^T d\|^2 \ge \frac{\|d\|^2}{\|L^{-1}\|^2}, \tag{2.8}$$

we obtain the second inequality of (2.6) by combining (2.7) and (2.8).

Lemma 2.1. *Suppose $\mu_1 < 0$. If we set $v = r_1 u^1$ and define d by* (2.3), *then*

$$d^T B_k d \le \frac{\mu_1}{c_2^2 c_4^2}\|d\|^2. \tag{2.9}$$

Proof. From (2.7), (2.8), and the first part of (2.6) we obtain

$$d^T B_k d = d_1 r_1^2 \le d_1 \frac{\|d\|^2}{\|L^{-1}\|^2} \le \frac{\mu_1}{\|L\|^2 \cdot \|L^{-1}\|^2}\|d\|^2.$$

Using property (ii) we have (2.9) immediately. □

We will see in section 5 that d of (2.3) with $v = r_1 u^1$ is an appropriate choice for a direction of negative curvature.

3. Algorithms.

In this section we describe the proposed trust region indefinite dogleg path algorithm. In each iteration, we shall solve a quadratic minimization subproblem

$$\min \left\{ q_k(\delta) = f_k + {g^{(k)}}^T \delta + \frac{1}{2}\delta^T B_k \delta \mid \delta \in \Gamma^{(k)}, \quad ||\delta|| \le \triangle_k \right\}, \tag{3.1}$$

where $\Gamma^{(k)}$ is a dogleg path. Let $\delta^{(k)}$ be the solution of the subproblem (3.1). Then either $x^{(k)} + \delta^{(k)}$ is accepted as a new iteration point or the trust region radius is reduced according to a comparison between the actual reduction of the objective function

$$\text{ared}(\delta^{(k)}) = f_k - f(x^{(k)} + \delta^{(k)}) \tag{3.2}$$

and the reduction predicted by the quadratic model

$$\text{pred}(\delta^{(k)}) = -{g^{(k)}}^T \delta^{(k)} - \frac{1}{2}{\delta^{(k)}}^T B_k \delta^{(k)}. \tag{3.3}$$

That is, if the reduction in the objective function is satisfactory, then we finish the current iteration by taking

$$x^{(k+1)} = x^{(k)} + \delta^{(k)}$$

and adjusting the trust region radius; otherwise the iteration is repeated at point $x^{(k)}$ with a reduced trust region radius. Now we are ready to state the algorithm.

Algorithm TRIDPM

Step 0. Choose parameters $0 < \eta_1 < \eta_2 < 1$, $0 < \gamma_1 < 1 < \gamma_2$, $\triangle_{max} > 0$; give a starting point $x^{(0)} \in R^n$ and an initial trust region radius $\triangle_0 < \triangle_{max}$. Set $k = 0$.

Step 1. Evaluate $f_k = f(x^{(k)})$, $g^{(k)} = \bigtriangledown f(x^{(k)})$.

Step 2. Termination test. If the iteration is not terminated, form a symmetric matrix B_k.

Step 3. Form a dogleg path $\Gamma^{(k)}$.

Step 4. Determine

$$\delta^{(k)} = \arg\min\{\ q_k(\delta)\ |\ \ \delta \in \Gamma^{(k)},\ ||\delta|| \le \triangle_k\}.$$

Step 5. Calculate ared$(\delta^{(k)})$, pred$(\delta^{(k)})$ and $\theta_k =$ared$(\delta^{(k)})/$pred$(\delta^{(k)})$.

Step 6. If $\theta_k < \eta_1$, then $\triangle_k = \gamma_1 \triangle_k$ and go to Step 4.

Step 7. $x^{(k+1)} = x^{(k)} + \delta^{(k)}$ and

$$\triangle_{k+1} = \begin{cases} \min\{\gamma_2 \triangle_k,\ \triangle_{max}\} & \text{if} \ \ \theta_k \ge \eta_2 \ \ \text{and} \ \ ||\delta^{(k)}|| = \triangle_k, \\ \\ \triangle_k & \text{otherwise.} \end{cases}$$

Step 8. Set $k \leftarrow k + 1$ and then go to Step 1.

The dogleg path $\Gamma^{(k)}$ in Step 3 can be formulated in the following ways.

1. If $B^{(k)}$ is positive definite, $\Gamma^{(k)}$ is Powell's single dogleg path

$$\Gamma_{Ps}^{(k)} = [x^{(k)},\ x_{cp}^{(k)},\ x_{np}^{(k)}]$$

or Dennis and Mei's double dogleg path

$$\Gamma_{Md}^{(k)} = [x^{(k)},\ x_{cp}^{(k)},\ \bar{x}_{np}^{(k)},\ x_{np}^{(k)}],$$

where the point

$$x_{cp}^{(k)} \overset{def}{=} x^{(k)} + \delta_{cp}^{(k)} \overset{def}{=} x^{(k)} - \beta_k g^{(k)}, \qquad \beta_k = \frac{{g^{(k)}}^T g^{(k)}}{{g^{(k)}}^T B_k g^{(k)}},$$

is called the Cauchy point, in which $\delta_{cp}^{(k)}$ is the minimizer of $q_k(\delta)$ in the steepest descent direction; the point

$$x_{np}^{(k)} \overset{def}{=} x^{(k)} + \delta_{np}^{(k)} \overset{def}{=} x^{(k)} - B_k^{-1} g^{(k)}$$

is called the Newton point, where $\delta_{np}^{(k)}$ is the global minimizer of $q_k(\delta)$ in the whole space R^n; and $\bar{x}_{np}^{(k)}$ is a point in the Newton direction:

$$\bar{x}_{np}^{(k)} = x^{(k)} - \gamma_k \delta_{np}^{(k)}$$

with γ_k satisfying the condition

$$\frac{({g^{(k)}}^T g^{(k)})^2}{{g^{(k)}}^T B_k g^{(k)} {g^{(k)}}^T B_k^{-1} g^{(k)}} \leq \gamma_k \leq 1.$$

2. If B_k is not positive definite, $\Gamma^{(k)}$ is an infinite dogleg path, and we give three choices for the path. Suppose the matrix B_k has been factorized into the form (2.1) and a direction d of negative curvature has been defined by (2.3) with $v \in \mathcal{C}$.

(1) If

$$\xi \|g^{(k)}\| \cdot \|L^{-T} v\| \geq {g^{(k)}}^T L^{-T} v > 0 \tag{3.4}$$

and

$$\frac{{g^{(k)}}^T B_k g^{(k)}}{|d^T B_k d|} \geq \max \left\{ \rho \frac{\|g^{(k)}\|^2}{\|d\|^2},\ \frac{\|g^{(k)}\|}{\|d\|} \left(\frac{\|g^{(k)}\|}{\|d\|} - 2 \right) \right\}, \tag{3.5}$$

in which the two parameters $\xi \in (1/2, 1)$ and $\rho \in (0, 1)$, then $\Gamma^{(k)}$ is the path

$$\Gamma_{Id1}^{(k)} = [x^{(k)},\ x_{\eta p}^{(k)},\ d),$$

where

$$x_{\eta p}^{(k)} \overset{def}{=} x^{(k)} + \delta_{\eta p}^{(k)}, \qquad \delta_{\eta p}^{(k)} = \eta_k \delta_{cp}^{(k)},$$

$$\eta_k = \frac{1 + \frac{{g^{(k)}}^T d}{\|g^{(k)}\| \|d\|}}{1 + \frac{\|g^{(k)}\|}{\|d\|} \frac{{g^{(k)}}^T B_k d}{{g^{(k)}}^T B_k g^{(k)}}}.$$

(2) If one of (3.4) and (3.5) does not hold, choose $\mu > 0$ such that

$$\text{the smallest eigenvalue of } B_k + \mu I \ \geq \omega', \tag{3.6}$$

where $\omega' > 0$ is a given constant, and such that for all k, μ are uniformly bounded if all B_k are. That is, if $\|B_k\| \leq M_2$ for a constant M_2, then there exists $M_3 > 0$ such that

$$\|B_k + \mu I\| \leq M_3 \quad \forall k. \tag{3.7}$$

An easy way to produce a μ satisfying (3.6) and (3.7) will be given in section 6. Let

$$\delta_B^{(k)} = -(B_k + \mu I)^{-1} g^{(k)}. \tag{3.8}$$

Define

$$\delta_{\mu p}^{(k)} = -\beta_\mu g^{(k)}, \quad \beta_\mu = \frac{{g^{(k)}}^T g^{(k)}}{{g^{(k)}}^T (B_k + \mu I) g^{(k)}},$$

$$\delta_{\mu B}^{(k)} = \frac{\|\delta_{\mu p}^{(k)}\|}{\|\delta_B^{(k)}\|} \delta_B^{(k)},$$

and make the direction d of negative curvature satisfy

$$d^T\delta_B^{(k)} \geq 0. \tag{3.9}$$

If

$$q_k(\delta_{\mu p}^{(k)}) \leq q_k(\delta_{\mu B}^{(k)}), \tag{3.10}$$

then choose the path

$$\Gamma_{Id2}^{(k)} = [x^{(k)},\ x_{\mu p}^{(k)},\ x_B^{(k)},\ d)$$

as $\Gamma^{(k)}$; otherwise $\Gamma^{(k)}$ is the path

$$\Gamma_{Id3}^{(k)} = [x^{(k)},\ x_B^{(k)},\ d),$$

where

$$x_{\mu p}^{(k)} = x^{(k)} + \delta_{\mu p}^{(k)}, \quad x_B^{(k)} = x^{(k)} + \delta_B^{(k)}.$$

The following lemma exposes our motivation of selecting the path $\Gamma_{Id1}^{(k)}$. These results shall be used in the later convergence analysis.

LEMMA 3.1. *The η_k defined above has the following properties:*

(i)

$$\frac{1-\xi}{1+c_2c_4/\rho} \leq \eta_k < 1. \tag{3.11}$$

(ii) *η_k is the solution to the equation*

$$-\nabla q_k(\eta\delta_{cp}^{(k)})^T\frac{g^{(k)}}{\|g^{(k)}\|} = \nabla q_k(\eta\delta_{cp}^{(k)})^T\frac{d}{\|d\|}.$$

And for $\eta_k < \eta \leq 1$,

$$-\nabla q_k(\eta\delta_{cp}^{(k)})^T\frac{g^{(k)}}{\|g^{(k)}\|} > \nabla q_k(\eta\delta_{cp}^{(k)})^T\frac{d}{\|d\|}.$$

(iii)

$$q_k(\eta\delta_{cp}^{(k)}) < q_k(\eta\delta_d^{(k)}) \quad \text{for} \quad \eta \in (0, \eta_k), \tag{3.12}$$

where $\delta_d^{(k)} = \|\delta_{cp}^{(k)}\|d/\|d\|$.

Proof. For simplicity, we suppress the superscript and subscript k in all proofs of the paper.

(i) Condition (3.4) means that

$$-\xi\|g\|\|d\| \leq g^Td < 0.$$

As in this case, (2.4) indicates $g^TBd = -d^TBd \geq 0$, and it is obvious that $\eta_k < 1$.

We know that $\|d\| = \|L^{-1}v\| \leq \|L^{-1}\|\|v\|$. Since

$$L^Tg = \sum_{i=1}^n r_iu^i,$$

but by the definition of the set $\mathcal{C}$, the vector v chosen in the algorithm is only a partial sum of the right side, we know that $\|v\| \le \|L^T g\|$, resulting in

$$\|d\| \le \|L\| \cdot \|L^{-1}\| \cdot \|g\| \le c_2 c_4 \|g\|.$$

Therefore, by the condition (3.5),

$$1 + \frac{\|g\|}{\|d\|} \frac{g^T Bd}{g^T Bg} \le 1 + \frac{1}{\rho} \frac{\|d\|}{\|g\|} \le 1 + c_2 c_4 / \rho,$$

which leads to the conclusion $\eta_k \ge \frac{1-\xi}{1+c_2 c_4/\rho}$.

(ii) By the definition of function $q(\delta)$, it is easy to know that the left side of the equation is

$$-\nabla q(\eta \delta_{cp})^T \frac{g}{\|g\|} = (\eta - 1)\|g\|,$$

while the right side is

$$\nabla q(\eta \delta_{cp})^T \frac{d}{\|d\|} = \frac{1}{\|d\|} \left(g^T d - \eta \|g\|^2 \frac{g^T Bd}{g^T Bg} \right).$$

Thus, it is straightforward to obtain the two conclusions.

(iii) We know that

$$q(\eta \delta_{cp}) = f - \eta \frac{\|g\|^4}{g^T Bg} + \frac{1}{2} \eta^2 \frac{\|g\|^4}{g^T Bg}$$

is a convex quadratic polynomial of η, whereas

$$q(\eta \delta_d) = f + \eta \frac{\|g\|^3}{g^T Bg} \frac{g^T d}{\|d\|} + \frac{1}{2} \eta^2 \frac{\|g\|^6}{(g^T Bg)^2} \frac{d^T Bd}{\|d\|^2}$$

is a concave quadratic polynomial. It is easy to obtain the unique nonzero solution to the equation $q(\eta \delta_{cp}) = q(\eta \delta_d)$, which is

$$\eta_k' = 2 \frac{1 + \frac{g^T d}{\|g\| \|d\|}}{1 + \frac{\|g\|^2}{\|d\|^2} \frac{|d^T Bd|}{g^T Bg}}.$$

Therefore, when $\eta \in (0, \eta_k')$,

$$q(\eta \delta_d) > q(\eta \delta_{cp}).$$

By condition (3.5), we have

$$1 + \frac{\|g\|^2}{\|d\|^2} \frac{|d^T Bd|}{g^T Bg} < 2 \left(1 + \frac{\|g\|}{\|d\|} \frac{g^T Bd}{g^T Bg} \right),$$

which immediately leads to the conclusion $\eta_k' > \eta_k$, and hence the proof is completed. □

Notice that as $\eta_k < 1$, the path $\Gamma_{Id1}^{(k)}$ turns its direction at the point $x_{\eta p}^{(k)}$, which is closer to $x^{(k)}$ than the Cauchy point $x_{cp}^{(k)}$, and we can see from the conclusions in (ii) that more reduction in the value of quadratic function $q_k(\delta)$ can be achieved by turning the direction of the path at point $x_{\eta p}^{(k)}$. The two vectors $\delta_{\mu p}^{(k)}$ and $\delta_{\mu B}^{(k)}$ have equal length but the predicted reduction along the first direction is larger, which is proved in the next lemma.

LEMMA 3.2. *Under the condition* (3.10)*,*

$$\text{pred}(\eta\delta_{\mu p}^{(k)}) \geq \text{pred}(\eta\delta_{\mu B}^{(k)}) \quad \forall\, \eta \in [0,1]. \tag{3.13}$$

Proof. By the definition of pred(.), (3.13) is equivalent to the condition

$$q(\eta\delta_{\mu B}) \geq q(\eta\delta_{\mu p}), \qquad \eta \in [0,1].$$

Notice that the graphs of

$$q(\eta\delta_{\mu B}) = f + g^T\delta_{\mu B}\eta + \frac{1}{2}\delta_{\mu B}^T B\delta_{\mu B}\eta^2$$

and

$$q(\eta\delta_{\mu p}) = f + g^T\delta_{\mu p}\eta + \frac{1}{2}\delta_{\mu p}^T B\delta_{\mu p}\eta^2$$

are two parabolas intersecting at $\eta = 0$. Since

$$-g^T\delta_B = g^T(B+\mu I)^{-1}g < \|g\|\|\delta_B\|,$$

their slopes at $\eta = 0$ have the relation

$$g^T\delta_{\mu p} = -\|\delta_{\mu p}\|\|g\| < \frac{\|\delta_{\mu p}\|}{\|\delta_B\|}g^T\delta_B = g^T\delta_{\mu B} < 0,$$

which means that for sufficiently small $\eta > 0$, the desired result is true. On the other hand, condition (3.10) tells us that this inequality also holds at $\eta = 1$. As the two parabolas can have at most one intersecting point when $\eta > 0$, we know immediately that the desired inequality must be true for all $\eta \in [0,1]$. □

As a matter of fact, if only theoretical analysis is concerned, when one of (3.4) and (3.5) does not hold, we can use Γ_{Id3} only to ensure all convergence properties. However, by Lemma 3.2, under condition (3.10) it is very likely that along the first piece of Γ_{Id2} the value of function q_k will reduce more quickly than along the first piece of Γ_{Id3}. Thus, so that the algorithm will be more efficient, we use indefinite path Γ_{Id2} when condition (3.10) holds.

4. Properties of indefinite paths. We want the curvilinear paths formulated to satisfy the following two properties proposed in [4]; that is, when point x proceeds from $x^{(k)}$ along the path,

(R1) the distance to $x^{(k)}$ is monotonically increasing, and

(R2) the value of $q_k(\delta)$ is monotonically decreasing.

These two properties ensure that for any given radius Δ_k, a unique solution $\delta^{(k)}$ of problem (3.1) along the path exists and can be found easily. In this section, we will show that the indefinite paths $\Gamma_{Id1}^{(k)} - \Gamma_{Id3}^{(k)}$ indeed satisfy both (R1) and (R2).

LEMMA 4.1. *Indefinite paths* Γ_{Id1}, Γ_{Id2}*, and* Γ_{Id3} *satisfy both properties* (R1) *and* (R2)*.*

Proof. As the proofs for the three paths are similar, we give only the proof for the path Γ_{Id2}.

Since

$$\begin{aligned}\delta_{\mu p}^T(\delta_B-\delta_{\mu p}) &= -\beta_\mu g^T[-(B+\mu I)^{-1}g+\beta_\mu g]\\ &= \beta_\mu[g^T(B+\mu I)^{-1}g-\beta_\mu g^Tg]\\ &= \beta_\mu\frac{g^T(B+\mu I)^{-1}g\cdot g^T(B+\mu I)g-(g^Tg)^2}{g^T(B+\mu I)g}\geq 0,\end{aligned}$$

and $\delta_B^Td\geq 0$, it is clear that path Γ_{Id2} satisfies property (R1).

The function $q(\delta)$ is obviously decreasing along the first piece of Γ_{Id2}. In fact, if $g^TBg>0$, then δ_{cp} is the minimizer of $q(\delta)$ in the steepest descent direction $-g$, and since $\|\delta_{\mu p}\|\leq\|\delta_{cp}\|$, $q(\delta)$ is decreasing before reaching $x_{\mu p}$. On the other hand, if $g^TBg\leq 0$, then $-g$ is a direction of negative curvature of B and hence $q(\delta)$ is always decreasing along the direction $-g$. Now let $x(\lambda)=x+\delta(\lambda)$, where $\delta(\lambda)=\delta_{\mu p}+\lambda(\delta_B-\delta_{\mu p})$ and $0\leq\lambda\leq 1$, be a point on the second piece of path Γ_{Id2}. Then, using

$$g^TB(B+\mu I)^{-1}g=g^Tg-\mu g^T(B+\mu I)^{-1}g, \tag{4.1}$$

$$g^T(B+\mu I)^{-1}B(B+\mu I)^{-1}g=g^T(B+\mu I)^{-1}g-\mu\|(B+\mu I)^{-1}g\|^2, \tag{4.2}$$

$$g^TBg=g^T(B+\mu I)g-\mu g^Tg, \tag{4.3}$$

we obtain

$$\begin{aligned}&\nabla q(\delta(\lambda))^T(\delta_B-\delta_{\mu p})\\ &= [g+B(\delta_{\mu p}+\lambda(\delta_B-\delta_{\mu p}))]^T(\delta_B-\delta_{\mu p})\\ &= -g^T(B+\mu I)^{-1}g+\beta_\mu g^Tg+\beta_\mu g^TB(B+\mu I)^{-1}g-\beta_\mu^2g^TBg\\ &\quad +\lambda g^T(B+\mu I)^{-1}B(B+\mu I)^{-1}g+\lambda\beta_\mu^2g^TBg-2\lambda\beta_\mu g^TB(B+\mu I)^{-1}g\\ &= (\lambda-1)(1+\mu\beta_\mu)\frac{g^T(B+\mu I)g\cdot g^T(B+\mu I)^{-1}g-(g^Tg)^2}{g^T(B+\mu I)g}\\ &\quad +\lambda\mu(\delta_{\mu p}^T\delta_B-\|\delta_B\|^2)\\ &< 0\quad\forall\lambda\in(0,1),\end{aligned}$$

because $\|\delta_{\mu p}\|<\|\delta_B\|$. This shows that $q(\delta)$ is monotonically decreasing along the second piece of Γ_{Id2}. Finally, let $x(\lambda)=x+\delta(\lambda)$, where $\delta(\lambda)=\delta_B+\lambda d$ and $\lambda>0$, be a point on the third piece of the path Γ_{Id2}. Then, using the facts $d^T\delta_B\geq 0$ and $d^TBd\leq 0$, we have

$$\begin{aligned}\nabla q(\delta(\lambda))^Td &= [g+B(\delta_B+\lambda d)]^Td\\ &= g^Td+d^TB\delta_B+\lambda d^TBd\\ &= g^Td-d^TB(B+\mu I)^{-1}g+\lambda d^TBd\\ &= -\mu d^T\delta_B+\lambda d^TBd\\ &\leq 0.\end{aligned}$$

Hence, $q(\delta)$ is also monotonically decreasing along the third piece of Γ_{Id2}. This completes the proof. □

5. Convergence properties.

We now analyze the convergence properties of Algorithm TRIDPM. The following assumptions are often used in this section.

AS1. $f(x)$ is twice continuously differentiable and bounded below.

AS2. $\|\nabla^2 f(x)\| \leq M_1$ in the level set $\mathcal{L}(x^{(0)}) \overset{def}{=} \{\, x \mid f(x) \leq f(x^{(0)})\}$, where M_1 is a constant.

AS3. $\bigtriangledown^2 f(x)$ is Lipschitz continuous over $\mathcal{L}(x^{(0)})$.

These mild assumptions are commonly used in the convergence analysis of most optimization algorithms. Shultz, Schnabel, and Byrd [22] presented some general conditions on the approximate solution $\delta^{(k)}$ of problem (1.2) in discussing the convergence properties of trust region-type algorithms. These conditions can be stated as follows.

C1. There exist ω_1, $\sigma_1 > 0$ such that for all $\Delta_k > 0$,

$$\text{pred}(\delta^{(k)}) \geq \|g^{(k)}\| \min\left\{\omega_1\Delta_k, \sigma_1\frac{\|g^{(k)}\|}{\|B_k\|}\right\}.$$

If $\|B_k\| \leq M_2$, then this condition can be replaced by

$$\text{pred}(\delta^{(k)}) \geq \|g^{(k)}\| \min\left\{\omega_1\Delta_k, \sigma_1\frac{\|g^{(k)}\|}{M_2}\right\}.$$

C2. There exists $\omega_2 > 0$ such that for all $\Delta_k > 0$,

$$\text{pred}(\delta^{(k)}) \geq -\omega_2\mu_1\Delta_k^2,$$

where μ_1 is the smallest eigenvalue of B_k.

C3. If B_k is positive definite and $\|B_k^{-1}g^{(k)}\| \leq \Delta_k$, then

$$\delta^{(k)} = -B_k^{-1}g^{(k)}.$$

Let a trust region algorithm be applied to minimize a function $f(x)$ satisfying the assumptions AS1 and AS2. It is shown in [22] that

1. If the approximate solution $\delta^{(k)}$ generated by the trust region algorithm satisfies condition C1 and $\|B_k\| \leq M_2$ for all k, then $g^{(k)}$ converges to 0, where M_2 is a constant.
2. If $B_k = \nabla^2 f(x^{(k)})$ and $\delta^{(k)}$ satisfies conditions C1 and C2, then $\nabla^2 f(x)$ is positive semidefinite at the accumulation points of the sequence $\{x^{(k)}\}$.
3. If $B_k = \nabla^2 f(x^{(k)})$, $\delta^{(k)}$ satisfies conditions C1 and C3, assumption AS3 holds, and $\nabla^2 f(x)$ is positive definite at limit x^* of the sequence $\{x^{(k)}\}$, then $x^{(k)}$ converges to x^* at a quadratic rate.

It follows from our choice of dogleg paths in the case of positive definite matrix $B^{(k)}$ that condition C3 is obviously satisfied for Algorithm TRIDPM and the algorithm is quadratically convergent if $B_k = \nabla^2 f(x^{(k)})$ and $x^{(k)}$ converges to x^*, where $\nabla^2 f(x^*)$ is positive definite. Thus, in the following, we prove only that the solution $\delta^{(k)}$ obtained at Step 4 of Algorithm TRIDPM satisfies conditions C1 and C2.

Lemma 5.1. *If $\|B_k\| \leq M_2$, then the solution δ at Step* 4 *of Algorithm* TRIDPM *satisfies condition* C1 *with*

$$\omega_1 = \frac{1}{2}\min\left\{1,\ \frac{\omega'}{M}\right\}, \qquad \sigma_1 = \frac{1-\xi}{2(1+c_2c_4/\rho)}, \tag{5.1}$$

where ω', ρ, and ξ are constants given in (3.6), (3.4), *and* (3.5), *and $M = \max\{M_2, M_3\}$ with M_3 given in* (3.7).

Proof. For δ on paths Γ_{Ps} and Γ_{Md} it is known that

$$\text{pred}(\delta) \geq \frac{1}{2}\|g\| \min\left\{\Delta,\ \frac{\|g\|}{\|B\|}\right\}. \tag{5.2}$$

For δ on path Γ_{Id1}, if $\|\delta_{\eta p}\| \geq \Delta$, then $\delta = -\Delta g/\|g\|$. As in this case, $\Delta \leq \|\delta_{cp}\| = \frac{\|g\|^3}{g^TBg}$, it is easy to obtain

$$\text{pred}(\delta) \geq \frac{1}{2}\Delta\|g\|. \tag{5.3}$$

If $\|\delta_{\eta p}\| < \Delta$, then $\delta = \delta_{\eta p} + \lambda d$ for some $\lambda > 0$ and it follows from property (R2) and (3.11) that

$$\text{pred}(\delta) \geq \text{pred}(\delta_{\eta p}) = \eta\left(1 - \frac{\eta}{2}\right)\frac{(g^Tg)^2}{g^TBg} \geq \frac{1}{2}\;\frac{1-\xi}{(1+c_2c_4/\rho)}\;\frac{\|g\|^2}{\|B\|}.$$

Thus, for δ on path Γ_{Id1}, we have

$$\text{pred}(\delta) \geq \frac{1}{2}\|g\|\,\min\left\{\Delta,\ \frac{1-\xi}{1+c_2c_4/\rho}\frac{\|g\|}{M}\right\}. \tag{5.4}$$

For δ on path Γ_{Id2}, if $\|\delta_{\mu p}\| \geq \Delta$, then $\delta = -\Delta g/\|g\|$ and (5.3) holds. If $\|\delta_{\mu p}\| < \Delta$, using the property (R2) and (4.3) and (3.7), we obtain

$$\begin{aligned}\text{pred}(\delta) \geq \text{pred}(\delta_{\mu p}) &= -g^T\delta_{\mu p} - \frac{1}{2}\delta_{\mu p}^TB\delta_{\mu p}\\ &= \frac{1}{2}\frac{(g^Tg)^2}{g^T(B+\mu I)g} + \frac{1}{2}\mu\frac{(g^Tg)^3}{(g^T(B+\mu I)g)^2}\\ &\geq \frac{1}{2}\frac{\|g\|^2}{\|B+\mu I\|} \geq \frac{1}{2}\frac{\|g\|^2}{M}.\end{aligned}$$

Thus, for δ on path Γ_{Id2}, we have

$$\text{pred}(\delta) \geq \frac{1}{2}\|g\|\,\min\left\{\Delta,\ \frac{\|g\|}{M}\right\}. \tag{5.5}$$

For δ on path Γ_{Id3}, if $\|\delta_B\| \geq \Delta$, then there exists a $\lambda \in (0,1]$ such that $\delta = \lambda\delta_B$ and $\|\lambda\delta_B\| = \Delta$. We then obtain, using (4.2),

$$\begin{aligned}\text{pred}(\delta) &= \lambda\left(1 - \frac{\lambda}{2}\right)g^T(B+\mu I)^{-1}g + \frac{1}{2}\mu\Delta^2\\ &\geq \lambda\left(1 - \frac{\lambda}{2}\right)\frac{\|g\|^2}{\|B+\mu I\|}.\end{aligned}$$

Using the fact

$$\|\delta_B\| \leq \|(B+\mu I)^{-1}\|\cdot\|g\| = \frac{\mathcal{K}(B+\mu I)}{\|B+\mu I\|}\|g\|, \tag{5.6}$$

it is immediate to obtain

$$\text{pred}(\delta) \geq \frac{1}{2}\|g\|\frac{\lambda\|\delta_B\|}{\mathcal{K}(B+\mu I)} = \frac{1}{2}\|g\|\frac{\Delta}{\mathcal{K}(B+\mu I)},$$

where $\mathcal{K}(\cdot)$ is the condition number of a matrix. It follows from (3.6) and (3.7) that

$$\mathcal{K}(B+\mu I) = \|B+\mu I\|\cdot\|(B+\mu I)^{-1}\| \leq \frac{M}{\omega'}. \tag{5.7}$$

Therefore,

$$\mathrm{pred}(\delta) \geq \frac{1}{2}\frac{\omega'}{M}\|g\|\Delta.$$

If $\|\delta_B\| \leq \Delta$, then $\delta = \delta_B + \lambda d$ for $\lambda > 0$. Using (R2), (4.2), and (3.7) we have

$$\begin{aligned}\mathrm{pred}(\delta) \geq \mathrm{pred}(\delta_B) &= -g^T\delta_B - \frac{1}{2}\delta_B^T B\delta_B \\ &= \frac{1}{2}g^T(B+\mu I)^{-1}g + \frac{1}{2}\mu\|\delta_B\|^2 \\ &\geq \frac{1}{2}\frac{\|g\|^2}{\|B+\mu I\|} \geq \frac{1}{2}\frac{\|g\|^2}{M}.\end{aligned}$$

Thus, for δ on path Γ_{Id3}, we have

$$\mathrm{pred}(\delta) \geq \frac{1}{2}\|g\| \min\left\{\frac{\omega'}{M}\Delta,\ \frac{\|g\|}{M}\right\}. \tag{5.8}$$

Combining (5.2), (5.4), (5.5), and (5.8) we obtain C1 with ω_1 and σ_1 given by (5.1). This completes the proof. □

Based on this lemma the following result can be immediately obtained from Theorem 2.2 of [22].

THEOREM 5.2. *Let $f(x)$ satisfy assumptions* AS1 *and* AS2 *and assume $\|B_k\| \leq M_2$ for all k. Then the sequence $\{g^{(k)}\}$ generated by Algorithm* TRIDPM *converges to* 0 *(first-order stationary point convergence). Moreover, if $f(x)$ also satisfies assumption* AS3, *$B_k = \nabla^2 f(x^{(k)})$ for all k, and $x^{(k)}$ converges to x^*, where $\nabla^2 f(x^*)$ is positive definite, then the convergence rate is quadratic.*

Now we prove the second-order stationary point convergence of the algorithm; i.e., if the sequence $\{x^{(k)}\}$ generated by the algorithm with $B_k = \nabla^2 f(x^{(k)})$ for all k converges to x^*, then $\nabla^2 f(x^*)$ is at least positive semidefinite.

LEMMA 5.3. *The solution δ at Step* 4 *of Algorithm* TRIDPM *satisfies condition C2 with $\omega_2 = c/2$ if*

$$d^T B d \leq c\mu_1\|d\|^2 \tag{5.9}$$

holds for the direction d of negative curvature, where $c \in (0,1)$ is a constant and μ_1 is the smallest eigenvalue of B.

Notice that according to Lemma 2.1, when the direction d is given by (2.3) with $v = r_1u^1$, the condition (5.9) holds.

Proof. Since condition C2 is clearly satisfied when the matrix B is positive definite, we consider only the case of indefinite B.

For δ on path Γ_{Id1}, if $\|\delta_{\eta p}\| \geq \Delta$, then $\delta = \lambda\delta_{cp},\ 0 < \lambda \leq \eta,\ \|\lambda\delta_{cp}\| = \Delta$, and it follows from (3.12) and (5.9) that

$$\begin{aligned}\mathrm{pred}(\delta) = \mathrm{pred}(\lambda\delta_{cp})\ &\geq\ \mathrm{pred}(\lambda\delta_d) \\ &= -\lambda\beta\frac{\|g\|}{\|d\|}g^T d - \frac{1}{2}\lambda^2\beta^2\frac{\|g\|^2}{\|d\|^2}d^T B d \\ &\geq -\frac{1}{2}\lambda^2\beta^2\|g\|^2\frac{d^T B d}{d^T d} \geq -\frac{1}{2}c\mu_1\Delta^2,\end{aligned}$$

where $\delta_d = \|\delta_{cp}\| d/\|d\|$. If $\|\delta_{\eta p}\| < \Delta$, then $\delta = \delta_{\eta p} + \lambda d$ for $\lambda > 0$, and $\|\delta\| = \Delta$, leading to

$$\begin{aligned}
\operatorname{pred}(\delta) &= -g^T(\delta_{\eta p} + \lambda d) - \frac{1}{2}(\delta_{\eta p} + \lambda d)^T B(\delta_{\eta p} + \lambda d) \\
&= \eta\left(1 - \frac{\eta}{2}\right)\frac{(g^Tg)^2}{g^TBg} - \lambda g^T d + \lambda\eta\beta g^T Bd - \frac{1}{2}\lambda^2 d^T Bd \\
&\geq \eta\left(1 - \frac{\eta}{2}\right)\frac{(g^Tg)^2}{g^TBg} - \lambda g^T d + \lambda\eta\beta g^T Bd - \frac{1}{2}c\mu_1\|\lambda d\|^2, \\
\|\lambda d\|^2 &= -\eta^2\beta^2 g^T g + 2\eta\lambda\beta g^T d + \Delta^2.
\end{aligned}$$

Thus, we have

$$\begin{aligned}
\operatorname{pred}(\delta) \geq{}& \eta\left(1 - \frac{1}{2}\eta + \frac{1}{2}c\mu_1\beta\eta\right)\frac{(g^Tg)^2}{g^TBg} - \lambda(1 + c\mu_1\beta\eta)g^T d \\
&+ \lambda\eta\beta g^T Bd - \frac{1}{2}c\mu_1\Delta^2.
\end{aligned}$$

From (3.11), (5.9), and (3.4) we obtain

$$\begin{aligned}
&1 + c\mu_1\beta\eta \geq 1 + c\mu_1\beta \geq 0, \\
&1 - \frac{1}{2}\eta + \frac{1}{2}c\mu_1\beta\eta \geq \frac{1}{2}(1 + c\mu_1\beta\eta) \geq 0.
\end{aligned}$$

Using conditions $g^T Bd \geq 0$ and $-g^T d > 0$ in the case of path Γ_{Id1}, we have

$$\operatorname{pred}(\delta) \geq -\frac{1}{2}c\mu_1\Delta^2.$$

Thus for δ on path Γ_{Id1}, we always have

$$\operatorname{pred}(\delta) \geq -\frac{1}{2}c\mu_1\Delta^2. \tag{5.10}$$

For δ on path Γ_{Id2}, if $\|\delta_{\mu p}\| \geq \Delta$, then $\delta = -\Delta g/\|g\|$. Denote $\hat{\delta} = \Delta\delta_B/\|\delta_B\|$. Then it follows from Lemma 3.2, $\|\delta_{\mu p}\| < \|\delta_B\|$, and the choice of μ that

$$\begin{aligned}
\operatorname{pred}(\delta) \geq \operatorname{pred}(\hat{\delta}) &= -g^T\hat{\delta} - \frac{1}{2}\hat{\delta}^T B\hat{\delta} \\
&= \frac{\Delta}{\|\delta_B\|}g^T(B + \mu I)^{-1}g - \frac{1}{2}\frac{\Delta^2}{\|\delta_B\|^2}[g^T(B + \mu I)^{-1}g - \mu\|\delta_B\|^2] \\
&= \frac{\Delta}{\|\delta_B\|}\left(1 - \frac{1}{2}\frac{\Delta}{\|\delta_B\|}\right)g^T(B + \mu I)^{-1}g + \frac{1}{2}\mu\Delta^2 \\
&\geq -\frac{1}{2}\mu_1\Delta^2. \qquad (5.11)
\end{aligned}$$

If $\|\delta_{\mu p}\| < \Delta \leq \|\delta_B\|$, then $\delta = (1 - \lambda)\delta_{\mu p} + \lambda\delta_B$ for a $\lambda \in (0, 1]$ and $\|\delta\| = \Delta$. Using (4.1)–(4.3) we obtain

$$\begin{aligned}
\operatorname{pred}(\delta) &= -g^T[(1 - \lambda)\delta_{\mu p} + \lambda\delta_B] - \frac{1}{2}[(1 - \lambda)\delta_{\mu p} + \lambda\delta_B]^T B[(1 - \lambda)\delta_{\mu p} + \lambda\delta_B] \\
&= (1 - \lambda)\beta_\mu g^T g + \lambda g^T(B + \mu I)^{-1}g - \frac{1}{2}(1 - \lambda)^2\beta_\mu^2 g^T Bg
\end{aligned}$$

$$
\begin{aligned}
&-\frac{1}{2}\lambda^2 g^T(B+\mu I)^{-1}B(B+\mu I)^{-1}g - \lambda(1-\lambda)\beta_\mu g^T B(B+\mu I)^{-1}g\\
&= (1-\lambda)\beta_\mu g^T g + \lambda g^T(B+\mu I)^{-1}g - \frac{1}{2}(1-\lambda)^2\beta_\mu g^T g\\
&\quad+\frac{1}{2}(1-\lambda)^2\beta_\mu^2\mu g^T g - \frac{1}{2}\lambda^2 g^T(B+\mu I)^{-1}g + \frac{1}{2}\lambda^2\mu\|\delta_B\|^2\\
&\quad-\lambda(1-\lambda)\beta_\mu g^T g + \mu\lambda(1-\lambda)\beta_\mu g^T(B+\mu I)^{-1}g\\
&= \left[1-\lambda-\frac{1}{2}(1-\lambda)^2-\lambda(1-\lambda)\right]\beta_\mu g^T g + \lambda\left(1-\frac{1}{2}\lambda\right)g^T(B+\mu I)^{-1}g\\
&\quad+\frac{1}{2}\mu[(1-\lambda)^2\|\delta_{\mu p}\|^2 + \lambda^2\|\delta_B\|^2 + 2\lambda(1-\lambda)\delta_{\mu p}^T\delta_B]\\
&= \frac{1}{2}(1-\lambda)^2\beta_\mu g^T g + \lambda\left(1-\frac{1}{2}\lambda\right)g^T(B+\mu I)^{-1}g + \frac{1}{2}\mu\|\delta\|^2\\
&\geq \frac{1}{2}\mu\Delta^2 \geq -\frac{1}{2}\mu_1\Delta^2.
\end{aligned}
$$

If $\|\delta_B\| < \Delta$, then $\delta = \delta_B + \lambda d$ for $\lambda > 0$, and $\|\delta\| = \Delta$. It has been proved in [22] that in this case,

$$\text{pred}(\delta) \geq -\frac{1}{2}c\mu_1\Delta^2. \tag{5.12}$$

Thus for δ on path Γ_{Id2} we have

$$\text{pred}(\delta) \geq -\frac{1}{2}c\mu_1\Delta^2. \tag{5.13}$$

Finally, for δ on path Γ_{Id3}, it comes directly from (5.11) for the case $\|\delta_B\| \geq \Delta$ (because in this case the solution $\delta = \hat{\delta}$) and from (5.12) for the case $\|\delta_B\| < \Delta$ that (5.13) holds. Therefore, the result of the lemma comes from (5.10) and (5.13). This completes the proof. □

Based on this lemma, we can immediately obtain, from Theorem 2.2 of [22], the following second-order stationary point convergence result.

THEOREM 5.4. *Suppose that $f(x)$ satisfies assumptions* AS1 *and* AS2 *and that Algorithm* TRIDPM *with $B_k = \nabla^2 f(x^{(k)})$ for all k is applied to $f(x)$. If the generated sequence $\{x^{(k)}\}$ converges to x^*, then $\nabla^2 f(x^*)$ is at least positive semidefinite.*

6. Numerical results. In this section we present numerical results for Algorithm TRIDPM. The matrix B_k is evaluated from exact Hessians of test functions and the Bunch–Parlett factorization [6] is employed to factorize B_k. If B_k is positive definite, Powell's single dogleg path is used; if B_k is not positive definite, a negative curvature direction d will be generated by formula (2.3) with v given in (2.5). As shown in Lemma 2.1, such a direction d satisfies condition (2.9). Then one of the three indefinite dogleg paths is selected, depending on the conditions given in section 3.

Notice that in order to obtain a $\mu = \mu^{(k)}$ which satisfies (3.6) and (3.7), we do not need to calculate the smallest eigenvalue μ_1. Suppose $B^{(k)} = \{b_{ij}\}$ $(i, j = 1, \ldots, n)$, and for $i = 1, \ldots, n$, calculate

$$c_i = \sum_{j\neq i}|b_{ij}| + 0.05 - b_{ii}.$$

Then we just take $\mu^{(k)}$ as

$$\mu^{(k)} = \max\{0, c_1, \ldots, c_n\}.$$

The rationality of this choice is that such $\mu^{(k)}$ must satisfy

$$\mu^{(k)} + b_{ii} \geq \sum_{j \neq i} |b_{ij}| + 0.05, \quad i = 1, \ldots, n,$$

so that $B^{(k)} + \mu^{(k)} I$ is diagonally dominant. Since this matrix is real symmetric, all its eigenvalues are real numbers. Using the Gerschgorin theorem, we know that the smallest eigenvalue of $B^{(k)} + \mu^{(k)} I$ is no less than 0.05; i.e., (3.6) holds with $\omega' = 0.05$. Also, it is easy to see that $\{\mu^{(k)}\}$ must be uniformly bounded if $\{B^{(k)}\}$ is.

The experiments are carried out in Fortran routines with single precision. The parameter values set in the program are $\eta_1 = 0.001$, $\eta_2 = 0.75$, $\gamma_1 = 0.1$, $\gamma_2 = 2$, $\Delta_0 = \max\{2, \|x^{(0)}\|/10\}$, and $\Delta_{max} = 10\Delta_0$. The values ξ=0.8 and $\rho = 0.1$ are used in conditions (3.4) and (3.5). The convergence criterion

$$\|g^{(k)}\| \leq 10^{-5} \quad \text{or} \quad f(x^{(k)}) - f(x^{(k+1)}) \leq 10^{-6} \max\{1.0, |f(x^{(k)})|\}$$

is used for the termination test; that is, when one of the two conditions is satisfied, computation stops. We also set a maximum iteration number, 500, to terminate calculation when this number is reached, but this type of termination did not occur in our experiments.

The first experiment is carried out on 14 standard unconstrained optimization test problems from Moré, Garbow, and Hillstrom [17]. The top half of Table 6.4 lists the problem numbers and names and the numbers in [17] for these functions. In [7], Byrd, Schnabel, and Shultz employed these problems to test their algorithm, which finds the solution of problem (1.7) as an approximate solution of problem (1.2). Their results show that their two-dimensional approximate method works as well as the more expensive exact full-dimensional minimization methods. We quote their results in Table 6.1 to show that the proposed indefinite dogleg path algorithm in this paper works as well as their algorithm. We tested all the problems Byrd, Schnabel, and Shultz reported in [7] except for the Watson problem (problem (20) in [17]), because the starting point for this problem is not clear.

Table 6.1 contains the results for this experiment, where IDSFA stands for Byrd, Schnabel, and Shultz's algorithm and TRIDPM for the algorithm proposed in this paper. ITR, NF, and NBF are the numbers of iterations, function evaluations, and matrix factorizations, respectively, needed to reach termination in the algorithms. NID stands for the number of indefinite matrices B_k that appeared in the iterations of Algorithm TRIDPM.

The results show that the behavior of the proposed indefinite dogleg path algorithm is as good as that of the IDSFA algorithm. While computing, most time is spent on matrix factorization. The average number of matrix factorizations per iteration, performed by the proposed algorithm on the Hessian matrices, in this experiment is 1.016, and the average number of factorizations on indefinite matrices is 1.105, whereas the two average numbers of factorizations performed by the IDSFA algorithm for the same set of test problems are 1.05 and 1.14, respectively. While the performances of the two methods for most test problems are comparable, the method TRIDPM is much better than the method IDSFA in solving Problems 5 and 6. It can be seen from Table 6.1 that all matrices obtained in using TRIDPM are positive definite. From our computational experiences, getting into a region where the Hessian matrices or their approximations are indefinite may delay convergence. As there is no corresponding information provided in [7], we do not know if the main cause for this

Table 6.1
Numerical results for small-scale problems.

		$10^k x^{(0)}$	TRIDPM				IDSFA		
Prob. no.	n	k	ITR	NF	NBF	NID	ITR	NF	NBF
1	3	0	9	11	9	3	10	12	12
		1	17	19	17	3	15	19	16
		2	18	22	18	0	16	20	16
2	6	0	48	59	48	29	47	63	51
3	3	0	2	3	2	0	2	3	2
4	10	0	14	15	14	0	14	15	14
		1	16	17	16	0	17	18	17
		2	23	24	23	7	23	24	23
5	10	0	14	15	14	0	31	43	31
		1	19	20	19	0	36	48	36
		2	28	29	28	0	43	57	43
6	4	0	6	7	6	0	75	100	75
		1	12	13	12	0	85	115	85
		2	18	19	18	0	83	112	83
6	10	0	10	11	10	0	92	123	92
		1	17	18	17	0	97	129	97
		2	23	24	23	0	105	139	105
7	4	0	9	10	9	0	8	9	8
		1	14	15	14	0	14	15	14
		2	20	21	20	0	20	21	20
8	3	0	26	29	26	8	25	29	25
9	10	0	11	15	12	6	9	12	9
		1	17	20	18	4	17	22	18
		2	12	13	15	3	14	15	14
10	2	0	26	32	28	3	22	27	22
		1	49	58	51	2	43	55	43
		2	120	139	120	1	110	146	110
11	4	0	13	14	13	0	15	16	15
		1	19	20	19	0	20	21	20
		2	25	26	25	0	26	27	26
12	2	0	8	10	8	3	9	11	11
		1	50	60	50	22	57	74	60
13	4	0	44	51	47	7	40	51	42
		1	51	57	53	7	45	59	47
		2	57	64	59	7	53	67	55
14	7	0	8	12	8	5	7	9	9
14	8	0	14	20	14	8	12	16	15
14	9	0	14	19	14	9	9	12	14
14	10	0	11	15	11	5	10	14	13

difference is that the moving trajectories of their method for these two test problems entered such a region.

The second experiment we conducted tests the performance of Algorithm TRIDPM for solving middle- and large-scale unconstrained problems. The test is carried out on 18 test functions quoted from [1], [15], [17], [19], [21], and [24]. The bottom half of Table 6.4 lists the problem numbers, names, and their references. The detailed information on these problems can be found in the corresponding references. We have not used any sparse technique for the matrix factorization. The experiment is only to show that the proposed indefinite dogleg path algorithm can also effectively solve middle- and large-scale optimization problems. Table 6.2 contains the results of this set of tests. The average number of matrix factorizations performed for this set of tests is 1.01, and the average number of factorizations for indefinite matrices is 1.02.

After we finished the preparation of this paper, Conn, Gould, and Toint [9] re-

Table 6.2
Numerical results for middle- and large-scale problems.

Prob. no.	n	ITR	NF	NBF	NID	Prob. no.	n	ITR	NF	NBF	NID
1	25	19	23	19	6	12	200	15	16	15	0
	50	25	30	25	15		500	15	16	15	0
2	50	7	8	7	0		1000	16	17	16	0
3	50	45	46	45	0	13	200	25	26	25	0
4	48	174	211	174	132		500	28	29	28	0
	100	347	418	347	302		1000	31	32	31	0
5	50	24	28	24	0	14	200	5	6	5	0
	100	17	18	17	0		500	5	6	5	0
6	50	13	17	13	9		1000	5	6	5	0
	100	22	26	22	19	15	200	7	8	7	0
7	50	16	22	19	3		500	8	9	8	0
	100	45	54	48	36		1000	8	9	8	0
8	100	117	142	118	64	16	200	6	7	6	0
	200	233	284	233	142		500	8	9	8	0
9	100	19	20	19	8		1000	7	8	7	0
	200	22	23	22	13	17	200	56	67	59	22
10	100	6	7	6	0		500	89	107	93	30
	200	6	7	6	0		1000	109	129	111	36
	500	6	7	6	0	18	200	29	34	32	19
11	100	21	24	21	0		500	25	28	25	17
	200	29	35	29	3		1000	48	56	51	38
	500	27	32	27	2						

Table 6.3
Numbers of iterations of NOPRC and TRIDMP on large-scale problems.

Prob. no.	n	NOPRC	TRIDMP	Prob. no.	n	NOPRC	TRIDMP
1	25	52	19	13	500	56	28
	50	78	25		1000	55	31
2	50	8	7	14	500	5	5
3	50	38	45		1000	5	5
4	100	18	347	15	500	10	8
5	100	25	17		1000	13	8
10	100	10	6	16	500	(81)	8
	500	10	6		1000	(136)	7
12	500	15	15	18	500	4	25
	1000	15	16		1000	4	48

ported the results of their numerical experiments with the LANCELOT package (release A) on a variety of large-scale unconstrained and constrained optimization problems. The purpose of their paper is to draw some conclusions on the respective merits of various variants of the algorithm in the LANCELOT package. The aim of the package is to find minimizers of smooth nonlinear functions with equality and simple bound constraints using an augmented Lagrangian approach. For unconstrained optimization, the algorithm reduces to a trust region–type method in which trust region subproblems are approximately solved using line search techniques; see [9] for a description of the algorithm. Variants of the algorithm are provided in the package, depending on different ways to generate gradients and Hessians of problem functions, use of scaling techniques, choices of different preconditioners, and direct or iterative solutions of the systems of linear equations in determining search directions.

From these variants we choose the variant NOPRC to compare with our algorithm because they both use exact gradient and Hessian evaluations but do not adopt scaling and preconditioning techniques, and therefore the comparison seems more reasonable.

Table 6.4
Lists of test problems.

Small-scale problems		
Prob. no.	Name of prob.	No. in [17]
1	Helical Valley Func.	7
2	Biggs EXP6 Func.	18
3	Gaussian Func.	9
4	Variably Dimension Func.	25
5	Penalty Func. I	23
6	Penalty Func. II	24
7	Brown and Dennis Func.	16
8	Gulf research Func.	11
9	Trigonometric Func.	26
10	Extended Rosenbrock Func.	21
11	Extended Powell Singular Func.	22
12	Beal Func.	5
13	Wood Func.	14
14	Chebyquad Func.	35

Middle- and large-scale problems		
Prob. no.	Name of prob.	Source
1	Chained Rosenbrock Func.	3.3 of [24]
2	Operation Research GOR	3.1 of [24]
3	Pseudo Penalty Func.	3.2 of [24]
4	Penalty Func. II	24 of [17]
5	Extended Woods Func.	14 of [17]
6	Trigonometric Func.	26 of [17]
7	Allgower Func.	2 of [15]
8	Generalized Rosenbrock Func.	4 of [19]
9	Zakharov Func.	305 of [21]
10	Extended Freudenstein and Roth	2 of [17]
11	Extended Rosenbrock Func.	21 of [17]
12	Extended Powell Singular Func.	22 of [17]
13	Penalty function I	23 of [17]
14	Broyden Tridiagonal Func.	6 of [1]
15	Broyden Banded Func.	7 of [1]
16	Broyden Seven Diagonal Func.	3.4 of [1]
17	Nearly Separable Func.	9 of [1]
18	Tridiagonal Func.	10 of [1]

Most test problems in our second set, except Problems 6–9 and 11, are also tested in [10], but only the numbers of iterations are listed for both and hence can be compared.

In Table 6.3 we list the numbers of iterations for all common test problems reported in both [10] for the variant NOPRC and Table 6.2 of this paper for Algorithm TRIDMP. The problem numbers in Table 6.3 are the same as in Table 6.2, and the parentheses around numbers indicate that the corresponding algorithm terminates at a different minimizer. From Table 6.3 we see that the performances of the two algorithms are comparable as the variant NOPRC is better for some problems (see Problems 4 and 18) while the algorithm TRIDMP is better for some other problems (for example, Problems 1, 13, and 16). It seems that Problem 4 has some irregular region(s) where the computation can make very little progress in each iteration, and our running of Algorithm TRIDMP enters such a region. However, it is very interesting that when we run the algorithm again by changing the value of the parameter γ_2 from 2 to 5 while maintaining other data, the number of iterations decreases drastically from 347 to 17. A possible reason for the improvement is that, with larger trust region radii, the iterative points can leave the troublesome region(s) more quickly.

The results obtained in our experiments show that the proposed indefinite dogleg path algorithm is easy to implement and very efficient for unconstrained optimization. This indefinite dogleg method has been successfully used as the main subproblem solver in several optimization problems. For example, in [8], Chen, Deng, and Zhang considered a partial update inexact Newton method for solving unary optimization problems by taking advantage of the special structure of the Hessian matrices of unary functions. As the approximate Hessian may not be positive definite, the method proposed in this paper was employed to solve their trust region subproblems. The numerical performance was very encouraging and improved the computational results of [13] remarkably.

Acknowledgments. The authors wish to express their sincere thanks to the referees for their constructive comments.

REFERENCES

[1] Y. Bing and G. Lin, *An efficient implementation of Merrill's method for sparse or partially separable systems of nonlinear equations*, SIAM J. Optim., 1 (1981), pp. 206–221.

[2] J. P. Bulteau and J. Ph. Vial, *Unconstrained Optimization by Approximation of a Projected Gradient Path*, Research Report 8352, Center for Operations Research and Econometrics, Université Catholique de Louvain, Belgium, 1983.

[3] J. P. Bulteau and J. Ph. Vial, *A restricted trust region algorithm for unconstrained optimization*, J. Optim. Theory Appl., 47 (1985), pp. 413–434.

[4] J. P. Buleau and J. Ph. Vial, *Curvilinear path and trust region in unconstrained optimization, a convergence analysis*, Math. Programming Stud., 30 (1987), pp. 82–101.

[5] J. R. Bunch and B. N. Parlett, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal., 8 (1971), pp. 639–655.

[6] J. R. Bunch, L. Kaufman, and B. N. Parlett, *Decomposition of a symmetric matrix*, Numer. Math., 27 (1976), pp. 95–109.

[7] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *Approximate solution of the trust region problem by minimization over two-dimensional subspaces*, Math. Programming, 40 (1988), pp. 247–263.

[8] L. Chen, N. Deng, and J. Zhang, *Modified partial-update Newton type algorithms for unary optimization*, J. Optim. Theory Appl., 97 (1998), pp. 385–406.

[9] A. R. Conn, N. Gould, and Ph. L. Toint, *Numerical experiments with the LANCELOT package (release A) for large-scale nonlinear optimization*, Math. Programming, 73 (1996), pp. 73–110.

[10] A. R. Conn, N. Gould, and Ph. L. Toint, *Intensive numerical tests with LANCELOT (release A): The complete results*, Tech. Report 92/15, Department of Mathematics, FUNDP, Namur, Belgium, 1992.

[11] J. E. Dennis and H. H. W. Mei, *Two new unconstrained optimization algorithms which use function and gradient values*, J. Optim. Theory Appl., 28 (1979), pp. 453–482.

[12] R. Fletcher, *Practical Methods of Optimization, Unconstrained Optimization*, Vol. 1, John Wiley, New York, 1980.

[13] D. Goldfarb and S. Wang, *Partial-update Newton methods for unary, factorable, and partially separable optimization*, SIAM J. Optim., 3 (1993), pp. 382–397.

[14] M. D. Hebden, *An Algorithm for Minimization Using Exact Second Derivatives*, Atomic Energy Research Establishment Report TP515, Harwell, England, 1973.

[15] M. Kojima and Y. Yamamoto, *A unified approach to the implementation of several fixed point algorithms and a new variable dimension algorithm*, Math. Programming, 28 (1984), pp. 288–328.

[16] J. J. Moré, *The Levenberg-Marquardt algorithm: Implementation and theory*, Lecture Notes in Math. 630, G. A. Watson, ed., Springer-Verlag, Berlin, Heidelberg, New York, 1978, pp. 105–116.

[17] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Software, 7 (1981), pp. 17–41.

[18] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[19] S. G. Nash, *Newton-type minimization via the Lanczos method*, SIAM J. Numer. Anal., 21 (1984), pp. 770–788.

[20] M. J. D. Powell, *A hybrid method for nonlinear equations*, in Numerical Methods for Nonlinear Algebraic Equations, Ph. Rabonowitz, ed., Gordon and Breach, New York, 1970, pp. 87–114.

[21] K. Schittkowski, *More Test Examples for Nonlinear Programming Codes*, Lecture Notes in Econom. and Math. Systems, 282, Springer-Verlag, New York, New York, 1987.

[22] G. A. Shultz, R. B. Schnabel, and R. H. Byrd, *A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties*, SIAM J. Numer. Anal., 22 (1985), pp. 47–67.

[23] D. C. Sorensen, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[24] Ph. L. Toint, *Some numerical results using a sparse matrix updating formula in unconstrained optimization*, Math. Comp., 32 (1978), pp. 839–851.

SIAM J. Optim.
Vol. 9, No. 3, pp. 668–689

PROXIMAL DECOMPOSITION VIA ALTERNATING LINEARIZATION*

KRZYSZTOF C. KIWIEL[†], CHARLES H. ROSA[‡], AND ANDRZEJ RUSZCZYŃSKI[§]

Abstract. A new approximate proximal point method for minimizing the sum of two convex functions is introduced. It replaces the original problem by a sequence of regularized subproblems in which the functions are alternately represented by linear models. The method updates the linear models and the prox center, as well as the prox coefficient. It is monotone in terms of the objective values and converges to a solution of the problem, if any. A dual version of the method is derived and analyzed. Applications of the methods to multistage stochastic programming problems are discussed and preliminary numerical experience is presented.

Key words. convex programming, large scale optimization, decomposition, proximal point methods, augmented Lagrangians, stochastic programming

AMS subject classifications. Primary, 65K05; Secondary, 90C25, 90C06, 90C15

PII. S1052623495288064

1. Introduction.

We present a method for solving structured convex optimization problems of the form

$$\text{minimize} \quad F(x) := h(x) + f(x), \tag{1.1}$$

where $h : \mathbb{R}^n \to (-\infty, +\infty]$ and $f : \mathbb{R}^n \to \mathbb{R}$ are closed proper convex functions.

Our method is an approximate version of the proximal point algorithm [Mar70, Roc76b] which generates a sequence

$$x^{k+1} = \arg\min_x F(x) + \tfrac{1}{2}\rho_k |x - x^k|^2 \quad \text{for } k = 1, 2, \dots , \tag{1.2}$$

starting from any point $x^1 \in \mathbb{R}^n$, where $| \cdot |$ is the Euclidean norm and $\{\rho_k\}$ is a sequence of positive numbers. To implement the iteration (1.2) approximately, our method employs a sequence of subproblems of the form

$$\min_x h(x) + \tilde{f}_k(x) + \tfrac{1}{2}\rho_k |x - x^k|^2 \tag{1.3}$$

and

$$\min_x \tilde{h}_k(x) + f(x) + \tfrac{1}{2}\rho_k |x - x^k|^2, \tag{1.4}$$

where $\tilde{f}_k$ and $\tilde{h}_k$ are linear models of f and h, respectively. This is the reason for naming our approach the *alternating linearization method.*

Our method makes it possible to exploit structural properties of h and f separately, which may be useful in many applications. Let us mention two examples, which will be treated in more detail later.

*Received by the editors June 22, 1995; accepted for publication (in revised form) May 4, 1998; published electronically April 19, 1999. This work was supported by the International Institute for Applied Systems Analysis, Laxenburg, Austria.

http://www.siam.org/journals/siopt/9-3/28806.html

[†] Systems Research Institute, Newelska 6, 01–447 Warsaw, Poland (kiwiel@ibspan.waw.pl).

[‡] Sabre Decision Technologies, 1 East Kirkwood Boulevard, Southlake, TX 76092 (rosa@sabre.com).

[§] Department of Management Science and Information Systems, Rutgers University, 94 Rockafeller Road, Piscataway, NJ 08854 (rusz@everest.rutgers.edu).

Example 1.1. Consider the separable problem with linking constraints:

$$\min \sum_{j=1}^{N} \psi_j(x_j) \quad \text{s.t.} \quad \sum_{j=1}^{N} A_j x_j = b,$$

where $\psi_j : \mathbb{R}^{n_j} \to (-\infty, +\infty]$ are closed proper convex functions and A_j are $m \times n_j$ matrices, $j = 1, \ldots, N$. Application of the *multiplier method* [Ber82, Hes69, Pow69, Roc76a] leads to subproblems of minimizing the augmented Lagrangian:

$$\min_x \sum_{j=1}^{N} \left(\psi_j(x_j) - \langle \lambda, A_j x_j \rangle\right) + \langle \lambda, b \rangle + \tfrac{1}{2}\rho |Ax - b|^2,$$

where $\lambda \in \mathbb{R}^m$ is the current vector of Lagrange multipliers, $\rho > 0$ is a penalty coefficient, $x = (x_1, \ldots, x_N)$, and $A = [A_1 \ \cdots \ A_n]$. This problem has the form (1.1) with $f(x) = \frac{1}{2}\rho|Ax - b|^2$, in which (1.3) is decomposable into independent subproblems for each $j = 1, \ldots, N$, while (1.4) is just a least squares problem.

Example 1.2. Let us now consider the decomposable problem with linking variables:

$$\min_y \varphi(y) + \sum_{j=1}^{N} \psi_j(y)$$

with closed proper convex functions $\varphi : \mathbb{R}^n \to (-\infty, +\infty]$ and $\psi_j : \mathbb{R}^n \to (-\infty, +\infty]$, $j = 1, \ldots, N$. Splitting variables and dualization [BeT89, p. 231] lead to the problem

$$\min_x \sum_{j=1}^{N} \psi_j^*(x_j) + \varphi^*\left(-\sum\nolimits_{j=1}^{N} x_j\right),$$

where φ^* and ψ_j^* are the conjugates of φ and ψ_j and $x_j \in \mathbb{R}^n$, $j = 1, \ldots, N$, are dual variables. This dual problem has the form (1.1), in which (1.3) decomposes into independent subproblems for $j = 1, \ldots, N$. All these subproblems and (1.4) are much easier to solve than the original formulation.

The general objective of our work has been pursued by many researchers; in particular, the well-known *operator splitting methods* should be mentioned here (see [Eck94, EcB92, EcF97, MOT95, MaT93, Spi85, Tse91, Tse90]). Their dual versions are known as *alternating direction methods* [BeT89, EcB92, EcF94, Fuk92, Gab83, KDLM96]. Other related recent research is described in [ChT94, FHN$^+$96, Tse97].

Our approach, although having parallel objectives, is fundamentally different. Contrary to earlier works, our method is *monotone* in terms of the values of the objective $F = h + f$. To achieve this, we employ two different types of updates of the models in (1.3) and (1.4). The first update changes only the approximations $\tilde{f}_k$ and $\tilde{h}_k$, while keeping x^k fixed; the second one updates x^k as well. In this way we ensure that $F(x^{k+1}) < F(x^k)$ whenever x^k is changed. We also allow changes in the value of the penalty coefficient ρ_k. On the other hand, our method is less general than some other ones because it requires that f be finite-valued; this, however, does not seem to limit its usefulness, at least in the applications that are of interest to us.

In section 2 we present the main idea of the method: approximate implementation of the proximal step by using alternating linearizations. In section 3 this idea is used within a descent algorithm for minimizing F. Its convergence is proved in section 4. The dual version of the method is described in section 5. In section 6 we discuss applications to stochastic programming. Preliminary computational experience is reported in section 7.

2. Proximal step by alternating linearization.

Let us first describe and analyze an algorithm that employs subproblems (1.3) and (1.4) for finding an approximation to the proximal point

$$p(\bar{x}) = \arg\min_x \; h(x) + f(x) + \tfrac{1}{2}\rho|x - \bar{x}|^2, \tag{2.1}$$

where $\bar{x} \in \mathbb{R}^n$ and $\rho > 0$ are fixed.

ALGORITHM 2.1.

Step 0: Choose $z_f^0 \in \mathbb{R}^n$ and $g_f^0 \in \partial f(z_f^0)$. Define $\tilde{f}_1(\cdot) = f(z_f^0) + \langle g_f^0, \cdot - z_f^0\rangle$. Set $k = 1$.

Step 1: Find the solution z_h^k of the following h-subproblem:

$$\min_x \; h(x) + \tilde{f}_k(x) + \tfrac{1}{2}\rho|x - \bar{x}|^2. \tag{2.2}$$

Set

$$g_h^k = -g_f^{k-1} - \rho(z_h^k - \bar{x}) \tag{2.3}$$

and define

$$\tilde{h}_k(\cdot) = h(z_h^k) + \langle g_h^k, \cdot - z_h^k\rangle. \tag{2.4}$$

Step 2: Find the solution z_f^k of the following f-subproblem:

$$\min_x \; \tilde{h}_k(x) + f(x) + \tfrac{1}{2}\rho|x - \bar{x}|^2. \tag{2.5}$$

Set

$$g_f^k = -g_h^k - \rho(z_f^k - \bar{x}) \tag{2.6}$$

and define

$$\tilde{f}_{k+1}(\cdot) = f(z_f^k) + \langle g_f^k, \cdot - z_f^k\rangle. \tag{2.7}$$

Step 3: Increase k by 1 and go to Step 1.

Our objective is to prove that $z_h^k \to p(\bar{x})$.

Remark 2.2. The necessary and sufficient condition of optimality for (2.2) has the form

$$0 \in \partial h(z_h^k) + g_f^{k-1} + \rho(z_h^k - \bar{x}), \tag{2.8}$$

so the vector g_h^k (cf. (2.3)) is the element of $\partial h(z_h^k)$ that satisfies this condition. Hence $\tilde{h}_k \le h$ by the subgradient inequality. Similarly, the vector g_f^k (cf. (2.6)) is the element of $\partial f(z_f^k)$ that satisfies the optimality condition for (2.5): $0 \in g_h^k + \partial f(z_f^k) + \rho(z_f^k - \bar{x})$. Therefore, $\tilde{f}_{k+1} \le f$ and $\tilde{F}_k := h + \tilde{f}_k$ is a lower approximation of the objective $F = h + f$.

Let us denote by

$$\eta_k = h(z_h^k) + \tilde{f}_k(z_h^k) + \tfrac{1}{2}\rho|z_h^k - \bar{x}|^2 \tag{2.9}$$

and

$$\eta_{k+1/2} = \tilde{h}_k(z_f^k) + f(z_f^k) + \tfrac{1}{2}\rho|z_f^k - \bar{x}|^2$$

the optimal values of (2.2) and (2.5), respectively. The way in which the successive linearizations $\tilde{f}_k$ and $\tilde{h}_k$ are generated ensures monotonicity of $\{\eta_k\}$:

$$\eta_k \leq \eta_{k+1/2} \leq \eta_{k+1}. \tag{2.10}$$

Indeed, the change from (2.2) to (2.5) at iteration k can be described in two steps:

(a) replace $h(\cdot)$ by $\tilde{h}_k(\cdot)$;

(b) replace $\tilde{f}_k(\cdot)$ by f.

By construction of $\tilde{h}_k$ (cf. (2.4)), operation (a) does not change the solution and value of (2.2), since $\tilde{h}_k(z_h^k) = h(z_h^k)$ and the gradient of $\tilde{h}_k$ is the subgradient of h at z_h^k that satisfies (2.8). Thus

$$\eta_k = \min_x \tilde{h}_k(x) + \tilde{f}_k(x) + \tfrac{1}{2}\rho|x - \bar{x}|^2.$$

Operation (b) can only increase the optimal value of the last problem, because $f \geq \tilde{f}_k$, so $\eta_{k+1/2} \geq \eta_k$. Similarly, replacing f by $\tilde{f}_{k+1}$ does not change the solution and value of (2.5), because g_f^k was chosen to satisfy the optimality condition (cf. Remark 2.2) and $\tilde{f}_{k+1}(z_f^k) = f(z_f^k)$. Replacing $\tilde{h}_k$ by h can only increase the optimal value, so $\eta_{k+1} \geq \eta_{k+1/2}$.

To estimate the increase from η_k to $\eta_{k+1/2}$ for operation (b), consider the family of relaxations of (2.5) at iteration k:

$$\min_x \{\, Q_k(x,\mu) = \tilde{h}_k(x) + (1-\mu)(\alpha_p^k + \langle p^k, x\rangle) + \mu(\alpha_g^k + \langle g^k, x\rangle) + \tfrac{1}{2}\rho|x - \bar{x}|^2 \,\}, \tag{2.11}$$

where $\mu \in [0,1]$, $p^k = g_f^{k-1}$, $\alpha_p^k = f(z_f^{k-1}) - \langle p^k, z_f^{k-1}\rangle$, and $\alpha_g^k = f(z_h^k) - \langle g^k, z_h^k\rangle$ for an arbitrary $g^k = g_f(z_h^k) \in \partial f(z_h^k)$. Since $\tilde{f}_k(\cdot) = \alpha_p^k + \langle p^k, \cdot\rangle$ and $\alpha_g^k + \langle g^k, \cdot\rangle$ are lower approximations of f, (2.11) is a relaxation of (2.5) for all $\mu \in [0,1]$. For $\mu = 0$ the solution and value of (2.11) coincide with those of (2.2). Thus, the difference between the optimal values of (2.5) and (2.2) can be estimated from below by the increase in the optimal value $\hat{Q}_k(\mu)$ of (2.11) when μ moves away from zero. Formally,

$$\eta_{k+1/2} - \eta_k \geq \max_{\mu\in[0,1]} \hat{Q}_k(\mu) - \hat{Q}_k(0).$$

Lemma 2.3. *The following inequalities hold for any* $g^k \in \partial f(z_h^k)$:

(i) $\max_{\mu\in[0,1]} \hat{Q}_k(\mu) - \hat{Q}_k(0) \geq \hat{Q}_k(\bar{\mu}_k) - \hat{Q}_k(0) \geq \frac{1}{2}\bar{\mu}_k\delta_k$,

(ii) $\eta_{k+1} \geq \eta_{k+1/2} \geq \eta_k + \frac{1}{2}\bar{\mu}_k\delta_k$,

where $\delta_k = F(z_h^k) - \tilde{F}_k(z_h^k) \geq 0$ *and* $\bar{\mu}_k = \min\{1, \delta_k\rho/|g^k - p^k|^2\}$.

Proof. Note that $\delta_k \geq 0$, since $f \geq \tilde{f}_k$, so $\bar{\mu}_k \in [0,1]$. By direct calculation, the solution of (2.11) has the form $\hat{x}(\mu) = \bar{x} - [g_h^k + p^k + \mu(g^k - p^k)]/\rho$, so using the definitions following (2.11) and the fact that $\hat{x}(0) = z_h^k$, the derivative of $\hat{Q}_k$ can be expressed as follows:

$$\begin{aligned}
\hat{Q}_k'(\mu) &= \langle g^k - p^k, \hat{x}(\mu)\rangle + \alpha_g^k - \alpha_p^k \\
&= \langle g^k - p^k, \hat{x}(\mu) - \hat{x}(0)\rangle + \big[\alpha_g^k + \langle g^k, \hat{x}(0)\rangle\big] - \big[\alpha_p^k + \langle p^k, \hat{x}(0)\rangle\big] \\
&= \langle g^k - p^k, \hat{x}(\mu) - \hat{x}(0)\rangle + F(z_h^k) - \tilde{F}_k(z_h^k) \\
&= -\mu|g^k - p^k|^2/\rho + \delta_k.
\end{aligned}$$

Thus

$$\hat{Q}_k(\bar{\mu}_k) - \hat{Q}_k(0) = \int_0^{\bar{\mu}_k} \hat{Q}_k'(\mu)d\mu = \bar{\mu}_k \left(\delta_k - \tfrac{1}{2}\bar{\mu}_k|g^k - p^k|^2/\rho\right).$$

Substituting the definition of $\bar{\mu}_k$ in the expression in the parentheses yields (i). Assertion (ii) follows from (i) and (2.10). □

THEOREM 2.4. *The sequences of points* $\{z_h^k\}$ *and approximations* $\{\tilde{F}_k\}$ *generated by Algorithm* 2.1 *have the following properties:*

(i) $|z_h^k - p(\bar{x})| \le \left\{[F(z_h^k) - \tilde{F}_k(z_h^k)]/\rho\right\}^{1/2}$ *for* $k = 1, 2, \ldots$.

(ii) $\lim_{k\to\infty} \left[F(z_h^k) - \tilde{F}_k(z_h^k)\right] = 0$.

(iii) $\lim_{k\to\infty} z_h^k = p(\bar{x})$.

Proof. Since $F \ge \tilde{F}_k$ and z_h^k solves the strongly convex problem (2.2), we have [Roc76b]

$$\begin{aligned} F(p(\bar{x})) + \tfrac{1}{2}\rho|p(\bar{x}) - \bar{x}|^2 &\ge \tilde{F}_k(p(\bar{x})) + \tfrac{1}{2}\rho|p(\bar{x}) - \bar{x}|^2 \\ &\ge \tilde{F}_k(z_h^k) + \tfrac{1}{2}\rho|z_h^k - \bar{x}|^2 + \tfrac{1}{2}\rho|p(\bar{x}) - z_h^k|^2. \end{aligned} \tag{2.12}$$

Similarly, $p(\bar{x})$ solves the strongly convex problem in (2.1), so

$$F(z_h^k) + \tfrac{1}{2}\rho|z_h^k - \bar{x}|^2 \ge F(p(\bar{x})) + \tfrac{1}{2}\rho|p(\bar{x}) - \bar{x}|^2 + \tfrac{1}{2}\rho|p(\bar{x}) - z_h^k|^2.$$

Adding the last two inequalities and simplifying, we get $F(z_h^k) - \tilde{F}_k(z_h^k) \ge \rho|p(\bar{x}) - z_h^k|^2$, which proves assertion (i). Next, (2.12) can be equivalently written as (cf. (2.9))

$$\tfrac{1}{2}\rho|p(\bar{x}) - z_h^k|^2 \le F(p(\bar{x})) + \tfrac{1}{2}\rho|p(\bar{x}) - \bar{x}|^2 - \eta_k. \tag{2.13}$$

By Lemma 2.3, $\{\eta_k\}$ is nondecreasing, so (2.13) implies that $\{z_h^k\}$ is bounded. Then $\{g^k\}$ is bounded as well, because $g^k \in \partial f(z_h^k)$ for all k and f is finite-valued (cf. [Roc70, Thm. 24.7]). By an analogous argument, using the inequality

$$\tfrac{1}{2}\rho|p(\bar{x}) - z_f^k|^2 \le F(p(\bar{x})) + \tfrac{1}{2}\rho|p(\bar{x}) - \bar{x}|^2 - \eta_{k+1/2},$$

we see that z_f^k and $p^k = g_f^{k-1} \in \partial f(z_f^{k-1})$ are bounded. By (2.13), the sequence $\{\eta_k\}$ is bounded from above, so Lemma 2.3 implies that it converges and $\bar{\mu}_k\delta_k \to 0$. Since $\{|g^k - p^k|\}$ is bounded, assertion (ii) follows from the definition of $\bar{\mu}_k$ (cf. Lemma 2.3). The final assertion is a consequence of (i) and (ii). □

Remark 2.5. Algorithm 2.1 can be used in the implementable proximal point schemes of [Aus86, CoL93, EcB92, GoT89, Gül91, Lem89, Roc76b]. Indeed, Theorem 2.4 ensures that for every $\epsilon > 0$ we can find in finitely many steps a point z_h^k such that $|z_h^k - p(\bar{x})| \le \epsilon$. An alternative scheme will be presented in the next section.

3. The alternating linearization method.

The algorithm below employs a simple descent test for terminating the loop of Algorithm 2.1 in order to update the prox center.

ALGORITHM 3.1.

Step 0: Select $x^1 \in \operatorname{dom} h$, $z_f^0 \in \mathbb{R}^n$, and $g_f^0 \in \partial f(z_f^0)$. Define $\tilde{f}_1(\cdot) = f(z_f^0) + \langle g_f^0, \cdot - z_f^0\rangle$. Choose parameters $\rho_1 \ge \rho_{\min} > 0$, $\kappa > 1$, $\beta_0 > 0$, $\beta_1 \in (0, 1)$. Set $k = 1$.

Step 1: Find the solution z_h^k of the h-subproblem:

$$\min_x \; h(x) + \tilde{f}_k(x) + \tfrac{1}{2}\rho_k|x - x^k|^2. \tag{3.1}$$

Set $g_h^k = -g_f^{k-1} - \rho_k(z_h^k - x^k)$ and define $\tilde{h}_k(\cdot) = h(z_h^k) + \langle g_h^k, \cdot - z_h^k \rangle$.

Step 2: Let $\tilde{F}_k = h + \tilde{f}_k$. Set

$$v_k = F(x^k) - \tilde{F}_k(z_h^k). \tag{3.2}$$

If

$$F(z_h^k) \le F(x^k) - \beta_1 v_k, \tag{3.3}$$

then set $x^{k+1} = z_h^k$ (descent step); otherwise set $x^{k+1} = x^k$ (null step).

Step 3: If $x^{k+1} = z_h^k$, then choose $\rho_{k+1} \in [\max\{\rho_{\min}, \rho_k/\kappa\}, \rho_k]$. If $x^{k+1} = x^k$ and

$$\delta_k := F(z_h^k) - \tilde{F}_k(z_h^k) \ge \beta_0 \frac{v_k}{|z_h^k - x^k|},$$

then choose $\rho_{k+1} \ge \rho_k$; else set $\rho_{k+1} = \rho_k$.

Step 4: Find the solution z_f^k of the f-subproblem:

$$\min_x \; \tilde{h}_k(x) + f(x) + \tfrac{1}{2}\rho_{k+1}|x - x^{k+1}|^2. \tag{3.4}$$

Set $g_f^k = -g_h^k - \rho_{k+1}(z_f^k - x^{k+1})$ and define $\tilde{f}_{k+1}(\cdot) = f(z_f^k) + \langle g_f^k, \cdot - z_f^k \rangle$.

Step 5: Increase k by 1 and go to Step 1.

We shall preserve the notation of the previous section, with only necessary changes. Thus

$$\eta_k = \tilde{F}_k(z_h^k) + \tfrac{1}{2}\rho_k|z_h^k - x^k|^2 \tag{3.5}$$

will denote the optimal value of (3.1), and $\eta_{k+1/2}$ that of (3.4).

By construction (cf. Remark 2.2), $g_f^k \in \partial f(z_f^k)$ and $\tilde{F}_k \le F$, so $\eta_k \le F(x^k)$ and $v_k \ge 0$. Thus (3.3) implies that $\{F(x^k)\}$ is nonincreasing and $\{x^k\} \subset \operatorname{dom} F$. It will become clear that if $v_k = 0$ or $\eta_k = F(x^k)$, then $x^k \in \operatorname{Arg\,min} F$. As observed by a referee, one can write the conditions of Steps 2 and 3 alternatively as

$$F(x^k) - F(z_h^k) \ge \beta_1[F(x^k) - \tilde{F}_k(z_h^k)],$$

$$F(x^k) - F(z_h^k) \le (1 - \beta_0/|z_h^k - x^k|)[F(x^k) - \tilde{F}_k(z_h^k)],$$

with some flavor of a trust-region scheme.

4. Convergence.

Let us first make a simple observation concerning the optimal values of (3.1) and (3.4).

LEMMA 4.1. *The following inequalities are true for all $k = 1, 2, \ldots$:*

(i) $\frac{1}{2}\rho_k|z_h^k - x^k|^2 \le \frac{1}{2}v_k \le F(x^k) - \eta_k \le v_k$,

(ii) $\frac{1}{2}\rho_{k+1}|z_f^k - x^{k+1}|^2 \le F(x^{k+1}) - \eta_{k+1/2}$.

Proof. Relations (3.2) and (3.5) yield $F(x^k) - v_k \le \eta_k$ and hence the right inequality of (i). Next, note that by construction (cf. Step 1),

$$-\rho_k(z_h^k - x^k) = g_h^k + g_f^{k-1} \in \partial\tilde{F}_k(z_h^k). \tag{4.1}$$

Therefore, the left inequality in (i) follows from the subgradient inequality, since

$$v_k = F(x^k) - \tilde{F}_k(z_h^k) \ge \tilde{F}_k(x^k) - \tilde{F}_k(z_h^k) \ge \rho_k|z_h^k - x^k|^2.$$

Thus

$$\eta_k = F(x^k) - v_k + \tfrac{1}{2}\rho_k|z_h^k - x^k|^2 \leq F(x^k) - \tfrac{1}{2}v_k,$$

which completes the proof of (i). Assertion (ii) can be obtained similarly. □

The following result is a simple consequence of Lemma 4.1 and Theorem 2.4.

COROLLARY 4.2. *If* $v_k = 0$*, then* $x^k \in \operatorname{Arg\,min} F$*.*

Proof. By Lemma 4.1(i) and (3.2), $z_h^k = x^k$ and $\tilde{F}_k(z_h^k) = F(z_h^k) = F(x^k)$. Then Theorem 2.4(i) yields $x^k = z_h^k = \arg\min F + \frac{1}{2}\rho_k|\cdot - x^k|^2$, so we have $x^k \in \operatorname{Arg\,min} F$ [Roc76b]. □

We split our convergence analysis into several stages, starting from the case of an infinite series of null steps. Our objective is to prove that in this case the optimal values of (3.1) and (3.4) converge to $F(x^{k_0})$, where x^{k_0} is the last point to which a descent step was made.

LEMMA 4.3. *If a null step is made at iteration* k*, then*

$$\eta_{k+1} \geq \eta_k + \tfrac{1}{2}(1 - \beta_1)\bar{\mu}_k v_k,$$

where $\bar{\mu}_k \geq \min\{1, (1 - \beta_1)v_k\rho_k/|g_f(z_h^k) - g_f^{k-1}|^2\}$ *for any* $g_f(z_h^k) \in \partial f(z_h^k)$*.*

Proof. If (3.3) fails, then $\delta_k = F(z_h^k) - \tilde{F}_k(z_h^k) = F(z_h^k) - F(x^k) + v_k > (1 - \beta_1)v_k$. Hence if $\rho_{k+1} = \rho_k$, then Lemma 2.3(ii) yields $\eta_{k+1/2} \geq \eta_k + \frac{1}{2}(1 - \beta_1)\bar{\mu}_k v_k$. When $\rho_{k+1} > \rho_k$, the minimum value of (3.4) can only be greater. Finally, $\eta_{k+1} \geq \eta_{k+1/2}$ as in (2.10). □

LEMMA 4.4. *If the set* $\mathcal{K} = \{k : x^{k+1} \neq x^k\}$ *is finite, then* $v_k \to 0$*.*

Proof. Let k_0 be such that $x^k = x^{k_0}$ for all $k \geq k_0$. By Lemma 4.3, $\{\eta_k\}$ is nondecreasing for $k \geq k_0$, and hence convergent, because $\eta_k \leq F(x^{k_0})$, so $\eta_{k+1} - \eta_k \to 0$. Since $\rho_k \geq \rho_{\min} > 0$ for all k, and $\{x^k\}$ is bounded, so are $\{z_h^k\}$ and $\{z_f^k\}$ (cf. Lemma 4.1), and hence also $g_f(z_h^k) \in \partial f(z_h^k)$ and $g_f^k \in \partial f(z_f^k)$, because f is locally Lipschitz (cf. [Roc70, Thm. 24.7]). Therefore, Lemma 4.3 yields $\bar{\mu}_k v_k \to 0$ and $v_k \to 0$. □

Let us now pass to the case of infinitely many descent steps.

LEMMA 4.5. *Suppose the set* $\mathcal{K} = \{k : x^{k+1} \neq x^k\}$ *is infinite and* $\inf F > -\infty$*. Then*

(i) $\sum_{k\in\mathcal{K}} v_k < \infty$;
(ii) $\lim_{k\to\infty} v_k = 0$;
(iii) $\lim_{k\to\infty} \left[F(x^k) - \eta_k\right] = 0$;
(iv) $\lim_{k\to\infty} \left[F(x^{k+1}) - \eta_{k+1/2}\right] = 0$.

Proof. For each $k \in \mathcal{K}$, a descent step occurs with $F(x^k) - F(x^{k+1}) \geq \beta_1 v_k \geq 0$. Summing these inequalities over k and using monotonicity and boundedness of $\{F(x^k)\}$, we get (i) and $v_k \to 0$ for $k \in \mathcal{K}$. In view of Lemma 4.1, $F(x^k) - \eta_k \to 0$ for $k \in \mathcal{K}$. To show convergence of the whole sequences, let us denote by $l(k)$ the number of the last iteration with a descent step preceding iteration k. By Lemma 4.3,

$$0 \leq F(x^k) - \eta_k \leq F(x^{l(k)+1}) - \eta_{l(k)+1}. \tag{4.2}$$

From (i) and Lemma 4.1 we obtain $F(x^{l(k)}) - \eta_{l(k)} \to 0$. It remains to relate $F(x^{l(k)+1}) - \eta_{l(k)+1}$ to $F(x^{l(k)}) - \eta_{l(k)}$. The changes in (3.1) following a descent step at iteration $l = l(k)$ can be decomposed into the following operations:

(a) linearization of h and the shift of the prox center x^l to $x^{l+1} = z_h^l$;
(b) the change of the penalty parameter ρ_l to $\rho_{l+1} \in [\rho_l/\kappa, \rho_l]$;
(c) replacement of $\tilde{f}_l$ by $\tilde{f}_{l+1}$.

Denote by $\eta_l^{(b)}$ the resulting optimal value of (3.1) after partial modifications (a) and (b), and let $\bar{F}_l = \tilde{h}_l + \tilde{f}_l$ ($\bar{F}_l$ is linear and $\bar{F}_l \le F$; cf. Remark 2.2). By construction, $\bar{F}_l(x^{l+1}) = \tilde{F}_l(x^{l+1})$ and $g^l_{\tilde{F}} = g^l_h + g^{l-1}_f \in \partial \tilde{F}_l(x^{l+1})$ is such that $g^l_{\tilde{F}} = \nabla \bar{F}_l(x^{l+1})$, $x^{l+1} - x^l = -g^l_{\tilde{F}}/\rho_l$ (cf. (4.1)) and

$$\eta_l = \min_x \ \bar{F}_l(x^{l+1}) + \langle g^l_{\tilde{F}}, x - x^{l+1}\rangle + \tfrac{1}{2}\rho_l |x - x^l|^2,$$

so

$$\eta_l = \bar{F}_l(x^{l+1}) + \tfrac{1}{2}|g^l_{\tilde{F}}|^2/\rho_l = \bar{F}_l(x^l) - \tfrac{1}{2}|g^l_{\tilde{F}}|^2/\rho_l.$$

Similarly,

$$\eta_l^{(b)} = \min_x \{\, \bar{F}_l(x^{l+1}) + \langle g^l_{\tilde{F}}, x - x^{l+1}\rangle + \tfrac{1}{2}\rho_{l+1}|x - x^{l+1}|^2 \,\} = \bar{F}_l(x^{l+1}) - \tfrac{1}{2}|g^l_{\tilde{F}}|^2/\rho_{l+1},$$

so

$$\bar{F}_l(x^l) - \eta_l = \frac{1}{2\rho_l}|g^l_{\tilde{F}}|^2 = \frac{\rho_{l+1}}{\rho_l}\left[\bar{F}_l(x^{l+1}) - \eta_l^{(b)}\right] \ge \frac{1}{\kappa}\left[\bar{F}_l(x^{l+1}) - \eta_l^{(b)}\right].$$

Finally, operation (c) is a hypothetical null step, so by Lemma 2.3

$$\eta_{l+1} \ge \eta_{l+1/2} \ge \eta_l^{(b)}.$$

Combining the last two relations and noting that at descent steps $F(x^{l+1}) \le F(x^l) = \bar{F}_l(x^{l+1}) + v_l$, we obtain for each descent step $l(k)$ the relation

$$F(x^{l(k)+1}) - \eta_{l(k)+1} \le \kappa \left[F(x^{l(k)}) - \eta_{l(k)}\right] + v_{l(k)}.$$

Since the right side of the above inequality converges to 0, and the left side is nonnegative, we must have $\lim_{k\to\infty} F(x^{l(k)+1}) - \eta_{l(k)+1} = 0$. Using this relation in (4.2) we conclude that $F(x^k) - \eta_k \to 0$ and $F(x^{k+1}) - \eta_{k+1/2} \to 0$; i.e., (iii) and (iv) hold. Assertion (ii) follows from Lemma 4.1. □

LEMMA 4.6. *Suppose the set $\mathcal{K} = \{k : x^{k+1} \ne x^k\}$ is infinite. If there exists a point $\tilde{x}$ such that $F(x^k) \ge F(\tilde{x})$ for all k, then $\{x^k\}$ converges to a point $\bar{x} \in \operatorname{dom} F$.*

Proof. Fix $k \in \mathcal{K}$. We have

$$|x^{k+1} - \tilde{x}|^2 = |x^k - \tilde{x}|^2 + 2\langle x^{k+1} - \tilde{x}, x^{k+1} - x^k\rangle - |x^{k+1} - x^k|^2. \tag{4.3}$$

By (4.1), $g^k_{\tilde{F}} = g^k_h + g^{k-1}_f = -\rho_k(x^{k+1} - x^k) \in \partial \tilde{F}_k(x^{k+1})$, so

$$\begin{aligned}\rho_k\langle x^{k+1} - \tilde{x}, x^{k+1} - x^k\rangle &= \langle \tilde{x} - x^{k+1}, g^k_{\tilde{F}}\rangle \\ &\le \tilde{F}_k(\tilde{x}) - \tilde{F}_k(x^{k+1}) \le F(\tilde{x}) - F(x^k) + v_k\end{aligned}$$

by (3.2). Using this inequality in (4.3) yields

$$|x^{k+1} - \tilde{x}|^2 \le |x^k - \tilde{x}|^2 + 2v_k/\rho_k, \qquad k \in \mathcal{K}.$$

Since $\{\rho_k\}$ is bounded away from zero by construction, the last inequality and assertion (i) of Lemma 4.5 imply that the sequence $\{x^k\}$ is bounded. Hence, it has an accumulation point $\bar{x}$. By monotonicity of $\{F(x^k)\}$ and closedness of F, $F(\bar{x}) \le F(x^k)$ for all k, so we can replace $\tilde{x}$ by $\bar{x}$ in the preceding argument, concluding that $\bar{x}$ is the only accumulation point, since $\sum_{k\in\mathcal{K}, k\ge l} v_k \to 0$ as $l \to \infty$. □

LEMMA 4.7. *If there exists a point $\tilde{x}$ such that $F(x^k) \ge F(\tilde{x})$ for all k, then*

(i) *$v_k \to 0$, $F(x^k) - \eta_k \to 0$, and $F(x^{k+1}) - \eta_{k+1/2} \to 0$, as $k \to \infty$;*

(ii) *the sequence $\{x^k\}$ converges to a point $\bar{x} \in \operatorname{Arg\,min} F$, and $F(x^k) \downarrow F(\bar{x})$.*

Proof. By Lemmas 4.4–4.6, $\{x^k\}$ converges to some $\bar{x} \in \operatorname{dom} F$ and assertion (i) holds.

By construction, the linear function $\bar{F}_k = \tilde{h}_k + \tilde{f}_k$ minorizes F, and $\bar{F}_k(z_h^k) = \tilde{F}_k(z_h^k) = F(x^k) - v_k$, so for all $x \in \mathbb{R}^n$,

$$F(x) \geq \bar{F}_k(x) = \bar{F}_k(z_h^k) + \langle g_h^k + g_f^{k-1}, x - z_h^k \rangle = F(x^k) + \langle g_h^k + g_f^{k-1}, x - z_h^k \rangle - v_k. \tag{4.4}$$

By Lemma 4.1(i),

$$\rho_k |z_h^k - x^k|^2 \leq v_k \to 0. \tag{4.5}$$

However, $\rho_k \geq \rho_{\min}$ for all k, so $z_h^k - x^k \to 0$. Similarly, $z_f^k - x^k \to 0$. Thus $z_h^k \to \bar{x}$ and $z_f^k \to \bar{x}$, so $g_f^k \in \partial f(z_f^k)$ are bounded, since f is locally Lipschitz. We have to consider two cases.

Case 1. There exists $\bar{\rho}$ such that $\rho_k \leq \bar{\rho}$ for all k. Since $z_h^k - x^k \to 0$, at Step 1

$$g_h^k + g_f^{k-1} = -\rho_k (z_h^k - x^k) \to 0. \tag{4.6}$$

Hence (4.4) yields $F(x) \geq \lim F(x^k) \geq F(\bar{x})$, using the closedness of F.

Case 2. $\limsup_k \rho_k = +\infty$. Since f is locally Lipschitz,

$$\begin{aligned} \delta_k = F(z_h^k) - \tilde{F}_k(z_h^k) &= f(z_h^k) - \tilde{f}_k(z_h^k) \\ &= f(z_h^k) - f(z_f^{k-1}) - \langle g_f^{k-1}, z_h^k - z_f^{k-1} \rangle \to 0. \end{aligned} \tag{4.7}$$

The penalty coefficient is increased infinitely many times, so (cf. Step 3) there must be a subsequence $\mathcal{K}$ such that

$$\delta_k \geq \beta_0 v_k / |z_h^k - x^k| \quad \text{for all } k \in \mathcal{K}.$$

Hence, dividing the inequality in (4.5) by $|z_h^k - x^k|$ and using (4.7), we get at Step 1

$$g_h^k + g_f^{k-1} = -\rho_k (z_h^k - x^k) \to 0, \qquad k \in \mathcal{K}. \tag{4.8}$$

Passing to the limit in (4.4) for $k \in \mathcal{K}$ and using (4.8), we obtain (ii) in this case, too. ▯

Our results can be summarized as follows.

THEOREM 4.8. *Algorithm* 3.1 *generates a sequence $\{x^k\}$ with the following properties:*

(i) *$F(x^k) \downarrow \inf F$.*

(ii) *If $\operatorname{Arg\,min} F \neq \emptyset$ then $\{x^k\}$ converges to a point $\hat{x} \in \operatorname{Arg\,min} F$.*

(iii) *If $\operatorname{Arg\,min} F = \emptyset$ then $|x^k| \to \infty$.*

(iv) *If $\operatorname{Arg\,min} F \neq \emptyset$ and the sequence $\{\rho_k\}$ is bounded, then the sequences $\{g_f^k\}$ and $\{g_h^k\}$ are bounded, $g_h^k + g_f^{k-1} \to 0$, $g_h^k + g_f^k \to 0$, and every accumulation point $(\hat{g}_f, \hat{g}_h)$ of $\{(g_f^k, g_h^k)\}$ satisfies the relations $\hat{g}_f \in \partial f(\hat{x})$, $\hat{g}_h \in \partial h(\hat{x})$, and $\hat{g}_f + \hat{g}_h = 0$.*

Proof. If $\operatorname{Arg\,min} F$ contains a point $\tilde{x}$, one has $F(x^k) \geq F(\tilde{x})$ for all k. Then by Lemma 4.7, $x^k \to \hat{x} \in \operatorname{Arg\,min} F$ and $F(x^k) \downarrow F(\hat{x}) = \inf F$, so (i) and (ii) hold in this case.

Suppose now that $\operatorname{Arg\,min} F = \emptyset$. If there existed $\tilde{x}$ such that $F(x^k) \geq F(\tilde{x})$ for all k, then Lemma 4.7 would imply convergence of $\{x^k\}$ to a minimizer of F, a contradiction. Therefore, for every $\tilde{x}$ we can find k such that $F(x^k) < F(\tilde{x})$. This implies that $F(x^k) \downarrow \inf F$ in this case, too; i.e., (i) is true. Moreover, if $\{x^k\}$ had a bounded subsequence, then (by the closedness of F) each of its accumulation points would minimize F, another contradiction. Therefore (iii) must be true.

As for (iv), we may use the proof of Lemma 4.7 with $\bar{x} = \hat{x}$. The sequence $g_f^k \in \partial f(z_f^k)$ is bounded and each of its accumulation points is in $\partial f(\hat{x})$, since ∂f is upper semicontinuous (cf. [Roc70, Thm. 24.4]).

Next, we have (4.6) and, by the definition of g_f^k, $g_f^k + g_h^k = -\rho_{k+1}(z_f^k - x^{k+1}) \to 0$. Thus $\{g_h^k\}$ must be bounded, too, and the required result follows from the upper semicontinuity of ∂h. □

Remark 4.9.

(i) Without boundedness of $\{\rho_k\}$ we obtain (iv) only on some subsequence, as follows from (4.8).

(ii) Inequality (4.4) may serve as a global lower bound for the objective value at iteration k. If $\operatorname{Arg\,min} F \neq \emptyset$, then, by (4.6) and (4.8), the right side of (4.4) approaches $\min F$ (at least on some subsequence even if $\{\rho_k\}$ is unbounded).

(iii) As observed by a referee, our framework may be generalized by considering other iterative methods for subproblem (1.2) that eventually satisfy descent criteria similar to (3.2)–(3.3) which ensure global convergence; cf. Remark 2.5 and [Kiw96]. Also, extensions to the case of Bregman or ϕ-divergence proximal terms can be developed along the lines of [Kiw96, Kiw97, Teb97]; we have restricted ourselves to the Euclidean norm in (1.2) for simplicity only.

5. Dual application. Let us now discuss the application of the alternating linearization method to structured problems of the form

$$\inf\nolimits_y \; \varphi(y) + \psi(My) \tag{5.1}$$

with closed proper convex functions $\varphi : \mathbb{R}^m \to (-\infty, +\infty]$, $\psi : \mathbb{R}^n \to (-\infty, +\infty]$, and an $n \times m$ matrix M. Splitting variables yields the problem

$$\inf\nolimits_{w,y} \; \varphi(y) + \psi(w), \tag{5.2a}$$

$$w - My = 0, \tag{5.2b}$$

with the Lagrangian $L(y, w, x) = \varphi(y) + \psi(w) + \langle x, My - w \rangle$, where $x \in \mathbb{R}^n$ is the vector of dual variables. The dual problem

$$\sup_x \left\{ L_D(x) := \inf_{y,w} L(y, w, x) \right\}$$

can be equivalently written as

$$\inf_x \left\{ \, F(x) = \psi^*(x) + \varphi^*(-M^T x) \, \right\}, \tag{5.3}$$

using the conjugates $\varphi^*(\cdot) = \sup_y \{\langle \cdot, y \rangle - \varphi(y)\}$, $\psi^*(\cdot) = \sup_w \{\langle \cdot, w \rangle - \psi(w)\}$ (see, e.g., [HUL93, section XII.5.4]). The dual problem (5.3) has the form (1.1) with

$$h(x) = \psi^*(x),$$

$$f(x) = \varphi^*(-M^T x).$$

Let us assume that $\varphi^* \circ (M^T)$ is finite-valued. Then both f and h are closed proper convex functions [Roc70, Thm. 12.2] and $\operatorname{dom} f = \mathbb{R}^n$. Therefore problem (5.3) satisfies all the assumptions required for applying the alternating linearization method.

The algorithm below will be shown to constitute a dual version of Algorithm 3.1.

ALGORITHM 5.1.

Step 0: Select $x^1 \in \operatorname{dom} h$ and calculate $F(x^1) = h(x^1)+f(x^1)$. Choose $z_f^0 \in \mathbb{R}^n$. Calculate

$$f(z_f^0) = -\min_y \left\{\varphi(y) + \langle z_f^0, My\rangle\right\}. \tag{5.4}$$

Choose a minimizer y^0 in the problem above. Select $\rho_1 \geq \rho_{\min} > 0$, $\kappa > 1$, $\beta_0 > 0$, $\beta_1 \in (0,1)$. Set $k = 1$.

Step 1: Find

$$w^k = \arg\min_w \ \psi(w) - \langle x^k, w\rangle + \tfrac{1}{2}\rho_k |w - My^{k-1}|^2, \tag{5.5}$$

and set

$$z_h^k = x^k - (w^k - My^{k-1})/\rho_k. \tag{5.6}$$

Step 2: Calculate

$$h(z_h^k) = \langle w^k, z_h^k\rangle - \psi(w^k), \tag{5.7}$$

$$f(z_h^k) = -\min_y \left\{ \varphi(y) + \langle z_h^k, My\rangle \right\}, \tag{5.8}$$

$$\tilde{f}_k(z_h^k) = -\left[\varphi(y^{k-1}) + \langle z_h^k, My^{k-1}\rangle\right]. \tag{5.9}$$

Set $F(z_h^k) = h(z_h^k) + f(z_h^k)$ and $\tilde{F}_k(z_h^k) = h(z_h^k) + \tilde{f}_k(z_h^k)$. Set $v_k = F(x^k) - \tilde{F}_k(z_h^k)$. If $F(z_h^k) \leq F(x^k) - \beta_1 v_k$, then set $x^{k+1} = z_h^k$; otherwise set $x^{k+1} = x^k$.

Step 3: Choose ρ_{k+1} as at Step 3 of Algorithm 3.1.

Step 4: Find

$$y^k \in \operatorname{Arg\,min}_y \ \varphi(y) + \langle x^{k+1}, My\rangle + \tfrac{1}{2}\rho_{k+1}|w^k - My|^2. \tag{5.10}$$

Step 5: Increase k by 1 and go to Step 1.

Of course, the applicability of Algorithm 5.1, and other related proximal-based dual methods, is limited to problems for which the necessary operations on conjugate functions can be implemented. Some examples are provided at the end of this section and in section 6.

The analysis of Algorithm 5.1 will be based on the following fact [Roc70, Thm. 23.5].

FACT 5.2. *For a closed proper convex function f the following conditions are equivalent*: $x^* \in \partial f(x)$, $x \in \partial f^*(x^*)$, $f(x) + f^*(x^*) = \langle x, x^*\rangle$, $x \in \operatorname{Arg\,min}\{f(\cdot) - \langle x^*, \cdot\rangle\}$.

THEOREM 5.3. *Algorithm* 5.1 *generates sequences* $\{x^k\}$, $\{y^k\}$, *and* $\{w^k\}$ *such that*

(i) *$F(x^k) \downarrow \inf F$.*
(ii) *If $\operatorname{Arg\,min} F \neq \emptyset$ then $\{x^k\}$ converges to a point $\hat{x} \in \operatorname{Arg\,min} F$.*
(iii) *If $\operatorname{Arg\,min} F = \emptyset$ then $|x^k| \to \infty$.*
(iv) *If $\operatorname{Arg\,min} F \neq \emptyset$ and the sequence $\{\rho_k\}$ is bounded, then the sequences $\{My^k\}$ and $\{w^k\}$ are bounded, $w^k - My^k \to 0$ and $w^k - My^{k-1} \to 0$. Further, each accumulation point $\hat{y}$ of $\{y^k\}$ is a solution of (5.1).*

Proof. We shall prove that Algorithm 5.1 is equivalent to Algorithm 3.1 applied to the dual problem (5.3).

First, let us note that the minimizer y^0 in (5.4) chosen at Step 0 (which exists because $\varphi^* \circ (M^T)$ is finite-valued) satisfies the relation $y^0 \in \partial\varphi^*(-M^T z_f^0)$. Therefore, by Fact 5.2, $-My^0 \in \partial f(z_f^0)$ and we can define $g_f^0 = -My^0$.

We shall use induction. Assume that for some k we have

$$y^{k-1} \in \partial\varphi^*(-M^T z_f^{k-1}) \tag{5.11}$$

and

$$g_f^{k-1} = -My^{k-1}. \tag{5.12}$$

By (5.12), problem (3.1) can be formulated as follows:

$$\min_x \psi^*(x) - \langle My^{k-1}, x\rangle + \tfrac{1}{2}\rho_k|x - x^k|^2. \tag{5.13}$$

We now show that (5.5) and (5.6) define its solution z_h^k. Indeed, the optimality condition for (5.5) yields

$$z_h^k = x^k - (w^k - My^{k-1})/\rho_k \in \partial\psi(w^k), \tag{5.14}$$

which by Fact 5.2 is equivalent to

$$w^k \in \partial\psi^*(z_h^k). \tag{5.15}$$

Using (5.6) we can rewrite the last relation as $My^{k-1} - \rho_k(z_h^k - x^k) \in \partial\psi^*(z_h^k)$, which is necessary and sufficient for the optimality of z_h^k in (5.13). From (5.15), using Fact 5.2, we obtain $\psi^*(z_h^k) = \langle w^k, z_h^k\rangle - \psi(w^k)$, which validates (5.7). Relation (5.8) follows directly from the definition. Next, (5.11) and Fact 5.2 yield

$$f(z_f^{k-1}) = \varphi^*(-M^T z_f^{k-1}) = -\varphi(y^{k-1}) - \langle M^T z_f^{k-1}, y^{k-1}\rangle.$$

Combining this relation with (5.12), we obtain

$$\begin{aligned}\tilde{f}_k(z_h^k) &= f(z_f^{k-1}) + \langle g_f^{k-1}, z_h^k - z_f^{k-1}\rangle \\ &= -\varphi(y^{k-1}) - \langle M^T z_f^{k-1}, y^{k-1}\rangle - \langle My^{k-1}, z_h^k - z_f^{k-1}\rangle,\end{aligned}$$

which is equivalent to (5.9). The remaining parts of Steps 2 and 3 are identical to those in Algorithm 3.1.

By direct calculation, using (5.12) and (5.6), we obtain

$$g_h^k = -g_f^{k-1} - \rho_k(z_h^k - x^k) = w^k. \tag{5.16}$$

Therefore, problem (3.4) can be written as

$$\min_x \langle w^k, x\rangle + \varphi^*(-M^T x) + \tfrac{1}{2}\rho_{k+1}|x - x^{k+1}|^2. \tag{5.17}$$

We now show that the point z_f^k, the solution of (5.17), has the form

$$z_f^k = x^{k+1} - (w^k - My^k)/\rho_{k+1}, \tag{5.18}$$

where y^k is given by (5.10). Indeed, the optimality condition for (5.10) reads

$$-M^T z_f^k = -M^T x^{k+1} + M^T(w^k - My^k)/\rho_{k+1} \in \partial\varphi(y^k), \tag{5.19}$$

which by Fact 5.2 is equivalent to the relation $y^k \in \partial\varphi^*(-M^T z_f^k)$; i.e., (5.11) holds for k. The last relation is equivalent to $-My^k \in \partial f(z_f^k)$ (Fact 5.2). Substitution of My^k from (5.18) yields the optimality condition for (5.17): $-w^k - \rho_{k+1}(z_f^k - x^{k+1}) \in \partial f(z_f^k)$. Finally, from (5.16) and (5.18) we get

$$g_f^k = -g_h^k - \rho_{k+1}(z_f^k - x^{k+1}) = -My^k, \tag{5.20}$$

which proves (5.12) for k and completes the induction.

Therefore, assertions (i)–(iii) follow from those of Theorem 4.8. To show (iv), observe that from (5.16) and (5.20), by Theorem 4.8(iv), the sequences $\{My^k\}$ and $\{w^k\}$ are bounded,

$$w^k - My^k \to 0, \tag{5.21}$$

and $w^k - My^{k-1} \to 0$. To complete the proof of (iv), let $(w^k, y^k) \to (\hat{w}, \hat{y})$, $k \in \mathcal{K}$. Taking limits in (5.14) and (5.19), we obtain $\hat{x} \in \partial\psi(\hat{w})$, $-M^T\hat{x} \in \partial\varphi(\hat{y})$, and, by (5.21), $\hat{w} - M\hat{y} = 0$. This proves the optimality of $(\hat{w}, \hat{y})$ in (5.2). □

As mentioned in sections 1 and 2, the alternating linearization method fits the framework of inexact proximal point algorithms and bears some resemblance to the operator splitting methods. Therefore it is not surprising that its dual version, Algorithm 5.1, is intimately related to augmented Lagrangian methods and alternating direction methods of multipliers [BeT89, DLMK$^+$94, EcB92, EcF94, Fuk92, Gab83, KDLM96].

Specifically, consider the augmented Lagrangian for (5.2):

$$\Lambda_\rho(y, w, x) = \varphi(y) + \psi(w) - \langle x, w - My\rangle + \tfrac{1}{2}\rho|w - My|^2, \tag{5.22}$$

where $x \in \mathbb{R}^n$ is the vector of multipliers and $\rho > 0$ is a penalty coefficient. Assuming that in Algorithm 5.1 the points x^k remain fixed at x and the penalty coefficients ρ_k fixed at ρ, we see that (5.5) and (5.10) implement the Gauss–Seidel (blockwise minimization) method for minimizing the augmented Lagrangian (5.22). Note, however, that in the alternating direction method the multipliers are updated after each blockwise minimization iteration. In Algorithm 5.1, the classical update (cf. (5.6))

$$x^{k+1} = x^k - (w^k - My^{k-1})/\rho_k$$

takes place only under the descent conditions of Step 2. Moreover, the penalty coefficient is allowed to change within the Gauss–Seidel loop as well as after the multiplier update.

Example 5.4. Consider the problem

$$\min_y \varphi(y) + \sum_{j=1}^N \psi_j(y),$$

with closed proper convex functions $\varphi : \mathbb{R}^m \to (-\infty, +\infty)$ and $\psi_j : \mathbb{R}^m \to (-\infty, +\infty]$, $j = 1, \dots, N$. This is a special case of (5.1) with $My = (y, y, \dots, y)$, $\psi(w) = \sum_{j=1}^N \psi_j(w_j)$, and $n = Nm$. The key operations of Algorithm 5.1 can be substantially simplified in this case. With $x = (x_1, \dots, x_N) \in \mathbb{R}^{Nm}$ problem (5.5), solved at Step 1, decomposes into parallel subproblems for $j = 1, \dots, N$:

$$w_j^k = \arg\min_{w_j} \psi_j(w_j) - \langle x_j^k, w_j \rangle + \frac{1}{2\rho_k} |w_j - y^{k-1}|^2,$$

$$(z_h^k)_j = x_j^k - (w_j^k - y^{k-1})/\rho_k,$$

while (5.10) takes the form

$$y^k = \arg\min_y \varphi(y) + \left\langle \sum_{j=1}^N x_j^{k+1}, y \right\rangle + \frac{1}{2\rho_{k+1}} \sum_{j=1}^N |w_j^k - y|^2.$$

We recognize some similarities with the algorithms of [FHN$^+$96, HaL88, MNS91, Tse91], but our approach has different rules for updating the multipliers and a variable penalty coefficient.

6. Applications to stochastic programming. We now consider an important class of optimization models known as multistage stochastic programming problems.

We use the modeling methodology developed in [RoW91] (see also [ChR95, MuR95, Rob91]). The basic object in the model is the *scenario tree*, whose levels $1, \dots, T$ (counted from the root to the leaves) correspond to time stages and each path from the root to a leaf (scenario) has exactly T nodes. With each scenario path j ($j = 1, \dots, N$) the following objects are associated: the decision subvector

$$w_j = (w_j(1), \dots, w_j(T)) \in \mathbb{R}^{q_1} \times \dots \times \mathbb{R}^{q_T},$$

the closed convex cost function $\psi_j : \mathbb{R}^{q_1} \times \dots \times \mathbb{R}^{q_T} \to (-\infty, +\infty]$, and the probability p_j. The entire decision vector $w = (w_1, \dots, w_N) \in \mathbb{R}^{qN}$, where $q = q_1 + \dots + q_T$, must satisfy the *nonanticipativity* constraint: for all $t = 1, \dots, T-1$ and for all pairs (i, j) of scenarios (paths) with identical first t nodes, one must have

$$w_i(\tau) - w_j(\tau) = 0, \qquad \tau = 1, \dots, t.$$

All these constraints (or a sufficient subset of them) can be put into one linear equation $Aw = \sum_{j=1}^N A_j w_j = 0$, where $A = [A_1 \cdots A_N]$ has dimension $m_A \times qN$. The entire problem can be formulated as follows:

$$\min \sum_{j=1}^N p_j \psi_j(w_j), \tag{6.1a}$$

$$\text{s.t.} \sum_{j=1}^N A_j w_j = 0. \tag{6.1b}$$

6.1. Augmented Lagrangian decomposition. Consider the augmented Lagrangian for (6.1):

$$\Lambda(w,\lambda) = \sum_{j=1}^{N} p_j\psi_j(w_j) + \left\langle \lambda, \sum_{j=1}^{N} A_j w_j \right\rangle + \frac{1}{2}\rho \left| \sum_{j=1}^{N} A_j w_j \right|^2, \tag{6.2}$$

where $\lambda \in \mathbb{R}^{m_A}$ and $\rho > 0$ is a penalty parameter. A solution of (6.1) can be obtained by the following method of multipliers (cf. [Ber82, Hes69, Pow69, Roc76a]).

Algorithm 6.1.

Step 0: Choose $\lambda^1 \in \mathbb{R}^{m_A}$. Set $l = 1$.

Step 1: Find $w^l \in \operatorname{Arg\,min}_w \Lambda(w, \lambda^l)$.

Step 2: Set $\lambda^{l+1} = \lambda^l + \rho A w^l$, increase l by 1, and go to Step 1. It remains to determine an efficient method for minimizing (6.2). In fact, the alternating linearization algorithm is a good candidate. To see this, note that the problem in question is nearly identical to that presented in Example 1.1. In particular, we have

$$h(w) = \sum_{j=1}^{N} \{ p_j \psi_j(w_j) + \langle \lambda, A_j w_j \rangle \}$$

and

$$f(w) = \tfrac{1}{2}\rho\, |Aw|^2 .$$

The functions h and f have all the properties required by the alternating linearization algorithm. The separability of h means that Step 1 of Algorithm 3.1 can be decomposed into parallel subproblems for $j = 1, \ldots, N$,

$$z_{h,j}^k = \arg\min_{w_j}\, p_j\psi_j(w_j) + \langle \lambda + \rho A z_f^k, A_j w_j \rangle + \tfrac{1}{2}\rho_k |w_j - w_j^k|^2,$$

whereas Step 4 requires solving the least squares problem:

$$z_f^k = \arg\min_{w}\, \langle g_h^k, w \rangle + \tfrac{1}{2}\rho |Aw|^2 + \tfrac{1}{2}\rho_{k+1} \sum_{j=1}^{N} |w_j - w_j^{k+1}|^2.$$

Algorithm 6.1 effectively comprises three layers of methods: the method of multipliers at the top level, the alternating linearization method at the middle layer, and the subproblem solver at the bottom layer. In the next section we show that the dual approach allows us to remove one of these layers and to develop a more efficient method.

6.2. Dual strategy. All nonanticipative vectors $w = (w_1, \ldots, w_N)$ form a linear subspace $\mathcal{L}$ of $\mathbb{R}^{qN}$. The orthogonal projection on $\mathcal{L}$ will be denoted $\Pi_{\mathcal{L}}$. Given w, its projection $u = \Pi_{\mathcal{L}} w$ can be calculated as follows (see [RoW91]). For every $j = 1, \ldots, N$ and $t = 1, \ldots, T$, we find the set of scenarios indistinguishable from scenario j until stage t,

$$I_j(t) = \{ i : \nu_\tau(i) = \nu_\tau(j),\ \tau = 1, \ldots, t \},$$

and we average $w_i(t)$ over this subset:

$$u_j(t) = \frac{1}{|I_j(t)|} \sum_{i \in I_j(t)} w_i(t).$$

Using the indicator function $\delta_{\mathcal{L}}$ of $\mathcal{L}$ we can formulate (6.1) equivalently as

$$\min_w \delta_{\mathcal{L}}(w) + \sum_{j=1}^{N} p_j \psi_j(w_j). \tag{6.3}$$

Let r majorize the Euclidean norm of a solution to (6.1) and let $\mathcal{B} = \{y \in \mathbb{R}^{qN} : |y| \le r\}$. With

$$\varphi(w) = \delta_{\mathcal{L}\cap\mathcal{B}}(w)$$

and

$$\psi(w) = \sum_{j=1}^{N} p_j \psi_j(w_j)$$

we can regard problem (6.3) as an instance of (5.1), where $M = I$ (the identity). The purpose of introducing $\mathcal{B}$ was to make φ^* finite-valued. For $x = (x_1, \dots, x_N) \in \mathbb{R}^{qN}$, we have

$$h(x) = -\sum_{j=1}^{N} \inf_{w_j} \{ \, p_j \psi_j(w_j) - \langle x_j, w_j \rangle \, \} , \tag{6.4a}$$

$$f(x) = \max_y \{ \, \langle -x, y \rangle : |y| \le r, \; y \in \mathcal{L} \, \} = r|\Pi_{\mathcal{L}} x|, \tag{6.4b}$$

and the entire algorithm simplifies as follows.

ALGORITHM 6.2.

Step 0: Select $x^1 \in \mathbb{R}^{qN}$ and calculate $F(x^1) = h(x^1) + f(x^1)$, using (6.4). Choose $z_f^0 \in \mathbb{R}^{qN}$. Calculate $f(z_f^0) = r|\Pi_{\mathcal{L}} z_f^0|$ and $y^0 = -r\Pi_{\mathcal{L}} z_f^0 / |\Pi_{\mathcal{L}} z_f^0|$ ($y^0 = 0$ if $z_f^0 \perp \mathcal{L}$). Choose $\rho_1 \ge \rho_{\min} > 0$, $\kappa > 1$, $\beta_0 > 0$, $\beta_1 \in (0, 1)$. Set $k = 1$.

Step 1: For scenarios $j = 1, \dots, N$, calculate

$$w_j^k = \arg\min_{w_j} \, p_j \psi_j(w_j) - \langle x_j^k, w_j \rangle + \frac{1}{2\rho_k} |w_j - y_j^{k-1}|^2$$

and set $z_h^k = x^k - (w^k - y^{k-1})/\rho_k$.

Step 2: Calculate

$$h(z_h^k) = \sum_{j=1}^{N} \left\{ \, \langle w_j^k, (z_h^k)_j \rangle - p_j \psi_j(w_j^k) \, \right\} ,$$

$$f(z_h^k) = r|\Pi_{\mathcal{L}} z_h^k|,$$

$$\tilde{f}_k(z_h^k) = -\langle z_h^k, y^{k-1} \rangle.$$

Set $F(z_h^k) = h(z_h^k) + f(z_h^k)$ and $\tilde{F}_k(z_h^k) = h(z_h^k) + \tilde{f}_k(z_h^k)$. Set $v_k = F(x^k) - \tilde{F}_k(z_h^k)$. If $F(z_h^k) \le F(x^k) - \beta_1 v_k$, then set $x^{k+1} = z_h^k$; otherwise set $x^{k+1} = x^k$.

Step 3: Choose ρ_{k+1} as at Step 3 of Algorithm 3.1.

Step 4: Calculate y^k as the orthogonal projection of $\tilde{y}^k = \Pi_{\mathcal{L}}(w^k - \rho_{k+1}x^{k+1})$ on the ball $\{y : |y| \le r\}$.

Step 5: Increase k by 1 and go to Step 1.

To justify Step 4 of Algorithm 6.2 we note that

$$\begin{aligned}
&\arg\min_y \left\{ \varphi(y) + \langle x^{k+1}, y\rangle + \frac{1}{2\rho_{k+1}}|w^k - y|^2 \right\} \\
&\qquad = \arg\min_y \left\{ \langle x^{k+1}, y\rangle + \frac{1}{2\rho_{k+1}}|w^k - y|^2 : |y| \le r,\ y \in \mathcal{L} \right\} \\
&\qquad = \arg\min_y \left\{ |w^k - \rho_{k+1}x^{k+1} - y|^2 : |y| \le r,\ y \in \mathcal{L} \right\}.
\end{aligned}$$

Algorithm 6.2 bears some similarities to the scenario aggregation method of [RoW91], which is a version of the alternating direction method of multipliers (see also [Spi85, ChT94]). There are differences, though, in the way the multipliers x^k are updated and in the variable penalty coefficient. It is worth noting that the descent test in the dual space (Step 2) does not require much work, because the values of $F = h + f$ are easily available. Compared to the augmented Lagrangian decomposition of section 6.1, we have only two layers of algorithms here: the alternating linearization method in the dual space and the subproblem solver.

7. Numerical illustration. We consider a multistage stochastic macroeconomic energy model described in detail in [Ros94]. The model has the form (6.1) with $N = 8$, $n = 610$, and $m_A = 3240$. Each function ψ_j has a simple analytic form, but its domain is defined by 398 constraints, of which 25 are nonlinear (with 85 nonlinear variables). Thus, of 4880 variables in the entire model, 680 are nonlinear variables. The scenario model was formulated in GAMS [BKM92], and MINOS [MuS82] was used to solve scenario subproblems (with default parameters).

TABLE 7.1
Results for the augmented Lagrangian decomposition method.

Outer iteration (l)	Alternating steps (k)	Descent steps	Null steps	$\frac{v^k}{1+\vert F(x^k)\vert}$	$\vert Aw^l\vert^2/2$
1	10	6	4	1.9E-3	1284
2	431	256	175	7.9E-7	1.429
3	24	11	13	4.5E-7	0.276
4	11	5	6	2.0E-7	0.133
5	13	9	4	1.6E-7	0.104
6	107	76	31	1.2E-7	0.076
7	1	1	0	1.2E-7	0.049

7.1. Augmented Lagrangian decomposition. Algorithm 6.1 was run with $\rho = 1$ and $\lambda^1 = 0$. At Step 1 we used Algorithm 3.1 with the following parameters: $\kappa = 2$, $\beta_0 = 1$, $\beta_1 = 0.1$, $\rho_1 = \rho$, $\rho_{\min} = \rho/1000$. It started from $x^1 = \arg\min_x\{h(x) + \frac{1}{2}|x|^2\}$ at $l = 1$ and from w^{l-1} otherwise and terminated when $\max\{v_k, |z_h^k - x^k|^2/2\} \le 0.1|Aw^{l-1}|^2/2$ (with $w^0 = x^1$).

Seven major iterations of Algorithm 6.1 were made; the accuracy of the final solution was comparable with that obtained by other methods [RoR96, Rus95]. Table 7.1 illustrates our results. Each row of the table corresponds to one iteration of the multiplier method (outer loop). Columns 2–5 provide information about the performance of the alternating linearization method (inner loop). The relative accuracy of

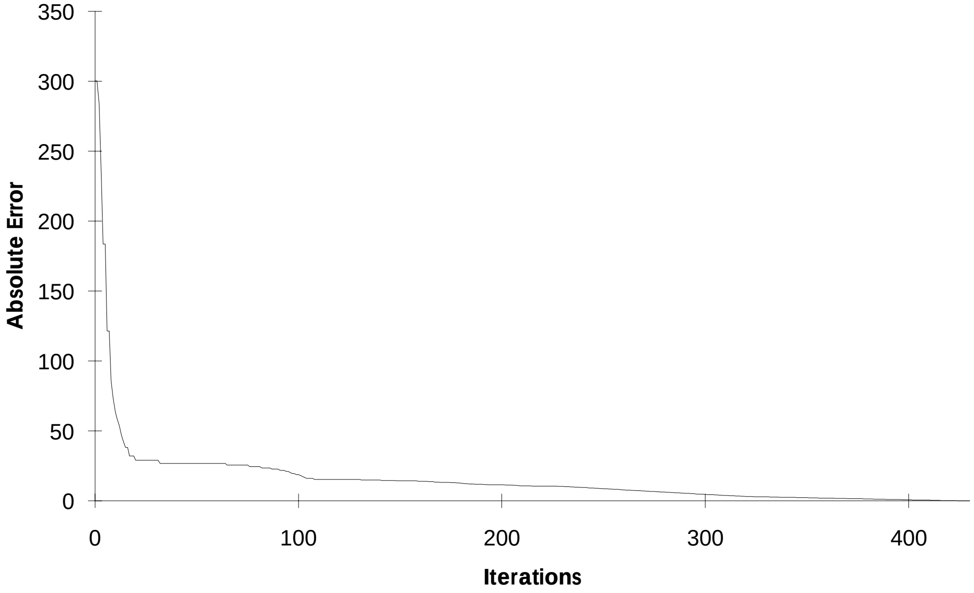


Fig. 7.1. *Absolute error in the objective value: major iteration* 2.

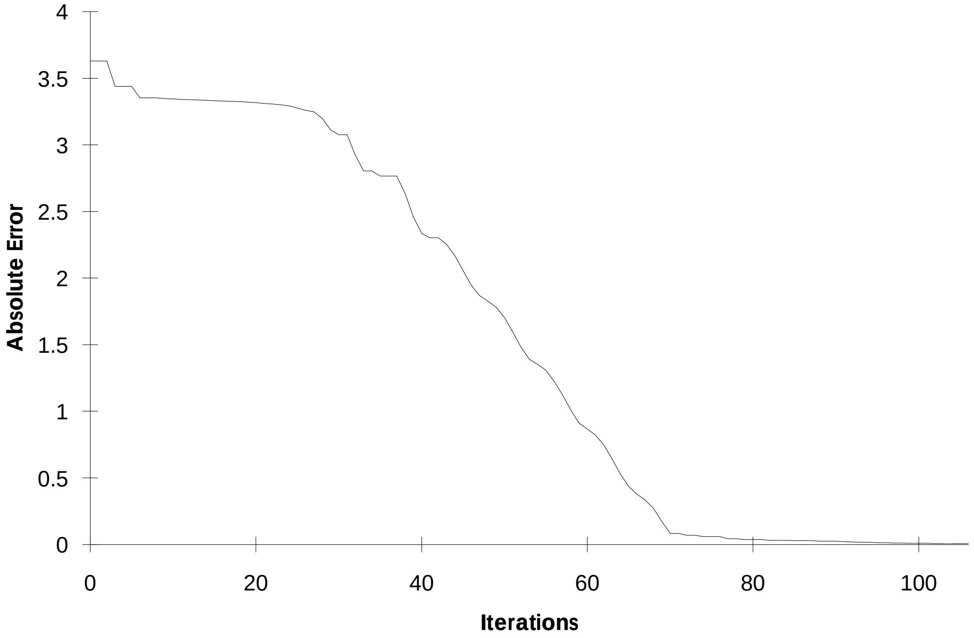


Fig. 7.2. *Absolute error in the objective value: major iteration* 6.

minimization in the inner loop was estimated by $v_k/(1+|F(x^k)|)$. The last column gives the error in the nonanticipativity constraints.

The progress of the alternating linearization method at major iterations 2 and 6

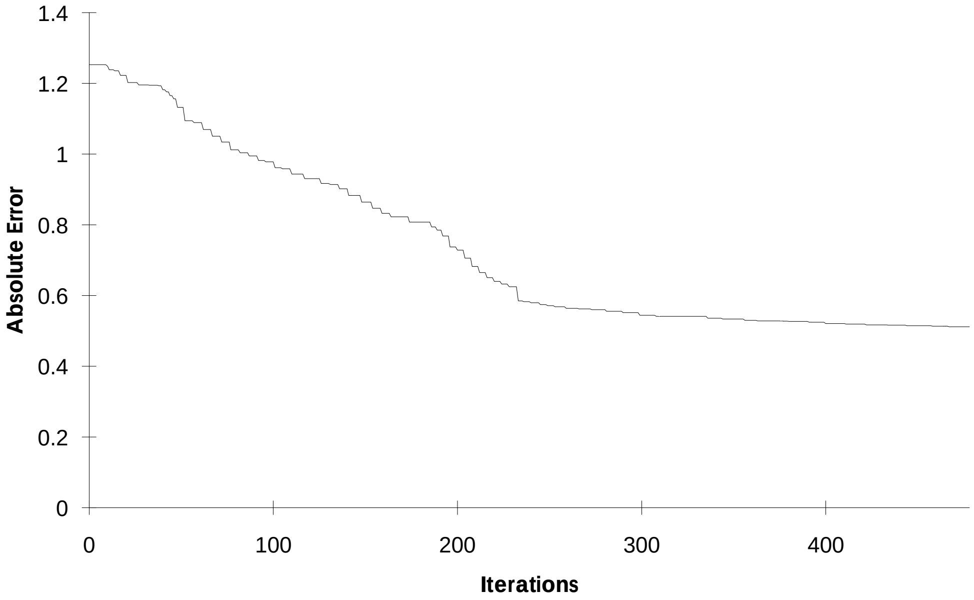


Fig. 7.3. *Dual method: absolute error in the objective value.*

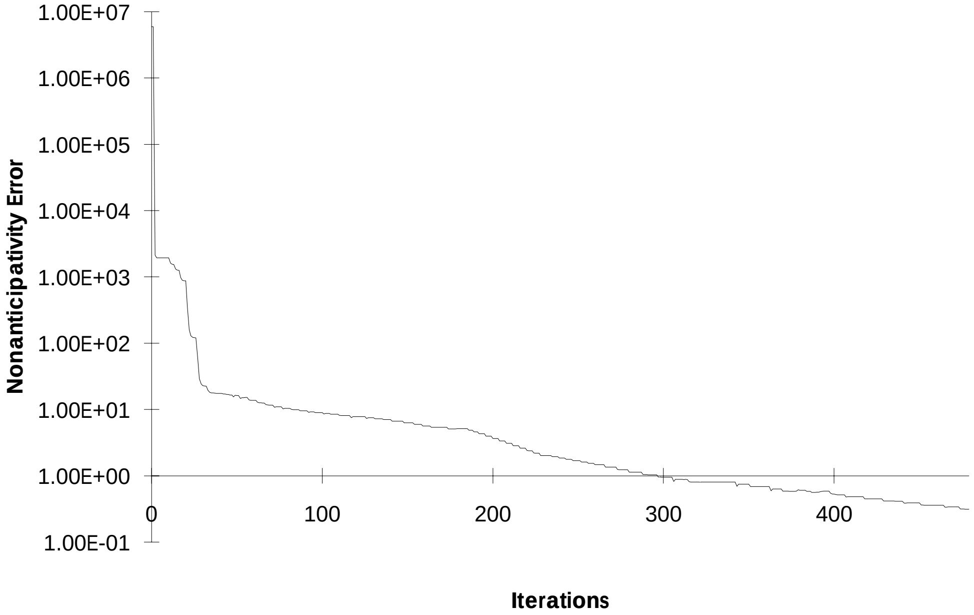


Fig. 7.4. *Dual method: nonanticipativity.*

is illustrated in Figures 7.1 and 7.2. The label "Iterations" refers to alternating steps (the inner loop).

The absolute error in the objective value was calculated as $F(x^k) - F(x^{k_*}) + v_{k_*}$, where k and k_* are the current and final iterations (alternating steps) of Algorithm 3.1, respectively. We see that the algorithm can attain relatively high accuracy.

7.2. Dual strategy. We chose $r = 3 \times 10^3$ large enough to majorize the solution obtained by other methods, so f (which may be interpreted as an exact penalty function) had rather steep walls. Accordingly, in Algorithm 6.2 we used a larger value of $\rho_1 = 10^6$. The other parameters were the same as in section 7.1. The starting point was $x^1 = 0$.

Figure 7.3 illustrates the progress of the method in terms of the absolute error in the dual objective—$F(x^k) - F_{\min}$ (where $F_{\min}$ is the known optimal value)—and Figure 7.4 shows the decrease in the measure of nonanticipativity of the current solution—$\frac{1}{2}|w^k - y^{k-1}|^2$. Again, we see that the method converges quickly at the initial stage, although the speed of convergence at the tail is not high, because of the essential nonsmoothness of f.

Summing up, this preliminary numerical experience indicates that the alternating linearization method, both in the primal and in the dual form, has the potential to become a useful tool for large scale nonsmooth optimization.

Acknowledgments. We wish to thank the associate editor and an anonymous referee for their comments, which allowed us to improve the paper.

REFERENCES

[Aus86] A. Auslender, *Numerical methods for nondifferentiable convex optimization*, Math. Programming Stud., 30 (1986), pp. 102–126.

[Ber82] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[BeT89] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[BKM92] A. Brooke, D. Kendrick, and A. Meeraus, *GAMS: A User's Guide*, Scientific Press, San Francisco, 1992.

[ChR95] B. J. Chun and S. M. Robinson, *Scenario analysis via bundle decomposition*, Ann. Oper. Res., 56 (1995), pp. 39–63.

[ChT94] G. Chen and M. Teboulle, *A proximal-based decomposition method for convex minimization problems*, Math. Programming, 64 (1994), pp. 81–101.

[CoL93] R. Correa and C. Lemaréchal, *Convergence of some algorithms for convex minimization*, Math. Programming, 62 (1993), pp. 261–275.

[DLMK$^+$94] R. De Leone, R. R. Meyer, and S. Kontogiorgis, Z. Zakarian and G. Zakeri, *Coordination in coarse-grained decomposition*, SIAM J. Optim., 4 (1994), pp. 777–793.

[EcB92] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Programming, 55 (1992), pp. 293–318.

[EcF94] J. Eckstein and M. Fukushima, *Some reformulations and applications of the alternating direction method of multipliers*, in Large Scale Optimization: State of the Art, W. W. Hager, D. W. Hearn, and P. M. Pardalos, eds., Kluwer, Dordrecht, the Netherlands, 1994, pp. 115–134.

[EcF97] J. Eckstein and M. C. Ferris, *Operator-splitting methods for monotone affine variational inequalities, with a parallel application to optimal control*, INFORMS J. Comput., 10 (1998), pp. 218–235.

[Eck94] J. Eckstein, *Some saddle-function splitting methods for convex programming*, Optim. Methods Softw., 4 (1994), pp. 75–83.

[FHN$^+$96] M. Fukushima, M. Haddou, V. H. Nguyen, J.-J. Strodiot, and E. Yamakawa, *A parallel descent algorithm for convex programming*, Comput. Optim. Appl., 5 (1996), pp. 5–37.

[Fuk92] M. Fukushima, *Application of the alternating direction method of multipliers to separable convex programming problems*, Comput. Optim. Appl., 1 (1992), pp. 93–111.

[Gab83] D. Gabay, *Applications of the method of multipliers to variational inequalities*, in Augmented Lagrangian Methods: Applications to the Solution of Boundary-

Value Problems, M. Fortin and R. Glowinski, eds., North-Holland, Amsterdam, 1983, pp. 299–331.

[GoT89] E. G. Golshtein and N. V. Tretyakov, *Modified Lagrange Functions; Theory and Optimization Methods*, Nauka, Moscow, 1989 (in Russian).

[Gül91] O. Güler, *On the convergence of the proximal point algorithm for convex minimization*, SIAM J. Control Optim., 29 (1991), pp. 403–419.

[HaL88] S.-P. Han and G. Lou, *A parallel algorithm for a class of convex programs*, SIAM J. Control Optim., 26 (1988), pp. 345–355.

[Hes69] M. R. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appl. 4 (1969), pp. 303–320.

[HUL93] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.

[KDLM96] S. Kontogiorgis, R. De Leone, and R. R. Meyer, *Alternating direction splittings for block angular parallel optimization*, J. Optim. Theory Appl., 90 (1996), pp. 1–29.

[Kiw96] K. C. Kiwiel, *A Bundle Bregman Proximal Method for Convex Nondifferentiable Minimization*, Tech. Report, Systems Research Institute, Warsaw, June 1996. Revised September 1997.

[Kiw97] K. C. Kiwiel, *Proximal minimization methods with generalized Bregman functions*, SIAM J. Control Optim., 35 (1997), pp. 1142–1168.

[Lem89] B. Lemaire, *The proximal algorithm*, in New Methods in Optimization and Their Industrial Uses, J. P. Penot, ed., International Series of Numerical Mathematics 87, Birkhäuser, Basel, 1989, pp. 73–87.

[Mar70] B. Martinet, *Régularisation d'inéquations variationelles par approximations successives*, RAIRO Rech. Opér., 4(R3) (1970), pp. 154–158.

[MaT93] P. Mahey and P.-D. Tao, *Partial regularization of the sum of two maximal monotone operators*, RAIRO Modél. Math. Anal. Numér. **27** (1993) 375–392.

[MNS91] K. Mouallif, V. H. Nguyen, and J.-J. Strodiot, *A perturbed parallel decompositon method for a class of nonsmooth convex minimization problems*, SIAM J. Control Optim., 29 (1991), pp. 829–847.

[MOT95] P. Mahey, S. Oualibouch, and P.-D. Tao, *Proximal decomposition on the graph of a maximal monotone operator*, SIAM J. Optim. **5** (1995) 454–466.

[MuR95] J. M. Mulvey and A. Ruszczyński, *A new scenario decomposition method for large-scale stochastic optimization*, Oper. Res., 43 (1995), pp. 477–490.

[MuS82] B. A. Murtagh and M. A. Saunders, *A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints*, Math. Programming Stud., 16 (1982), pp. 84–117.

[Pow69] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, London, 1969, pp. 283–298.

[Rob91] S. M. Robinson, *Extended scenario analysis*, Ann. Oper. Res., 31 (1991), pp. 385–398.

[Roc70] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[Roc76a] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.

[Roc76b] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[RoR96] C. H. Rosa and A. Ruszczyński, *On augmented Lagrangian decomposition methods for multistage stochastic programs*, Ann. Oper. Res., 64 (1996), pp. 289–309.

[Ros94] C. H. Rosa, *Pathways of Economic Development in an Uncertain Environment: A Finite Scenario Approach to the U.S. Region under Carbon Emission Restrictions*, WP-94-41, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1994.

[RoW91] R. T. Rockafellar and R. J.-B. Wets, *Scenarios and policy aggregation in optimization under uncertainty*, Math. Oper. Res., 16 (1991), pp. 1–23.

[Rus95] A. Ruszczyński, *On convergence of an augmented Lagrangian decomposition method for sparse convex optimization*, Math. Oper. Res., 20 (1995), pp. 634–656.

[Spi85] J. E. Spingarn, *Applications of the method of partial inverses to convex programming: Decomposition*, Math. Programming, 32 (1985), pp. 199–223.

[Teb97] M. Teboulle, *Convergence of proximal-like algorithms*, SIAM J. Optim., 7 (1997), pp. 1069–1083.

[Tse90] P. Tseng, *Further applications of a splitting algorithm to decomposition in varia-*

tional inequalities and convex programming, Math. Programming, 48 (1990), pp. 249–263.

[Tse91] P. Tseng, *Applications of a splitting algorithm to decomposition in convex programming and variational inequalities*, SIAM J. Control Optim., 29 (1991), pp. 119–138.

[Tse97] P. Tseng, *Alternating projection-proximal methods for convex programming and variational inequalities*, SIAM J. Optim., 7 (1997), pp. 951–965.

SIAM J. OPTIM.
Vol. 9, No. 3, pp. 690–706

NONSMOOTH CONSTRAINED OPTIMIZATION AND MULTIDIRECTIONAL MEAN VALUE INEQUALITIES[*]

DIDIER AUSSEL[†], JEAN-NOËL CORVELLEC[†], AND MARC LASSONDE[‡]

Abstract. We establish a general Fermat rule for the problem of minimizing a lower semicontinuous function on a convex subset of a Banach space. Our basic tool is a constrained variational principle derived from the "smooth" variational principle of Borwein and Preiss. Specializing the Fermat rule to the case when the convex set is a "drop," we obtain a multidirectional Rolle-type inequality from which, in turn, we deduce a multidirectional mean value inequality, in the line of Clarke and Ledyaev. We follow the abstract approach of our previous paper [*Trans. Amer. Math. Soc.*, 347 (1995), pp. 4147–4161], thus covering all standard situations met in applications, while stressing the links between the results and the few key properties that are needed.

Key words. subdifferential, smooth norms, constrained variational principle, Fermat rule, optimality condition, multidirectional mean value inequality

AMS subject classifications. Primary 49J52; Secondary 49J45, 49K27

PII. S105262349732339X

1. Introduction.

Consider the constrained optimization problem

$$(\mathcal{P}) \qquad \text{minimize} \quad f(x) \quad \text{subject to } x \in C,$$

where C is a nonempty closed convex subset of a Banach space X, and $f : X \to \mathbb{R} \cup \{+\infty\}$ is lower semicontinuous, bounded below on C, and finite at some point of C. In case the function f is "smooth," the well-known Fermat rule provides a necessary condition for a point $\bar{x} \in C$ to be a solution to $(\mathcal{P})$. Here, by "smooth," we understand that either f is differentiable at the potential solution $\bar{x}$ or f is convex lower semicontinuous and satisfies a so-called constraint qualification. What can be said in the most general situations where $(\mathcal{P})$ may have no solution and f is not "smooth"?

We study this problem in the first part of the paper. We begin with a constrained version of the smooth variational principle, from which we derive a necessary condition for a sequence to be minimizing for $(\mathcal{P})$. Roughly speaking, we establish that any minimizing sequence for $(\mathcal{P})$ is close to another minimizing sequence, the points of which satisfy an approximate Fermat rule. When $\bar{x}$ is a solution of $(\mathcal{P})$ and f is either differentiable at $\bar{x}$ or convex qualified, we recover the classical results.

In the rest of the paper, we assume that the constraint set is a "drop." In that case, the Fermat rule can be refined to yield first a multidirectional Rolle-type inequality, and next, with some extra work, a multidirectional mean value inequality in the line of Clarke and Ledyaev. This result, in turn, provides interesting complements to our bootstrapping Fermat rule.

The case where the constraint set is compact deserves special attention, since it is both easier and important for applications. We find it convenient to give a self-

[*] Received by the editors June 13, 1997; accepted for publication (in revised form) July 15, 1998; published electronically May 21, 1999.

http://www.siam.org/journals/siopt/9-3/32339.html

[†] Département de Mathématiques, Université de Perpignan, 66860 Perpignan Cedex, France (aussel@univ-perp.fr, corvellec@univ-perp.fr).

[‡] Département de Mathématiques, Université des Antilles et de la Guyane, 97159 Pointe-à-Pitre Cedex, Guadeloupe, France (lassonde@univ-ag.fr).

contained treatment of this case in an appendix. This can be read independently from the rest; it provides a clear overview of the interconnections between the results.

This paper is a sequel to [3], where a mean value inequality and subdifferential criteria were established using an abstract notion of subdifferential. Here we use the same approach to express the aforementioned results in terms of our abstract subdifferential, thus pointing out the only properties pertinent to the theory. Examples of settings covered by our approach are specified at appropriate places in the text.

After this paper was submitted for publication, we were informed of a closely related work of Zhu [18], where the multidirectional mean value inequality is established for the subdifferentials $\partial^{\#}$ (see section 2 for the notation) in spaces with a $\#$-differentiable norm.

2. Subdifferential and associated smooth concepts. Throughout this paper, X stands for a real Banach space, X^* for its topological dual, and $\langle \cdot, \cdot \rangle$ for the duality pairing. Let C be a nonempty convex subset of X. For $x \in X$ and $\| \cdot \|$ a norm on X, we set

$$d_C(x) := \inf_{c \in C} \|x - c\| \, ,$$

and for $\delta \geq 0$, we let $B_\delta(C) = \{x \in X \mid d_C(x) \leq \delta\}$.

All the functions $f : X \to \mathbb{R} \cup \{+\infty\}$ considered are lower semicontinuous. As usual, we set $\operatorname{dom} f = \{x \in X \mid f(x) < \infty\}$ and we write $x_n \to_f x$ to express that the sequence $(x_n, f(x_n)) \subset \operatorname{dom} f \times \mathbb{R}$ converges to $(x, f(x))$. For a set-valued operator $A : X \to X^*$, we let $\operatorname{dom} A = \{x \in X \mid A(x) \neq \emptyset\}$.

We recall from [3] the abstract notions of *subdifferential operator* ∂ and of corresponding *∂-smoothness of a norm.*

DEFINITION 2.1. *We call* subdifferential operator*, denoted by ∂ any operator that associates a subset $\partial f(x)$ of X^* to any lower semicontinuous $f : X \to \mathbb{R} \cup \{+\infty\}$, any space X, and any $x \in X$ and that satisfies the following properties:*

(P1) $\partial f(x) = \{x^* \in X^* \mid \langle x^*, y - x\rangle + f(x) \leq f(y) \quad \forall y \in X\}$ *whenever f is convex;*

(P2) $0 \in \partial f(x)$ *whenever f attains a local minimum at $x \in \operatorname{dom} f$;*

(P3) $\partial(f + g)(x) \subset \partial f(x) + \partial g(x)$ *whenever g is real-valued, convex, continuous, and* ∂-differentiable *at x;*

where g ∂-differentiable *at x means that both $\partial g(x)$ and $\partial(-g)(x)$ are nonempty.*

Remark 2.1. The consideration of abstract subdifferentials was initiated by Ioffe [12] in a different context; see also Correa, Jofré, and Thibault [7], Thibault and Zagrodny [16], and Ioffe and Penot [13]. Our definition, however, is less restrictive.

DEFINITION 2.2. *A norm $\|.\|$ on X is said to be* ∂-smooth *if the functions of the following form are ∂-differentiable:*

$$x \mapsto \Delta_2(x) := \sum_n \mu_n \|x - v_n\|^2,$$

where $\mu_n \geq 0$, the series $\sum_n \mu_n$ is convergent, and the sequence (v_n) converges in X.

Remark 2.2. In the definition of a ∂-smooth norm as given in [3, *Definition* 2.2], it is required that only the functions Δ_2 with $\sum_n \mu_n = 1$ be ∂-differentiable. Allowing $\sum_n \mu_n$ to take any nonnegative value is necessary for the proof of [3, *Theorem* 3.1], while it does not affect the rest of the paper. It is also required in [3, *Definition* 2.2] that for any compact segment $[a, b] \subset X$, the function $x \mapsto d^2_{[a,b]}(x)$ be ∂-differentiable. It turns out, however, that this property is automatically fulfilled as soon as the above condition on the functions Δ_2 holds; see Proposition 2.3 below.

As was already observed in [3], the $\partial^{\#}$ subdifferentials considered by Borwein and Preiss [4], the Ioffe approximate subdifferential ∂^I (see, e.g., Ioffe and Penot [13]), and the Clarke–Rockafellar subdifferential ∂^{CR} (see, e.g., Clarke [5]), among others, satisfy properties (P1), (P2), and (P3) of Definition 2.1. We recall that a function f is ∂-differentiable at x for a subdifferential $\partial = \partial^{\#}$ if and only if $\partial f(x) = -\partial(-f)(x)$ contains a single element, which is the usual Gâteaux derivative $\nabla^G f(x)$, while a function is ∂^I- or ∂^{CR}-differentiable at x whenever it is Lipschitz continuous near x. It follows that a norm on X is $\partial^{\#}$-smooth if and only if it is ${}^{\#}$-differentiable off the origin, and that any norm on X is ∂^I- and ∂^{CR}-smooth. Other noteworthy examples are the viscosity subdifferentials. See [3] for more details.

The following proposition provides crucial examples of ∂-differentiable functions. Recall that a subset A of a normed space $(X, \|.\|)$ is said to be *proximinal* if every point in X has a closest point in A (with respect to $\|.\|$). Any boundedly weakly compact set is proximinal; in particular, any nonempty closed convex set in a reflexive Banach space is proximinal.

Proposition 2.3. *Let X be a Banach space with a ∂-smooth norm and let $C \subset X$ be convex and proximinal. Then, for any $\delta \geq 0$, the function*

$$x \mapsto d^2_{B_\delta(C)}(x)$$

is ∂-differentiable.

Proof. For any $\delta \geq 0$, $B_\delta(C)$ is clearly convex and also proximinal: if $x \notin B_\delta(C)$ and x_C is a closest point to x in C, then $y := x_C + \delta(x - x_C)/\|x - x_C\|$ is a closest point to x in $B_\delta(C)$. The proof thus reduces to show that for every $x \in X$ and every nonempty $A \subset X$, the set $\partial d^2_A(x)$ is nonempty provided A is convex, while the set $\partial(-d^2_A)(x)$ is nonempty provided A is proximinal. The first assertion follows from property (P1) and convex analysis because d^2_A is convex continuous. Now assume that A is proximinal. Let x_A be a closest point to x in A and define $\varphi : X \to \mathbb{R}$ by $\varphi(y) := \|y - x_A\|^2$. Since $\varphi - d^2_A$ attains its minimum at x, and φ is convex, continuous, and ∂-differentiable (because $\|.\|$ is ∂-smooth), we derive from properties (P2) and (P3) that $0 \in \partial\varphi(x) + \partial(-d^2_A)(x)$, proving the nonemptiness of $\partial(-d^2_A)\ (x)$. □

This motivates the next definition.

Definition 2.4. *A pair (X, C), with $\emptyset \neq C \subset X$, is said to be* ∂-smooth *if X admits a ∂-smooth renorm such that for any $\delta \geq 0$, the function*

$$x \mapsto d^2_{B_\delta(C)}(x)$$

is ∂-differentiable.

Combining Proposition 2.3 with classical renorming theorems, we easily get examples of ∂-smooth pairs (X, C), e.g.,

—X and C arbitrary, with $\partial = \partial^I$ or $\partial = \partial^{CR}$;

—X having a ∂-smooth renorm, C boundedly weakly compact and convex, with ∂ arbitrary;

—X reflexive, C closed and convex, with $\partial = \partial^F$ (where ∂^F stands for the Fréchet subdifferential);

—X superreflexive, C closed and convex, with $\partial = \partial^{HS}$ (where ∂^{HS} stands for the Hölder-smooth subdifferential);

—$X = L^p$, with $2 \leq p < \infty$, C closed and convex, with $\partial = \partial^{LS}$ (where ∂^{LS} stands for the Lipschitz-smooth subdifferential);

—X Hilbert space, C closed and convex, with $\partial = \partial^\pi$ (where ∂^π stands for the proximal subdifferential).

3. The ε-variational principle for constrained minimization problems.

The ε-variational principle of Ekeland [11] and its smooth version by Borwein and Preiss [4] are aimed at unconstrained optimization problems. In this section, we provide a variant of the smooth principle suitable for *constrained* optimization problems. The proof is based on the following adaptation of Borwein and Preiss's theorem; see [3, Theorem 3.1].

THEOREM 3.1. *Let X be a Banach space with a ∂-smooth norm and let $f : X \to \mathbb{R} \cup \{+\infty\}$ be lower semicontinuous. Let $A \subset X$ be a closed set and let $\varepsilon > 0$ be a given constant. Suppose that $x_0 \in A$ and $\lambda > 0$ satisfy*

$$B_\lambda(x_0) \subset A \quad \text{and} \quad f(x_0) < \inf_A f + \varepsilon.$$

Then there exists $\bar{x}$ in X verifying

$$\|\bar{x} - x_0\| < \lambda, \qquad f(\bar{x}) < \inf_A f + \varepsilon, \qquad \text{and}$$

$$0 \in \partial f(\bar{x}) + 2(\varepsilon/\lambda)B^*,$$

where B^* is the dual closed unit ball.

In the general framework of a nonsmooth function $f : X \to \mathbb{R} \cup \{+\infty\}$ to be minimized on a constraint set C, the natural value to be considered is

$$r_C(f) := \sup_{\delta > 0} \inf_{B_\delta(C)} f.$$

Note that $r_C(f)$ is independent of the norm used to describe the topology of X. Plainly, $r_C(f) \leq \inf_C f$. The following proposition lists important cases where equality holds.

PROPOSITION 3.2. *Let X be a Banach space, $f : X \to \mathbb{R} \cup \{+\infty\}$ lower semicontinuous, and $C \subset X$ with* $\operatorname{dom} f \cap C \neq \emptyset$. *Then, $r_C(f) = \inf_C f$ in the following cases:*

(1) *$C = X$;*

(2) *f is uniformly continuous on a uniform neighborhood of C;*

(3a) *f is $\mathcal{T}$-lower semicontinuous on a neighborhood of C and C is $\mathcal{T}$-compact, where $\mathcal{T}$ is any vector space topology on X that is weaker than the norm topology;*

(3b) *f is $\mathcal{T}$-*inf*-compact and C is $\mathcal{T}$-closed, where $\mathcal{T}$ is any vector space topology on X which is weaker than the norm topology;*

(4) *$X = \mathbb{R}_+(\operatorname{dom} f - C)$, f is convex lower semicontinuous, and C is closed and convex.*

Proof. Case (1) is obvious, and cases (2), (3a), and (3b) can easily be proved directly. Otherwise, observe that $r_C(f) = \inf_C f$ if and only if the function $x \mapsto h(x) := \inf_{y \in X}(f(y) + \psi_C(y - x))$ is lower semicontinuous at 0. Then, to prove case (4), invoke a standard result of convex analysis stating that, under such a qualification condition, h is in fact continuous at 0. □

Our constrained version of the ε-variational principle can now be stated.

THEOREM 3.3. *Let X be a Banach space with a ∂-smooth norm; let $C \subset X$ be nonempty, closed, and convex such that the function d_C^2 is ∂-differentiable; and let $f : X \to \mathbb{R} \cup \{+\infty\}$ be lower semicontinuous. Let $\varepsilon > 0$ be a given constant and suppose that $x_0 \in X$ satisfies*

$$x_0 \in C \quad \text{and} \quad f(x_0) < r_C(f) + \varepsilon.$$

Then, for any $\lambda > 0$ *such that* f *is bounded below on* $B_\lambda(x_0)$, *there exist* $\bar{x} \in X$ *and* $K > 0$ *verifying*

$$\|\bar{x} - x_0\| < \lambda, \qquad f(\bar{x}) < f(x_0) + \varepsilon, \qquad \textit{and}$$
$$0 \in \partial f(\bar{x}) + K\partial d_C^2(\bar{x}) + 2(\varepsilon/\lambda)B^*.$$

Proof. Let $0 < \varepsilon' < \varepsilon$ such that $f(x_0) < r_C(f) + \varepsilon'$. By the definition of $r_C(f)$, there exists $\delta > 0$ such that

$$f(x_0) < \inf_{B_\delta(C)} f + \varepsilon'.$$

Set $A := B_\delta(C) \cup B_\lambda(x_0)$ and let $K > 0$ such that

$$f(x_0) < \inf_A f + K\delta^2 + \varepsilon'.$$

Consider $g := f + Kd_C^2$. We have

$$g(x_0) = f(x_0) < \inf_A g + \varepsilon;$$

indeed, if $x \in B_\delta(C)$, then $g(x) \geq f(x) > f(x_0) - \varepsilon'$, while if $x \in A \setminus B_\delta(C)$, then $g(x) = f(x) + Kd_C^2(x) \geq f(x) + K\delta^2 > f(x_0) - \varepsilon'$.

Applying Theorem 3.1, we find $\bar{x} \in X$ satisfying

$$\|\bar{x} - x_0\| < \lambda, \qquad g(\bar{x}) < \inf_A g + \varepsilon, \qquad 0 \in \partial g(\bar{x}) + 2(\varepsilon/\lambda)B^*.$$

It follows that $f(\bar{x}) \leq g(\bar{x}) < g(x_0) + \varepsilon = f(x_0) + \varepsilon$, and, using (P3), that

$$0 \in \partial f(\bar{x}) + K\partial d_C^2(\bar{x}) + 2(\varepsilon/\lambda)B^*. \qquad \square$$

Remark 3.1. As was observed in [3], if f is "smooth," i.e., $f = g + h$ with g Gâteaux-differentiable and h convex, Theorem 3.1 remains true for $\partial = \partial^G$ without requiring that the norm be ∂^G-smooth. Equally in that case, Theorem 3.3 remains true without assuming that the norm is ∂^G-smooth or that d_C^2 is ∂-differentiable.

4. Fermat rules. As a straightforward consequence of Theorem 3.3, we obtain a general Fermat rule for nonsmooth constrained minimization problems.

THEOREM 4.1. *Let* (X, C) *be a* ∂*-smooth pair, with* C *closed and convex, and let* $f : X \to \mathbb{R} \cup \{+\infty\}$ *be lower semicontinuous and bounded below on a uniform neighborhood of* C. *Assume that* $\operatorname{dom} f \cap C \neq \emptyset$, *and let* (y_n) *be a sequence in* X *such that*

$$d_C(y_n) \to 0 \quad \textit{and} \quad f(y_n) \to r_C(f).$$

Then, there exist a subsequence (y_{k_n}) *of* (y_n) *and sequences* $(x_n) \subset \operatorname{dom} \partial f$ *and* $x_n^* \in \partial f(x_n)$ *such that*

(i) $\|x_n - y_{k_n}\| \to 0$, $f(x_n) \to r_C(f)$;

(ii) $\langle x_n^*, c - x_n\rangle \geq -(1/n)\|c - x_n\|$ *for all* $c \in C$ *and all* $n \in \mathbb{N}$.

Proof. Without any loss of generality, we may assume that the given norm of X is ∂-smooth and that for any $\delta \geq 0$ the function $d_{B_\delta(C)}^2$ is ∂-differentiable. Note that the assumptions imply that $r_C(f) \in \mathbb{R}$. Define a sequence $\gamma_n \to 0^+$ such that

$$r_C(f) + 1/2n^2 < r_{B_{\gamma_n}(C)}(f) + 1/n^2.$$

Then, take a subsequence (y_{k_n}) of (y_n) such that

$$d_C(y_{k_n}) < \gamma_n \quad \text{and} \quad f(y_{k_n}) < r_{B_{\gamma_n}(C)}(f) + 1/n^2.$$

Clearly, we may assume that f is bounded below on $B_{\gamma_n}(C) \cup B_{2/n}(y_{k_n})$. For each $n \in \mathbb{N}$, apply Theorem 3.3 to $B_{\gamma_n}(C)$ with $\varepsilon := 1/n^2$, $x_0 := y_{k_n}$, and $\lambda := 2/n$ to obtain sequences $(x_n) \subset X$ and $K_n > 0$ such that

$$\|x_n - y_{k_n}\| \leq 2/n,$$

$$f(x_n) < f(y_{k_n}) + 1/n^2,$$

$$0 \in \partial f(x_n) + K_n \partial d^2_{B_{\gamma_n}(C)}(x_n) + (1/n)B^*.$$

The above inequalities yield assertion (i). The third expression gives $x_n^* \in \partial f(x_n)$, $\xi_n^* \in \partial d^2_{B_{\gamma_n}(C)}(x_n)$, and $\beta_n^* \in B^*$ satisfying

$$0 = x_n^* + K_n \xi_n^* + (1/n)\beta_n^*,$$

from which assertion (ii) follows, since for any $c \in C$ it holds that

$$\langle \xi_n^*, c - x_n \rangle \leq 0,$$

so that

$$\langle x_n^*, c - x_n \rangle \geq -(1/n)\langle \beta_n^*, c - x_n \rangle \geq -(1/n)\,\|c - x_n\|. \qquad \square$$

Remark 4.1. Because of Remark 3.1, if f is "smooth," Theorem 4.1 is still valid for arbitrary pairs (X, C), with C closed and convex, and $\partial = \partial^G$. Of course, this remark applies also to the following corollaries and to the results of the subsequent sections.

Theorem 4.1 covers a wide range of situations. As an illustration, we consider three typical special cases:

—$C = X$;

—f is Lipschitz near a local minimum over C;

—$f = g + h$ with g Fréchet differentiable, h convex "qualified," and f attaining a local minimum over C.

COROLLARY 4.2. *Let X be a Banach space with a ∂-smooth renorm, and $f\colon X \to \mathbb{R} \cup \{+\infty\}$ be lower semicontinuous and bounded below. Assume that* $\operatorname{dom} f \neq \emptyset$, *and let (y_n) be a minimizing sequence. Then, there exist sequences $(x_n) \subset \operatorname{dom} \partial f$, $x_n^* \in \partial f(x_n)$ such that*

$$\|x_n - y_n\| \to 0, \qquad f(x_n) \to \inf_X f, \qquad \textit{and} \quad \|x_n^*\| \to 0.$$

Proof. Letting $C = X$ in Theorem 4.1, it is easy to see from the proof that the conclusions hold without passing to a subsequence of (y_n). Assertion (ii) clearly gives that $\|x_n^*\| \to 0$. $\square$

If, in Corollary 4.2, we assume further that $f = g + h : X \to \mathbb{R} \cup \{+\infty\}$ with g Gâteaux-differentiable and h convex, then X may be any Banach space (see Remark 4.1), and the conclusion reads thusly: there exist $(x_n) \subset \operatorname{dom} \partial h$ and $x_n^* \in \partial h(x_n)$ such that

$$\|x_n - y_n\| \to 0, \qquad f(x_n) \to \inf_X f, \qquad \text{and} \quad \|g'(x_n) + x_n^*\| \to 0.$$

This improves, e.g., Aubin and Ekeland [1, Corollary 7, p. 259].

We now elaborate on the case when f achieves a local minimum over C at point $x_0 \in C$. In that case, an approximate variational inequality involving x_0 can be obtained (see Corollary 6.2 below) but, as simple examples show, one cannot expect that $\partial f(x_0)$ be nonempty without further regularity or qualification conditions. We end this section by showing how two such classical conditions can be recovered from Theorem 4.1. The first result is well known for the case $\partial = \partial^{CR}$; the second one can be obtained through standard arguments of convex analysis.

Given a subdifferential ∂, $f : X \to \mathbb{R} \cup \{+\infty\}$ lower semicontinuous and $x \in X$, we let $\widehat{\partial} f(x)$ be the set of weak* cluster points of sequences $x_n^* \in \partial f(x_n)$ as $x_n \to_f x$.

Corollary 4.3. *Let (X, C) be a ∂-smooth pair, with C closed and convex, and $f : X \to \mathbb{R} \cup \{+\infty\}$ be lower semicontinuous. Assume that f attains a local minimum over C at x_0 and that the following regularity condition holds:*

(R) *∂f is locally bounded at x_0.*

Then, there exists $x_0^* \in \widehat{\partial} f(x_0)$ such that

$$\langle x_0^*, c - x_0 \rangle \geq 0 \quad \textit{for all } c \in C.$$

Proof. Let B be a ball around x_0 such that $f(x_0) = \min_{C \cap B} f$ and $\partial f(B)$ is bounded. Then f is Lipschitz on B (see [3, Theorem 5.2]), so $f(x_0) = r_{C \cap B}(f)$ by Proposition 3.2 (2). Applying Theorem 4.1 with $y_n := x_0$, we get sequences $x_n \to_f x_0$, $x_n^* \in \partial f(x_n)$ such that

$$\langle x_n^*, c - x_n \rangle \geq -(1/n) \, \|c - x_n\| \quad \text{for all} \quad c \in C \cap B.$$

Since the sequence (x_n^*) eventually lies in the bounded set $\partial f(B)$, it has a weak* cluster point $x_0^* \in \widehat{\partial} f(x_0)$. Clearly, $\langle x_0^*, c - x_0 \rangle \geq 0$ for all $c \in C \cap B$. The conclusion follows. □

The regularity condition (R) in Corollary 4.3 is satisfied in particular if f is Lipschitz near x_0 and $\partial f \subset \partial^{CR} f$. Thus, Corollary 4.3 applies whenever f is Lipschitz near x_0, and, e.g.,

—X is any Banach space and $\partial = \partial^I$ or ∂^{CR}, or

—X is reflexive and $\partial = \partial^F$, or

—X is a Hilbert space and $\partial = \partial^\pi$.

Moreover, if X admits a Gâteaux renorm (for example, X reflexive or X separable), then instead of $\widehat{\partial}$ in Corollary 4.3 we may use the sequential limit of ∂ given by

$$\widetilde{\partial} f(x) := \{x^* \in X^* \mid x_n^* \xrightarrow{w^*} x^*, x_n^* \in \partial f(x_n), x_n \to x\},$$

because in that case every bounded sequence of X^* has a weak* converging subsequence.

Corollary 4.4. *Let X be a Banach space, $C \subset X$ be nonempty closed convex, and $f := g + h$, with $g : X \to \mathbb{R}$ Fréchet differentiable and $h : X \to \mathbb{R} \cup \{+\infty\}$ convex lower semicontinuous. Assume that f attains a local minimum over C at x_0 and that the following qualification condition holds:*

(Q) *$X = \mathbb{R}_+(\operatorname{dom} h - C)$.*

Then, there exists $x_0^* \in \partial h(x_0)$ such that

$$\langle \nabla^F g(x_0) + x_0^*, c - x_0 \rangle \geq 0 \quad \textit{for all} \quad c \in C.$$

Proof. For each $k \in \mathbb{N}$, let $\gamma_k > 0$ such that the convex lower semicontinuous function $\varphi_k := \nabla^F g(x_0) + (1/k) \| \,.- x_0 \| + h$ attains its minimum over $C_k := C \cap B_{\gamma_k}(x_0)$

at x_0. Of course, $X = \mathbb{R}_+(\operatorname{dom}\varphi_k - C_k)$, and because of Proposition 3.2 (4), $r_{C_k}(\varphi_k) = \varphi(x_0)$. Applying Theorem 4.1, we get sequences $x_n \to_{\varphi_k} x_0$ and $y_n^* \in \partial\varphi_k(x_n)$ such that

$$\langle y_n^*, c - x_n\rangle \geq -(1/n)\,\|c - x_n\| \quad \text{for all} \quad c \in C_k.$$

Writing $y_n^* = \nabla^F g(x_0) + (1/k)\beta_n^* + x_n^*$ with $\beta_n^* \in \partial(\|\ .-x_0\|)(x_n)$ and $x_n^* \in \partial h(x_n)$, we obtain

$$\langle \nabla^F g(x_0) + x_n^*, c - x_n\rangle \geq -(1/n + 1/k)\,\|c - x_n\| \quad \text{for all} \quad c \in C_k.$$

Since $x_n \to_{\varphi_k} x_0$, we have $x_n \to_h x_0$. Combining the above inequality with the qualification condition, we derive that (x_n^*) is pointwise bounded, hence norm bounded by the uniform boundedness principle. If x_0^* is a weak* cluster point of (x_n^*), then $x_0^* \in \partial h(x_0)$ and we deduce from the previous inequality that

$$\langle \nabla^F g(x_0) + x_0^*, c - x_0\rangle \geq -(1/k)\,\|c - x_0\| \quad \text{for all} \quad c \in C_k.$$

By convexity of C and homogeneity, this inequality actually holds for all $c \in C$, and the conclusion follows from the arbitrariness of k. ◻

5. Multidirectional Rolle-type inequalities.

For a nonempty closed convex set $C \subset X$ and $a \in X$, we let

$$[a, C] := \{x \in X \mid x = a + t(c - a) \text{ for some } t \in [0,1] \text{ and some } c \in C\}$$

be the closed convex "drop" joining a and C.

PROPOSITION 5.1. *Let $D = [a, C]$, with $C \subset X$ nonempty, closed, convex, and $a \in X$. Let $x \in X$ be such that $d_D(x) < d_C(x)$. Then, for any $V := B_\delta(D)$, $\delta \geq 0$, and any $\xi^* \in \partial d_V^2(x)$, it holds that*

$$\langle \xi^*, d - a\rangle \leq 0 \quad \textit{for all } d \in D.$$

Proof. It follows from the classical chain rule of subdifferential calculus that $\xi^* = 2d_V(x)\zeta^*$, where $\zeta^* \in \partial d_V(x)$. If $x \in V$, then $\xi^* = 0$, so we assume that $x \notin V$. In this case, we have $d_D(x) = d_V(x) + \delta$, hence $\zeta^* \in \partial d_D(x)$. Let $k \in \mathbb{N}$ such that

$$d_D(x) + 2/k < d_C(x),$$

and let $x_k \in D$ such that

$$\|x - x_k\| \leq d_D(x) + 1/k^2.$$

For any $d \in D$, we have

$$\begin{aligned}\langle \xi^*, d - x_k\rangle &= \langle \xi^*, d - x\rangle + \langle \xi^*, x - x_k\rangle\\ &= 2d_V(x)(\langle \zeta^*, d - x\rangle + \langle \zeta^*, x - x_k\rangle)\\ &\leq 2d_V(x)(-d_D(x) + \|x - x_k\|)\\ &\leq 2d_V(x)/k^2.\end{aligned}$$

On the other hand, it is easily seen that $d_C(x_k) \geq 1/k$, hence $x_k - a = t(\bar{c} - x_k)$ for some $\bar{c} \in C$ and $0 \leq t \leq k\|x_k - a\|$. So, for any $d \in D$, it holds that

$$\begin{aligned}\langle \xi^*, d - a\rangle &= \langle \xi^*, d - x_k\rangle + \langle \xi^*, x_k - a\rangle\\ &= \langle \xi^*, d - x_k\rangle + t\langle \xi^*, \bar{c} - x_k\rangle\\ &\leq 2d_V(x)(1/k^2 + t/k^2)\\ &\leq 2d_V(x)(1/k^2 + \|x_k - a\|/k).\end{aligned}$$

Since $\|x_k - a\|$ is bounded (by $\|x - a\| + d_D(x) + 1$), letting $k \to \infty$ yields the result. □

Combining the above proposition with Theorem 3.3, we readily obtain a multidirectional Rolle-type inequality.

THEOREM 5.2. *Let (X, D) be a ∂-smooth pair with $D = [a, C]$, where C is closed and convex and $a \in X$. Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be lower semicontinuous and bounded below on a uniform neighborhood of D. Assume that* $\operatorname{dom} f \cap D \neq \emptyset$ *and that (y_n) is a sequence in X such that*

$$d_D(y_n) \to 0, \qquad f(y_n) \to r_D(f), \qquad \text{and} \quad (y_n) \subset X \setminus B_\delta(C) \quad \text{for some } \delta > 0.$$

Then, there exist a subsequence (y_{k_n}) of (y_n) and sequences $(x_n) \subset \operatorname{dom}\partial f$ and $x_n^* \in \partial f(x_n)$ such that

(i) $\|x_n - y_{k_n}\| \to 0, f(x_n) \to r_D(f)$ *;*

(ii) $\langle x_n^*, d - x_n\rangle \geq -(1/n)\|d - x_n\|$ *for all $d \in D$ and all $n \in \mathbb{N}$;*

(iii) $\langle x_n^*, d - a\rangle \geq -(1/n)\|d - a\|$ *for all $d \in D$ and all $n \in \mathbb{N}$.*

Proof. The sequences (y_{k_n}), (x_n), and (x_n^*) are constructed as in the proof of Theorem 4.1, with D in place of C. To prove assertion (iii), recall that $x_n^* \in \partial f(x_n)$ satisfies

$$0 = x_n^* + K_n\xi_n^* + (1/n)\beta_n^*,$$

with $\xi_n^* \in \partial d^2_{B_{\gamma_n}(D)}(x_n)$ and $\beta_n^* \in B^*$. Since $d_D(y_{k_n}) \to 0$, $(y_{k_n}) \subset X \setminus B_\delta(C)$ and $\|x_n - y_{k_n}\| \to 0$, we have that $d_D(x_n) < d_C(x_n)$ for large n. It follows from Proposition 5.1 that for any $d \in D$ it holds that

$$\langle \xi_n^*, d - a\rangle \leq 0,$$

whence

$$\langle x_n^*, d - a\rangle \geq -(1/n)\langle \beta_n^*, d - a\rangle \geq -(1/n)\|d - a\|. \qquad \square$$

In view of applications of Theorem 5.2, it is worth noting that if f satisfies $r_D(f) < r_C(f)$, then any sequence (y_n) verifying $d_D(y_n) \to 0$ and $f(y_n) \to r_D(f)$ automatically satisfies $(y_n) \subset X \setminus B_\delta(C)$ for some $\delta > 0$, eventually. Another useful situation is described in the following corollary.

COROLLARY 5.3. *Let X, $D := [a, C]$, and f be as in Theorem 5.2. Assume that there exists $x_0 \in D \setminus C$ such that $f(x_0) \leq r_C(f)$. Then, there exist sequences $(x_n) \subset \operatorname{dom}\partial f$, $x_n^* \in \partial f(x_n)$ such that*

(i) $d_D(x_n) \to 0,\ f(x_n) \to r_D(f), (x_n) \subset X \setminus B_\delta(C)$ *for some $\delta > 0$;*

(ii) $\langle x_n^*, d - x_n\rangle \geq -(1/n)\|d - x_n\|$ *for all $d \in D$ and all $n \in \mathbb{N}$;*

(iii) $\langle x_n^*, d - a\rangle \geq -(1/n)\|d - a\|$ *for all $d \in D$ and all $n \in \mathbb{N}$.*

Furthermore, if $f(x_0) = r_D(f)$, we may choose (x_n) so that $x_n \to_f x_0$.

Proof. If $f(x_0) = r_D(f)$, the result follows from Theorem 5.2 with $y_n = x_0$ for all $n \in \mathbb{N}$. If $f(x_0) > r_D(f)$, then $r_D(f) < r_C(f)$ and the result follows from Theorem 5.2 through the above remark. □

Remark 5.1. Corollary 5.3 contains our previous result [3, Theorem 4.1], which is the case when C is a singleton (so that D is a compact segment) and $f(x_0) = r_D(f) = \min_D f$.

6. Multidirectional mean value inequalities. The approximate mean value theorem of Zagrodny [17] has proved to be a powerful tool in nonsmooth analysis (see, e.g., Correa, Jofré, and Thibault [7], Thibault and Zagrodny [16], and Aussel, Corvellec, and Lassonde [2, 3]). The multidirectional mean value inequality of Clarke and Ledyaev [6] is a (nontrivial) partial extension of Zagrodny's theorem: instead of dealing with segments $[a, c]$, it is concerned with "drops" $[a, C]$, where C is a closed bounded convex set. In this section, we establish a general multidirectional mean value inequality containing both Zagrodny's and Clarke and Ledyaev's results, by applying Corollary 5.3 in the Banach space $X \times \mathbb{R}$. To this end, we need specific assumptions concerning the ∂-smoothness of the space $X \times \mathbb{R}$ and the behavior of ∂ on a certain class of functions of $X \times \mathbb{R}$.

Throughout this section, we assume that X and ∂ satisfy the following:

(S) For every nonempty closed convex $C \subset X \times \mathbb{R}$, the pair $(X \times \mathbb{R}, C)$ is ∂-smooth.

(P4) If $\varphi : X \times \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is such that $\varphi(x,t) = f(x) + \alpha t$, where $f : X \to \mathbb{R} \cup \{+\infty\}$ is lower semicontinuous and $\alpha \in \mathbb{R}$, then $\partial\varphi(x,t) \subset \partial f(x) \times \{\alpha\}$.

These properties are verified in most situations met in applications, e.g.,

—X is a Banach space and $\partial = \partial^I$ or ∂^{CR};

—X is a reflexive Banach space and $\partial = \partial^F$;

—X is a superreflexive Banach space and $\partial = \partial^{HS}$;

—X is an L^p space, with $2 \le p < +\infty$, and $\partial = \partial^{LS}$;

—X is a Hilbert space and $\partial = \partial^\pi$.

THEOREM 6.1. *Let X and ∂ satisfy* (S) *and* (P4), *and let $D = [a, C]$, with $C \subset X$ nonempty, closed, convex, and $a \in X$. Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be lower semicontinuous and bounded below on a uniform neighborhood of D. Assume that $a \in \operatorname{dom} f$, and let $r \in \mathbb{R}$ with $r \le r_C(f)$. Then, there exist sequences $(x_n) \subset \operatorname{dom} \partial f$ and $x_n^* \in \partial f(x_n)$ such that*

(i) *$d_D(x_n) \to 0\,,\ f(x_n) \to \rho$ with $r_D(f) \le \rho \le r_D(f) + |r - f(a)|$;*

(ii) *$\langle x_n^*, d - x_n\rangle \ge -|r - f(a)| - (1/n)\|d - x_n\| - 1/n$ for all $d \in D$ and all n;*

(iii) *$\langle x_n^*, c - a\rangle \ge r - f(a) - (1/n)\|c - a\| - 1/n$ for all $c \in C$ and all n.*

Furthermore, if $f(a) = r = r_D(f)$, we may choose (x_n) so that $x_n \to_f a$, while if $f(a) \le r$ and $a \notin C$, we may choose (x_n) so that $(x_n) \subset X \setminus B_\delta(C)$ for some $\delta > 0$.

Proof. Supply $X \times \mathbb{R}$, say, with the Euclidean product norm

$$\|(x,t)\|_{X\times\mathbb{R}} := (\|x\|^2 + t^2)^{1/2}\,;$$

define $\varphi : X \times \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ by

$$\varphi(x,t) := f(x) - (r - f(a))t$$

and let $\tilde{a} := (a, 0)$, $\tilde{C} := C \times \{1\}$, $\tilde{D} := [\tilde{a}, \tilde{C}]$, so that

$$\tilde{D} = \{(x,t) \in X \times [0,1] \mid x = a + t(c - a) \text{ for some } c \in C\}.$$

We claim that

$$\varphi(\tilde{a}) = f(a) \le r_{\tilde{C}}(\varphi)\,.$$

Indeed, let $\varepsilon > 0$ be an arbitrary constant, and let $\delta > 0$ be such that $r - \varepsilon < \inf_{B_\delta(C)} f$. It follows that $f(a) - \varepsilon < \inf_{B_\delta(C)} f - (r - f(a))$, whence

$$f(a) - \varepsilon < \inf_{B_\delta(C)} f - (r - f(a))t \quad \text{for any } t \text{ close to 1.}$$

Therefore, some $\gamma > 0$ exists such that

$$f(a) - \varepsilon < \varphi(x,t) \quad \text{for all } (x,t) \in B_\gamma(C) \times B_\gamma(\{1\}),$$

which implies $f(a) - \varepsilon \leq r_{\tilde{C}}(\varphi)$. The claim is proved.

On the other hand, φ is lower semicontinuous and bounded below on a uniform neighborhood of $\tilde{D}$, and $\tilde{a} \in \tilde{D} \setminus \tilde{C}$. Thanks to (S), we may apply Corollary 5.3 to φ, $\tilde{D}$, and $\tilde{a}$, getting sequences $(\tilde{x}_n) \subset \operatorname{dom} \partial\varphi$ and $\tilde{x}_n^* \in \partial\varphi(\tilde{x}_n)$ with

$$d_{\tilde{D}}(\tilde{x}_n) \to 0, \qquad \varphi(\tilde{x}_n) \to r_{\tilde{D}}(\varphi); \tag{6.1}$$

$$\langle \tilde{x}_n^*, \tilde{d} - \tilde{x}_n \rangle \geq -(1/n)\|\tilde{d} - \tilde{x}_n\| \quad \text{for all } \tilde{d} \in \tilde{D} \text{ and all } n \in \mathbb{N}; \tag{6.2}$$

$$\langle \tilde{x}_n^*, \tilde{d} - \tilde{a} \rangle \geq -(1/n)\|\tilde{d} - \tilde{a}\| \quad \text{for all } \tilde{d} \in \tilde{D} \text{ and all } n \in \mathbb{N}. \tag{6.3}$$

Writing $\tilde{x}_n := (x_n, t_n)$, we derive from (6.1) that, up to a subsequence, $t_n \to \tau \in [0,1]$, $d_D(x_n) \to 0$, and $f(x_n) \to \rho := r_{\tilde{D}}(\varphi) + (r - f(a))\tau$. From the very definition of $r_D(f)$, we have that $\rho \geq r_D(f)$. To complete the proof of assertion (i), it remains to show that

$$\rho \leq r_D(f) + |r - f(a)|.$$

The following straightforward observation will be helpful: for any $\delta > 0$ and any $x \in B_\delta(D)$, there exists $t \in [0,1]$ such that $(x,t) \in B_\delta(\tilde{D})$. Assume first that $r - f(a) \geq 0$. Then, $\varphi(x,t) \leq f(x)$ for all $x \in X$ and $t \geq 0$, which, combined with the previous observation, gives $r_{\tilde{D}}(\varphi) \leq r_D(f)$, whence the result. Assume now that $r - f(a) \leq 0$. Then, $\varphi(x,t) \leq f(x) - (r - f(a))$ for all $x \in X$ and $t \leq 1$, so that, as above, we get $r_{\tilde{D}}(\varphi) \leq r_D(f) - (r - f(a))$, whence the result again.

Now, according to (P4) we have $\tilde{x}_n^* = (x_n^*, -(r - f(a)))$ with $x_n^* \in \partial f(x_n)$. Substitution in (6.2) gives the following: for each $d \in a + t(C - a)$, with $0 \leq t \leq 1$, and each $n \in \mathbb{N}$, it holds that

$$\begin{aligned}\langle x_n^*, d - x_n \rangle &\geq (t - t_n)(r - f(a)) - (1/n)(\|d - x_n\| + |t - t_n|)\\ &\geq -|r - f(a)| - (1/n)\|d - x_n\| - 1/n - |\tau - t_n|(|r - f(a)| + 1/n),\end{aligned}$$

from which assertion (ii) follows if we agree that (6.2) could be given with $1/2n$ instead of $1/n$ and that t_n could be such that $|\tau - t_n|(|r - f(a)| + 1) \leq 1/2n$. Similarly, (6.3) yields

$$\langle x_n^*, c - a \rangle \geq r - f(a) - (1/n)\|c - a\| - 1/n \quad \text{for all } c \in C \text{ and all } n \in \mathbb{N}.$$

This is assertion (iii), so the first part of the theorem is proved.

The last statements follow from observations on the above proof. Assume that $f(a) \leq r$. If $r_{\tilde{D}}(\varphi) = \varphi(\tilde{a}) = f(a)$, by Corollary 5.3 we may choose $\tilde{x}_n \to_\varphi \tilde{a}$, which implies that $x_n \to_f a$. This case holds in particular whenever $f(a) = r = r_D(f)$. If $r_{\tilde{D}}(\varphi) < \varphi(\tilde{a}) = f(a)$, we have

$$\begin{aligned}\rho &= r_{\tilde{D}}(\varphi) + (r - f(a))\tau\\ &\leq r_{\tilde{D}}(\varphi) + r - f(a)\\ &< r \leq r_C(f);\end{aligned}$$

but $f(x_n) \to \rho < r_C(f)$ and $d_D(x_n) \to 0$, so necessarily $x_n \notin B_\delta(C)$ for some $\delta > 0$ and large n. □

Remark 6.1. If C is weakly compact, then the set $\tilde{D}$ considered in the proof is also weakly compact. In that case, the property (S) can therefore be relaxed to simply

(S′) $X \times \mathbb{R}$ admits a ∂-smooth renorm,

which, for all natural subdifferentials, amounts to saying that X admits a ∂-smooth renorm (see Fact 1 in the Appendix). This remark also applies to the forthcoming results.

Theorem 6.1 brings interesting complements to the Fermat rule (Theorem 4.1).

Corollary 6.2. *Let X and ∂ satisfy* (S) *and* (P4)*; let $C \subset X$ be nonempty, closed, and convex; and let $f : X \to \mathbb{R} \cup \{+\infty\}$ be lower semicontinuous. Assume that* $\operatorname{dom} f \cap C \neq \emptyset$, *and that $x_0 \in C$ verifies $f(x_0) = r_C(f)$. Then, there exist sequences* $(x_n) \subset \operatorname{dom} \partial f$ *and $x_n^* \in \partial f(x_n)$ such that*

(i) $x_n \to_f x_0$*;*

(ii) $\langle x_n^*, c - x_n\rangle \geq -(1/n)\|c - x_n\| - 1/n$ *for all $c \in C$ and all $n \in \mathbb{N}$;*

(iii) $\langle x_n^*, c - x_0\rangle \geq -(1/n)\|c - x_0\| - 1/n$ *for all $c \in C$ and all $n \in \mathbb{N}$.*

Proof. Apply Theorem 6.1 with $a := x_0$ (so that $D = C$) and $r := f(a) = r_D(f)$. □

Corollary 6.3. *Let X and ∂ satisfy* (S′) *and* (P4), *let Y be a closed vector subspace of X, and let $f : X \to \mathbb{R} \cup \{+\infty\}$ be lower semicontinuous. Assume that f attains a local minimum over Y at $x_0 \in Y$ and that either*

(i) *Y is finite-dimensional, or*

(ii) *Y is reflexive and f is weakly lower semicontinuous near x_0, or*

(iii) *Y is reflexive and f is uniformly continuous near x_0.*

Then, there exist sequences $(x_n) \subset \operatorname{dom} \partial f$ and $x_n^* \in \partial f(x_n)$ such that

(a) $x_n \to_f x_0$*;*

(b) $\|x_n^*\|_{Y^*} \to 0$.

Proof. Apply Corollary 6.2 with $C := B_\delta(x_0) \cap Y$, where $\delta > 0$ is such that $f(x_0) = \min_C f$ and f is weakly lower semicontinuous (respectively, uniformly continuous) around C in case (ii) (respectively, (iii)). Note that C is compact in case (i) and weakly compact in the other cases, so that $\min_C f = r_C(f)$ by Proposition 3.2. Moreover, (S) can indeed be relaxed to (S′); see Remark 6.1. Assertion (iii) of Corollary 6.2 obviously gives assertion (b) above. □

Assuming that the convex set C in Theorem 6.1 is bounded, we obtain a more precise and somewhat simpler result.

Theorem 6.4. *Let X and ∂ satisfy* (S) *and* (P4), *and let $D = [a, C]$, with $C \subset X$ nonempty, closed, bounded, convex, and $a \in X$. Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be lower semicontinuous and bounded below on a uniform neighborhood of D. Assume that $a \in \operatorname{dom} f$, and let $r \in \mathbb{R}$ with $r \leq r_C(f)$. Then, there exist $0 \leq \tau < 1$ and sequences $(x_n) \subset \operatorname{dom} \partial f$ and $x_n^* \in \partial f(x_n)$ such that*

(i) $d_{a+\tau(C-a)}(x_n) \to 0\,,\ f(x_n) \to \rho$ *with* $r_D(f) \leq \rho \leq r_D(f) + |r - f(a)|$*;*

(ii) *for any* $0 \leq t \leq 1$, $\liminf_{n\to\infty}\ \inf_{x \in a+t(C-a)}\ \langle x_n^*, x - x_n\rangle \geq (t - \tau)(r - f(a))$*;*

(iii) $\liminf_{n\to\infty} \inf_{c\in C} \langle x_n^*, c - a\rangle \geq r - f(a)\,.$

Furthermore, if $f(a) = r = r_D(f)$, we may choose (x_n) so that $x_n \to_f a$, while if $f(a) \leq r$ and $a \notin C$, we may choose (x_n) so that $(x_n) \subset X \setminus B_\delta(C)$ for some $\delta > 0$.

Proof. The proof is the same as that of Theorem 6.1, except for the following details. Observe that Corollary 5.3 provides the sequence $(\tilde{x}_n = (x_n, t_n))$ with

$$d_{\tilde{D}}(\tilde{x}_n) \to 0, \qquad \varphi(\tilde{x}_n) \to r_{\tilde{D}}(\varphi), \qquad (\tilde{x}_n) \subset X \times \mathbb{R} \setminus B_\delta(\tilde{C}) \quad \text{for some } \delta > 0\,.$$

From the boundedness of C, it is thus easily seen that there exists $0 \leq \tau < 1$ such that, up to a subsequence, $t_n \to \tau$ and $d_{a+\tau(C-a)}(x_n) \to 0$.

Also, recalling that $\tilde{x}_n^* = (x_n^*, -(r - f(a)))$ with $x_n^* \in \partial f(x_n)$, (6.2) and (6.3), respectively, yield

$$\liminf_{n\to\infty} \inf_{x\in a+t(C-a)} \langle x_n^*, x - x_n\rangle \geq \lim_{n\to\infty} (t-t_n)(r-f(a)) = (t-\tau)(r-f(a)), \quad 0 \leq t \leq 1,$$
$$\liminf_{n\to\infty} \inf_{c\in C} \langle x_n^*, c-a\rangle \geq r - f(a)\,. \qquad \square$$

The case $f(a) \leq r_C(f)$ in Theorem 6.4 is worth stating explicitly (compare with Corollary 5.3).

COROLLARY 6.5. *Let* X *and* ∂ *satisfy* (S) *and* (P4), *and let* $D = [a, C]$, *with* $C \subset X$ *nonempty, closed, bounded, convex, and* $a \in X$. *Let* $f : X \to \mathbb{R} \cup \{+\infty\}$ *be lower semicontinuous and bounded below on a uniform neighborhood of* D. *Assume that* $a \in \operatorname{dom} f \setminus C$ *and that* $f(a) + \varepsilon \leq r_C(f)$ *with* $\varepsilon \geq 0$. *Then, there exist* $0 \leq \tau < 1$ *and sequences* $(x_n) \subset \operatorname{dom} \partial f$ *and* $x_n^* \in \partial f(x_n)$ *such that*

(i) $d_D(x_n) \to 0$, $f(x_n) \to \rho$ *with* $r_D(f) \leq \rho \leq r_D(f) + \varepsilon$, *and* $(x_n) \subset X \setminus B_\delta(C)$ *for some* $\delta > 0$*;*

(ii) *for any* $t \geq 0$, $\liminf_{n\to\infty} \inf_{x\in a+t(C-a)} \langle x_n^*, x - x_n\rangle \geq (t-\tau)\varepsilon$*;*

(iii) *for any* $t \geq 0$, $\liminf_{n\to\infty} \inf_{x\in a+t(C-a)} \langle x_n^*, x - a\rangle \geq t\varepsilon$.

Proof. Apply Theorem 6.4 with $r := f(a) + \varepsilon$ to get assertion (i), assertion (ii) for $0 \leq t \leq 1$, and assertion (iii) for $t = 1$. Then observe that, if $t > 1$, it holds that

$$\inf_{x\in a+t(C-a)} \langle x_n^*, x - x_n\rangle \geq \inf_{c\in C} \langle x_n^*, c - x_n\rangle + (t-1) \inf_{c\in C} \langle x_n^*, c - a\rangle$$

so that

$$\liminf_{n\to\infty} \inf_{x\in a+t(C-a)} \langle x_n^*, x - x_n\rangle \geq (1-\tau)\varepsilon + (t-1)\varepsilon = (t-\tau)\varepsilon\,,$$

while if $t \geq 0$ it holds that

$$\liminf_{n\to\infty} \inf_{x\in a+t(C-a)} \langle x_n^*, x - a\rangle = t \liminf_{n\to\infty} \inf_{c\in C} \langle x_n^*, c - a\rangle \geq t\varepsilon. \qquad \square$$

Remark 6.2. (a) Theorem 6.1, as well as Theorem 6.4, contains the following: the Clarke–Ledyaev multidirectional mean value theorem [6, Theorem 2.1], where X is a Hilbert space, C is bounded, and $\partial = \partial^\pi$ is the proximal subdifferential; Theorem 3.2 in Radulescu and Clarke [15], where X is a uniformly smooth Banach space, C is bounded, and $\partial = \partial^F$; the main results in Luc [14], where C is norm (respectively, weakly) compact, f is norm (respectively, weakly) lower semicontinuous, and ∂ satisfies more restrictive properties than ours; and our previous result [3, Theorem 4.2], where C is a singleton, which is an extension of Zagrodny's approximate mean value inequality [17] for abstract subdifferential. Compare also with [15, Theorem 3.1], where X is a Banach space admitting a Lipschitz C^1 bump function, f is locally Lipschitz, and $\partial = \partial^F$: this result is obtained via the variational principle of Deville, Godefroy, and Zizler [9] instead of the variational principle of Borwein and Preiss.

(b) For $\partial = \partial^F$, cases (i) and (iii) of Corollary 6.3 are established in Deville and Ivanov [10, Theorem 2.1] and Deville and El Haddad [8, Theorem II-4], respectively, with slightly different assumptions (and completely different proofs). The case of a one-dimensional space Y is contained in [3, Theorem 4.1].

(c) Corollary 6.5 can be used to derive "subdifferential criteria," as in the case when C is a singleton; see [3, Corollary 4.3]. For example, we have the following *weak monotonicity* result generalizing [6, Theorem 6.1]:

Let X and ∂ satisfy (S) *and* (P4)*; let $Y \subset X$ be nonempty, closed, bounded, and convex; and let $f : X \to \mathbb{R} \cup \{+\infty\}$ be lower semicontinuous. If*

$$\inf_{y \in Y} \langle x^*, y\rangle \leq 0 \quad \textit{for all} \;\; x^* \in \operatorname{Im} \partial f,$$

then $f(a) \geq r_{a+\tau Y}(f)$ for all $a \in X$ and all $\tau > 0$.

Indeed, if we suppose that $a \in X$ and $\tau > 0$ are such that $f(a) < r_{a+\tau Y}(f)$, applying Corollary 6.5 (iii) with $C := a + \tau Y$, we get $x \in \operatorname{dom} \partial f$, $x^* \in \partial f(x)$ such that

$$\inf_{z \in a+(1/\tau)(C-a)} \langle x^*, z - a\rangle = \inf_{y \in Y} \langle x^*, y\rangle > 0.$$

Appendix: The compact case. This appendix provides complete, self-contained proofs of our main results in the (much easier) case where the constraint set is (weakly) compact. We use the notions of *subdifferential operator* ∂ and *∂-smooth norm* as given in [3] (see Definitions 2.1 and 2.2). We assume further that ∂ satisfies the following natural properties:

(P0) $\partial g(t) = \{g'(t)\}$ whenever $g : \mathbb{R} \to \mathbb{R}$ is of class C^∞;

(P4)$'$ If $\varphi : X \times \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is such that $\varphi(x, t) = f(x) + g(t)$, where $f : X \to \mathbb{R} \cup \{+\infty\}$ is lower semicontinuous and $g : \mathbb{R} \to \mathbb{R}$ is of class C^∞, then $\partial\varphi(x, t) \subset \partial f(x) \times \{g'(t)\}$ with equality if either f or g is a constant function.

Of course, property (P4)$'$ is a strengthening of property (P4).

Hereafter, X denotes a Banach space with a ∂-smooth renorm, C a nonempty, $\mathcal{T}$-compact, convex subset of X, and $f : X \to \mathbb{R} \cup \{+\infty\}$ a $\mathcal{T}$-lower semicontinuous function, where $\mathcal{T}$ is either the weak or the norm topology of X.

The proofs of our results are based on Borwein and Preiss's smooth variational principle in the form given in [3, Theorem 3.1] (also see our Theorem 3.1), and on the following two elementary facts.

Fact 1. $X \times \mathbb{R}$ admits a ∂-smooth renorm.

Indeed, if $\|.\|$ is a ∂-smooth norm on X, then the corresponding Euclidean product norm on $X \times \mathbb{R}$, namely,

$$\|(x, t)\|_{X\times\mathbb{R}} = (\|x\|^2 + t^2)^{1/2},$$

is ∂-smooth. Thus, let

$$\begin{aligned} \Delta_2(x, t) \;&:=\; \textstyle\sum_n \mu_n \|(x, t) - (v_n, \tau_n)\|^2_{X\times\mathbb{R}} = \sum_n \mu_n \|x - v_n\|^2 + \sum_n \mu_n (t - \tau_n)^2 \\ &=:\; \Delta_2(x) + \Delta_2(t), \end{aligned}$$

where μ_n and $(v_n, \tau_n) \subset X \times \mathbb{R}$ are as in Definition 2.2. It follows from (P1) and convex analysis that $\partial\Delta_2(x, t)$ is not empty. Now, define $\widetilde{\Delta}_2$ and $\overline{\Delta}_2$ on $X \times \mathbb{R}$ by

$$\widetilde{\Delta}_2(x, t) := \Delta_2(x)\,, \qquad \overline{\Delta}_2(x, t) := \Delta_2(t)$$

so that $\Delta_2 = \widetilde{\Delta}_2 + \overline{\Delta}_2$. According to (P4)$'$, $\partial(-\widetilde{\Delta}_2)(x, t) = \partial(-\Delta_2)(x) \times \{0\}$ and $\partial(-\overline{\Delta}_2)(x, t) = \{0\} \times \{-\Delta_2'(t)\}$; hence, $\widetilde{\Delta}_2$ and $\overline{\Delta}_2$ are both convex continuous and ∂-differentiable. Writing $-\Delta_2 + \overline{\Delta}_2 = -\widetilde{\Delta}_2$, we deduce from (P3) that

$$\emptyset \neq \partial(-\widetilde{\Delta}_2)(x, t) \subset \partial(-\Delta_2)(x, t) + \partial\overline{\Delta}_2(x, t)\,,$$

proving that $\partial(-\Delta_2)(x,t)$ is nonempty. $\square$

Fact 2. $x \mapsto d_D^2(x) := \inf\{\|x-d\|^2 \mid d \in D\}$ is ∂-differentiable whenever $D \subset X$ is nonempty, weakly compact, convex and $\|.\|$ is ∂-smooth.

Indeed, it follows from (P1) and convex analysis that the set $\partial d_D^2(x)$ is nonempty. On the other hand, let x_D be a closest point to x in D and define $\varphi : X \to \mathbb{R}$ by $\varphi(y) := \|y - x_D\|^2$. Since $\varphi - d_D^2$ attains its minimum at x, and φ is convex, continuous, and ∂-differentiable (because $\|.\|$ is ∂-smooth), we derive from properties (P2) and (P3) that $0 \in \partial\varphi\ (x)\ +\ \partial\ (-d_D^2)\ (x)$, proving the nonemptiness of $\partial(-d_D^2)\ (x)$. $\square$

THEOREM A (multidirectional mean value inequality). *Let $D = [a, C]$ with $a \in \operatorname{dom} f$, and let $r \in \mathbb{R}$ with $r \le \min_C f$. Then, there exist $0 \le \tau < 1$, $x_0 \in a + \tau(C - a)$, and sequences $(x_n) \in \operatorname{dom}\partial f$ and $x_n^* \in \partial f(x_n)$ such that*

(i) $x_n \to_f x_0$ *and* $f(x_0) - \min_D f \le |r - f(a)|$;

(ii) *for any* $0 \le t \le 1, \liminf_{n\to\infty} \inf_{x\in a+t(C-a)} \langle x_n^*, x - x_n\rangle \ge (t-\tau)(r - f(a))$;

(iii) $\liminf_{n\to\infty} \inf_{c\in C} \langle x_n^*, c - a\rangle \ge r - f(a)$.

Furthermore, if $f(a) = r = \min_D f$, we may choose $x_0 = a$, while if $f(a) \le r$ and $a \notin C$, we may choose $x_0 \notin C$.

THEOREM B (Fermat rule). *Assume that $C \cap \operatorname{dom} f \neq \emptyset$, and that $x_0 \in C$ verifies $f(x_0) = \min_C f$. Then, there exist sequences $(x_n) \subset \operatorname{dom}\partial f$ and $x_n^* \in \partial f(x_n)$ such that*

(i) $x_n \to_f x_0$;

(ii) $\liminf_{n\to\infty} \inf_{c\in C} \langle x_n^*, c - x_n\rangle \ge 0$;

(iii) $\liminf_{n\to\infty} \inf_{c\in C}\langle x_n^*, c - x_0\rangle \ge 0$.

THEOREM C (multidirectional Rolle-type inequality). *Let $D = [a, C]$ with $a \in X$. Assume that there exists $x_0 \in D \setminus C$ such that $f(x_0) = \min_D f$. Then, there exist sequences $(x_n) \subset \operatorname{dom}\partial f$ and $x_n^* \in \partial f(x_n)$ such that*

(i) $x_n \to_f x_0$;

(ii) $\liminf_{n\to\infty} \inf_{d\in D} \langle x_n^*, d - x_n\rangle \ge 0$;

(iii) $\liminf_{n\to\infty} \inf_{d\in D}\langle x_n^*, d - a\rangle \ge 0$.

We first observe that the above three theorems can be easily derived from each other:

A $\Rightarrow$ **B**. In Theorem A, let $a :=\ x_0$ (so that $D = C$) and $r :=\ f\ (a)\ =\ \min_D f$.

B $\Rightarrow$ **C**. Since $x_0 \in [a, C] \setminus C$, we have $x_0 - a = t(c - x_0)$ for some $t \ge 0$ and some $c \in C$, so that assertion (iii) of Theorem B gives

$$\liminf_{n\to\infty} \inf_{d\in D} \langle x_n^*, d - a\rangle = \liminf_{n\to\infty} \big(\inf_{d\in D}\langle x_n^*, d - x_0\rangle + t\langle x_n^*, c - x_0\rangle\big) \ge 0.$$

C $\Rightarrow$ **A**. Consider the $\mathcal{T}$-lower semicontinuous function $\varphi : X \times \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ given by

$$\varphi(x,t) = f(x) - (r - f(a))t,$$

and set $\tilde{a} := (a, 0)$, $\tilde{C} := C \times \{1\}$, and $\tilde{D} := [\tilde{a}, \tilde{C}]$. Since

$$\varphi(\tilde{a}) = f(a) \le f(a) + f(c) - r = \varphi(c,1) \quad \text{for all } c \in C,$$

there exists $\tilde{x}_0 \in \tilde{D} \setminus \tilde{C}$ such that $\varphi(\tilde{x}_0) = \min_{\tilde{D}} \varphi$. Because the space $X \times \mathbb{R}$ has a ∂-smooth renorm (Fact 1), we may apply Theorem C to find sequences $(\tilde{x}_n) \subset \operatorname{dom}\partial\varphi$ and $\tilde{x}_n^* \in \partial\varphi(\tilde{x}_n)$ such that

$$\tilde{x}_n \to_\varphi \tilde{x}_0\,, \tag{A.1}$$

$$\liminf_{n\to\infty} \inf_{\tilde{d}\in\tilde{D}} \langle \tilde{x}_n^*, \tilde{d} - \tilde{x}_n \rangle \geq 0\,, \tag{A.2}$$

$$\liminf_{n\to\infty} \inf_{\tilde{d}\in\tilde{D}} \langle \tilde{x}_n^*, \tilde{d} - \tilde{a} \rangle \geq 0\,. \tag{A.3}$$

Writing $\tilde{x}_0 = (x_0, \tau)$ with $\tau < 1$ and $\tilde{x}_n = (x_n, \tau_n)$, then $x_0 \in a + \tau(C - a)$, $\tau_n \to \tau$, and $x_n \to_f x_0$, because of (A.1). Moreover, for all $t \in [0, 1]$ and $x \in a + t(C - a)$, we have $\varphi(\tilde{x}_0) = f(x_0) - (r - f(a))\tau \leq \varphi(x, t) = f(x) - (r - f(a))t$, which gives $f(x_0) \leq f(x) + |r - f(a)|$, proving assertion (i). Now, according to property (P4), we can write $\tilde{x}_n^* = (x_n^*, -(r - f(a)))$ with $x_n^* \in \partial f(x_n)$. It is easily seen that assertions (ii) and (iii) follow from (A.2) and (A.3), respectively.

The last statements come from the fact that if $f(a) = r = \min_D f$, then

$$\varphi(a, 0) = f(a) \leq f(x) = \varphi(x, t)$$

for all $(x, t) \in \tilde{D}$, so we may choose $\tilde{x}_0 = \tilde{a}$, while if $f(a) \leq r$, then

$$\varphi(a, 0) = f(a) \leq f(c) - r + f(a) \leq f(c) - (r - f(a))t = \varphi(c, t)$$

for all $(c, t) \in C \times [0, 1]$; so if $\tilde{a} \notin C \times [0, 1]$, we may also choose $\tilde{x}_0 \notin C \times [0, 1]$.

To complete the proofs of our theorems, it thus suffices to give a direct proof of Theorem C. This goes exactly as in the case where $D = [a, c]$ is a compact segment; see [3, Theorem 4.1]. We briefly recall the arguments.

Direct proof of Theorem C. Let $\|\cdot\|$ be a ∂-smooth renorm on X and A be a closed neighborhood of D on which f is bounded below. For any natural number n such that $B_{1/n}(x_0)$ is contained in A, let $\gamma_n > 0$ be such that

$$f(x_0) < \inf_{B_{\gamma_n}(D)} f + 1/n^2.$$

Note that this is always possible since f is $\mathcal{T}$-lower semicontinuous and D is $\mathcal{T}$-compact. Then let $K_n > 0$ be such that

$$f(x_0) < \inf_A f + K_n \gamma_n^2 + 1/n^2,$$

and consider the function $f_n := f + K_n d_D^2$. Clearly,

$$f_n(x_0) = f(x_0) < \inf_A f_n + 1/n^2.$$

Applying [3, Theorem 3.1] to f_n with $\varepsilon := 1/n^2$ and $\lambda := 1/n$ produces a sequence $(x_n) \subset A$ such that

$$\|x_0 - x_n\| < 1/n,$$
$$f(x_n) \leq f_n(x_n) < f(x_0) + 1/n^2, \text{ and}$$
$$\partial f_n(x_n) \cap (2/n)B^* \neq \emptyset.$$

Since f is lower semicontinuous, the first two formulae show that $x_n \to_f x_0$. The function d_D^2 being ∂-differentiable (Fact 2), the third formula combined with (P3) shows that there exist $x_n^* \in \partial f(x_n)$, $\xi_n^* \in \partial d_D^2(x_n)$, and $\beta_n^* \in B^*$ with $x_n^* + K_n \xi_n^* = (2/n)\beta_n^*$.

From elementary subdifferential calculus of convex analysis, we infer that

$$\langle \xi_n^*, d - x_n \rangle \leq 0 \quad \text{for all} \quad d \in D, \tag{A.4}$$

$$\langle \xi_n^*, d - Px_n \rangle \leq 0 \quad \text{for all} \quad d \in D, \tag{A.5}$$

where $Px_n \in D$ is such that $\|x_n - Px_n\| = d_D(x_n)$. We get from (A.4) that $\langle x_n^*, d - x_n \rangle \geq (2/n)\langle \beta_n^*, d - x_n \rangle$, and assertion (ii) follows. Since $Px_n \to x_0$ and $x_0 \notin C$, we may assume that $Px_n \notin C$. Hence $Px_n - a = t_n(y_n - Px_n)$ for some $t_n \geq 0$ and $y_n \in C$. According to (A.5), for any $d \in D$ we have $\langle \xi_n^*, d - a \rangle = \langle \xi_n^*, d - Px_n \rangle + t_n \langle \xi_n^*, y_n - Px_n \rangle \leq 0$, which yields $\langle x_n^*, d - a \rangle \geq (2/n)\langle \beta_n^*, d - a \rangle$, and proves assertion (iii). □

Theorems A, B, and C are special cases of, respectively, Theorem 6.4, Corollary 6.2, and Corollary 5.3 given in the text. They are sufficient to obtain the results of Luc and of Deville and Ivanov mentioned in Remark 6.2. For various corollaries, please refer to the main text.

REFERENCES

[1] J.-P. Aubin and I. Ekeland, *Applied Nonlinear Analysis*, Wiley-Interscience, New York, 1984.

[2] D. Aussel, J.-N. Corvellec, and M. Lassonde, *Subdifferential characterization of quasiconvexity and convexity*, J. Convex Anal., 1 (1994), pp. 195–201.

[3] D. Aussel, J.-N. Corvellec, and M. Lassonde, *Mean value property and subdifferential criteria for lower semicontinuous functions*, Trans. Amer. Math. Soc., 347 (1995), pp. 4147–4161.

[4] J. M. Borwein and D. Preiss, *A smooth variational principle with applications to subdifferentiability and to differentiability of convex functions*, Trans. Amer. Math. Soc., 303 (1987), pp. 517–527.

[5] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley-Interscience, New York, 1983; reprinted as Classics Appl. Math. 5, SIAM, Philadelphia, PA, 1990.

[6] F. H. Clarke and Yu. S. Ledyaev, *Mean value inequalities in Hilbert space*, Trans. Amer. Math. Soc., 344 (1994), pp. 307–324.

[7] R. Correa, A. Jofré, and L. Thibault, *Subdifferential monotonicity as characterization of convex functions*, Numer. Funct. Anal. Optim., 15 (1994), pp. 531–535.

[8] R. Deville and E. M. El Haddad, *The subdifferential of the sum of two functions in Banach spaces,* I. *First order case*, J. Convex Anal., 3 (1996), pp. 295–308.

[9] R. Deville, G. Godefroy, and V. Zizler, *A smooth variational principle with applications to Hamilton-Jacobi equations in infinite dimensions*, J. Funct. Anal., 111 (1993), pp. 197–212.

[10] R. Deville and M. Ivanov, *Smooth variational principles with constraints*, Arch. Math., 69 (1997), pp. 418–426.

[11] I. Ekeland, *On the variational principle*, J. Math. Anal. Appl., 47 (1974), pp. 324–353.

[12] A. Ioffe, *Approximate subdifferentials and applications.* I: *The finite dimensional theory*, Trans. Amer. Math. Soc., 281 (1984), pp. 389–416.

[13] A. Ioffe and J.-P. Penot, *Subdifferentials of performance functions and calculus of coderivatives of set-valued mappings*, Serdica Math. J., 22 (1996), pp. 359–384.

[14] D. T. Luc, *A strong mean value theorem and applications*, Nonlinear Anal., 26 (1996), pp. 915–923.

[15] M. L. Radulescu and F. H. Clarke, *The multidirectional mean value theorem in Banach spaces*, Canad. Math. Bull., 40 (1997), pp. 88–102.

[16] L. Thibault and D. Zagrodny, *Integration of subdifferentials of lower semicontinuous functions on Banach spaces*, J. Math. Anal. Appl., 189 (1995), pp. 33–58.

[17] D. Zagrodny, *Approximate mean value theorem for upper subderivatives*, Nonlinear Anal., 12 (1988), pp. 1413–1428.

[18] Q. J. Zhu, *Clarke-Ledyaev mean value inequalities in smooth Banach spaces*, Nonlinear Anal., 32 (1998), pp. 315–324.

SIAM J. OPTIM.
Vol. 9, No. 3, pp. 707–728

HOMOGENEOUS ANALYTIC CENTER CUTTING PLANE METHODS FOR CONVEX PROBLEMS AND VARIATIONAL INEQUALITIES*

YU. NESTEROV[†] AND J.-PH. VIAL[‡]

Abstract. In this paper we consider a new analytic center cutting plane method in an extended space. We prove the efficiency estimates for the general scheme and show that these results can be used in the analysis of a feasibility problem, the variational inequality problem, and the problem of constrained minimization. Our analysis is valid even for problems whose solution belongs to the boundary of the domain.

Key words. cutting plane, analytic centers, convex problems, variational irregularities

AMS subject classifications. 65K05, 65K10

PII. S1052623497324813

1. Introduction.

Cutting plane methods are designed to solve convex problems with the following property. A so-called oracle provides first-order information in the form of cutting planes that separate the query point from the set of solutions. Given a sequence of query points, the oracle provides a set of cutting planes that generates a polyhedral relaxation of the solution set. As the sequence of query points increases, the relaxation becomes increasingly refined, until one obtains a solution to the original problem to the given degree of accuracy. In a worst-case analysis, one assumes that the oracle provides, each time, the least informative cutting plane. In that respect, the choice of some kind of center of the current relaxation is rather intuitive, as it should force the oracle to cut off at each iteration a significant part of the current relaxation. One can conceive of many possible centers, but the analytic center—a concept first introduced by Sonnevend [20][1]— is well adapted. Analytic centers underlie the theory of most interior point methods; their analytical properties are well studied, and there are powerful algorithms to compute them or to retrieve a new center after one side of the polyhedron has been shifted. We name this class of cutting plane method the analytic center cutting plane method (ACCPM).

Goffin, Haurie, and Vial [6] proposed the first ACCPM; they also provided some evidence of its practical efficiency. See also [3, 4, 5]. Atkinson and Vaidya [2] and Nesterov [15] gave the first complexity analysis of some closely related methods. The proof technique of [15] has been subsequently applied to analyzing the original ACCPM method for different problems: finding a point in a convex set [7, 8, 10], minimizing a convex function [1, 11], or solving a variational inequality problem [9]. An interesting extension of the method concerns the case where part of the information on the problem is given under the form of some self-concordant functions. By incor-

*Received by the editors July 18, 1997; accepted for publication (in revised form) June 23, 1998; published electronically May 21, 1999.

http://www.siam.org/journals/siopt/9-3/32481.html

[†] CORE, Catholic University of Louvain, 34 voie du Roman Pays, 1348 Louvain-la-Neuve, Belgium (nesterov@core.ucl.ac.be).

[‡] HEC/LOGILAB, University of Geneva, 102 bd Carl-Vogt, 1211-Geneva 4, Switzerland (jpvial@uni2a.unige.ch). The research of this author was partially supported by Fonds National Suisse grant 12-42503.94.

[1] In this paper, among many other applications, Sonnevend also mentioned the possibility of using the analytic center of a polytope in nondifferentiable optimization schemes.

porating this information directly into the algorithm, one can presumably enhance the practical convergence. This issue has been addressed by shifting nonlinear cuts, in the case of quadratic constraints [12] or a nonlinear objective [13].

All the quoted papers dealing with the complexity analysis, except [2], use an inequality due to Nesterov [15]. This inequality bounds the growth of the Hessian of the barrier function as the new cutting planes are added to the existing collection. This inequality is rather weak; besides, it involves the dimension n of the space in which the problem is posed. This introduces an undesirable factor n in the complexity analysis and thus calls for another approach to the problem.

In this paper, we propose an approach that circumvents some of the difficulties and shortcomings of the previous papers. The new cutting plane scheme offers the further advantage of a unified framework to deal with three different convex problems for which part of the information is given under the form of self-concordant functions. The main idea is to embed the original problem in an extended space and to apply a homogeneous ACCPM to this new conic formulation. This scheme uses a logarithmic barrier for the cutting planes, a ν-self-concordant barrier for the feasible set, and a proximal term. We present here an idealized version of the method based on the exact analytic center. Although it is not implementable, this idealized version is much simpler to analyze. However, let us mention that one can remove the assumption of an exact analytic center at the cost of some technicalities. In the idealized framework, we are able to derive, at any feasible point, a bound on the weighted average of the slacks to the cutting planes. We use this inequality in three different cases. We first apply it to the problem of finding a point in a closed convex set. We next deal with monotone variational inequalities over a bounded convex set. Finally, we consider the problem of minimizing a convex function over a compact convex set.

In the three cases, we can bound the number of iterations by a quantity of order $O(\frac{\exp(\sqrt{\nu})}{\varepsilon^2})$, where ν is the parameter of the self-concordant barrier for the explicit feasible set and ε is the required accuracy. For the simple feasibility problem, we can take $\nu = 2$. In the minimization of a convex function over a convex set, we can further improve the result to get $O(\frac{\nu}{\varepsilon^2})$. Note that the complexity estimate is independent of the dimension n of the underlying space. Thus, for large n our result is optimal (see [14]).

Notation. Given a symmetric positive definite matrix B, we define the norm

$$\| u \|_B = \langle Bu, u\rangle^{\frac{1}{2}}.$$

2. Homogeneous cutting plane scheme. In this section, we consider a homogeneous feasibility problem and propose a homogeneous cutting plane scheme to solve it. The feasibility problem of interest is

$$\text{find } x \in K \bigcap X^*, \qquad x \neq 0, \tag{2.1}$$

where K is a closed convex cone with nonempty interior and X^* is a closed convex cone. As will be shown in the next sections, the formulation of problem (2.1) is general enough to include a variety of convex problems of interest.

The natural black-box description of problem (2.1) can be produced using the concept of the homogeneous separation oracle.

DEFINITION 2.1. *A separation oracle is a mapping $g(x)$ such that $\langle g(x), x - x^*\rangle \geq 0$ for any $x^* \in X^*$. The oracle is homogeneous if $g(tx) = g(x)$ for any $x \in \operatorname{int} K$ and $t > 0$, and if $\langle g(x), x\rangle = 0$.*

Thus, in this paper, we always assume the following.

Assumption 2.2.

(1) *Problem* (2.1) *is endowed with a homogeneous separation oracle* $g(x)$, $x \in \operatorname{int} K$, *with* $\| g(x) \| = 1$.

(2) K *is equipped with a* ν*-normal barrier* $F(x)$.

We recall that a ν-normal barrier $F(x)$ is a convex function

$$F(y) \geq F(x) + \langle F'(x), y - x \rangle, \tag{2.2}$$

which is self-concordant and ν-logarithmically homogeneous:

$$F(\tau x) = F(x) - \nu \ln \tau, \qquad x \in \operatorname{int} K, \qquad \tau > 0. \tag{2.3}$$

(See Definition 2.3.2 in [17, p. 40].) Since, for logarithmically homogeneous functions, we have $F'(\tau x) = \frac{1}{\tau} F'(x)$, the convexity condition (2.2) can be written in a stronger form:

$$F(y) \geq F(x) - \nu \ln \left[\frac{1}{\nu} \langle -F'(x), y \rangle \right]. \tag{2.4}$$

(To see this, replace x by τx in (2.2) and find the maximum of the right-hand side in $\tau > 0$.)

The homogeneous cutting plane scheme can be briefly described as follows:

$$\begin{array}{ll} (0) & \text{Set } F_0(x) = \frac{\rho}{2} \| x \|^2 + F(x). \\ (1) & k\text{th iteration } (k \geq 0): \\ & \text{(a) compute } x_k = \arg\min\limits_x F_k(x); \\ & \text{(b) set } F_{k+1}(x) = F_k(x) - \ln\langle g(x_k), x_k - x \rangle. \qquad \Box \end{array} \tag{2.5}$$

The coefficient ρ must be positive, but it is otherwise arbitrary. Indeed, the aim of the proximal term is just to normalize the iterates. We prove this statement at the end of the section. Below, we refer to this x_k as the proximal analytic center generated by $F_k(x)$ (or simply as the analytic center).

As stated above, the cutting plane method assumes that the exact minimizer x_k of the function $F_k(x)$ can be computed. In practice, one can only compute approximate minimizers satisfying the usual proximity condition

$$\| F_k'(x_k) \|_{[F_k''(x_k)]^{-1}} \leq \eta < \frac{3 - \sqrt{5}}{2}.$$

Following [15], it is possible to carry out the analysis under this milder hypothesis at the cost of greater technicalities. The issues are then twofold. First, it must be checked that an appropriate choice of η allows us to compute an approximate minimizer in a bounded number of Newton steps after adding a cutting plane. Next, any time we use the first-order optimality condition associated with an exact minimizer, we should replace the equation by an appropriate inequality. Similarly, the inequality on the potential change that is used in the proofs should be weakened appropriately. We shall not perform the detailed analysis in this paper.

Let us write the extensive form of function F_k*:*

$$F_k(x) = \frac{\rho}{2} \| x \|^2 + F(x) - \sum_{i=0}^{k-1} \ln\langle g(x_i), x_i - x \rangle.$$

The first-order optimality condition for the minimizer x_k is

$$\rho x_k + F'(x_k) = -\sum_{i=0}^{k-1} \frac{g(x_i)}{\langle g(x_i), x_i - x_k\rangle}. \tag{2.6}$$

LEMMA 2.3. *The equation $\rho \parallel x_k \parallel^2 = k + \nu$ holds for all $k = 0, 1, \ldots$.*

Proof. Since F is logarithmically homogeneous, one has (see Theorem 2.3.13 of [17])

$$\langle F'(x_k), x_k\rangle = -\nu.$$

Using the first-order optimality condition (2.6), we get

$$\rho \parallel x_k \parallel^2 = k - \langle F'(x_k), x_k\rangle = k + \nu,$$

since $\langle g(x_i), x_i\rangle = 0$. □

In what follows, we shall use the notation

$$\lambda_{ik} = \frac{1}{\langle g(x_i), x_i - x_k\rangle} > 0 \qquad \text{for} \quad 0 \leq i < k-1 \qquad \text{and} \quad S_k = \sum_{i=0}^{k-1} \lambda_{ik}.$$

Let $x \in K$. We introduce the quantity

$$\mu_k(x) = \frac{1}{S_k} \sum_{i=0}^{k-1} \lambda_{ik} \langle g(x_i), x_i - x\rangle,$$

which will play a central role in the analysis. Note that $\mu_k(x)$ is the weighted average of the slacks at x in the cutting plane inequalities.

LEMMA 2.4. *For all $x \in K$,*

$$\mu_k(x) \leq \sqrt{\rho} \frac{\sqrt{\nu + k}}{S_k} \parallel x \parallel. \tag{2.7}$$

Proof. Multiplying (2.6) by x and using $\langle g(x_i), x_i\rangle = 0$, we get

$$\rho\langle x_k, x\rangle + \langle F'(x_k), x\rangle = \sum_{i=0}^{k-1} \lambda_{ik} \langle g(x_i), x_i - x\rangle = S_k \mu_k(x). \tag{2.8}$$

In view of Corollary 2.3.1 of [17, p. 39], we have, for all $x \in K$,

$$\langle F'(x_k), x\rangle \leq 0.$$

Consequently,

$$S_k \mu_k(x) \leq \rho\langle x_k, x\rangle \leq \rho \parallel x_k \parallel \cdot \parallel x \parallel = \sqrt{\rho}\sqrt{\nu + k} \parallel x \parallel .$$

The lemma is proved. □

Lemma 2.4 calls for an analysis of the behavior of S_k as k increases. This behavior is closely related to the potential values $F_k(x_k)$. Denote $F_k^* = F_k(x_k)$, where $x_k = \arg\min_x F_k(x)$. We also define the constants

$$\theta_1 = \tfrac{1}{2}(\sqrt{5} - 1) - \ln \tfrac{\sqrt{5}+1}{2} > 0 \quad \text{and} \quad \theta_2 = \tfrac{\sqrt{5}+1}{2}.$$

Lemma 2.5. *For any $k \geq 0$,*

$$\sqrt{\rho} \parallel x_{k+1} - x_k \parallel \leq \parallel x_{k+1} - x_k \parallel_{F''_k(x_{k+1})} \leq \theta_2$$

and

$$F^*_{k+1} \geq F^*_k + \theta_1 - \ln \frac{\theta_2}{\sqrt{\rho}}.$$

Proof. The first-order optimality conditions on $F_{k+1}(x) = F_k(x) - \ln\langle g(x_k), x_k - x\rangle$ at x_{k+1} are

$$F'_{k+1}(x_{k+1}) = F'_k(x_{k+1}) + \frac{g(x_k)}{\langle g(x_k), x_k - x_{k+1}\rangle} = 0.$$

Multiplying by $x_k - x_{k+1}$ and using $F'_k(x_k) = 0$ yields

$$\langle F'_k(x_{k+1}) - F'_k(x_k), x_{k+1} - x_k\rangle = 1. \tag{2.9}$$

From (2.9) and Lemma A.6 in the appendix, we get

$$1 = \langle F'_k(x_{k+1}) - F'_k(x_k), x_{k+1} - x_k\rangle \geq \frac{\parallel x_k - x_{k+1} \parallel^2_{F''_k(x_{k+1})}}{1 + \parallel x_k - x_{k+1} \parallel_{F''_k(x_{k+1})}}. \tag{2.10}$$

Therefore,

$$\parallel x_k - x_{k+1} \parallel_{F''_k(x_{k+1})} \leq \tfrac{\sqrt{5}+1}{2} = \theta_2. \tag{2.11}$$

Since $\rho I \preceq F''_k$, we get

$$\sqrt{\rho} \parallel x_k - x_{k+1} \parallel \leq \parallel x_k - x_{k+1} \parallel_{F''_k(x_{k+1})} \leq \theta_2,$$

hence the bound on $\parallel x_{k+1} - x_k \parallel$.

To prove the second part of the theorem, we first show that $(\sqrt{5}-1)/2$ is a lower bound for $\parallel x_k - x_{k+1} \parallel_{F''_k(x_k)}$. Indeed, assume $\parallel x_k - x_{k+1} \parallel_{F''_k(x_k)} \leq (\sqrt{5}-1)/2 < 1$. Since $F_k(x)$ is a self-concordant function, in view of Lemma A.7 in the appendix, we have

$$1 = \langle F'_k(x_{k+1}) - F'_k(x_k), x_{k+1} - x_k\rangle \leq \frac{\parallel x_k - x_{k+1} \parallel^2_{F''_k(x_k)}}{1 - \parallel x_k - x_{k+1} \parallel_{F''_k(x_k)}}.$$

Hence,

$$\parallel x_k - x_{k+1} \parallel_{F''_k(x_k)} \geq \tfrac{\sqrt{5}-1}{2} \tag{2.12}$$

always holds. By Lemma A.6 in the appendix and with $F'_k(x_k) = 0$, we get

$$F_k(x_{k+1}) \geq F^*_k + \omega(\parallel x_k - x_{k+1} \parallel_{F''_k(x_k)}),$$

where $\omega(t) = t - \ln(1+t)$. Hence,

$$F_k(x_{k+1}) \geq F^*_k + \tfrac{\sqrt{5}-1}{2} - \ln \tfrac{\sqrt{5}+1}{2} = F^*_k + \theta_1.$$

Finally, from the definition of F_{k+1} and $\| g(x_k) \|= 1$, we have

$$\begin{aligned} F^*_{k+1} = F_{k+1}(x_{k+1}) &= F_k(x_{k+1}) - \ln\langle g(x_k), x_k - x_{k+1}\rangle \\ &\geq F^*_k + \theta_1 - \ln \| x_k - x_{k+1} \| \qquad (2.13) \\ &\geq F^*_k + \theta_1 - \ln \frac{\theta_2}{\sqrt{\rho}}. \qquad (2.14) \end{aligned}$$

□

The next lemma gives a lower bound for S_k.

LEMMA 2.6. *For any $k \geq 0$ we have*

$$S_k \geq \theta_3 k\sqrt{\rho} \cdot \exp\left\{ \frac{1}{k}(F(x_0) - F(x_k)) \right\},$$

where $\theta_3 = \frac{1}{\theta_2} \exp(\theta_1 - \frac{1}{2})$.

Proof. Using the inequality between the arithmetic and the geometric means, we have

$$S_k = \sum_{i=0}^{k-1} \frac{1}{\langle g(x_i), x_i - x_k\rangle} \geq k \exp\left\{ \frac{1}{k} \sum_{i=0}^{k-1} \ln \frac{1}{\langle g(x_i), x_i - x_k\rangle} \right\}.$$

From the definition of F_k and Lemmas 2.3 and 2.5,

$$\begin{aligned} \sum_{i=0}^{k-1} \ln \frac{1}{\langle g(x_i), x_i - x_k\rangle} &= -\frac{\rho}{2} \| x_k \|^2 - F(x_k) + F_k(x_k) \\ &\geq \left(-\frac{\nu + k}{2} - F(x_k) \right) + \left(k\left(\theta_1 - \ln \frac{\theta_2}{\sqrt{\rho}}\right) + F(x_0) + \frac{\nu}{2} \right) \\ &= k\left(\theta_1 - \frac{1}{2} - \ln \frac{\theta_2}{\sqrt{\rho}}\right) + F(x_0) - F(x_k). \end{aligned}$$

Hence,

$$S_k \geq \frac{k\sqrt{\rho}}{\theta_2} \exp\left\{\theta_1 - \frac{1}{2}\right\} \cdot \exp\left\{ \frac{1}{k}(F(x_0) - F(x_k)) \right\}. \qquad \square$$

We now state the main result of this section.

THEOREM 2.7. *For any $x \in K$,*

$$\mu_k(x) \leq \frac{\sqrt{k+\nu}}{k\theta_3} \exp\left\{ \frac{1}{k}\Big(F(x_k) - F(x_0)\Big) \right\} \| x \| . \qquad (2.15)$$

Besides,

$$F(x_k) - F(x_0) \leq k\sqrt{\nu}\theta_2. \qquad (2.16)$$

Proof. Combining inequality (2.7) with Lemma 2.6 yields the first inequality. To prove the second inequality, we use

$$\begin{aligned} F(x_{k+1}) - F(x_k) &\leq \langle F'(x_{k+1}), x_{k+1} - x_k\rangle \\ &\leq \| F'(x_{k+1}) \|_{[F''(x_{k+1})]^{-1}} \cdot \| x_{k+1} - x_k \|_{F''(x_{k+1})} . \end{aligned}$$

Recall that $F'' \preceq F''_k$. Hence,

$$\| x_{k+1} - x_k \|_{F''(x_{k+1})} \leq \| x_{k+1} - x_k \|_{F''_k(x_{k+1})} .$$

By Lemma 2.5, $\| x_{k+1} - x_k \|_{F''_k(x_{k+1})} \leq \theta_2$. On the other hand, F is a ν-self-concordant barrier; thus

$$\| F'(x_{k+1}) \|_{[F''(x_{k+1})]^{-1}} \leq \sqrt{\nu}.$$

The theorem is proved. □

By inserting (2.16) into (2.15), we get the general bound

$$\mu_k(x) \leq \frac{\sqrt{k+\nu}}{k\theta_3} e^{\theta_2\sqrt{\nu}} \| x \|. \tag{2.17}$$

This inequality is useful if the parameter ν is small. For large values of ν, the exponential term on the right-hand side considerably weakens the bound on μ_k.

To conclude this section, we prove that the proximal term is just a convenient way to normalize the iterates.

LEMMA 2.8. *Let $\rho_j > 0$, $j = 1, 2$, be two arbitrary positive numbers. Let $\{x_i^j\}_{i=0}^k$, $j = 1, 2$, be the sequences of proximal analytic centers generated by the standard scheme with $\rho = \rho_j$, $j = 1, 2$, respectively. Then, $x_i^2 = \tau x_i^1$ for all $i = 0, 1, \ldots, k$, with $\tau = \sqrt{\frac{\rho_1}{\rho_2}}$.*

Proof. Consider the sequence $\{\hat{x}_i = \tau x_i^1\}_{i=0}^k$, where $\tau = \sqrt{\frac{\rho_1}{\rho_2}}$. Define the sequence of potentials

$$\hat{F}_k(x) = \frac{\rho_2}{2} \| x \|^2 + F(x) - \sum_{i=0}^{k-1} \ln\langle g(\hat{x}_i), \hat{x}_i - x\rangle.$$

Taking the derivative of $\hat{F}_k$ at $\hat{x}_k$ and using the fact that g is homogeneous and F is a ν-normal barrier, we get

$$\begin{aligned}
\hat{F}'_k(\hat{x}_k) &= \rho_2 \hat{x}_k + F'(\hat{x}_k) + \sum_{i=0}^{k-1} \frac{g(\hat{x}_i)}{\langle g(\hat{x}_i), \hat{x}_i - \hat{x}_k\rangle} \\
&= \rho_2 \tau x_k^1 + \frac{1}{\tau} F'(x_k^1) + \sum_{i=0}^{k-1} \frac{g(x_i^1)}{\langle g(x_i^1), \tau(x_i^1 - x_k^1)\rangle} \\
&= \frac{1}{\tau}\left\{ \rho_1 x_k^1 + F'(x_k^1) + \sum_{i=0}^{k-1} \frac{g(x_i^1)}{\langle g(x_i^1), x_i^1 - x_k^1\rangle} \right\} = 0.
\end{aligned}$$

Hence, $\{\hat{x}_i\}_{i=0}^k$ coincides with the sequence of proximal analytic centers $\{x_i^2\}_{i=0}^k$ generated by the algorithm with $\rho = \rho_2$. □

Since ρ is arbitrary, we may well choose $\rho = \nu$. By Lemma 2.3, this choice implies $\| x_0 \| = 1$. For the sake of simpler formulas, we shall assume throughout the rest of the paper that $\rho = \nu$, and thus $\| x_0 \| = 1$.

3. Convex feasibility problems.

Suppose we are given the following convex feasibility problem:

$$\text{find} \quad y \in Y^*,$$

where Y^* is a closed bounded convex set with a nonempty interior. Thus there are constants ε and R and a point $\bar{y} \in Y^*$ such that

$$B(\bar{y}, \varepsilon) \subset Y^* \subset B(0, R),$$

where $B(\bar{y}, \varepsilon)$ is the Euclidean ball centered at $\bar{y}$ with radius ε. (The assumption does not imply that $\bar{y}$ is known, only that it exists.) We assume that ε and R are known constants. Finally, we assume that for any $\hat{y} \notin Y^*$ a separation oracle returns a vector $h(\hat{y})$:

$$\langle h(\hat{y}), \hat{y} - y \rangle \geq 0 \quad \text{for all} \quad y \in Y^*,$$

with $\| \; h(\hat{y}) \; \|= 1$; if $y \in Y^*$, the oracle confirms that a solution has been found. Note that for any $\hat{y} \notin Y^*$ the point $\bar{y} + \epsilon h(\hat{y})$ belongs to Y^*. Therefore, we have the following inequality:

$$0 \leq \langle h(\hat{y}), \hat{y} - (\bar{y} + \epsilon h(\hat{y})) \rangle = \langle h(\hat{y}), \hat{y} - \bar{y} \rangle - \epsilon. \tag{3.1}$$

To embed the problem in an extended space, we define

$$X^* = \{ x = (y, t) \mid y = ty^*, \, y^* \in Y^*, \, t > 0 \}.$$

We also define the cone

$$K = \{ x = (y, t) \mid y = t\bar{y}, \, \| \; \bar{y} \; \| \leq R, \, t > 0 \},$$

and the associated barrier

$$F(x) = -\ln \left(t^2 R^2 - \| \; y \; \|^2 \right).$$

This barrier is ν-normal, with parameter $\nu = 2$.

Let us construct the separation oracle for X^*. Assume $x = (y, t) \notin X^*$, with $t > 0$; then, $\langle h(\frac{y}{t}), \frac{y}{t} - y^* \rangle \geq 0$ for all $y^* \in Y^*$. Let

$$\hat{g}(x) = \left(h \left(\frac{y}{t} \right), - \left\langle h \left(\frac{y}{t} \right) \frac{y}{t} \right\rangle \right)$$

and

$$g(x) = \frac{\hat{g}(x)}{\| \; \hat{g}(x) \; \|}.$$

Note that for all $x \in K$, we have $\| \; \hat{g}(x) \; \| \leq \sqrt{1 + R^2}$. From this definition we check that g is a homogeneous separation oracle, with $\| \; g(x) \; \|= 1$, $\langle g(x), x \rangle = 0$, and $g(\tau x) = g(x)$ for all $\tau > 0$.

The convex feasibility problem is now embedded in a homogeneous problem of the form (2.1). We can apply the homogeneous cutting plane algorithm with a stopping criterion. Assume that $x \notin X^*$, i.e., that $y/t \notin Y^*$. Let $\bar{x} = (\bar{y}, 1)$. In view of (3.1), we get

$$\langle \hat{g}(x), x - \bar{x} \rangle = -\langle \hat{g}(x), \bar{x} \rangle = - \left\langle h \left(\frac{y}{t} \right), \bar{y} \right\rangle + \left\langle h \left(\frac{y}{t} \right), \left(\frac{y}{t} \right) \right\rangle \geq \varepsilon.$$

Hence,

$$\langle g(x), x - \bar{x} \rangle \geq \frac{\varepsilon}{\| \; \hat{g}(x) \; \|} \geq \frac{\varepsilon}{\sqrt{1 + R^2}}.$$

At the kth iteration of the cutting plane algorithm, either the algorithm stops with $x_k \in X^*$ or, by (2.17),

$$\frac{\varepsilon}{\sqrt{1+R^2}} \leq \min_i \langle g(x_i), x_i - \bar{x} \rangle \leq \mu_k(\bar{x}) \leq \frac{\sqrt{k+2}}{k\theta_3} e^{\theta_2\sqrt{2}} \parallel \bar{x} \parallel$$
$$\leq \frac{\sqrt{k+2}}{k\theta_3} e^{\theta_2\sqrt{2}} \sqrt{1+R^2}.$$

This inequality implies that the iteration number is bounded by

$$k \leq \frac{(1+R^2)^2}{\varepsilon^2} M, \qquad \text{with} \quad M = 3\left(\frac{e^{\theta_2\sqrt{2}}}{\theta_3}\right)^2.$$

To conclude this section, we point out that the unconstrained minimization of a nondifferentiable convex function f with a known optimal value of the function $f^* = f(y^*)$ can be converted into a simple convex feasibility problem. Indeed, the level set $\{y \mid f(y) \leq f^* + \eta\}$ contains the ball $B(y^*, \frac{\eta}{L})$, where L is the Lipschitz constant for $f(x)$. However, section 5 provides a sharper analysis for minimization problems.

4. Variational inequalities.

Let $H(y)$ be a multivalued operator defined on a closed bounded set Q. We associate with it the variational inequality problem

$$\text{(4.1)} \quad \text{find } y^* \in Q: \quad \langle h_y, y - y^* \rangle \geq 0 \quad \text{for all} \quad y \in Q \quad \text{and all} \quad h_y \in H(y).$$

Such a y^* is called a weak solution to the variational inequality problem. (For a discussion relating strong and weak solutions, see [17].)

ASSUMPTION 4.1.

(1) *Q is bounded and R is a constant such that, for all $y \in Q$, $\parallel y \parallel \leq R$.*

(2) *The mapping H is uniformly bounded on Q and is monotone; i.e., $\parallel h_y \parallel \leq L$ for all $y \in Q$, and*

$$\langle h_u - h_y, u - y \rangle \geq 0$$

for all u, $y \in Q$ and any $h_u \in H(u)$, $h_y \in H(y)$.

Denote by Y^* the set of solutions to (4.1). For practical purposes, we need to enlarge this definition to include approximate solutions. To this end, we introduce the so-called *gap function*

$$\phi(y) = \max_{u \in Q} \{\langle h_u, y - u \rangle \mid \quad h_u \in H(u)\}.$$

Clearly, $\phi(y)$ is a closed convex function, which is strictly positive for all $y \in Q \setminus Y^*$ and $\phi(y) = 0$ for all $y \in Y^*$. Given $\epsilon > 0$, we define an ϵ-approximate solution $\bar{y}$ by $\phi(\bar{y}) \leq \epsilon$. In what follows we will give a complexity estimate for finding an ϵ-approximate solution. Finally, we observe that there is an obvious separation oracle for Y^*. Given $u \in Q$ and $h_u \in H(u)$, the following inequality holds:

$$\langle h_u, u - y \rangle \geq 0 \quad \text{for all} \quad y \in Y^*.$$

Embedding in an extended space. Let us transform the problem (4.1) into a conic form. To this end let us introduce a projective variable $t > 0$ and set

$$K = \left\{ x = (y, t) \;\mid\; t > 0,\ \frac{y}{t} \in Q \right\}.$$

Denote by $F(x)$ a ν-self-concordant barrier for the cone K. It is known that any ν-self-concordant barrier $H(y)$ for Q can be transformed into a self-concordant barrier for K (see [17]):

$$F(x) = c_1 H\left(\frac{y}{t}\right) - c_2 \nu \ln t,$$

where c_i are some absolute constants. However, the theoretical values of c_i are rather large. Therefore, in some particular cases it is reasonable to find a self-concordant barrier directly for the cone K. For example, if

$$Q = \{y \;\mid\; \langle a_i, y\rangle \le b_i,\ i = 1 \dots m\},$$

then

$$K = \{x = (y, t) \;\mid\; \langle a_i, y\rangle \le t b_i,\ i = 1 \dots m,\ t \ge 0\},$$

and we can use the logarithmic barrier

$$F(x) = -\sum_{i=1}^{m} \ln(t b_i - \langle a_i, y\rangle) - \ln t, \qquad \nu = m + 1.$$

If Q is a set defined by convex quadratic inequalities,

$$Q = \{y \;\mid\; \langle A_i y, y\rangle + \langle a_i, y\rangle \le b_i,\ i = 1 \dots m\},$$

then

$$K = \{x = (y, t) \;\mid\; \langle A_i y, y\rangle \le t(t b_i - \langle a_i, y\rangle),\ i = 1 \dots m,\ t \ge 0\},$$

and we can use the self-concordant barrier

$$F(x) = -\sum_{i=1}^{m} \ln[t(t b_i - \langle a_i, y\rangle) - \langle A_i y, y\rangle], \qquad \nu = 2m.$$

Note that in both cases the barrier can be represented as a restriction of a *self-scaled barrier* [19, 18] onto an affine hyperplane. For the first example this transformation is straightforward. For the second one, it proceeds as follows:

$$K_i = \{z = (y, t, \xi_i, \tau_i) \mid \langle A_i y, y\rangle + \xi_i^2 \le \tau_i^2,\ t \ge 0\},$$

with

$$\xi_i = \tfrac{1}{2}[t - (t b_i - \langle a_i, y\rangle)] \quad \text{and} \quad \tau_i = \tfrac{1}{2}[t + (t b_i - \langle a_i, y\rangle)].$$

The self-scaled property will be used in section 5 only.

Our problem in the extended space is to approximate, in the sense made precise earlier, a point from the intersection $K \bigcap X^*$, with

$$X^* = \{x = (y, t) \;\mid\; y = t y^*,\ y^* \in Y^*,\ t \ge 0\}.$$

Without loss of generality, we can put forth the following assumption.

ASSUMPTION 4.2. *The set Q contains the origin,* $0 \in Q$, *and* $F'_y(0, t) = 0$.

In other words, $y_0 = 0$ is the analytic center of the set Q. Then, $x_0 = (0, t_0)$ with $t_0 = 1$, since $\| x_0 \| = 1$ (see Lemma 2.3). Note that $F'_y(0, t) = 0$ implies that $0 \in \operatorname{int} Q$.

Separation oracle. Let us construct for problem (4.1) a separation oracle satisfying the assumptions of problem (2.1). For any $x = (y, t) \in \operatorname{int} K$, define $y(x) = y/t \in Q$ and

$$\hat{g}(x) = \left(h_{y(x)}, -\langle h_{y(x)}, y(x)\rangle\right).$$

Then, for any $x \in \operatorname{int} K$, we have $\langle \hat{g}(x), x\rangle = 0$ and $\hat{g}(\tau x) = \hat{g}(x)$ for $\tau > 0$.

The oracle enjoys a simple property that will prove useful in the analysis.

LEMMA 4.3. *Let $\bar{x} = (\bar{y}, \bar{\tau}) \in K$. Then,*

$$\langle \hat{g}(x), x - \bar{x}\rangle = \bar{\tau}\langle h_{y(x)}, y(x) - y(\bar{x})\rangle. \tag{4.2}$$

Proof. The proof is a direct consequence of the definition of $\hat{g}$:

$$\begin{aligned}\langle \hat{g}(x), x - \bar{x}\rangle &= -\langle \hat{g}(x), \bar{x}\rangle = -\langle h_{y(x)}, \bar{y}\rangle + \bar{\tau}\langle h_{y(x)}, y(x)\rangle \\ &= \bar{\tau}\langle h_{y(x)}, y(x) - y(\bar{x})\rangle. \qquad \square\end{aligned}$$

The lemma will often be used with $\bar{\tau} = 1$. Note also that, for $x^* \in X^*$, we have

$$\langle \hat{g}(x), x - x^*\rangle \geq 0.$$

Finally, let us define $g(x) = \hat{g}(x)/ \parallel \hat{g}(x) \parallel$. Note that

$$\parallel \hat{g}(x) \parallel = \left[\parallel h_{y(x)} \parallel^2 + \langle h_{y(x)}, y(x)\rangle^2\right]^{\frac{1}{2}} \leq L\sqrt{1 + R^2}. \tag{4.3}$$

Complexity estimate. Assume $\{x_i\}_{i=0}^{\infty}$ is a sequence generated by the algorithm. Define

$$\pi_{ik} = \frac{\lambda_{ik}}{\parallel \hat{g}(x_i) \parallel}, \qquad P_k = \sum_{i=0}^{k-1} \pi_{ik},$$

and

$$\bar{y}_k = \frac{1}{P_k}\sum_{i=0}^{k-1} \pi_{ik} y(x_i).$$

Let $u \in Q$ and $h_u \in H(u)$. Since H is monotone,

$$\begin{aligned}\langle h_u, \bar{y}_k - u\rangle &= \frac{1}{P_k}\sum_{i=0}^{k-1} \pi_{ik}\langle h_u, y(x_i) - u\rangle \\ &\leq \frac{1}{P_k}\sum_{i=0}^{k-1} \pi_{ik}\langle h_{y(x_i)}, y(x_i) - u\rangle. \end{aligned} \tag{4.4}$$

Let $v = (u, 1)$. In view of Lemma 4.3, we have

$$\begin{aligned}\frac{1}{P_k}\sum_{i=0}^{k-1} \pi_{ik}\langle h_{y(x_i)}, y(x_i) - u\rangle &= \frac{1}{P_k}\sum_{i=0}^{k-1} \pi_{ik}\langle \hat{g}(x_i), x_i - v\rangle \\ &= \frac{1}{P_k}\sum_{i=0}^{k-1} \lambda_{ik}\langle g(x_i), x_i - v\rangle = \frac{S_k}{P_k}\mu_k(v).\end{aligned}$$

Note that

$$P_k = \sum_{i=0}^{k-1} \frac{\lambda_{ik}}{\parallel \hat{g}(x_i) \parallel} \geq \frac{S_k}{L\sqrt{1+R^2}} \tag{4.5}$$

and $\parallel v \parallel \leq \sqrt{1+R^2}$. Therefore, in view of inequality (2.17), we get the following bound:

$$\begin{aligned}\phi(\bar{y}_k) &\leq \frac{S_k}{P_k} \max_v \{\mu_k(v) \mid v = (u,1) \in K\} \\ &\leq \frac{S_k}{P_k} \cdot \frac{\sqrt{k+\nu}}{k\theta_3} e^{\theta_2\sqrt{\nu}} \sqrt{1+R^2} \leq \frac{\sqrt{k+\nu}}{k\theta_3} e^{\theta_2\sqrt{\nu}} L(1+R^2).\end{aligned}$$

Thus, we have proved the following theorem.

THEOREM 4.4. *The proximal analytic center cutting plane method yields an ϵ-approximate solution for problem* (4.1) *after k iterations, with k satisfying*

$$\frac{k}{\sqrt{k+\nu}} \leq \frac{L(1+R^2)}{\epsilon\theta_3} e^{\theta_2\sqrt{\nu}}.$$

The result of Theorem 4.4 exhibits a quadratic dependence on R. For this problem class, we should rather expect a linear dependence. We shall show in the next section that a proper scaling of the variable y may restore this property.

To illustrate the necessity to use the average of the sequence of iterates and not the last analytic center, we apply the homogeneous cutting plane algorithm to the operator

$$h(y_1, y_2) = \begin{pmatrix} -y_2 \\ y_1 \end{pmatrix}$$

on the box

$$-1 \leq y_1 \leq 2 \quad \text{and} \quad -1 \leq y_2 \leq 1.$$

Figure 4.1 plots the iterates of the homogeneous cutting plane method when projected back into the original affine space. This same picture also shows the plot of the candidate solution $\bar{y}$. Clearly, the sequence of analytic centers does not converge to the solution point $(0,0)$, while the sequence of candidate solutions does.

Figure 4.2 displays the evolution of the upper bound for the gap function. This bound is computed from (4.4). More precisely, we solve the problem

$$\max \left\{ \frac{1}{P_k} \sum_{i=0}^{k-1} \pi_{ik} \langle h_{y(x_i)}, y(x_i) - u \rangle \mid u \in Q \right\}.$$

Note that the objective is linear. Thus the solution lies at one of the corners of the box. We hope that the procedure can be extended to more complex examples, thus providing a practical stopping criterion for an implementation of the algorithm. Note that the decrease of the bound is linear, a fact that seems to be quite typical of analytic center cutting plane schemes. Of course, no conclusion on the actual behavior of the method should be drawn from this simplistic example.

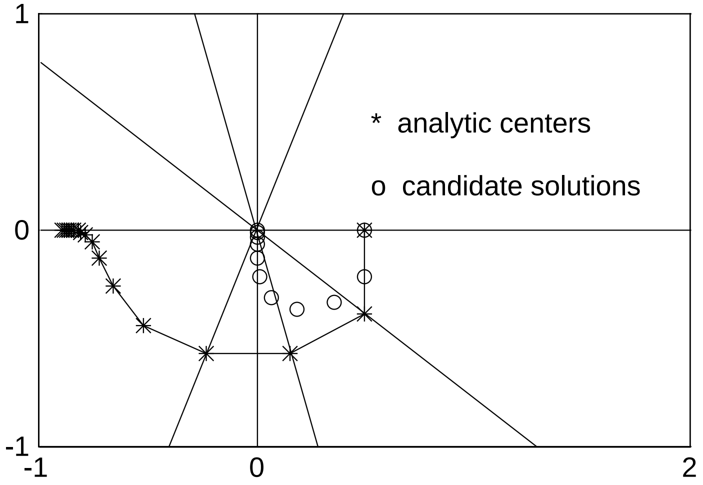


FIG. 4.1. *Iterates in the affine space.*

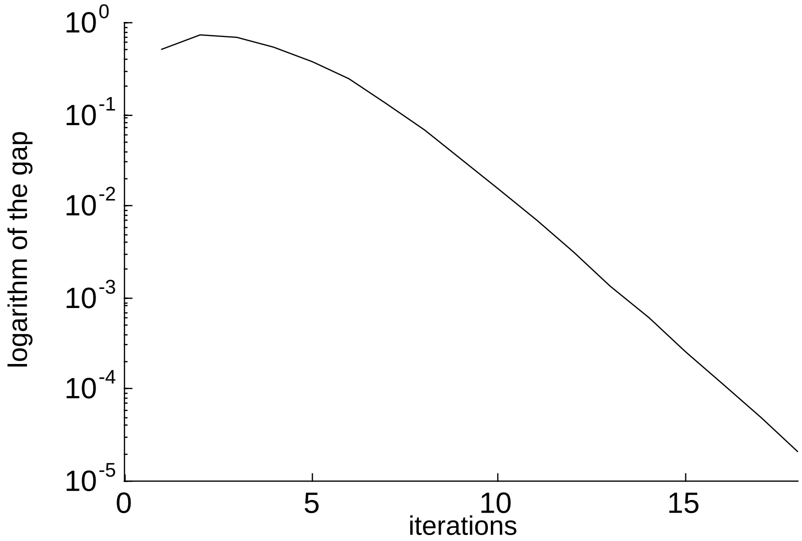


FIG. 4.2. *Bound for the gap function.*

5. Nonsmooth constrained minimization. Consider the problem

$$\min\{f(y) \mid \; y \in Q\}, \tag{5.1}$$

where Q is a closed bounded convex set with nonempty interior and the function $f(y)$ is convex and subdifferentiable on some open convex set containing Q. Then the subgradients of $f(y)$ are uniformly bounded on Q by some constant L. Denote by R any constant such that $\parallel y \parallel \leq R$ for all $y \in Q$ and by Y^* the set of the optimal solutions to (5.1).

Using the same argumentation as in section 4 we can embed the problem (5.1) into a conic form and provide it with a separation oracle satisfying the assumptions of the problem (2.1). To this end let us introduce a projective variable $t > 0$ and set

$$K = \left\{ x = (y, t) \ \mid \ t > 0, \ \frac{y}{t} \in Q \right\}.$$

Denote by $F(x)$ a ν-self-concordant barrier for the cone K. Just as before, we assume that $0 \in Q$ and $F'_y(0, t) = 0$. Then $x_0 = (0, t_0)$ with $t_0 = 1$, since $\| x_0 \| = 1$. Thus, our problem is to approximate a point from the intersection $K \bigcap X^*$, with

$$X^* = \{ x = (y, t) \ \mid \ y = t y^*, \ y^* \in Y^*, \ t \geq 0 \}.$$

The separation oracle for X^* can be defined as follows. Let $x = (y, t) \in \operatorname{int} K$. Define $y(x) = x/t \in Q$ and

$$\hat{g}(x) = \left(f'(y(x)), -\langle f'(y(x)), y(x) \rangle \right),$$

where $f'(u)$ is a subgradient of $f(u)$ at $u \in Q$. Then, for any $x \in K$, we have $\langle \hat{g}(x), x \rangle = 0$ and $\hat{g}(\tau x) = \hat{g}(x)$ for $\tau > 0$. Moreover, as in section 4, we have

$$\langle \hat{g}(x), x - \bar{x} \rangle = \bar{\tau} \langle f'(y(x)), y(x) - y(\bar{x}) \rangle \geq \bar{\tau} (f(y(x)) - f(y(\bar{x}))) \tag{5.2}$$

for any $\bar{x} = (\bar{y}, \bar{\tau}) \in K$. In particular, for $\bar{x} = x^* \in X^*$, we have

$$\langle \hat{g}(x), x - x^* \rangle \geq 0.$$

Let us set $g(x) = \hat{g}(x) / \| \hat{g}(x) \|$. Then

$$\| \hat{g}(x) \| = \left[\| f'(y(x)) \|^2 + \langle f'(y(x)), y(x) \rangle^2 \right]^{1/2} \leq L \sqrt{1 + R^2}. \tag{5.3}$$

Complexity estimate. Suppose we generate a sequence of the analytic centers $\{x_k\}_{k=0}^{\infty}$ using the scheme (2.5). Denote $y_k = y(x_k)$. Then, in view of inequalities (5.2), (5.3), and the definition of μ_k, we have

$$\begin{aligned} \min_{0 \leq i \leq k-1} f(y_i) - f(y(x)) &\leq \tfrac{1}{t} \min_{0 \leq i \leq k-1} \langle \hat{g}(x_i), x_i - x \rangle \\ &\leq \tfrac{1}{t} \min_{0 \leq i \leq k-1} \langle g(x_i), x_i - x \rangle L \sqrt{1 + R^2} \leq \tfrac{1}{t} \mu_k(x) L \sqrt{1 + R^2}. \end{aligned} \tag{5.4}$$

Inserting the bound for $\mu_k(x)$ from (2.15) into (5.4) yields

$$\min_{0 \leq i \leq k-1} f(y_i) - f(y(x)) \leq \frac{\sqrt{k + \nu}}{k \theta_3 t} \exp \left\{ \frac{1}{k} (F(x_k) - F(x_0)) \right\} \| x \| \cdot L \sqrt{1 + R^2}. \tag{5.5}$$

Let us fix some $x^* = (y^*, 1) \in X^*$ and $\alpha \in (0, 1)$. Consider the point

$$x_\alpha = (1 - \alpha) x^* + \alpha x_0 = ((1 - \alpha) y^*, 1).$$

Then $y_\alpha \equiv y(x_\alpha) \in Q$. There are two possibilities. First, it might be that $\langle g(x_i), x_\alpha - x_i \rangle \geq 0$ for some $i \in [0 \ldots k-1]$. Then, in view of our assumption, we have

$$0 \geq \langle \hat{g}(x_i), x_i - x_\alpha \rangle = \langle \hat{g}(x_i), -x_\alpha \rangle = -(1 - \alpha) \langle f'(y_i), y^* \rangle + \langle f'(y_i), y_i \rangle.$$

Consequently,

$$\begin{aligned} f(y_\alpha) &\geq f(y_i) + \langle f'(y_i), y_\alpha - y_i \rangle \\ &= f(y_i) + \langle f'(y_i), (1 - \alpha) y^* - y_i \rangle \geq f(y_i). \end{aligned}$$

Therefore,

$$\min_{0\le i\le k-1} f(y_i) \le f(y_\alpha) \le (1-\alpha)f^* + \alpha f(0),$$

implying that,

$$\min_{0\le i\le k-1} f(y_i) - f^* \le \alpha(f(0)-f^*) \le \alpha L \parallel y^* \parallel \le \alpha LR. \tag{5.6}$$

Let us assume now that $\langle g(x_i), x_i - x_\alpha\rangle \ge 0$ for all $i \in [0\ldots k-1]$. Then, in view of (2.6), we have

$$\begin{aligned}\langle -F'(x_k), x_\alpha\rangle &= \left\langle \nu x_k + \sum_{i=0}^{k-1} \frac{g(x_i)}{\langle g(x_i), x_i - x_k\rangle}, x_\alpha \right\rangle \\ &= \nu\langle x_k, x_\alpha\rangle + \sum_{i=0}^{k-1} \frac{\langle g(x_i), x_\alpha - x_i\rangle}{\langle g(x_i), x_i - x_k\rangle} \\ &\le \nu\langle x_k, x_\alpha\rangle \le \sqrt{\nu(k+\nu)}\cdot \parallel x_\alpha \parallel .\end{aligned}$$

The last inequality follows from Lemma 2.3. Therefore, using (2.4), we have

$$F(x_k) \le F(x_\alpha) + \nu \ln \frac{\langle -F'(x_k), x_\alpha\rangle}{\nu} \le F(x_\alpha) + \nu \ln\left[\sqrt{1+\frac{k}{\nu}}\cdot \parallel x_\alpha \parallel\right].$$

In what follows we assume that $F(x)$ is a restriction of some self-scaled barrier [19, 18]. More precisely, we assume that the set Q consists of points x, for which there exist some s (dependent on x) such that $Ax + s = b$ and $s \in \hat{K}$, where $\hat{K}$ is a symmetric cone (or self-scaled cone in the terminology of [19, 18]) endowed with a self-scaled barrier $\Phi(s)$. We assume that $F(x) = \Phi(b - Ax)$.

Let

$$p = -(y^*, 0) = x_0 - x^*, \qquad \sigma = \frac{1}{\sup\{\gamma \mid x_0 - \gamma p \in K\}}.$$

Then $x_\alpha = x_0 - \beta p$, with $\beta = 1-\alpha$. Note that $\sigma \le 1$ and $\parallel p \parallel_{F''(x_k)} \le \sqrt{\nu}\sigma$. (See Proposition 3.2 in [18].) Moreover,

$$\langle F'(x_0), x_\alpha - x_0\rangle = \langle F'_y(x_0), y_\alpha\rangle = 0.$$

Therefore, in view of inequality (4.7) of [19], we have

$$\begin{aligned}F(x_\alpha) - F(x_0) &\le \tfrac{\|p\|^2_{F''(x_k)}}{\sigma^2}(-\sigma\beta - \ln(1-\beta\sigma)) \\ &\le \nu(-\beta - \ln(1-\beta)) = \nu\left(\alpha - 1 + \ln\tfrac{1}{\alpha}\right) \le \nu \ln \tfrac{1}{\alpha}.\end{aligned}$$

Combining these inequalities we obtain

$$\begin{aligned}F(x_k) - F(x_0) &\le \nu \ln\tfrac{1}{\alpha} + \nu \ln\left[\sqrt{1+\tfrac{k}{\nu}}\cdot \parallel x_\alpha \parallel\right] \\ &= \nu \ln\left[\sqrt{1+\tfrac{k}{\nu}}\cdot \tfrac{1}{\alpha} \parallel x_\alpha \parallel\right].\end{aligned}$$

Substituting this inequality in (5.5) with $x = x_\alpha$, we get

$$\min_{0\le i\le k-1} f(y_i) - f^* \le f(y_\alpha) - f^* + \tfrac{\sqrt{k+\nu}}{k\theta_3}\left[\sqrt{1+\tfrac{k}{\nu}}\cdot\tfrac{1}{\alpha}\right]^{\nu/k} \parallel x_\alpha \parallel^{1+\nu/k} \cdot L\sqrt{1+R^2}$$

$$\le \alpha LR + \tfrac{\sqrt{k+\nu}}{k\theta_3}\left[\sqrt{1+\tfrac{k}{\nu}}\cdot\tfrac{1}{\alpha}\right]^{\nu/k} \cdot L(1+R^2)^{1+\frac{\nu}{2k}}.$$

Since $(1+k/\nu)^{\nu/k} < e$, we obtain for $\alpha = 1/\sqrt{1+k/\nu}$

$$\min_{0\le i\le k-1} f(y_i) - f^* \le \frac{\sqrt{\nu}LR}{\sqrt{k+\nu}} + \frac{e\sqrt{k+\nu}}{k\theta_3}\cdot L(1+R^2)^{1+\frac{\nu}{2k}}. \tag{5.7}$$

Clearly, the upper bound given by (5.6) is smaller than the one provided by (5.7). Therefore, (5.7) proves the following theorem.

THEOREM 5.1. *For any $k \ge 1$ we have*

$$\min_{0\le i\le k-1} f(y_i) - f^* \le \frac{L}{\sqrt{k+\nu}}\left[\sqrt{\nu} + \frac{e}{\theta_3}\left(1+\frac{\nu}{k}\right)\right][1+R^2]^{1+\frac{\nu}{2k}}.$$

Scaling. The result of Theorem 5.1 exhibits a quadratic dependence on R. That is not a standard dependence, since for our problem class it should be proportional to LR. In order to improve the situation we need only introduce a scaling parameter in our scheme. Indeed, let our initial problem be

$$\min_{y\in Q_1} \phi(y). \tag{5.8}$$

Let us assume that this problem satisfies the assumptions on problem (5.1). Namely, we assume that $\parallel \phi'(y) \parallel \le L_1$ and $\parallel y \parallel \le R_1$ for all $y \in \operatorname{int} Q_1$. Let us fix some $\kappa > 0$ and apply our minimization scheme to problem (5.1) with

$$f(y) = \phi(\kappa y), \qquad Q = \frac{1}{\kappa}Q_1.$$

Then the parameters of this problem become

$$L = \kappa L_1, \qquad R = \frac{1}{\kappa}R_1.$$

Note that both problems have the same optimal value. The sequence of objective function values is the same for the two minimizing sequences, $\{y_i\}_{i=0}^\infty$ and $\{\kappa y_i\}_{i=0}^\infty$, which are generated for the first and the second problem, respectively. Therefore, in view of Theorem 5.1, we have

$$\min_{0\le i\le k-1} \phi(\kappa y_i) - \phi^* \le \frac{L_1}{\sqrt{k+\nu}}\left[\sqrt{\nu} + \frac{e}{\theta_3}\left(1+\frac{\nu}{k}\right)\right]\cdot\kappa\left[1+\left(\frac{R_1}{\kappa}\right)^2\right]^{1+\frac{\nu}{2k}}.$$

Thus, if we make the choice $\kappa = \gamma R_1$ with some $\gamma > 0$, we get

$$\min_{0\le i\le k-1} \phi(\kappa y_i) - \phi^* \le \frac{L_1 R_1}{\sqrt{k+\nu}}\left[\sqrt{\nu} + \frac{e}{\theta_3}\left(1+\frac{\nu}{k}\right)\right]\gamma[1+\gamma^{-2}]^{1+\frac{\nu}{2k}}.$$

The factor depending on γ in the right-hand side of this inequality approaches $\gamma + \frac{1}{\gamma}$ as $k \to \infty$. Asymptotically the best choice is $\gamma = 1$. However, if we do not have exact information about R_1, we must pay for it, but the price is only an absolute multiplicative factor.

Note that the same reasoning also applies to the efficiency estimates for the convex feasibility problem and for variational inequalities.

Iterations in the original space. Let us discuss the interpretation of the homogeneous analytic center cutting plane scheme for the constrained minimization problem. Recall that at each iteration of this scheme we minimize the potential

$$\frac{1}{2} \parallel x \parallel^2 + F(x) - \sum_{i=0}^{k-1} \ln\langle g(x_i), x_i - x\rangle,$$

where $g(x) = \hat{g}(x)/ \parallel \hat{g}(x) \parallel$ and

$$\hat{g}(x) = \left(f'\left(\frac{y}{t}\right), -\left\langle f'\left(\frac{y}{t}\right), \left(\frac{y}{t}\right)\right\rangle\right)$$

with $\frac{y}{t} \in Q$. Thus, in fact, we deal with the following potential:

$$\frac{1}{2} \parallel x \parallel^2 + F(x) - \sum_{i=0}^{k-1} \ln[-\langle \hat{g}(x_i), x\rangle].$$

Let us represent a point $x \in \operatorname{int} K$ as $x = (t\hat{y}, t)$ with $\hat{y} \in \operatorname{int} Q$ and $t > 0$. Note that

$$-\langle \hat{g}(x_i), x\rangle = -\langle f'(\hat{y}_i), t\hat{y}\rangle + \langle f'(\hat{y}_i), \hat{y}_i\rangle t = t\langle f'(\hat{y}_i), \hat{y}_i - \hat{y}\rangle.$$

Moreover,

$$F(x) = F(t\hat{y}, t) = F(\hat{y}, 1) - \nu \ln t$$

and $\hat{F}(\hat{y}) = F(\hat{y}, 1)$ is a ν-self-concordant barrier for the set Q. Thus, in terms of the variables $\hat{y}$, our potential is

$$\begin{aligned} &\frac{1}{2} \parallel x \parallel^2 + \hat{F}(\hat{y}) - \nu \ln t - \sum_{i=0}^{k-1} \ln\langle f'(\hat{y}_i), \hat{y}_i - \hat{y}\rangle - k \ln t \\ &\quad = \frac{1}{2} t^2 (1 + \parallel \hat{y} \parallel^2) + \hat{F}(\hat{y}) - \sum_{i=0}^{k-1} \ln\langle f'(\hat{y}_i), \hat{y}_i - \hat{y}\rangle - (k+\nu) \ln t. \end{aligned} \tag{5.9}$$

Let y be given. The potential achieves its minimum value at

$$t_k^* = \sqrt{\frac{k+\nu}{1 + \parallel \hat{y} \parallel^2}}.$$

Therefore, replacing t by t_k^* in (5.9), we get the following function in the $\hat{y}$-space:

$$\psi_k(\hat{y}) = \frac{k+\nu}{2} \ln(1 + \parallel \hat{y} \parallel^2) + \hat{F}(\hat{y}) - \sum_{i=0}^{k-1} \ln\langle f'(\hat{y}_i), \hat{y}_i - \hat{y}\rangle + c_k,$$

where $c_k = \frac{k+\nu}{2}\,(1 - \ln(\nu + k))$.

Thus, the homogeneous analytic center method can be seen as a standard analytic center scheme augmented by the logarithm of a proximal term. Note that this logarithmic term is quasi-convex in $\hat{y}$, but the convexity or even quasi-convexity of the function ψ_k is under question. These considerations indicate that the practical implementation of the proposed scheme must be done in the extended space.

6. Conclusion. As pointed out in section 2, a practical implementation of the algorithm must work with approximate analytic centers. A complexity estimate for the implementable version of the algorithm can be obtained by using the standard argumentation, based on the theory of self-concordant functions. One should prove two things: the bound on the number of iterations is of the same order as with exact analytic centers; and the number of auxiliary Newton steps to compute an approximate center at each iteration is bounded by an absolute constant. (See [15] for an example of this reasoning.)

Note that in the proposed schemes the complexity of finding an analytic center increases as the number of cutting planes increases. Therefore, it would be interesting to study the possibility of dropping old or shallow cutting planes. However, up to now there is no known scheme which can bound the number of cutting planes for the analytic center cutting plane methods (see [2] and references therein). Solving this problem would have obvious practical consequences, but it would also significantly improve the theoretical complexity results. This is therefore an important open question.

The complexity result of this paper is of the same order as for the proximal analytic center method of [15]. However, the new scheme is much more flexible in terms of the accuracy of the initial information. (In [15] it is necessary to choose a parameter $R >\| y \|$ for all $y \in Q$.) Besides, we managed to prove the complexity result of an analytic center scheme for the constrained problems whose solutions may belong to the boundary of the basic feasible set. These results seem to be new.

Finally, we would like to recall two earlier comments. First, in section 2 we showed that the proximal term could be multiplied by an arbitrary constant without changing the iterates. Secondly, in section 5, we gave evidence that the embedding into an extended space is necessary for both theoretical and practical purposes.

Appendix. In section 2 we used results on self-concordant functions from the notes [16]. Since the notes have not yet appeared in the open literature, we include them in the appendix for the sake of completeness.

Let us consider a *closed convex* function $f(x) \in C^3(\operatorname{dom} f)$ with *open* domain.

DEFINITION A.1. *We call a function f* self-concordant *if the inequality*

$$| D^3 f(x)[u,u,u] | \leq 2 \| u \|^{3/2}_{f''(x)}$$

holds for any $x \in \operatorname{dom} f$ and $u \in R^n$.

Let us fix $x \in \operatorname{dom} f$ and $u \in R^n$, $u \neq 0$. Consider the two functions of one variable:

$$\psi(t) = \langle f''(x+tu)u, u\rangle$$

and

$$\phi(t) = \psi(t)^{-1/2}.$$

Clearly, $\operatorname{dom} \psi = \{t \in R \mid \ x + tu \in \operatorname{dom} f\}$ and $\operatorname{dom} \phi = \{t \in \operatorname{dom} \psi \mid \psi(t) > 0\}$. Note that $\operatorname{dom} \psi$ and $\operatorname{dom} \phi$ are open.

LEMMA A.2. *For all $t \in \operatorname{dom} \phi$ we have $| \phi'(t) | \leq 1$.*

Proof. Indeed,

$$\phi'(t) = -\frac{f'''(x+tu)[u,u,u]}{2\langle f''(x+tu)u, u\rangle^{3/2}}.$$

Therefore, $| \phi'(t) | \leq 1$ in view of Definition A.1. □

Corollary A.3. *If* $\operatorname{dom}\phi \neq \phi$, *then* $\operatorname{dom}\phi = \operatorname{dom}\psi$.

Proof. Let $\hat{t} \in \operatorname{dom}\phi$. Denote by Δ the largest connected open interval such that $\hat{t} \in \Delta \subseteq \operatorname{dom}\phi$. Then $\partial\Delta \subseteq \partial\operatorname{dom}\phi$.

Assume first that an end point of Δ, $\bar{t}$ belongs to the intersection $\partial\operatorname{dom}\phi \bigcap \operatorname{dom}\psi$. Then, for any sequence $t_i \in \Delta$, $t_i \to \bar{t}$, we have

$$\phi(t_i) \leq \phi(t_0) + | t_i - \hat{t} | .$$

Then

$$\psi(\bar{t}) = \lim_{i\to\infty} \psi(t_i) \geq \lim_{i\to\infty} \frac{1}{[\phi(t_0)+ | t_i - \hat{t} |]^2} \geq \frac{1}{[\phi(t_0)+ | \Delta |]^2} > 0.$$

Therefore, $\bar{t} \in \operatorname{dom}\phi$. This is a contradiction, which proves that $\bar{t} \in \partial\operatorname{dom}\psi$. Thus, $\operatorname{dom}\phi = \operatorname{dom}\psi$. □

Corollary A.4. *Either* $\psi(0) = 0$ *and* $x + tu \in \operatorname{dom} f$ *for all* $t \in R$, *or* $(-\phi(0), \phi(0)) \subseteq \operatorname{dom}\phi$*; i.e.,* $x + tu \in \operatorname{dom} f$ *for all* t *such that* $-\phi(0) < t < \phi(0)$.

Proof. Assume first that $\psi(0) > 0$. In view of Lemma A.2 and Corollary A.3, we have $\phi(t) \geq \phi(0) - | t |$ for all $t \in \operatorname{dom}\psi$. Therefore, for all t such that $|t| \leq \phi(0) - \epsilon$ with some $\epsilon > 0$, we have that $\psi(t)$ is uniformly bounded. Hence, $[-\phi(0)+\epsilon, \phi(0)-\epsilon] \subseteq \operatorname{dom}\psi$.

Consider now the case when $\psi(0) = 0$. Then $\operatorname{dom}\phi = \emptyset$. This means that $\psi(t) = \langle f''(x + tu)u, u\rangle = 0$ for all t. Therefore, the function $f(x + tu)$ is linear in t. Hence, $x + tu \in \operatorname{dom} f$ for any t. □

Denote $\| u \|_x = \langle f''(x)u, u\rangle^{1/2}$. Let us consider the following ellipsoid:

$$W^0(x; r) = \{y \in R^n \mid \| y - x \|_x < r\},$$

$$W(x; r) = \operatorname{Cl}\left(W^0(x; r)\right) = \{y \in R^n \mid \| y - x \|_x \leq r\}.$$

This ellipsoid is called the *Dikin* ellipsoid of function f at x.

Lemma A.5.

(1) *For any* $x \in \operatorname{dom} f$ *we have* $W^0(x; 1) \subseteq \operatorname{dom} f$.

(2) *For all* x, $y \in \operatorname{dom} f$ *the following inequality holds:*

$$\| y - x \|_y \geq \frac{\| y - x \|_x}{1+ \| y - x \|_x}. \tag{A.1}$$

(3) *If* $\| y - x \|_x < 1$, *then*

$$\| y - x \|_y \leq \frac{\| y - x \|_x}{1- \| y - x \|_x}. \tag{A.2}$$

Proof. (1) In view of Corollary A.4, either $\| u \|_x = 0$ and the line $x + tu$ belongs to $\operatorname{dom} f$, or $0 \in \operatorname{dom}\phi$ and $(-\phi(0), \phi(0)) \in \operatorname{dom}\phi$ with $\phi(0) = 1/ \| u \|_x$. The latter implies that $\operatorname{dom} f$ contains the set $\{y = x + tu \mid\ t^2 \| u \|_x^2 < 1\}$, which is identical to $W^0(x; 1)$.

(2) Let us choose $u = y - x$. Then

$$\phi(1) = \frac{1}{\| y - x \|_y}, \qquad \phi(0) = \frac{1}{\| y - x \|_x},$$

and $\phi(1) \leq \phi(0) + 1$ in view of Lemma A.2. This is the same as (A.1).

(3) If $\| y - x \|_x < 1$, then $\phi(0) > 1$, and in view of Lemma A.2, $\phi(1) \geq \phi(0) - 1$. This is the same as (A.2). □

LEMMA A.6. *For any x, $y \in \operatorname{dom} f$ we have*

$$\langle f'(y) - f'(x), y - x \rangle \geq \frac{\| y - x \|_x^2}{1 + \| y - x \|_x}, \tag{A.3}$$

$$f(y) \geq f(x) + \langle f'(x), y - x \rangle + \omega(\| y - x \|_x), \tag{A.4}$$

where $\omega(t) = t - \ln(1 + t)$.

Proof. Denote $y_\tau = x + \tau(y - x)$, $\tau \in [0, 1]$, and $r = \| y - x \|_x$. Then, in view of (A.1), we have

$$\begin{aligned} \langle f'(y) - f'(x), y - x \rangle &= \int_0^1 \langle f''(y_\tau)(y - x), y - x \rangle d\tau = \int_0^1 \frac{1}{\tau^2} \| y_\tau - x \|_{y_\tau}^2 \, d\tau \\ &\geq \int_0^1 \frac{r^2}{(1 + \tau r)^2} d\tau = r \int_0^r \frac{1}{(1 + t)^2} dt = \frac{r^2}{1 + r}. \end{aligned}$$

Further, using (A.3), we obtain

$$\begin{aligned} f(y) - f(x) - \langle f'(x), y - x \rangle &= \int_0^1 \langle f'(y_\tau) - f'(x), y - x \rangle d\tau \\ &= \int_0^1 \frac{1}{\tau} \langle f'(y_\tau) - f'(x), y_\tau - x \rangle d\tau \\ &\geq \int_0^1 \frac{\| y_\tau - x \|_x^2}{\tau(1 + \| y_\tau - x \|_x)} d\tau \\ &= \int_0^1 \frac{\tau r^2}{1 + \tau r} d\tau = \int_0^r \frac{t dt}{1 + t} = \omega(r). \end{aligned}$$ □

LEMMA A.7. *Let $x \in \operatorname{dom} f$ and $\| y - x \|_x < 1$. Then*

$$\langle f'(y) - f'(x), y - x \rangle \leq \frac{\| y - x \|_x^2}{1 - \| y - x \|_x}, \tag{A.5}$$

$$f(y) \leq f(x) + \langle f'(x), y - x \rangle + \omega_*(\| y - x \|_x), \tag{A.6}$$

where $\omega_*(t) = -t - \ln(1 - t)$.

Proof. Denote $y_\tau = x + \tau(y - x)$, $\tau \in [0, 1]$, and $r = \| y - x \|_x$. Since $\| y_\tau - x \| < 1$, in view of (A.2) we have

$$\begin{aligned} \langle f'(y) - f'(x), y - x \rangle &= \int_0^1 \langle f''(y_\tau)(y - x), y - x \rangle d\tau = \int_0^1 \frac{1}{\tau^2} \| y_\tau - x \|_{y_\tau}^2 \, d\tau \\ &\leq \int_0^1 \frac{r^2}{(1 - \tau r)^2} d\tau = r \int_0^r \frac{1}{(1 - t)^2} dt = \frac{r^2}{1 - r}. \end{aligned}$$

Further, using (A.5), we obtain

$$\begin{aligned} f(y) - f(x) - \langle f'(x), y - x\rangle &= \int_0^1 \langle f'(y_\tau) - f'(x), y - x\rangle d\tau \\ &= \int_0^1 \frac{1}{\tau}\langle f'(y_\tau) - f'(x), y_\tau - x\rangle d\tau \\ &\leq \int_0^1 \frac{\| y_\tau - x \|_x^2}{\tau(1 - \| y_\tau - x \|_x)} d\tau = \int_0^1 \frac{\tau r^2}{1 - \tau r} d\tau \\ &= \int_0^r \frac{tdt}{1-t} = \omega_*(r). \qquad \Box \end{aligned}$$

Acknowledgment. B. Büeler worked out the computations on the small example of section 4. We thank him for letting us use his results. We are also grateful for his comments on the paper.

REFERENCES

[1] A. Altman and K. C. Kiwiel, *A note on some cutting plane methods for convex feasibility and minimization problems*, Comput. Optim. Appl., 5 (1996), pp. 175–180.

[2] D. S. Atkinson and P. M. Vaidya, *A cutting plane algorithm for convex programming that uses analytic centers*, Math. Programming, 69 (1995), pp. 1–43.

[3] O. Bahn, O. du Merle, J.-L. Goffin, and J.-P. Vial, *A cutting plane method from analytic centers for stochastic programming*, Math. Programming, 69 (1995), pp. 45–73.

[4] O. Bahn, J.-L. Goffin, J.-P. Vial, and O. du Merle, *Experimental behaviour of an interior point cutting plane algorithm for convex programming: An application to geometric programming*, Discrete Appl. Math., 49 (1994), pp. 2–23.

[5] J.-L. Goffin, J. Gondzio, R. Sarkissian, and J.-P. Vial, *Solving nonlinear multicommodity flow problems by the analytic center cutting plane method*, Math. Programming, 76 (1997), pp. 131–154.

[6] J.-L. Goffin, A. Haurie, and J.-P. Vial, *Decomposition and nondifferentiable optimization with the projective algorithm*, Management Sci., 38 (1992), pp. 284–302.

[7] J.-L. Goffin, Z. Q. Luo, and Y. Ye, *On the complexity of a column generation algorithm for convex and quasiconvex feasibility problems*, in Large Scale Optimization: State of the Art, W. Hager, D. Hearn, and P. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA, 1993, pp. 187–196.

[8] J.-L. Goffin, Z.-Q. Luo, and Y. Ye, *Complexity analysis of an interior cutting plane method for convex feasibility problems*, SIAM J. Optim., 6 (1996), pp. 638–652.

[9] J.-L. Goffin, P. Marcotte, and D. Zhu, *An analytic center cutting plane method for pseudomonotone variational inequalities*, Oper. Res. Lett., 20 (1997), pp. 1–6.

[10] J.-L. Goffin and J.-P. Vial, *Shallow, Deep and Very Deep Cuts in the Analytic Center Cutting Plane Method*, Math. Programming, 84 (1999), pp. 89–103.

[11] K. C. Kiwiel, *Efficiency of the analytic center cutting plane method for convex minimization*, SIAM J. Optim., 7 (1997), pp. 336–346.

[12] Z. Luo and J. Sun, *An analytic center based column generation algorithm for convex quadratic feasibility problems*, SIAM J. Optim., 9 (1999), pp. 217–235.

[13] F. S. Mokhtarian and J. L. Goffin, *A nonlinear analytic center cutting plane method for a class of convex programming problems*, SIAM J. Optim., 8 (1998), pp. 1108–1131.

[14] A. Nemirovsky and D. Yudin, *Informational Complexity and Efficient Methods for Solution of Convex Extremal Problems*, John Wiley, New York, 1983.

[15] Y. Nesterov, *Complexity estimates of some cutting plane methods based on the analytic center*, Math. Programming, 69 (1995), pp. 149–176.

[16] Y. Nesterov, *Introductory Lectures on Convex Optimization*, CORE, Louvain, Belgium, 1996.

[17] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Stud. Appl. Math. 13, Philadelphia, 1994.

[18] Yu. E. Nesterov and M. J. Todd, *Primal-dual interior-point methods for self-scaled cones*, SIAM J. Optim., 8 (1998), pp. 324–364.

[19] Y. Nesterov and M. Todd, *Self-scaled cones and interior-point methods in nonlinear programming*, Math. Oper. Res., 22 (1997), pp. 1–42.

[20] G. Sonnevend, *New algorithms in convex programming based on a notion of "centre" (for systems of analytic inequalities) and on rational extrapolation*, in Trends in Mathematical Optimization: Proceedings of the 4th French–German Conference on Optimization, Irsee, Germany, April 1986, Internat. Ser. Numer. Math. 84, K. H. Hoffmann, J.-B. Hiriat-Urruty, C. Lemarechal, and J. Zowe, eds., Birkhäuser, Basel, 1988, pp. 311–327.

SIAM J. Optim.
Vol. 9, No. 3, pp. 729–754

A POTENTIAL REDUCTION NEWTON METHOD FOR CONSTRAINED EQUATIONS*

RENATO D. C. MONTEIRO† AND JONG-SHI PANG‡

Abstract. Extending our previous work [T. Wang, R. D. C. Monteiro, and J.-S. Pang, *Math. Programming*, 74 (1996), pp. 159–195], this paper presents a general potential reduction Newton method for solving a constrained system of nonlinear equations. A major convergence result for the method is established. Specializations of the method to a convex semidefinite program and a monotone complementarity problem in symmetric matrices are discussed. Strengthened convergence results are established in the context of these specializations.

Key words. potential reduction algorithm, constrained equation, Newton method, interior point methods, global convergence, potential function, complementarity problems, variational inequality, semidefinite programming, primal-dual methods

AMS subject classifications. 65K05, 90C25, 90C33

PII. S1052623497318980

1. Introduction.

In the paper [36], we have introduced the problem of solving a system of nonlinear equations subject to additional constraints on the variables, i.e., a *constrained* system of equations. We have demonstrated that constrained equations (CEs) provide a unifying framework for the study of complementarity problems of various types, including the standard nonlinear complementarity problem and the Karush–Kuhn–Tucker system of a variational inequality. Postulating a partitioning property of the CE, we have introduced an interior point potential reduction algorithm for solving the CE and have applied this method to convex programs and monotone complementarity problems of different kinds. The goal of this paper is to present a potential reduction Newton method for solving a CE, without assuming the existence of the partitioning property that is key to the previous work.

The central problem studied in section 2 of this paper is as follows. Let $H : \Re^n \to \Re^n$ be a given mapping from the real Euclidean space $\Re^n$ into itself and let Ω be a given closed subset of $\Re^n$. The constrained equation defined by the pair (Ω, H) is to find a vector $x \in \Re^n$ such that

$$H(x) = 0, \quad x \in \Omega.$$

We refer the reader to [36] for the initial motivation to study the CE. The method proposed in this paper for solving the CE (Ω, H) combines ideas from the classical damped Newton method for solving the unconstrained system of equations $H(x) = 0$, $x \in \Re^n$, and the family of interior point methods for solving constrained optimization and complementarity problems. A general convergence theory for the proposed

*Received by the editors March 26, 1997; accepted for publication (in revised form) June 22, 1998; published electronically May 21, 1999.

http://www.siam.org/journals/siopt/9-3/31898.html

†School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205 (monteiro@isye.gatech.edu). The research of this author was supported by the National Science Foundation under grants INT-9600343 and CCR-9700448 and the Office of Naval Research under grant N00014-94-1-0340.

‡Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, MD 21218-2682 (jsp@vicp1.mts.jhu.edu). The research of this author was supported by the National Science Foundation under grant CCR-9624018 and by the Office of Naval Research under grant N00014-93-1-0228.

method is presented in section 2.4. Unlike the previous study [36], where we assume that the function $H(x)$ has a certain partition conformal to the set Ω, we make no such assumption herein. Instead, the present work is based on a set of broad hypotheses on the pair (Ω, H).

In sections 3 and 4, we consider applications of our results to a monotone complementarity problem and a semidefinite convex program on the cone of positive semidefinite matrices. These applications yield new interior point methods for solving these problems whose convergence can be established under some mild assumptions. It should be noted that many interior point methods for the linear version of these problems have been proposed in the literature (e.g., see [1, 2, 3, 4, 6, 9, 10, 11, 12, 15, 16, 19, 20, 22, 23, 24, 25, 26, 27, 29, 32, 34, 35, 37]).

We explain some terminology and fix the notation used throughout the paper. For a given subset S of $\Re^n$, we let int S, cl S, and bd S denote, respectively, the interior, closure, and boundary of S. If the mapping H is (Fréchet) differentiable at a point x in its domain, the Jacobian matrix of H at x is denoted $H'(x)$; thus the (i, j)-entry of $H'(x)$ is equal to $\partial H_i(x)/\partial x_j$ for $i, j = 1, \ldots, n$. We write $H'(x; v) \equiv H'(x)v$ for any vector $v \in \Re^n$; thus $H'(x; v)$ is the Fréchet derivative of H at x along the direction v. If $H(x, y)$ is a function of two arguments $(x, y) \in \Re^{n+m}$, then H'_x denotes the partial Jacobian matrix of H with respect to the variable x. For a real-valued function $\phi : \Re^n \to \Re$, we write $\nabla\phi(x)$ for the gradient vector of ϕ at the vector $x \in \Re^n$. The p-norm of a vector x is denoted by $\|x\|_p$; in particular, its 2-norm or Euclidean norm is denoted by $\|x\|$. For a vector $a \in \Re^n$, we let $[0, a]$ denote the line segment joining the origin and a. For a positive vector u, we let u^{-1} denote the vector whose components are the reciprocals of the corresponding components of u. For a mapping $G : M \to N$ with domain M, range N, and subsets $D \subset M$ and $E \subset N$, we let

$$G(D) \equiv \{ G(u) : u \in D \} \quad \text{and} \quad G^{-1}(E) \equiv \{ u \in M : G(u) \in E \}.$$

The set of real matrices of order n is denoted by $\mathcal{M}^n$; the subset of symmetric matrices in $\mathcal{M}^n$ is denoted by $\mathcal{S}^n$. The set $\mathcal{M}^n$ forms a finite-dimensional inner-product vector space with the inner product given by

$$X \bullet Y \equiv \operatorname{tr}(X^T Y), \quad (X, Y) \in \mathcal{M}^n,$$

where "tr" denotes the trace of a matrix. This inner product induces the Frobenius norm for matrices given by

$$\| X \|_F \equiv \sqrt{\operatorname{tr}(X^T X)}, \quad X \in \mathcal{M}^n.$$

The subsets of $\mathcal{S}^n$ consisting of the positive semidefinite and positive definite matrices are denoted by $\mathcal{S}^n_+$ and $\mathcal{S}^n_{++}$, respectively. For two matrices A and B in $\mathcal{S}^n$, we write $A \preceq B$ if $B - A \in \mathcal{S}^n_+$; similarly, $A \prec B$ means $B - A \in \mathcal{S}^n_{++}$. For any matrix $A \in \mathcal{S}^n_+$, $A^{1/2}$ denotes the square root of A; i.e., $A^{1/2}$ is the unique matrix in $\mathcal{S}^n_+$ such that $(A^{1/2})^2 = A$.

2. Description and analysis of the algorithm. In this section, we describe the potential reduction Newton algorithm for solving the CE (Ω, H), where Ω is a closed subset of $\Re^n$ and H is a continuous mapping from $\Re^n$ into itself. This section is divided into four subsections as follows: in the first subsection, we lay down the basic assumptions on the pair (Ω, H); in the second subsection, we give some results which guarantee the existence of a solution for the CE (Ω, H); in the third subsection, we present the detailed statement of the algorithm; in the fourth subsection, we establish a convergence theorem for the algorithm.

2.1. Basic assumptions. We introduce several key assumptions on the pair (Ω, H). Subsequently, these assumptions will be verified in the context of several applications of the CE. Among these assumptions, we postulate the existence of a closed convex subset S that relates to the range of H and possesses certain special properties. Based on such a set S and a corresponding potential function p, an algorithm for solving the CE is developed. Part of the generality of the present framework stems from the freedom in the choice of S. There are two immediate benefits of this generality. One is that our framework provides a unified basis for the study of many iterative algorithms for solving nonlinear equations and mathematical programs. More importantly, the other benefit is that new algorithms can be constructed with novel choices of S. Of particular interest is the construction of sets S and associated potential functions that depend on given starting points. These details will appear in subsequent sections. The blanket assumptions are as follows.

(A1) The closed set Ω has a nonempty interior.

(A2) There exists a closed convex set $S \subset \Re^n$ such that

(a) $0 \in S$;

(b) the (open) set $\Omega_{\mathcal{I}} \equiv H^{-1}(\text{int } S) \cap \text{int } \Omega$ is nonempty;

(c) the set $H^{-1}(\text{int } S) \cap \text{bd } \Omega$ is empty.

(A3) H is continuously differentiable on $\Omega_{\mathcal{I}}$, and $H'(x)$ is nonsingular for all $x \in \Omega_{\mathcal{I}}$.

Assumption (A1) is needed for the applicability of an interior point method. The sets S and $\Omega_{\mathcal{I}}$ in assumption (A2) contain the key elements of the proposed algorithm. (As noted by a referee, if H is considered to be a mapping with domain Ω, conditions (b) and (c) in (A2) are equivalent to the condition that $\emptyset \neq H^{-1}(\text{int } S) \subset \text{int } \Omega$.) Whereas S pertains to the range of H, $\Omega_{\mathcal{I}}$ pertains to the domain. Initiated at a vector x^0 in $\Omega_{\mathcal{I}}$, the algorithm generates a sequence of iterates $\{x^k\} \subset \Omega_{\mathcal{I}}$ so that the sequence $\{H(x^k)\} \subset \text{int } S$ will eventually converge to zero, thus accomplishing the goal of solving the CE (Ω, H), at least approximately. Assumption (A3) facilitates the application of a Newton scheme for the generation of $\{x^k\}$; this scheme relies on a potential function for the set $\Omega_{\mathcal{I}}$ that is induced by such a function for int S. Specifically, we postulate the existence of a potential function $p : \text{int } S \to \Re$ satisfying the following properties:

(A4) for every sequence $\{u^k\} \subset \text{int } S$ such that

$$\text{either } \lim_{k\to\infty} \|u^k\| = \infty \text{ or } \lim_{k\to\infty} u^k = \bar{u} \in \text{bd } S \setminus \{0\},$$

we have

$$\lim_{k\to\infty} p(u^k) = \infty. \tag{1}$$

(A5) p is continuously differentiable on its domain and $u^T \nabla p(u) > 0$ for all nonzero $u \in \text{int } S$.

A condition equivalent to (A4) is stated in the following straightforward result.

Lemma 1. *Condition* (A4) *holds if and only if for all* $\gamma \in \Re$ *and* $\varepsilon > 0$, *the set*

$$\Lambda(\varepsilon, \gamma) \equiv \{\, u \in \text{int } S : p(u) \leq \gamma, \ \|u\| \geq \varepsilon \,\}$$

is compact.

The notion of the central path has played a fundamental role in all interior point methods for solving optimization and complementarity problems [7, 13, 14]. Inspired by this notion, we introduce an important vector a that will be used to define a modified Newton direction that is key to the generation of the iterates for solving the CE (Ω, H). Although the vector a is inspired by the central vector of all ones in the case where S is the nonnegative orthant, since our present setting is very broad, the vector a should not be thought of as just a "central vector" for int S; instead, a is closely linked with the potential function p, which itself is fairly loosely restricted.

(A6) There exists a pair $(a, \bar{\sigma}) \in \Re^n \times (0, 1]$ such that

$$\|a\|^2 \left(u^T \nabla p(u)\right) \ \geq \ \bar{\sigma}\,(a^T u)(a^T \nabla p(u)) \quad \forall\ u \in \text{int } S.$$

Trivially, (A6) holds with $a = 0$ and any $\bar{\sigma} \in (0, 1]$. It follows that the entire development in this paper holds with $a = 0$. Nevertheless, the interesting case is when $a \neq 0$. The purpose of (A6) is to identify a broad class of such vectors a for which one can establish the convergence of the potential reduction algorithm of section 2.3. For many problems (such as those described in this paper), a nonzero vector a satisfying (A6) can be identified easily; for others, we could always resort to the zero vector.

The basic role of the potential function p is to keep the sequence $\{H(x^k)\}$ away from the set bd $S \backslash \{0\}$ while leading it toward the zero vector. Hence, its role is slightly different from that of a standard barrier function used in nonlinear programming, which in contrast penalizes an iterate when it gets close to *any* boundary point of S.

Our framework includes the most basic case of solving a smooth system of unconstrained equations. This case corresponds to $\Omega = \Re^n$. In this case, we may simply take S to be the entire space $\Re^n$ (so that bd $S = \emptyset$), $p(u)$ to be the function $\|u\|^2$, a to be any vector, and $\bar{\sigma} = 1$. It is then clear that (A2) and (A4)–(A6) all hold easily.

Another simple case to illustrate the above assumptions (with an unspecified Ω) is when S is the nonnegative orthant $\Re^n_+$. In what follows, we establish the validity of conditions (A4)–(A6) for the function

$$p(u) \ = \ \zeta \, \log u^T u - \sum_{i=1}^{n} \log u_i, \quad u \ > \ 0$$

and the pair $(a, \bar{\sigma}) = (e, 1)$, where $\zeta > n/2$ is an arbitrary scalar and e is the n-dimensional vector of all ones. (Note: The ℓ_1-norm of u, instead of $u^T u$, could also be used in the first logarithmic term. The analysis remains the same with the constant ζ properly adjusted.) Clearly, p is norm-coercive on $\Re^n_{++}$; i.e.,

$$\lim_{\substack{u > 0 \\ \|u\| \to \infty}} \ p(u) \ = \ \infty,$$

because for $u > 0$,

$$\begin{aligned} p(u) &\geq \zeta \left(2 \log \left(\sum_{i=1}^{n} u_i \right) - \log n \right) - \sum_{i=1}^{n} \log u_i \\ &> (2\,\zeta - n) \, \log \left(\sum_{i=1}^{n} u_i \right) - (\zeta - n) \, \log n, \end{aligned}$$

where the first and second inequalities follow from the fact that $\|u\|_1 \leq \sqrt{n}\|u\|$ and $n \log(\sum_{i=1}^{n} u_i) - \sum_{i=1}^{n} \log u_i \ \geq \ n \log n$, respectively. Moreover, for any positive sequence $\{u^k\}$ converging to a nonzero nonnegative vector with at least one zero component, the limit (1) clearly holds. Thus (A4) follows. Since

$$u^T \nabla p(u) \,=\, u^T \left(\frac{2\,\zeta}{\|\, u \,\|^2}\, u - u^{-1} \right) \,=\, 2\,\zeta - n \,>\, 0,$$

(A5) holds. Moreover, with $(a, \bar{\sigma}) = (e, 1)$, we now show that (A6) also holds. Indeed, we have for $u > 0$,

$$a^T \nabla p(u) \,=\, \frac{2\,\zeta \sum_{i=1}^n u_i}{\sum_{i=1}^n u_i^2} - \sum_{i=1}^n u_i^{-1}\,;$$

thus

$$\begin{aligned} \frac{(a^T \nabla p(u))\,(a^T u)}{\|a\|^2} &= n^{-1} \left[\frac{2\,\zeta\, \|u\|_1^2}{\|u\|^2} - \left(\sum_{i=1}^n u_i^{-1} \right) \left(\sum_{i=1}^n u_i \right) \right] \\ &\leq 2\,\zeta - n = u^T \nabla p(u), \end{aligned}$$

where the last inequality follows from the fact that $\|u\|_1 \,\leq\, \sqrt{n}\,\|u\|$ and from the arithmetic-geometric mean inequality.

Other choices for the function p exist for $S = \Re^n_+$. The above choice will be generalized to the case where S involves the cone of symmetric positive semidefinite matrices.

Admittedly, the set S, function p, and vector a as stated in the general assumptions (A2) and (A4)–(A6) are somewhat abstract. In particular, a question raised by a referee is whether our framework is applicable to a linear program over a general convex cone, the latter being an elegant problem that has received substantial interest in the optimization community in recent years. Needless to say, to be amenable to our framework, the cone linear program has to be written in the form of a CE. (We are convinced that this can be done via duality theory.) After this conversion, the ability to identify S, p, and a depends on how much we know about the given cone. We believe that for cones arising most frequently in applications (such as the well-known quadratic cone), this set, function, and vector can be identified (although the identification could entail considerable additional efforts). For general cones without additional properties, the applicability of our approach is not clear. A careful investigation may reveal some interesting connection between S, p, and a and certain intrinsic conic properties; nevertheless, such an investigation is clearly beyond the scope of this paper.

2.2. Existence of solutions. In this subsection, we study conditions that guarantee the existence of solutions of the CE (Ω, H). We start by giving a few definitions. Assume that M and N are two metric spaces and that $G : M \to N$ is a map between these two spaces. The map G is said to be *proper* with respect to a set $E \subset N$ if $G^{-1}(K) \subset M$ is compact for every compact set $K \subset E$. If G is proper with respect to N, we will simply say that G is proper. For $D \subset M$, and $E \subset N$ such that $G(D) \subset E$, the restricted map $\tilde{G} : D \to E$ defined by $\tilde{G}(u) \equiv G(u)$ for all $u \in D$ is denoted by $G|_{(D,E)}$; if $E = N$, then we write this $\tilde{G}$ simply as $G|_D$. We will also refer to $G|_{(D,E)}$ as "G restricted to the pair (D, E)," and to $G|_D$ as "G restricted to D." We say that (V_1, V_2) forms a *partition* of the set V if $V_1 \subset V$, $V_2 \subset V$, $V_1 \cup V_2 = V$, and $V_1 \cap V_2 = \emptyset$. A metric space M is said to be *connected* if there exists no partition

$(\mathcal{O}_1, \mathcal{O}_2)$ for which both $\mathcal{O}_1$ and $\mathcal{O}_2$ are nonempty and open. A metric space M is said to be *path-connected* if for any two points $u_0, u_1 \in M$, there exists a continuous $p : [0,1] \to M$ such that $p(0) = u_0$ and $p(1) = u_1$.

The following result and its proof can be found in Monteiro and Pang [17] (see Corollary 1 of this reference).

PROPOSITION 1. *Let M and N be two metric spaces and $F : M \to N$ be a continuous map. Let $M_0 \subset M$ and $N_0 \subset N$ be given sets satisfying the following conditions: $F|_{M_0}$ is a local homeomorphism and $\emptyset \neq F^{-1}(N_0) \subset M_0$. Assume that F is proper with respect to some set E such that $N_0 \subset E \subset N$. Then F restricted to the pair $(F^{-1}(N_0), N_0)$ is a proper local homeomorphism. If, in addition, N_0 is connected, then $F(M_0) \supseteq N_0$ and $F(\mathrm{cl}\ M_0) \supseteq E \cap \mathrm{cl}\ N_0$.*

Using Proposition 1, we now derive two existence results for the CE (Ω, H).

THEOREM 1. *Assume that conditions* (A1)–(A3) *hold and that there exists a convex set $E \subset S$ such that $0 \in E$, $E \cap H(\Omega_{\mathcal{I}})$ is nonempty, and $H : \Re^n \to \Re^n$ is proper with respect to E. Then*

(a) *$E \subset H(\Omega)$; in particular, CE (Ω, H) has a solution;*

(b) *H restricted to the pair $(\,\Omega_{\mathcal{I}} \cap H^{-1}(E),\ E \cap \mathrm{int}\ S\,)$ is a proper local homeomorphism.*

Proof. To apply Proposition 1, let $M \equiv \Omega$, $N \equiv \Re^n$, $M_0 \equiv \Omega_{\mathcal{I}}$, $N_0 \equiv E \cap \mathrm{int}\ S$, and $F \equiv H|_{\Omega}$. Using (A2) and the assumption that $E \cap H(\Omega_{\mathcal{I}}) \neq \emptyset$, we easily see that $\emptyset \neq F^{-1}(N_0) \subset M_0$. Moreover, by (A3) and the inverse function theorem, it follows that $F|_{M_0}$ is a local homeomorphism. Since F is proper with respect to E by assumption, it follows from Proposition 1 that

$$H(\Omega) \supseteq H(\mathrm{cl}\ \Omega_{\mathcal{I}}) = F(\mathrm{cl}\ M_0) \supseteq E \cap \mathrm{cl}\ N_0 = E \cap \mathrm{cl}\ (E \cap \mathrm{int}\ S) = E,$$

where the last equality follows from the fact that $\mathrm{cl}\ (E \cap \mathrm{int}\ S) = (\mathrm{cl}\ E) \cap \mathrm{cl}\ (\mathrm{int}\ S) = (\mathrm{cl}\ E) \cap S$, by elementary properties of convex sets (see section 2.1 in Chapter 3 of [5]). Hence, (a) holds. It also follows from Proposition 1 that F restricted to the pair $(F^{-1}(N_0),\ N_0)$ is a proper local homeomorphism. Since by (A2) and the definition of F, we have

$$F^{-1}(N_0) \,=\, \Omega \cap H^{-1}(E \cap \mathrm{int}\ S) \,=\, \Omega_{\mathcal{I}} \,\cap\, H^{-1}(E),$$

we conclude that (b) holds. $\square$

THEOREM 2. *Assume that conditions* (A1)–(A3) *hold and that H is proper with respect to S. Then* (i) *$S \subset H(\Omega)$ and* (ii) *H restricted to $\Omega_{\mathcal{I}}$ maps each path-connected component of $\Omega_{\mathcal{I}}$ homeomorphically onto* int S. *In particular, CE (Ω, H) has a solution.*

Proof. Conclusion (i) follows immediately from Theorem 1(a) with $E = S$. Using Theorem 1(b) with $E = S$, we conclude that H restricted to the pair $(\Omega_{\mathcal{I}}, \mathrm{int}\ S)$ is a proper local homeomorphism. If $\mathcal{T} \subset \Omega_{\mathcal{I}}$ is a path-connected component of $\Omega_{\mathcal{I}}$, then H restricted to the pair $(\mathcal{T},\ \mathrm{int}\ S)$ is a proper local homeomorphism since $\mathcal{T}$ is both open and closed with respect to $\Omega_{\mathcal{I}}$. Since every proper local homeomorphism from a path-connected set into a convex set is a homeomorphism (see, for example, Theorem 1 of [17]), (ii) follows. $\square$

2.3. The algorithm.

The algorithm for solving the CE (Ω, H) is a modified, damped Newton method applied to the equation $H(x) = 0$. Referring the reader to [28] for the basic family of Newton methods for solving this unconstrained equation, we highlight the modifications to deal with the presence of the constraint set Ω. In essence, there are two major modifications. One, the Newton equation to compute the search directions is modified using the (central) vector a in assumption (A6). Two, the merit function for the line searches is based on the merit function:

$$\psi(x) \equiv p(H(x)), \quad x \in \Omega_{\mathcal{I}}. \tag{2}$$

This is different from the norm functions of H that are the common merit functions used in a classical damped Newton method. Note that by (A3) and (A5) the function ψ is continuously differentiable on $\Omega_{\mathcal{I}}$.

With the above explanation, we now give the full details of the Newton method for solving the CE (Ω, H) under the setting given in the last subsection.

Step 0. (Initialization) Let a vector $x^0 \in \Omega_{\mathcal{I}}$ and scalars $\rho \in (0,1)$ and $\alpha \in (0,1)$ be given. Let a sequence of scalars $\{\sigma_k\} \subset [0, \bar{\sigma})$ also be given. (The scalar $\bar{\sigma}$ is as given in assumption (A6).) Set the iteration counter $k = 0$.

Step 1. (Computing the modified Newton direction) Solve the system of linear equations

$$H(x^k) + H'(x^k; d) = \sigma_k \frac{a^T H(x^k)}{\|a\|^2} a \tag{3}$$

to obtain the search direction d^k. (The right-hand side of the above equation is assumed to be zero if $a = 0$. This convention will be assumed throughout our presentation.)

Step 2. (Armijo line search) Let m_k be the smallest nonnegative integer m such that $x^k + \rho^m d^k \in \Omega_{\mathcal{I}}$ and

$$\psi(x^k + \rho^m d^k) - \psi(x^k) \le \alpha \rho^m \nabla\psi(x^k)^T d^k.$$

Set $x^{k+1} \equiv x^k + \rho^{m_k} d^k$.

Step 3. (Termination test) If

$$\| H(x^{k+1}) \| \le \text{ prescribed tolerance,}$$

stop; accept x^{k+1} as an approximate solution of the CE (Ω, H). Otherwise, return to Step 1 with k replaced by $k + 1$.

By (A3) and the fact that $x^k \in \Omega_{\mathcal{I}}$, the Newton equation (3) has a unique solution which we have denoted by d^k. The following lemma guarantees that d^k is a descent direction for the function ψ at x^k. This property, along with the openness of $\Omega_{\mathcal{I}}$, ensures that the integer m_k can be determined in a finite number of trials (starting with $m_k = 0$ and increasing it by one at each trial), thus guaranteeing the well-definedness of the next iterate x^{k+1}.

LEMMA 2. *Suppose that conditions* (A5) *and* (A6) *hold. Assume also that* $x \in \Omega_{\mathcal{I}}$, $d \in \Re^n$, *and* $\sigma \in \Re$ *are such that*

$$H(x) \ne 0, \qquad 0 \le \sigma < \bar{\sigma}, \tag{4}$$

$$H'(x; d) = -H(x) + \sigma \frac{a^T H(x)}{\|a\|^2} a, \tag{5}$$

where $a \in \Re^n$ *and* $\bar{\sigma} \in [0,1]$ *are as in condition* (A6). *Then,* $\nabla\psi(x)^T d < 0$.

Proof. Let $u \equiv H(x)$. Then, $0 \neq u \in \text{int } S$ due to (4) and the assumption that $x \in \Omega_{\mathcal{I}}$. This together with (2), (5), (4), (A5), and (A6) imply

$$\begin{aligned}\nabla\psi(x)^T d = \nabla p(H(x))^T H'(x; d) \ &= \ \nabla p(u)^T \left(-u + \sigma \, \frac{a^T u}{\|a\|^2} \, a \right) \\ &\leq -\nabla p(u)^T u \left(1 - \frac{\sigma}{\bar{\sigma}} \right) < 0,\end{aligned}$$

as claimed. □

2.4. A convergence result. In what follows, we state and prove a limiting property of an infinite sequence of iterates $\{x^k\}$ generated by the algorithm. Before stating the theorem, we observe that such a sequence necessarily belongs to the set $\Omega_{\mathcal{I}}$; thus $\{H(x^k)\} \subset \text{int } S$. Since the sequence $\{x^k\}$ is infinite, we have $H(x^k) \neq 0$ for all k. Theorem 3 below contains four conclusions, (a)–(d). The first three of these do not assert the boundedness of the sequence $\{x^k\}$; this boundedness is established under the assumptions of statement (d), which implies the existence of a solution of the CE (Ω, H). A consequence of statement (c) in the theorem is

$$\inf \{ \, \| \, H(x) \, \| \, : \, x \, \in \, \Omega \, \} \, = \, 0;$$

consequently, CE (Ω, H) has "ε-solutions" for every $\varepsilon > 0$ in the sense that for any such ε, there exists a vector $x^\varepsilon \in \Omega$ satisfying $\| \, H(x^\varepsilon) \, \| \, \leq \, \varepsilon$; moreover x^ε can be computed by the potential reduction Newton method starting at the given vector x^0.

THEOREM 3. *Assume conditions* (A1)–(A6) *hold and that* $\limsup_k \sigma_k < \bar{\sigma}$. *Let* $\{x^k\}$ *be any infinite sequence produced by the potential reduction Newton algorithm. Then, the following statements hold:*

(a) *the sequence* $\{H(x^k)\}$ *is bounded;*

(b) *any accumulation point of* $\{x^k\}$*, if it exists, solves the CE* (Ω, H)*; in particular, if* $\{x^k\}$ *is bounded, then the CE* (Ω, H) *has a solution.*

Moreover, for any closed subset E *of* S *containing the sequence* $\{H(x^k)\}$,

(c) *if* H *is proper with respect to* $E \cap \text{int } S$*, then* $\lim_{k\to\infty} H(x^k) \, = \, 0$*;*

(d) *if* H *is proper with respect to* E*, then* $\{x^k\}$ *is bounded.*

Proof. Let $\gamma \equiv \psi(x^0)$ and $u^k \equiv H(x^k) \in \text{int } S$ for all k. Clearly, $p(u^k) = \psi(x^k) \leq \psi(x^0) = \gamma$ for all k. Hence, for any $\varepsilon > 0$ we have $\{u^k\} \subset \Lambda(\varepsilon, \gamma) \cup \{u \in \Re^n : \|u\| \leq \varepsilon\}$. Since by Lemma 1 the set $\Lambda(\varepsilon, \gamma)$ is compact, and hence bounded, we conclude that $\{u^k\}$ is bounded. Hence, (a) follows.

To show (b), let x^∞ be an accumulation point of $\{x^k\}$. Clearly $x^\infty \in \Omega$ because Ω is a closed set. Assume for contradiction that $u^\infty \equiv H(x^\infty) \neq 0$. Let $\{x^k : k \in \kappa\}$ be a subsequence converging to x^∞ and assume without loss of generality that $\{\sigma_k : k \in \sigma\}$ converges to some scalar σ_∞. Since $\sigma_k \geq 0$ for all k and $\limsup_k \sigma_k < \bar{\sigma}$, we must have $\sigma_\infty \in [0, \bar{\sigma})$. Since $p(u^k) \leq p(u^0) = \gamma$ for all k and

$$\lim_{k(\in\kappa)\to\infty} u^k \, = \, u^\infty \, \neq \, 0,$$

there exists $\varepsilon > 0$ such that the subsequence $\{u^k : k \in \kappa\} \subset \Lambda(\varepsilon, \gamma)$. Since by Lemma 1 the set $\Lambda(\varepsilon, \gamma)$ is compact, we conclude that $u^\infty = H(x^\infty) \in \Lambda(\varepsilon, \gamma) \subset \text{int } S$, and hence that $x^\infty \in H^{-1}(\text{int } S)$. By assumption (A2), it follows that $x^\infty \in \Omega_{\mathcal{I}}$. Hence, by assumption (A3), $H'(x^\infty)^{-1}$ exists. This implies that the sequence $\{d^k : k \in \kappa\}$ converges to a vector d^∞ satisfying

$$H(x^\infty) + H'(x^\infty; d^\infty) \, = \, \sigma_\infty \, \frac{a^T H(x^\infty)}{\|a\|^2} \, a.$$

Hence, it follows from Lemma 2 that $\nabla\psi(x^\infty)^T d^\infty < 0$.

Since $\{x^k : k \in \kappa\}$ converges to $x^\infty \in \Omega_{\mathcal{I}}$ where ψ is continuous, it follows that $\{\psi(x^k) : k \in \kappa\}$ converges. This implies that the whole sequence $\{\psi(x^k)\}$ converges due to the fact that it is monotonically decreasing. Using the relation

$$\psi(x^{k+1}) - \psi(x^k) \,=\, \psi(x^k + \rho^{m_k} d^k) - \psi(x^k) \,\le\, \alpha\, \rho^{m_k}\, \nabla\psi(x^k)^T d^k \,<\, 0$$

for all k, we conclude that

$$\lim_{k\to\infty} \rho^{m_k}\, \nabla\psi(x^k)^T d^k \,=\, 0$$

and hence that

$$\lim_{k(\in\kappa)\to\infty} \rho^{m_k} \,=\, 0$$

because

$$\lim_{k(\in\kappa)\to\infty} \nabla\psi(x^k)^T d^k \,=\, \nabla\psi(x^\infty)^T d^\infty \,<\, 0.$$

Thus

$$\lim_{k(\in\kappa)\to\infty} m_k \,=\, \infty,$$

which implies that $m_k \ge 2$ for all $k \in \kappa$ sufficiently large. Consequently, by the definition of m_k, we deduce that

$$\frac{\psi(x^k + \rho^{m_k-1} d^k) - \psi(x^k)}{\rho^{m_k-1}} \,>\, \alpha\, \nabla\psi(x^k)^T d^k$$

for all $k \in \kappa$ sufficiently large. Letting $k \in \kappa$ tend to infinity in the above expression, we obtain

$$\nabla\psi(x^\infty)^T d^\infty \,\ge\, \alpha \nabla\psi(x^\infty)^T d^\infty,$$

which contradicts the fact that $\alpha < 1$ and $\nabla\psi(x^\infty)^T d^\infty < 0$. Consequently, we must have $H(x^\infty) = 0$, and hence (b) follows.

Assume now that E is a closed subset of S containing the sequence $\{H(x^k)\}$. To prove (c), assume for contradiction that for an infinite subset $\kappa \subset \{0, 1, 2, \ldots\}$, we have

$$\liminf_{k(\in\kappa)\to\infty} \|\, u^k \,\| \,>\, 0.$$

By an argument similar to that employed above, we conclude that for some $\varepsilon > 0$ we have $\{u^k : k \in \kappa\} \subset \Lambda(\varepsilon, \gamma) \cap E$. By Lemma 1 and the fact that E is closed, we conclude that $\Lambda(\varepsilon, \gamma) \cap E$ is a compact subset of int $S \cap E$. Since H is proper with respect to int $S \cap E$, the inverse image of $\Lambda(\varepsilon, \gamma) \cap E$ under H is compact, and hence bounded. This implies that $\{x^k : k \in \kappa\}$ is bounded. By (b), every accumulation point of the latter subsequence is a zero of H. This contradiction establishes (c).

Finally, using (a) and the fact that E is closed, we conclude that $\{u^k\}$ is contained in a compact subset E_1 of E. Since H is proper with respect to E, it follows that the set $H^{-1}(E_1) \supset \{x^k\}$ is bounded. Hence, (d) follows. □

The framework of the CE (Ω, H) that we have set forth so far is very broad. In addition to not assuming any sign restriction on the components of H (like we did in [36]; see Assumption 1 therein), as we have mentioned before, the freedom in the choice of the set S and the associated potential function p and vector a adds to the versatility of the framework. The results in the next two sections will demonstrate how S, p, and a can easily be constructed in important cases under very mild assumptions.

3. Monotone complementarity problems in symmetric matrices. We consider a mixed complementarity problem defined on the cone of symmetric positive semidefinite matrices. The linear version of this problem was introduced by Kojima, Shindoh, and Hara [10] and has received a great deal of research attention recently. In what follows, we consider a nonlinear version of this problem defined in [18]. This reference contains a fairly extensive bibliography on interior point methods for solving optimization and complementarity problems defined on the cone of semidefinite matrices; it will be the source for several results that will be used freely in the subsequent development.

3.1. Implicit mixed complementarity problems. We recall the framework considered in [18]. Let $F : \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^m \to \mathcal{S}^n \times \Re^m$ be a given mapping. The mixed complementarity problem in symmetric matrices is to find a triple $(X, Y, z) \in \mathcal{S}^n \times \mathcal{S}^n \times \Re^m$ satisfying

$$F(X,Y,z) = 0, \quad X \bullet Y = 0, \quad (X,Y) \in \mathcal{S}^n_+ \times \mathcal{S}^n_+. \tag{6}$$

As explained in [18] and the references therein, there are several equivalent ways of stating the complementarity condition $X \bullet Y = 0$, each leading to a different interior point method for solving the above problem. In what follows, we consider the equivalent formulation of this problem as the CE defined by the pair (Ω, H), where the set Ω and the map $H : \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^m \to \mathcal{S}^n \times \mathcal{S}^n \times \Re^m$ are defined by

$$\Omega \equiv \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^m, \tag{7}$$

$$H(X,Y,z) \equiv \begin{pmatrix} (XY+YX)/2 \\ F(X,Y,z) \end{pmatrix}, \quad (X,Y,z) \in \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^m. \tag{8}$$

Similar treatment can be applied to other equivalent formulations and to generalizations of the basic problem (6). Throughout the following discussion, F is assumed to be continuous on its domain and continuously differentiable on $\mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \times \Re^m$.

Associated with the above mapping H, define the set

$$\mathcal{U} \equiv \{ (X,Y) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} : XY + YX \in \mathcal{S}^n_{++} \}. \tag{9}$$

The set $\mathcal{U}$ was introduced in [31] and subsequently used in the papers [8, 33] for the analysis of primal-dual semidefinite programming algorithms based on the Alizadeh–Haeberly–Overton (AHO) direction [2]. It has also been used in [18] for the study of the fundamental properties of the interior point map (8). The fundamental role of the set $\mathcal{U}$ in the study of the problem (6) is well explained in the above-cited references. It has been shown in Lemma 1 of [18] that

$$\mathcal{U} = \{ (X,Y) \in \mathcal{S}^n_+ \times \mathcal{S}^n_+ : XY + YX \in \mathcal{S}^n_{++} \}. \tag{10}$$

We introduce an important assumption on the mapping F that will be used to verify the nonsingularity of the Jacobian matrix $H'(X,Y,z)$.

(B1) The mapping F is (X,Y)-*differentiably-monotone* at every triple $(X,Y,z) \in \mathcal{U} \times \Re^m$; i.e., for any such triple,

$$\left. \begin{array}{c} F'(\,(X,Y,z);\,(dX,dY,dz)\,) = 0 \\ (dX,dY,dz) \in \mathcal{S}^n \times \mathcal{S}^n \times \Re^m \end{array} \right\} \Longrightarrow dX \bullet dY \geq 0. \tag{11}$$

(B2) The mapping F is *z-differentiably-injective* at every triple $(X, Y, z) \in \mathcal{U} \times \Re^m$; i.e., for any such triple,

$$F'(\,(X, Y, z);\ (0, 0, dz)\,) \;=\; 0 \;\Longrightarrow\; dz \;=\; 0. \tag{12}$$

The following lemma asserts that the basic assumptions (A1)–(A3) in section 2.1 are valid under the above hypotheses.

LEMMA 3. *Consider the CE* (Ω, H) *with* Ω *and* H *defined by* (7) *and* (8), *and let* $S \equiv \mathcal{S}^n_+ \times \mathcal{S}^n \times \Re^m$. *If conditions* (B1) *and* (B2) *hold, then*

$$\Omega_{\mathcal{I}} \;\equiv\; H^{-1}(\text{int } S) \cap \text{int } \Omega \;=\; \mathcal{U} \times \Re^m;$$

moreover, the pair (Ω, H) *and the set* S *satisfy conditions* (A1), (A2), *and* (A3).

Proof. Only the second assertion requires a proof. Conditions (A1) and (A2)(a) obviously hold. Clearly $\mathcal{U}$ is an open set; since $(I, I) \in \mathcal{U}$, (A2)(b) holds. Moreover, it is easy to see that the alternative representation (10) implies (A2)(c). Next we establish that (A3) holds under (B1) and (B2). This amounts to showing that for every $(X, Y, z) \in \Omega_{\mathcal{I}} = \mathcal{U} \times \Re^m$, the following implication holds:

$$\left.\begin{array}{r} H'(\,(X, Y, z);\ (dX, dY, dz)\,) \;=\; 0 \\ (dX, dY, dz) \;\in\; \mathcal{S}^n \times \mathcal{S}^n \times \Re^m \end{array}\right\} \;\Longrightarrow\; (dX, dY, dz) \;=\; 0.$$

Assume the left-hand condition holds. Then,

$$X(dY) + (dY)X + Y(dX) + (dX)Y \;=\; 0, \tag{13}$$

$$F'(\,(X, Y, z);\ (dx, dy, dz)\,) \;=\; 0. \tag{14}$$

Condition (B1) and (14) imply that $dX \bullet dY \geq 0$. This together with (13) and the fact that $(X, Y) \in \mathcal{U}$ yield $dX = dY = 0$ (see the proof of Theorem 3.1(iii) of [31]). In turn, this together with (14) imply

$$F'(\,(X, Y, z);\ (0, 0, dz)\,) \;=\; 0,$$

which yields $dz = 0$ due to (B2). □

From the above result, we see that the set $\mathcal{U}$ is naturally associated with the map H given by (8). We observe that, based on the analysis of Monteiro and Zanjácomo [21], it can be shown that $H'(X, Y, z)$ is invertible over the set $\mathcal{U}' \times \Re^m$ with $\mathcal{U}'$ given by

$$\mathcal{U}' \;\equiv\; \left\{ (X, Y) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \;:\; \left\| X^{1/2} Y X^{1/2} - \mu I \right\| \;\leq\; \frac{1}{2}\,\mu \right\},$$

where $\mu \equiv (X \bullet Y)/n$. However, the set $\mathcal{U}'$ does not fit well with the map H in the sense of Lemma 3 even for different choices of the set S. Instead, $\mathcal{U}'$ naturally arises in connection with the interior point map $\tilde{H}(X, Y, z) \equiv (X^{1/2} Y X^{1/2}, F(X, Y, z))$ by choosing the set S as

$$S \;\equiv\; \left\{ U \in \mathcal{S}^n_{++} \;:\; \left\| U - \left(\frac{\text{tr}\, U}{n} \right) I \right\|_F \;\leq\; \frac{1}{2}\, \frac{\text{tr}\, U}{n} \right\}.$$

Even though this provides a viable alternative approach, we will not pursue it any further.

Next we deal with conditions (A4)–(A6). For this purpose, consider the potential function $p : \mathcal{S}^n_{++} \times \mathcal{S}^n \times \Re^m \to \Re$ defined by

$$p(M,N,v) \equiv \zeta \log \left(\|M\|_F^2 + \|N\|_F^2 + \|v\|^2 \right) - \log(\det M) \tag{15}$$

for every $(M,N,z) \in \mathcal{S}^n_{++} \times \mathcal{S}^n \times \Re^m$, where $\zeta > n/2$ is an arbitrary constant.

LEMMA 4. *The potential function* (15), *the vector* $a \equiv (I,0,0) \in \mathcal{S}^n \times \mathcal{S}^n \times \Re^m$, *and the scalar* $\bar{\sigma} \equiv 1$ *satisfy conditions* (A4), (A5), *and* (A6).

Proof. Since, for a matrix $Z \in \mathcal{S}^n$, $\|Z\|_F^2$ is equal to the sum of the squares of the n eigenvalues of Z, and $\det Z$ is equal to the product of these eigenvalues, the verification of (A4) for the function $p(M,N,v)$ is the same as in the previous case of a nonnegatively constrained equation (discussed at the end of section 2.1). Noting that

$$\nabla p(M,N,v) = \begin{pmatrix} \dfrac{2\zeta}{\|M\|_F^2 + \|N\|_F^2 + \|v\|^2} M - M^{-1} \\ \dfrac{2\zeta}{\|M\|_F^2 + \|N\|_F^2 + \|v\|^2} N \\ \dfrac{2\zeta}{\|M\|_F^2 + \|N\|_F^2 + \|v\|^2} v \end{pmatrix},$$

we have

$$(M,N,v) \bullet \nabla p(M,N,v) = 2\zeta - n > 0,$$

and thus (A5) holds. We now show that (A6) is satisfied with the given a and $\bar{\sigma}$. Indeed we have

$$(I,0,0) \bullet \nabla p(M,N,v) = \frac{2\zeta}{\|M\|_F^2 + \|N\|_F^2 + \|v\|^2} \operatorname{tr}(M) - \operatorname{tr}(M^{-1}),$$

which implies

$$\begin{aligned} &[\,(I,0,0) \bullet \nabla p(M,N,v)\,]\,[\,(I,0,0) \bullet (M,N,v)\,] \\ &= \frac{2\zeta}{\|M\|_F^2 + \|N\|_F^2 + \|v\|^2} (\operatorname{tr}(M))^2 - \operatorname{tr}(M^{-1}) \operatorname{tr}(M). \end{aligned}$$

Noting that (i) $\operatorname{tr}(M)$ equals the sum of the eigenvalues of M, (ii) $\operatorname{tr}(M^{-1})$ equals the sum of the inverses of the same eigenvalues, and (iii) $\|M\|_F^2 = \operatorname{tr}(M^2)$ equals the sum of these eigenvalues squared, it follows from the same derivation as at the end of section 2.1 that condition (A6) holds. □

According to (2), the potential function (15) induces the following merit function on the set $\Omega_{\mathcal{I}} = \mathcal{U} \times \Re^m$:

$$\begin{aligned} \psi(X,Y,z) &\equiv p(H(X,Y,z)) \\ &= \zeta \log \left(\frac{\|\, XY + YX \,\|_F^2}{4} + \|\, F(X,Y,z) \,\|_{F,2}^2 \right) - \log \left(\det \left(\frac{XY + YX}{2} \right) \right), \end{aligned}$$

for any triple $(X,Y,z) \in \mathcal{U} \times \Re^m$. Here, $\|\cdot\|_{F,2}$ denotes the norm on $\mathcal{S}^n \times \Re^m$ defined by $\|(N,v)\|_{F,2}^2 \equiv \|N\|_F^2 + \|v\|^2$ for every $(N,v) \in \mathcal{S}^n \times \Re^m$.

We now give a detailed description of a specialized algorithm for solving the mixed complementarity problem in symmetric matrices (6), based on the potential reduction Newton method for solving the CE (Ω, H) with Ω, H, S, $p : \operatorname{int} S \to \Re$, a and $\bar{\sigma}$ defined as in (7), (8), Lemma 3, (15), and Lemma 4, respectively.

Step 0. (Initialization) Let a pair of matrices $(X^0, Y^0) \in \mathcal{U}$, a vector $z^0 \in \Re^m$, and scalars $\rho \in (0,1)$ and $\alpha \in (0,1)$ be given. Let a sequence of scalars $\{\sigma_k\}$ also be given, where $\sigma_k \in [0,1)$ for all k. Set the iteration counter $k = 0$.

Step 1. (Computing the modified Newton direction) Solve the system of linear equations:

$$\begin{pmatrix} \left[X^kY^k + Y^kX^k + X^k(dY) + (dY)X^k + Y^k(dX) + (dX)Y^k\right]/2 \\ F(X^k, Y^k, z^k) + F'((X^k, Y^k, z^k); (dX, dY, dz)) \end{pmatrix} = \begin{pmatrix} \sigma_k \mu_k I \\ 0 \end{pmatrix}$$

$$(dX, dY, dz) \in \mathcal{S}^n \times \mathcal{S}^n \times \Re^m,$$

where $\mu_k \equiv \operatorname{tr}(X^kY^k)/n$, to obtain the search triple (dX^k, dY^k, dz^k).

Step 2. (Armijo line search) Let m_k be the smallest nonnegative integer m such that

$$\begin{pmatrix} X^k + \rho^m \, dX^k \\ Y^k + \rho^m \, dY^k \end{pmatrix} \in \Omega_{\mathcal{I}}$$

and

$$\begin{aligned} &\psi(X^k + \rho^m dX^k, Y^k + \rho^m dY^k, z^k + \rho^m dz^k) - \psi(X^k, Y^k, z^k) \\ &\qquad \le \alpha \, \rho^m \, \psi'(\,(X^k, Y^k, dz^k);\, (dX^k, dY^k, dz^k)\,). \end{aligned}$$

Set

$$\begin{pmatrix} X^{k+1} \\ Y^{k+1} \\ dz^{k+1} \end{pmatrix} \equiv \begin{pmatrix} X^k + \rho^{m_k} \, dX^k \\ Y^k + \rho^{m_k} \, dY^k \\ z^k + \rho^{m_k} \, dz^k \end{pmatrix}.$$

Step 3. (Termination test) If

$$\| \, H(X^{k+1}, Y^{k+1}, z^{k+1}) \, \| \le \text{ prescribed tolerance},$$

stop; accept the triple $(X^{k+1}, Y^{k+1}, z^{k+1})$ as an approximate solution of the problem (6). Otherwise, return to Step 1 with k replaced by $k + 1$.

We observe that the direction obtained in Step 1 of the above algorithm is an extension of the AHO direction introduced in [2] in the context of semidefinite programming.

As an immediate consequence of Lemma 3, Lemma 4, and Theorem 3, we have the following convergence result for the above algorithm.

THEOREM 4. *Assume that conditions* (B1) *and* (B2) *hold and* $\limsup_k \sigma_k < 1$. *Let* $\{(X^k, Y^k, z^k)\}$ *be any infinite sequence produced by the above algorithm for solving problem* (6). *Then, the following statements hold:*

(a) *the sequence* $\{H(X^k, Y^k, z^k)\}$ *is bounded;*

(b) *any accumulation point of* $\{(X^k, Y^k, z^k)\}$*, if it exists, solves the problem* (6)*; in particular, if* $\{(X^k, Y^k, z^k)\}$ *is bounded, then problem* (6) *has a solution.*

We now make a few remarks. The above theorem guarantees neither that $\{(X^k, Y^k, z^k)\}$ is bounded nor that it has an accumulation point. The conclusion that $\{(X^k, Y^k, z^k)\}$ is bounded would follow from Theorem 3(d) with $E = S$ if we could prove that the map H is proper with respect to the set $S \equiv \mathcal{S}^n_+ \times \mathcal{S}^n \times \Re^m$. Unfortunately, this requirement is rather strong. For monotone mixed complementarity problems, we state in Proposition 2 below a result (from Monteiro and Pang [18, Lemma 2]) asserting that the map H is proper with respect to $\mathcal{S}^n \times F(\mathcal{U} \times \Re^m)$. Hence, if the latter set contains the set $S = \mathcal{S}^n_+ \times \mathcal{S}^n \times \Re^m$, or equivalently if the equality $F(\mathcal{U} \times \Re^m) = \mathcal{S}^n \times \Re^m$ holds, then the sequence generated by the above algorithm $\{(X^k, Y^k, z^k)\}$ is bounded. Intuitively, the equality $F(\mathcal{U} \times \Re^m) = \mathcal{S}^n \times \Re^m$ might hold for maps F satisfying some kind of strong monotonicity condition. But since this type of condition is fairly restrictive, we do not pursue this issue any further.

Another possible approach which would guarantee the boundedness of $\{(X^k, Y^k, z^k)\}$ is to reduce the set S so as to have $S \subset \mathcal{S}^n \times F(\mathcal{U} \times \Re^m)$. This approach requires some knowledge of the set $F(\mathcal{U} \times \Re^m)$. We will see that for the complementarity problems studied in sections 3.2 and 4, enough information about the set $F(\mathcal{U} \times \Re^m)$ is available to allow us to choose a set S together with a potential function $p : \text{int } S \to \Re$ satisfying the inclusion $S \subset \mathcal{S}^n \times F(\mathcal{U} \times \Re^m)$ and the conditions (A1)–(A6) of section 2.1.

Before stating the properness result mentioned above, we give a few basic definitions.

Definition 1. *A mapping* $J(X, Y, z)$ *defined on a subset* $\text{dom}(J)$ *of* $\mathcal{M}^n \times \mathcal{M}^n \times \Re^m$ *is said to be* (X, Y)-equilevel-monotone *on a subset* $\mathcal{V} \subset \text{dom}(J)$ *if for any* $(X, Y, z) \in \mathcal{V}$ *and* $(X', Y', z') \in \mathcal{V}$ *such that* $J(X, Y, z) = J(X', Y', z')$*, there holds* $(X' - X) \bullet (Y' - Y) \geq 0$*. When* $\mathcal{V} = \text{dom}(J)$*, we will simply say that* J *is* (X, Y)*-equilevel-monotone.*

In the following two definitions, we assume that W, Z, and N are three normed spaces and that $\phi(w, z)$ is a function defined on a subset of $W \times Z$ with values in N.

Definition 2. *The function* $\phi(w, z)$ *is said to be* z-bounded *on a subset* $\mathcal{V} \subset \text{dom}(\phi)$ *if for every sequence* $\{(w^k, z^k)\} \subset \mathcal{V}$ *such that* $\{w^k\}$ *and* $\{\phi(w^k, z^k)\}$ *are bounded, the sequence* $\{z^k\}$ *is also bounded. When* $\mathcal{V} = \text{dom}(\phi)$*, we will simply say that* ϕ *is* z*-bounded.*

Definition 3. *The function* $\phi(w, z)$ *is said to be* z-injective *on a subset* $\mathcal{V} \subset \text{dom}(\phi)$ *if the following implication holds*: $(w, z) \in \mathcal{V}$, $(w, z') \in \mathcal{V}$*, and* $\phi(w, z) = \phi(w, z')$ *implies* $z = z'$*. When* $\mathcal{V} = \text{dom}(\phi)$*, we will simply say that* ϕ *is* z*-injective.*

The following is the promised result from Lemma 2 of Monteiro and Pang [18].

Proposition 2. *Let* $F : \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^m \to \mathcal{S}^n \times \Re^m$ *be a continuous map and let* $H : \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^m \to \mathcal{S}^n \times \mathcal{S}^n \times \Re^m$ *be the map defined by* (8)*. Assume that the map* F *is* (X, Y)*-equilevel-monotone and* z*-bounded on its domain. If the map* H *restricted to* $\mathcal{U} \times \Re^m$ *is a local homeomorphism, then* H *is proper with respect to* $\mathcal{S}^n \times F(\mathcal{U} \times \Re^m)$*.*

3.2. Standard complementarity problem. In this section, we consider the standard nonlinear complementarity problem (NCP) in symmetric matrices:

$$X \bullet f(X) = 0, \quad X \succeq 0, \quad f(X) \succeq 0, \tag{16}$$

where $f : \mathcal{S}^n_+ \to \mathcal{S}^n$ is a given continuous mapping that is continuously differentiable on $\mathcal{S}^n_{++}$. This problem is a special case of the implicit mixed complementarity problem of section 3.1, where $m = 0$ (i.e., the free variable z is not present) and $F : \mathcal{S}^n_+ \times \mathcal{S}^n_+ \to \mathcal{S}^n$ is given by

$$F(X, Y) \equiv Y - f(X) \quad \forall (X, Y) \in \mathcal{S}^n_+ \times \mathcal{S}^n_+. \tag{17}$$

We make the following assumption on the mapping f.

(C1) $f : \mathcal{S}^n_+ \to \mathcal{S}^n$ is monotone on $\mathcal{S}^n_+$; i.e., for all X and X' in $\mathcal{S}^n_+$,

$$(X - X') \bullet (f(X) - f(X')) \geq 0.$$

LEMMA 5. *If condition* (C1) *holds, then the map* $F : \mathcal{S}^n_+ \times \mathcal{S}^n_+ \to \mathcal{S}^n$ *defined by* (17) *satisfies condition* (B1) *of section* 3.1.

Proof. By (C1), it follows that for every $X \in \mathcal{S}^n_+$, the linear map $f'(X)$ is monotone in the sense that

$$U \bullet f'(X; U) \geq 0 \quad \forall U \in \mathcal{S}^n. \tag{18}$$

To verify (B1), assume that $(dX, dY) \in \mathcal{S}^n \times \mathcal{S}^n$ satisfies $F'(X, Y)(dX, dY) = 0$, or equivalently that $dY - f'(X; dX) = 0$. Then, by (18), we have

$$dX \bullet dY = dX \bullet f'(X; dX) \geq 0.$$

This shows that implication (11) holds for $m = 0$, and since implication (12) holds vacuously for $m = 0$, (C1) follows. □

It is possible to solve the NCP (16) with the use of the potential reduction algorithm described in section 3.1. However, the sequence of iterates $\{(X^k, Y^k)\}$ generated by this algorithm might not be bounded. We now develop a different potential reduction algorithm in which the set S is reduced so as to satisfy $S \subset \mathcal{S}^n_+ \times F(\mathcal{U})$, thus ensuring the boundedness of the sequence $\{(X^k, Y^k)\}$ (see the discussion at the end of the previous subsection).

To describe the alternative algorithm, it is sufficient to identify the pair (Ω, H), the set S, the potential function $p : \text{int } S \to \Re$, and the vector a and scalar $\bar{\sigma}$ in condition (A6). We let $\Omega \equiv \mathcal{S}^n_+ \times \mathcal{S}^n_+$ and define $H : \mathcal{S}^n_+ \times \mathcal{S}^n_+ \to \mathcal{S}^n \times \mathcal{S}^n$ by

$$H(X, Y) \equiv \begin{pmatrix} (XY + YX)/2 \\ F(X, Y) \end{pmatrix}, \quad (X, Y) \in \mathcal{S}^n_+ \times \mathcal{S}^n_+, \tag{19}$$

where F is given by (17). Moreover, we let $S \equiv \mathcal{S}^n_+ \times \mathcal{S}^n_+$ and $p : \text{int } S \to \Re$ be defined by

$$p(M, N) \equiv \zeta \log \left(\|M\|_F^2 + \|N\|_F^2 \right) - \log(\det M) - \log(\det N)$$

for every $(M, N) \in \mathcal{S}^n_{++} \times \mathcal{S}^n_{++}$, where $\zeta > n$ is an arbitrary constant. Finally, we let $a \equiv (I, I)$ and $\bar{\sigma} \equiv 1$. Clearly, the set $\Omega_{\mathcal{I}}$ and the merit function $\psi : \Omega_{\mathcal{I}} \to \Re$ become

$$\Omega_{\mathcal{I}} = \{ (X, Y) \in \mathcal{U} : Y \succ f(X) \}$$

and

$$\psi(X, Y) \equiv \zeta \log \left(\frac{\|XY + YX\|_F^2}{4} + \| Y - f(X) \|^2 \right)$$
$$- \log \left(\det \left(\frac{XY + YX}{2} \right) \right) - \log(\det(Y - f(X))) \text{ for } (X, Y) \in \Omega_{\mathcal{I}}.$$

Lemma 6. *The pair* (Ω, H)*, the set* S*, the potential function* $p : \operatorname{int} S \to \Re$*, the vector* a*, and the scalar* $\bar{\sigma}$ *defined above satisfy conditions* (A1)–(A6) *of section* 2.1.

Proof. Condition (A2)(b) follows from the fact that $(I, \delta I) \in \Omega_{\mathcal{I}}$ for all $\delta > 0$ sufficiently large. The other conditions are either straightforward or are shown using Lemma 5 and the same arguments used in the proofs of Lemmas 3 and 4. □

Before giving the convergence result for the potential reduction Newton method in the above framework, we state the following result, which will be used to establish boundedness of the iterates generated by this method.

Lemma 7. *Suppose that* $f : \mathcal{S}^n_+ \to \mathcal{S}^n$ *is a continuous map that is continuously differentiable on* $\mathcal{S}^n_{++}$ *and satisfies condition* (C1)*. Then, for the maps* F *and* H *defined by* (17) *and* (19)*, respectively, we have*

(a) $F(\mathcal{U}) = F(\mathcal{S}^n_{++} \times \mathcal{S}^n_{++})$*;*

(b) *if* $0 \in F(\mathcal{S}^n_+ \times \mathcal{S}^n_+)$*, then* H *is proper with respect to* $\mathcal{S}^n \times \mathcal{S}^n_{++}$*;*

(c) *if* $0 \in F(\mathcal{S}^n_{++} \times \mathcal{S}^n_{++})$*, then* H *is proper with respect to* $\mathcal{S}^n \times \mathcal{S}^n_+$*.*

Proof. By Proposition 4(a) and Corollary 3 of [18] with $m = 0$, it follows that $\{I\} \times F(\mathcal{S}^n_{++} \times \mathcal{S}^n_{++}) \subset H(\mathcal{S}^n_+ \times \mathcal{S}^n_+)$. Using this inclusion, we easily see that statement (a) holds.

We next show (b). By Lemma 6, $H'(X, Y)$ is invertible for all $(X, Y) \in \mathcal{U}$. Thus H restricted to $\mathcal{U}$ is a local homeomorphism. Thus it follows from Lemma 2 that H is proper with respect to $\mathcal{S}^n \times F(\mathcal{U})$. Hence, (b) follows once we prove that $\mathcal{S}^n_{++} \subset F(\mathcal{U}) = F(\mathcal{S}^n_{++} \times \mathcal{S}^n_{++})$. Let $U \in \mathcal{S}^n_{++}$ be arbitrary. Since $0 \in F(\mathcal{S}^n_+ \times \mathcal{S}^n_+)$, there exists $(\tilde{X}, \tilde{Y}) \in \mathcal{S}^n_+ \times \mathcal{S}^n_+$ such that $\tilde{Y} = f(\tilde{X})$. For $\epsilon > 0$, let $X_\epsilon \equiv \tilde{X} + \epsilon I$ and $Y_\epsilon \equiv U + f(X_\epsilon) = U + \tilde{Y} + f(X_\epsilon) - f(\tilde{X})$. Clearly, $X_\epsilon \succ 0$ for every $\epsilon > 0$. By the continuity of f and the fact that $U + \tilde{Y} \succ 0$, we have $Y_\epsilon \succ 0$ for $\epsilon > 0$ sufficiently small. Since $U = Y_\epsilon - f(X_\epsilon)$, it follows that U belongs to $F(\mathcal{S}^n_{++} \times \mathcal{S}^n_{++})$.

We omit the proof of (c), which is similar to that of (b). □

We will skip the straightforward formulation of the potential reduction Newton method specialized to the above choices of the pair (Ω, H), set S, potential function $p : \operatorname{int} S \to \Re$, vector a, and scalar $\bar{\sigma}$; instead, we directly give the convergence properties of the method. Among the three conclusions (a), (b), and (c) of Theorem 5, (b) provides a constructive proof that a feasible monotone complementarity problem in symmetric matrices on the positive semidefinite cone always has "ε-solutions"; (c) implies the well-known fact that for such a problem, strict feasibility yields solvability.

Theorem 5. *Let* $f : \mathcal{S}^n_+ \to \mathcal{S}^n$ *be a continuous function which is continuously differentiable on* $\mathcal{S}^n_{++}$ *and satisfies condition* (C1)*. Suppose that* $\{(X^k, Y^k)\}$ *is a sequence generated by the potential reduction Newton method with the pair* (Ω, H)*, set* S*, potential function* $p : \operatorname{int} S \to \Re$*, vector* a*, and scalar* $\bar{\sigma}$ *as specified above. Then, the following statements hold:*

(a) *every accumulation point of* $\{(X^k, Y^k)\}$ *is a solution of the NCP* (16)*;*

(b) *if there exists* $\tilde{X} \in \mathcal{S}^n_+$ *such that* $f(\tilde{X}) \in \mathcal{S}^n_+$*, then* $\lim_{k\to\infty} H(X^k, Y^k) = 0$*;*

(c) *if there exists* $\hat{X} \in \mathcal{S}^n_{++}$ *such that* $f(\hat{X}) \in \mathcal{S}^n_{++}$*, then the sequence* $\{(X^k, Y^k)\}$ *is bounded.*

Proof. Statement (a) follows from Theorem 3(b). To prove statement (b), note first that the assumption implies that $0 \in F(\mathcal{S}^n_+ \times \mathcal{S}^n_+)$. Hence, by Lemma 7(b), we conclude that H is proper with respect to $\mathcal{S}^n \times \mathcal{S}^n_{++}$. It follows from Theorem 3(c) with $E = S$ that $\{H(X^k, Y^k)\}$ converges to zero. The proof of (c) follows similarly from Lemma 7(c) and Theorem 3(d) with $E = S$. □

Statement (a) is within expectation; statement (b) is interesting because its assumption is the feasibility of the NCP in symmetric matrices (16). A consequence of statement (b) is that feasibility of this problem (which is also monotone by assumption (C1)) is sufficient for the sequence $\{H(X^k, Y^k)\}$ to converge to zero, although no boundedness of the sequence $\{(X^k, Y^k)\}$ is asserted. The latter assertion is established under the strict feasibility of the problem (16); this is statement (c).

4. Convex semidefinite programs. In this section we consider the convex semidefinite program studied in [18, 30], namely,

$$\begin{aligned} &\text{minimize} && \theta(x) \\ &\text{subject to} && G(x) \preceq 0, \\ &&& h(x) = 0, \end{aligned} \tag{20}$$

where $\theta : \Re^m \to \Re$, $G : \Re^m \to \mathcal{S}^n$, and $h : \Re^m \to \Re^p$ are given smooth mappings. Under a suitable constraint qualification, if x^* is a locally optimal solution of the semidefinite program, then there must exist $(\eta^*, U^*) \in \Re^p \times \mathcal{S}^n_+$ such that

$$\nabla_x L(x^*, U^*, \eta^*) = 0, \quad U^* G(x^*) = 0, \quad U^* \succeq 0, \tag{21}$$

where $L : \Re^m \times \mathcal{S}^n \times \Re^p \to \Re$ is the Lagrangian function defined by

$$L(x, U, \eta) \equiv \theta(x) + U \bullet G(x) - \eta^T h(x) \quad \text{for } (x, U, \eta) \in \Re^m \times \mathcal{S}^n \times \Re^p. \tag{22}$$

Clearly, the first-order optimality condition (21) and the feasibility of x^* is equivalent to the implicitly mixed complementarity problem (6) in which the map $F : \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^{p+m} \to \mathcal{S}^n \times \Re^{p+m}$ is defined by

$$F(U, V, \eta, x) \equiv \begin{pmatrix} V + G(x) \\ h(x) \\ \nabla_x L(x, U, \eta) \end{pmatrix} \quad \forall (U, V, \eta, x) \in \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^{p+m}, \tag{23}$$

and the following correspondence of variables are made: $(U, V) \leftrightarrow (X, Y)$ and $(\eta, x) \leftrightarrow z$. Hence, as in section 3.1, the feasibility of x^* and the first-order optimality condition (21) can be formulated as the CE (Ω, H), where the set Ω and the map $H : \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^{p+m} \to \mathcal{S}^n \times \Re^{p+m}$ are defined by

$$\Omega \equiv \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^{p+m}, \tag{24}$$

$$H(U, V, \eta, x) \equiv \begin{pmatrix} (UV + VU)/2 \\ F(U, V, \eta, x) \end{pmatrix} \quad \text{for } (U, V, \eta, x) \in \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^{p+m}. \tag{25}$$

Our goal is to solve the CE (Ω, H) by the potential reduction Newton method. For this purpose, we make several blanket assumptions on problem (20). These are all fairly standard assumptions; in particular, (D4) is a second-order sufficiency condition. The assumptions are as follows.

(D1) The objective function $\theta : \Re^m \to \Re$ is twice continuously differentiable and convex.

(D2) The map $G : \Re^m \to \mathcal{S}^n$ is twice continuously differentiable and *positive semidefinite convex* (psd-convex); that is,

$$G(tx + (1-t)y) \preceq tG(x) + (1-t)G(y) \quad \forall x, y \in \Re^m \ \forall t \in (0, 1).$$

(D3) The map $h : \Re^m \to \Re^p$ is affine, and the (constant) gradients $\{\nabla h_j(x)\}_{j=1}^p$ are linearly independent.

(D4) For every $(x, U, \eta) \in \Re^m \times \mathcal{S}^n_{++} \times \Re^p$, the following implication holds:

$$\left.\begin{array}{c} h'(x;v) = 0 \\ G'(x;v) = 0 \\ v \neq 0 \end{array}\right\} \implies v^T L''_{xx}(x, U, \eta) v \; > \; 0.$$

(D5) The feasible set

$$X \equiv \{ x \in \Re^m : G(x) \preceq 0; \; h(x) = 0 \}$$

is nonempty and bounded.

We propose below a new interior point method for solving the convex semidefinite program (20) based on the potential reduction Newton algorithm of section 2.3. This method not only generalizes the algorithm developed in section 4.2 of [36] to the context of the nonlinear semidefinite programming problem, but it also allows for a more general choice of starting points. The new algorithm uses a novel potential function ψ which depends on the starting point. A key advantage of the new algorithm is that good convergence properties can be established for arbitrary starting points. This differs from the results in [36], which either require the starting point to satisfy the linear equality constraint $h(x) = 0$ (Theorem 5 in the reference) or do not guarantee the boundedness of the sequence of multipliers (Theorem 4 in the reference).

Let $(U^0, V^0, \eta^0, x^0) \in \mathcal{U} \times \Re^{p+m}$ denote an arbitrary starting point and let $c^0 \equiv h(x^0)$ and $G^0 \in \mathcal{S}^n$ be any matrix such that

$$G(x^0) \prec G^0 \prec G(x^0) + V^0 \quad \text{if } c^0 \neq 0;$$
$$G^0 \succ 0 \quad \text{if } c^0 = 0.$$

Define

$$(26)\quad S \equiv \begin{cases} \left\{ (A, B, c, d) \in \mathcal{S}^n_+ \times \mathcal{S}^n \times \Re^{p+m} : B \succeq \dfrac{c^T c^0}{\|c^0\|^2} G^0 \right\} & \text{if } c^0 \neq 0, \\ \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^{p+m} \quad \text{if } c^0 = 0. \end{cases}$$

Note that S depends on the starting point when $h(x^0) \neq 0$.

The following technical lemma is a partial restatement of Lemma 6 of [18] and is used in the subsequent Lemma 9 to establish that the CE (Ω, H) and the set S defined above satisfy conditions (A1)–(A3) of section 2.1.

LEMMA 8. *Assume that $G : \Re^m \to \mathcal{S}^n$ is psd-convex and $h : \Re^m \to \Re^p$ is an affine function. Then the following statements hold:*

(a) *for every $U \in \mathcal{S}^n_+$, the function $x \in \Re^m \mapsto U \bullet G(x)$ is convex;*

(b) *if condition* (D5) *holds then, for every $\bar{B} \in \mathcal{S}^n$ and $\bar{\gamma} \in \Re$, the set*

$$\{ x \in \Re^m : G(x) \preceq \bar{B}, \; \| h(x) \| \leq \bar{\gamma} \}$$

is bounded.

LEMMA 9. *Assume that problem* (20) *satisfies conditions* (D1)–(D4). *The following three statements hold:*

(a) *the map F defined by* (23) *satisfies* (B1) *and* (B2) *of section* 3.1;

(b) *the pair* (Ω, H) *with* Ω *and* H *defined by* (25) *and* (24), *respectively, and the set* S *defined by* (26) *satisfy conditions* (A1), (A2), *and* (A3) *of section* 2.1; *and*

(c) *the map* H *restricted to the set* $\mathcal{U} \times \Re^{p+m}$ *is a local homeomorphism.*

Proof. Since the case where $c^0 = 0$ is easy to deal with, the proof below focuses on the case where $c^0 \neq 0$. Conditions (A1) and (A2)(a) are obvious. Clearly, we have

$$\Omega_{\mathcal{I}} = \left\{ (U, V, \eta, x) \in \mathcal{U} \times \Re^{p+m} \,:\, V + G(x) \succ \frac{h(x^0)^T h(x)}{\|h(x^0)\|^2} G^0 \right\}, \tag{27}$$

which is nonempty because it contains the tuple (U^0, V^0, η^0, x^0). Moreover, using (10) we easily see that the set $H^{-1}(\text{int } S) \cap \text{bd } \Omega$ is empty. We have thus proved that condition (A2) holds. Using the same arguments as in the proof of Lemma 3, we can show that if statement (a) holds, then $H'(U, V, \eta, x)$ is nonsingular for every $(U, V, \eta, x) \in \mathcal{U} \times \Re^{p+m}$; in particular, we can conclude that (A3) holds due to (27), and that H restricted to the set $\mathcal{U} \times \Re^{p+m}$ is a local homeomorphism by the inverse function theorem. Thus the remaining proof is to show that F satisfies (B1) and (B2). For this purpose, assume that $(U, V, x, \eta) \in \mathcal{U} \times \Re^{p+m}$ satisfies

$$F'((U, V, x, \eta); (dU, dV, dx, d\eta)) = 0$$

for some $(dU, dV, dx, d\eta) \in \mathcal{S}^n \times \mathcal{S}^n \times \Re^{p+m}$ or, equivalently,

$$dV + G'(x; dx) = 0, \tag{28}$$

$$L''_{xx}(x, U, \eta)dx + \sum_{i,j=1}^{n} dU_{ij} \nabla G_{ij}(x) - \sum_{k=1}^{\ell} d\eta_k \nabla h_k(x) = 0, \tag{29}$$

$$h'(x; dx) = 0. \tag{30}$$

Lemma 8(a) together with conditions (D1), (D2), and (D3) and the fact that $U \succeq 0$ imply that $L(x, U, \eta)$ is a convex function of x. Hence, we have $dx^T L''_{xx}(x, U, \eta)dx \geq 0$. Multiplying (29) on the left by dx^T and using this last observation together with (28) and (30), we obtain

$$dU \bullet dV = -dU \bullet G'(x; dx) + d\eta^T h'(x; dx) = dx^T L''_{xx}(x, U, \eta)dx \geq 0. \tag{31}$$

Thus F satisfies (B1). Assume now that

$$F'((U, V, x, \eta); (0, 0, dx, d\eta)) = 0.$$

Then all the relations above hold with $(dU, dV) = (0, 0)$. In particular, (28), (30), and (31) imply that $h'(x; dx) = 0$, $G'(x; dx) = 0$, and $dx^T L''_{xx}(x, U, \eta)dx = 0$. Hence, we conclude that $dx = 0$ due to (D4). Using this and the fact that relation (29) holds with $dU = 0$, we obtain

$$\sum_{k=1}^{\ell} d\eta_k \nabla h_k(x) = 0,$$

which in turn implies that $d\eta = 0$ due to (D3). We have thus shown that F satisfies (B2). $\square$

Associated with the set S, we now introduce the following potential function $p : \text{int } S \to \Re$ defined for any tuple $(A, B, c, d) \in \text{int } S$ by

$$p(A,B,c,d) \equiv \zeta \log\left(\| A \|_F^2 + \left\| B - \frac{c^T c^0}{\| c^0 \|} G^0 \right\|_F^2 + \| c \|^2 + \| d \|^2 \right)$$

$$- \log(\det A) - \log\left(\det\left(B - \frac{c^T c^0}{\| c^0 \|^2} G^0 \right) \right), \tag{32}$$

where ζ is a suitable constant.

We establish in the next result that if $\zeta \geq 3n/2$, then the above potential function satisfies conditions (A4), (A5), and (A6) of section 2.1.

Lemma 10. *If $\zeta \geq 3n/2$, then the potential function* (32), *the tuple $a \equiv (I, 0, 0, 0) \in \mathcal{S}^n \times \mathcal{S}^n \times \Re^{p+m}$, and the constant $\bar{\sigma} \equiv 1/2$ satisfy conditions* (A4), (A5), *and* (A6) *of section* 2.1.

Proof. The verification of (A4) is similar to that of Lemma 4. Define

$$\tau \equiv \| A \|_F^2 + \left\| B - \frac{c^T c^0}{\| c^0 \|} G^0 \right\|_F^2 + \| c \|^2 + \| d \|^2,$$

$$\tilde{B} \equiv B - \frac{c^T c^0}{\| c^0 \|} G^0.$$

It is easy to see that

$$\nabla p(A,B,c,d) = \begin{pmatrix} \dfrac{2\,\zeta}{\tau} A - A^{-1} \\ \dfrac{2\,\zeta}{\tau} \tilde{B} - \tilde{B}^{-1} \\ \dfrac{2\,\zeta}{\tau} \left(c - \dfrac{\tilde{B} \bullet G^0}{\|c^0\|^2} c^0 \right) + \dfrac{\tilde{B}^{-1} \bullet G^0}{\|c^0\|^2} c^0 \\ \dfrac{2\,\zeta}{\tau} d \end{pmatrix}.$$

The definition of τ and $\tilde{B}$ together with a simple algebraic manipulation reveals that

$$\nabla p(A,B,c,d) \bullet (A,B,c,d) = 2(\zeta - n) > 0 \quad \text{for all } (A,B,c,d) \in \text{int } S,$$

and hence that (A5) holds. Moreover, using the fact that

$$(\text{tr}P)^2 \leq n \| P \|_F^2 \quad \text{and} \quad (\text{tr}P^{-1})\,(\text{tr}P) \geq n^2$$

for every $P \in \mathcal{S}^n$ and $\zeta \geq 3n/2$, we obtain for every $(A,B,c,d) \in \text{int } S$,

$$\frac{[\,\nabla p(A,B,c,d) \bullet (I,0,0,0)\,]\,[(A,B,c,d) \bullet (I,0,0,0)]}{\| (I,0,0,0) \|_F^2}$$

$$= \frac{1}{n} \left[\frac{2\zeta}{\tau} (\text{tr}A)^2 - (\text{tr}A^{-1})\,(\text{tr}A) \right] \leq \frac{2\zeta\,(\text{tr}A)^2}{n \| A \|_F^2} - \frac{(\text{tr}A^{-1})\,(\text{tr}A)}{n}$$

$$\leq 2\,\zeta - n < 4\,\zeta - 4n = 2\,[\,p(A,B,c,d) \bullet (A,B,c,d)\,].$$

Hence (A6) holds with $a = (I, 0, 0, 0)$ and $\bar{\sigma} = 1/2$. □

The next two results will be used in Theorem 3 to establish the boundedness of the sequence of iterates generated by the potential reduction Newton method under the framework of this section.

LEMMA 11. *Assume that problem* (20) *satisfies conditions* (D1)–(D5). *Then the map* $H : \mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^{p+m} \to \mathcal{S}^n \times \Re^{p+m}$ *defined in* (25) *is proper with respect to the set* $\mathcal{S}^n \times F(\mathcal{U} \times \Re^{p+m})$.

Proof. Using Proposition 4(a) and Lemma 7 of [18], we conclude that the map F defined in (23) is (U, V)-equilevel monotone on $\mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^{p+m}$. Moreover, by Proposition 4(c) and Lemma 9 of [18], it follows that F is (η, x)-bounded on $\mathcal{S}^n_+ \times \mathcal{S}^n_+ \times \Re^{p+m}$. Since, by Lemma 9, the map H restricted to $\mathcal{U} \times \Re^{m+p}$ is a local homeomorphism, we conclude from Proposition 2 that H is proper with respect to $\mathcal{S}^n \times F(\mathcal{U} \times \Re^{p+m})$. □

In the next result we describe in more detail the set $F(\mathcal{U} \times \Re^{p+m})$ for the map F given by (23).

LEMMA 12. *Assume that problem* (20) *satisfies conditions* (D1)–(D5). *Then* $F(\mathcal{U} \times \Re^{p+m}) = \mathcal{F} \times \Re^m$, *where* F *is the map given by* (23) *and*

$$\mathcal{F} \equiv \{ (B, c) \in \mathcal{S}^n \times \Re^p : \exists x \in \Re^m \text{ such that } G(x) \prec B \text{ and } h(x) = c \}.$$

Moreover, $\mathcal{F}$ *is a convex set.*

Proof. The inclusion $F(\mathcal{U} \times \Re^{p+m}) \subset \mathcal{F} \times \Re^m$ follows straightforwardly from the definition of the map F and the set $\mathcal{U}$. Assume now that $(B, c, d) \in \mathcal{F} \times \Re^m$. We have proved in Lemma 10 of [18] that if conditions (D1)–(D5) hold and $(0, 0) \in \mathcal{F}$, then $(0, 0, 0) \in F(\mathcal{U} \times \Re^{p+m})$. Consider now the problem

$$\begin{aligned} &\text{minimize} && \tilde{\theta}(x) \\ &\text{subject to} && \tilde{G}(x) \preceq 0, \quad \tilde{h}(x) = 0, \end{aligned}$$

where $\tilde{\theta}(x) \equiv \theta(x) - d^T x$, $\tilde{G}(x) \equiv G(x) - B$, and $\tilde{h}(x) \equiv h(x) - c$ for all $x \in \Re^m$. It is easy to see that the functions $\tilde{\theta}$, $\tilde{G}$, and $\tilde{h}$ also satisfy conditions (D1)–(D5). Hence, applying Lemma 10 of [18] to this new problem, we conclude that $(0, 0, 0) \in \tilde{F}(\mathcal{U} \times \Re^{p+m})$, where $\tilde{F}$ is defined like the function F in (23) with θ, G, and h replaced by $\tilde{\theta}$, $\tilde{G}$, and $\tilde{h}$, respectively. A simple verification shows that $(0, 0, 0) \in \tilde{F}(\mathcal{U} \times \Re^{p+m})$ is equivalent to $(B, c, d) \in F(\mathcal{U} \times \Re^{p+m})$. We have thus shown that $F(\mathcal{U} \times \Re^{p+m}) \supseteq \mathcal{F} \times \Re^m$. Using conditions (D2) and (D3), and some standard arguments, we can easily show that $\mathcal{F}$ is a convex set. □

We establish one technical lemma, which will be used to prove an important conclusion of the main result of this section, Theorem 6.

LEMMA 13. *Let* $\{U^k\}$ *and* $\{V^k\}$ *be two sequences in* $\mathcal{S}^n_{++}$ *such that*

$$\lim_{k \to \infty} (U^k V^k + V^k U^k) = 0.$$

Then

$$\lim_{k \to \infty} (U^k)^{1/2} V^k (U^k)^{1/2} = 0. \tag{33}$$

Proof. Since $(U^k)^{1/2} V^k (U^k)^{1/2}$ is a symmetric matrix, its eigenvalues are all real. Since

$$(U^k)^{-1/2} (U^k V^k) (U^k)^{1/2} = (U^k)^{1/2} V^k (U^k)^{1/2},$$

it follows that all the eigenvalues of U^kV^k are real too. This implies that the eigenvalues of $(U^kV^k)^2$ are all positive. Therefore,

$$2\,\|\,U^kV^k\,\|_F^2 \;\le\; 2\,\|\,U^kV^k\,\|_F^2 + 2\,\mathrm{tr}\left(U^kV^k\right)^2 \;=\; \|\,U^kV^k + V^kU^k\,\|_F^2.$$

Since the right-hand norm converges to zero as $k \to \infty$, the same holds for the left-hand norm. Thus the spectrum of U^kV^k converges to the single element $\{0\}$. Since this spectrum is the same as that of $(U^k)^{1/2}V^k(U^k)^{1/2}$, the desired limit (33) follows. □

The following is the main convergence result of the potential reduction Newton method specialized to the convex semidefinite program (20). A noteworthy remark about this result is that part (d) does not require the sequence of multipliers $\{(U^k,\eta^k)\}$ to be bounded.

THEOREM 6. *Suppose that problem* (20) *satisfies conditions* (D1)–(D5), *and that* $\{(U^k,V^k,\eta^k,x^k)\}$ *is a sequence generated by the potential reduction Newton method of section* 2.3 *initialized at an arbitrary tuple* $(U^0,V^0,\eta^0,x^0)\} \in \mathcal{U}\times\Re^{p+m}$, *and with* (Ω,H), S, $p : \mathrm{int}\,S \to \Re$ *given by* (24), (25), (26), *and* (32), *respectively,* $a \equiv (I,0,0,0) \in \mathcal{S}^n\times\mathcal{S}^n\times\Re^{p+m}$, *and* $\bar\sigma \equiv 1/2$. *Assume also that* $\zeta \ge 3/2$ *and* $\limsup_k \sigma_k < 1/2$. *Then, the following statements hold:*

(a) *every accumulation point of* $\{(U^k,V^k,\eta^k,x^k)\}$ *is a solution of the CE* (Ω,H);

(b) *the sequence* $\{(V^k,x^k)\}$ *is bounded; thus* $\{x^k\}$ *has at least one accumulation point;*

(c) $\lim_{k\to\infty} H(U^k,V^k,\eta^k,x^k) = 0$;

(d) *every accumulation point of the sequence* $\{x^k\}$ *is an optimal solution of problem* (20);

(e) *if there exists* $\bar x \in \Re^m$ *such that* $h(\bar x) = 0$ *and* $G(\bar x) \prec 0$ *(that is, problem* (20) *has a Slater point), then the whole sequence* $\{(U^k,V^k,\eta^k,x^k)\}$ *is bounded.*

Proof. By Lemmas 9 and 10, the assumptions of the theorem guarantee that (Ω,H), S, $p : \mathrm{int}\,S \to \Re$, $a = (I,0,0,0)$, and $\bar\sigma = 1/2$ satisfy conditions (A1)–(A6) of section 2.1. Hence, by Theorem 3, we conclude that statement (a) holds and that the sequence $\{H(U^k,V^k,\eta^k,x^k)\}$ is bounded. By the definition of H, this implies that $\{V^k + G(x^k)\}$ and $\{h(x^k)\}$ are bounded, and hence $\{x^k\} \subset \{x \in \Re^m : G(x) \preceq \bar B,\ \|h(x)\| \le \bar\gamma\}$ for some $(\bar B,\bar\gamma) \in \mathcal{S}^n\times\Re$. Since by Lemma 8(b) the latter set is bounded, we conclude that $\{x^k\}$ is bounded. Clearly, this and the fact that $\{V^k + G(x^k)\}$ is bounded imply that $\{V^k\}$ is also bounded. Hence, statement (b) follows.

The proofs of statements (c) and (e) are based on statements (c) and (d) of Theorem 3. For simplicity, we assume in the remaining proof that $c^0 \equiv h(x^0) \ne 0$; the proof when $c^0 = 0$ is analogous. Define

$$E \equiv \mathcal{S}^n_+ \times \left\{ (B,c) \in \mathcal{S}^n\times[0,c^0] \,:\, B \succeq \frac{c^Tc^0}{\|c^0\|^2}\,G^0 \right\} \times \Re^m.$$

Note that E is a closed subset of S. Moreover, using (D3) and the fact that the third component of a is zero, we easily see that $\{h(x^k)\} \subset [0,c^0]$. Clearly, this implies that $\{H(U^k,V^k,\eta^k,x^k)\} \subset E$. In view of (c) and (d) of Theorem 3, statements (c) and (e) follow once we establish that the map H is proper with respect to

$$E \cap \mathrm{int}\,S = \mathcal{S}^n_{++} \times \left\{ (B,c) \in \mathcal{S}^n\times[0,c^0] \,:\, B \succ \frac{c^Tc^0}{\|\,c^0\,\|^2}\,G^0 \right\} \times \Re^m$$

and also proper with respect to E under the assumption that $(0,0) \in \mathcal{F}$. We prove first the properness assertion with respect to int $S \cap E$. By Lemmas 11 and 12, we know that H is proper with respect to $\mathcal{S}^n \times F(\mathcal{U} \times \Re^{p+m}) = \mathcal{S}^n \times \mathcal{F} \times \Re^m$. Hence, it suffices to show that int $S \cap E$ is contained in $\mathcal{S}^n \times \mathcal{F} \times \Re^m$, or equivalently that

$$\left\{ (B,c) \in \mathcal{S}^n \times [0,c^0] : B \succ \frac{c^T c^0}{\|c^0\|^2} G^0 \right\} \subset \mathcal{F}. \tag{34}$$

Using the definition of $\mathcal{F}$ and Lemma 8(b), it is easy to see that

$$\text{cl } \mathcal{F} = \{ (B,c) \in \mathcal{S}^n \times \Re^p : \exists\, x \in \Re^m \text{ such that } G(x) \preceq B \text{ and } h(x) = c \}. \tag{35}$$

Moreover, it follows immediately from the definition of $\mathcal{F}$ and (35) that

$$(B,c) \in \mathcal{F} \Rightarrow (B',c) \in \mathcal{F} \quad \forall B' \succeq B, \tag{36}$$

$$(B,c) \in \text{cl } \mathcal{F} \Rightarrow (B',c) \in \mathcal{F} \quad \forall B' \succ B. \tag{37}$$

Let (B,c) be an arbitrary element of the left-hand set in (34). Since $c \in [0,c^0]$, we have $c = tc^0$ for some $t \in [0,1]$. Hence,

$$B \succ \frac{c^T c^0}{\|c^0\|^2} G^0 = tG^0. \tag{38}$$

Since $(0,0) \in \text{cl } \mathcal{F}$ by (D5), $(G^0,c^0) \in \mathcal{F}$ by (26), and cl $\mathcal{F}$ is a convex set due to Lemma 12 and Proposition III.1.2.7 of [5], we conclude that $(tG^0,tc^0) = t(G^0,c^0) + (1-t)(0,0) \in \text{cl } \mathcal{F}$. Hence, by (37) and (38), we have $(B,c) = (B,tc^0) \in \mathcal{F}$. Hence, (34) holds.

Assume now that $(0,0) \in \mathcal{F}$. To prove the properness assertion with respect to E, it suffices to show that $E \subset \mathcal{S}^n \times \mathcal{F} \times \Re^m$ or, equivalently, that

$$\left\{ (B,c) \in \mathcal{S}^n \times [0,c^0] : B \succeq \frac{c^T c^0}{\| c^0 \|^2} G^0 \right\} \subset \mathcal{F}. \tag{39}$$

If (B,c) is in the left-hand set, then we have $c = tc^0$ and $B \succeq tG^0$ for some $t \in [0,1]$. Since $(0,0) \in \mathcal{F}$ by assumption, $(G^0,c^0) \in \mathcal{F}$ by (26), and $\mathcal{F}$ is convex by Lemma 12, we conclude that $(tG^0,tc^0) \in \mathcal{F}$. Hence, by (36) and the fact that $B \succeq tG^0$, we have $(B,c) = (B,tc^0) \in \mathcal{F}$. Hence, (39) holds.

Finally, we prove statement (d). For each k, let $B^k \equiv G(x^k) + V^k$, $\tilde{B}^k \equiv G(x^k) + (U^k)^{-1}$, and $d^k \equiv \nabla_x L(x^k,U^k,\eta^k)$. It follows that x^k is an optimal solution of the convex program

$$\min \left\{ f(x) - (d^k)^T x - \log\det\left(\tilde{B}^k - G(x)\right) \;:\; h(x) = h(x^k) \right\}, \tag{40}$$

due to the fact that x^k together with the multiplier pair (U^k,η^k) satisfy the optimality condition for this problem. Now let x^∞ be an arbitrary accumulation point of $\{x^k\}$. Clearly, x^∞ is a feasible solution of (20) due to Theorem 6(c). To show the global optimality of x^∞, assume that $\tilde{x}$ is an arbitrary feasible solution of (20). Let $t_k \in [0,1]$ be such that $h(x^k) = t_k h(x^0)$ and define $\tilde{x}^k \equiv t_k x^0 + (1-t_k)\tilde{x}$. Clearly, $\tilde{x}^k$ is feasible to (40). Since $\{t_k\}$ converges to zero, it follows that $\{\tilde{x}^k\}$ converges to $\tilde{x}$. Moreover, since $H(U^k,V^k,\eta^k,x^k) \in S$, by the definition of S (26), we have for each k (cf. (38)),

$$B^k \succ t_k G^0.$$

Hence, it follows that

$$\begin{aligned} &f(\tilde{x}^k) - (d^k)^T\tilde{x}^k - \log\det\left(\tilde{B}^k - G(\tilde{x}^k)\right) \\ &\geq f(x^k) - (d^k)^T x^k - \log\det\left(\tilde{B}^k - G(x^k)\right) \\ &= f(x^k) - (d^k)^T x^k + \log\det\left(U^k\right) \end{aligned}$$

for all k. Rearranging this inequality, we obtain

$$\begin{aligned} &f(\tilde{x}^k) - f(x^k) - (d^k)^T(\tilde{x}^k - x^k) \\ &\qquad \geq \log\det\left((U^k)^{1/2}\left[\tilde{B}^k - G(\tilde{x}^k)\right](U^k)^{1/2}\right) \\ &\qquad = \log\det\left(I + (U^k)^{1/2}\left[G(x^k) - G(\tilde{x}^k)\right](U^k)^{1/2}\right) \\ &\qquad = \log\det\left(I - (U^k)^{1/2}V^k(U^k)^{1/2} + (U^k)^{1/2}\left[B^k - G(\tilde{x}^k)\right](U^k)^{1/2}\right) \\ &\qquad \geq \log\det\left(I - (U^k)^{1/2}V^k(U^k)^{1/2}\right), \end{aligned}$$

where the last inequality follows from the fact that

$$B^k - G(\tilde{x}^k) \succeq B^k - t_k G(x^0) - (1-t_k)G(\tilde{x}) \succeq B^k - t_k G(x^0) \succ B^k - t_k G^0 \succ 0.$$

Hence, as k goes to ∞, we may invoke Lemma 13 to conclude that $f(\tilde{x}) - f(x^\infty) \geq 0$. We have thus proved that x^∞ is an optimal solution of (20). □

Assuming that $G^0 \succ 0$, it is possible to show that the potential function (32), $a \equiv (I, I, 0, 0)$, and $\bar{\sigma} = 1$ satisfy the inequality in condition (A6) for every (A, B, c, d) in the set $E \cap \operatorname{int} S$, where E is defined as in the proof of Theorem 6. Using this fact, it is possible to establish a convergence result similar to Theorem 6 for $a \equiv (I, I, 0, 0)$ and $\bar{\sigma} = 1$. The interesting point to note is that Theorem 3 still holds if we assume the inequality in condition (A6) to be valid only for points in the sequence $\{H(x^k)\}$. Details are omitted.

Acknowledgment. The authors would like to thank the two anonymous referees for their constructive comments.

REFERENCES

[1] F. Alizadeh, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.

[2] F. Alizadeh, J.-P. Haeberly, and M. Overton, *Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results*, SIAM J. Optim., 8 (1998), pp. 746–768.

[3] R. M. Freund, *Complexity of an Algorithm for Finding an Approximate Solution of a Semidefinite Program with No Regularity Condition*, Working Paper OR 302-94, Operations Research Center, Massachusetts Institute of Technology, Cambridge, December 1994.

[4] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. Optim., 6 (1996), pp. 342–361.

[5] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms* I, Comprehensive Study in Mathematics 305, Springer-Verlag, New York, 1993.

[6] F. Jarre, *An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices*, SIAM J. Control and Optim., 31 (1993), pp. 1360–1377.

[7] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*, Lecture Notes in Comput. Sci. 538, Springer-Verlag, Berlin, 1991.

[8] M. Kojima, M. Shida, and S. Shindoh, *A predictor-corrector interior-point algorithm for the semidefinite linear complementarity problem using the Alizadeh-Haeberly-Overton search direction*, SIAM J. Optim, 9 (1999), pp. 444–465.

[9] M. Kojima, M. Shida, and S. Shindoh, *Local convergence of predictor-corrector infeasible-interior-point algorithms for SDPs and SDLCPs*, Math. Programming, 80 (1998), pp. 129–160.

[10] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices*, SIAM J. Optim., 7 (1997), pp. 86–125.

[11] C.-J. Lin and R. Saigal, *A Predictor-Corrector Method for Semi-Definite Programming*, Working Paper, Dept. of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1995.

[12] Z.-Q. Luo, J. F. Sturm, and S. Zhang, *Superlinear convergence of a symmetric primal-dual path following algorithm for semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 59–81.

[13] L. McLinden, *The complementarity problem for maximal monotone multifunctions*, in Variational Inequalities and Complementarity Problems, R. Cottle, F. Giannessi, and J.-L. Lions, eds., John Wiley, New York, 1980, pp. 251–270.

[14] N. Megiddo, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming: Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 131–158.

[15] R. D. C. Monteiro, *Primal-dual path-following algorithms for semidefinite programming*, SIAM J. Optim., 7 (1997), pp. 663–678.

[16] R. D. C. Monteiro, *Polynomial convergence of primal-dual algorithms for semidefinite programming based on the Monteiro and Zhang family of directions*, SIAM J. Optim., 8 (1998), pp. 797–812.

[17] R. D. C. Monteiro and J.-S. Pang, *Properties of an interior-point mapping for mixed complementarity problems*, Math. Oper. Res., 21 (1996), pp. 629–654.

[18] R. D. C. Monteiro and J.-S. Pang, *On two interior-point mappings for nonlinear semidefinite complementarity problems*, Math. Oper. Res., 23 (1998), pp. 39–60.

[19] R. D. C. Monteiro and T. Tsuchiya, *Polynomial convergence of a new family of primal-dual algorithms for semidefinite programming*, SIAM J. Optim., 9 (1999), pp. 551–577.

[20] R. D. C. Monteiro and T. Tsuchiya, *Polynomiality of Primal-Dual Algorithms for Semidefinite Linear Complementarity Problems Based on the Kojima-Shindoh-Hara Family of Directions*, Manuscript, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, August 1996; Math. Programming, to appear.

[21] R. D. C. Monteiro and P. Zanjácomo, *A note on the existence of the Alizadeh-Haeberly-Overton direction for semidefinite programming*, Math. Programming, 78 (1997), pp. 393–396.

[22] R. D. C. Monteiro and Y. Zhang, *A unified analysis for a class of path-following primal-dual interior-point algorithms for semidefinite programming*, Math. Programming, 81 (1998), pp. 281–299.

[23] Yu. E. Nesterov and A. S. Nemirovskii, *Polynomial barrier methods in convex programming*, Ekonomika i Mat. Metody, 24 (1988), pp. 1084–1091 (in Russian).

[24] Yu. E. Nesterov and A. S. Nemirovskii, *Self-Concordant Functions and Polynomial Time Methods in Convex Programming*, Preprint, Central Economic & Mathematical Institute, USSR Acad. Sci. Moscow, 1989.

[25] Yu. E. Nesterov and A. S. Nemirovskii, *Interior Point Methods in Convex Programming: Theory and Applications*, SIAM, Philadelphia, PA, 1994.

[26] Yu. E. Nesterov and M. Todd, *Self-scaled barriers and interior-point methods for convex programming*, Math. Oper. Res., 22 (1997), pp. 1–42.

[27] Yu. E. Nesterov and M. Todd, *Primal-dual interior-point methods for self-scaled cones*, SIAM J. Optim., 8 (1998), pp. 324–364.

[28] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, San Diego, 1970.

[29] F. A. Potra and R. Sheng, *A superlinearly convergent primal-dual infeasible-interior-point algorithm for semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 1007–1028.

[30] A. Shapiro, *First and second order analysis of nonlinear semidefinite programs*, Math. Programming, 77 (1997), pp. 301–320.

[31] M. Shida, S. Shindoh, and M. Kojima, *Existence and uniqueness of search directions in interior-point algorithms for the SDP and the monotone SDLCP*, SIAM J. Optim., 8 (1998), pp. 387–396.

[32] J. F. Sturm and S. Zhang, *Symmetric Primal-Dual Path-Following Algorithms for Semidefinite Programming*, Report 9554/A, Econometric Institute, Erasmus University, Rotterdam, The Netherlands, November 1995.

[33] M. J. Todd, K. C. Toh, and R. H. Tütüncü, *On the Nesterov–Todd direction in semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 769–796.

[34] P. Tseng, *Search directions and convergence analysis of some infeasible path-following methods for the monotone semi-definite LCP*, Optim. Methods. Softw., 9 (1998), pp. 245–268.

[35] L. Vandenberghe and S. Boyd, *A primal-dual potential reduction method for problems involving matrix inequalities*, Math. Programming, 69 (1995), pp. 205–236.

[36] T. Wang, R. D. C. Monteiro, and J.-S. Pang, *An interior point potential reduction method for constrained equations*, Math. Programming, 74 (1996), pp. 159–195.

[37] Y. Zhang, *On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 365–386.

SIAM J. OPTIM.
Vol. 9, No. 3, pp. 755–778

A PRACTICAL ALGORITHM FOR GENERAL LARGE SCALE NONLINEAR OPTIMIZATION PROBLEMS[*]

PAUL T. BOGGS[†], ANTHONY J. KEARSLEY[‡], AND JON W. TOLLE[§]

Abstract. We provide an effective and efficient implementation of a sequential quadratic programming (SQP) algorithm for the general large scale nonlinear programming problem. In this algorithm the quadratic programming subproblems are solved by an interior point method that can be prematurely halted by a trust region constraint. Numerous computational enhancements to improve the numerical performance are presented. These include a dynamic procedure for adjusting the merit function parameter and procedures for adjusting the trust region radius. Numerical results and comparisons are presented.

Key words. nonlinear programming, interior point, SQP, merit function, trust region, large scale

AMS subject classifications. 49M37, 65K05, 90C30

PII. S105262349426722X

1. Introduction.

In a series of recent papers, [3], [6], and [8], the authors have developed a new algorithmic approach for solving large, nonlinear, constrained optimization problems. This proposed procedure is, in essence, a sequential quadratic programming (SQP) method that uses an interior point algorithm for solving the quadratic subproblems and achieves global convergence through the application of a special merit function and a trust region strategy. Over the past several years the theory supporting this approach has been analyzed and strengthened. This theory is presented in a companion paper [4]. In addition, implementations of the algorithm have been extensively tested on a variety of large problems, including standard test problems and problems of engineering and scientific origin, ranging in size from several hundred to several thousand variables with up to several thousand constraints. Specific strategies have been developed for handling the parameters utilized by the algorithm and for dealing with nontrivial pathologies (e.g., linearly dependent active constraint gradients or inconsistent linearized constraints in the quadratic subproblem) that often occur in large scale problems. In this paper we present the results of these efforts.

Based on its theoretical foundation and on our numerical experience we are confident that this algorithm provides an efficient means for attacking a large, sparse, nonlinear program with equality or inequality constraints. Rigorous comparison of algorithms for large nonlinear problems is notoriously difficult, especially given the extensive set of options typically available in codes for such problems. Nevertheless,

[*] Received by the editors May 6, 1994; accepted for publication (in revised form) November 14, 1997; published electronically May 21, 1999. This work was performed by an employee of the U.S. Government or under U.S. Government contract. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/siopt/9-3/26722.html

[†] Computational Sciences and Mathematics Research Department, Sandia National Laboratories, Livermore, CA 94550 (ptboggs@ca.sandia.gov).

[‡] Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213-3890.

[§] Departments of Mathematics and Operations Research, University of North Carolina, Chapel Hill, NC 27599.

our algorithm, with the (conservative) default parameter settings, has been successful on problems that have caused difficulties for other algorithms, and consequently we are encouraged to believe that it is competitive at the current stage in the development of methods for solving these large problems.

Below we give an outline of our basic procedure, and in the succeeding sections we provide more specific detail on the component parts of the implemented algorithm, including the strategies and safeguards that we have used. We also exhibit and comment on the results of some of our numerical tests. This paper relies heavily on the results from the paper [4] on the theory for motivation of the basic ideas.

We assume the general nonlinear programming problem to be of the form

$$(NLP) \qquad \begin{array}{l} \min\limits_{x} f(x) \\ \text{subject to } g(x) \leq 0, \end{array}$$

where $f : \mathcal{R}^n \rightarrow \mathcal{R}^1$ and $g : \mathcal{R}^n \rightarrow \mathcal{R}^m$ are smooth functions. Nonlinear equality constraints are not included in our description here in order to avoid distracting technicalities. The modifications necessary for their insertion can be inferred from [6]. Nonlinear equality constraints are included in our code and in some of the problems we tested. The SQP method is the backbone of our algorithm. (See [7] for a review of these techniques.) At the kth step we have an iterate, x^k, denoting the current approximation to the solution of (NLP). In addition to the x-iterate we also maintain a nonnegative iterate, $z^k \in \mathcal{R}^m$, which measures the infeasibility at x^k. At this stage (NLP) is modeled by a quadratic program of the form

$$(QP) \qquad \begin{array}{l} \min\limits_{\delta} \nabla f(x^k)^{\mathsf{T}}\delta + \frac{1}{2}\delta^{\mathsf{T}}B^k\delta \\ \text{subject to } \nabla g(x^k)^{\mathsf{T}}\delta + g(x^k) \leq 0. \end{array}$$

Here B^k is taken to be an appropriate approximation to the Hessian of the Lagrangian for (NLP), i.e.,

$$B^k \approx H_{xx}\ell(x^k, \lambda^k),$$

where

$$\ell(x, \lambda) = f(x) + g(x)^{\mathsf{T}}\lambda$$

and H_{xx} represents the Hessian with respect to x of the function to which it is applied. (See section 4.5 for a discussion of the choice of B^k used in our numerical experiments.) In this form (QP) generates a step that provides a search direction for improving the current iterate.

There are two significant points to be made concerning this phase of our algorithm. First, we apply an interior point quadratic program solver to (QP); more specifically, we use the method found in [1], where solutions are calculated by solving a sequence of low-dimensional quadratic programs. Pertinent details of this solver and its properties relative to its use in our SQP method can be found in section 2. Second, we do not try to solve (QP) with complete accuracy at each iteration; rather, we often terminate the interior point method prematurely. In particular, we halt the quadratic program solver when the steplength exceeds a "trust region radius" that is modified at each iteration according to how well the improvement in our merit function is predicted. Thus our algorithm can be said to be a "truncated Newton method" in the sense of [18] (see also [15]). This particular merit function and a more useful "working version" are discussed in section 3, and our strategy for updating the trust region radius is given in section 4.2.

The output of the (QP) solver is a vector that determines the direction of the step in the x-variable, which in turn yields a step direction for the "slack" variable z as explained in section 3. The combined step direction of these two variables is a descent direction for the working version of the merit function and also for constraint infeasibility; thus we can choose the steplength in this direction to decrease the merit function or the infeasibility of the iterate. The choice of steplength determines the new iterate x^{k+1} and also the new value z^{k+1}. The strategy for choosing the steplength and other algorithmic details, including the modifications and safeguards necessary to make an implementation robust, is given in section 4.

The results of our numerical tests are contained in section 5. These results demonstrate the overall effectiveness of the procedure and highlight the beneficial effect of our trust region strategy and other procedures. Finally, in section 6 we briefly consider weaknesses in the current version of the algorithm and suggest possible avenues of research to improve its efficiency.

For a discussion of the theoretical and practical questions related to large scale nonlinear programming, see the recent surveys [12], [14], and [21].

2. An interior point QP solver. Interior point methods for linear programming have been demonstrated to be very successful, especially on large problems, and recent research has led to their extension to quadratic programs. A particular method, the method of optimizing over low-dimensional subspaces, has performed well on linear programs and has been extended to the quadratic programming case (see [1], as well as [2] and the references contained therein). This method, for which good numerical results for quadratic programs have been reported, has properties that make it particularly compatible with the SQP algorithm we are describing in this paper. A brief description of the essential features of this method and their importance for our purposes follow. The many details of the actual algorithm that are not reported here may be found in the above references.

The quadratic program that we solve, (QP), has the form

$$\begin{aligned}&\min_{s}\ c^{\mathsf{T}}s+\tfrac{1}{2}s^{\mathsf{T}}Qs\\&\text{subject to } A^{\mathsf{T}}s+b\le 0,\end{aligned}\tag{2.1}$$

where $c, s\in\mathcal{R}^n$, $Q\in\mathcal{R}^{n\times n}$, $A\in\mathcal{R}^{n\times m}$, and $b\in\mathcal{R}^m$. The assumptions on (2.1) that are necessary to apply the interior point algorithm are that the problem be bounded, that A have full column rank, and that there exist feasible points (i.e., that the constraints be consistent). Note that Q can be indefinite and that no assumption of a full-dimensional interior is required. If equality constraints are present, they are handled by writing them as two inequalities.

An important prerequisite for solving (2.1) by an interior point method is a feasible initial point. Our algorithm uses a "big M" method to construct the Phase I problem

$$\begin{aligned}&\min_{s,\theta}\ c^{\mathsf{T}}s+\tfrac{1}{2}s^{\mathsf{T}}Qs+M\theta\\&\text{subject to } A^{\mathsf{T}}s+b-e\theta\le 0,\end{aligned}\tag{2.2}$$

where e is a vector of ones and θ is the "artificial" variable. Clearly, for θ^* large enough, the point $(s,\theta)=(0,\theta^*)$ is feasible for (2.2), and if M is sufficiently large, the algorithm applied to (2.2) will reduce θ until the artificial variable is nonpositive, at which point the current value of s is feasible and the $M\theta$ and $e\theta$ terms are dropped. If no such value of the artificial variable can be found, then (2.2) is not consistent and the algorithm stops. As discussed below, we make use of the step obtained from (2.2) even if it is not feasible for (QP). Note that when equality constraints are present, the entire solution procedure takes place in Phase I and θ will always be present.

The defining characteristic of the algorithm is that it proceeds by solving a sequence of low-dimensional subspace approximations to (2.1). In our application we follow the reported results in which the dimension of the subspace is taken as 3. The following is an outline of the O3D (for "optimizing over three-dimensional" subspaces) version of the algorithm. As the variable θ is treated essentially the same as the components of s in the O3D algorithm (see, however, step 6 below), the dependence on θ is incorporated into the formulation given in (2.1).

O3D Algorithm for Quadratic Programming

```
1. Given a feasible point, $s^0$; set $j := 0$.
2. Generate three independent search directions $p_i,\ i = 1, 2, 3$, and let $P^j$ be
   the matrix whose columns are $p_i$.
3. Form and solve the restricted quadratic program
       $\min_{\zeta} c^{\mathsf{T}}\tilde{s} + \frac{1}{2}\tilde{s}^{\mathsf{T}} Q\tilde{s}$
       subject to $A^{\mathsf{T}}\tilde{s} + b \le 0$,
   where $\tilde{s} = s^j + P^j\zeta$ and $\zeta \in \mathcal{R}^3$. Call the solution $\zeta^*$.
4. Set $s^{j+1} := s^j + \rho P^j \zeta^*$ for an appropriate value of the steplength $\rho \in (0, 1)$.
5. If stopping criteria are met, exit.
6. Go to 2. (At this step, if the component of the vector $s$ corresponding to
   the artificial variable $\theta$ has become nonpositive, it is eliminated from the
   problem.)
```

The search directions in step 2 are solutions to

$$\left[AD^2A^{\mathsf{T}} + Q/\beta\right] p_i = t_i, \qquad i = 1, 2, 3, \tag{2.3}$$

where β is a scalar depending on the current iterate,

$$D = \operatorname{diag}\{1/r_k, \qquad k = 1, \ldots, m\},$$

with $r_k = -(As + b)_k$, and the t_i are particular values chosen such that one of these directions is always a descent direction with respect to the objective function. The steplength ρ is set to the lesser of 99% of the distance to the boundary or the distance to the minimum of the objective function.

The form of the matrix in (2.3) allows for efficient exploitation of the sparsity. Note that if Q is positive semidefinite, then the matrix in (2.3) is positive definite for all interior points; otherwise, it may not be. In the latter case, a modification similar to that in [20] is used. In our application of this algorithm, using this procedure obviates the need for the matrix B^k to be positive definite, which in turn allows us to use the Hessian of the Lagrangian or a finite difference approximation thereof.

The standard stopping criterion for the algorithm is that at least one of the following holds: (a) the relative change in two successive values of the objective function is small; (b) the relative difference between the primal and the dual objective function values is small; or (c) the difference between two successive iterates is small. For use in our SQP algorithm we have added (d) the length of the solution vector exceeds a specified value. This additional condition has been implemented to allow for trust region strategies; in particular, this criterion will cause the algorithm to halt if (QP) is unbounded. In any case, the terminal vector will be a useful direction in the context of our purposes; this point will be discussed in the next section.

The most recent version of O3D described in [1] contains an option to perform a special "recentering step" after each subspace optimizing step (step 4) that has generally improved the efficiency. This option is not used in the results reported here. (See section 6 for a further comment.)

3. Updating the iterates: The merit functions. In this section we review the definitions and properties of our merit functions and provide formulas for updating the iterates. The reader is referred to the companion paper [4] for proofs and motivations of these concepts.

As stated in section 1, at each iteration our algorithm yields a pair (x^k, z^k), where x^k is an approximation to the solution of (NLP) and z^k is the corresponding approximate slack vector. The step directions for the updated values of these approximations are based on the (approximate) solution, (δ^k, θ^k), to the quadratic program

$$\begin{array}{l} \min\limits_{\delta,\theta} \nabla f(x^k)^\mathsf{T}\delta + \frac{1}{2}\delta^\mathsf{T} B^k \delta + M\theta \\ \text{subject to } \nabla g(x^k)^\mathsf{T}\delta + g(x^k) - e\,\theta \le 0, \end{array} \tag{3.1}$$

obtained as described in the preceding section. The vector δ^k gives the step direction for x^k, and we determine the step direction, q^k, for the slack vector z^k by the formula

$$q^k = -\left[\nabla g(x^k)^\mathsf{T}\delta^k + g(x^k) + z^k - e\,\theta^k\right]. \tag{3.2}$$

Note that if δ^k is feasible for (QP), then $\theta^k = 0$ and hence

$$q^k = -\left[\nabla g(x^k)^\mathsf{T}\delta^k + g(x^k) + z^k\right].$$

In this case $z^k + q^k$ is the slack vector for (QP) corresponding to δ^k and thus is the slack variable for the linear approximation of $g(x^{k+1})$. Given the step direction we then update the iterate by means of the formulas

$$\begin{aligned} x^{k+1} &= x^k + \alpha\delta^k, \\ z^{k+1} &= z^k + \alpha q^k \end{aligned}$$

for some value of the steplength parameter α. Observe that if $z^k \ge 0$, then the fact that (δ^k, θ^k) is feasible for (3.1) means that z^{k+1} will be nonnegative if $\alpha \in [0, 1]$. In our algorithm the nonnegativity of the slack vector iterates is preserved and, in fact, it sometimes turns out to be useful to maintain the z^k at a positive level (see section 4.8).

It is important to emphasize that the δ^k are determined by (QP), the quadratic approximation to (NLP), and are not dependent on the choice of z^k. The z^k are generated solely for use with the merit function described below. That is, we *do not* solve the slack variable problem. A comment on the notation is also in order at this point: We denote the iterate by (x^k, z^k) and the step by (δ^k, q^k), whereas conventional notation would be to use

$$\begin{pmatrix} x^k \\ z^k \end{pmatrix} \text{ and } \begin{pmatrix} \delta^k \\ q^k \end{pmatrix}.$$

It should be clear from the context what is meant.

In optimization algorithms the value of a steplength parameter is generally chosen so as to reduce the value of a suitably chosen merit function. Typically, a merit function for (NLP) is a scalar-valued function that has an unconstrained minimum at x^*, a solution to (NLP). Because a reduction in this function implies that progress is being made toward the solution, it can be used to determine an appropriate steplength in a given search direction.

In [5] and [6] a merit function for equality-constrained problems was derived that has important properties vis-à-vis the steps generated by the SQP algorithm. Using a slack-variable formulation of (NLP), a merit function for the inequality constrained problem can be constructed having the form

$$\psi_d(x,z) = f(x) + \bar{\lambda}(x,z)^{\mathsf{T}}\bar{c}(x,z) + \frac{1}{d}\bar{c}(x,z)^{\mathsf{T}}\bar{A}(x,z)^{-1}\bar{c}(x,z), \tag{3.3}$$

where z is nonnegative, d is a scalar,

$$\begin{aligned}\bar{c}(x,z) &= g(x) + z,\\ \bar{A}(x,z) &= \nabla g(x)^{\mathsf{T}}\nabla g(x) + Z,\\ \bar{\lambda}(x,z) &= -\bar{A}(x,z)^{-1}\nabla g(x)^{\mathsf{T}}\nabla f(x),\end{aligned}$$

and

$$Z = \operatorname{diag}\{z_1, \ldots, z_m\}.$$

We use this merit function (and its approximations defined below) for choosing the value of the steplength parameter α. As noted above, the approximate slack vectors generated by our algorithm, z^k, always remain nonnegative; thus the nonnegativity constraint on the z for ψ_d imposes no theoretical difficulty.

The function $\bar{c}(x,z)$ defined above plays an important role in our algorithm, as it is used to measure the feasibility of the pair (x,z). That is, if we define the function

$$r(x,z) = \|\bar{c}(x,z)\|^2, \tag{3.4}$$

where $\|\cdot\|$ denotes the standard Euclidean norm, and set

$$\mathcal{C}_\eta = \{(x,z) \,:\, r(x,z) \le \eta \text{ and } z \ge 0\}, \tag{3.5}$$

then $\mathcal{C}_0$ corresponds to the feasible set of (NLP) and hence (x^k, z^k) is close to feasible if it is in $\mathcal{C}_\eta$ for small η.

For d sufficiently small the merit function ψ_d has the desirable property that a solution of (NLP) corresponds to a (constrained) minimum of ψ_d. In addition, if d is small and δ^k is the exact solution to (QP) (which implies that $\theta^k = 0$), then the step (δ^k, q^k) is a descent direction for ψ_d when (x^k, z^k) is sufficiently close to feasibility. Despite these useful properties, ψ_d has two deficiencies that limit its use in an efficient algorithm. First, (δ^k, q^k) is a descent direction of ψ_d only near feasibility, and, second, the evaluation of ∇f and ∇g and additional nontrivial computational algebra are required to assess a prospective point. In order to overcome these difficulties, the *approximate merit function*

$$\psi_d^k(x,z) = f(x) + \bar{c}(x,z)^{\mathsf{T}}\bar{\lambda}^k + \frac{1}{d}\bar{c}(x,z)^{\mathsf{T}}(\bar{A}^k)^{-1}\bar{c}(x,z),$$

where

$$\bar{A}^k = \nabla g(x^k)^{\mathsf{T}} \nabla g(x^k) + Z^k,$$
$$\bar{\lambda}^k = -(\bar{A}^k)^{-1} \nabla g(x^k)^{\mathsf{T}} \nabla f(x^k)$$

is developed as a "working" version of ψ_d at (x^k, z^k). As the values of $\bar{\lambda}^k$ and $\bar{A}^k$ are fixed, ψ_d^k can be more easily evaluated than ψ_d in a line search algorithm for choosing an appropriate value of α. This approximate merit function, ψ_d^k, not only has essentially the same properties as ψ_d with respect to the step (δ^k, q^k), but it has the stronger property that the step is a descent direction for ψ_d^k *everywhere*. Moreover, for η sufficiently small and (x^k, z^k) outside of a ball around the solution a "sufficient" reduction in ψ_d^k implies a sufficient reduction in ψ_d. (We mean by sufficient reduction that a Wolfe condition is satisfied.) Thus we are able to use ψ_d^k as a surrogate for ψ_d for testing the progress of our iterates toward a minimum.

A further important property of the step δ^k, under the assumption that it is the exact solution to (QP), is that it is a descent direction for the function r defined by (3.4). Thus a basic algorithm for the case where the (QP) can be solved exactly is as follows: Given an initial value of η use the steps (δ^k, q^k) to reduce r until the iterates are in $\mathcal{C}_\eta$. Once the iterates are contained in $\mathcal{C}_\eta$, if a sufficient reduction in ψ_d^k does not yield a sufficient reduction in ψ_d, then reduce η. If, in the course of the algorithm, η remains bounded away from zero, then convergence follows from the fact that the Wolfe condition is satisfied for ψ_d. If η goes to zero, then convergence follows from the observation that the radius of the ball in which the Wolfe condition is not satisfied also goes to zero. This is essentially the algorithm for which global convergence is proved in the paper on the theory [4].

In this paper we are primarily interested in enhancements that convert the theoretical algorithm into one that is practical and efficient. This requires that we make provisions for situations when the assumptions under which we performed the convergence analysis are not valid and that we adopt numerical procedures to reduce the computational effort. As we note below, not all of these modifications have been (or even can be) theoretically justified, but we believe that the firm foundation of the underlying algorithm and the evidence accumulated in extensive numerical testing validate their use.

In the implementation of our algorithm a trust region constraint is used that possibly truncates the quadratic programming algorithm before an exact solution is achieved. In this case the theory described above does not apply for the step (δ^k, q^k) obtained from the approximate solution, (δ^k, θ^k), to (3.1). Although a general convergence theory based on this step is not yet available, it is shown in the theory paper [4] that if the approximate solution is obtained from the O3D algorithm and if θ^k is not too large, then the resulting step has the appropriate descent properties for the functions r, ψ_d, and ψ_d^k at (x^k, z^k). In particular, convergence can be achieved if θ^k goes to zero in a suitable manner. These properties justify our use of the truncation procedure to speed up the algorithm. It is important to note that this approximation procedure also allows us to handle the difficulty that arises in SQP methods when the quadratic subproblem is inconsistent.

4. The truncated SQP algorithm. In this section we give a somewhat detailed description of our algorithm. Initially we assume that the Hessian approximations, B^k, are positive definite, the matrices $\bar{A}^k$ are nonsingular, and the linearized constraints in (QP) are consistent. In real-world applications these assumptions are not always valid, so we have tried to make our algorithm flexible enough to perform well in situations where these assumptions fail to hold. We describe some of these adaptations at the end of this section.

The implementation of the algorithm depends upon four important parameters that need to be either computed or modified throughout the course of the algorithm. The *globalization parameter*, η, was introduced in (3.5). It is a measure of the size of the domain about the feasible region in which the direction (δ^k, q^k) is a descent direction for the true merit function ψ_d. A current estimate of η is maintained in the algorithm. The *trust region parameter*, τ, is an upper bound on the (weighted) norm of our approximate solution to (QP),

$$\|D\delta\| \leq \tau,$$

where D is a positive definite diagonal matrix. The trust region radius τ is updated at every iteration. The parameter, α, is the *steplength parameter*. It determines the length of the step in the variables (x, z) in the direction (δ^k, q^k). It is chosen to guarantee progress toward the solution in decreasing either the merit function or infeasibility. Finally, d, the *merit function parameter*, must be small enough to guarantee that the theoretical properties described in the preceding section are valid. Although the theory allows arbitrarily small values of d, the algorithm becomes very slow if d is too small, thus it is monitored throughout the algorithm and either increased or decreased as appropriate.

The outline of the algorithm is followed by specific comments on the procedures and their justifications. This version contains some of the practical modifications described above. To simplify the notation we define

$$(x_\alpha, z_\alpha) = (x^k + \alpha\delta^k, z^k + \alpha q^k).$$

Recall that r is given by (3.4).

```
Basic Truncated SQP Algorithm
  1. Initialization: Given $x^0$, $B^0$, $\tau$, $\eta$, and $d$
      a. Initialize the slack variable $z^0 \geq 0$.
      b. Set $k := 0$.
  2. Calculation of the basic trust region step:
      a. While $\|\delta\| < \tau$, iterate (using O3D) on

            $\min_{\delta} \nabla f(x^k)^\mathsf{T} \delta + \frac{1}{2}\delta^\mathsf{T} B^k \delta + M\theta$
            subject to $\nabla g(x^k)^\mathsf{T}\delta + g(x^k) - e\theta \leq 0$

         to obtain $\delta^k$ and $\theta^k$.
      b. Set

            $q^k = \begin{cases} -\left[\nabla g(x^k)^\mathsf{T}\delta^k + g(x^k) + z^k - e\theta^k\right] & \text{if } \theta^k > 0, \\ -\left[\nabla g(x^k)^\mathsf{T}\delta^k + g(x^k) + z^k\right] & \text{otherwise.} \end{cases}$

      c. Decrease $d$ if necessary.
  3. Computation of the steplength parameter:
      a. Choose $\alpha \in (0, 1]$ such that $\psi_d^k$ is sufficiently reduced.
      b. If $(x^k, z^k) \notin \mathcal{C}_\eta$ then reduce $\alpha$ if necessary until $r$ is sufficiently reduced.
      c. If $(x^k, z^k) \in \mathcal{C}_\eta$ then reduce $\alpha$ if necessary so that $(x_\alpha, z_\alpha) \in \mathcal{C}_\eta$.
```

4. Update of the estimate of the globalization parameter:
   a. If
   $$\psi_d(x_\alpha, z_\alpha) > \psi_d(x^k, z^k),$$
   set $\eta = \frac{1}{2} r(x^k, z^k)$.
5. Update of the variables and check for termination:
   a. Set
   $$x^{k+1} := x^k + \alpha\delta^k,$$
   $$z^{k+1} := z^k + \alpha q^k.$$
   b. If convergence criteria are met, quit.
   c. Update B^k to B^{k+1}.
6. Adjustment of the merit function and trust region parameters:
   a. Update d if necessary.
   b. Adjust the trust region radius τ.
7. Return:
   a. Set $k := k + 1$.
   b. Go to step 2.

4.1. The globalization parameter. The globalization step is based on work in [6] and [4]. In step 3 we require that the approximate merit function be reduced and, in addition, if the current iterate lies outside the set $\mathcal{C}_\eta$, we require that the constraint infeasibilities also be reduced. This is possible as a result of the descent properties described in section 3. If we have a good estimate of η and $(x^k, z^k) \in \mathcal{C}_\eta$, then the true merit function can also be reduced; if this is not the case, then our estimate of η is too large and we reduce its value in step 4. This procedure will eventually lead to a sufficiently small value of η. Note that this arrangement allows steps that may increase the merit function, but only in a controlled way. It also allows steps that may increase the constraint infeasibilities, but only when inside of $\mathcal{C}_\eta$.

4.2. Updating τ. Our procedure for updating τ, the trust region radius, in step 6b is similar to the standard strategy used in trust region algorithms (see [17] or [31]) in that we base the decision on how to change τ on a comparison of a predicted relative reduction, $pred_k$, and an actual relative reduction, $ared_k$, in a function used to measure the progress toward the solution. (Various formulas for the predicted relative reduction, $pred_k$, have been suggested for different merit functions, especially for equality constrained programming problems; see, for example, [19].) What is distinctive about our procedure is that we use different functions for computing $pred_k$ and $ared_k$ depending on the current status of the algorithm. When the linearized constraints are satisfied we use the approximate merit function to compute the predicted and actual reductions. When the trust region constraint causes O3D to terminate in Phase I, i.e., when the linearized constraints are not satisfied, predicted and actual reductions in infeasibility are used.

In the case when a feasible solution to (QP) is obtained, then ψ_d^k is used to compute the predicted and actual reductions. Our method for defining $pred_k$ differs from the standard methods used in unconstrained optimization because the step-finding subproblem is not based solely on the merit function and, moreover, the trust region constraint does not appear explicitly in the subproblem. Nevertheless, in updating τ we want to assess how well an approximation to ψ_d^k agrees with ψ_d^k in the direction (δ^k, q^k). Since (QP) uses a quadratic approximation of the Lagrangian for the objective function with linearized constraints, we form our approximation to ψ_d^k based on a quadratic approximation to the function ψ_1^k given by

$$\psi_1^k(x,z) = f(x) + \bar{c}(x,z)^\mathsf{T}\bar{\lambda}^k$$

and a linear approximation to

$$\psi_2^k(x,z) = \bar{c}(x,z)^\mathsf{T}(\bar{A}^k)^{-1}\bar{c}(x,z).$$

Note that $\psi_d^k(x,z) = \psi_1^k(x,z) + (1/d)\psi_2^k(x,z)$. Based on these considerations and the results of [16] we define the predicted relative reduction by

$$\begin{aligned} pred_k = \Bigg\{ & -\alpha^k \nabla\psi_1^k(x^k,z^k)^\mathsf{T}(\delta^k,q^k) - \frac{(\alpha^k)^2}{2}(\delta^k,q^k)^\mathsf{T}\nabla^2\psi_1^k(x^k,z^k)(\delta^k,q^k) \\ & - \frac{\alpha^k}{d}\nabla\psi_2^k(x^k,z^k)^\mathsf{T}(\delta^k,q^k)\Bigg\} \Big/ \psi_d^k(x^k,z^k), \end{aligned} \tag{4.1}$$

where the derivatives are with respect to x and z and the steplength parameter α^k is the size of the most recently accepted step. The value of the actual relative reduction, $ared_k$, is taken to be the difference in the values of ψ_d^k at the points (x^{k+1}, z^{k+1}) and (x^k, z^k) divided by the value of $\psi_d^k(x^k, z^k)$. A valid criticism of the formula for $pred_k$ is its dependence on higher order derivatives. Therefore, we use the available approximation of the Hessian of the Lagrangian for $\nabla^2\psi_1^k$. For example, cell-centered finite difference approximations to the Hessian of the Lagrangian function were used in the numerical results presented here, unless analytic second derivative formulas were readily available.

The above choice for $pred_k$ is not used when the step returned by O3D is not feasible. In these situations the resulting step is dominated by a feasibility-improving component, and it makes little sense for the adjustment to τ to be determined by ψ_d^k; rather, a comparison of the predicted and actual improvements in constraint infeasibility seems more appropriate. Therefore, in this case the function $r(x,z)$ is used for comparison purposes. The values of $pred_k$ and $ared_k$ are given as follows for the case when the O3D algorithm terminates in Phase I:

$$pred_k = \left\{ r(x^k,z^k) - \left\| \alpha^k \nabla g(x^k)^\mathsf{T}\delta^k + g(x^k) + z^{k+1} \right\|^2 \right\} / r(x^k,z^k)$$

and

$$ared_k = \left\{ r(x^k,z^k) - r(x^{k+1},z^{k+1}) \right\} / r(x^k,z^k).$$

These heuristics for choosing $pred_k$ and $ared_k$ appear to work well. Specifically, they allow the trust region radius, τ, to be increased even in the event that the step returned by O3D does not satisfy linearized constraints or it results in an increase in the true merit function. In our experience, the alternative formulas based solely on constraint violations are never employed close to the solution. Indeed, the iterates preceding convergence have always been observed to be well inside $\mathcal{C}_\eta$, where satisfying the linearized constraints and decreasing the merit functions usually pose no problem.

4.3. The steplength α. The steplength α is determined in step 3 of the algorithm. The "sufficient decrease" referred to in 3a and 3b requires that the Wolfe condition be satisfied. For a given function ϕ and potential step w from point v, this condition requires that α satisfy

$$\phi(v + \alpha w) \leq \phi(v) + \sigma\, \alpha\, \nabla\phi(v)^{\mathsf{T}} w$$

for some fixed $\sigma \in (0, 1)$. In the numerical experiments reported in section 5 we employed a simple backtracking procedure (with factor one-half) to find α to satisfy this condition for both ψ_d^k and for r. We have also experimented with more sophisticated line search methods motivated by unconstrained optimization techniques as in [18], but the observations to date suggest that the more complicated line searches result in very little improvement of our algorithm, except when the iterates are quite far from the solution.

4.4. Adjusting *d*. Choosing an effective value for the merit function parameter d is essential in our algorithm. While it is clear that (in a compact set) a sufficiently small value of d will ensure that the results given in [4] are valid, there are three very important practical reasons why the parameter must be adjusted rather than fixed. First, if the angle between the direction generated by O3D and the gradient of the approximate merit function becomes nearly orthogonal, the steps might become too small. We adjust d to avoid this possibility. Second, the *approximate* merit function, ψ_d^k, is changing at each iteration, and it is possible a previous iterate might be acceptable to the current ψ_d^k; i.e., cycling might occur. This worry can also be alleviated by adjusting d. A third reason for changing d is to allow for larger steps. It is seen from the theory and has been verified by numerical experience that if d is too small then the form of the merit function forces the path of the iterates to follow the "nearly active" constraints closely. This causes the algorithm to take very small steps and, in particular, to be slow in moving away from a nonoptimal active set. By making it possible to increase d we can significantly improve the algorithm's performance.

In the implementation of our algorithm there are two opportunities to adjust d: in step 2, after solving the quadratic subproblem, and in step 6, after the step has been taken. In the first of these adjustments d can only be decreased; in the second, the parameter may be increased or decreased.

In step 2, the angle between the gradient of the approximate merit function $\nabla\psi_d^k$ and the step direction (δ^k, q^k) is computed. If these two vectors become nearly orthogonal, we conclude that d is not small enough to ensure a good decrease in ψ_d^k, and we decrease the parameter. To be more specific, we compute

$$w(d) = \frac{(\nabla\psi_d^k(x_k, z_k))^{\mathsf{T}}(\delta^k, q^k)}{\left\|\nabla\psi_d^k(x_k, z_k)\right\| \cdot \|(\delta^k, q^k)\|}.$$

If $w(d) \geq -.1$ we calculate a value $\hat{d}$ so that $w(\hat{d}) \approx -.5$. We safeguard the procedure by not allowing more than a certain percentage decrease in d. In the current version we use 50%.

If d was not decreased in step 2 we consider modifying it after a step has been taken (step 6). Here the primary concern is to avoid cycling. To do so we compute an interval for the penalty parameter as follows. For a fixed integer κ we seek a value of the parameter, $\bar{d}$, such that

$$\psi_{\bar{d}}^k(x^k, z^k) < \psi_{\bar{d}}^k(x^{k-i}, z^{k-i}), \qquad i = 1, \ldots, \kappa. \tag{4.2}$$

Inequality (4.2) implies that none of the past κ iterates will be acceptable to the approximate merit function with the new value of $\bar{d}$. (Thus if $\kappa = k$, no cycling would be possible.) To accomplish this, we use the decomposition

$$\psi_d^k = \psi_1^k + \frac{1}{d}\psi_2^k, \tag{4.3}$$

where ψ_1^k and ψ_2^k are defined in section 4.2. We then compute the values of $\psi_1^k(x^{k-i}, z^{k-i})$ and $\psi_2^k(x^{k-i}, z^{k-i})$, $i = 1, \ldots, \kappa$, and consider the inequalities

$$\psi_1^k(x^k, z^k) + \frac{1}{d}\psi_2^k(x^k, z^k) < \psi_1^k(x^{k-i}, z^{k-i}) + \frac{1}{d}\psi_2^k(x^{k-i}, z^{k-i}). \tag{4.4}$$

We define d_i^u and d_i^l to be the upper and lower values of d that ensure that inequality (4.4) is satisfied. Then letting

$$d^u = \min\{d_i^u : i = 1, \ldots, \kappa\} \tag{4.5}$$

and

$$d^l = \max\{d_i^l : i = 1, \ldots, \kappa\}, \tag{4.6}$$

we obtain an interval (d^l, d^u). Assuming that this interval exists, it is the case that if the value of d for the next step is chosen in this interval, the next iterate will not return to one of the previous κ iterates. In practice a value of $\kappa \approx 5$ is usually more than sufficient to prevent cycling. If the interval doesn't exist, then we make no change.

Given that we can choose d to avoid cycling, our second objective at this juncture is to increase d to allow bigger steps. If the d^u is larger than the current d, then we can safely increase d without worrying about possible cycling. However, we safeguard this increase in two ways. First, we require that the predicted reduction based on the approximate merit function must be greater than the predicted reduction of infeasibility in the linearized constraints. This restriction prevents d from being increased prematurely due primarily to a large decrease in constraint infeasibilities. Specifically, writing the predicted reduction in ψ_d^k (see (4.1)) as

$$\mathcal{P}_\mathcal{Q} + \frac{1}{d}\mathcal{P}_\mathcal{L},$$

we insist that for a new value of d

$$\mathcal{P}_\mathcal{Q} + \frac{1}{d}\mathcal{P}_\mathcal{L} > \mathcal{P}_\mathcal{L}. \tag{4.7}$$

Second, we use a maximum allowable change (currently a factor of 2) to limit the growth of d. Computationally, these simple procedures for updating d appear to be effective, especially in the presence of highly nonlinear constraints and poorly scaled problems.

4.5. The Hessian approximation. In the numerical experimentation reported here, we have used a finite difference approximation to the Hessian of the Lagrangian as B^k. Although the Hessian of the Lagrangian at a strong solution is positive definite on the appropriate subspace, it may be indefinite in general. Even if it is positive definite, the finite difference approximation may not be. We experimented with two approaches for handling this possibility. First, we simply modified the approximate Hessian matrix by adding nonnegative elements to the diagonal ensuring that the Cholesky factorization of the matrix had positive elements along its diagonal (see [20]). This modification was easy to implement, but it was observed to slow convergence on some problems. While this modification guarantees that a positive definite matrix will be delivered to the (QP) solver, if it takes place when the iterates get close to the solution, it generally precludes local q-superlinear convergence.

An alternative to modifying the approximate Hessian of the Lagrangian is simply to allow O3D to iterate on the indefinite QP subproblem, halting the iterations when the solution exceeds the trust region radius. We implemented this approach and it seemed to yield superior results to those obtained by making the approximate Hessian positive definite (especially when the iterates were close to a solution) although, theoretically, we can prove only that we obtain a descent direction when the approximate Hessian is positive definite.

4.6. Convergence criteria. The convergence criteria used are standard and similar to those in [3]. We first insist that the constraints be satisfied to a close tolerance; specifically, we require

$$\left\|\max(g(x^k), 0)\right\|_\infty \le 10^{-6}. \tag{4.8}$$

We also require that either

$$\frac{\|\nabla f(x^k) + \nabla g(x^k)\lambda^k\|}{|f(x^k)|} \le 10^{-7} \tag{4.9}$$

or

$$\left\|x^k - x^{k-1}\right\|_\infty \le 10^{-8}(1 + \left\|x^k\right\|). \tag{4.10}$$

The criterion (4.9) is a stronger indication that a KKT point has been reached. The weaker criterion (4.10) suggests that progress slowed drastically and that iterates may or may not have drawn close to a solution. For this reason criterion (4.9) is usually preferable to criterion (4.10). The Lagrange multipliers returned by the quadratic program are used in (4.9) unless the trust region constraint determines the approximate solution of the (QP). In that case, we use the least squares approximation to the multipliers, replacing all negative multipliers with machine zeros. In all of the problems solved to date, the trust region never comes into play when the iterates get close to the solution; therefore, the (QP) multipliers are used for the convergence test at the solution.

4.7. Inconsistent quadratic subproblems. One difficulty that can occur when making linear approximations to nonlinear constraints is that (QP) may be inconsistent. In this case O3D will, even if it runs to completion, not exit Phase I and will return a positive value of the artificial variable. (Note that this always occurs if equality constraints are present.) For small θ the resulting direction is a descent direction for ψ_d^k and for r. As a result, the step taken in this direction will generally decrease infeasibility, making it less likely that an inconsistent set of linearized constraints will be encountered during subsequent iterations.

More recent versions of our algorithm include a constraint relaxation procedure that appears to yield an acceptable step, δ_k, even in the event that inconsistent linearizations of constraints are encountered. Because this situation did not surface during the numerical experiments presented in this paper, we do not include a description of our perturbation procedure. We do note, however, that we have encountered important application problems where this procedure was crucial to the performance of our algorithm (see, for example, [24]).

4.8. Updating slack variables. One difficulty in our algorithm is the updating of slacks in the event that the SQP step does not satisfy the linearized constraints well enough, i.e., θ^k is not small enough. This can occur when (QP) is inconsistent or when a trust region bound is encountered during the solution of (QP). In this case our slack variable updating scheme would ensure that nonnegative slacks remain nonnegative, but the direction may not be one of descent. We resolve this dilemma by opting for descent, i.e., computing q^k with $\theta^k = 0$ and replacing any negative slacks using the following rule:

$$\text{If } z_i^{k+1} < 0, \text{ then set}$$
$$z_i^{k+1} = \begin{cases} \epsilon_{Mach}, & g_i(x^{k+1}) \geq 0, \\ -g_i(x^{k+1}), & g_i(x^{k+1}) < 0, \end{cases}$$

where ϵ_{Mach} is machine epsilon. This is sometimes referred to as "closing" the constraints (see, for example, [33]).

4.9. Linearly dependent constraint gradients. Linearly dependent constraint gradients cause many theoretical and computational difficulties in constrained optimization. In our theoretical algorithm we obtain convergence even when there are linearly dependent constraint gradients provided the approximate multipliers do not become unbounded. In practice although O3D has no difficulty in dealing with this problem, evaluating the merit function and computing the least squares approximation to the Lagrange multipliers become problematical. Computational experience shows that we solve many problems with degeneracy in the constraints. Simply maintaining slacks to be positive as described above allows us to factor the crucial matrices and continue with the algorithm. However, the algorithm failed to solve some problems that had a large amount of degeneracy in the linearized constraint matrix. This was, of course, problem dependent but it was observed that the current implementation can usually solve problems where up to 25% of the constraint gradients are linearly dependent. This degeneracy causes the performance of the merit functions to deteriorate. In particular, the least squares approximation to Lagrange multipliers seems to be especially poor, resulting in only very small steps being allowed, even close to the solution.

5. Numerical results. The modified algorithm was coded in Fortran and is installed on a SPARCstation 10 using *IEEE floating point arithmetic* (64 bit). The current implementation is being used to solve a wide variety of medium to large scale problems. In this section we report the results of a set of performance tests designed specifically to answer questions about the trust region strategy and the procedure to update the penalty parameter, d. We conclude the section with the results of our algorithm applied to some test problems that are publicly available. We emphasize that all of the problems were solved with the same default settings of the parameters (see Table 5.1); i.e., no attempt was made to pick parameter settings to optimize performance on individual problems.

Although in many of the applications some analytic derivatives were available, no use of analytic derivative information was used in these numerical experiments. When possible, first and second derivatives were computed using forward and central

Table 5.1
Numerical values of default parameters.

Parameter	Value
M	$10\min\{10^7, \\|\nabla f(x_0)\\|_\infty \min\{10^3, \\|\nabla f(x_0)\\|_\infty\}\}$
θ^*	$2\\|g(x_0)\\|_\infty$
τ_0	$(\\|g(x_0)\\|_\infty + \\|x_0\\|_2)$
η_0	$(1 + \\|c(x_0, z_0)\\|_\infty)^2$
z_0	$\epsilon_{Mach} + \max(-g(x_0), \epsilon_{Mach})$
σ	10^{-4}

finite differences, respectively. A costly one-time calculation provided a zero/nonzero stencil of the Hessian of the Lagrangian and the Jacobian matrix of the constraint function. These stencils were then used for the duration of the solution process. For some problems, these finite difference approximations are not convenient to use. This can be the case with control problems governed by partial differential equations (see [29] or [30]). If the partial differential equation is solved using a finite element method, with piecewise linear elements, then evaluating the derivative of the objective function with respect to the control variables can be quite cumbersome. In such cases, which occurred in the control problems in our test suite, one can approximate the first derivatives of the objective function by solving an adjoint problem with a computational cost comparable to one function evaluation. (For examples, see [22].) The objective function portion of the Hessian of the Lagrangian can then be approximated with forward finite differences.

A set of eight problems was chosen as the first test suite. These problems ranged in size from 500 to 1000 variables and from 1000 to 2000 constraints. The first four are relatively straightforward nonlinear programming test examples, while the last four are from actual applications: two discretized control problems, a density estimation problem from statistics, and a "molecular distance" problem. A more complete description of these problems is found in the appendix. The problems all have nonlinear inequality constraints and exploitable sparsity. Problem 4 (NLP4) was designed to have a controllable percentage of linear dependency in the constraint gradients to demonstrate any weaknesses in the algorithm associated with this difficulty. We ran three versions of our algorithm on each problem: using a positive definite modification of the Hessian matrix, as discussed in section 4, with and without the trust region strategy, and using the unmodified Hessian with the trust region. (Using the unmodified Hessian results in failure in most cases if no trust region strategy is employed.) In addition, each problem was run from two starting points: one, labeled "c," which was close to the solution in the sense that each of the variables was of the same order of magnitude as in the solution, and a distant start, labeled "f."

The results of the numerical tests on these problems are summarized in Tables 5.2–5.4. The first two columns of each table give the number of SQP iterations ("nl-i") and the total number of O3D iterations ("qp-i"). The next two columns contain the stopping criterion that was met and the value of the gradient of the Lagrangian at the solution. Unless the algorithm failed (which is denoted by "Failure" in the tables), feasibility condition (4.8) was satisfied for all solutions. The stopping criterion is denoted by either a 1 or a 2 depending on whether (4.9) or (4.10) was satisfied. If both conditions were satisfied, a 3 appears in the column. The remaining columns give information about the values of the parameter d for each run: columns 5–8 give the initial, maximum, minimum, and final values of this parameter, and the final column gives the last iteration at which d was changed.

Table 5.2
Modified Hessians with no trust region.

Problem	nl-i	qp-i	Conv	$\|\nabla_x l\|_\infty$	d_0	Max d	Min d	Final d	Last d-cha
NLP1-c	37	1435	2	1.2e-7	1.00e00	2.08e00	4.45e-2	9.35e-1	34
NLP1-f	49	1656	1	7.3e-8	1.00e00	1.07e00	5.89e-2	9.20e-1	46
NLP2-c	66	2211	1	1.2e-7	1.00e00	2.83e00	6.89e-2	1.72e00	61
NLP2-f	71	2369	1	3.1e-8	6.98e-1	3.00e00	8.31e-2	1.51e00	64
NLP3-c	29	983	1	6.7e-8	1.00e00	1.12e00	8.13e-1	1.03e00	22
NLP3-f	39	1314	1	4.2e-8	1.00e00	1.05e00	9.84e-1	1.03e00	31
NLP4-c	–	6		Failure	1.00e00	–	–	–	Failure
NLP4-f	–	6		Failure	1.00e00	–	–	–	Failure
Truss-c	103	3561	1	4.4e-8	9.87e-1	1.01e00	9.57e-2	9.57e-1	100
Truss-f	110	3799	2	1.9e-7	1.00e00	1.08e00	8.93e-2	8.93e-1	106
Stat-c	135	4561	3	1.1e-8	1.00e00	2.33e00	8.25e-2	9.70e-1	129
Stat-f	144	4805	1	3.3e-8	1.00e00	2.16e00	7.77e-2	8.49e-1	140
BCHeat-c	257	5398	1	7.8e-8	9.18e-1	1.98e00	5.23e-2	1.24e00	254
BCHeat-f	289	5971	1	1.9e-7	1.00e00	4.21e00	4.92e-2	1.37e00	281
Molec-c	37	1376	1	9.8e-9	9.88e-1	1.21e1	1.17e-2	9.81e-1	34
Molec-f	41	1437	2	6.5e-6	1.00e00	2.38e0	1.49e-1	5.52e-1	39

Table 5.3
Modified Hessians with trust region.

Problem	nl-i	qp-i	Conv	$\|\nabla_x l\|_\infty$	d_0	Max d	Min d	Final d	Last d-cha
NLP1-c	94	1412	2	3.2e-7	1.00e00	8.58e00	3.13e-3	6.55e-2	88
NLP1-f	108	2947	2	6.8e-8	1.00e00	7.50e00	1.23e-3	2.60e-2	99
NLP2-c	213	2744	1	1.6e-7	1.00e00	3.85e00	1.28e-3	6.48e-1	209
NLP2-f	231	2963	1	5.4e-8	6.98e-1	1.10e01	5.67e-3	8.57e-1	221
NLP3-c	42	932	1	3.7e-8	1.00e00	1.64e00	4.27e-1	9.83e-1	39
NLP3-f	44	946	1	9.1e-8	1.00e00	1.53e00	2.71e-1	9.22e-1	38
NLP4-c	199	2582	2	9.2e-6	1.00e00	1.41e00	8.92e-1	1.18e00	49
NLP4-f	201	2599	2	5.2e-7	1.00e00	2.01e00	9.94e-1	1.21e00	55
Truss-c	195	3528	1	6.1e-8	9.87e-1	1.13e00	9.87e-1	1.11e00	189
Truss-f	195	3544	2	2.9e-7	1.00e00	1.47e00	1.00e00	1.47e00	188
Stat-c	144	4519	3	2.3e-8	1.00e00	2.39e00	1.00e00	1.53e00	140
Stat-f	150	4581	1	4.2e-8	1.00e00	2.48e00	1.00e00	1.89e00	144
BCHeat-c	257	2898	1	8.1e-8	9.18e-1	4.15e00	1.38e-1	4.10e00	249
BCHeat-f	289	3071	1	9.9e-8	1.00e00	3.74e00	2.44e-1	3.89e00	281
Molec-c	39	546	1	1.3e-7	9.88e-1	1.71e01	3.74e-2	2.22e00	36
Molec-f	44	621	1	6.6e-8	1.00e00	1.48e00	7.39e-2	9.52e-2	38

The results of the tests illustrate that using the unmodified Hessian with the trust region was most effective in reducing the number of O3D iterations *and* the number of SQP iterations. The trust region strategy prevented long, unprofitable steps from being generated when far from the solution, and the use of the unmodified Hessian allowed the trust region to become inactive near the solution, thus allowing rapid local convergence. Requiring the Hessian to be positive definite often precluded rapid local (q-superlinear) convergence and, when used in conjunction with the trust region strategy, resulted in the trust region's being active close to the solution.

The results also show that the value of the parameter d varied over several orders of magnitude. The procedures discussed in section 4 that allowed the value of d to

TABLE 5.4
Unmodified Hessians with trust region.

Problem	nl-i	qp-i	Conv	$\|\nabla_x l\|_\infty$	d_0	Max d	Min d	Final d	Last d-cha
NLP1-c	43	820	2	2.1e-7	1.00e00	3.69e00	5.13e-2	7.13e-1	38
NLP1-f	46	913	3	1.7e-8	1.00e00	6.58e00	6.27e-2	6.14e-1	39
NLP2-c	51	1330	1	9.8e-8	1.00e00	4.11e00	9.65e-2	5.95e-1	44
NLP2-f	53	1351	1	1.1e-7	6.98e-1	2.94e00	1.20e-1	2.47e-1	48
NLP3-c	35	832	1	4.5e-8	1.00e00	1.89e00	2.46e-2	3.79e-1	29
NLP3-f	39	867	1	7.3e-8	1.00e00	1.57e00	2.22e-2	4.52e-1	28
NLP4-c	–	–		Failure	–	–	–	–	Failure
NLP4-f	–	–		Failure	–	–	–	–	Failure
Truss-c	94	2242	1	3.9e-8	9.87e-1	1.03e00	1.26e-1	9.11e-1	87
Truss-f	96	2261	1	6.6e-8	1.00e00	1.33e00	5.67e-2	7.84e-1	85
Stat-c	121	1577	3	1.1e-8	1.00e00	2.58e00	1.57e-1	9.34e00	114
Stat-f	121	1585	1	4.7e-8	1.00e00	2.19e00	1.65e-1	7.03e00	111
BCHeat-c	231	2498	1	1.2e-7	9.18e-1	3.24e00	6.04e-2	8.83e-1	226
BCHeat-f	239	2871	3	2.4e-8	1.00e00	1.61e01	3.89e-2	4.98e-1	222
Molec-c	39	550	1	8.76e-8	9.88e-1	1.02e00	4.34e-2	6.04e-1	36
Molec-f	45	658	2	2.3e-7	1.00e00	8.78e00	1.35e-2	6.53e-1	42

increase or decrease greatly enhanced the algorithm; earlier tests using either a fixed value of d or only allowing a reduction in d yielded inferior results.

Another modification in our algorithm, not reflected in the table or included in the description in the preceding section, was made to force the O3D algorithm to take a minimum number of steps. We found that when the trust region radius τ became small the algorithm would sometimes exit O3D after only one iteration, resulting in a poor step direction. This poor step would result in a further decrease in τ, and eventually the algorithm would fail. When we required a minimum number of steps to be taken in O3D (our choice was seven), this problem disappeared.

Recently a collection of test problems has become available for the testing and comparing of optimization algorithms (see [13]). The problems in the Constrained and Unconstrained Testing Enviroment (CUTE) are quickly becoming standards with which researchers can establish the viability and effectiveness of their numerical algorithms. These problems are replacing the smaller and well-scaled test problems of Hock and Schittkowski [25] and Schittkowski [32], which were not intended to be used to test large scale algorithms. Our results on the CUTE test problems are summarized in Tables 5.5–5.7. These problems were solved to the same stopping conditions as the problems above. Likewise, the same table format was used to present these numerical results. For a detailed description of these problems and their structure, motivation, and sources, see [9].

While it appears that the CUTE test problem set is rich in both large and small scale unconstrained and equality constrained test problems, at present there are not many large scale problems that include inequality constraints (and particularly nonlinear inequality constraints). We chose problems that reflected the class of problems our algorithm was designed to solve. At least one inequality constraint was present in each problem. The number of variables or constraints was large enough so that the exploitation of special sparsity structure was important. The problems we selected from CUTE to report on were CORKSCREW, MANNE, SVANBERG, and ZIGZAG. The associated problem sizes are recorded in Table 5.8.

TABLE 5.5
Modified Hessians with no trust region.

Problem	nl-i	qp-i	Conv	$\|\nabla_x l\|_\infty$	d_0	Max d	Min d	Final d	Last d-cha
CORKSCREW-c	4	73	1	2.6e-8	1.d0	1.d0	1.d0	1.d0	0
CORKSCREW-f	5	90	1	3.3e-8	1.d0	1.d0	1.d0	1.d0	0
MANNE-c	8	144	1	1.9e-8	1.d0	1.d0	1.d0	1.d0	0
MANNE-f	8	146	2	1.3e-7	1.d0	1.d0	1.d0	1.d0	0
SVANBERG-c	6	111	1	1.9e-8	1.d0	1.d0	1.d0	1.d0	0
SVANBERG-f	6	111	1	3.9e-8	1.d0	1.d0	1.d0	1.d0	0
ZIGZAG-c	5	93	1	1.8e-8	1.d0	1.d0	1.d0	1.d0	0
ZIGZAG-f	6	99	1	9.9e-9	1.d0	1.d0	1.d0	1.d0	0

TABLE 5.6
Modified Hessians with trust region.

Problem	nl-i	qp-i	Conv	$\|\nabla_x l\|_\infty$	d_0	Max d	Min d	Final d	Last d-cha
CORKSCREW-c	4	71	1	4.1e-8	1.d0	1.d0	1.d0	1.d0	0
CORKSCREW-f	5	87	1	5.2e-8	1.d0	1.d0	1.d0	1.d0	0
MANNE-c	8	141	1	2.8e-8	1.d0	1.d0	8.51d-1	8.51d-1	2
MANNE-f	8	142	1	5.4e-8	1.d0	1.d0	8.13d-1	8.13d-1	3
SVANBERG-c	5	91	1	2.5e-8	1.d0	1.d0	1.d0	1.d0	0
SVANBERG-f	6	100	1	1.3e-8	1.d0	1.d0	1.d0	1.d0	0
ZIGZAG-c	5	89	1	1.8e-8	1.d0	1.d0	1.d0	1.d0	0
ZIGZAG-f	5	91	1	1.6e-8	1.d0	1.d0	1.d0	1.d0	0

It is worth commenting that much of the machinery developed in this paper deals with effectively handling nonlinear inequality constraints. The performance of our algorithm on the CUTE test problem set is, therefore, slightly deceiving since many of the constraints in these problems are simple bounds on the primal variables or are purely linear. (For instance, approximately 83% of the constraints in CORKSCREW, 50% of the constraints in MANNE, and 66% of the constraints in ZIGZAG were linear, and many of them were equality constraints.) Although these caused no problem for our algorithm, the structure of these constraints was not completely exploited and the extra machinery of our code resulted in an overhead with no performance benefit. Clearly, an algorithm designed specifically to deal with linear equality constraints should outperform our algorithm on these problems. The problem on which our algorithm appeared to perform best was SVANBERG, a problem with only inequality constraints (a substantial number of which are nonlinear).

We succeeded in solving all four problems with a reasonable number of inner and outer iterations. However, many of our algorithmic enhancements contributed little to the solution process. The measure of distance to feasibility (the η-tube strategy), the nonmonotone updating of penalty parameter d, and the trust region strategy were essentially dormant during the solution process regardless of the iterates' proximity to the solution or to feasibility. In fact, the only evidence of our enhancements on the small number of CUTE test problems that we solved occurred when d was decreased slightly while solving the problem MANNE employing modified Hessians with a trust region strategy (see the third and fourth rows of Table 5.6). It is noteworthy that the iterates that resulted from solving this problem with the penalty parameter artificially held fixed at $d = 1$ were identical to iterates that resulted for the adjusted d solution. This appears to illustrate that in this case the adjustment of d was purely superficial.

Table 5.7
Unmodified Hessians with trust region.

Problem	nl-i	qp-i	Conv	$\|\nabla_x l\|_\infty$	d_0	Max d	Min d	Final d	Last d-cha
CORKSCREW-c	3	39	1	1.1e-8	1.d0	1.d0	1.d0	1.d0	0
CORKSCREW-f	4	43	1	1.9e-8	1.d0	1.d0	1.d0	1.d0	0
MANNE-c	5	64	1	2.4e-8	1.d0	1.d0	1.d0	1.d0	0
MANNE-f	6	75	1	1.2e-8	1.d0	1.d0	1.d0	1.d0	0
SVANBERG-c	3	30	3	1.0e-8	1.d0	1.d0	1.d0	1.d0	0
SVANBERG-f	3	38	3	9.5e-9	1.d0	1.d0	1.d0	1.d0	0
ZIGZAG-c	3	38	1	9.3e-9	1.d0	1.d0	1.d0	1.d0	0
ZIGZAG-f	4	41	1	4.8e-8	1.d0	1.d0	1.d0	1.d0	0

Table 5.8
Minimization parameters.

Problem	Variables	Constraints
CORKSCREW	96	159
MANNE	300	600
SVANBERG	500	1500
ZIGZAG	304	1206

6. Future directions. In this paper we have discussed in some detail an SQP algorithm for solving large scale nonlinear problems. The numerical results with default parameter settings indicate that the procedures that we have implemented are robust, effective, and efficient; the convergence theory in [4] provides a sound theoretical basis for the procedure. Nevertheless, there are several areas in which the techniques used here can be improved to allow the solution of larger and more difficult problems.

Algorithmically, we observe that the current implementation requires the factorization of both $(\nabla g^{\mathsf{T}}\nabla g + Z)$ and $(\nabla g \nabla g^{\mathsf{T}})$, the latter in O3D. While the sparse matrix package makes this reasonable for the problems that we have currently considered, it is clearly expensive to maintain both.

The results reported here use analytic or finite difference Hessian approximations. An examination of the details of O3D reveals that a limited memory BFGS or limited memory SR1 could be readily incorporated into the code. We have done some experimentation with such techniques; the results will be reported elsewhere [26].

Many of the problems that we have seen have been degenerate, and this significantly slows the convergence of the method. The primary culprit is the extremely poor multiplier estimates provided by the least squares procedure. Improvements in this area are certainly required.

In some problems (not reported here) that have nonlinear equality constraints, we have occasionally observed significant difficulty in trying to satisfy the linearized equality constraints, i.e., in completing Phase I. In these cases we have had some success in relaxing the constraints [26]. In the context of O3D, this can be accomplished by simply fixing the artificial variable at some positive value and continuing the O3D iterations. In this approach, we often find that O3D converges, and the "recentering" procedure mentioned in section 2 has led to further improvements. The theory in [4] supports these ideas. The details will, again, be reported elsewhere.

Appendix. Problem Descriptions.

Nonlinear Program # 1 (NLP1)

$$\begin{aligned}
&\min f(x) = \tfrac{1}{2}((x_1 - x_{100})x_2 + x_{101})^2\\
&\text{subject to}\\
&\qquad x_1 x_{i+1} + \left(1 + \frac{2}{i}\right) x_i x_{100} + x_{101} \le 0, \qquad i = 1, \ldots, 99,\\
&\qquad (\sin(x_j))^2 - \tfrac{1}{2} \le 0, \qquad j = 1, \ldots, 100,\\
&\qquad \sin((x_j)^2) \ge 0, \qquad j = 1, \ldots, 100,\\
&\qquad x_j \le j, \qquad j = 1, \ldots, 100,\\
&\qquad -x_j \le 1, \qquad j = 1, \ldots, 100,\\
&\qquad (x_1 + x_{100})^2 = 1.
\end{aligned}$$

Explanation. This problem with 101 variables and 500 constraints is taken from [13], where it was used to illustrate separability in nonlinear programming.

Nonlinear Program # 2 (NLP2)

$$\begin{aligned}
&\min f(x) = 1000 \left[\left(\sum_{i=1}^{n} x_i^3 \right)^2 - \left(\sum_{i=1}^{n} x_i^2 \right) \left(\sum_{i=1}^{n} x_i^4 \right) \right]\\
&\text{subject to}\\
&\qquad x_1 \ge 0 \quad \text{and} \quad x_n \le 1,\\
&\qquad x_i - x_{i+1} \le 0, \qquad i = 1, \ldots, (n-1),\\
&\qquad x_i^2 - x_i x_{i+1}^2 \le 0, \qquad i = 1, \ldots, (n-1).
\end{aligned}$$

Explanation. There are many local extrema for this problem; we made no special effort to locate global minima. The objective function is highly nonlinear and has a dense Hessian, but the constraints have sparse banded first derivatives. Our example uses $n = 250$.

Nonlinear Program # 3 (NLP3)

$$\begin{aligned}
&\min f(x) = \sum_{i=1}^{n} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right]\\
&\text{subject to}\\
&\qquad x_1 \ge 0 \quad \text{and} \quad x_n \ge 0,\\
&\qquad x_i - x_{i+1} \le 0, \qquad i = 1, 3 \ldots, (n-1),\\
&\qquad 4x_{i+1} - x_i^2 - 4 \le 0 \qquad i = 1, 3 \ldots, (n-1),\\
&\qquad 2x_{i+1} + x_i - 1 \le 0 \qquad i = 1, 3 \ldots, (n-1).
\end{aligned}$$

Explanation. The objective function here is Rosenbrock's function. The objective function is nonlinear and has a tridiagonal Hessian. The constraints have sparse banded first derivatives. We solved the problem with $n = 250$.

Nonlinear Program # 4 (NLP4)

$$
\begin{aligned}
&\min f(x) = x^{\mathsf{T}} L^2 x\\
&\text{subject to}\\
&\qquad i - (x_i^2 + x_{2i}^2) \le 0, \qquad i = 1, \dots, (n/2),\\
&\qquad \sqrt{2i} - (x_i + x_{2i}) \le 0, \qquad i = 1, \dots, (n/2),\\
&\qquad \log(x_i + x_{i+1} + x_{i+2}) - x_i + x_{i+1} + x_{i+2} \le 0,\\
&\qquad i = 1, \dots, n-2.
\end{aligned}
$$

Explanation. The matrix L is the discretized tridiagonal Laplacian operator, so the objective function is convex and quadratic. The constraints are nonlinear and the gradients of the active constraints at the solution are linearly dependent. The problem on which we reported results has $n = 1000$.

Truss Problem (Truss)

$$
\begin{aligned}
&\min_x \rho(c^{\mathsf{T}} x)\\
&\text{subject to}\\
&\qquad S(x)^{-1} F - b \le 0,\\
&\qquad X(x) G S(x)^{-1} F - \beta x \le 0.
\end{aligned}
$$

Explanation. This problem chooses the state variables $x \in \mathcal{R}^n$ to minimize the weight of an optimal n-bar truss design, subject to constraints on the deflection and stress of the truss. The function ρ is the density of the material and in our problem was a nonconvex polynomial, $\rho(\zeta) = \zeta^4 - \zeta^2 + 1$. c is a vector containing the lengths of the bars in the truss. The matrix S is the positive definite stiffness matrix, G is a matrix that represents the geometry of the truss and design, and F is the vector of applied forces. The vector b and scalar β form bounds on the maximum allowable deflections in the state variables and the maximum allowable stress in the truss. We solved a problem with $n = 500$ and 1500 constraints.

Maximum Penalized Likelihood Estimate (Stat)

$$
\begin{aligned}
&\min_x f(x(t)) = -\prod_{i=1}^{n} x(t) e^{\phi_\mu(x(t))}\\
&\text{subject to}\\
&\qquad x \in H^2(-\infty, \infty),\\
&\qquad \int_{-\infty}^{\infty} x(t)^2 dt = 1,\\
&\qquad -x(t) \le 0 \quad \text{for all} \;\; t.
\end{aligned}
$$

Explanation. This particular maximum penalized likelihood estimator is sometimes referred to as "the second estimate of Gaskins and Good" (see, e.g., [34] or [35]). We discretize this problem by taking a finite random sample of t_i's, say, $t_i \in [\alpha, \beta]$. $\phi(x)$ is defined by

$$
\phi(x) = \alpha \int_{-\infty}^{\infty} x'(t)^2 dt + \beta \int_{-\infty}^{\infty} x''(t)^2 dt, \tag{A.1}
$$

and given $\mu > 0$, the regularized function $\phi_\mu(x)$ is defined by

$$\phi_\mu(x) = \phi(x)\mu \int_{-\infty}^{\infty} x(t)^2 dt. \tag{A.2}$$

The discrete approximate of $x(t)$ was taken to be a cubic spline. The resulting problem had 500 variables and 1000 constraints.

Boundary Control of Heat Equation (BCHeat)

$$\begin{aligned}
&\min f(x,y) = \int_0^T \left[(x(1,t) - x_d(t))^2 + ay(t)\right] dt \\
&\text{subject to} \\
&\quad C(x(z,t))x_t(z,t) - \nabla(\lambda(x(z,t))\nabla x(z,t)) = f(z,t) \quad \text{on} \;\; \Omega \times [0,T], \\
&\quad \lambda(x(z,t))\nabla x(z,t) = b(z,t) \quad \text{on} \;\; \partial\Omega \times [0,T], \\
&\quad x(z,0) = x_0 \quad \text{on} \;\; \Omega, \\
&\quad x \in L^2(0,T;H^1(\Omega)), \\
&\quad y \in L^2(0,T).
\end{aligned}$$

Explanation. The desired profile is denoted by $x_d(t)$ and Ω is a square in $\mathcal{R}^2$. The inequality constraints are quadratic and linear and arise from enforcing the space conditions $x \in L^2(0,T) \times H^1(\Omega)$ and $y \in L^2(0,T)$. Our discretization results in 500 variables and 1200 constraints. Similar problems have been solved by Newton's method (see [10]), conjugate gradient methods (see [11]), and reduced methods (see [28]).

Molecule Distance Problem (Molec)

$$\begin{aligned}
&\min_{x \in \mathcal{R}^{3d}} \|\Delta - D(X)\|_F \\
&\text{subject to} \\
&\qquad a_{ij} \le D_{ij} \le b_{ij}.
\end{aligned}$$

Explanation. Here $\Delta, D(X), a, b \in \mathcal{R}^{m \times m}$ and $X \in \mathcal{R}^{n \times 3}$, where n is the number of atoms and m is the number of interatomic distances ($2m = n^2 - n$). Δ is a set of observed data, X is a configuration of atoms (their locations in $\mathcal{R}^3$), and D is a transformation into the space of "distance matrices." The bound matrices a, b are upper and lower bounds based on estimating errors in measurements. This problem arises in the processing of NMR data for visualization of large proteins and organic molecules (see, e.g., [23] and [27]). The results in the tables correspond to a problem we solved with 100 variables and 5000 constraints.

REFERENCES

[1] P. T. Boggs, P. D. Domich, and J. E. Rogers, *An interior-point method for general large scale quadratic programming problems*, Ann. Oper. Res., 62 (1996), pp. 419–437.

[2] P. T. Boggs, P. D. Domich, J. E. Rogers, and C. Witzgall, *An interior point method for linear and quadratic programming problems*, Mathematical Programming Society Committee on Algorithms Newsletter, 19 (1991), pp. 32–40.

[3] P. T. Boggs, A. J. Kearsley, and J. W. Tolle, *A Merit Function for Inequality Constrained Nonlinear Programming Problems*, Internal Report 4702, National Institute of Standards and Research, Gaithersburg, MD, 1991.

[4] P. T. Boggs, A. J. Kearsley, and J. W. Tolle, *A global convergence analysis of an algorithm for large scale nonlinear programming problems*, SIAM J. Optim., to appear.

[5] P. T. Boggs and J. W. Tolle, *A family of descent functions for constrained optimization*, SIAM J. Numer. Anal., 21 (1984), pp. 1146–1161.

[6] P. T. Boggs and J. W. Tolle, *A strategy for global convergence in a sequential quadratic programming algorithm*, SIAM J. Numer. Anal., 26 (1989), pp. 600–623.

[7] P. T. Boggs and J. W. Tolle, *Sequential Quadratic Programming*, in Acta Numerica, 1995, Cambridge University Press, Cambridge, UK, 1995, pp. 1–51.

[8] P. T. Boggs, J. W. Tolle, and A. J. Kearsley, *A truncated SQP algorithm for large scale nonlinear programming problems*, in Advances in Optimization and Numerical Analysis: Proceedings of the Sixth Conference on Numerical Analysis and Optimization, S. Gomez and J.-P. Hennart, eds., Kluwer, Norwell, MA, 1994, pp. 69–78.

[9] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. T. Toint, *CUTE: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[10] J. Burger and M. Pogu, *Functional and numerical solution of a control problem originating from heat transfer*, J. Optim. Theory Appl., 68 (1991), pp. 49–73.

[11] C. Carthel, R. Glowinski, and J. L. Lions, *On exact and approximate boundary controllabilities for the heat equation. A numerical approach*, J. Optim. Theory Appl., 82 (1994), pp. 429–484.

[12] T. F. Coleman, *Large scale numerical optimization: Introduction and overview*, in Encyclopedia of Computer Science and Technology, Marcel Dekker, New York, 1992.

[13] A. R. Conn, N. I. M. Gould, and P. T. Toint, *Lancelot: A Fortran Package for Large-Scale Nonlinear Optimization*, Ser. Comput. Math. 17, Springer-Verlag, Heidelberg, New York, 1992.

[14] A. R. Conn, N. I. M. Gould, and P. T. Toint, *Large-scale nonlinear constrained optimization*, in Proceedings of the Second International Conference on Industrial and Applied Mathematics, SIAM, Philadelphia, 1992, pp. 51–70.

[15] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[16] J. Dennis, Jr., M. El-Alem, and M. C. Maciel, *A global convergence theory for general trust-region-based algorithms for equality constrained optimization*, SIAM J. Optim., 7 (1997), pp. 177–207.

[17] J. E. Dennis, Jr., and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[18] S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422.

[19] M. El-Alem, *A robust trust-region algorithm with a nonmonotonic penalty parameter scheme for constrained optimization*, SIAM J. Optim., 5 (1995), pp. 348–378.

[20] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, New York, 1981.

[21] P. E. Gill, M. A. Saunders, W. Murray, and M. H. Wright, *Constrained nonlinear programming*, in Optimization, G. L. Nemhauser, A. H. G. R. Kan, and M. J. Todd, eds., North-Holland, Amsterdam, 1989, pp. 171–210.

[22] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, Berlin, 1984.

[23] W. Glunt, T. L. Hayden, and M. Raydan, *Molecular conformations from distance matrices*, J. Comput. Chem., 14 (1993), pp. 114–120.

[24] M. Gockenbach and A. J. Kearsley, *Optimal signal sets for non-Gaussian detectors*, SIAM J. Optim., 9 (1999), pp. 316–326.

[25] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Econom. and Math. Systems 187, Springer-Verlag, Berlin, 1981.

[26] A. J. Kearsley, *The Use of Optimization Techniques in the Solution of Partial Differential Equations from Science and Engineering*, Ph.D. thesis, Rice University, Houston, TX, 1996.

[27] A. J. Kearsley, R. A. Tapia, and M. Trosset, *The solution of the metric stress and stress problems in multidimensional scaling using Newton's method*, Comput. Statist., 13 (1998), pp. 369–396.

[28] F.-S. Kupfer and E. W. Sachs, *Numerical solution of a nonlinear parabolic control problem by a reduced SQP method*, Comput. Optim. Appl., 1 (1992), pp. 113–135.

[29] J. L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations*, Springer-Verlag, Berlin, 1971.

[30] J. L. Lions, *Exact controllability, stabilization and perturbations for distributed systems*, SIAM

Rev., 30 (1988), pp. 1–68.

[31] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[32] K. Schittkowski, *More Test Examples for Nonlinear Programming Codes*, Lecture Notes in Econom. and Math. Systems 282, Springer-Verlag, Berlin, 1987.

[33] R. A. Tapia, *On the role of slack variables in quasi-Newton methods for constrained optimization*, in Numerical Optimization of Dynamical Systems, L. C. W. Dixon and G. P. Szegö, eds., North-Holland, Amsterdam, 1980, pp. 235–246.

[34] R. A. Tapia and J. R. Thompson, *Nonparametric Probability Density Estimation*, The Johns Hopkins University Press, Baltimore, MD, 1978.

[35] J. R. Thompson and R. A. Tapia, *Nonparametric Function Estimation, Modeling, and Simulation*, SIAM, Philadelphia, 1990.

SIAM J. OPTIM.
Vol. 9, No. 3, pp. 779–802

SIMULATED ANNEALING: SEARCHING FOR AN OPTIMAL TEMPERATURE SCHEDULE*

HARRY COHN† AND MARK FIELDING†

Abstract. A sizable part of the theoretical literature on simulated annealing deals with a property called convergence, which asserts that the simulated annealing chain is in the set of global minimum states of the objective function with probability tending to 1. However, in practice, the convergent algorithms are considered too slow, whereas a number of nonconvergent ones are usually preferred. We attempt a detailed analysis of various temperature schedules. Examples will be given of when it is both practically and theoretically justified to use boiling, fixed temperature, or even fast cooling schedules which have a small probability of reaching global minima. Applications to traveling salesman problems of various sizes are also given.

Key words. simulated annealing, temperature, cooling, Markov chain, convergence, inhomogeneous chain, fundamental matrix, time to absorption

AMS subject classifications. 60J05, 60K35

PII. S1052623497329683

1. Introduction and summary.

Suppose that a function f is defined on a finite (but large) set of states S. The aim of simulated annealing (SA) is to find a state x such that $f(x) = \min_{y\in S} f(y)$. Because, for some large S, such an aim is not in general feasible in a reasonable time frame, we may confine ourselves to finding a near optimal x, i.e., a state x for which $f(x)$ is close to $\min_{y\in S} f(y)$.

For each state x in S, define a set $N(x)$, called the set of neighbors of x. Write $\mathcal{N}$ for the family of neighborhoods $\{N(x),\ x \in S\}$.

A neighbor choosing matrix $\boldsymbol{G}$ with entries $G(x, y)$ is defined for each x and y in S such that $G(x, y) > 0$ if and only if $y \in N(x)$. The matrix $\boldsymbol{G}$ is called a *generation* matrix.

Define

$$P_T(x,y) = \begin{cases} G(x,y)\exp(-[f(y)-f(x)]^+/T) & \text{if } y \neq x, \\ 1-\sum_{z\neq x} P_T(x,z) & \text{if } y = x, \end{cases}$$

with $a^+ = \max(a, 0)$.

The parameter T is called *temperature*. Write $\boldsymbol{P_n}$ for the transition probability corresponding to $T = T_n$, where T_n is the temperature at time n. The sequence $\{T_n\}$ is called a *temperature schedule*. If $\lim_{n\to\infty} T_n = 0$ we say that $\{T_n\}$ is a *cooling schedule*; we say it is a *fixed temperature schedule* if $T_n = T$ for all n.

An initial probability distribution and the sequence of one-step transition probabilities $\{\boldsymbol{P_n}\}$ define an inhomogeneous Markov chain $\{X_n\}$. This chain will be called an *SA chain*. The SA chain is the basis of the SA algorithm. It originates from an idea that goes back to the paper by Metropolis et al. [24] and was followed up by many other contributors (see, e.g., Aarts and van Laarhoven [2], Aarts and Korst [3], Chiang and Chow [5], Connolly [9], Connors and Kumar [10], Gelfand and Mitter [11],

*Received by the editors October 30, 1997; accepted for publication (in revised form) July 1, 1998; published electronically June 30, 1999.
http://www.siam.org/journals/siopt/9-3/32968.html

†Department of Mathematics and Statistics, University of Melbourne, Parkville 3052, Victoria, Australia (harry@ms.unimelb.edu.au).

Geman and Geman [12], Hajek [15], Hwang and Sheu [16], Romeo and Sangiovanni-Vincentelli [28]). Recently, Niemiro and Pokarowski [26] and Niemiro [27] have clarified the asymptotic behavior of the SA chain by relating it to the theory of the tail events (see Cohn [6], [7], [8]).

Write $P^{(m,n)}(x,y) = P(X_n = y|X_m = x)$ for $m < n$. We shall say that y *is reachable from* x if there exist an integer p and states $x = x_0, x_1, x_2, \ldots, x_p = y$ such that $x_{k+1} \in N(x_k)$ for $0 \leq k < p$. It is easy to see that if y is reachable from x then there must be a number p such that $P^{(m,m+p)}(x,y) > 0$ for any m.

We assume that $(S, \mathcal{N})$ is irreducible, i.e., that any state x is reachable from any state y.

We shall say that state y is *reachable at height* h from state x if h is the smallest number such that $x = y$ and $f(x) \leq h$ or if there is a sequence of states $x = x_0, x_1, \ldots, x_p = y$ for some $p \geq 1$ such that $x_{k+1} \in N(x_k)$ for $0 \leq k < p$ and $f(x_k) \leq h$ for $0 \leq k \leq p$.

State x is said to be a *local minimum* if no state y with $f(y) < f(x)$ is reachable from x at height $f(x)$. The *depth* of x, $d(x)$, is defined to be ∞ if x is a global minimum; otherwise it is the smallest number h, $h > 0$, such that some y with $f(y) < f(x)$ can be reached from x at height $f(x) + h$.

We assume that y is reachable from x at height h if and only if x is reachable from y at height h. This assumption is called *weak reversibility.*

Write S^* for the global minimum set of states, i.e., the set of states x with $f(x) = \min_{y \in S} f(y)$. We say that the SA chain (or algorithm) is *convergent* if

$$\lim_{n\to\infty} P(X_n \in S^*) = 1. \tag{1.1}$$

Hajek [15] identified the smallest value of c for which an SA chain with cooling schedule of the form

$$T_n = \frac{c}{\log(n + n_0)}$$

is convergent. (Here n_0 is a positive integer.) It was proved in [15] that the SA algorithm is convergent if and only if $c \geq d^*$, where d^* is the largest depth of the local minima, which are not global minima.

Thus $T_n = d^*/\log(n + n_0)$ gives the fastest logarithmic-type cooling schedule leading to convergence. Such a cooling schedule is called *canonical*, and d^* is said to be the *canonical constant.* It is important to stress that it would be wrong to assume that a canonical cooling schedule necessarily reaches global minimum faster than other schedules.

A convergent chain obtains optimality in the long run even if we adopt a *memoryless algorithm*, i.e., an algorithm that does not recall the past values of the chain. An algorithm that stores the best solution of all iterations will be called a *memory algorithm.*

The aim of this paper is to study the behavior of the SA algorithm in terms of temperature schedules. It turns out that the key critical points for the limit behavior of the SA chain occur in the range of logarithmic cooling schedules. We shall describe a number of optimality criteria corresponding to various situations. Then we study some theoretical properties of algorithms that are used in practice. It turns out that there is no theoretical reason why some temperature schedules that are attached to nonconvergent SA chains should be overruled. Examples are given to illustrate each case.

2. Which optimality? An algorithm needs to specify a stopping rule (time), i.e., a time when the process is terminated and a decision is adopted. This stopping rule, denoted by τ, may or may not depend on the values taken by the chain up to the stopping time and may be random. It is also a function of the temperature schedule and other parameters of the algorithm as well as the optimality criterion adopted for the problem.

It is usually assumed that an optimal (near optimal) algorithm is one that

(i) with probability 1 reaches the global (near global) minimum in finite time, and

(ii) is faster than other algorithms.

Both desiderata need to be qualified. It will turn out that (i) may have a different meaning for memoryless algorithms than for those with memory. Besides, we shall further see that (i) may be done away with if we consider multiple run algorithms. As far as (ii) is concerned, there are a number of principles for optimality based on τ, the most common one being searching for the algorithm that attains the minimum of $E(\tau)$. However, such a criterion would exclude algorithms with $P(\tau = \infty) > 0$. The choice of the algorithm and the corresponding optimality criterion may be problem dependent. A discussion of various criteria of optimality follows.

2.1. Convergent algorithms. A convergent chain, defined by (1.1), ensures that the global minimum is eventually reached if a sufficiently large number of iterations is allowed. If convergence fails for a cooling schedule $\{T_n\}$, there is a positive probability that the SA chain will never reach a global minimum state (see Hajek [15]). Some authors seem to assume that convergence is a necessary property of a successful algorithm. As previously mentioned, the canonical schedule, despite being the fastest tending to 0, is not necessarily optimal, i.e., the fastest in reaching a global minimum.

In fact, convergence is not a necessary attribute of a successful algorithm either. It may be necessary in relation to a memoryless algorithm. For memory algorithms, there is no a priori reason for a cooling schedule with property (1.1) to be preferred to temperature schedules that do not satisfy (1.1).

By the same token, one should not a priori eliminate convergent algorithms from the search of optimal schedules.

2.2. Regular algorithms. For memory algorithms, (2.1) is replaced by the less restrictive requirement that S^* be reached with probability 1. In such a case, a criterion for optimality should depend only on how early S^* is reached.

Suppose that, for any given temperature schedule $\{T_n\}$, we define the stopping time τ to be the first n such that X_n hits S^*. An algorithm for which

$$P(\tau < \infty) = 1 \tag{2.1}$$

is said to be *regular* and *defective* otherwise.

For memory algorithms, τ is the variable that should be optimized. The relevant sequence of random variables is $\{\min(f(X_1), \ldots, f(X_n))\}$, and (2.1) is equivalent to

$$\lim_{n\to\infty} P(\min(f(X_1), \ldots, f(X_n)) = \min_{x\in S} f(x)) = 1. \tag{2.2}$$

Obviously (2.2), or equivalently (2.1), is a convergence property, and it is easy to see that it holds for a much larger class of temperature schedules than the ones satisfying (1.1). For example, all the chains corresponding to fixed temperature schedules satisfy it, as they are ergodic Markov chains with stationary transition probabilities. It is well known that for such chains all states are recurrent and (2.1) holds. On the other hand, if $\{\pi_x\}$ is the stationary probability distribution, then $\lim_{n\to\infty} P(X_n = x) = \pi_x > 0$ for all x and therefore

$$\lim_{n\to\infty} P(X_n \in S^*) = 1 - \sum_{x \notin S^*} \pi_x < 1,$$

so that (1.1) fails.

We shall see later an example where it may be optimal to boil, i.e., to let T_n tend to ∞. This case corresponds to a Markov chain with stationary transition probabilities given by the generation matrix.

We shall show that there are problems where a fixed optimal temperature may be identified.

For regular algorithms, a criterion for τ^* to be optimal is

$$E(\tau^*) = \min_{\tau \in \mathcal{T}} E(\tau),$$

where $\mathcal{T}$ is the class of stopping times attached to all temperature schedules. This criterion is often used in operations research.

A number of papers have pointed out the potential usefulness of memory algorithms (see, e.g., Kirkpatrick [20], Gelfand and Mitter [11] for cooling schedules and Connolly [9] for a fixed temperature algorithm).

We shall study the properties of τ for fixed temperature schedules in a later section.

2.3. Defective algorithms. It may seem natural to consider optimality with respect to the class of all temperature schedules defining a regular algorithm. However, a closer examination does not justify such a criterion. We also need to consider temperature schedules that may correspond to defective algorithms, even if $P(\tau < \infty)$ is not even close to 1. In fact, such algorithms are the ones mostly used in practice.

Suppose that we want to allow for a fixed number of iterations N and choose the algorithm that performs the best within N iterations.

Consider the case when the numbers N and p are suitably chosen such that, for a stopping time τ corresponding to some temperature schedule,

$$P(\tau \leq \mathrm{N}) \geq p.$$

Define $\mathcal{T}$ to be the class of all regular and defective τ, where τ is the first hitting time of S^* (or near optimal states). An optimality criterion for such a case will be satisfied by a stopping time τ^* in $\mathcal{T}$ such that

$$P(\tau^* \leq \mathrm{N}) = \sup_{\tau \in \mathcal{T}} P(\tau \leq \mathrm{N}).$$

In fact, we may achieve a property close to (2.2) in terms of some number, say k, of independent runs of size N. Indeed, if $\{X_n^{(i)},\ n = 1, \ldots, \mathrm{N}\}$ is the ith run with $i = 1, \ldots, k$, then

$$P\left(\min_{i \in \{1,\ldots,k\}} \min\left(f\left(X_1^{(i)}\right), \ldots, f\left(X_{\mathrm{N}}^{(i)}\right)\right) = \min_{x \in S} f(x)\right) \geq 1 - (1-p)^k. \tag{2.3}$$

By suitably choosing k such that the right-hand side of (2.3) is as large as desired, and adopting the stopping rule $\tau_{\mathrm{N}} = \min(\tau, \mathrm{N})$, we may ensure both the quality of the algorithm and a limitation on the number of iterations.

2.4. Near optimality. If optimality requires too many iterations, near optimality may be a suitable alternative. The latter is in fact the case with most algorithms used in practice. In fact, many feasible algorithms will only provide a near optimal solution.

In some cases, getting a near minimum, say, within two percent of the global minimum, can be achieved with a drastic reduction in the number of iterations required for finding a global minimum state. An improvement from two percent to, say, one percent may result in a huge increase in the number of iterations, which is not always practical.

3. Fixed temperature schedules.

If a simulated annealing chain is run with a fixed temperature, then, under minor conditions on the generation matrix, the "best state so far" will, with probability 1, ultimately become a global minimum, and the expected time and variance of the time until reaching a global minimum will be finite. This follows from the classical theory of finite Markov chains.

We will see that for some small and medium-size problems, the fixed temperature schedules seem to work better than the simulated algorithms based on a cooling schedule.

Define a Markov chain $\{X_n^*\}$ with one absorbing state representing the states of S^* lumped together. The states outside S^*, as well as the transition probabilities among themselves, remain unchanged. Clearly, the first time the chain $\{X_n^*\}$ reaches S^* is a stopping time, say, τ. Such a case is well known in the theory of Markov chains (see, e.g., Kemeny and Snell [18, Theorem 3.5.3]). Denote by $\boldsymbol{Q}$ the transition matrix corresponding to the states outside S^*. The matrix $\boldsymbol{N} = (\boldsymbol{I} - \boldsymbol{Q})^{-1}$ is called *fundamental*. The square matrix $\boldsymbol{I}$, called identity matrix, has the diagonal entries equal to 1 and is 0 elsewhere. Let $\boldsymbol{A}$ be an arbitrary finite matrix. The matrix $\boldsymbol{A_{sq}}$ is formed from $\boldsymbol{A}$ by squaring all entries. We denote by $\boldsymbol{\xi}$ a column matrix having all components equal to 1.

The following result is extracted from Theorem 3.5.4 of [18].

THEOREM 3.1. (i) *The ith component of $\boldsymbol{N\xi}$ is the mean number of steps needed to reach S^* given that the chain starts in i.*

(ii) *The ith component of $(\boldsymbol{2N} - \boldsymbol{I})\boldsymbol{N\xi} - (\boldsymbol{N\xi})_{sq}$ is the variance of the same function.*

3.1. An optimal boiling schedule example.

In Hajek [15], a small problem instance, shown here as Figure 3.1, is given for SA consisting of 26 states. The chain is used by Hajek to illustrate convergent schedules. Ironically, it turns out that boiling to ∞ is the optimal temperature schedule.

Shown is the neighborhood structure as well as the cost associated with each state. The states have been numbered arbitrarily to give the state space $S = \{1, 2, \ldots, 26\}$. The relationship $y \in N(x)$ is represented by an arrow from x to y. So the set of neighbors of state 9, for example, is $N(9) = \{8, 10, 13\}$, and for state 3, we get $N(3) = \{2\}$.

There are six local minima, states 1, 2, 10, 12, 17, and 26, and the set of global minima is $S^* = \{1, 2, 26\}$. It is easy to check that this chain is weakly reversible.

We shall assume that the generation matrix is given by $G(x, y) = 1/|N(x)|$ for all $x \in S$ and $y \in N(y)$.

We note that this example is only trivial in size. It does, however, allow us to examine the application of Markov chain theory to SA in a way not plausible for practical problems. That is the explicit examination of the transition matrix. It may also raise interesting questions about the behavior of SA in real-life problems.

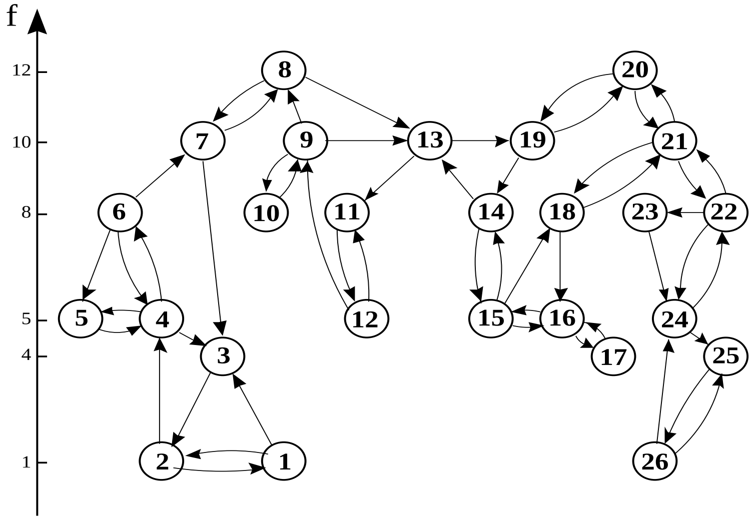


FIG. 3.1. *A* 26*-state example given in Hajek* [15].

TABLE 3.1

Performance of fixed temperature schedules for a 26*-state SA chain. Mean and standard deviation of time to hitting a global minimum are given.*

T	E(τ)	SD(τ)
0	∞	∞
1	2964.04	2949.57
2	250.77	252.25
3	129.00	126.93
4	97.87	94.48
5	84.73	80.73
10	67.02	62.16
50	58.67	53.48
100	57.91	52.70
∞	57.20	51.97

TABLE 3.2

Performance of logarithmic cooling schedules for a 26*-state SA chain. Estimates of mean and standard deviation of time until hitting a global minimum are given. The values at* $c = \infty$ *follow from the calculations with fixed temperature.*

c	E(time)	SD(time)
6	∞	∞
8	1026.74	3464.46
10	259.78	499.17
50	64.45	61.70
80	60.39	56.06
100	60.44	55.74
200	58.72	53.84
500	57.81	53.58
1000	57.78	52.50
∞	57.20	51.97

To get from the local minimum state 12 to a state of lower cost, it is necessary to climb at least five units, so the depth of state 12 is equal to 5. Similarly, the depth of state 10 is 2, the depth of state 17 is 6, and the depth of the globally minimal states is defined as being infinite. The cups associated with the local minima that are not global minima are $\{10\}$, $\{11, 12\}$, and $\{14, 15, 16, 17, 18\}$.

The generation matrix for Hajek's 26-state example is irreducible, and as a result, the homogeneous SA Markov chain is also irreducible. Thus all states are recurrent. For Hajek's example, we investigate how the value of the temperature influences the time it takes a homogeneous SA chain to reach a global minimum. To this end we consider the Markov chain formed by making all global minima absorbing states. For $x = 1, 2, 26$, we set $G(x, y) = 1$ for $y = x$ and 0 otherwise. Given $\boldsymbol{Q}$, we can go on to calculate the fundamental matrix $\boldsymbol{N} = (\boldsymbol{I} - \boldsymbol{Q})^{-1}$. This can then be used to calculate the mean and variance of the time it takes a homogeneous SA Markov chain to reach a global minimum given by Theorem 3.1.

Performing these calculations for Hajek's example, we find, if we start the SA chain in, say, state 13, that the mean time until absorption is equal to

$$\begin{aligned}&(8 + 64\,\delta + 22\,\delta^2 + 190\,\delta^3 + 85\,\delta^4 + 243\,\delta^5 + 318\,\delta^6 + 180\,\delta^7\\ &+ 600\,\delta^8 + 107\,\delta^9 + 632\,\delta^{10} + 101\,\delta^{11} + 391\,\delta^{12} + 127\,\delta^{13}\\ &+ 135\,\delta^{14} + 118\,\delta^{15} + 24\,\delta^{16} + 65\,\delta^{17} + 2\,\delta^{18} + 18\,\delta^{19} + 2\,\delta^{21}\,)\\ &/\,(\,\delta^6\,(\,4 + 16\,\delta^2 + 21\,\delta^4 + 13\,\delta^6 + \delta^7 + 3\,\delta^8 + \ \delta^9 + \ \delta^{11}\,)\,)\end{aligned}$$

with $\delta = \exp(-1/T)$, and the variance is equal to

$$\begin{aligned}&(64 + 896\,\delta + 4288\,\delta^2 + 5888\,\delta^3 + 26380\,\delta^4 + 25352\,\delta^5 + 82400\,\delta^6\\ &+ 90128\,\delta^7 + 168711\,\delta^8 + 236270\,\delta^9 + 267229\,\delta^{10} + 444772\,\delta^{11}\\ &+ 387603\,\delta^{12} + 616478\,\delta^{13} + 557847\,\delta^{14} + 658450\,\delta^{15} + 747564\,\delta^{16}\\ &+ 582882\,\delta^{17} + 849119\,\delta^{18} + 481773\,\delta^{19} + 779677\,\delta^{20} + 418840\,\delta^{21}\\ &+ 571970\,\delta^{22} + 376007\,\delta^{23} + 340216\,\delta^{24} + 305693\,\delta^{25} + 174206\,\delta^{26}\\ &+ 202119\,\delta^{27} + 86565\,\delta^{28} + 102605\,\delta^{29} + 45346\,\delta^{30} + 38468\,\delta^{31}\\ &+ 23053\,\delta^{32} + 10177\,\delta^{33} + 9792\,\delta^{34} + 1782\,\delta^{35} + 3092\,\delta^{36}\\ &+ 184\,\delta^{37} + 654\,\delta^{38} + 8\,\delta^{39} + 80\,\delta^{40} + 4\,\delta^{42}\,)\\ &/\ \ (\,\delta^{12}(\,4 + 16\,\delta^2 + 21\,\delta^4 + 13\,\delta^6 + \delta^7 + 3\,\delta^8 + \delta^9\,\delta^{11}\,)^2\,).\end{aligned}$$

We see from Table 3.1 that, for Hajek's example, SA with a fixed temperature will find a global minimum more quickly, on average, for larger temperatures. The optimum strategy based on $E(\tau)$ is to take $T = \infty$, i.e., to adopt the "boiling" schedule which corresponds to a Markov chain with transition matrix given by the generation matrix. This strategy is the one that accepts all moves with probability 1.

Shown in Table 3.2 are the results from simulations performed for cooling schedules of a logarithmic type, including the canonical cooling schedule. Ten thousand runs were performed at each value of c. Again, it is apparent that the optimal strategy is to adopt boiling.

4. A state classification and eventual traps. As in the homogeneous case, the states of an inhomogeneous Markov chain may be classified as positive, null, recurrent, or transient. However, some of the definitions used for the homogeneous chains do not seem to carry over, whereas other definitions for inhomogenous chains which reduce to the classical ones are available. A state classification for finite and countable inhomogeneous chains is given in [8]. We shall adapt it here to the particular case of a SA chain. Also, the atomic sets of the tail σ-field (see [8]) admit in this case a neat representation in terms of some sets, which we shall call *eventual traps.*

A state x will be said to be *null* if $\lim_{n\to\infty} P(X_n = x) = 0$ and *positive* if $\lim_{n\to\infty} P(X_n = x) > 0$. Such a classification is not a dichotomy for inhomogeneous chains, but in the case of an SA chain, it is (see [26]). Let $\{A_n\}$ be a sequence of events. Write $\{A_n \text{ i.o.}\} = \cap_{n=1}^{\infty} \cup_{m=n}^{\infty} A_m$, where i.o. stands for *infinitely often* and $\{A_n \text{ ult.}\} = \cup_{n=1}^{\infty} \cap_{m=n}^{\infty} A_m$, where ult. stands for *ultimately.* We say that $\lim_{n\to\infty} A_n = A$ almost surely (a.s.) if $P(\{A_n \text{ ult.}\}) = P(\{A_n \text{ i.o.}\})$ and A is an event differing from $\{A_n \text{ ult.}\}$ only by a set of probability 0. We say that a state x is *recurrent* if

$$P(X_n = x \text{ i.o.}) > 0,$$

and *transient* otherwise. A positive state x is always recurrent and is called *positive recurrent.* Null states may be transient or recurrent. A null state which is recurrent will be called *null recurrent.* These definitions were given in [8].

We say that A is a recurrent class if it contains only recurrent states and for any $x \in A$

$$\{X_n = x \text{ i.o.}\} = \{X_n \in A \text{ i.o.}\} \text{ a.s.}$$

We say that the recurrent class A is an *eventual trap* if

(i) $\lim_{n\to\infty} P(X_n \in A) > 0$ and

(ii) $P(X_n = x \text{ i.o.}) = P(X_n \in A \text{ i.o.}) = P(X_n \in A \text{ ult.})$ for any $x \in A$.

We use the term eventual trap as distinct from trap to emphasize that a Markov chain reaching A may have a positive probability of escaping from A at all times, but as $n \to \infty$ such an escape becomes less and less likely and the chain must end up in an eventual trap with probability 1.

Remark. For an SA chain, it turns out that if A is an eventual trap with $\lim_{n\to\infty} P(X_n \in A) < 1$, then A^c, the complementary set to A, must contain at least one eventual trap.

A result of one of the present authors (see [8] and the references therein) describing the tail σ-field of a finite inhomogeneous Markov chain leads to the assertion that a chain of SA type has a finite number of disjoint eventual traps $A_1, \dots, A_t$ such that

$$\lim_{n\to\infty} P(X_n \in \cup_{i=1}^{t} A_i) = 1.$$

Obviously, the number of eventual traps does not exceed the cardinality of S.

5. Weak and strong ergodicity: Conditional convergence. Write $P^{(m,n)}(x, y) = P(X_n = y | X_m = x)$ for $m < n$. We shall say that $\{X_n\}$ is *weakly ergodic* if for any m, x, y, and z,

$$\lim_{n\to\infty} (P^{(m,n)}(x, z) - P^{(m,n)}(y, z)) = 0.$$

A sufficient condition for weak ergodicity is the existence of some constant u such that

$$\sum_{k=1}^{\infty} \min_{x,y} P^{(k,k+u)}(x, y) = \infty. \tag{5.1}$$

However, it is easy to see that (5.1) requires that for all x

$$\sum_{n=1}^{\infty} P(X_n = x) = \infty.$$

In general, the above property is not necessary for weak ergodicity.

We say that $\{X_n\}$ is *strongly ergodic* if there is a probability distribution $\pi = (\pi_1, \ldots, \pi_s)$ on S such that for any x, y, and $m \geq 1$,

$$\lim_{n\to\infty} P^{(m,n)}(x, y) = \pi_y. \tag{5.2}$$

It is easy to see that strong ergodicity implies weak ergodicity. For properties of weakly and strongly ergodic chains, see Seneta [29].

We say that $\{X_n\}$ is *conditionally convergent* if for some numbers $\pi_{x,y}^{(m)}$ and any $m \geq 1$,

$$\lim_{n\to\infty} P^{(m,n)}(x, y) = \pi_{x,y}^{(m)}. \tag{5.3}$$

In the literature of inhomogeneous Markov chains, such chains are known as *convergent* (see Mukherjea [25] and Cohn [7]), but that term has been used before in relation to property (1.1), so *conditional convergence* will be used for (5.3). For weakly ergodic chains, conditional convergence is equivalent to strong convergence, as defined in (5.2).

6. Slow cooling schedules. Let us write $\underline{f}= \min_{\{x\in S\}} f(x)$ and denote by $\bar{d}$ the number with the property

$$\sum_{k=1}^{\infty} \exp(-\bar{d}/T_k) = \infty$$

and

$$\sum_{k=1}^{\infty} \exp(-d/T_k) < \infty$$

for any $d > \bar{d}$.

If $\bar{d} > 0$ we shall say that $\{T_n\}$ is a slow cooling schedule. If $\bar{d} = 0$ we say that $\{T_n\}$ is a fast cooling schedule.

The following result is extracted from Niemiro and Pokarowski [26] and Niemiro [27].

THEOREM 6.1. *Suppose that $\{X_n\}$ is an SA chain with $\bar{d} > 0$. Then*

(i) *there exist t recurrent classes $A_1, \ldots, A_t$ which are eventual traps;*

(ii) *the chain is conditionally convergent;*

(iii) *if $x \in A_i$ then $y \in A_i$ if and only if y is reachable from x at height lower than or equal to $\underline{f}(A_i) + \bar{d}$, where $\underline{f}(A_i) = \min_{z\in A_i} f(z)$;*

(iv) *if $\bar{S}$ is the set consisting of deepest states of $A_1, \ldots, A_t$, then* $\lim_{n\to\infty} P(X_n \in \bar{S}) = 1$*;*

(v) *if $x \in \bar{S}$, then* $\lim_{n\to\infty} P(X_n = x) > 0$.

Notice in particular that a convergent chain may admit either only one eventual trap (the case of a weakly ergodic chain) or several eventual traps, each of them containing some global minimum states on which the whole probability mass will eventually concentrate.

On the other hand, it is easy to see that by decreasing $\bar{d}_1$ to, say, $\bar{d}_2$, the number of eventual traps does not decrease, because any eventual trap for $\bar{d}_2$ is either an eventual trap for $\bar{d}_1$ or belongs to a partition of an eventual trap for $\bar{d}_1$. Thus, if $\bar{d}$ is smaller than d^*, the SA algorithm does not converge because the set of eventual traps will necessarily include some that do not contain any global minima states. Such eventual traps attract the chain to local minima. Since each of the limit probabilities $\lim_{n\to\infty} P(X_n \in A_i)$ is positive, property (2.1) also fails for chains of this kind. Indeed, notice that $\lim_{n\to\infty} P(X_n \in S^*) = 1 - \sum_{i\in\Lambda_{\bar{d}}} \lim_{n\to\infty} P(X_n \in A_i)$, where $\{A_i : i \in \Lambda_{\bar{d}}\}$ is the collection of eventual traps corresponding to $\bar{d}$ which do not contain global minima states. Clearly, $\lim_{n\to\infty} P(X_n \in S^*)$ becomes smaller as $\bar{d}$ decreases and, as a result, the number of eventual traps increases. If $\bar{d}$ is sufficiently small, then any local minima states may form the bottom of some eventual trap. This is the reason why some heuristics cooling faster than logarithmic are not convergent. Such algorithms may end up in a local minimum. We shall describe a number of such algorithms later in the paper.

Theorem 6.2. *A convergent chain is weakly ergodic if and only if one of the following two statements holds:*

(i) *There is only one global minimum state.*

(ii) *If x and y are two global minima states, then x is reachable from y at height smaller than or equal to $\underline{f} + \bar{d}$.*

Proof. It is easy to see that if (i) holds, the only global minima state, say x, is in a recurrent class A which is an eventual trap. However, A is the only eventual trap since $\lim_{n\to\infty} P(X_n \in A) \geq \lim_{n\to\infty} P(X_n = x) = 1$. Thus $\{X_n\}$ has a trivial tail σ-field, which implies weak ergodicity (see [6]).

To prove (ii) notice that by Theorem 6.1 (iii) all the global minima states must be in one recurrent class which is the unique eventual trap. This completes the proof.

Corollary 6.3. *A convergent chain is not weakly ergodic if and only if there exist two global minima states x and y such that y is reachable from x at height higher than $\underline{f} + \bar{d}$.*

Theorem 6.4. *A weakly ergodic SA chain corresponding to a cooling schedule is convergent and strongly ergodic.*

Proof. Any weakly ergodic chain has a trivial tail σ-field and therefore could not admit more than one eventual trap. However, the only cooling schedules that are not convergent are the ones that admit several eventual traps, with at least one having no global minimum states. This proves convergence. Strong ergodicity follows from Theorem 6.1 and weak ergodicity.

Remark. If we do not confine ourselves to cooling schedules, then weak ergodicity may not imply convergence, as we have seen in the case of fixed temperature schedules.

To summarize the above results on convergence, we conclude that

(i) the canonical cooling schedule may result in a chain that is not weakly ergodic;

(ii) the canonical constant d^* is the cutoff point for $\bar{d}$ below which the process exhibits a *phase transition*, with its class of eventual traps increasing to include some local minima traps.

Lemma 6.5. *Suppose that $\sum_{n=1}^{\infty} P_n(x,y) = \infty$, where $\liminf_{n\to\infty} P(X_n = x) > 0$. Then $P(X_n = x, X_{n+1} = y \text{ i.o.}) > 0$.*

Proof. Write $A_n = \{X_n = x, X_{n+1} = y\}$. We shall show that a divergent part of the Borel–Cantelli-type lemma holds for the events $\{A_n\}$. Write $\mathcal{F}_n$ for the σ-field generated by $X_1, \ldots, X_n$. The Markov property of $\{X_n\}$ yields

$$P(A_n|\mathcal{F}_n) = P_n(x,y)1_{\{X_n=x\}}. \tag{6.1}$$

According to the Borel–Cantelli–Levy lemma,

$$P(A_n \text{ i.o.}) > 0 \text{ if and only if } P\left(\sum_{n=1}^{\infty} P(A_n|\mathcal{F}_n) = \infty\right) > 0. \tag{6.2}$$

Consider now the random variable

$$Y_n = \frac{\sum_{k=1}^{n} P_k(x,y)1_{\{X_k=x\}}}{\sum_{k=1}^{n} P_k(x,y)}. \tag{6.3}$$

Notice that the denominator in (6.3) tends to ∞ as $n \to \infty$. Since $0 \le Y_n \le 1$, for $\{Y_n\}$ to converge in probability to 0, it is necessary that $E(Y_n) \to 0$. However, this is not the case as $\liminf_{n\to\infty} E(Y_n) \ge \liminf_{n\to\infty} P(X_n = x) > 0$, which implies $P(\sum_{k=1}^{\infty} P_k(x,y)1_{\{X_k=x\}} = \infty) > 0$; the proof is concluded on account of (6.1) and (6.2).

This lemma provides a criterion of recurrence for a state y which is reachable in one step from a positively recurrent state.

THEOREM 6.6. *If $\{X_n\}$ is convergent, then*

(i) *$\{f(X_n)\}$ converges in probability to $\underline{f}$;*

(ii) *$\{f(X_n)\}$ converges a.s. to $\underline{f}$ if and only if*

$$\sum_{n=1}^{\infty} \exp\left(-\frac{d(x)}{T_n}\right) < \infty \tag{6.4}$$

for any state x with $x \notin S^*$ and $x \in N(y)$, where y is a global minimum state.

Proof. Since f is constant on S^*, (i) follows from the definition of convergence (1.1).

To prove (ii), notice that (6.4) implies

$$\sum_{n=1}^{\infty} P(\{X_n \in S^*\} \cap \{X_{n+1} \notin S^*\}) < \infty.$$

By a Borel–Cantelli-type lemma given by Barndorff-Nielsen [4], the above implies that $P(\{X_n \in S^* \text{ ult.}\}) = 1$. Thus all states outside S^* are transient, which proves the first implication of (ii).

Assume now that (6.4) fails. Thus there exists a state x with $f(x) > \underline{f}$, $x \in N(y)$, where y is a global minimum state, and

$$\sum_{n=1}^{\infty} \exp\left(-\frac{d(x)}{T_n}\right) = \infty.$$

According to Theorem 6.1, all states of S^* are positive. Thus we can use Lemma 6.5 to conclude that x is recurrent. However, in this case, $P(\{f(X_n) \ge f(x) > \underline{f} \text{ i.o.}\}) > 0$, contradicting the almost sure convergence of $\{f(X_n)\}$ to $\underline{f}$. This completes the proof of (ii).

7. Critical points for the SA chains. Next we shall identify a number of critical points for the constant c of an SA chain with logarithmic temperature schedule.

THEOREM 7.1. *Suppose that the SA chain $\{X_n\}$ admits a cooling schedule $\{T_n = c/\log(n_0 + n)\}$ for some constant c.*

1. *Define c_0 to be the smallest $h > 0$ such that*

$$\exists x \in M,\ y \in N(x) : d(x) < h \text{ and } f(x) < f(y) \leq f(x) + h,$$

where M is the set of all local minima, including global minima. Then c_0 is the smallest c such that null recurrent states exist. For $c < c_0$, the SA chain assumes only positive recurrent and transient states, and its collection of eventual traps is maximal in number.

2. *Define*

$$c_1 = \min_{x \in M, x \notin S^*} d(x),$$

where M is the set of all local minima states. c_1 is the smallest c such that the number of recurrent classes that are eventual traps decreases.

3. *Define*

$$c_2 = d^* = \max_{x \in M, x \notin S^*} d(x).$$

Then c_2 is the smallest c such that the algorithm is convergent. It is also the smallest c such that all local minima that are not global minima are null states, or the smallest c such that the only positive recurrent states are global minima states.

4. *Define c_3 to be the smallest c such that a null recurrent local minimum exists. c_3 is the smallest h such that there exist a local or global minimum state x and a local minimum state y with $d(x) > h$ and y reachable at height $f(x) + h$ from x.*

5. *Define*

$$h^* = \max_{x,y \in S^*} \{h : y \text{ is reachable from } x \text{ at height } \underline{f} + h\}$$

and $c_4 = \max\{d^*, h^*\}$. Then c_4 is the smallest c for which weak ergodicity occurs.

6. *Define*

$$c_5 = \max_{x \in S} f(x) - \min_{y \in S} f(y).$$

Then c_5 is the smallest c for which all states are recurrent.

7. *Define $c_6 = +\infty$ in the case when the transition probabilities of the SA chain do not depend on the temperature. Then c_6 is the only c for which all states are positive recurrent.*

Proof. Notice first that by simple manipulations we deduce that for any c_i with $i \in \{1, \ldots, c_6\}$ we have for $\alpha \geq c_i$

$$\sum_{k=1}^{\infty} \exp(-\alpha/T_k) = \infty$$

and

$$\sum_{k=1}^{\infty} \exp(-\alpha/T_k) < \infty$$

for $\alpha < c_i$.

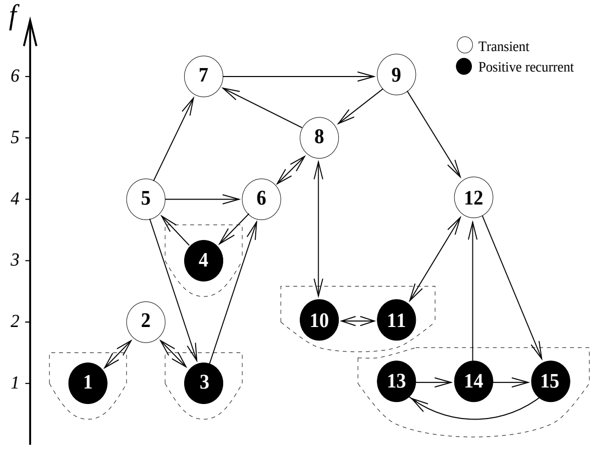


FIG. 7.1. $0 \leq c < 1$. *All local and global minima are positive states. It is a defective algorithm. No null recurrent states exist. The chain may freeze in any (connected set of) local minima including global minima. There are four eventual traps. This also is the case for all fast cooling schedules.*

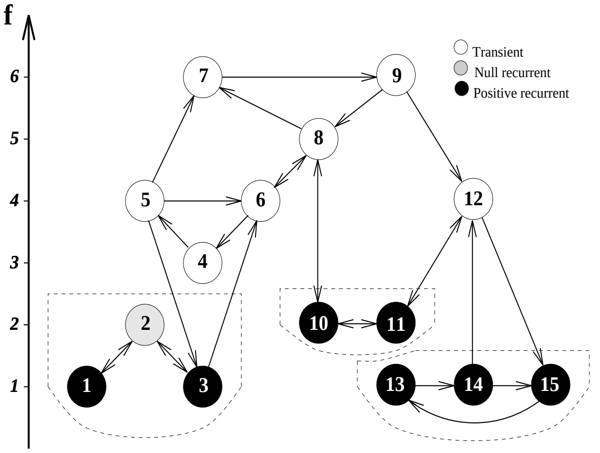


FIG. 7.2. $1 \leq c < 2$. *A null recurrent state first occurs. One local minimum is rendered transient. If the SA chain becomes trapped in the eventual trap containing the null recurrent state, then, strictly speaking, the chain will never freeze. That is, the null recurrent state will be visited infinitely often. Such visits will, however, become less and less frequent and further apart. There are three eventual traps.*

To prove point 1 we take into account that for $c = c_0$ the positive states will remain the same as for $c < c_0$ but, according to Lemma 6.5, the set of null recurrent states will increase.

To prove point 2, notice that for $c = c_1$, at least two eventual traps for $c < c_1$ become merged in one eventual trap. This follows from Theorem 6.1(iii).

Point 3 is also a consequence of Theorem 6.1, because for $c = c_2$ the eventual traps containing global minima must contain all local minima as well. As the probability mass concentrates in the bottom states of a recurrent eventual trap the chain must be convergent.

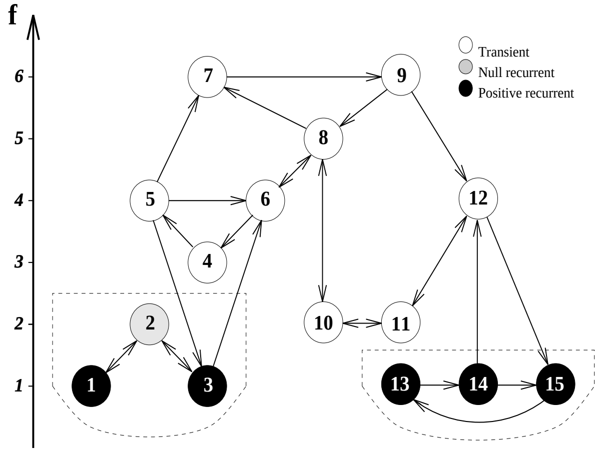


FIG. 7.3. $2 \leq c < 3$. *The canonical cooling schedule is reached. The chain is convergent. There are two eventual traps. All eventual traps contain global minima. Only global minima are positive recurrent states.*

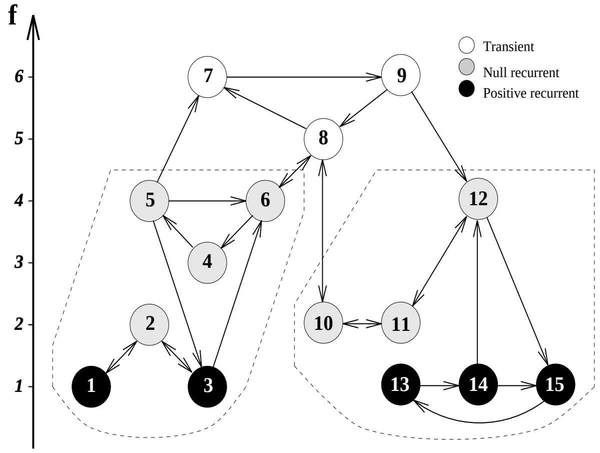


FIG. 7.4. $3 \leq c < 4$. *More states become null recurrent, including a local minimum. The two eventual traps have increased in size.*

To prove point 4, notice that c_3 is defined in such a way that we may choose x to be a bottom state of an eventual trap which makes it positive recurrent, and the condition of Theorem 6.1(iii) is satisfied, implying that y is recurrent. It is easy to see that x and y belong to the same recurrent class for which x is a bottom state and $f(y) > f(x)$. This makes y a null state.

Point 5 follows from the observation that $c = c_4$ does not allow two eventual traps, and this is equivalent to weak ergodicity.

We leave the proofs of 6 and 7 to the reader.

Remark. It has turned out that the critical points identified above belong to a logarithmic cooling schedule. For cooling schedules that go faster to 0 than a logarithmic one, we can easily see that the SA chain behavior is the one described for $c < c_0$. For temperature schedules that are slower than logarithmic, the SA chain

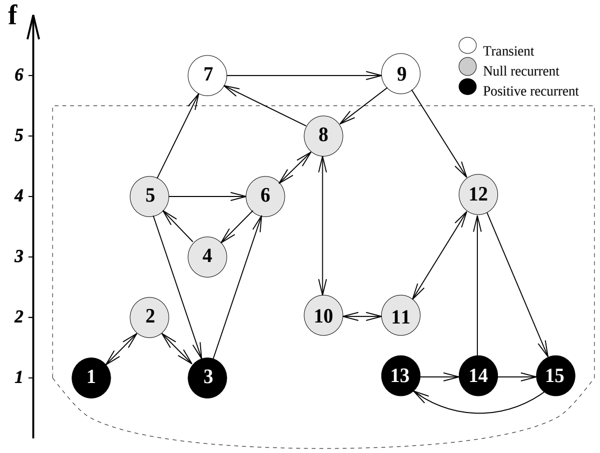


FIG. 7.5. $4 \le c < 5$. *The chain is weakly ergodic. All global minima are contained in the single eventual trap.*

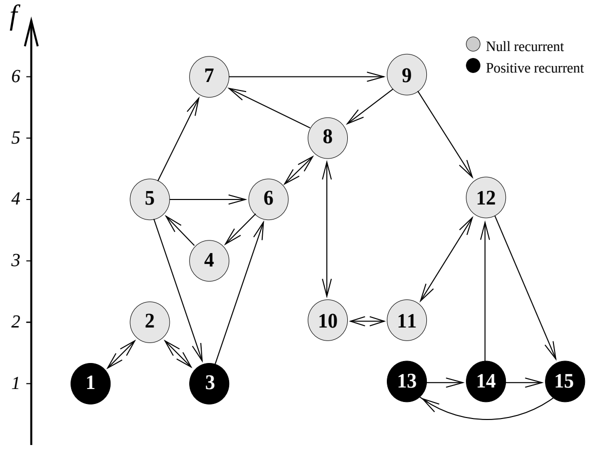


FIG. 7.6. $c \ge 5$. *There is one eventual trap, incorporating the entire state space. All states are recurrent, but only global minima are positive recurrent.*

behavior is as in the case $c \ge c_5$. For temperature schedules with a subsequence of $\{T_n\}$ bounded away from 0, we get a weakly ergodic $\{X_n\}$ with all states positive recurrent.

We shall consider now an example of an SA chain with 15 states to illustrate the asymptotic behavior of slow cooling schedules described above. The example is shown in Figures 7.1–7.7, where the properties of states are depicted at various values of c. The dotted graphs delineate the eventual traps. Marked in black are the positive recurrent, in gray the null recurrent, and in white the transient states. For this example we get $c_0 = c_1 = 1$, $c_2 = 2$, $c_3 = 3$, $c_4 = 4$, and $c_5 = 5$.

8. Fast cooling schedules. Most of the algorithms applied to large problems are of the fast cooling type and are therefore nonconvergent. This is the case for the algorithms of Aarts and van Laarhoven [1], Kirkpatrick, Gellat, and Vecchi [19], and

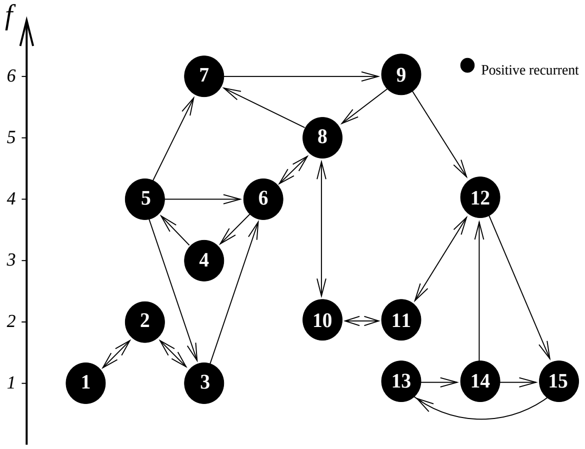


FIG. 7.7. $c = +\infty$. *The case where boiling is employed and all transitions are accepted. No cooling takes place. All states are positive recurrent. This also is the case for any fixed positive temperature.*

Lundy and Mees [23], which satisfy the condition

$$\sum_{n=1}^{\infty} P_n(x,y) < \infty \tag{8.1}$$

for any x, y with $f(y) > f(x)$.

Consider now the Markov chain $\{X_n^A\}$ with state space A and transition probability matrix R, with entries

$$R(x,y) = \begin{cases} G(x,y) & \text{if } f(y) < f(x), \\ 0 & \text{if } f(y) > f(x), \\ 1 - \sum_{z \neq x} G(x,z) & \text{if } f(y) = f(x) \end{cases}$$

for $x, y \in A$. We shall attach such Markov chains to any (local minima) recurrent class A.

THEOREM 8.1. *If* (8.1) *holds, then*

(i) $\{f(X_n)\}$ *converges a.s. to a random variable* W *whose probability mass is concentrated on the set of local and global minima;*

(ii) *all the states except for global and local minima are transient;*

(iii) $\{X_n\}$ *eventually freezes in a set of states of constant objective function* f*;*

(iv) *if* x *belongs to a recurrent class* A *consisting of local or global minima states, then* $\lim_{n\to\infty} P(X_n = x) = P(\Lambda)\pi_x$*, where* $\{\pi_x,\ x \in A\}$ *is the stationary distribution of* $\{X_n^A\}$*, and* $\Lambda = \lim_{n\to\infty}\{X_n \in A\}$ *a.s.*

Proof. We shall show that if A is a recurrent class consisting only of global or local minima of constant f-value, then A is an eventual trap. We shall prove first that for such A we get

$$P(\{X_n \in A \text{ ult.}\}) > 0. \tag{8.2}$$

This is equivalent to showing that

$$\lim_{n\to\infty} P(\cap_{m=n}^{\infty}\{X_m \in A\}) > 0.$$

Write

$$\alpha_n = \max_{x \in A, y \notin A} P_n(x, y). \tag{8.3}$$

By conditioning, we get

$$P(\cap_{m=n}^{r}\{X_m \in A\}) \geq P(\cap_{m=n}^{r-1}\{X_m \in A\})(1-\alpha_{r-1}) \geq$$

$$\cdots\cdots\cdots\cdots$$

$$\geq P(\{X_m \in A\})(1-\alpha_m)\cdots(1-\alpha_{r-1}). \tag{8.4}$$

Letting r tend to ∞ in (8.4) and recalling that $\sum_n \alpha_n < \infty$ we get that $P(\{X_n \in A \text{ ult.}\}) > 0$ and (8.2) is proved.

Notice now that

$$\sum_{n=1}^{\infty} P(\{X_n \in A\} \cap \{X_{n+1} \notin A\}) < \infty, \tag{8.5}$$

which in conjunction with (8.2) and the Barndorff–Nielsen–Borel–Cantelli-type lemma [4] imply that A is an eventual trap. The transient states do not have an a.s. contribution in the limit. (Notice that f is constant on A but its value may differ for various eventual traps being the value of the bottom states which are global or local minima of f.) This completes the proof of (i).

Obviously, (ii) follows from (i).

It is easy to see that (iii) follows from (i) and (ii).

To prove (iv), notice that the assumption of irreducibility and accessibility of states from each other makes any chain $\{X_n^A\}$ ergodic and irreducible. Thus

$$\lim_{n\to\infty} P^{(m,n)}(y, x) = \pi_x$$

for $x, y \in A$, and

$$\lim_{n\to\infty} P^{(m,n)}(y, x) = 0$$

for $x \notin A$. But

$$P(X_n = x) = \sum_{y \in S} P(X_m = y) P^{(m,n)}(y, x).$$

Thus, if $x \in A$,

$$\lim_{n\to\infty} P(X_n = x) = \lim_{m\to\infty} P(X_m \in A)\pi_x = P(\Lambda)\pi_x,$$

and the proof is finished.

9. Some traveling salesman examples. We next investigate the relative performance of a number of cooling schedules used in applications to which we add a fixed temperature schedule. There is no claiming that the algorithms chosen are the most appropriate for the problems. The aim of the exercise is to use statistical analysis to ascertain the quality of various algorithms which appear to be problem dependent. Clearly, for some problems there is a need for faster, nonconvergent cooling than logarithmic cooling. We examine the performance of fast cooling schedules such as Aarts and van Laarhoven [1] and Lundy and Mees [23] and a basic geometric schedule as first introduced by Kirkpatrick, Gellat, and Vecchi [19]. The traveling salesman problems (TSPs) considered vary in size from 48 to 442 cities.

We consider the relative performance of these algorithms allowing a fixed number of iterations N, for an appropriately chosen N.

9.1. Application of SA to the TSP. For the TSP, we consider a path leading through all of n cities, starting in an arbitrary city and finally returning to it. A distance (or possibly time or cost) is given between each pair of cities. We consider here the symmetric TSP, where the distance is the same in either direction. The objective is to identify the path that has the smallest total distance. There are $(n-1)!/2$ possible paths.

The neighborhood structure we employ for the TSP is that generated by 2-opt moves. Consider the cities and the path of the TSP as the vertices and edges of a graph. A 2-opt move is simply the process of deleting and replacing two edges of the graph to yield a new path for the TSP. There are $n(n-3)/2$ different paths that can be created by such a move. (Note that once one edge has been deleted, if either of the neighboring edges is then deleted, it is possible only to reconstruct the original path, leaving $n-3$ edges to choose from.)

The TSP is often stated as a *benchmark* problem for testing optimization procedures. SA is often outperformed by specially tailored algorithms. The merits of SA lie in its ease of implementation and its applicability to a wide range of problems. It is our aim to use the observations of SA on TSPs to gain valuable insight into what criteria constitute an optimal temperature schedule for problems in general.

9.2. The problem instances. We have considered the six problem instances of the TSP examined in Aarts and van Laarhoven [2]. Each problem is labeled by the initials of the author(s) of the reference to it, followed by the number of cities. The problem instances are `gr48` and `gr442` from [14], `gr120` from [13], `kt57` from [17], `kroA100` from [21], and `lin318` from [22]. (We have taken `lin318` in the form of a TSP rather than a Hamiltonian circuit.)

9.3. The different schedules. Following are the rules for updating the temperature in each of the schedules considered.

Aarts: Temperature is held fixed during each loop of $R = \max_{x \in S} |N(x)|$ iterations. At the end of each loop the temperature is dropped according to the rule

$$T_{k+1} = T_k \Bigg/ \left(1 + \frac{T_k \log(1+\delta)}{3\sigma_k}\right),$$

where σ_k is the standard deviation of the observed values of the cost function during the kth loop of the algorithm.

Geometric: The temperature is again held fixed during each loop. We have set the length of each loop to be the same as for Aarts. At the end of each loop the temperature is dropped according to the rule

$$T_{k+1} = \alpha T_k.$$

It is worth noting that we found that the number of iterations performed at each loop had little if any effect on the algorithm's performance, provided the value of α was adjusted appropriately.

Lundy: With Lundy's schedule the temperature is to be dropped after each iteration according to the rule

$$T_{n+1} = \frac{T_n}{1 + \beta T_n},$$

or equivalently,

$$T_n = \frac{T_0}{1 + n\beta T_0}.$$

To keep this algorithm in the same form as above, we update the temperature at the end of each loop of the same number of iterations as above. Again, we did not find this to alter the performance of the algorithm.

Logarithmic: Here again, the temperature is to be updated after each iteration, the rule for which is

$$T_n = \frac{c}{\log(n + n_0)}.$$

Again we update this temperature at the end of each loop of R iterations.

Fixed temperature: In a fixed temperature algorithm an appropriate temperature must be found. We have done so experimentally, by running a fixed temperature schedule for a range of temperatures and choosing the temperature which gives the best performance, say, the best average solution in N iterations. Connolly [9] gives a method for determining a fixed temperature by first running a fast cooling algorithm and noting the temperature at which the best solution found first occurred.

9.4. Method used in comparing the schedules. In an attempt to make a fair comparison of the different schedules, the following method is used.

Measure of performance: We measure the performance of each algorithm by the average best solution found in the N iterations. Results of the algorithms with regards to $P(\tau \leq \mathrm{N})$ are also given, where τ is taken as the time until reaching a global minimum, as well as within one and two percent of the global minimum.

To choose N*:* We wish to choose an N for each problem instance that is sufficiently large, but not too large, for the algorithms to find good heuristic solutions. We have chosen Aarts's algorithm to roughly determine such an N, but Lundy's or the geometric algorithm also could have been used. First, 100 runs of Aarts's algorithm are performed with the parameter setting ($\delta = 0.1$) recommended by its authors. In choosing N, we consider the number of iterations taken until first visiting the best solution found in each run. The maximum of these is taken, after removing outliers. An outlier is taken as a value more than 1.5 times the interquartile range greater than the third quartile. The initial temperature is determined experimentally to yield an initial acceptance ratio of 0.95.

Determining the parameters of the schedules: Once N has been chosen for a given problem, the parameters of Lundy's algorithm and the geometric schedule are

Table 9.1

The parameter settings experimentally found for five temperature schedules for various TSPs. N *is the number of iterations to be allowed for each algorithm and is set according to Aarts's schedule.*

Problem	N	T_0	Aarts	Geom.	Lundy	Logar.	Fixed
			δ	α	β	c	T
gr48	509760	2800	0.1	0.98700	2.546×10^{-7}	250	20
kt57	857223	6000	0.1	0.98920	6.173×10^{-8}	500	40
kroA100	4205532	11500	0.1	0.99220	1.196×10^{-8}	650	45
gr120	7104240	2900	0.1	0.99300	3.704×10^{-8}	150	11
lin318	102173400	11800	0.1	0.99615	1.498×10^{-9}	450	25
gr442	242935584	2420	0.1	0.99670	5.669×10^{-9}	45	2.3

determined experimentally to yield approximately the same N, when determined in the same way. The initial temperature is set as above. For fixed temperature, the optimum temperature is found experimentally by trying various temperatures and finding the one that yields on average the best solution in the N iterations. The logarithmic schedule is very slow and cooling from a high to a low temperature in the given amount of time is not possible. We therefore set $n_0 = 2$ in order to maximize the overall change in temperature, and we determine the optimal value for c in the same way as we determine the optimum fixed temperature.

Stopping the algorithms: Once parameters are chosen, the algorithms are rerun. Upon reaching N iterations the temperature is set to zero, and the algorithms are allowed to (quickly) settle in a local minimum. For the logarithmic schedule the optimum value of c is found, with this final freezing included in the algorithm. This final freezing is also included when searching for the optimal fixed temperature.

9.5. Results. Tables 9.1–9.5 show the results from running the five abovementioned temperature schemes on the six TSP instances. One hundred runs are performed for each instance under each temperature schedule. Table 9.1 shows the number of iterations allowed for each problem instance and the parameters experimentally determined for each algorithm. Reported are the quality of final (best) solutions, iterations taken to reach these solutions, and the proportion of runs reaching global or near global minima solutions. Global minima solutions were found only for the 48, 57, and 100 cities instances.

9.6. Remarks regarding simulations. 1. From the simulations carried out, we see that it is worthwhile having a handful of algorithms available in the application of SA to a particular problem.

2. In the case of the TSP, we see that for smaller problems, the fixed and logarithmic schedules seem to perform as well as and better than the fast cooling schedules. For larger problems the fast cooling schedules seem to perform better. It appears that in such cases the schedule of Lundy and Mees outperforms the Aarts and van Laarhoven and the geometric schedules.

3. The results are likely to differ for different applications of SA. Lundy and Mees's algorithm initially cools more rapidly than the other two fast cooling schedules, and it spends more time at smaller temperatures. It may be the case, however, that the slower initial cooling of the other schedules is crucial in other applications.

4. We see that for the 48-city and 120-city TSPs, fixed temperature and the logarithmic schedule outperform the fast cooling schedules. The results suggest that it is not simply the size of the problem that is important but the structure as well. It may be the case that for applications other than the TSP, the structure of the

Table 9.2

A comparison of different temperature schedules, in a fixed number of iterations, for various TSPs. Mean and standard deviation given for 100 *runs in each case. The solution in each run is taken as the best solution visited.*

Problem	Average best solution (% above global)				
	Aarts	Geom.	Lundy	Logar.	Fixed
gr48	0.88	0.66	0.38	0.24	0.25
kt57	1.07	0.84	0.40	0.56	0.59
kroA100	0.96	0.75	0.49	0.49	0.54
gr120	1.83	1.40	1.07	0.69	0.85
lin318	1.73	1.45	1.34	2.16	2.37
gr442	1.66	1.34	1.05	2.00	2.10
	Standard deviation (% above global)				
gr48	0.69	0.54	0.40	0.28	0.28
kt57	0.85	0.77	0.54	0.69	0.69
kroA100	0.70	0.52	0.38	0.38	0.56
gr120	0.72	0.66	0.50	0.39	0.43
lin318	0.52	0.45	0.45	0.67	0.79
gr442	0.51	0.37	0.39	0.47	0.45

Table 9.3

Mean and standard deviation of iterations taken until finding the best solution of each run.

Problem	Average iterations until best in run				
	Aarts	Geom.	Lundy	Logar.	Fixed
gr48	478375	440705	280476	280260	247698
kt57	809653	763375	540635	474612	494819
kroA100	4113819	3821121	2927120	2703681	2843458
gr120	6828916	6305855	4467879	4932393	4586938
lin318	100592216	95016256	72002496	99782344	96906464
gr442	237841104	226062032	160714880	222006640	206698016
	Standard deviation of iterations until best				
gr48	15981	27161	81559	138893	145977
kt57	20622	36932	117102	226153	240173
kroA100	64690	173474	720508	1257878	1339295
gr120	114856	318401	1194903	1658310	2017348
lin318	835365	2079017	11570513	9369117	16177562
gr442	1823134	6554549	31531302	34883836	52499600

Table 9.4

Estimates for $P(\tau \leq \mathrm{N})$, where τ is the time to reaching a global minimum, for N *as given in Table* 9.1*. A global minimum was never reached in any of the runs for the larger problems.*

Problem	Proportion reaching global minimum				
	Aarts	Geom.	Lundy	Logar.	Fixed
gr48	0.05	0.17	0.35	0.30	0.34
kt57	0.04	0.06	0.28	0.31	0.30
kroA100	0.03	0.06	0.10	0.04	0.00

problem means that fixed and logarithmic schedules are suited to large problems too.

5. We do not know whether the logarithmic cooling schedule used is convergent, as we have not identified the canonical constant. Indeed, d^* is not readily available, and to get it, when feasible, may require much more extensive work than finding an optimal state. In fact, as we pointed out before, convergence is not relevant to the success of the algorithm.

Table 9.5

Estimates for $P(\tau \leq \mathrm{N})$ when τ is taken, respectively, as the time to reaching solutions with the objective function at most one percent and two percent larger than the global minimum.

Problem	Proportion reaching within 1% of global				
	Aarts	Geom.	Lundy	Logar.	Fixed
gr48	0.68	0.81	0.97	1.00	1.00
kt57	0.58	0.71	0.86	0.67	0.65
kroA100	0.62	0.76	0.91	0.93	0.88
gr120	0.11	0.27	0.44	0.80	0.69
lin318	0.07	0.13	0.19	0.01	0.02
gr442	0.12	0.23	0.39	0.02	0.00
	Proportion reaching within 2% of global				
gr48	0.92	0.97	1.00	1.00	1.00
kt57	0.89	0.89	0.99	1.00	1.00
kroA100	0.93	0.98	0.99	1.00	0.96
gr120	0.57	0.81	0.95	0.99	0.99
lin318	0.70	0.88	0.92	0.52	0.34
gr442	0.77	0.98	1.00	0.49	0.43

6. We have seen that obtaining a global minimum is plausible for some small to medium-size problems. For the 100-city TSP, using Lundy's schedule, we get for the time until reaching a global minimum,

$$P(\tau < N) \approx 0.10,$$

for $N = 4205532$. Using $k = 50$ reruns (2.3) becomes

$$P(\min_{i \in \{1,\ldots,k\}} \min(f(X_1^{(i)}), \ldots, f(X_{\mathrm{N}}^{(i)})) = \min_{x \in S} f(x)) \approx 0.995,$$

and with $k = 100$ we get a probability of 0.99997 of reaching the global minimum.

10. Concluding remarks. 1. We have looked at the limit behavior of the SA chain as a function of its temperature schedule. The quality of an algorithm depends on its parameters, and the temperature schedule is only one ingredient of an algorithm. However, the limit behavior of the SA chain is determined only by its temperature schedule.

2. We have a three-type classification for an algorithm: convergent, regular, and defective. Examples are provided to illustrate situations when boiling gives the optimal algorithm, when logarithmic or fixed temperature outperform a number of faster cooling schedules, or when defective algorithms are better for the problem.

3. We characterized the limit behavior of an algorithm in terms of recurrence, transience, and eventual traps. It turns out that a convergent chain may have several eventual traps or may consist of one eventual trap, as in the weakly ergodic case. A regular algorithm is not necessarily convergent. It may be weakly ergodic but not convergent. A convergent chain or a chain with a fixed temperature will exhibit a lot of changes in its objective function values, as there are usually recurrent states that are neither global nor local minima. Such changes will become less and less frequent but will not disappear. In contrast, a defective chain does not have recurrent states outside global or local minima states and will eventually have its objective function value frozen in a local or global minimum.

4. The critical points for algorithms where the asymptotic behavior changes are all in the range of logarithmic cooling schedules. There are two extreme types of behavior: the first, when each local mimimum is an eventual trap, and the second,

when all states are recurrent. It may seem that the first case does not lead to a good algorithm. However, for large problems these type of schedules usually outperform the convergent and regular ones. It may also seem that the latter compares unfavorably to the canonical cooling schedule which prescribes a convergent chain with the minimal number of recurrent states. However, such an impression is also deceptive.

5. When using a memoryless algorithm for a convergent chain or a memory algorithm for a regular chain, we know that reaching global minima may be achieved with probability as large as desired if we let the chain run a sufficiently long time. However, that may not be feasible in practice, as it may require an excessively long time. In contrast, for defective algorithms we know that the probability of reaching optimality is limited, often by a small number. However, repeated independent runs may ensure a high quality for such algorithms, which are often used in practice.

Acknowledgment. The authors are thankful to the referees for a number of useful comments.

REFERENCES

[1] E. H. L. Aarts and P. J. M. van Laarhoven, *Statistical cooling: A general approach to combinatorial optimization problems*, Philips J. Res., 40 (1985), pp. 193–226.

[2] E. H. L. Aarts and P. J. M. van Laarhoven, *Simulated Annealing: Theory and Applications*, Reidel, Dordrecht, the Netherlands, 1987.

[3] E. H. L Aarts and J. Korst, *Simulated Annealing and Bolzman Machines*, John Wiley, New York, 1989.

[4] O. Barndorff-Nielsen, *On the limit behaviour of extreme order statistics*, Ann. Math. Statist., 34 (1963), pp. 992–1002.

[5] T. S. Chiang and Y. Chow *On the convergence rate of annealing processes*, SIAM J. Control Optim., 26 (1988), pp. 1455–1470.

[6] H. Cohn, *On a paper by Doeblin on non-homogeneous Markov chains*, Adv. Appl. Prob., 13 (1981), pp. 388–401.

[7] H. Cohn, *On a class of non-homogeneous Markov chains*, Math. Proc. Cambridge Philos. Soc., 92 (1982), pp. 527–534.

[8] H. Cohn, *Products of stochastic matrices and applications*, Internat. J. Math Math. Sci., 12 (1988), pp. 209–233.

[9] D. T. Connolly, *An improved annealing scheme for the QAP*, European J. Oper. Res., 46 (1990), pp. 93–100.

[10] D. P. Connors and P. R. Kumar, *Balance of recurrence order in time-inhomogeneous Markov chains with applications to simulated annealing*, Probab. Engrg. Inform. Sci., 2 (1988), pp. 157–184.

[11] S. B. Gelfand and S. K. Mitter, *Analysis of simulated annealing for optimization*, in Proc. 24th Conference on Decision and Control, Ft. Lauderdale, FL, 1985, pp. 779–786.

[12] S. Geman and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Trans. Pattern. Anal. Machine Intelligence, 6 (1984), pp. 721–741.

[13] M. Grötschel, *Polyedrische Charakterisierungen Kombinatorischer Optimierungsprobleme*, Hain, Meisenheim am Glan, 1977.

[14] M. Grötschel and O. Holland, *Solution of large-scale symmetric traveling salesman problems*, Math. Programming, 51 (1991), pp. 141–202.

[15] B. Hajek, *Cooling schedules for optimal annealing*, J. Math. Oper. Res., 13 (1988), pp. 311–329.

[16] C. R. Hwang and S. J. Sheu, *Singular perturbed Markov chains and the exact behaviors of simulated annealing processes*, J. Theoret. Probab., 5 (1992), pp. 223–249.

[17] R. L. Karg and G. L. Thompson, *A heuristic approach to solving traveling salesman problems*, J. Management Sci., 10 (1964), pp. 225–248.

[18] J. G. Kemeny and J. L. Snell *Finite Markov Chains*, Springer-Verlag, New York, 1976.

[19] S. Kirkpatrick, C. D. Gellat, and M. P. Vecchi *Optimization by simulated annealing*, Science, 220 (1983), pp. 671–680.

[20] S. Kirkpatrick, *Optimization by simulated annealing: Quantitative studies*, J. Statist. Phys., 34 (1983), pp. 975–986.

[21] P. D. Krolak, W. Felts, and G. Marble, *A man-machine approach toward solving the traveling salesman problem*, Comm. ACM, 14 (1971), pp. 327–334.

[22] S. Lin and B. W. Kernighan, *An effective heuristic algorithm for the traveling salesman problem*, J. Oper. Res., 21 (1973), pp. 498–516.

[23] M. Lundy and A. Mees *Convergence of an annealing algorithm*, Math. Programming, 34 (1986), pp. 111–124.

[24] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. Teller, and E. Teller, *Equations of state calculations by fast computing mashines*, J. Chem. Phys., 21 (1953), pp. 1087–1092.

[25] A. Mukherjea, *A new result on the convergence of non-homogeneous Markov chains*, Trans. Amer. Math. Soc., 90 (1981), pp. 167–182.

[26] W. Niemiro, *Limit distributions of simulated annealing Markov chains*, Disc. Math. Algebra Stochastic Methods, 15 (1997), pp. 241–269.

[27] W. Niemiro and P. Pokarowski, *Tail events of some non-homogeneous Markov chains*, Ann. Appl. Probab., 5 (1995), pp. 261–293.

[28] F. Romeo and A. Sangiovanni-Vincentelli, *Probabilistic hill climbing algorithms: Properties and applications*, Proc. Chapel Hill Conference on VLSI, H. Fuchs, ed., Computer Science Press, Rockville, MD, 1985, pp. 393–417.

[29] E. Seneta, *Non-Negative Matrices and Markov Chains*, Springer-Verlag, New York, 1981.

DEDICATION TO JOHN E. DENNIS, JR. ON THE OCCASION OF HIS 60TH BIRTHDAY

SIAM Journal on Optimization would not exist were it not for the vision, energy, and dedication of John E. Dennis, Jr. It was he who, in the late 1980s, recognized the need for a SIAM journal focusing broadly on optimization. He was inspired partly by the success of the SIAM conferences on optimization held regularly since 1984 and partly by the example of the Mathematical Programming Society, whose effectiveness in establishing optimization, especially its algorithmic aspects, as a discipline rested both on its international symposia and on its well-respected journal. John had long played an important role in SIAM, serving as an editor for *SIAM Journal on Numerical Analysis* from 1975 to 1981, as co-chair of the second SIAM Conference on Optimization in Houston in 1987, as a member of the SIAM Council from 1985 to 1990, as chair of the SIAM activity group on optimization from 1989 to 1992, and as an advocate for optimization as a subject that has a natural home in SIAM. When *SIAM Journal on Optimization* was established, he was a natural choice as Editor-in-Chief, serving from 1990 to 1994, when he stepped down to become the Chair of the Mathematical Programming Society. With the passage of time, it is clear that the establishment of the journal brought optimization to full status as one of the leading disciplinary areas within SIAM.

The two of us have known John since the late 1970s when we were graduate students, one of us as his advisee. He was very supportive to us as young scientists and by his support played a critical role throughout our careers, for which we are enormously grateful. John is particularly proud of the success of his more than thirty Ph.D. graduates at Cornell and Rice; seven of these former students are authors of papers in this special issue. John has always made a special point of showing interest in young scientists beginning their careers, whether or not they studied with him; in our view, nothing a senior scientist can do is more important than that.

John began his career in Utah as a functional analyst and only later turned to computational mathematics. He has authored or coauthored dozens of well known papers in optimization and applied mathematics. To single out one contribution is difficult, but he is particularly well known for his pioneering convergence analysis of quasi-Newton methods (also known as secant or variable metric methods) with C. G. Broyden and J. J. Moré, and his survey paper with Moré in *SIAM Review* (1977) became required reading for a generation of graduate students. In more recent years John's special interest has been multidisciplinary optimization, emphasizing industrial application of optimization, especially in the aeronautical and oil industries. Nothing could be closer to the central mission of SIAM.

John has long been an advocate of electronic publication, and we find it especially appropriate that, exploiting this medium, we are publishing this special issue of *SIAM Journal on Optimization* actually on his 60th birthday. We are honored to be able to dedicate this issue to John Dennis. Finally, we also salute his family: Ann, Jed and Katie, of whom he is so proud. Happy Birthday, John, and many happy returns!

Michael L. Overton and Robert B. Schnabel

SIAM J. Optim.
Vol. 9, No. 4, pp. 803–812

LINEAR PROGRAMMING IN $O\big(\frac{n^3}{\ln n}L\big)$ OPERATIONS*

K. M. ANSTREICHER†

To John Dennis on the occasion of his 60th birthday.

Abstract. We show that the complexity to solve linear programming problems, using standard linear algebra, can be reduced to $O([n^3/\ln n]L)$ operations, where n is the number of variables in a standard-form problem with integer data of bit size L. Our technique combines partial updating with a preconditioned conjugate gradient method, in a scheme first suggested by Nesterov and Nemirovskii.

Key words. linear programming, interior point algorithm, partial updating, conjugate gradient method

AMS subject classification. 90C25

PII. S1052623497323194

1. Introduction.

Consider a standard-form linear program and its dual:

$$\begin{array}{lll} \text{LP}: & \min & c^Tx \\ & \text{s.t.} & Ax=b, \\ & & x\ge 0 \end{array} \qquad \begin{array}{lll} \text{LD}: & \max & b^Ty \\ & \text{s.t.} & A^Ty+s=c, \\ & & s\ge 0, \end{array}$$

where A is an $m\times n$ matrix. We assume without loss of generality that the rows of A are linearly independent. For the purpose of stating complexity results we may assume that the data of LP is integral and let L denote the bit size of the problem.

Karmarkar's [5] celebrated projective algorithm solves LP in $O(n^4L)$ operations, where here and throughout the paper we use "operations" to refer to arithmetic operations and comparisons in infinite precision. The overall complexity for Karmarkar's basic algorithm arises from $O(nL)$ iterations, each requiring $O(n^3)$ operations. Using a "partial updating" technique, Karmarkar also devised a modified algorithm that reduced the average work per iteration to $O(n^{2.5})$ operations, while retaining the $O(nL)$ iteration complexity, for an overall complexity of $O(n^{3.5}L)$ operations. Subsequently Renegar [11] devised a "path following" method that reduced the number of iterations to $O(\sqrt{n}L)$, while still requiring $O(n^3)$ operations per iteration. By adapting Karmarkar's partial updating strategy to Renegar's path following algorithm, Gonzaga [4] and Vaidya [15] obtained the first algorithms for LP with an overall complexity of $O(n^3L)$ operations. Many subsequent papers have obtained $O(n^3L)$ methods for LP, by using partial updating in a variety of algorithmic frameworks.

A small number of papers have discussed the use of fast matrix multiplication to improve the complexity of interior point methods for linear programming; see, for example, [14]. "Fast matrix multiplication" refers to the fact that the multiplication of two $m\times m$ matrices, and the inversion of an $m\times m$ matrix, can both be performed in $O(m^{2+\alpha})$ operations for $\alpha<1$ [2]. Nesterov and Nemirovskii [10, Chapter 8] consider a number of different strategies for reducing the complexity of interior point methods

*Received by the editors June 20, 1997; accepted for publication (in revised form) February 2, 1999; published electronically September 24, 1999.

http://www.siam.org/journals/siopt/9-4/32319.html

†Department of Management Sciences, University of Iowa, Iowa City, IA 52242 (kurt-anstreicher@uiowa.edu). This research was conducted while visiting the Center for Operations Research and Econometrics, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, with support from a CORE fellowship.

for linear and quadratic programming, including combinations of partial updating, fast matrix multiplication, and iterative methods for approximately solving positive definite linear equations. For $\alpha = 1$ the various "acceleration" methods considered in [10] all produce $O(n^3L)$ methods for LP, but when $\alpha < 1$ lower overall complexities are obtained. For all $0 < \alpha < 1$ the best overall complexity is obtained by a method that combines partial updating with a preconditioned conjugate gradient (PCG) method to approximately solve the Newton equations produced on each iteration of a path following algorithm. The same method, with additional consideration of parallelization, is also described in [9].

In this paper we reexamine Nesterov and Nemirovskii's PCG algorithm, using standard linear algebra ($\alpha = 1$). We find, somewhat surprisingly, that the algorithm can be specified so as to have an overall complexity slightly below $O(n^3L)$, specifically $O([n^3/\ln n]L)$ operations. To our knowledge, this is the first complexity result for LP below $O(n^3L)$ using standard linear algebra. The algorithm we analyze is actually simpler than Nesterov and Nemirovskii's PCG method, because some rather complex details required to get the best possible results with $\alpha < 1$ are eliminated. We present our complexity analysis in the simplest possible algorithmic setting, that of a short-step path following algorithm for LD, so as to concentrate as much as possible on the complexity improvement from the PCG strategy. However, it is important to note that the same complexity improvement could be obtained by applying the strategy described here to virtually *any* $O(n^3L)$ partial updating algorithm for linear programming or linearly constrained quadratic programming, including barrier function methods, potential reduction methods, and primal-dual algorithms.

Notation. We use standard notation. For $s \in \Re^n$, $S = \mathrm{Diag}(s)$ denotes the $n \times n$ diagonal matrix with $S_{ii} = s_i$, $i = 1, \dots, n$. We use e to denote a vector of arbitrary dimension with each component equal to 1. For symmetric matrices B and D, $B \preceq D$ denotes that $D - B$ is positive semidefinite. For B positive definite, we use $\|y\|_B = \sqrt{y^T B y}$.

2. A short-step path following algorithm.

The algorithm we consider is a modified version of a short-step path following method due to Roos and Vial [12]. For (y, s) feasible in LD, and a scalar $\mu > 0$, consider the measure

$$\delta(s, \mu) = \min_{x \mid Ax=b} \left\| \frac{Xs}{\mu} - e \right\|. \tag{2.1}$$

Define

$$\begin{aligned} p_y(s,\mu) &= -(AS^{-2}A^T)^{-1}(AS^{-1}e - b/\mu),\\ p_s(s,\mu) &= -A^T p_y(s,\mu),\\ x(s,\mu) &= \mu(S^{-1}e - S^{-2}p_s(s,\mu)). \end{aligned}$$

Then $p_y = p_y(s, \mu)$ is the Newton direction for the logarithmic barrier function

$$f(y,\mu) = \frac{-b^T y}{\mu} - \sum_{i=1}^{n} \ln(s_i(y)),$$

where $s(y) = c - A^T y$, and $p_s = p_s(s, \mu)$ is the corresponding direction in the slack variables s. Moreover, it is straightforward to show that $x(s, \mu)$ is the solution of the minimization problem that defines $\delta(s, \mu)$, from which it follows that

$$\delta(s,\mu) = \|S^{-1}p_s\| = \|p_y\|_H = \|g\|_{H^{-1}}, \tag{2.2}$$

where $H = H(y) = AS^{-2}A^T$, $g = g(y) = (AS^{-1}e - b/\mu)$. Note that (2.2) implies that $\delta(s(y), \mu) = 0$ if and only if y is the minimizer of $f(\cdot, \mu)$. The set of such minimizers for $\mu \in (0, \infty)$ is called the *central trajectory* for LD. The quantity $\delta(s(y), \mu)$ can be considered to be a measure of the proximity of y to the central trajectory.

In what follows, we will assume that $\tilde{p}_y = \tilde{p}_y(s, \mu)$ is an *approximate* solution of the Newton equations $Hp = -g$. Letting $\tilde{p}_s = \tilde{p}_s(s, \mu) = -A^T\tilde{p}_y$, we will assume that $(\tilde{p}_y, \tilde{p}_s)$ satisfies

$$\|S^{-1}(\tilde{p}_s - p_s)\| = \|\tilde{p}_y - p_y\|_H \le \gamma_1 \|p_y\|_H, \tag{2.3}$$

where $0 \le \gamma_1 < 1$. For such an approximate solution $(\tilde{p}_y, \tilde{p}_s)$ we consider a step of the form

$$y' = y + \tilde{p}_y, \qquad s' = s + \tilde{p}_s. \tag{2.4}$$

The next lemma extends the well-known convergence result of Roos and Vial [12] for Newton steps ($\gamma_1 = 0$) to the case of the approximate Newton steps ($\gamma_1 > 0$) used in our algorithm.

LEMMA 2.1. *Let (y', s') be as in (2.4), where $(\tilde{p}_y, \tilde{p}_s)$ satisfy (2.3), and $(1 + \gamma_1)\delta(s, \mu) < 1$. Then $s' > 0$, and $\delta(s', \mu) \le \gamma_1\delta(s, \mu) + (1 + \gamma_1)\delta(s, \mu)^2$.*

Proof. We have $S^{-1}\tilde{p}_s = S^{-1}p_s + S^{-1}(\tilde{p}_s - p_s)$, so (2.2) and (2.3) together imply that

$$\|S^{-1}\tilde{p}_s\| = \|\tilde{p}_y\|_H \le (1 + \gamma_1)\delta(s, \mu) < 1, \tag{2.5}$$

the last by assumption. Then $s' > 0$ follows from (2.4). Next, by definition, we have

$$\begin{aligned}
\delta(s', \mu) &= \min_{x \mid Ax = b} \left\| \frac{Xs'}{\mu} - e \right\| \\
&\le \left\| \frac{S'x(s, \mu)}{\mu} - e \right\| \\
&= \|(S + \tilde{P}_s)(S^{-1}e - S^{-2}p_s) - e\| \\
&\le \|S^{-1}(\tilde{p}_s - p_s)\| + \|S^{-2}\tilde{P}_s p_s\|. \qquad (2.6)
\end{aligned}$$

Moreover,

$$\|S^{-2}\tilde{P}_s p_s\| \le \|S^{-1}\tilde{p}_s\| \, \|S^{-1}p_s\| \le (1 + \gamma_1)\|S^{-1}p_s\|^2, \tag{2.7}$$

where the last inequality uses (2.5) and (2.2). The proof is completed by combining (2.3), (2.6), and (2.7). □

We will use Lemma 2.1, with appropriate γ_1, to control the proximity measure $\delta(\cdot, \cdot)$ following reductions in the parameter μ. The effect of such reductions on $\delta(s, \cdot)$ is given in the following very well known lemma [12]. We give the proof for completeness.

LEMMA 2.2. *Let $0 < \mu' \le \mu$. Then*

$$\delta(s, \mu') \le \left(\frac{\mu}{\mu'}\right)\delta(s, \mu) + \left(\frac{\mu}{\mu'} - 1\right)\sqrt{n}.$$

Proof. Using (2.2) and the definition of $p_s(\cdot, \cdot)$, we have

$$\begin{aligned}
\delta(s, \mu') &= \|S^{-1}p_s(s, \mu')\| \\
&= \left\| \left(\frac{\mu}{\mu'}\right) S^{-1}p_s(s, \mu) + \left(1 - \frac{\mu}{\mu'}\right) S^{-1}A^T(AS^{-2}A^T)^{-1}AS^{-1}e \right\| \\
&\le \left(\frac{\mu}{\mu'}\right) \|S^{-1}p_s(s, \mu)\| + \left(\frac{\mu}{\mu'} - 1\right)\sqrt{n},
\end{aligned}$$

where the last inequality uses $\|e\| = \sqrt{n}$. □

The algorithm that we will employ to solve LP/LD is as follows.

ALGORITHM (modified short-step path following).

```
Given $\epsilon > 0$, $\gamma_1 > 0$, $\gamma_2 > 0$, $y^0$, $s^0$, $\mu^0$, $k := 0$.
Do Until $\mu^k \le \epsilon/(2n)$
    $\mu^{k+1} := (1 - \gamma_2/\sqrt{n})\mu^k$
    Compute $\tilde{p}_y = \tilde{p}_y(s^k, \mu^{k+1})$ satisfying (2.3)
    $y^{k+1} := y^k + \tilde{p}_y$, $s^{k+1} := s^k + \tilde{p}_s$
    Perform updates
    $k := k + 1$
End
```

For each k, the computation of $\tilde{p}_y = \tilde{p}_y(s^k, \mu^{k+1})$ will be accomplished using a PCG method, described in the next section. The PCG method requires that a certain approximation of H^{-1} be maintained via rank-1 updates; the "Perform updates" step of the algorithm refers to these rank-1 changes. The updating procedure is described in detail in section 4.

The original algorithm of Roos and Vial [12] is simply the above method with $\gamma_1 = 0$, so $\tilde{p}_y$ and $\tilde{p}_s$ are replaced by p_y and p_s, respectively. In this case the PCG method and updating steps are not used; instead, the true Newton direction p_y is obtained by direct factorization of $H = AS^{-2}A^T$.

Below we give a complexity result for the number of iterations required by the modified short-step path following algorithm to approximately solve a linear programming problem, under standard assumptions regarding initialization. See, for example, Monteiro and Adler [7] for details on satisfying these initialization assumptions for an arbitrary linear program.

THEOREM 2.3. *Let $\gamma_1 = .20$, $\gamma_2 = .10$, $n \ge 4$. Suppose that the above algorithm is initialized with y^0, $s^0 > 0$, μ^0 such that $\delta(s^0, \mu^0) \le .20$. Then $s^k > 0$, and $\delta(s^k, \mu^k) \le .20$ for all k. Moreover, if $n\mu^0 = 2^{O(L)}$ and $\epsilon = 2^{-2L}$, the algorithm terminates with s^k, and $x^k = x(s^k, \mu^k) > 0$ having $(x^k)^T s^k \le \epsilon$, in $k = O(\sqrt{n}L)$ iterations.*

Proof. Assume that $\delta(s^k, \mu^k) \le .20$. From Lemma 2.2 we have

$$\begin{aligned} \delta(s^k, \mu^{k+1}) &\le \frac{1}{1 - \gamma_2/\sqrt{n}} \delta(s^k, \mu^k) + \sqrt{n} \frac{\gamma_2/\sqrt{n}}{1 - \gamma_2/\sqrt{n}} \\ &\le \frac{1}{1 - .10/2}(.20 + .10) \\ &< .32, \end{aligned} \tag{2.8}$$

where the second inequality uses $n \ge 4$. From Lemma 2.1 we then obtain $s^{k+1} > 0$, and

$$\delta(s^{k+1}, \mu^{k+1}) \le .2(.32) + (1.2)(.32)^2 < .19,$$

so by induction, $s^k > 0$ and $\delta(s^k, \mu^k) \le .20$ for all k. That the algorithm terminates in $O(\sqrt{n}L)$ iterations follows easily from the assumption on μ^0, and the fact that $\mu^k = (1 - \gamma_2/\sqrt{n})^k \mu^0$. That $x^k = x(s^k, \mu^k) > 0$ is immediate from $s^k > 0$, and $\delta(s^k, \mu^k) = \|X^k s^k/\mu - e\| < 1$. Finally, for each k we have

$$\|X^k s^k - \mu^k e\| = \mu^k \delta(s^k, \mu^k) \le .2\mu^k.$$

It follows that $(x^k)^T s^k = e^T X^k s^k \le n\mu^k + .2\sqrt{n}\mu^k < 2n\mu^k$, and therefore $\mu^k \le \epsilon/(2n)$ implies that $(x^k)^T s^k \le \epsilon$. □

Theorem 2.3 gives a complexity result in terms of the parameter L, which is commonly taken to be the bit size of an instance of LP with integer data. The complexity of the algorithm can alternatively be given directly in terms of the termination tolerance ϵ. In particular, it is easy to show that under the conditions of Theorem 2.3, but with the condition on μ^0 changed to $n\mu^0 = O(1/\epsilon)$, the number of steps of the algorithm required to obtain $(x^k)^T s^k \le \epsilon$ is $O(\sqrt{n}\,|\ln \epsilon|)$.

3. The PCG method. On each iteration of the algorithm in the previous section we must approximately solve a system of the form

$$Hp = -g, \tag{3.1}$$

where $H = AS^{-2}A^T$. We accomplish this using a PCG method. Before describing our exact methodology we review some basic properties of the ordinary conjugate gradient (CG) method; see, for example, [3] or [6] for more details. The CG method is an iterative algorithm for solving a system of the form

$$Qv = q, \tag{3.2}$$

where Q is an $m \times m$ positive definite matrix. The algorithm produces a sequence v^i, $i \ge 0$, where v^0 is given. The computation of v^{i+1} from v^i requires $O(1)$ inner products of vectors in $\Re^m$, and matrix-vector products using the matrix Q. There is a variety of results concerning the convergence of the algorithm. For example [6], it is very well known that if Q has k distinct eigenvalues, then v^k solves (3.2). Here we will use the fact [6, p. 258] that if $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum eigenvalues of Q, then

$$\|v^i - v^*\|_Q \le 2\left(\frac{1-\sqrt{\lambda_{\min}/\lambda_{\max}}}{1+\sqrt{\lambda_{\min}/\lambda_{\max}}}\right)^i \|v^0 - v^*\|_Q, \tag{3.3}$$

where $v^* = Q^{-1}q$ is the solution of (3.2). Letting $v^0 = 0$, (3.3) implies that in order to obtain $\|v^i - v^*\|_Q \le \gamma_1 \|v^*\|_Q$, it suffices to have

$$i \ln\left(1 - \frac{2\sqrt{\lambda_{\min}/\lambda_{\max}}}{1+\sqrt{\lambda_{\min}/\lambda_{\max}}}\right) \le \ln\left(\frac{\gamma_1}{2}\right). \tag{3.4}$$

Since $\lambda_{\min}/\lambda_{\max} \le 1$, and $\ln(1-t) \le -t$ for $0 \le t < 1$, (3.4) certainly holds for

$$i \ge \ln\left(\frac{2}{\gamma_1}\right)\sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}}\,. \tag{3.5}$$

The PCG method for solving a system of the form (3.1) is based on applying a symmetric transformation to H before applying the CG method. In particular, consider a positive definite matrix W, and let

$$Q = W^{-1/2}HW^{-1/2}, \qquad q = -W^{-1/2}g, \qquad v = W^{1/2}p.$$

The systems (3.1) and (3.2) are then clearly equivalent. Moreover, for $v = W^{1/2}p$ we have

$$\|v\|_Q = \sqrt{v^TQv} = \sqrt{p^THp} = \|p\|_H. \tag{3.6}$$

Let $p_y = -H^{-1}g = W^{-1/2}v^*$ be the solution of (3.1). It then follows from (3.6) that if the CG method is applied to (3.2) starting at $v^0 = 0$, and $p^i = W^{-1/2}v^i$ for $i \geq 0$, the number of steps required to obtain

$$\|p^i - p_y\|_H \leq \gamma_1 \|p_y\|_H$$

is bounded exactly as in (3.5), where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum eigenvalues of $Q = W^{-1/2}HW^{-1/2}$. Finally, it is well known [3, p. 151] that the PCG method can be implemented so as to require only the matrix W^{-1} (or alternatively, a suitable factorization of W), and not $W^{-1/2}$. Each step of the method, which obtains p^{i+1} from p^i, requires $O(1)$ inner products of vectors in $\Re^m$, and matrix-vector products involving the matrices H and W^{-1}.

Recall that in our case, $H = AS^{-2}A^T$. *It is important to note that H is never explicitly formed.* Since the PCG method is used to solve (3.1), we only require matrix-vector products using H, which can be obtained from A and s. Our preconditioning matrix W will be of the form $W = A\tilde{S}^{-2}A^T$, where

$$\frac{1}{\rho} \leq \left(\frac{\tilde{s}_i}{s_i}\right)^2 \leq \rho, \qquad i = 1, \ldots, n, \tag{3.7}$$

and $\rho > 1$. The parameter ρ is *not* assumed to be $O(1)$, as in the usual construction of partial updating interior point algorithms (see, for example, [1], [4], [5], [15]). From (3.7) it follows that

$$\frac{1}{\rho}W \preceq H \preceq \rho W,$$

and therefore

$$\frac{1}{\rho}I \preceq W^{-1/2}HW^{-1/2} \preceq \rho I.$$

Letting $\lambda_{\min}$ and $\lambda_{\max}$ denote the minimum and maximum eigenvalues of $Q = W^{-1/2}HW^{-1/2}$, as above, we then have

$$\sqrt{\lambda_{\max}/\lambda_{\min}} \leq \rho. \tag{3.8}$$

THEOREM 3.1. *Suppose that the PCG method is applied to* (3.1), *using* $W = A\tilde{S}^{-2}A^T$, *where* $\tilde{s}$ *satisfies* (3.7). *Then for* $p^0 = 0$, *in* $i = \lceil \rho \ln(2/\gamma_1) \rceil$ *steps, each requiring* $O(mn)$ *operations, the algorithm obtains* $\tilde{p}_y = p^i$ *such that* $\|\tilde{p}_y - p_y\|_H \leq \gamma_1\|p_y\|_H$.

Proof. The number of steps required follows immediately from (3.5) and (3.8). That each step requires $O(mn)$ operations follows from the fact that a matrix-vector product involving $H = AS^{-2}A^T$ can be obtained in $O(mn)$ operations given the matrix A and vector s. □

Note that while Theorem 3.1 bounds the number of steps of the PCG method required to obtain $\|\tilde{p}_y - p_y\|_H \leq \gamma_1\|p_y\|_H$, it is never necessary to evaluate the quantities involved in this condition. In particular, p_y is never known; if it was, the path following algorithm could simply use p_y in place of the approximation $\tilde{p}_y$.

4. Partial updating analysis. In this section we consider the "partial updating" procedure that is used to maintain the matrices $(W^k)^{-1}$ required in the PCG method of the previous section. Recall that $W^k = A(\tilde{S}^k)^{-2}A^T$, where for each $k \geq 0$,

$s = s^k$, $\tilde{s} = \tilde{s}^k$ satisfy (3.7). The analysis required here is quite standard and has been employed in many papers on partial updating interior point methods. Our exact proofs are based on the analysis of a projective partial updating method in [1, section 3], which itself is based on the original partial updating analysis of Karmarkar [5]. To begin, we give a precise statement of the update procedure that is mentioned in the algorithm in section 2. We assume that $\tilde{s}^0 = s^0$.

Update Procedure.

```
For $i = 1 : n$
    If $(1/\rho) \le (\tilde{s}_i^k / s_i^{k+1})^2 \le \rho$
    Then $\tilde{s}_i^{k+1} := \tilde{s}_i^k$
    Else $\tilde{s}_i^{k+1} := s_i^{k+1}$
Next $i$
```

We assume that the matrices $(W^k)^{-1}$ are explicitly available. As a result, each "update" $\tilde{s}_i^{k+1} \neq \tilde{s}_i^k$ requires a rank-1 change in $(W^k)^{-1}$, which can be performed in $O(m^2)$ operations. Alternatively, a suitable factorization of W^k can be maintained; see, for example, Shanno [13] for details on maintaining a Cholesky factorization. For each $k \ge 0$ and $i = 1, \dots, n$, define the "discrepancies"

$$\delta_i^k = \ln(\tilde{s}_i^k / s_i^k), \qquad \tilde{\delta}_i^k = \ln(\tilde{s}_i^k / s_i^{k+1}).$$

Then $|\delta_i^k| \le .5 \ln(\rho)$ for each i and k, by construction, and

$$\delta_i^{k+1} = \begin{cases} \tilde{\delta}_i^k & \text{if } |\tilde{\delta}_i^k| \le .5\ln(\rho), \\ 0 & \text{otherwise,} \end{cases}$$

the second case corresponding to an update of index i at the end of iteration k. Finally, note that

$$\tilde{\delta}_i^k - \delta_i^k = \ln(s_i^{k+1}/s_i^k). \tag{4.1}$$

Lemma 4.1 (see Karmarkar [5]). *If index i is updated at the end of iteration k_1, and is next updated at the end of iteration $k_2 > k_1$, then $\sum_{k=k_1+1}^{k_2} |\ln(s_i^{k+1}/s_i^k)| \ge .5 \ln \rho$.*

Proof. The proof is identical to that of [1, Lemma 3.1], using (4.1) in place of [1, (3.2)]. □

Proposition 4.2 (see [1, Lemma 3.3]). *If $0 < \gamma < 1$, $u \ge \gamma$, then $|\ln u| \le |1 - u|\,|\ln \gamma|/(1-\gamma)$.*

Theorem 4.3. *Assume that the modified short-step path following algorithm is initiated with $\tilde{s}^0 = s^0$ and that $\|(S^k)^{-1}\tilde{p}_s^k\| \le \bar{\gamma} < 1$ for all k. Then, if the algorithm is run for M iterations, and N is the total number of updates required on these iterations,*

$$N \le \frac{2M\sqrt{n}\,|\ln(1-\bar{\gamma})|}{\ln \rho}.$$

Proof. Let n_i denote the number of updates of index i on iterations $0, \dots, M-1$. Repeatedly applying Lemma 4.1, starting with $k_1 = -1$, obtains

$$n_i(.5 \ln \rho) \le \sum_{k=0}^{M-1} |\ln(s_i^{k+1}/s_i^k)|, \qquad i = 1, \dots, n. \tag{4.2}$$

Summing (4.2) over $i = 1, \dots, n$ then results in

$$N \ln \rho \le 2 \sum_{i=1}^{n} \sum_{k=0}^{M-1} |\ln(s_i^{k+1}/s_i^k)| = 2 \sum_{k=0}^{M-1} \sum_{i=1}^{n} |\ln(1 + (\tilde{p}_s^k)_i / s_i^k)|. \tag{4.3}$$

But $\|(S^k)^{-1}\tilde{p}_s^k\| \le \bar{\gamma} < 1$ implies that $(\tilde{p}_s^k)_i/s_i^k \ge -\bar{\gamma}$ for each i, so $1+(\tilde{p}_s^k)_i/s_i^k \ge 1-\bar{\gamma}$ for all i and k. Applying Proposition 4.2 with $\gamma = 1-\bar{\gamma}$, (4.3) implies that

$$N \ln\rho \le \frac{2|\ln(1-\bar{\gamma})|}{\bar{\gamma}} \sum_{k=0}^{M-1}\sum_{i=1}^{n} |(\tilde{p}_s^k)_i/s_i^k| \le 2M\sqrt{n}\,|\ln(1-\bar{\gamma})|,$$

where the final inequality uses $\|(S^k)^{-1}\tilde{p}_s^k\|_1 \le \sqrt{n}\|(S^k)^{-1}\tilde{p}_s^k\| \le \bar{\gamma}\sqrt{n}$. □

COROLLARY 4.4. *Under the assumptions of Theorem* 2.3, *if the algorithm of section* 2 *is applied to LD, and N is the total number of updates performed on all iterations, then $N = O(nL/\ln\rho)$.*

Proof. For each $k \ge 0$ we have

$$\|(S^k)^{-1}\tilde{p}_s^k\| = \|(S^k)^{-1}\tilde{p}_s(s^k,\mu^{k+1})\| \le (1+\gamma_1)\delta(s^k,\mu^{k+1}) < 1.2(.32) < .40,$$

where the first inequality uses (2.5) and the second uses (2.8). The assumptions of Theorem 4.3 are then satisfied with $\bar{\gamma} = .40$. In addition, from Theorem 2.3, the number of iterations required by the algorithm is $M = O(\sqrt{n}L)$. The result then follows immediately from Theorem 4.3. □

5. Overall complexity. From Theorems 2.3 and 3.1, and Corollary 4.4, we can easily obtain the following result for the overall complexity of the algorithm of section 2.

THEOREM 5.1. *Suppose that the algorithm of section* 2 *is applied to LD, under the assumptions of Theorem* 2.3. *Then the total number of arithmetic operations required by the algorithm is $O(nm^2 + (n^{1.5}m\rho + nm^2/\ln\rho)L)$.*

Proof. By Theorem 2.3 the algorithm requires $O(\sqrt{n}L)$ iterations. On each iteration the PCG method requires $O(nm\rho)$ operations, by Theorem 3.1, and the remaining work per iteration, exclusive of performing updates, is $O(nm)$. Thus the total nonupdating work is $O(n^{1.5}m\rho L)$. In addition, the algorithm requires $O(nm^2)$ operations to form $(A(S^0)^{-2}A^T)^{-1}$, and a total of $O(nm^2L/\ln\rho)$ operations to perform all subsequent updates, by Corollary 4.4. □

Note that if $\rho = \Theta(1)$, then the overall complexity given by Theorem 5.1 is $O((n^{1.5}m + nm^2)L) \le O(n^{1.5}m^{1.5}L) \le O(n^3L)$ operations, which is identical to that obtained for $O(\sqrt{n}L)$-iteration methods using partial updating; see, for example, [4] and [15]. However, we now show that ρ can be chosen so that the PCG algorithm has an overall complexity below $O(n^3L)$ operations.

COROLLARY 5.2. *Suppose that the algorithm of section* 2 *is applied to LD, under the assumptions of Theorem* 2.3, *with $\rho = \beta_1 m^{\beta_2}$, where β_1 and β_2 are absolute constants with $\beta_1 > 0$, $0 < \beta_2 < 1/2$. Then for $L = \Omega(\ln m)$, the total number of arithmetic operations required by the algorithm is*

$$O\bigg(\frac{n^{1.5}m^{1.5}}{\ln m}L\bigg) \le O\bigg(\frac{n^3}{\ln n}L\bigg).$$

Proof. Using the given form of ρ, we obtain

$$\begin{aligned} n^{1.5}m\rho + \frac{nm^2}{\ln\rho} &= \beta_1 n^{1.5}m^{1+\beta_2} + \frac{nm^2}{\ln(\beta_1)+\beta_2\ln(m)} \\ &= O\bigg(n^{1.5}m^{1+\beta_2} + \frac{nm^2}{\ln m}\bigg) \\ &= O\bigg(\frac{n^{1.5}m^{1.5}}{\ln m}\bigg), \end{aligned}$$

where the final inequality uses $\beta_2 < 1/2$ and $m \le n$. The corollary then follows from Theorem 5.1 and the assumption that $L = \Omega(\ln m)$. $\square$

Note that the total complexity given in Corollary 5.2 can also be considered to include the $O(n^2 m)$ operations necessary to obtain an exact optimal solution to LP or LD or both from the approximately optimal solutions output by the algorithm, using a standard "rounding" procedure.

As described following Theorem 2.3, complexity results like those given here in terms of L can alternatively be given in terms of the termination tolerance ϵ. For example, it is easy to show that under the assumptions of Corollary 5.2, but where the assumption on μ^0 in Theorem 2.3 is replaced by $n\mu^0 = O(1/\epsilon)$, the total number of operations required by the algorithm to obtain $(x^k)^T s^k \le \epsilon$ is

$$O\left(n^{1.5} m^{1.5} \left[1 + \frac{|\ln \epsilon|}{\ln m}\right]\right).$$

In addition to Corollary 5.2, which holds for any m and n, it is interesting to consider how ρ may be chosen so as to obtain the lowest possible complexity bound for a given m and n. Differentiating the bound of Theorem 5.1, this "optimal" ρ satisfies

$$\rho(\ln \rho)^2 = \frac{m}{\sqrt{n}}. \tag{5.1}$$

We do not have an analytic solution of (5.1), but we consider a few representative cases:

1. If $m = \Theta(n)$, then the optimal ρ is slightly less than $\Theta(\sqrt{n})$. For example, using $\rho = \Theta(\sqrt{n}/\ln n)$ gives an overall complexity bound of $O([n^3/\ln n]L)$ operations for the PCG algorithm, as in Corollary 5.2.

2. If $m = \Theta(\sqrt{n})$, then the optimal ρ is $\Theta(1)$, and using this ρ in the PCG algorithm obtains an overall complexity of $O(n^2 L)$ operations.

3. If $m = o(\sqrt{n})$, then the optimal ρ is of the form $1 + \Theta(m^{1/2}/n^{1/4})$, and the overall complexity of the PCG algorithm using this ρ is $O(n^{1.5} m L) = o(n^2 L)$ operations.

As described in the introduction, the PCG algorithm was originally devised by Nesterov and Nemirovskii [9], [10] so as to obtain a complexity below $O(n^3 L)$ operations using fast matrix multiplication. It is worthwhile to note that in the algorithm described here, as in Karmarkar's original partial updating method, there are *no* matrix operations, other than matrix-vector products, following the initial factorization or inversion of $A(S^0)^{-2}A^T$. Thus, to obtain a complexity improvement using fast matrix multiplication, it is necessary to replace some of the algorithm's rank-1 updating with higher rank matrix operations. There are basically two ways to accomplish this:

1. Replace a sequence of rank-1 updates on a single iteration with a single block update of $(W^k)^{-1}$, using the Sherman–Morrison–Woodbury formula.

2. Periodically perform additional "refresh" steps, where some or all of the $\tilde{s}_i^k$ are reset to the correct values s_i^k, and use block updating, or full recomputation of $(W^k)^{-1}$, to perform these refreshes.

See Nesterov and Nemirovskii [9], [10] for details. Finally, it is worthwhile to note that results similar to those obtained here could be proved using iterative techniques other than the PCG method to approximately solve the Newton equations (3.1) that arise on each iteration. For example, Nesterov and Nemirovskii [10] consider the use of the steepest descent method, and also the "optimal" method for smooth convex programming. (See Nesterov [8] for more details on the optimal method.) Either of these methods, combined with the preconditioning strategy described in section 3, could be used to obtain algorithms for LP with overall complexities of $O([n^3/\ln n]L)$ operations.

Acknowledgment. I would like to thank Yuri Nesterov for several conversations on the topic of this paper.

REFERENCES

[1] K. M. Anstreicher, *A standard form variant, and safeguarded linesearch, for the modified Karmarkar algorithm*, Math. Programming, 47 (1990), pp. 337–351.

[2] D. Coppersmith and S. Winograd, *Matrix multiplication via arithmetic progressions*, in Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1986, pp. 1–6.

[3] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, New York, 1981.

[4] C. C. Gonzaga, *An algorithm for solving linear programming problems in $O(n^3L)$ operations*, in Progress in Mathematical Programming, N. Megiddo, ed., Springer-Verlag, New York, 1987.

[5] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[6] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed., Addison-Wesley, New York, 1984.

[7] R. D. C. Monteiro and I. Adler, *Interior path following primal-dual algorithms. Part* I: *Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[8] Y. Nesterov, *Introductory Lectures on Convex Programming, Vol.* I: *Basic Course*, Center for Operations Research and Econometrics, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 1996.

[9] Y. Nesterov and A. Nemirovskii, *Acceleration and parallelization of the path-following interior point method for a linearly constrained quadratic programming problem*, SIAM J. Optim., 1 (1991), pp. 548–564.

[10] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, 1994.

[11] J. Renegar, *A polynomial-time algorithm, based on Newton's method, for linear programming*, Math. Programming, 40 (1988), pp. 59–93.

[12] C. Roos and J.-Ph. Vial, *A polynomial method of approximate centers for linear programming*, Math. Programming, 54 (1992), pp. 295–305.

[13] D. F. Shanno, *Computing Karmarkar projections quickly*, Math. Programming, 41 (1988), pp. 61–71.

[14] P. M. Vaidya, *Speeding-up linear programming using fast matrix multiplication*, in Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science, 1989, pp. 338–343.

[15] P. M. Vaidya, *An algorithm for linear programming which requires $O(((m+n)n^2+(m+n)^{1.5}n)L)$ arithmetic operations*, Math. Programming, 47 (1990), pp. 175–201.

SIAM J. OPTIM.
Vol. 9, No. 4, pp. 813–832

FREE MATERIAL DESIGN VIA SEMIDEFINITE PROGRAMMING: THE MULTILOAD CASE WITH CONTACT CONDITIONS*

A. BEN-TAL[†], M. KOČVARA[‡], A. NEMIROVSKI[†], AND J. ZOWE[‡]

To John Dennis on the occasion of his 60th birthday.

Abstract. Free material design deals with the question of finding the stiffest structure with respect to one or more given loads which can be made when both the distribution of material and the material itself can freely vary. The case of one single load has been discussed in several recent papers, and an efficient numerical approach was presented in [M. Kocvara, M. Zibulevsky, and J. Zow, *RAIRO Modél. Math. Anal. Numér.* 32 (1998), pp. 255–281]. We attack here the multiload situation (understood in the worst-case sense), which is of much more interest for applications but also significantly more challenging from both the theoretical and the numerical points of view. After a series of transformation steps we reach a problem formulation for which we can prove existence of a solution; a suitable discretization leads to a semidefinite programming problem for which modern polynomial time algorithms of interior point type are available. A number of numerical examples demonstrates the efficiency of our approach.

Key words. structural optimization, material optimization, topology optimization, semidefinite programming

AMS subject classifications. 73K40, 90C90, 90C25

PII. S1052623497327994

1. Introduction.

One of the basic problems of structural engineering is to design the stiffest structure of a given volume, occupying some fixed domain $\Omega \subset \mathbb{R}^{dim}$ (dim = 2,3) with boundary Γ, which is capable of carrying a given set of external loads. The desired optimal structure is considered to be a continuum elastic body, and the design variables are the *material properties* which may vary from point to point. Thus the aim is to optimize not only the distribution of material but also the material properties themselves, and we are looking for the ultimately best structure among all possible elastic continua, in a framework of what is now usually referred to as "free material design."

Optimization of structures is traditionally performed through the variation of sizing variables (e.g., thicknesses of bars in a truss) and shape variables (e.g., splines defining the boundary of a body). With the appearance of composites and other advanced man-made materials it has been natural to extend this variation to the material choice itself. The basic problem setting of "free material design" that we will deal with goes back to the work of Bendsøe et al. [5] and Ringertz [14], where representing material properties as elements of the unrestricted set of positive semidefinite constitutive tensors with the trace of the stiffness tensor as a measure of resource ("cost") was suggested. In mathematical language this leads to an optimization prob-

*Received by the editors September 29, 1997; accepted for publication (in revised form) August 17, 1998; published electronically September 24, 1999. This work was supported by GIF contract I0455-214.06/95. The research of the second and fourth authors was supported by BMBF project 03ZO7BAY. The research of the fourth author was supported by grant A1075707 from the Czech Academy of Sciences.

http://www.siam.org/journals/siopt/9-4/32799.html

[†]Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, 32000 Haifa, Israel (morbt@ie.technion.ac.il, nemirovs@ie.technion.ac.il).

[‡]Institute of Applied Mathematics, University of Erlangen, Martensstr. 3, D-91058 Erlangen, Germany (kocvara@am.uni-erlangen.de, zowe@am.uni-erlangen.de). M. Kočvara is on leave from the Czech Academy of Sciences.

lem with an objective function (stiffness) which is the result of an inner optimization. More precisely, one minimizes (with respect to material properties) the compliance (a certain global measure of the stiffness of the structure), where the compliance itself is the outcome of a lower optimization level (minimization of potential energy). The resulting minimax problem looks rather complicated: in two (three) space dimensions, the design variables are the 6 (21) defining elements of the symmetric elasticity tensor and these variables are allowed to vary pointwise throughout the structure. The case of single-load design (SLD) was treated in [5] and, emphasizing the numerical aspect, in [16]. There it is shown that one can analytically reduce the problem to one with only a single design variable at each point (in addition to the displacement vector), namely, the *trace* of the elasticity tensor. The elements of the optimal tensor itself are then fully recoverable from the optimal trace and the related displacements. A finite element discretization of the above reduced problem leads to a mathematical programming formulation, which is identical in form to maximal stiffness optimization problems for trusses, and the very efficient interior point–based software developed for truss problems (see, e.g., [1, 9, 10]) can be used almost immediately in this framework of material optimization. In [16] this computational approach to SLD is discussed in detail, and a number of examples demonstrate its efficiency.

For most applications, however, the assumption of a single acting load is too restrictive and may lead to a structure which is highly unstable with respect to small load perturbations. Hence one is interested in a structure which is stable with respect to a whole scenario of independent loads and which is the stiffest one in the worst-case sense. This multiload feature complicates the situation substantially since it leads to a blow-up in the dimension, and further, the above-mentioned reduction process leads to an integral over an eigenvalue problem which is hard to eliminate when discretizing for a numerical approach. All this excludes a direct transfer of the tools, which are successful in the SLD case, to the multiload situation. *Multiload design* (MLD) requires essentially new tools. Only some first steps in the direction of a theoretical treatment of the MLD can be found in the literature [2]; we are not aware of reports on numerical approaches. Our paper tries to fill this gap.

2. Problem formulation and existence theorem. We study the optimization of the design of a *continuum* structure that is loaded by multiple independent forces. In order to deal with the problem in a very general form, we consider the *distribution of the material in space* as well as the *material properties at each point* as design variables. The idea of treating the material itself as a function of the space variable goes back to [5, 14] and also has been studied in other contexts in [3, 4, 6]. This present text develops in this framework a theory for the MLD case with additional contact conditions. We start from the infinite-dimensional problem setting, prove existence of a solution after a reformulation of the problem, and, after discretization, reach a finite-dimensional formulation expressed as a *semidefinite program* and, as such, accessible to modern numerical interior point methods.

For an easier understanding of the physical background we begin with a sketch of the single-load model. Let $\Omega \subset \mathbb{R}^{dim}$, dim $= 2, 3$, be a bounded domain (the elastic body) with a Lipschitz boundary Γ. We use the standard notation $[H^1(\Omega)]^{dim}$ and $[H_0^1(\Omega)]^{dim}$ for Sobolev spaces of functions $v : \Omega \to \mathbb{R}^{dim}$. By $u(x) = (u_1(x), \ldots, u_{dim}(x))$ with $u \in [H^1(\Omega)]^{dim}$ (in short, $u \in H^1(\Omega)$) we denote the *displacement vector* at point x of the body under load. Further

$$e_{ij}(u(x)) = \frac{1}{2}\left(\frac{\partial u_i(x)}{\partial x_j} + \frac{\partial u_j(x)}{\partial x_i}\right) \qquad \text{for } i, j = 1, \ldots, \text{dim}$$

denotes the *(small-)strain tensor*, and $\sigma_{ij}(x), i,j = 1,\ldots,\text{dim}$, the *stress tensor.*

We assume that our system is governed by linear Hooke's law; i.e., the stress is a linear function of the strain

$$\sigma_{ij}(x) = E_{ijkl}(x)e_{kl}(u(x)) \quad \text{(in tensor notation)}, \tag{2.1}$$

where $E(x)$ is the so-called (plain-stress) *elasticity tensor* of order 4; this tensor characterizes the behavior of material at point x. To unburden the notation we will often skip the variable x in u, e, E, etc. The strain and stress tensors are symmetric (e.g., $e_{ij} = e_{ji}$) and E also is symmetric in the following sense:

$$E_{ijkl} = E_{jikl} = E_{ijlk} = E_{klij} \qquad \text{for } i,j,k,l = 1,\ldots,\text{dim}.$$

These symmetries allow us to avoid the tensor notation, which is not commonly used in the optimization community, and interpret the 2-tensors e and σ as vectors

$$e = (e_{11}, e_{22}, \sqrt{2}e_{12})^T \in \mathbb{R}^3, \qquad \sigma = (\sigma_{11}, \sigma_{22}, \sqrt{2}\sigma_{12})^T \in \mathbb{R}^3$$

for dim = 2 and analogously as vectors in $\mathbb{R}^6$ for dim = 3. Correspondingly, the 4-tensor E can be written as a symmetric 3×3 matrix,

$$E = \begin{pmatrix} E_{1111} & E_{1122} & \sqrt{2}E_{1112} \\ & E_{2222} & \sqrt{2}E_{2212} \\ \text{sym.} & & 2E_{1212} \end{pmatrix}, \tag{2.2}$$

for dim = 2 and as a symmetric 6×6 matrix for dim = 3. In this notation, (2.1) reads as

$$\sigma(x) = E(x)e(u(x)).$$

Since E will be understood as a matrix in our paper, we will use double indices for the elements of E; the correspondence between E_{ij} and the tensor components E_{ijkl} is clear from (2.2). To allow switches from material to no-material, it is natural to work with ($d = 3$ or 6)

$$E \in [L^\infty(\Omega)]^{d\times d} \qquad (\text{in short, } E \in L^\infty(\Omega)).$$

For a consistent notation, we will always use $d = 3$ in connection with dim = 2 and $d = 6$ when dim = 3.

We consider a partitioning of the boundary Γ into two parts: $\Gamma = \overline{\Gamma}_u \cup \overline{\Gamma}_f$, where Γ_u and Γ_f are open in Γ, and $\Gamma_u \cap \Gamma_f = \emptyset$. Further, we put

$$\mathcal{H} = \{u \in [H^1(\Omega)]^{dim} \mid G(s)u(s) = 0 \text{ for } s \in \Gamma_u\},$$

$G(s)$ being a measurable matrix-valued function defining the *boundary conditions*, so that $[H_0^1(\Omega)]^{dim} \subset \mathcal{H} \subset [H^1(\Omega)]^{dim}$; we assume that the admissible displacement fields belong to $\mathcal{H}$.

The boundary conditions on Γ_f are specified by the surface traction ("external load")

$$f \in [L^2(\Gamma_f)]^{dim} \qquad (\text{in short,} \quad f \in L^2(\Gamma_f)).$$

To allow for more general situations, we require that u stays within a (nonempty) closed convex set $U \subset \mathcal{H}$. This U can be given, e.g., by *unilateral contact condition* (for details, see [10, 13]).

For given elasticity matrix E and acting load f, the potential energy as a function of the displacement $u \in U$ is given by

$$-\frac{1}{2}\int_{\Omega} \langle Ee(u), e(u)\rangle \, dx + F(u), \tag{2.3}$$

where we have put

$$F(u) := \int_{\Gamma_f} f \cdot u \, dx. \tag{2.4}$$

We recall once more that E, u, and f in (2.3), (2.4) are functions of x, which is omitted only to economize the notation. The system is in equilibrium (outer and inner forces balance each other) for u, which maximizes the concave term (2.3), i.e., u which solves

$$\sup_{u \in U} \left\{ -\frac{1}{2}\int_{\Omega} \langle Ee(u), e(u)\rangle \, dx + F(u) \right\}. \tag{2.5}$$

Nature always tries to reach the equilibrium (2.5). The supremum in (2.5) is equal to what engineers often call *compliance* of the system. It is a measure of the stiffness of the structure: the less the compliance, the more rigid the structure with respect to f. It is now the interest of the designer to choose under physical and economical constraints the material function $E \in Ł^{\infty}(\Omega)$ such that the "sup" in (2.5) becomes as small as possible. Physics tells us that $E(x)$ has to to be a symmetric and positive semidefinite matrix almost everywhere (a.e.) on Ω, which we write as

$$E = E^T \succeq 0 \qquad \text{a.e. in } \Omega. \tag{2.6}$$

The diagonal elements of $E(x)$ measure the stiffness of the material at x in the coordinate directions. Hence it makes sense to use as resource (cost) constraint the *trace* of E (with $d = 3$ or 6 according to dim $= 2$ or 3),

$$\operatorname{tr}(E(x)) := \sum_{i=1}^{d} E_{ii}(x), \tag{2.7}$$

and to require, with some given positive α,

$$\int_{\Omega} \operatorname{tr}(E(x)) \, dx \le \alpha. \tag{2.8}$$

The trace is invariant under orthogonal transformations; hence our constraint does not depend on the coordinate system.

Further, to exclude singularities (e.g., on the boundary Γ_f) we demand that, with some fixed $r^+, r^- \in L^{\infty}(\Omega)$, $0 \le r^- < r^+$,

$$r^- \le \operatorname{tr}(E) \le r^+ \qquad \text{a.e. in } \Omega. \tag{2.9}$$

It is convenient to summarize the feasible design functions in a set

$$\mathcal{E} := \left\{ E \in L^{\infty}(\Omega) \mid \begin{array}{l} E \text{ is of form (2.2) and} \\ \text{satisfies (2.6), (2.8), and (2.9)} \end{array} \right\}. \tag{2.10}$$

With this definition, the SLD problem becomes

$$\inf_{E \in \mathcal{E}} \sup_{u \in U} \left\{ -\frac{1}{2}\int_{\Omega} \langle Ee(u), e(u)\rangle \, dx + F(u) \right\}. \tag{2.11}$$

Obviously, a minimizing E in (2.11) will be optimal only for the *one* considered load f and might be extremely unstable (may even collapse) under loads other than f (even of small magnitude). Hence a more realistic approach requires us to look for a structure which can withstand a whole collection of independent loads $f^1, \dots, f^L$ from $L^2(\Gamma_f)$, acting at different times; further, the design should be the "best possible" one. In an engineering context, the worst-case aspect makes most sense. This leads to the following MLD problem, in which we seek the design function E which yields the smallest possible worst-case compliance:

$$\inf_{E\in\mathcal{E}} \sup_{\ell=1,\dots,L} \sup_{u^\ell\in U^\ell} \left\{ -\frac{1}{2}\int_\Omega \langle Ee(u^\ell), e(u^\ell)\rangle\, dx + F^\ell(u^\ell) \right\}; \tag{2.12}$$

here we have put, in accordance with (2.4),

$$F^\ell(u) := \int_{\Gamma_f} f^\ell \cdot u\, dx \qquad \text{for } \ell = 1,\dots,L. \tag{2.13}$$

Further, the sets U^ℓ in (2.12) allow individual contact conditions for the loads f^ℓ; hence we can work in (2.12) with different rigid obstacles and indeed we solve a coupled *multiple-load* and *multiple-obstacle* problem. One could even go one step further and consider different partitioning of Γ for each load-case. However, in technical practice the (noncontact) boundary conditions are usually the same for all load-cases and thus we assume $f^\ell \in \Gamma_f$ for all $\ell = 1,\dots,L$.

To be more precise for the numerical part later we assume that the sets U^ℓ in (2.12) can be written in the form

$$U^\ell := \left\{ u \in H^1(\Omega) \mid g^\ell(u) \le \delta^\ell \right\} \tag{2.14}$$

with linear functions g^ℓ and suitable right-hand sides δ^ℓ for $\ell = 1,\dots,L$. Further, to exclude trivial situations, let

$$U^\ell \neq \emptyset \qquad \text{for } \ell = 1,\dots,L.$$

All our forthcoming efforts aim at finding an efficient analytical and computational way to solve the MLD (2.12). We start with two steps which convert (2.12) to an "equivalent" but more easily accessible problem. First let us eliminate the discrete inner "$\sup_{\ell=1,\dots,L}$" in (2.12). With a *weight vector* λ for the loads, which runs over the unit simplex

$$\Lambda := \left\{ \lambda \in \mathbb{R}^L \mid \sum_{\ell=1}^{L} \lambda_\ell = 1,\ \lambda_\ell \ge 0 \text{ for } \ell = 1,\dots,L \right\}, \tag{2.15}$$

we get from a standard linear programming argument the following equivalent representation of (2.12):

$$\inf_{E\in\mathcal{E}} \sup_{\substack{\lambda\in\Lambda \\ (u^1,\dots,u^L)\in U^1\times\cdots\times U^\ell}} \sum_{\ell=1}^{L} \left\{ -\frac{1}{2}\int_\Omega \lambda_\ell \langle Ee(u^\ell), e(u^\ell)\rangle\, dx + \lambda_\ell F^\ell(u^\ell) \right\}. \tag{2.16}$$

The objective function in (2.16) is linear (and thus convex) in the inf-variable E; it is, however, not concave in the sup-argument $(u^1,\dots,u^\ell;\lambda)$. This is in contrast to the SLD case, where λ reduces to 1 and (2.16) specializes to (2.11), which is convex-concave in (E, u). This convex-concave feature of SLD allows one to use convex analysis and to prove an existence result for (2.11). Further, after applying a minimax switch and after discretizing, one reaches a mathematical programming formulation for SLD which is of extremely simple structure (linear objective and quadratic constraints) and which is open to powerful modern interior point methods; see [16]. The loss of the convex-concave character in the MLD (2.16) excludes a direct transfer of this approach to the MLD case. Here we use a trick and show that after a simple change of variables we reach a convex-concave formulation for this case also.

We begin by noting that the inf-sup value in (2.16) remains the same when restricting λ to the half-open set

$$\Lambda_0 := \{\lambda \in \Lambda \mid \lambda_\ell > 0 \text{ for } \ell = 1, \dots, L\} \tag{2.17}$$

and passing from the variable $(u^1, \dots, u^L; \lambda)$ to

$$(v^1 := \lambda_1 u^1, \dots, v^L := \lambda_L u^L; \lambda).$$

This step converts (2.12)–(2.16) to

$$\inf_{E\in\mathcal{E}} \sup_{(\boldsymbol{v},\lambda)\in\mathcal{V}} \sum_{\ell=1}^{L} \left\{ -\frac{1}{2} \int_\Omega \lambda_\ell^{-1} \langle E e(v^\ell), e(v^\ell)\rangle \, dx + F^\ell(v^\ell) \right\}, \tag{2.18}$$

where we have put $\boldsymbol{v} := (v^1, \dots, v^L)$ and

$$\mathcal{V} := \left\{(\boldsymbol{v}; \lambda) \mid \lambda \in \Lambda_0, g^\ell(v^\ell) - \lambda_\ell \delta^\ell \le 0 \text{ for } \ell = 1, \dots, L\right\}$$

with g^ℓ and δ^ℓ from (2.13). $\mathcal{V}$ is again a convex set. Further—and this is the purpose of this substitution—the objective function in (2.18),

$$\mathcal{F}(E; (\boldsymbol{v}; \lambda)) := \sum_{\ell=1}^{L} \left\{ -\frac{1}{2} \int_\Omega \lambda_\ell^{-1} \langle E e(v^\ell), e(v^\ell)\rangle \, dx + F^\ell(v^\ell) \right\}, \tag{2.19}$$

is now concave in $(\boldsymbol{v}, \lambda) = (v^1, \dots, v^L; \lambda) \in \mathcal{V}$; this follows easily from the concavity of $-x^2/y$ in $(x, y) \in \mathbb{R} \times \mathbb{R}_+ \setminus \{0\}$. Since, as before, $\mathcal{F}$ is linear (and thus convex) in E, our convex-concave inf-sup problem (2.18) is open to the machinery of convex analysis. From a theorem due to Moreau [11] we get the following existence result.

Theorem 2.1 (existence of an optimal design tensor for MLD). *There exists $E^* \in \mathcal{E}$ such that*

$$\sup_{(\boldsymbol{v};\lambda)\in\mathcal{V}} \mathcal{F}(E^*; (\boldsymbol{v}; \lambda)) = \min_{E\in\mathcal{E}} \sup_{(\boldsymbol{v};\lambda)\in\mathcal{V}} \mathcal{F}(E; (\boldsymbol{v}; \lambda)).$$

Further,

$$\inf_{E\in\mathcal{E}} \sup_{(\boldsymbol{v};\lambda)\in\mathcal{V}} \mathcal{F}(E; (\boldsymbol{v}; \lambda)) = \sup_{(\boldsymbol{v};\lambda)\in\mathcal{V}} \inf_{E\in\mathcal{E}} \mathcal{F}(E; (\boldsymbol{v}; \lambda)).$$

Proof. The claim follows from [11] if we can guarantee that

(i) $\mathcal{V}$ is a convex set;
(ii) $\mathcal{F}(E; \cdot)$ is concave for fixed $E \in \mathcal{E}$;
(iii) $\mathcal{E} \subset L^\infty(\Omega)$ is convex and weak*-compact;

(iv) $\mathcal{F}(\cdot;(\boldsymbol{v};\lambda))$ is convex and lower semicontinuous on $\mathcal{E}$ (equipped with the weak*-topology of $L^\infty(\Omega)$) for fixed $(\boldsymbol{v};\lambda) \in \mathcal{V}$.

Conditions (i) and (ii) were already discussed above, and the convexity in (iii) and (iv) is obvious. The limit E of a sequence of elements $E_n \in \mathcal{E}$ again satisfies (2.6), (2.8), and (2.9); hence $\mathcal{E}$ is closed in $L^\infty(\Omega)$. From (2.6) and (2.9) one easily deduces that $\mathcal{E}$ lies in a norm ball of $L^\infty(\Omega)$. Thus the weak*-compactness of $\mathcal{E}$ follows from the Alaoglu's theorem (see, e.g., [15, Theorem III.10.2]. The function $\mathcal{F}(\cdot;(\boldsymbol{v};\lambda))$ is linear in E (for fixed $(\boldsymbol{v};\lambda)$) and obviously continuous on $\mathcal{E}$ (as a subset of $L^\infty(\Omega)$). The continuity in the weak*-topology follows from the very definition of this topology. □

Note that (2.12), (2.16), and (2.18) yield the same objective values but that in (2.18) we work with a restricted domain of definition (Λ replaced by Λ_0). Obviously, we can extend Λ_0 in (2.18) to Λ for the price of working with an extended-valued variant of $\mathcal{F}$. We avoid these technicalities here since it is the design function E we are really interested in, and for such E we use an existence result with Theorem 2.1.

3. Discretization and semidefinite reformulation.

Given the existence of an optimal elasticity matrix E^* for (2.18), we ask how to "compute" this E^*. The results of this section supply the key to this question; it is shown that after a finite element discretization of (2.18), the problem can be reduced to a *semidefinite program*, for which efficient computational tools are available.

3.1. The discretized problem.

To simplify the notation, we use the same symbols for the discrete objects (vectors) as for the "continuum" ones (functions). Assume that Ω is partitioned into M quadrilateral elements Ω_m of volumes ω_m. Let N be the number of nodes (vertices of the elements). Assume that E is approximated by a function that is constant on each element Ω_m; i.e., it is fully characterized by a collection $E = (E_1, \dots, E_M)$ of $d \times d$ matrices E_m—the values of E on the elements. The feasible set $\mathcal{E}$ is replaced by its discrete counterpart

$$\mathcal{E} := \left\{ E \in \mathbb{R}^{d\times dM} \;\middle|\; \begin{array}{l} E_m = E_m^T \succeq 0 \text{ and } r_m^- \le \operatorname{tr}(E_m) \le r_m^+ \text{ for } m = 1,\dots,M, \\ \sum\limits_{m=1}^{M} \operatorname{tr}(E_m)\omega_m \le \alpha \end{array} \right\}.$$

Further, assume that the displacement vector u^ℓ corresponding to the load-case ℓ is approximated by a continuous function that is tri/bilinear (linear in each coordinate) on every element. Such a function can be written as

$$u^\ell(x) = \sum_{n=1}^{N} u_n^\ell \vartheta_n(x),$$

where u_n^ℓ is the value of u^ℓ at the nth node and ϑ_n is the basis function associated with nth node. (For details, see [7].) Recall that, at each node, the displacement has dim components, hence $u \in \mathbb{R}^D$, $D \le \dim \cdot N$. (D could be less than $\dim \cdot N$ because of boundary conditions which enforce the displacements of certain nodes to lie in given subspaces of $\mathbb{R}^{dim}$.)

Further, we define the discrete version of the set U^ℓ of admissible displacements. We assume that the set is given by unilateral contact conditions. The introduction of these conditions is quite technical, and the details can be found in [10]. Here we introduce only vectors $\delta^\ell \in \mathbb{R}^r$ (representing the gaps between the contact surfaces and the rigid obstacles) and $r \times M$ matrices C^ℓ (defining the nodes of the contact surface and the direction to the obstacle). The set of admissible displacements for the discretized problem takes the form

$$U^\ell := \{u^\ell \in \mathbb{R}^D \mid C^\ell u^\ell \leq \delta^\ell\}. \tag{3.1}$$

For basis functions $\vartheta_n, n = 1, \ldots, N$, we define the matrix (which are again functions of x)

$$B_n = \begin{pmatrix} \frac{\partial \vartheta_n}{\partial x_1} & 0 \\ 0 & \frac{\partial \vartheta_n}{\partial x_2} \\ \frac{1}{2}\frac{\partial \vartheta_n}{\partial x_2} & \frac{1}{2}\frac{\partial \vartheta_n}{\partial x_1} \end{pmatrix}$$

for dim = 2 and an analogous matrix for dim = 3. Now, for an element Ω_m, let $\mathcal{D}_m$ be an index set of nodes belonging to this element. The value of the approximate strain tensor e on element Ω_m is then (adding the variable x as a subscript)

$$e_x(u^\ell) = \sum_{n \in \mathcal{D}_m} B_n(x) u_n^\ell \qquad \text{on } \Omega_m;$$

recall that u_n^ℓ has dim components.

Finally, the discrete version of the linear functional $F^\ell(u^\ell)$ is $(f^\ell)^T u^\ell$ with $f^\ell \in \mathbb{R}^D$, the load discretized in a standard way by means the basis functions ϑ_n.

Analogously to $\mathcal{E}$, define

$$\mathcal{E}_h := \Bigg\{ E = \{E_m\}_{m=1}^M \mid E_m \in \Sigma_+^d,\ m = 1, \ldots, M; \\ [0 \leq] \quad r_m^- \leq \operatorname{tr}(E_m) \leq r_m^+ \quad [< \infty],\ m = 1, \ldots, M; \\ \sum_{m=1}^M \omega_m \operatorname{tr}(E_m) \leq \alpha \Bigg\}, \tag{3.2}$$

where Σ^p denotes the space of symmetric $p \times p$ matrices and Σ_+^p is the cone of positive semidefinite matrices from Σ^p. As a discretized version of the original problem we thus obtain

$$\min_{E = \{E_m\}_{m=1}^M \in \mathcal{E}_h} \phi(E), \\ \phi(E) := \sup_{\ell = 1, \ldots, L} \sup_{u \in U^\ell} \left[-\sum_{m=1}^M \operatorname{tr}\left(E_m \int_{\Omega_m} e_x(u) e_x^T(u) dx \right) + 2(f^\ell)^T u \right]. \tag{3.3}$$

Now, for each element Ω_m there exists a finite set of points x_{ms} and positive weights χ_{ms}^2, $s = 1, \ldots, S$, such that

$$\int_{\Omega_m} e_x(u) e_x^T(u) dx = \sum_{s=1}^S \chi_{ms}^2 e_{x_{ms}}(u) e_{x_{ms}}^T(u)$$

for all $u \in \mathbb{R}^D$; e.g., one can take $S = 4$ for dim = 2, linear $B_n(\cdot)$, and rectangular Ω_m.

Let us define linear matrix-valued functions

$$\zeta_m(u) = \omega_m^{-1/2} [\chi_{m1} e_{x_{m1}}(u); \chi_{m2} e_{x_{m2}}(u); \ldots; \chi_{mS} e_{x_{mS}}(u)],\ m = 1, \ldots, M,$$

taking values in the space of $d \times S$ matrices; then the objective function in (3.3) can be rewritten equivalently as

$$\phi(E) = \sup_{\ell=1,\dots,L} \sup_{u \in U^\ell} \left[-\sum_{m=1}^{M} \omega_m \mathrm{tr}(E_m \zeta_m(u)\zeta_m^T(u)) + 2(f^\ell)^T u \right].$$

From now on we make the following assumptions:

(A) The linear inequalities defining the polyhedral sets U^ℓ, $\ell = 1, \dots, L$, satisfy the Slater condition: for every ℓ, there exists u_0^ℓ such that $C^\ell u_0^\ell < \delta^\ell$.

(B) The mapping $u \mapsto \{\zeta_m(u)\}_{m=1}^M$ has trivial kernel on $\mathbb{R}^D$. (This is actually the assumption which excludes rigid body motion of the construction.)

(C) $r_m^- < r_m^+$, $m = 1, \dots, M$, and $\sum_{m=1}^M r_m^- \omega_m < \alpha$.

3.2. The main results. We formulate two main results related to the discretized problem (3.3). (For proofs, see section 6.)

THEOREM 3.1. *Under assumptions* A, B, *and* C, *the semidefinite program*

$$\begin{array}{rcl}
 & \textit{maximize} & \\
\psi(v,\nu,\rho^+,\rho^-) & = & -\alpha\nu + 2\sum_{\ell=1}^{L}(f^\ell)^T v^\ell + \sum_{m=1}^{M}(s_m^-\rho_m^- - s_m^+\rho_m^+), \\
 & \textit{subject to} & \\
\mathcal{A}_m(v,\nu,\rho^+,\rho^-) & := & \begin{pmatrix} (\nu+\rho_m^+-\rho_m^-)I_d & \zeta_m(v^1) & \zeta_m(v^2) & \cdots & \zeta_m(v^L) \\ \zeta_m^T(v^1) & \lambda_1 I_S & & & \\ \zeta_m^T(v^2) & & \lambda_2 I_S & & \\ \vdots & & & \ddots & \\ \zeta_m^T(v^L) & & & & \lambda_L I_S \end{pmatrix} \\
 & \succeq & 0, \qquad m = 1,\dots,M, \\
\mathrm{Diag}(\lambda_\ell\delta^\ell - C^\ell v^\ell) & \succeq & 0, \qquad \ell = 1,\dots,k, \\
 & & \\
\mathrm{Diag}(\rho^+) & \succeq & 0, \\
\mathrm{Diag}(\rho^-) & \succeq & 0, \\
\nu & \geq & 0, \\
\sum_{\ell=1}^{L}\lambda_\ell & = & 1.
\end{array} \tag{3.4}$$

(C^ℓ, δ^ℓ *are given by* (3.1)*) with the design variables*

$$(\boldsymbol{v};\lambda) = (v^1,\dots,v^L;\lambda) \in (\mathbb{R}^D)^L \times \mathbb{R}^L, \quad \rho^\pm \in \mathbb{R}^M, \quad \nu \in \mathbb{R}$$

and constants

$$s_m^\pm = \omega_m r_m^\pm$$

is dual to the problem of interest (3.3) *in the sense that the optimal value* ϕ^* *of* (3.3) *is equal to the optimal value* ψ^* *of* (3.4).

Theorem 3.1 deals with optimal values of (3.3), (3.4) but does not answer the crucial question of how to recover a (nearly) optimal solution to the original (primal) problem from a (nearly) optimal solution to its dual problem. In order to derive such a recovering routine, recall the notion of a central approximate solution to a semidefinite program. Problem (3.4) is of the generic form

$$\max\{c^T x \mid \mathcal{A}x \succeq 0, e^T x = 1\}, \tag{SDP}$$

where the design vector x varies in $\mathbb{R}^n$ and $x \mapsto \mathcal{A}x$ is an affine mapping of $\mathbb{R}^n$ into space Σ of symmetric matrices of a given block-diagonal structure. Assuming the problem (SDP) to be strictly feasible (there exists x with $e^T x = 1$ and positive definite $\mathcal{A}x$), one can equip the relative interior $\mathcal{X}'$ of a feasible set $\mathcal{X}$ of the problem with the standard barrier

$$\boldsymbol{B}(x) = -\ln \operatorname{Det}(\mathcal{A}x).$$

Now let $t > 0$. A point $x(t) \in \mathcal{X}'$ is called *central approximate solution* to (SDP) *associated with the value t of the penalty parameter* if $x(t)$ minimizes the aggregate

$$-tc^T x + \boldsymbol{B}(x) \tag{3.5}$$

over $\mathcal{X}'$.

We are about to establish the following theorem.

Theorem 3.2. *Under assumptions* A, B, *and* C *we have the following:*

(i) *Central approximate solutions to* (3.4) *exist for every value* $t > 0$ *of the penalty parameter*

(ii) *A central approximate solution*

$$x(t) = ((v^1(t), \ldots, v^L(t); \lambda(t)), \nu(t), \rho^+(t), \rho^-(t))$$

to (3.4) *associated with a large value of the penalty parameter can be explicitly converted to a good approximate solution to* (3.3) *as follows. Let*

$$W_m := t^{-1}\mathcal{A}_m^{-1}(x(t)) = \begin{pmatrix} \Xi_m & Q_m^T \\ Q_m & R_m \end{pmatrix}, \qquad m = 1, \ldots, M,$$

Ξ_m *being* $d \times d$ *block, and let*

$$E_m^+ = \omega_m^{-1}\Xi_m, \qquad m = 1, \ldots, M.$$

Then $E^+ = \{E_m^+\}_{m=1}^M$ *is a feasible solution to* (3.3)*, and the value of the objective of the latter problem at* E^+ *is larger than the optimal value* ϕ^* *of* (3.3) *by at most* $\Delta(t)$*, where*

$$\Delta(t) = t^{-1}\left[N(kS + D + 2) + \sum_{\ell=1}^{L} \dim(\delta^\ell) + 1\right].$$

4. Computational issues. The semidefinite problem (3.4) can be efficiently solved by modern interior point polynomial time methods; the most attractive seem to be the path-following algorithms, since they automatically generate (nearly) central approximate solutions with the value of the penalty parameter growing linearly at the rate $(1 + O(\vartheta^{-2}))$, where

$$\vartheta = M(kS + d) + 2M + \sum_{\ell=1}^{L} \dim(\delta^\ell) + 1$$

is the total row size of matrices from Σ. The computational effort per iteration (i.e., per increasing the penalty parameter in the aforementioned ratio) is dominated by the necessity of assembling and solving (with respect to d) the Newton system

$$[\nabla^2 \boldsymbol{B}(x)]d = b,$$

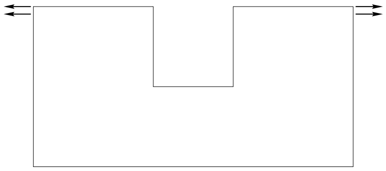


Fig. 5.1. *Example* 1.

$x \in \mathcal{V}'$ and b being given. It is easily seen that for (3.4) the latter task requires $O(L^3D^3)$ arithmetic operations. The theoretical upper bound on the number of iterations required to recover, via the scheme of Theorem 3.2, an ϵ-optimal solution to the original problem (3.3) (i.e., a feasible solution to (3.3) with the value of the objective greater than the optimal one by at most ϵ) is

$$\sqrt{\vartheta}\ln(\vartheta\epsilon^{-1}V),$$

where the *scale factor* V depends on the numerical values of the data. The practical behavior of a good interior point method as applied to (3.4) is even better than the one predicted by the theoretical complexity bound, and the typical number of iterations required to solve (3.3) to a reasonably high accuracy is 20–40.

The illustrated numerical results reported in the next section were obtained with the aid of the projective method [8] implemented in the LMI Toolbox for use with MATLAB—the only interior point solver for semidefinite programs that we had at our disposal. Unfortunately, this method is *not* a path-following method; this is why we were enforced to combine it with an additional (and computationally relatively cheap) interior point routine, based on Theorem 3.2, which, given a good feasible solution to (3.4), updates it into a central solution of the same quality and uses this "refined" solution to recover a nearly optimal solution to the problem of interest.

5. Examples.

Results of three numerical examples are presented in this section. The values of the "density" function ρ are depicted by gradations of gray: full black corresponds to high density, white to zero density (no material), etc.

Example 1. We consider a typical example of structural design: The two forces (or force and fixed boundary) are opposite to each other and there is a hole in between because of technological reasons. The geometry of domain Ω and the forces are depicted in Figure 5.1. The forces are considered as a single load. Because of symmetry, we could compute only one half of the original domain. The resulting values of the "density" function ρ for 29×29 mesh are presented in Figure 5.2; the figure is composed from two computational domains to obtain the original body.

Example 2. Let us now generalize Example 1 to a symmetric two-sided body shown in Figure 5.3. The body can be loaded by the forces on either the left- or the right-hand side. Therefore this example has to be considered as MLD (two-load case). Again, symmetry allows us to compute only one half of the original domain. The resulting values of the "density" function ρ for 37×25 mesh are also presented in Figure 5.3. Again, the figure is composed from two computational domains to get the full body.

Example 3. In this example we try to model a spanner. The geometry of domain Ω is depicted in Figure 5.4. The nut (depicted in full black in Figure 5.4) is considered to present a rigid obstacle for the spanner. Hence the spanner is in unilateral contact with the nut and there are no other boundary conditions. The loads are also shown in Figure 5.4. Note that the problem is nonlinear because of the unilateral contact conditions and that for positive vertical force we get a different design than for a negative one; hence we have to consider these two forces as two independent loads. The resulting values of the "density" function ρ for 37×22 discretization are shown in Figure 5.5. We also performed a more detailed analysis of the most interesting part around the nut: Figure 5.6 shows the values of ρ for 31×31 discretization of this part.

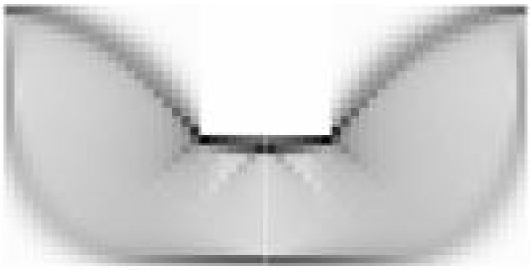


FIG. 5.2. *Example* 1.

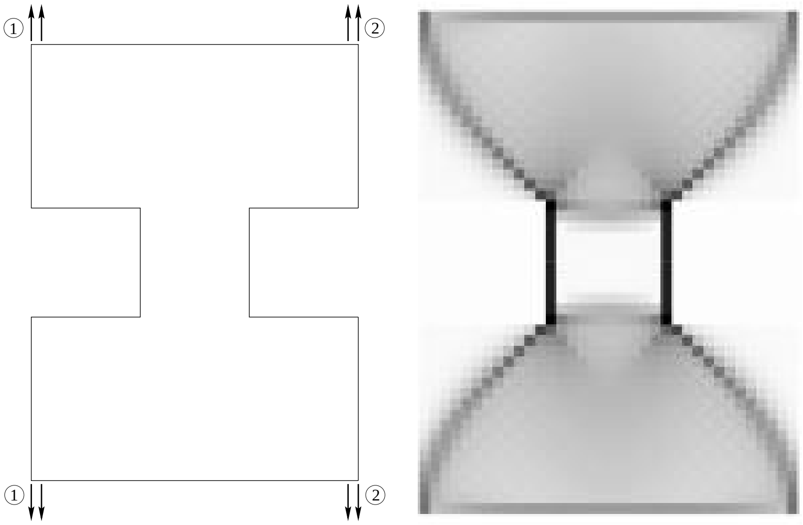


FIG. 5.3. *Example* 2.

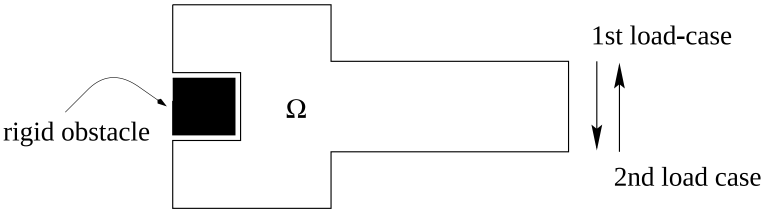


FIG. 5.4. *Example* 3.

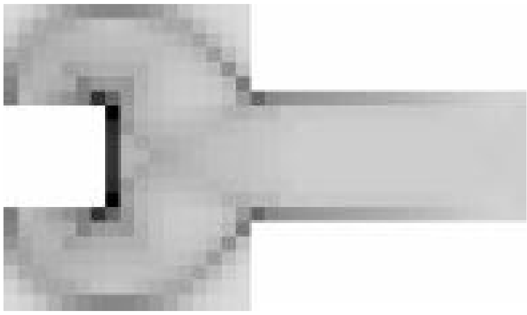


Fig. 5.5. *Example* 3.

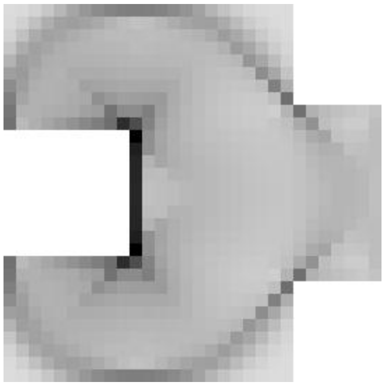


Fig. 5.6. *Example* 3.

6. Proofs of Theorems 3.1 and 3.2.

6.1. From the primal (3.3) to the dual (3.4). Recall the definitions of Λ (2.15), Λ_0 (2.17), and $\boldsymbol{v} = (v^1, \dots, v^L)$. Similarly to section 2, let

$$\begin{array}{rcl} \mathcal{V}' & = & \{(\boldsymbol{v}; \lambda) \in (\mathbb{R}^d)^L \times \mathbb{R}^L \mid C^\ell v^\ell < \lambda_\ell \delta^\ell, \lambda \in \Lambda_0\}, \\ \mathcal{V} & = & \mathrm{cl}\,\mathcal{V}' = \{(\boldsymbol{v}; \lambda) \in (\mathbb{R}^d)^L \times \mathbb{R}^L \mid C^\ell v^\ell \leq \lambda_\ell \delta^\ell, \lambda \in \Lambda\}. \end{array}$$

As in section 2, we can rewrite the function $\phi(\cdot)$ as

$$\begin{array}{rcl} \phi(E) & = & \displaystyle\sup_{\substack{(u^1,\lambda_1;\dots;u^L,\lambda_L): \\ \lambda\in\Lambda_0, u^\ell \in U^\ell}} \sum_{\ell=1}^{L} \left[2\lambda_\ell (f^\ell)^T u^\ell - \lambda_\ell \sum_{m=1}^{M} \omega_m \mathrm{tr}(E_m \zeta_m(u^\ell)\zeta_m^T(u^\ell)) \right] \\ & = & \displaystyle\sup_{(\boldsymbol{v};\lambda)\in\mathcal{V}'} \left[2\sum_{\ell=1}^{L} (f^\ell)^T v^\ell - \sum_{m=1}^{M}\sum_{\ell=1}^{L} \omega_m \lambda_\ell^{-1} tr(E_m \zeta_m(v^\ell)\zeta_m^T(v^\ell)) \right] \end{array}$$

so that (3.3) is only the problem

$$\min_{E\in\mathcal{E}_h} \sup_{(\boldsymbol{v};\lambda)\in\mathcal{V}'} \widehat{T}(E; (\boldsymbol{v}; \lambda)),$$

with

$$\widehat{T}(E, (\boldsymbol{v}; \lambda)) = 2\sum_{\ell=1}^{L} (f^\ell)^T v^\ell - \sum_{\ell=1}^{L}\sum_{m=1}^{M} \omega_m \lambda_\ell^{-1} \mathrm{tr}(E_m \zeta_m(v^\ell)\zeta_m^T(v^\ell)),$$

where $\mathcal{E}_h$ is defined in (3.2). By penalizing the linear inequalities in $\mathcal{E}_h$ and taking the supremum with respect to the penalty coefficients, we can rewrite the latter problem equivalently as

$$\min_{E\in\mathcal{P}} \sup_{\substack{(\boldsymbol{v};\lambda)\in\mathcal{V}',\\ \nu\geq 0,\sigma^+,\sigma^-\in\mathbb{R}^M_+}} T(E;(\boldsymbol{v};\lambda),\nu,\sigma^+,\sigma^-), \tag{6.1}$$

with

$$\begin{aligned} T(E;(\boldsymbol{v};\lambda),\nu,\sigma^+,\sigma^-) \; &= \; 2\sum_{\ell=1}^{L}(f^\ell)^T v^\ell - \sum_{\ell=1}^{L}\sum_{m=1}^{M}\omega_m\lambda_\ell^{-1}\mathrm{tr}(E_m\zeta_m(v^\ell)\zeta_m^T(v^\ell)) \\ &\quad -\nu\left[\alpha - \sum_{m=1}^{M}\omega_m\mathrm{tr}(E_m)\right] \\ &\quad -\sum_{m=1}^{M}[\sigma^-(\mathrm{tr}(E_m)-r_m^-)+\sigma^+(r_m^+-\mathrm{tr}(E_m))], \\ \mathcal{P} = \{\{E_m\}_{m=1}^M \mid E_m\in\Sigma_+^d, m=1,\ldots,M\}. \end{aligned}$$

The optimal value in (6.1), due to the origin of the problem, is exactly the optimal value ϕ^* of (3.3). Now let us pass from (3.3) to the problem with swapped infimum and supremum,

$$\sup_{\substack{(\boldsymbol{v};\lambda)\in\mathcal{V}',\\ \nu\geq 0,\sigma^+,\sigma^-\in\mathbb{R}^M_+}} \inf_{E\in\mathcal{P}} T(E;(\boldsymbol{v};\lambda),\nu,\sigma^+,\sigma^-), \tag{6.2}$$

and let ϕ^{**} be the optimal value in the latter problem. Note that by weak duality inequality

$$\phi^* \geq \phi^{**}. \tag{6.3}$$

By passing from $E=\{E_m\}_{m=1}^M$ to new variable $F=\{F_m\}_{m=1}^M$, $F_m=\omega_m E_m$, and setting

$$\rho^\pm := \frac{1}{\omega_m}\sigma_m, \quad s_m^\pm = \omega_m r_m^\pm, \quad m=1,\ldots,M,$$

we can rewrite the objective

$$\psi((\boldsymbol{v};\lambda),\nu,\rho^+,\rho^-) := \inf_{E\in\mathcal{P}} T(E;(\boldsymbol{v};\lambda),\nu,\sigma^+,\sigma^-)$$

of problem (6.2) as

$$\begin{aligned} \psi((\boldsymbol{v};\lambda),\nu,\rho^+,\rho^-) \; &= \; \inf_{F\in\mathcal{P}}\Bigg\{-\alpha\nu + 2\sum_{\ell=1}^{L}(f^\ell)^T v^\ell + \sum_{m=1}^{M}(s_m^-\rho_m^- - s_m^+\rho_m^+) \\ &\qquad - \sum_{m=1}^{M}\Bigg[\sum_{\ell=1}^{L}\lambda_\ell^{-1}\mathrm{tr}(F_m\zeta_m(v^\ell)\zeta_m^T(v^\ell)) \\ &\qquad\qquad + (\rho_m^- - \rho_m^+ - \nu)tr(F_m)\Bigg]\Bigg\}. \end{aligned} \tag{6.4}$$

Now, denoting by $\mu_{\max}(A)$ the largest eigenvalue of a symmetric matrix A and taking into account the evident relation

$$\max_{B\in\Sigma^d_+, tr(B)=r\geq 0} \operatorname{tr}(BC) = r\mu_{\max}(C)$$

which is valid for an arbitrary symmetric $d \times d$ matrix C, we can easily continue the above computation:

$$\begin{array}{rcl}
\psi((\boldsymbol{v};\lambda),\nu,\rho^+,\rho^-) & = & -\alpha\nu + 2\sum\limits_{\ell=1}^{L}(f^\ell)^T v^\ell + \sum\limits_{m=1}^{M}(s^-_m\rho^-_m - s^+_m\rho^+_m), \\
 & & \text{if } \mu_{\max}\left(\sum\limits_{\ell=1}^{L}\lambda_\ell^{-1}\zeta_m(v^\ell)\zeta_m^T(v^\ell)\right) \leq \nu + \rho^+_m - \rho^-_m, \\
 & & \quad m = 1,\ldots,M, \text{ and } \nu \geq 0, \\
\psi((\boldsymbol{v};\lambda),\nu,\rho^+,\rho^-) & = & -\infty, \\
 & & \text{otherwise.}
\end{array}$$

Thus the problem (6.2) becomes the optimization problem

$$\begin{array}{ll}
 & \text{maximize} \\
 & \psi((\boldsymbol{v};\lambda),\nu,\rho^+,\rho^-) = -\alpha\nu + 2\sum\limits_{\ell=1}^{L}(f^\ell)^T v^\ell + \sum\limits_{m=1}^{M}(s^-_m\rho^-_m - s^+_m\rho^+_m) \\
\text{s.t.} & \\
(6.5) & \mu_{\max}\left(\sum\limits_{\ell=1}^{L}\lambda_\ell^{-1}\zeta_m(v^\ell)\zeta_m^T(v^\ell)\right) \leq \nu + \rho^+_m - \rho^-_m, \quad m = 1,\ldots,M, \\
 & (\boldsymbol{v};\lambda) \in \mathcal{V}', \\
 & \rho^\pm \in \mathbb{R}^M_+, \\
 & \nu \geq 0.
\end{array}$$

Let I_p denote the unit $p \times p$ matrix, and let us write $A \succeq B$ whenever A, B are symmetric matrices of the same size and $A - B \succeq 0$. For positive λ_ℓ and rectangular $q \times p$ matrices $Z_\ell, \ell = 1,\ldots,L$, one clearly has

$$\sum_{\ell=1}^{L}\lambda_\ell^{-1}Z_\ell Z_\ell^T = [Z_1;Z_2;\ldots;Z_L][\operatorname{Diag}(\lambda_1 I_p,\lambda_2 I_p,\ldots,\lambda_L I_p)]^{-1}[Z_1;Z_2;\ldots;Z_L]^T$$

and therefore

$$\begin{array}{c}
a \geq \mu_{\max}\left(\sum\limits_{\ell=1}^{L}\lambda_\ell^{-1}Z_\ell Z_\ell^T\right), \\
\Updownarrow \\
aI_q \succeq [Z_1;Z_2;\ldots;Z_L][\operatorname{Diag}(\lambda_1 I_p,\lambda_2 I_p,\ldots,\lambda_L I_p)]^{-1}[Z_1;Z_2;\ldots;Z_L]^T \\
\Updownarrow \\
\begin{pmatrix} aI_q & [Z_1;Z_2;\ldots;Z_L] \\ [Z_1;Z_2;\ldots;Z_L]^T & \operatorname{Diag}(\lambda_1 I_p,\lambda_2 I_p,\ldots,\lambda_L I_p) \end{pmatrix} \succeq 0,
\end{array}$$

the concluding equivalence being given by the standard result on Schur's complement.

We conclude that (6.5) is equivalent to the problem

$$
\begin{array}{ll}
& \text{maximize} \\
& \psi((\boldsymbol{v};\lambda),\nu,\rho^+,\rho^-) = -\alpha\nu + 2\sum_{\ell=1}^{L}(f^\ell)^T v^\ell + \sum_{m=1}^{M}(s_m^-\rho_m^- - s_m^+\rho_m^+) \\
\text{s.t.} & \\
& \mathcal{A}_m(v,\nu,\rho^+,\rho^-) := \begin{pmatrix} (\nu+\rho_m^+-\rho_m^-)I_d & \zeta_m(v^1) & \zeta_m(v^2) & \cdots & \zeta_m(v^L) \\ \zeta_m^T(v^1) & \lambda_1 I_S & & & \\ \zeta_m^T(v^2) & & \lambda_2 I_S & & \\ \vdots & & & \ddots & \\ \zeta_m^T(v^L) & & & & \lambda_L I_S \end{pmatrix} \succeq 0, \\
& v \in \mathcal{V}', \\
& \rho^\pm \in \mathbb{R}_+^M, \\
& \nu \geq 0.
\end{array}
\tag{6.6}
$$

Problem (6.6) is "almost" the problem (3.4); the only difference is that the "unclosed" inequalities $(\boldsymbol{v};\lambda) \in \mathcal{V}'$, i.e.,

$$C^\ell v^\ell < \lambda_\ell \delta^\ell, \quad \lambda_\ell > 0, \quad \sum_\ell \lambda_\ell = 1,$$

of (6.6) in (3.4) are replaced with their closed versions $(\boldsymbol{v};\lambda) \in \mathcal{V}$, i.e.,

$$C^\ell v^\ell \leq \lambda_\ell \delta^\ell, \quad \lambda_\ell \geq 0, \quad \sum_\ell \lambda_\ell = 1.$$

It is immediately seen that this modification does not vary the optimal value. Indeed, (6.6) is clearly feasible. (In fact, it is even strictly feasible: there exists a feasible solution to the problem that makes all its inequalities strict. To get such a solution, it suffices to choose arbitrary $v \in \mathcal{V}'$ and positive vectors $\rho^\pm$ and then to extend this collection by large enough positive ν.) Due to feasibility of the problem, the standard approximation arguments demonstrate that its optimal value clearly remains unchanged when we pass from "unclosed" constraint $v \in \mathcal{V}'$ to its "closed" form $v \in \mathcal{V}$, thus arriving at the program (3.4). Consequently (see (6.3)),

$$\phi^* \geq \psi^*, \tag{6.7}$$

ψ^* being the optimal value in (3.4).

6.2. Proof of Theorem 3.2(i). Problem (3.4) is of the form (SDP); from the general theory of interior point methods (see [12]) it is known that existence of central approximate solutions to (SDP) is guaranteed by strict feasibility of the program (which indeed is the case for (3.4)) along with boundedness of the level sets of the objective

$$X(a) = \{x \mid \mathcal{A}x \succeq 0, e^T x = 1, c^T x \geq a\}$$

for every real a. Thus, all we need in order to prove (i) is to verify the boundedness of the level sets $X(a)$.

Consider a sequence

$$\{y_j = ((v^{1,j},\ldots,v^{L,j};\lambda_{1,j},\ldots,\lambda_{L,j}),\nu_j,\rho^{+,j},\rho^{-,j})\}_{j=1}^\infty$$

of points from $X(a)$, and let us prove that the sequence is bounded. Let $\pi_j = \max_{m=1,\ldots,M}[\nu_j + \rho_m^{+,j}]$. Since the matrices $\mathcal{A}_m(y_j)$ are positive semidefinite and $0 \le \lambda_{\ell,j}$, $\sum_\ell \lambda_{\ell,j} = 1$, we have $\|\zeta_m(v^{i,j})\| \le C\sqrt{\pi_j}$ for some constant C and all m, i, j. By assumption B, this observation yields that

$$\|v^{i,j}\| \le C'\sqrt{\pi_j} \tag{6.8}$$

for all i, j. It follows that the objective of (3.4) at y_j is at most

$$\begin{aligned}\theta_j &= -\alpha\nu_j + O(\sqrt{\pi_j}) + \sum_{m=1}^{M}(s_m^-\rho_m^{-,j} - s_m^+\rho_m^{+,j})\\ &= O(\sqrt{\pi_j}) - \left\{\sum_{m=1}^{M} s_m^-(\nu_j + \rho_m^{+,j} - \rho_m^{-,j})\right\}_1\\ &\quad - \left\{\left(\alpha - \sum_{m=1}^{M} s_m^-\right)\nu_j\right\}_2 - \left\{\sum_{m=1}^{M}(s_m^+ - s_m^-)\rho_m^{+,j}\right\}_3.\end{aligned}$$

Now, the quantities $\nu_j + \rho_m^{+,j} - \rho_m^{-,j}$ are nonnegative (they are diagonal entries of positive semidefinite matrices $\mathcal{A}_m(y_j)$), so that $\{\cdot\}_1 \ge 0$ and

$$0 \le \rho_m^{-,j} \le \pi_j. \tag{6.9}$$

By assumption C, we have $\{\cdot\}_2 + \{\cdot\}_3 \ge \kappa\pi_j$ with some positive κ, so that $\theta_j \le O(\sqrt{\pi_j}) - \kappa\pi_j$. On the other hand, θ_j is an upper bound on $\psi(y_j)$, and therefore the sequence $\{\theta_j\}$ is bounded below; thus, the sequence π_j is bounded, which, in view of (6.9) and (6.8), implies boundedness of $\{y_j\}$.

6.3. Proof of Theorem 3.2(ii) and Theorem 3.1. Recall the following.

For every feasible solution E to the problem of interest (3.3)*, the value of the objective at the solution is equal to*

$$\sup_{(\boldsymbol{v};\lambda)\in\mathcal{V}',\nu\ge 0,\rho^\pm\in\mathbb{R}^M_+} T(E;(\boldsymbol{v};\lambda),\nu,\rho^+,\rho^-), \tag{6.10}$$

with T given in (6.1)*.*

Now let $x(t) = ((v^1(t),\ldots,v^L(t);\lambda(t)),\nu(t),\rho^+(t),\rho^-(t))$ be a central approximate solution to (3.4), and let $W = t^{-1}[\mathcal{A}x(t)]^{-1}$, where $(\mathcal{A}, e)$ are the data from the representation of (3.4) in the generic form (SDP). Note that W is a block-diagonal positive definite matrix, and that its first N diagonal blocks are the matrices

$$W_m = \begin{pmatrix} \Xi_m & Q_m^T \\ Q_m & R_m \end{pmatrix}, m = 1, \ldots, M,$$

mentioned in (ii). Due to the structure of constraints in (3.4), the remaining diagonal blocks in W are k diagonal matrices W_{M+i} of the row sizes $\dim(\delta^\ell)$ associated with the constraints $\mathrm{Diag}(\lambda_\ell\delta^\ell - C^\ell v^\ell) \succeq 0$, $\ell = 1, \ldots, L$, two more diagonal $N \times N$ matrices W_{M+k+1}, W_{M+k+2} associated with the constraints $\mathrm{Diag}(\rho^+) \succeq 0$, $\mathrm{Diag}(\rho^-) \succeq 0$, respectively, and 1×1 matrix W_{M+k+3} associated with the constraint $\nu \ge 0$.

The fact that $x(t)$ minimizes the aggregate (3.5) over $\mathcal{X}'$ means that the vector

$$\mathcal{A}^*W + c$$

is proportional to the vector e defining, via the equality constraint $e^T x = 1$, the affine span of $\mathcal{X}$; here $\mathcal{A}^*$ is the operator conjugate to $\mathcal{A}$, i.e., $\mathrm{tr}(y[\mathcal{A}x]) = (\mathcal{A}^* y)^T x$ for all $x \in \mathbb{R}^N, y \in \Sigma$. Now the only nonzero component of vector e for the problem (3.4) is the λ-component, and this latter component is composed of ones. Substituting in the relation

$$\mathcal{A}^* W + c = \theta e \tag{6.11}$$

the particular data of (3.4), we end up with the following system of relations (where $\mathrm{diag}(Q)$ denotes the diagonal of a square matrix and $\mathrm{diag}_i(Q)$ is the ith diagonal entry of the matrix):

$$\begin{array}{ll}
\text{(a.1)} & \sum\limits_{m=1}^{M} tr(\Xi_m) + W_{M+k+3} = \alpha; \\
\text{(a.2)} & \mathrm{tr}(\Xi_m) + \mathrm{diag}_m(W_{M+k+1}) = s_m^+, \qquad m = 1, \dots, M; \\
\text{(a.3)} & \mathrm{tr}(\Xi_m) - \mathrm{diag}_m(W_{M+k+2}) = s_m^-, \qquad m = 1, \dots, M; \\
\text{(b)} & 2\sum\limits_{m=1}^{M} \mathrm{tr}(Z_m^T(w) Q_m) \\
\text{(6.12)} & -\sum\limits_{\ell=1}^{L} \mathrm{tr}(W_{M+i}\mathrm{Diag}(C^\ell w^\ell)) = -2\sum\limits_{\ell=1}^{L} (f^\ell)^T w^\ell \quad \text{for all } w = (w^1, \dots, w^L), \\
 & \qquad Z_m^T(w) = [\zeta_m(w^1); \dots; \zeta_m(w^L)]; \\
\text{(c)} & \sum\limits_{m=1}^{M} \mathrm{tr}(R_m \pi(\lambda)) \\
 & +\sum\limits_{\ell=1}^{L} \mathrm{tr}(W_{M+i}\mathrm{Diag}(\delta^\ell \lambda_\ell)) = \theta \sum\limits_{\ell=1}^{L} \lambda_\ell \quad \text{for all } \lambda \in \mathbb{R}^L,
\end{array}$$

where $\pi(\lambda)$, $\lambda \in \mathbb{R}^L$, is the $kS \times kS$ diagonal matrix where the first S diagonal entries are equal to λ_1, the next S entries are equal to λ_2, and so on.

Note that (6.12(a)) along with evident positive definiteness of all W_m (and, consequently, of all E_m^+) demonstrate that E^+ is a feasible solution to (3.3).

We have

$$\begin{array}{rcl}
-\psi^* & \leq & -c^T x(t) \\
 & = & (\mathcal{A}^* W - \theta e)^T x(t) \\
 & & \quad \text{(see (6.11))} \\
 & = & \mathrm{tr}(W[\mathcal{A}x]) - \theta \\
 & & \quad \text{(since } e^T x(t) = 1) \\
 & = & t^{-1}[N(kS + D + 2) + \sum\limits_{\ell=1}^{L} \dim(\delta^\ell) + 1] - \theta \\
 & & \quad \text{(since } W = t^{-1}[\mathcal{A}x(t)]^{-1}) \\
 & = & \Delta(t) - \theta.
\end{array} \tag{6.13}$$

According to (6.13), we have

$$\theta \leq \psi^* + \Delta(t). \tag{6.14}$$

Now let $(\boldsymbol{v};\lambda) \in \mathcal{V}'$, $\nu \geq 0$, $\rho^{\pm} \in \mathbb{R}^M_+$. Let us derive an upper bound for the quantity $T(E^+;(\boldsymbol{v};\lambda),\nu,\rho^+,\rho^-)$. The matrices

$$\begin{aligned}
A_m &= \begin{pmatrix} \sum_{\ell=1}^L \lambda_\ell^{-1}\zeta_m(v^\ell)\zeta_m^T(v^\ell) & Z_m^T(v^1,\dots,v^\ell) \\ Z_m(v^1,\dots,v^\ell) & \pi(\lambda) \end{pmatrix}, \qquad m = 1,\dots,M,\\
A_{M+i} &= \mathrm{Diag}(\lambda_\ell\delta^\ell - C^\ell v^\ell), \qquad i,\ell = 1,\dots,k,\\
A_{M+k+1} &= \mathrm{Diag}(\{\omega_m^{-1}\rho_m^+\}_{m=1}^M),\\
A_{M+k+2} &= \mathrm{Diag}(\{\omega_m^{-1}\rho^- l\}_{m=1}^M),\\
A_{M+k+3} &= \nu
\end{aligned}$$

clearly are positive semidefinite, so that

$$\begin{aligned}
0 \leq\ & \sum_{m=1}^{M+k+3}\mathrm{tr}(W_m A_m)\\
=\ & \sum_{m=1}^M \mathrm{tr}\left(\Xi_m\sum_{\ell=1}^L\lambda_\ell^{-1}\zeta_m(v^\ell)\zeta_m^T(v^\ell)\right)\\
& +2\sum_{m=1}^M\mathrm{tr}(Z_m^T(v^1\dots,v^L)Q_m) + \sum_{m=1}^M\mathrm{tr}(R_m\pi(\lambda))\\
& +\sum_{\ell=1}^L\mathrm{tr}(\mathrm{Diag}(\lambda_\ell\delta^\ell - C^\ell v^\ell)W_{M+i})\\
& +\sum_{m=1}^M\omega_m^{-1}[\rho_m^+\mathrm{diag}_m(W_{M+k+1}) + \rho_m^-\mathrm{diag}_m(W_{M+k+2})]\\
& +\nu W_{M+k+3}
\end{aligned}$$

$$\begin{aligned}
=\ & \sum_{m=1}^M \mathrm{tr}\left(\Xi_m\sum_{\ell=1}^L\lambda_\ell^{-1}\zeta_m(v^\ell)\zeta_m^T(v^\ell)\right) - 2\sum_{\ell=1}^L(f^\ell)^T v^\ell\\
& +\theta\sum_{\ell=1}^L\lambda_\ell\\
& +\sum_{m=1}^M\omega_m^{-1}[\rho_m^+(s_m^+ - \mathrm{tr}(\Xi_m)) + \rho_m^-(\mathrm{tr}(\Xi_m) - s_m^-)]\\
& +\nu W_{M+k+3}\\
& \qquad \text{(we have used (6.12(a.2, a.3, b, c)))}\\
=\ & \sum_{m=1}^M \mathrm{tr}\left(\Xi_m^+\sum_{\ell=1}^L\lambda_\ell^{-1}\zeta_m(v^\ell)\zeta_m^T(v^\ell)\right) - 2\sum_{\ell=1}^L(f^\ell)^T v^\ell\\
& +\sum_{m=1}^M\omega_m^{-1}[\rho_m^+(s_m^+ - \mathrm{tr}(\Xi_m)) + \rho_m^-(\mathrm{tr}(\Xi_m) - s_m^-)]\\
& +\nu\left(\alpha - \sum_{m=1}^M\mathrm{tr}(\Xi_m)\right) + \theta\\
& \qquad \text{(we have used (6.12(a.1)))}\\
=\ & -T(E^+;(\boldsymbol{v};\lambda),\nu,\rho^+,\rho^-) + \theta\\
& \qquad \text{(see (6.1)),}
\end{aligned}$$

which means

$$T(E^{+};(\boldsymbol{v};\lambda),\nu,\rho^{+},\rho^{-}) \le \theta \le \psi^{*} + \Delta(t),$$

the concluding inequality being given by (6.14). Applying (6.10), we conclude that the value of the objective of (3.3) at E^{+} is greater than ψ^{*} by at most $\Delta(t)$. Since the optimal value in (3.3) is $\phi^{*} \ge \psi^{*}$ (see (6.7)), this observation completes the proof.

REFERENCES

[1] A. Ben-Tal and M. Zibulevsky, *Penalty/barrier multiplier methods for convex programming problems*, SIAM J. Optim., 7 (1997), pp. 347–366.

[2] M. Bendsøe, *Optimization of Structural Topology, Shape and Material*, Springer-Verlag, Heidelberg, 1995.

[3] M. P. Bendsøe and A. Díaz, *Optimization of material properties for Mindlin plate design*, Structural Optim., 6 (1993), pp. 268–270.

[4] M. P. Bendsøe, A. Díaz, R. Lipton, and J. E. Taylor, *Optimal design of material properties and material distribution for multiple loading conditions*, Internat. J. Numer. Methods Engrg., 38 (1995), pp. 1149–1170.

[5] M. P. Bendsøe, J. M. Guades, R. Haber, P. Pedersen, and J. E. Taylor, *An analytical model to predict optimal material properties in the context of optimal structural design*, J. Appl. Mech., 61 (1994), pp. 930–937.

[6] M. P. Bendsøe, J. M. Guades, S. Plaxton, and J. E. Taylor, *Optimization of structure and material properties for solids composed of softening material*, Internat. J. Solids Structures, 33 (1995), pp. 1179–1813.

[7] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, New York, Oxford, 1978.

[8] P. Gahinet and A. Nemirovski, *The projective method for solving linear matrix inequalities*, Math. Programming Ser. B, 77 (1997), pp. 163–190.

[9] F. Jarre, M. Kočvara, and J. Zowe, *Optimal truss design by interior-point methods*, SIAM J. Optim., 8 (1998), pp. 1084–1107.

[10] M. Kočvara, M. Zibulevsky, and J. Zowe, *Mechanical design problems with unilateral contact*, RAIRO Modél. Math. Anal. Numér., 32 (1998), pp. 255–281.

[11] J.-J. Moreau, *Théorèmes "inf-sup,"* C. R. Acad. Sci. Paris, 258 (1964), pp. 2720–2722.

[12] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Stud. Appl. Math. 13, SIAM, Philadelphia, 1994.

[13] J. Petersson, *On stiffness maximization of variable thickness sheet with unilateral contact*, Quart. Appl. Math., 54 (1996), pp. 541–550.

[14] U. Ringertz, *On finding the optimal distribution of material properties*, Structural Optim., 5 (1993), pp. 265–267.

[15] A. E. Taylor and D. C. Lay, *Introduction to Functional Analysis*, Krieger, Malabar, FL, 1980.

[16] J. Zowe, M. Kočvara, and M. Bendsøe, *Free material optimization via mathematical programming*, Math. Programming Ser. B, 79 (1997), pp. 445–468.

SIAM J. Optim.
Vol. 9, No. 4, pp. 833–862

A GLOBAL CONVERGENCE ANALYSIS OF AN ALGORITHM FOR LARGE-SCALE NONLINEAR OPTIMIZATION PROBLEMS[*]

PAUL T. BOGGS[†], ANTHONY J. KEARSLEY[‡], and JON W. TOLLE[§]

This work is dedicated, with respect and admiration, to John Dennis
on the occasion of his 60th birthday

Abstract. In this paper we give a global convergence analysis of a basic version of an SQP algorithm described in [P. T. Boggs, A. J. Kearsley, and J. W. Tolle, *SIAM J. Optim.*, 9 (1999), pp. 755–778] for the solution of large-scale nonlinear inequality-constrained optimization problems. Several procedures and options have been added to the basic algorithm to improve the practical performance; some of these are also analyzed. The important features of the algorithm include the use of a constrained merit function to assess the progress of the iterates and a sequence of approximate merit functions that are less expensive to evaluate. It also employs an interior point quadratic programming solver that can be terminated early to produce a truncated step.

Key words. sequential quadratic programming, global convergence, merit function, large-scale problems

AMS subject classifications. 49M37, 65K05, 90C30

PII. S1052623497316026

1. Introduction.

In this report we consider an algorithm to solve the inequality-constrained minimization problem

$$\begin{aligned} &\min_x \ f(x) \\ &\text{subject to} \ \ g(x) \leq 0, \end{aligned} \tag{1.1}$$

where $x \in \mathcal{R}^n$, and $f : \mathcal{R}^n \to \mathcal{R}$ and $g : \mathcal{R}^n \to \mathcal{R}^m$ are smooth functions, in the case when the dimensions of the problem, n and/or m, are large. The algorithm in question has been reported in several papers and technical reports (see, for example, [2], [6], and [13]); the purpose here is to provide a rigorous analysis of the global convergence properties of a basic version of the algorithm. We also analyze certain procedures that have been added to the algorithm to improve the practical performance. The algorithm is an extension of the sequential quadratic programming (SQP) method (see [5]). That is, at each iteration of the typical SQP algorithm a quadratic program is solved to obtain the step direction. In particular, given current approximations, x^k and λ^k, to a solution and a corresponding multiplier of (1.1), the quadratic program

$$\begin{aligned} &\min_\delta \ \nabla f(x^k)^{\mathrm{t}}\delta + \tfrac{1}{2}\delta^{\mathrm{t}}B^k\delta \\ &\text{subject to} \ \ \nabla g(x^k)^{\mathrm{t}}\delta + g(x^k) \leq 0 \end{aligned} \tag{1.2}$$

[*] Received by the editors February 5, 1997; accepted for publication (in revised form) January 26, 1999; published electronically September 24, 1999. This work was performed by an employee of the U.S. Government or under U.S. Government contract. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/siopt/9-4/31602.html

[†] Computational Science and Mathematics Research Department, Sandia National Laboratories, Livermore, CA 94550 (ptboggs@ca.sandia.gov).

[‡] Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213-3890 (anthonyk@andrew.cmu.edu).

[§] Departments of Mathematics and Operations Research, University of North Carolina, Chapel Hill, NC 27599-3250 (tolle@or.unc.edu).

is formed and, if feasible, solved. The matrix B^k is generally taken to be an approximation of the Hessian with respect to x of the Lagrangian function of (1.1) at (x^k,λ^k). The solution, δ^k, is then used to generate the next approximation, x^{k+1}, by

$$x^{k+1} = x^k + \alpha\delta^k,$$

where α is a scalar *steplength* determined by a line search. A new multiplier approximation, λ^{k+1}, can also be obtained from the quadratic program, for example, by using a multiplier, μ^k, associated with δ^k.

The most crucial factors in the application of the SQP method to problems of the form (1.1) are the choice of the approximating matrices B^k to be used in the quadratic programs, the accuracy to which these quadratic subproblems are solved, and the choice of a *merit function* with which to measure progress toward a solution in the line search step. The matrix B^k determines how well the quadratic program models the true problem (1.1) as well as how easily the quadratic program can be solved; this choice may be constrained by sparsity or other considerations. The accuracy of the solution of (1.2) can have a profound effect on the overall efficiency of the algorithm; i.e., approximately solving (1.2), especially in early iterations, often results in less overall work. The merit function determines the choice of steplength α. A good merit function will balance the sometimes conflicting goals of decreasing f and decreasing infeasibility.

Much of the theory underlying the possible choices of the B^k and the ideal properties of a merit function can be found in [5]. In this work there can also be found references to the extensive literature concerning how best to implement an SQP algorithm. Most of this research is directed toward solving small- to medium-sized problems, most often with only equality constraints. Although (1.1) contains only inequality constraints, the algorithm described herein is designed to be applied to large-scale problems with general equality and inequality constraints; the equality constraints can be included without significantly changing the analysis.

Relatively few theoretical and computational algorithms for large-scale nonlinear programming problems have been proposed in the literature. One of the earliest methods is MINOS [15], a projected Lagrangian method originally developed for linear constraints. A more recent example is LANCELOT [9], which is an augmented Lagrangian method employing an ∞-norm trust region, with numerous options helpful in solving various classes of problems. Examples of SQP approaches include [14] and [12]. Some algorithms provide the option of using interior point methods to solve for the descent direction (see, for example, [11]), while others involve various direct extensions of the interior point ideas to the nonlinear setting, including [10]. Our approach differs significantly from these other methods. In particular, we may approximately solve the quadratic subproblem using an interior point method with a variation of a trust region, and we employ a different method of testing for acceptance of a step, incorporating an inexpensively evaluated approximate merit function that changes at each iteration. This paper is intended to provide a theoretical underpinning for our algorithm; [2] and [13] provide details of specific implementations as well as the results of numerical experiments on practical test problems. The performance of these algorithms on this test set, in our view, amply justifies this effort.

In section 2 we describe the merit function and its relationship to the original problem (1.1). Since our merit function depends on estimates of the nonnegative slack variables, we need to specify our procedure to update these. The complete steps are defined in section 3 along with the definition of the approximate merit functions that we employ. The assumptions and their implications for our analysis are set forth in section 4. In sections 5 and 6 we derive the fundamental descent properties for steps generated by solving the QP subproblems completely, and present the basic algorithm. Section 7 contains the main global convergence theorem for the algorithm. In particular, we obtain actual convergence of the iterates to a critical point of (1.1) as compared to the weaker result that limit points are critical points. In section 8 we analyze the descent properties of steps formed by approximately solving (1.2), illustrating how convergence can be achieved. Finally, in section 9 we comment on other aspects of the practical implementation of the algorithm.

2. The merit function and its properties.

An important facet of our algorithm is the merit function ψ, which is a scalar-valued function such that a reduction in ψ implies progress toward a solution. Typically, ψ is chosen such that an unconstrained minimum of ψ corresponds to a solution of (1.1). However, to obtain our merit function for the inequality-constrained problem we introduce *nonnegative* slack variables $z \in \mathcal{R}^m$ so that the feasibility constraints for (1.1) become

$$g(x) + z = 0.$$

With the addition of these slack variables we can derive a merit function based on a weighted ℓ_2 exact penalty function for the resulting equality-constrained program. The merit function has the form

$$\psi_d(x,z) = f(x) + \bar{\lambda}(x,z)^{\mathsf{t}}\bar{c}(x,z) + \frac{1}{d}\bar{c}(x,z)^{\mathsf{t}}[\bar{A}(x,z)]^{-1}\bar{c}(x,z), \tag{2.1}$$

where

$$\begin{aligned}\bar{c}(x,z) &= g(x) + z,\\ \bar{A}(x,z) &= \nabla g(x)^{\mathsf{t}}\nabla g(x) + Z,\\ \bar{\lambda}(x,z) &= -[\bar{A}(x,z)]^{-1}\nabla g(x)^{\mathsf{t}}\nabla f(x),\end{aligned}$$

d is a (small) positive parameter, e is the vector of ones, and

$$Z = \operatorname{diag}\{z_1, \ldots, z_m\}.$$

Here and throughout the paper the symbol $\nabla h(x)$ will denote the Jacobian (or, in the case of a scalar function, the gradient) of the function $h(x)$. If the Jacobian refers to differentiation with respect to only a subset of the variables, this will be indicated by a subscript, i.e., $\nabla_x\bar{\lambda}(x,z)$. The set $\{(x,z) : \bar{c}(x,z) = 0 \text{ and } z \geq 0\}$ can be thought of as the feasible set for (1.1) and the function $\bar{\lambda}(x,z)$ can be interpreted as a weighted least squares approximation to the Lagrange multiplier vector for (1.1). A motivation for including the weighting factor $\bar{A}(x,z)^{-1}$ can be found in [8]. Further details and references for this merit function, including its derivation, can be found in [6].

Before relating the minimization of $\psi_d(x,z)$ to the solution of (1.1), we introduce some notation to be used in the remainder of the paper. The Lagrangian function for (1.1) will be denoted by

$$L(x,\lambda) = f(x) + g(x)^{\mathsf{t}}\lambda,$$

and the Hessian of this Lagrangian by $HL(x,z)$. In particular, $H_{xx}L$ will denote the Hessian with respect to the vector x. A *first-order solution* to (1.1) will be denoted by (x^*,λ^*); that is, x^* and λ^* will satisfy

$$\nabla f(x^*) + \nabla g(x^*)\lambda^* = 0, \tag{2.2}$$

$$g(x^*) \leq 0, \tag{2.3}$$

$$\lambda^* \geq 0, \tag{2.4}$$

$$g(x^*)^{\mathrm{t}}\lambda^* = 0. \tag{2.5}$$

If, in addition, x^* and λ^* satisfy strict complementary slackness, the Hessian matrix $H_{xx}L(x^*, z^*)$ is positive definite on the tangent space to the active constraint set at x^*, and the set $\{\nabla g_i(x^*) : g_i(x^*) = 0\}$ is linearly independent, then we will call (x^*,λ^*) a *strong* solution to (1.1). We shall denote the set of first-order solutions (or critical points) to (1.1) by $\mathcal{S}$. That is,

$$\mathcal{S} = \{(x, z) : (x, \lambda) \text{ is a first-order solution of (1.1) for some } \lambda \text{ and } z = -g(x)\}.$$

For positive ϵ we denote an ϵ-neighborhood of $\mathcal{S}$ by

$$\mathcal{S}_\epsilon = \{(x, z) : \|(x, z) - (x^*, z^*)\| < \epsilon \text{ for some } (x^*, z^*) \in \mathcal{S}\}.$$

As noted above, feasibility for (1.1) can be expressed in terms of the x and z variables. Accordingly, we represent an η-neighborhood of the feasible set as

$$\mathcal{C}_\eta = \{(x, z) : r(x, z) \leq \eta\}, \tag{2.6}$$

where

$$r(x, z) = \|\bar{c}(x, z)\|^2. \tag{2.7}$$

In this notation, $\mathcal{C}_0$ is the feasible set.

Since the z variables must be nonnegative, the minimizers of the merit function $\psi_d(x, z)$ defined above have to be considered *constrained* optimal points. That is, they are solutions of

$$\begin{array}{l} \min\limits_{x,z} \ \psi_d(x, z) \\ \text{subject to } \ z \geq 0. \end{array} \tag{2.8}$$

We have that $(\hat{x}, \hat{z}, \hat{\omega})$ is a first-order solution of (2.8) if

$$\nabla_x \psi_d(\hat{x}, \hat{z}) = 0, \tag{2.9}$$

$$\nabla_z \psi_d(\hat{x}, \hat{z}) - \hat{\omega} = 0, \tag{2.10}$$

$$\hat{z} \geq 0, \tag{2.11}$$

$$\hat{\omega} \geq 0, \tag{2.12}$$

$$\hat{z}^{\mathrm{t}}\hat{\omega} = 0. \tag{2.13}$$

Let

$$\mathcal{M}(\hat{z}) = \{s = (s_x, s_z) : (s_z)_j = 0 \text{ if } \hat{z}_j = 0\}. \tag{2.14}$$

Then the second-order condition for (2.8) is that

$$s^{\mathrm{t}} H\psi_d(\hat{x}, \hat{z})\, s > 0$$

for all $s \in \mathcal{M}(\hat{z}),\ s \neq 0$. Since the active constraint gradients for (2.8) are always linearly independent, if this second-order condition and strict complementary slackness ($\hat{z}_j = 0$ implies $\hat{\omega}_j > 0$) hold, then a first-order solution is a strong solution to (2.8).

The following useful formulas for the derivatives of $\psi_d(x, z)$ are easily derived:

$$\begin{aligned} \nabla_x \psi_d(x, z) = {} & \nabla_x L(x, \bar{\lambda}(x, z)) + \nabla_x \bar{\lambda}(x, z) \bar{c}(x, z) \\ & + \frac{2}{d} \nabla g(x) [\bar{A}(x, z)]^{-1} \bar{c}(x, z) + V_1(x, z) \end{aligned} \tag{2.15}$$

and

$$\begin{aligned} \nabla_z \psi_d(x, z) = {} & \bar{\lambda}(x, z) + \nabla_z \bar{\lambda}(x, z) \bar{c}(x, z) \\ & + \frac{2}{d} [\bar{A}(x, z)]^{-1} \bar{c}(x, z) + V_2(x, z), \end{aligned} \tag{2.16}$$

where V_1 and V_2 are $O(r(x, z))$. From the expression for $\bar{\lambda}(x, z)$ we also obtain the expressions

$$\begin{aligned} \nabla_x \bar{\lambda}(x, z) = & - W(x, \nabla_x L(x, \bar{\lambda}(x, z))) \\ & - H_{xx} L(x, \bar{\lambda}(x, z)) \nabla g(x) [\bar{A}(x, z)]^{-1} \end{aligned} \tag{2.17}$$

and

$$\nabla_z \bar{\lambda}(x, z) = -\bar{\Lambda}(x, z)\, [\bar{A}(x, z)]^{-1}, \tag{2.18}$$

where $W(x, y) = O(y)$ uniformly in x and

$$\bar{\Lambda}(x, z) = \operatorname{diag}\left\{ \bar{\lambda}_1(x, z), \ldots, \bar{\lambda}_m(x, z) \right\}.$$

The following propositions establish the relationships between the solutions of (1.1) and (2.8). We assume in every case that $\bar{A}(x^*, z^*)$ is nonsingular and hence positive definite (see assumption A4 in section 4).

PROPOSITION 2.1. *If* (x^*, λ^*) *is a first-order solution for* (1.1) *and* z^* *is set equal to* $-g(x^*)$*, then* $\bar{\lambda}(x^*, z^*) = \lambda^*$ *and the triple* (x^*, z^*, λ^*) *is a first-order solution for* (2.8). *In addition, if* (x^*, λ^*) *is a strong solution to* (1.1) *and* d *is sufficiently small, then* (x^*, z^*, λ^*) *is a strong solution to* (2.8).

Proof. If (x^*, λ^*) is a first-order solution to (1.1), then from (2.2),

$$[\bar{A}(x^*, z^*)]^{-1} \nabla g(x^*)^{\mathrm{t}} \nabla f(x^*) + [\bar{A}(x^*, z^*)]^{-1} \nabla g(x^*)^{\mathrm{t}} \nabla g(x^*) \lambda^* = 0.$$

It follows from the definition of $\bar{\lambda}(x^*, z^*)$ and $\bar{A}(x^*, z^*)$ that

$$-\bar{\lambda}(x^*, z^*) + \lambda^* - [\bar{A}(x^*, z^*)]^{-1} Z^* \lambda^* = 0.$$

But since the complementary slackness conditions (2.5) and the definition of z^* imply

$$Z^* \lambda^* = 0, \tag{2.19}$$

it follows that

$$\lambda^* = \bar{\lambda}(x^*, z^*). \tag{2.20}$$

Then, since $\bar{c}(x^*, z^*) = 0$, (2.15) and (2.16) yield

$$\nabla_x \psi_d(x^*, z^*) = \nabla f(x^*) + \nabla g(x^*)\, \lambda^* = 0$$

and

$$\nabla_z \psi_d(x^*, z^*) = \lambda^* \geq 0.$$

These equations, together with (2.19), imply that (x^*,z^*,λ^*) is a first-order solution for (2.8). Now assume that (x^*, λ^*) is a strong solution to (1.1). Then strict complementary slackness for (2.8) follows from the definition of z^* and (2.19) since strict complementary slackness holds for (1.1). We now assume that the second-order condition holds for (1.1), i.e., that

$$v^{\mathsf{t}} H_{xx} L(x^*, \lambda^*)\, v > 0$$

if $v \neq 0$ and $\nabla g_i(x^*)^{\mathsf{t}} v = 0$ for all i such that $g_i(x^*) = 0$. To construct the Hessian of $\psi_d(x, z)$, we use (2.15), (2.16), (2.17), and (2.18) together with $\bar{c}(x^*, z^*) = 0$ and $\nabla_x L(x^*, z^*) = 0$ to obtain

$$\begin{aligned}
H_{xx}\psi_d(x^*, z^*) &= H_{xx} L(x^*, z^*) - \nabla g(x^*)[\bar{A}(x^*, z^*)]^{-1} \nabla g(x^*)^{\mathsf{t}} H_{xx} L(x^*, z^*) \\
&\quad - H_{xx} L(x^*, z^*) \nabla g(x^*)[\bar{A}(x^*, z^*)]^{-1} \nabla g(x^*)^{\mathsf{t}} \\
&\quad + \frac{2}{d} \nabla g(x^*)[\bar{A}(x^*, z^*)]^{-1} \nabla g(x^*)^{\mathsf{t}}, \\
H_{x,z}\psi_d(x^*, z^*) &= -H_{xx} L(x^*, z^*) \nabla g(x^*)[\bar{A}(x^*, z^*)]^{-1} \\
&\quad - \nabla g(x^*) \bar{\Lambda}(x^*, z^*)[\bar{A}(x^*, z^*)]^{-1} + \frac{2}{d} \nabla g(x^*)[\bar{A}(x^*, z^*)]^{-1}, \\
H_{zz}\psi_d(x^*, z^*) &= -2\, \bar{\Lambda}(x^*, z^*)[\bar{A}(x^*, z^*)]^{-1} + \frac{2}{d}[\bar{A}(x^*, z^*)]^{-1}.
\end{aligned}$$

Let $\mathcal{M}(z^*)$ be the set defined in (2.14). Then, for $s \in \mathcal{M}(z^*)$, (2.19) implies

$$\bar{\Lambda}(x^*, z^*)\, s_z = 0.$$

Thus

$$\begin{aligned}
s^{\mathsf{t}} H\psi_d(x^*, z^*)\, s &= s_x^{\mathsf{t}} H_{xx} L(x^*, z^*)\, s_x \\
&\quad - 2\, s_x^{\mathsf{t}} \nabla g(x^*)[\bar{A}(x^*, z^*)]^{-1} \nabla g(x^*)^{\mathsf{t}} H_{xx} L(x^*, z^*)\, s_x \\
&\quad - 2\, s_x^{\mathsf{t}} H_{xx} L(x^*, z^*) \nabla g(x^*)[\bar{A}(x^*, z^*)]^{-1} s_z \\
&\quad - 2\, s_x^{\mathsf{t}} \nabla g(x^*) \bar{\Lambda}(x^*, z^*)[\bar{A}(x^*, z^*)]^{-1} s_z \\
&\quad + \frac{2}{d} (\nabla g(x^*)^{\mathsf{t}} s_x + s_z)^{\mathsf{t}} [\bar{A}(x^*, z^*)]^{-1} (\nabla g(x^*)^{\mathsf{t}} s_x + s_z).
\end{aligned}$$

If

$$\nabla g(x^*)^{\mathsf{t}} s_x + s_z \neq 0,$$

then for d sufficiently small, this quadratic form is positive. If this vector is zero, then the quadratic form reduces to

$$s^{\mathsf{t}} H\psi_d(x^*, z^*)\, s = s_x^{\mathsf{t}} H_{xx} L(x^*, z^*)\, s_x.$$

But $\nabla g_i(x^*)^{\mathsf{t}} s_x = -(s_z)_i = 0$ for i such that $g_i(x^*) = -{z^*}_i = 0$, and hence the second-order condition for (1.1) implies that $s_x^{\mathsf{t}} H_{xx} L(x^*, z^*)\, s_x$ is positive. A standard argument now shows that the matrix $H\psi_d(x^*, z^*)$ must be positive definite on the set

$\mathcal{M}(z^*)$. Since the active constraint gradients for (2.8) are always linearly independent, we have shown that (x^*, z^*, λ^*) is a strong solution to (2.8). $\square$

The converse of the above proposition is not true in general since (2.8) may have solutions $(\hat{x}, \hat{z})$ for which $\hat{z} \neq -g(\hat{x})$; but for d sufficiently small, these nonfeasible solutions are far from the feasible set $\mathcal{C}_0$. The following proposition is a partial converse; it guarantees that any solution to (2.8) that is in $\mathcal{C}_0$ is a solution to (1.1).

PROPOSITION 2.2. *If (x^*,z^*) is a first-order solution to* (2.8) *with multiplier $\bar{\lambda}(x^*, z^*)$ and $z^* = -g(x^*)$, then $(x^*, \bar{\lambda}(x^*, z^*))$ is a first-order solution to* (1.1). *Moreover, if (x^*,z^*) is a strong solution to* (2.8) *and d is sufficiently small, then $(x^*, \bar{\lambda}(x^*, z^*))$ is a strong solution to* (1.1).

The proof of this proposition is very similar to that of Proposition 2.1 and hence is omitted.

3. The iteration steps and the approximate merit functions. In our algorithm we generate a sequence of iterates, (x^k, z^k), where x^k is a current approximation to x^* and z^k is a corresponding approximation to the optimal slack vector $-g(x^*)$. At each iteration, we compute a step for updating the slack variables as follows: if δ^k is the step computed at x^k using the quadratic program (1.2), then the corresponding step for z^k is taken to be

$$q^k = -\left[\nabla g(x^k)^{\mathsf{t}}\delta^k + g(x^k) + z^k\right]; \tag{3.1}$$

i.e., $z^k + q^k$ is the slack vector for (1.2). We then update the pair of iterates by

$$(x^{k+1}, z^{k+1}) = (x^k, z^k) + \alpha\,(\delta^k, q^k)$$

for some steplength α determined by a line search using our merit function ψ_d. If δ^k is feasible for (1.2) and $\alpha \in (0, 1]$, then z^{k+1} is nonnegative (provided z^k is nonnegative). Thus the nonnegativity of the slack variables can easily be maintained as long as the linearized constraints are satisfied. In section 8 we consider a slightly different update for z when this is not the case.

A comment on the notation is in order: We denote the iterate by (x^k, z^k) and the step by (δ^k, q^k), whereas conventional notation would be to use

$$\begin{pmatrix} x^k \\ z^k \end{pmatrix} \text{ and } \begin{pmatrix} \delta^k \\ q^k \end{pmatrix}.$$

It should be clear from the context what is meant.

Because $\psi_d(x, z)$ involves the gradients of the objective function and the constraints, carrying out line searches for this function can be quite expensive, an especially critical factor in solving large-scale problems. Consequently, at each iterate generated by the algorithm, we identify a corresponding (local) *approximate merit function* to act as a surrogate for $\psi_d(x, z)$ in determining an appropriate α. These approximate merit functions, which are formed by keeping the gradient terms in $\psi_d(x, z)$ fixed, are more easily evaluated than $\psi_d(x, z)$. At the kth iterate the approximate merit function is defined as

$$\psi_d^k(x, z) = f(x) + \bar{c}(x, z)^{\mathsf{t}}\bar{\lambda}^k + \frac{1}{d}\bar{c}(x, z)^{\mathsf{t}}\bar{A}_k{}^{-1}\bar{c}(x, z), \tag{3.2}$$

where

$$\bar{A}_k = \nabla g(x^k)^{\mathsf{t}}\nabla g(x^k) + Z^k$$

and

$$\bar{\lambda}^k = -\bar{A}_k{}^{-1}\nabla g(x^k)^{\mathrm{t}}\nabla f(x^k).$$

The gradient formulas for ψ_d^k are somewhat simpler than those for ψ_d given in (2.15) and (2.16); namely,

$$\nabla_x\psi_d^k(x,z) = \nabla_x L(x,\bar{\lambda}^k) + \frac{2}{d}\nabla g(x)[\bar{A}_k]^{-1}\bar{c}(x,z) \tag{3.3}$$

and

$$\nabla_z\psi_d^k(x,z) = \bar{\lambda}^k + \frac{2}{d}[\bar{A}_k]^{-1}\bar{c}(x,z). \tag{3.4}$$

While simpler in form, the $\psi_d^k(x,z)$ cannot be used directly in a global convergence theory since they change from iterate to iterate. Nevertheless, they play a prominent role in our algorithm. A major part of this paper is devoted to demonstrating how these approximate merit functions can be used in conjunction with $\psi_d(x,z)$ to force convergence.

4. Basic assumptions. In proving global convergence we need to make some fundamental assumptions that will guarantee that our algorithm is well defined. Of course, these assumptions can rarely be assured in practice; therefore, some safeguards must be incorporated into an implementation of the algorithm to ensure that the algorithm will continue if a particular assumption fails to hold. In theory, the inclusion of these and other modifications may not guarantee global convergence, but the analysis here provides a firm foundation for our existing code as well as for future developments and enhancements of the algorithm.

The basic assumptions we make are the following:

A1. All points, (x^k, z^k), generated by the algorithm lie in $\mathcal{G}$, a compact set of $\mathcal{R}^n \times \mathcal{R}_+^m$, where $\mathcal{R}_+^m$ is the set of nonnegative m-dimensional vectors.

A2. The matrices used in (1.2) are chosen from $\mathcal{B}$, a compact set of positive definite $n \times n$ matrices.

A3. There exists a constant $K > 0$ such that for each $(x^k, z^k) \in \mathcal{G}$ and $B^k \in \mathcal{B}$ there is a solution, δ^k, to (1.2) and a corresponding multiplier vector, μ^k, that satisfy

$$\left\|(\delta^k, \mu^k)\right\| \leq K.$$

A4. For each $(x^k, z^k) \in \mathcal{G}$ the matrix $\bar{A}_k$ is positive definite.

A5. The set $\mathcal{S}$ is finite.

The implication of the first assumption is that all of our analysis will take place in the compact set $\mathcal{G}$. In particular, the sets $\mathcal{C}$ and $\mathcal{S}$ are considered to be subsets of $\mathcal{G}$. The first assumption is a strong condition; however, it is clear that virtually any minimization algorithm can, for certain problems, generate iterates that wander off to infinity following a path on which the function and infeasibility are decreasing. The alternative to making this assumption is to restrict the class of problems being considered (e.g., requiring (1.1) to be convex). As we offer our algorithm as an effective solution technique for *general* nonlinear programs, we prefer to require A1.

Although there are alternative procedures for choosing the matrices B^k, the use of positive definite matrices, while not ideal from a theoretical view (see the discussion in [5]), is perhaps the most popular because it simplifies the problem of solving the quadratic programming subproblem and is often necessary to obtain a descent direction for the merit function. Assumption A2 requires that there exist positive constants ρ_1 and ρ_2 such that for all $B \in \mathcal{B}$

$$\rho_1 \, \|y\|^2 \leq y^{\mathrm{t}} B \, y \leq \rho_2 \, \|y\|^2 \tag{4.1}$$

for all $y \in \mathcal{R}^n$. (Here and throughout the remainder of the paper, $\|\cdot\|$ refers to the ℓ_2 norm.) In our algorithm we have implemented provisions that allow the possibility of maintaining a positive definite approximation to the Hessian of the Lagrangian so that A2 is not a severe restriction. We have also successfully used the algorithm when B_k is not positive definite, but this case has not been thoroughly analyzed.

Since assumption A3 requires that each quadratic program be feasible, it is a fairly restrictive requirement. It is not uncommon for infeasible quadratic programs to be encountered in practical applications, especially in the event that the number of constraints greatly exceeds the number of variables, and so a useful implementation of an SQP algorithm must have a procedure that addresses this possibility. A discussion of our approach to this difficulty can be found in section 8. Given the feasibility of the quadratic programs, assumption A2 guarantees that a unique solution to the quadratic program must exist at each point of $\mathcal{G}$. Assumptions A2 and A3, however, do not guarantee a unique multiplier. In fact, unbounded multipliers may exist, but A3 does force a bounded choice. For example, the minimum norm multiplier could be used. The boundedness of the solution and a corresponding multiplier is used to ensure that the solution is a continuous function of the point (x^k,z^k) and the matrix B^k (Lemma 5.1). Note that the assumption is significantly weaker than the common assumption that the active constraint gradients of (1.2) be linearly independent; it also does not require strict complementary slackness for the multipliers.

To employ the merit function $\psi_d(x, z)$ (as well as the approximate merit functions) we must be sure that it is well defined, i.e., that the matrix $\bar{A}(x, z)$ is nonsingular. A4 is less restrictive than it might appear at first. To see this, we observe that since the nonnegativity of the slack variables will be maintained, the matrix $\bar{A}(x, z)$ will always be positive semidefinite. If we partition the index set of the constraints into two subsets a and u, we can write (without loss of generality)

$$g(x) = \begin{pmatrix} g_a(x) \\ g_u(x) \end{pmatrix}$$

and, in a corresponding manner,

$$z = \begin{pmatrix} z_a \\ z_u \end{pmatrix}.$$

Then $\bar{A}(x, z)$ is positive definite at (x, z) if $\nabla g_a(x)$ has full column rank and $z_u > 0$. If, for instance, the index set a corresponds to a set of linearly independent active constraint gradients for (1.2) we have only to require the slacks corresponding to the inactive constraints to be positive. We can also ensure the nonsingularity of $\bar{A}(x, z)$ by maintaining the positivity of the slack vector z, the easy implementation of which is guaranteed by the updating rule for these variables. (See section 9 for more details.)

Finally, the last assumption assures that all of the first-order solutions to (1.1) are isolated, which is the case of most interest.

5. Properties of the steps. In this section we prove some fundamental properties of the steps $\{(\delta^k, q^k)\}$ with respect to the merit function, the approximate merit functions, and feasibility. Recall that δ^k is obtained as a solution to (1.2) and q^k is given by (3.1). The first-order conditions for (1.2) are

$$B^k\delta^k + \nabla g(x^k)\mu^k = -\nabla f(x^k), \tag{5.1}$$

$$\nabla g(x^k)^{\mathrm{t}}\delta^k + g(x^k) \le 0, \tag{5.2}$$

$$\mu^k \ge 0, \tag{5.3}$$

$$\left[\nabla g(x^k)^{\mathrm{t}}\delta^k + g(x^k)\right]^{\mathrm{t}}\mu^k = 0, \tag{5.4}$$

where μ^k is an optimal multiplier vector. Using (5.1), (5.4), and the definitions of $\bar{A}_k$ and q^k, the following relation between μ^k and $\bar{\lambda}^k$, defined in (3.2), can be derived:

$$\bar{\lambda}^k - \mu^k = [\bar{A}_k]^{-1}\nabla g(x^k)^{\mathrm{t}}B^k\delta^k + [\bar{A}_k]^{-1}U_k\, q^k, \tag{5.5}$$

where $U_k = \operatorname{diag}\left\{\mu_1^k, \ldots, \mu_m^k\right\}$. The step is also related to the feasibility function $\bar{c}(x^k, z^k)$ as follows. Since at any iterate (x^k,z^k)

$$\nabla\bar{c}(x^k, z^k) = \begin{pmatrix} \nabla g(x^k) \\ I \end{pmatrix},$$

we have from (3.1) that

$$\bar{c}(x^k, z^k) = g(x^k) + z^k = -\left[\nabla g(x^k)^{\mathrm{t}}\delta^k + q^k\right]$$

and hence

$$\nabla\bar{c}(x^k, z^k)^{\mathrm{t}}(\delta^k, q^k) = \nabla g(x^k)^{\mathrm{t}}\delta^k + q^k = -\bar{c}(x^k, z^k). \tag{5.6}$$

We begin by showing that the step (δ^k, q^k) defined from (1.2) and (3.1) is a continuous function of the data and that $\mathcal{S}$ is just the set of points for which this step is zero.

Lemma 5.1. *The pair (δ^k,q^k) is a continuous function of (x^k,z^k,B^k) in $\mathcal{G} \times \mathcal{B}$.*

Proof. Let $\{(x^k,z^k)\}$ be a sequence in $\mathcal{G}$ converging to $(\hat{x}, \hat{z})$ and let $\{B^k\}$ in $\mathcal{B}$ converge to $\hat{B}$. Then, by assumption A3, for each k there exists a multiplier μ^k for (1.2) such that the sequence $\{(\delta^k, \mu^k)\}$ is bounded. Let $(\hat{\delta}, \hat{\mu})$ be a limit point of this sequence. Then there exist subsequences $\{(\delta^{k_j}, \mu^{k_j})\}$ satisfying (5.1)–(5.4) for x^{k_j} and B^{k_j}. Taking the limit it follows that $(\hat{\delta}, \hat{\mu})$ is an optimal solution pair at $\hat{x}$ and $\hat{B}$. The uniqueness of the solution of (1.2) establishes the continuity of δ^k, and the continuity of q^k follows immediately from (3.1). □

Proposition 5.2. *Let $\{(x^k, z^k)\}$ be a sequence of points in $\mathcal{G}$ and let $\{B^k\}$ be a sequence of matrices from $\mathcal{B}$. Suppose that $\{(x^k, z^k)\} \to (\hat{x}, \hat{z})$ and that the corresponding sequence $\{(\delta^k, q^k)\}$ obtained from solving* (1.2) *and choosing q^k by* (3.1) *has a subsequence converging to zero. Then $(\hat{x}, \hat{z}) \in \mathcal{S}$.*

Proof. Without loss of generality, assume $\{(\delta^k, q^k)\} \to (\hat{\delta}, \hat{q}) = (0, 0)$ and $\{B^k\} \to \hat{B}$. Then, by the preceding lemma, $\hat{\delta} = 0$ is the solution to (1.2) when $x = \hat{x}$ and $B = \hat{B}$ (in fact, for any $B \in \mathcal{B}$). Because $\hat{q} = 0$, it follows from (3.1) that $\hat{z} = -g(\hat{x})$. The multiplier vectors μ^k can be taken to be bounded by assumption A3 and hence (without loss of generality) to converge to a nonnegative vector $\hat{\mu}$ that satisfies the complementary slackness conditions for (1.2) at $\hat{x}$. Because $\hat{\delta} = 0$ these first-order conditions (5.1)–(5.4) imply that $(\hat{x}, \hat{\mu})$ satisfy the first-order conditions for (1.1) and hence $(\hat{x}, \hat{z}) \in \mathcal{S}$. □

If we assume that d is sufficiently small (section 9 contains a brief discussion of this assumption), then by Propositions 2.1 and 2.2 it follows that to solve (1.1) it is sufficient to find a solution to (2.8) that satisfies $\bar{c}(x,z) = 0$. If we are to obtain convergence to a solution of (2.8), then (δ^k, q^k) should either decrease the function ψ_d (and also ψ_d^k) or decrease infeasibility or both. The next propositions give conditions under which these objectives can be achieved.

The first of these propositions is a direct consequence of the definition of $r(x,z)$ and (5.6).

PROPOSITION 5.3. *Given $(x^k, z^k) \in \mathcal{G}$, we have*

$$\nabla r(x^k, z^k)^{\mathrm{t}} (\delta^k, q^k) = -2 r(x^k, z^k).$$

Since $r(x^k, z^k)$ is positive unless $(x^k, z^k) \in \mathcal{C}_0$ (i.e., unless the current iterate is already feasible), the proposition guarantees that moving in the direction of the step (δ^k, q^k) will initially decrease infeasibility.

The next proposition shows that the step (δ^k, q^k) is a descent direction for ψ_d^k at (x^k, z^k) as well. Prior to stating the result we prove a useful lemma.

LEMMA 5.4. *Let $\mathcal{W}$ be a compact subset of $\mathcal{R}^p$. For $d > 0$ define the function $k_d(w,y)$ for $w \in \mathcal{W}$ and $y = (y_1, y_2) \in \mathcal{R}^n \times \mathcal{R}^m$ by*

$$k_d(w,y) = -\zeta_0 \|y_1\|^2 + \zeta_1 \|y\| \, \|E(w)\, y_1 + y_2\| - \frac{\zeta_2}{d} \|E(w)\, y_1 + y_2\|^2,$$

where ζ_0, ζ_1, and ζ_2 are positive constants and $E(w)$ is a continuous matrix-valued function. Then there exist positive constants $\bar{d}$ and $\bar{\kappa}$ such that for $d \le \bar{d}$

$$k_d(w,y) \le -\,\bar{\kappa}\, \|y\|^2$$

for all $w \in \mathcal{W}$ and $y \in \mathcal{R}^n \times \mathcal{R}^m$.

Proof. Because of the form of $k_d(w,y)$, we need only show that there exists a $\bar{\kappa} > 0$ such that $k_d(w,y) \le \bar{\kappa}$ for all $w \in \mathcal{W}$ and $y \in \mathcal{R}^n \times \mathcal{R}^m$ with $\|y\| = 1$. We define

$$M_1 = \{ y : \, E(w)\, y_1 + y_2 = 0, \|y\| = 1 \}$$

and

$$M_2 = \{ y : \, y_1 = 0, \|y_2\| = 1 \}.$$

For $y \in M_1$ we have that the right-hand side of $k_d(w,y)$ is equal to $-\,\zeta_0\, \|y_1\|^2$. Since M_1 and M_2 are compact and disjoint, there is a positive ξ such that $\|y_1\| > \xi$ for all $y \in M_1$. It follows that for some ν sufficiently small

$$k_d(w,y) \le -\frac{\zeta_0\, \xi^2}{2} \tag{5.7}$$

for y in the set

$$M_\nu = \{ y : \, \|y\| = 1 \text{ and } \|y - \hat{y}\| \le \nu \text{ for some } \hat{y} \in M_1 \}.$$

Moreover,

$$\epsilon = \min \{ \|E(w)\, y_1 + y_2\| : y \notin M_\nu, \|y\| = 1 \} > 0$$

so that for $y \notin M_\nu$ and $\|y\| = 1$,

$$k_d(w, y) \leq -\zeta_0 \, \|y_1\|^2 + \zeta_1 \, \|E(w)\, y_1 + y_2\| - \frac{\zeta_2}{d}\epsilon^2. \tag{5.8}$$

Now letting $\chi = \max \{\|E(w)\, y_1 + y_2\| : \|y\| = 1\}$ we see that for $y \notin M_\nu$ and for

$$d \leq \frac{\zeta_2 \, \epsilon^2}{2 \, \zeta_1 \, \chi},$$

the right-hand side of (5.8) is less than $-\zeta_1 \, \chi$. The lemma follows from this inequality and (5.7). □

PROPOSITION 5.5. *There exist positive constants $\bar{d}$ and κ such that for $d \leq \bar{d}$ and for all $(x^k, z^k) \in \mathcal{G}$*

$$\nabla \psi_d^k(x^k, z^k)^{\mathsf{t}} (\delta^k, q^k) \leq -\kappa \, \left\|(\delta^k, q^k)\right\|^2.$$

Proof. Using (3.3) and (3.4) we have

$$\begin{aligned} \triangle \equiv \nabla \psi_d^k(x^k, z^k)^{\mathsf{t}} (\delta^k, q^k) = {} & \nabla_x L(x^k, \bar{\lambda}^k)^{\mathsf{t}} \delta^k + (\bar{\lambda}^k)^{\mathsf{t}} q^k \\ & + \frac{2}{d} \bar{c}(x^k, z^k)^{\mathsf{t}} [\bar{A}_k]^{-1} (\nabla g(x^k)^{\mathsf{t}} \delta^k + q^k). \end{aligned}$$

Now

$$\nabla_x L(x^k, \bar{\lambda}^k) = \nabla_x L(x^k, \mu^k) + \nabla g(x^k)(\bar{\lambda}^k - \mu^k).$$

Hence, using (5.6) and (5.1), we can write

$$\begin{aligned} \triangle = {} & -(\delta^k)^{\mathsf{t}} B^k \delta^k + (\bar{\lambda}^k - \mu^k)^{\mathsf{t}} (\nabla g(x^k)^{\mathsf{t}} \delta^k + q^k) \\ & + (\mu^k)^{\mathsf{t}} q^k - \frac{2}{d} (\nabla g(x^k)^{\mathsf{t}} \delta^k + q^k)^{\mathsf{t}} [\bar{A}_k]^{-1} (\nabla g(x^k)^{\mathsf{t}} \delta^k + q^k). \end{aligned}$$

We have, from the definition of q^k and (5.4), that

$$(\mu^k)^{\mathsf{t}} q^k = -(\mu^k)^{\mathsf{t}} (\nabla g(x^k)^{\mathsf{t}} \delta^k + g(x^k) + z^k) = -(\mu^k)^{\mathsf{t}} z^k \leq 0.$$

Thus, from (4.1), (5.5), and assumptions A1–A3, we obtain the inequality

$$\begin{aligned} \triangle \leq {} & -\rho_1 \left\|\delta^k\right\|^2 \\ & + \xi \, \left\|(\delta^k, q^k)\right\| \, \left\|\nabla g(x^k)^{\mathsf{t}} \delta^k + q^k\right\| \\ & - \frac{2}{d} (\nabla g(x^k)^{\mathsf{t}} \delta^k + q^k)^{\mathsf{t}} [\bar{A}_k]^{-1} (\nabla g(x^k)^{\mathsf{t}} \delta^k + q^k). \end{aligned}$$

Now the preceding lemma can be applied to the term on the right with $w = (x, z, B)$, $W = \mathcal{G} \times \mathcal{B}$, and $y = (\delta, q)$ to obtain the desired result. □

The preceding result is quite strong and will play an important role in our convergence theory. In addition to demonstrating that (δ^k, q^k) is a descent direction of ψ_d^k, it gives a useful bound on the rate of decrease in that direction. The step (δ^k, q^k) does not have this same global property with respect to the true merit function $\psi_d(x, z)$. However, near feasibility, a similar result can be obtained. The required proximity to $\mathcal{C}_0$ depends on d (see (2.6)).

PROPOSITION 5.6. *For each sufficiently small positive d there exist positive constants $\eta(d)$ and $\hat{\kappa}$ such that*

$$\nabla\psi_d(x^k,z^k)^{\mathsf{t}}(\delta^k,q^k) \le -\,\hat{\kappa}\,\left\|(\delta^k,q^k)\right\|^2$$

for each $(x^k,z^k) \in \mathcal{C}_{\eta(d)}$.

Proof. Observe that $\left\|\bar{c}(x^k,z^k)\right\| \le K\,\left\|(\delta^k,q^k)\right\|$ for some constant K. Thus it is seen from (2.15)–(3.4) that for some constant $\hat{K}$

$$\begin{aligned}\hat{\triangle} &\equiv \nabla\psi_d(x^k,z^k)^{\mathsf{t}}(\delta^k,q^k)\\ &= \nabla\psi_d^k(x^k,z^k)^{\mathsf{t}}(\delta^k,q^k) + \bar{c}(x^k,z^k)^{\mathsf{t}}\nabla\bar{\lambda}(x^k,z^k)^{\mathsf{t}}(\delta^k,q^k)\\ &\quad+\frac{1}{d}\hat{K}\left\|(\delta^k,q^k)\right\|^2\left\|\bar{c}(x^k,z^k)\right\|.\end{aligned}$$

Using the inequalities from the preceding proposition and (5.6) as well as A3, we obtain

$$\begin{aligned}\hat{\triangle} &\le -\rho_1\left\|\delta^k\right\|^2 + \hat{\xi}\,\left\|(\delta^k,q^k)\right\|\,\left\|\nabla g(x^k)^{\mathsf{t}}\delta^k + q^k\right\|\\ &\quad-\frac{2}{d}(\nabla g(x^k)^{\mathsf{t}}\delta^k + q^k)^{\mathsf{t}}[\bar{A}_k]^{-1}(\nabla g(x^k)^{\mathsf{t}}\delta^k + q^k)\\ &\quad+\frac{1}{d}\hat{K}\left\|(\delta^k,q^k)\right\|^2\left\|\bar{c}(x^k,z^k)\right\|.\end{aligned}$$

The lemma can now be applied to the first three terms on the right-hand side as in Proposition 5.5 to obtain

$$\hat{\triangle} \le -\kappa\,\left\|(\delta^k,q^k)\right\|^2 + \frac{1}{d}\hat{K}\left\|(\delta^k,q^k)\right\|^2\sqrt{\eta}$$

for d sufficiently small. If for fixed d we choose $\eta(d)$ so that

$$\eta(d) \le \left(\frac{d\,\kappa}{2\,\hat{K}}\right)^2,$$

we get the desired result with $\hat{\kappa} = \kappa/2$. □

The above propositions show that near $\mathcal{C}_0$ the direction (δ^k,q^k) is a descent direction for both the approximate and the true merit function, and both functions have the same rate of decrease. The next proposition shows that these directional derivatives are indeed nearly identical provided that the iterate is close to feasibility but away from the set of first-order solutions, $\mathcal{S}$.

PROPOSITION 5.7. *For every $\beta > 1$ and every $\epsilon > 0$, there exists a positive constant $\eta(\beta,\epsilon)$ such that*

$$\begin{aligned}\beta\nabla\psi_d(x^k,z^k)^{\mathsf{t}}(\delta^k,q^k) &\le \nabla\psi_d^k(x^k,z^k)^{\mathsf{t}}(\delta^k,q^k)\\ &\le \frac{1}{\beta}\nabla\psi_d(x^k,z^k)^{\mathsf{t}}(\delta^k,q^k)\end{aligned}$$

for $(x^k,z^k) \in \mathcal{C}_{\eta(\beta,\epsilon)} - \mathcal{S}_\epsilon$ and for any choice of $B^k \in \mathcal{B}$.

Proof. For a given $\epsilon > 0$ and $\eta_0 > 0$, let

$$\nu_\epsilon = \min\left\{\left\|(\delta^k,q^k)\right\| : (x^k,z^k) \in \mathcal{C}_{\eta_0} - \mathcal{S}_\epsilon,\ B^k \in \mathcal{B}\right\}$$

and note that ν_ϵ is positive by virtue of Proposition 5.2. There exists a constant K independent of (x^k, z^k) such that for $(x^k, z^k) \in \mathcal{C}_{\eta_0} - \mathcal{S}_\epsilon$ and (x^+, z^+) a corresponding closest point in $\mathcal{C}_0$,

$$[\nabla\psi_d(x^k, z^k) - \nabla\psi_d(x^+, z^+)]^{\mathrm{t}}(\delta^k, q^k) \leq K \left\|(x^k, z^k) - (x^+, z^+)\right\| \left\|(\delta^k, q^k)\right\|$$

and

$$[\nabla\psi_d^k(x^k, z^k) - \nabla\psi_d^k(x^+, z^+)]^{\mathrm{t}}(\delta^k, q^k) \leq K \left\|(x^k, z^k) - (x^+, z^+)\right\| \left\|(\delta^k, q^k)\right\|.$$

From (2.15)–(2.16) it is seen that

$$\nabla\psi_d^k(x^+, z^+) = \nabla\psi_d(x^+, z^+).$$

Hence,

$$\begin{aligned} \triangle &\equiv |\nabla\psi_d(x^k, z^k)^{\mathrm{t}}(\delta^k, q^k) - \nabla\psi_d^k(x^k, z^k)^{\mathrm{t}}(\delta^k, q^k)| \\ &\leq 2\,K \left\|(x^k, z^k) - (x^+, z^+)\right\| \left\|(\delta^k, q^k)\right\|. \end{aligned}$$

Because of the compactness of $\mathcal{C}_0$ there exists a continuous function $\theta(\eta)$ such that $\theta(0) = 0$, and if $(x^k, z^k) \in \mathcal{C}_\eta$, then $\left\|(x^k, z^k) - (x^+, z^+)\right\| \leq \theta(\eta)$. Now, if $(x^k, z^k) \in \mathcal{C}_\eta - \mathcal{S}_\epsilon$ for $\eta \leq \eta_0$, we have from the definition of ν_ϵ in Proposition 5.5 that

$$\begin{aligned} \triangle &\leq \frac{2\,K}{\nu_\epsilon} \left\|(x^k, z^k) - (x^+, z^+)\right\| \left\|(\delta^k, q^k)\right\|^2 \\ &\leq -\frac{2\,K}{\nu_\epsilon\,\kappa} \left\|(x^k, z^k) - (x^+, z^+)\right\| \nabla\psi_d^k(x^k, z^k)^{\mathrm{t}}(\delta^k, q^k) \\ &\leq -\frac{2\,K\,\theta(\eta)}{\nu_\epsilon\,\kappa} \nabla\psi_d^k(x^k, z^k)^{\mathrm{t}}(\delta^k, q^k). \end{aligned}$$

The proposition now follows for a given β by choosing η sufficiently small. □

The preceding propositions ensure that at (x^k, z^k) the step generated by solving the quadratic program and using (3.1) is a descent direction for r, ψ_d^k, and, if (x^k, z^k) is close enough to the feasible set, ψ_d. In our algorithm we take a step in the direction of (δ^k, q^k) and choose a steplength so that the new point is a satisfactory choice for (x^{k+1}, z^{k+1}). That is, we will set

$$(x_\alpha, z_\alpha) = (x^k, z^k) + \alpha(\delta^k, q^k)$$

for some appropriate choice of α. In unconstrained optimization a standard criterion for ensuring that a *sufficient* relative decrease is obtained for an objective function $\phi(w)$ in a descent direction v is that the steplength α satisfy

$$\phi(w + \alpha\, v) - \phi(w) \leq \sigma\,\alpha\,\nabla\phi(w)^{\mathrm{t}}v \tag{5.9}$$

for some constant $\sigma \in (0, 1/2)$. This is often called the Goldstein–Armijo condition (see [17]). We use this test for both decreasing infeasibility (as measured by r) and for moving toward optimality (as measured by ψ_d^k and ψ_d).

The requirement that infeasibility be decreased will be imposed when (x^k, z^k) is outside of $\mathcal{C}_\eta$ for the current value of η. At that point we will require that the step from (x^k, z^k) to (x^{k+1}, z^{k+1}) should yield a sufficient relative decrease in r by choosing α so that

$$\begin{aligned} r(x_\alpha, z_\alpha) &\leq r(x^k, z^k) + \alpha\,\sigma\,\nabla r(x^k, z^k)^{\mathrm{t}}(\delta^k, q^k) \\ &= (1 - 2\,\sigma\,\alpha) r(x^k, z^k), \end{aligned}$$

where the equality follows from Proposition 5.3.

When the current iterate, (x^k, z^k), is far from feasibility our algorithm will apply condition (5.9) to force a sufficient relative decrease in the approximate merit function. That is, we require

$$\psi_d^k(x_\alpha, z_\alpha) - \psi_d^k(x^k, z^k) \le \sigma\,\alpha\,\nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k), \tag{5.10}$$

which can be satisfied because of Proposition 5.5. The step direction may not be a descent direction for ψ_d in general, but it is close to feasibility (Proposition 5.6). The next proposition shows that satisfying (5.10) at (x^k,z^k) guarantees that the same α satisfies a similar condition for ψ_d provided $(x^k, z^k) \in \mathcal{C}_\eta - \mathcal{S}_\epsilon$ for certain values of ϵ and η.

Proposition 5.8. *Choose $\sigma \in (0, 1/2)$ and suppose that for $(x^k, z^k) \in \mathcal{C}_\eta$, α is chosen so that $(x_\alpha, z_\alpha) \in \mathcal{C}_\eta$ and (5.10) holds. Then, for each $\epsilon > 0$ and $\gamma \in (0, 1)$, there exists an $\eta(\epsilon, \gamma)$ such that if $\eta < \eta(\epsilon, \gamma)$ and $(x^k, z^k) \in \mathcal{C}_\eta - \mathcal{S}_\epsilon$,*

$$\psi_d(x_\alpha, z_\alpha) - \psi_d(x^k, z^k) \le \gamma\,\sigma\,\alpha\,\nabla\psi_d(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k). \tag{5.11}$$

In addition, for each η sufficiently small but fixed and $\gamma \in (0, 1)$, there exists an $\epsilon(\eta, \gamma)$ such that if $\epsilon > \epsilon(\eta, \gamma)$ and $(x^k, z^k) \in \mathcal{C}_\eta - \mathcal{S}_\epsilon$, then (5.11) also holds.

Proof. Suppose $(x^k, z^k) \in \mathcal{C}_\eta$ and α is chosen so that the hypotheses hold. Then, since $\psi_d(x^k, z^k) = \psi_d^k(x^k, z^k)$, we have

$$\begin{aligned}
\triangle &\equiv \psi_d(x_\alpha, z_\alpha) - \psi_d(x^k, z^k) \\
&= \psi_d^k(x_\alpha, z_\alpha) - \psi_d^k(x^k, z^k) + \ \left[\psi_d(x_\alpha, z_\alpha) - \psi_d^k(x_\alpha, z_\alpha)\right] \\
&\le \sigma\,\alpha\,\nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k) \\
&\quad + \ \left[\psi_d(x_\alpha, z_\alpha) - \psi_d^k(x_\alpha, z_\alpha)\right].
\end{aligned}$$

Now, from the definitions of ψ_d and ψ_d^k, we have that

$$\begin{aligned}
\psi_d(x_\alpha, z_\alpha) - \psi_d^k(x_\alpha, z_\alpha) &= c(x_\alpha, z_\alpha)^{\mathsf{t}}\,[\bar\lambda(x_\alpha, z_\alpha) - \bar\lambda^k\,] \\
&\qquad + \frac{1}{d} c(x_\alpha, z_\alpha)^{\mathsf{t}}\,[\bar A(x_\alpha, z_\alpha)^{-1} - \bar A^k] c(x_\alpha, z_\alpha),
\end{aligned}$$

and hence for η sufficiently small, there is a constant K independent of (x^k, z^k) such that

$$|\psi_d(x_\alpha, z_\alpha) - \psi_d^k(x_\alpha, z_\alpha)| \le K\,\alpha\,\sqrt{\eta}\,\left(1 + \frac{\sqrt{\eta}}{d}\right).$$

Now suppose that $\epsilon > 0$ and η_0 are fixed. Then for d and $\beta > 1$ fixed and η small we have, using Proposition 5.7 and the definition of ν_ϵ in its proof and Proposition 5.5,

$$\begin{aligned}
\triangle &\le \frac{\sigma\,\alpha}{\beta}\,\nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k) + \hat K\,\alpha\,\sqrt{\eta}\,\frac{\left\|(\delta^k, q^k)\right\|^2}{\nu_\epsilon^2} \\
&\le \frac{\alpha\,\sigma}{\beta}\,\left[1 - \frac{\hat K\,\sqrt{\eta}}{\sigma\,\nu_\epsilon^2\,\kappa}\right]\,\nabla\psi_d(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k).
\end{aligned}$$

The proof now follows by choosing η sufficiently small or ϵ sufficiently large. □

In summary, we have shown that:

- First-order points of (1.1) correspond to first-order points of (2.8) and, specifically, strong solutions of (1.1) correspond to strong solutions of (2.8) for d small enough. Thus, reducing $\psi_d(x, z)$ while keeping z nonnegative implies that progress toward the solution is being made.
- The steps (δ, q) are continuous functions of (x, z) and only vanish at first-order points of (1.1). Thus an algorithm based on these steps with steplengths bounded away from zero and bounded above cannot "stall" before reaching first-order points.
- There is a tube $\mathcal{C}_\eta$ around the feasible region in which the step (δ, q) is a descent direction for the true merit function $\psi_d(x, z)$, the approximate merit function $\psi_d^k(x, z)$, and the function r (r gives a measure of infeasibility). Furthermore, a sufficient relative decrease in the approximate merit function implies a sufficient relative decrease in the true merit function, except possibly in a small ball around first-order points. This last point is essential in our convergence analysis.
- Outside of the η-tube, the approximate merit function and the function r are reduced by the step (δ, q), but this implies that the iterates can be forced into $\mathcal{C}_\eta$ and suggests an adaptive procedure for determining an appropriate η.

These results form the basis for the algorithm described next.

6. The basic algorithm.

In this section we give a description of our algorithm and comment further on its motivation. The underlying idea of the algorithm is to use the descent properties of the step (δ, q) with respect to $\psi_d(x, z)$, $\psi_d^k(x, z)$, and r to determine dynamically an appropriate value of η and to ensure that the iterates remain in the η-tube. Global convergence of this algorithm is shown in section 7. What distinguishes this algorithm is the use of the approximate merit functions that, far from feasibility, determine efficient steplengths that are likely to force the iterates toward optimality as well as feasibility and, near feasibility, provide relatively simple surrogates for the true merit function. A further distinguishing factor is that, unlike some other algorithms, we do not require reduction of infeasibility at every step.

In the description of the algorithm, it is assumed that $d > 0$ is sufficiently small and that constants $\nu > 0$, but sufficiently small, and $\sigma \in (0, 1/2)$ have been specified. Recall that

$$(x_\alpha, z_\alpha) = (x^k, z^k) + \alpha(\delta^k, q^k).$$

Algorithm.

1. Given x^0, λ^0, B^0, and $z^0 \geq 0$, set $k = 0$ and $\eta = r(x^k, z^k)$.
2. Compute (δ^k, q^k).
3. If $(x^k, z^k) \notin \mathcal{C}_\eta$, then compute α by backtracking line search starting at 1 such that

   $$\psi_d^k(x_\alpha, z_\alpha) \leq \psi_d^k(x^k, z^k) + \alpha\, \sigma\, \nabla \psi_d^k(x^k, z^k)^{\mathsf{t}} (\delta^k, q^k)$$

   and

   $$\begin{aligned} r(x_\alpha, z_\alpha) &\leq r(x^k, z^k) + \alpha\, \sigma\, \nabla r(x^k, z^k)^{\mathsf{t}} (\delta^k, q^k) \\ &\leq (1 - 2\, \sigma\, \alpha)\, r(x^k, z^k); \end{aligned}$$

   go to step 6.

4. If $(x^k, z^k) \in \mathcal{C}_\eta$, then compute α by backtracking line search starting at 1 such that

$$\psi_d^k(x_\alpha, z_\alpha) \le \psi_d^k(x^k, z^k) + \alpha\, \sigma\, \nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k)$$

and

$$(x_\alpha, z_\alpha) \in \mathcal{C}_\eta.$$

5. If

$$\psi_d(x_\alpha, z_\alpha) > \psi_d(x^k, z^k) + \tfrac{1}{2}\alpha\, \sigma\, \nabla\psi_d(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k)$$

or

$$\nabla\psi_d(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k) > -\nu \left\|(\delta^k, q^k)\right\|^2,$$

then set $\eta = \frac{1}{2}r(x^k, z^k)$.

6. Set $(x^{k+1}, z^{k+1}) = (x_\alpha, z_\alpha)$; update B^k; set $k = k + 1$; return to step 2.

The crucial parts of the algorithm are the procedures for picking the steplength α and adjusting the parameters η and d. The results of the previous section suggest our choices. Here, we make the assumption that the parameter d is initially small enough that the basic propositions of the preceding section are satisfied. In the actual implementation of our algorithm we do have a heuristic procedure for adjusting d (see section 9), but for the theoretical convergence analysis given here we do not include this modification.

Note that the steplength parameter α is always chosen from $(0, 1]$ by a backtracking method; this assures that the step will not become too small and also that the variable z will remain nonnegative. The specific criteria for choosing α depend on where the iterate is relative to the current value of η. We always require that condition (5.10) be satisfied. If $(x^k, z^k) \notin \mathcal{C}_\eta$, then we also require that the Goldstein–Armijo condition be satisfied for the function $r(x, z)$. By Propositions 5.3 and 5.5, this can always be done.

If $(x^k, z^k) \in \mathcal{C}_\eta$, then (in step 4) we also require that $(x_\alpha, z_\alpha) \in \mathcal{C}_\eta$; i.e., we do not allow the iterates to leave the η-tube once having entered it. Observe that this does not require r to be reduced at each iteration, but rather allows the algorithm the flexibility to increase and decrease r inside the η-tube. The computed step is then tested to see if the true merit function $\psi_d(x, z)$ satisfies the Goldstein–Armijo condition for the constant $\sigma/2$ (step 5). If η is small enough, then Proposition 5.8 ensures that such a decrease will occur (if (x^k, z^k) is not too close to the solution set $\mathcal{S}$). If the condition for $\psi_d(x, z)$ is not satisfied for the value of α, then we take this as a signal that the current value of η is too large and we decrease η to one-half the current value of $r(x^k, z^k)$. Thus, when the value of η is decreased, it is decreased by at least a factor of one-half, so that either the sequence of η values tends to zero or else the Goldstein–Armijo condition for $\psi_d(x, z)$ is eventually satisfied for all iterates. Technically, the satisfaction of the first test in step 5 does not guarantee that η is small enough for the conclusion of Proposition 5.6 to hold. The second test is included to ensure that the step has the desired properties. (See section 9.4.)

7. A global convergence theorem.

In this section we state and prove the main result of this paper, namely, that under appropriate conditions the sequence of iterates generated by the algorithm of the preceding section will converge to a first-order solution of (1.1).

In proving the convergence, the following standard results (see, e.g., [16]) are crucial. They establish the convergence properties of a descent algorithm under certain conditions on the steps. For the statement of these lemmas, we assume that $\phi(w)$ is a smooth function bounded below with bounded level sets. Given an initial w^0, the sequence of iterates $\{w^k\}$ is generated according to

$$w^{k+1} = w^k + \alpha_k v^k,$$

where the v^k satisfy

$$\nabla\phi(w^k)^{\mathsf{t}} v^k < 0 \tag{7.1}$$

for each k.

Lemma 7.1. *Let $\sigma \in (0, 1/2)$, $1 \geq \alpha^* > 0$, and $\xi > 1$ be given. Suppose that the α_k are determined by a backtracking line search, i.e.,*

$$\alpha_k = \max\left\{\alpha \in \mathcal{A} \ : \ \phi(w^k + \alpha\, v^k) \leq \phi(w^k) + \sigma\, \alpha\, \nabla\phi(w^k)^{\mathsf{t}} v^k\right\}, \tag{7.2}$$

where

$$\mathcal{A} = \{\alpha : \alpha = \xi^{-l}\, \alpha^*, l = 0, 1, 2, \ldots\}.$$

If there are positive constants ρ and γ such that for each k the v^k satisfy

$$\nabla\phi(w^k)^{\mathsf{t}} v^k \leq -\rho\, \left\|\nabla\phi(w^k)\right\| \left\|v^k\right\| \tag{7.3}$$

and

$$\left\|v^k\right\| \geq \gamma \left\|\nabla\phi(w^k)\right\|, \tag{7.4}$$

then

$$\lim_{k\to\infty} \left\|\nabla\phi(w^k)\right\| = 0. \tag{7.5}$$

Lemma 7.2. *Let $\sigma \in (0, 1/2)$ and assume that the α_k are chosen so that at each iteration*

$$\phi(w^{k+1}) \leq \phi(w^k) \tag{7.6}$$

and that there are an infinite subsequence $\{k_j\}$ and positive constants ρ and ν such that for each k_j

$$\alpha_{k_j} \geq \nu, \tag{7.7}$$

$$\phi(w^{k_j+1}) - \phi(w^{k_j}) \leq \sigma\, \alpha\, \nabla\phi(w^{k_j})^{\mathsf{t}} v^{k_j}, \tag{7.8}$$

and

$$\nabla\phi(w^{k_j})^{\mathsf{t}} v^{k_j} \leq -\rho \left\|v^{k_j}\right\|^2. \tag{7.9}$$

Then

$$\lim_{j\to\infty} \left\|v^{k_j}\right\| = 0. \tag{7.10}$$

In the algorithm, the iterates must eventually enter the set $\mathcal{C}_\eta$ for the current value of η. The following lemma states that this takes a finite number of steps.

LEMMA 7.3. *Given a fixed value of η, if the current iterate, (x^k,z^k), is not in $\mathcal{C}_\eta$ then the iterates will reach $\mathcal{C}_\eta$ in a finite number of steps.*

Proof. Assume that $(x^k, z^k) \notin \mathcal{C}_\eta$ for all k. From Proposition 5.3 we see that

$$\nabla r(x^k, z^k)^{\mathrm{t}}(\delta^k, q^k) = -2\, r(x^k, z^k) \leq -2\, \eta$$

for all k. Since $\left\|(\delta^k, q^k)\right\|$ and $\left\|\nabla r(x^k, z^k)\right\|$ are bounded away from zero for $(x^k, z^k) \notin \mathcal{C}_\eta$, this inequality implies that conditions (7.1), (7.3), and (7.4) are satisfied for the function $r(x, z)$. From the choice of α in step 3 of the algorithm, it follows that the hypotheses of Lemma 7.1 are satisfied for $r(x, z)$. Then $\nabla r(x^k, z^k)$ tends to zero which, by the above equality, forces $r(x^k, z^k)$ to zero, thus contradicting the assumption. □

We note that in step 4 of the algorithm, when $(x^k, z^k) \in \mathcal{C}_\eta$, α is chosen by a backtracking search so that (5.10) is satisfied and also so that the new iterate will remain in $\mathcal{C}_\eta$. Obviously, both of these conditions can be satisfied for α small enough; however, it will be important for the convergence proof to have the steplengths not get too small. The next two lemmas give lower bounds on the steplengths for these two conditions. The first shows that a steplength of $O(\sqrt{\eta})$ will suffice to keep $(x_\alpha, z_\alpha) \in \mathcal{C}_\eta$, while the second shows that (5.10) can be satisfied by a steplength bounded away from zero for all $(x^k, z^k) \in \mathcal{C}_\eta$.

LEMMA 7.4. *Let $\eta > 0$ be given and for $(x^k, z^k) \in \mathcal{C}_\eta$ let*

$$\zeta^k = \sup\{\bar{\alpha} : (x_\alpha, z_\alpha) \in \mathcal{C}_\eta \textit{ for } \alpha \in (0, \bar{\alpha}]\}.$$

Set

$$\zeta^* = \inf\{\zeta^k : (x^k, z^k) \in \mathcal{C}_\eta\}.$$

Then $\zeta^* > 0$.

Proof. By the Taylor series expansion of r, the compactness of $\mathcal{C}_\eta$, and Lemma 5.1, there exists a positive constant Γ depending only on η, such that for $(x^k, z^k) \in \mathcal{C}_\eta$

$$r(x_\alpha, z_\alpha) \leq r(x^k, z^k) + \alpha\, \nabla r(x^k, z^k)^{\mathrm{t}}(\delta^k, q^k) + \alpha^2 \Gamma.$$

But from Proposition 5.3 we have

$$r(x_\alpha, z_\alpha) \leq (1 - 2\alpha)\, \eta + \alpha^2\, \Gamma.$$

Thus for

$$\alpha \leq \frac{2\sqrt{\eta}}{\Gamma},$$

we have $(x_\alpha, z_\alpha) \in \mathcal{C}_\eta$. □

LEMMA 7.5. *There exists a positive constant ζ such that if $(x^k, z^k) \in \mathcal{C}_\eta$ and α is chosen by a backtracking line search so that* (5.10) *is satisfied, then*

$$\alpha \geq \zeta.$$

Proof. Suppose that α_k is chosen by a backtracking line search starting at 1 (see Lemma 7.1) so that

$$\psi_d^k(x_\alpha, z_\alpha) \le \psi_d^k(x^k, z^k) + \alpha_k\, \sigma\, \nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k).$$

It follows from the definition of the backtracking method that if $\alpha_k \ne 1$, then

$$\psi_d^k((x^k, z^k) + \xi\, \alpha_k\, (\delta^k, q^k)) - \psi_d^k(x^k, z^k) \ge \xi\, \sigma\alpha_k\, \nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k). \tag{7.11}$$

On the other hand, from the smoothness of $\psi_d^k(x,z)$ and the compactness of $\mathcal{C}_\eta$ we have

$$\begin{aligned} -\psi_d^k((x^k, z^k) + \xi\, \alpha_k\, (\delta^k, q^k)) + \psi_d^k(x^k, z^k) \\ \ge -\,\xi\, \alpha_k\, \nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k) - \xi\, \alpha_k^2\, \Gamma\, \left\|(\delta^k, q^k)\right\|^2 \end{aligned} \tag{7.12}$$

for some constant Γ that is independent of k. Adding (7.11) and (7.12) and simplifying give

$$\alpha_k \ge \frac{-(1-\sigma)}{\Gamma}\ \frac{\nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k)}{\left\|(\delta^k, q^k)\right\|^2}. \tag{7.13}$$

Combining (7.13) with Proposition 5.5 yields

$$\alpha_k \ge \frac{\kappa\,(1-\sigma)}{\Gamma},$$

which is the desired result. □

We are now ready to prove the main theorem.

THEOREM 7.6. *Assume* A1–A5 *and that d and ν are sufficiently small. Then the sequence of iterates $\{(x^k, z^k)\}$ converges to a point (x^*, z^*) in $\mathcal{S}$; i.e., x^* is the x-coordinate of a first-order solution of* (1.1).

Proof. There are two cases to consider.

Case 1. There is a positive number η^* that is the smallest value of η attained in the algorithm. Then, from some fixed index on, all of the iterates lie in $\mathcal{C}_{\eta^*}$ and the conditions

$$\begin{aligned} \psi_d(x^k + \alpha\, \delta^k, z^k + \alpha\, q^k) \le \psi_d(x^k, z^k) \\ +\ \tfrac{1}{2}\alpha\, \sigma\, \nabla\psi_d(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k) \end{aligned} \tag{7.14}$$

and

$$\nabla\psi_d(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k) \le -\nu\, \left\|(\delta^k, q^k)\right\|^2 \tag{7.15}$$

are satisfied. To apply Lemma 7.2 to the function ψ_d^k, we see that, by virtue of the way the α_k are chosen, Lemma 7.5 implies that the α_k are bounded away from zero and so (7.7) holds. Moreover, (7.1), (7.6), (7.8), and (7.9) are direct consequences of (7.14), (7.15), and Proposition 5.5. Thus, by Lemma 7.2, $\{(\delta^k, q^k)\} \to 0$ and the result follows from Proposition 5.2.

Case 2. At infinitely many of the iterates the value of η is changed. Since the size of η is cut in half at each change, the values of η tend to zero and all of the limit points of the sequence of iterates $\{(x^k, z^k)\}$ lie in $\mathcal{C}_0$. Therefore, by Proposition 5.6, the second condition in step 4 of the algorithm is satisfied for all k sufficiently large and the Goldstein–Armijo condition on ψ must fail infinitely often. Denote by $\{k_j\}$ the sequence of indices for which (7.14) fails and by η_j the values of η at these iterates. From Proposition 5.8 it follows that for each j there is an ϵ_j such that

$$(x^{k_j}, z^{k_j}) \in \mathcal{S}_{\epsilon_j} \cap \mathcal{C}_{\eta_j}$$

and $\epsilon_j \to 0$ as $j \to \infty$. It follows that at least one of the points in $\mathcal{S}$ is a limit point of $\{x^k, z^k\}$. Let L denote the set of points in $\mathcal{S}$ that are limit points of the sequence of iterates. Let $\epsilon > 0$ be given. It follows from A5 that there is a constant K_1 such that for each $(\tilde{x}, \tilde{z}) \in L$ and each (x, z), with $||(x, z) - (\tilde{x}, \tilde{z})|| < \epsilon$,

$$|\psi_d(x, z) - \psi_d(\tilde{x}, \tilde{z})| \le K_1 \, \epsilon.$$

Moreover, assumption A1, Proposition 5.6, and the fact that the steplength parameter is bounded by 1 ensure that there is a constant K_2 such that

$$|\psi_d(x^{k+1}, z^{k+1}) - \psi_d(x^k, z^k)| \le K_2 ||(\delta^k, q^k)||^2$$

for all iterates $(x^k, z^k) \in \mathcal{C}_\eta$ for η sufficiently small. Thus it follows from the above inequalities and Lemma 5.1 that there is a constant K_3 such that for j sufficiently large, if

$$||(x^{k_j}, z^{k_j}) - (\tilde{x}, \tilde{z})|| \le \epsilon_j,$$

then

$$|\psi_d(x^{k_j+1}, z^{k_j+1}) - \psi_d(\tilde{x}, \tilde{z})| \le K_3 \epsilon_j$$

for all $(\tilde{x}, \tilde{z}) \in L$. Since the iterations between k_j and k_{j+1} must result in a decrease in ψ_d, it must be that ψ_d has the same value, say $\tilde{\psi}_d$, at all points of L and at all other limit points of the sequence of iterates. Now suppose that (x^l, z^l) is any limit point of the sequence not contained in L. Then for k large and (x^k, z^k) close to (x^l, z^l), (7.14) and (7.15) imply that

$$\psi_d(x^{k+1}, z^{k+1}) - \psi_d(x^k, z^k) \le \tfrac{1}{2}\alpha \, \sigma \, \nabla\psi_d(x^k, z^k)^{\mathrm{t}}(\delta^k, q^k) \le -\tfrac{1}{2}\alpha\sigma \, \hat{\kappa}||(\delta^k, q^k)||^2.$$

But from the preceding, it follows that the left side of these inequalities is tending to zero, while from Proposition 5.2 the right side is bounded away from zero. This contradiction implies that there are no limit points of the sequence outside of L. But now, since the steps near the points in L tend to zero and the points in L are a positive distance apart, it follows that if there is more than one point in L then there must be a subsequence of iterates that is bounded away from the set L. By A1 this subsequence has a limit point that is not in L. Therefore, we can conclude that the sequence of iterates has exactly one limit point, which is in $\mathcal{S}$. □

8. Approximate solution of the quadratic subproblem. The algorithm in section 6 assumes that the quadratic subproblem can be solved exactly to generate the step δ^k. In practice this may not be realistic for two reasons. First, the quadratic subproblem may be infeasible—a far from uncommon occurrence in large-scale problems, especially if there is a very large number of nonlinear constraints. Second, even if the quadratic subproblem is feasible, the cost of obtaining an accurate solution may be prohibitive; moreover, at the beginning of the algorithm, where the quadratic problem is not necessarily a faithful representation of the nonlinear program, an accurate solution may not lead to a more useful step than an approximate solution. In this section we address these issues, suggesting how approximate solutions to a modified quadratic program can be used to generate useful steps. While the theoretical development is not as complete as we would like, our numerical experience using these approaches has been quite successful (see [2] and [13]).

We begin by formulating a modified quadratic program that can be solved even when (1.2) is infeasible. This problem is just the quadratic version of the phase 1 or "big M" problem used in linear programming:

$$\tag{8.1} \begin{aligned} &\min_{(s,\theta)} \nabla f(x^k)^{\mathrm{t}} s + \tfrac{1}{2} s^{\mathrm{t}} B^k s + M\,\theta \\ &\text{subject to } \nabla g(x^k)^{\mathrm{t}} s + g(x^k) - e\,\theta \le 0, \\ &\qquad\qquad\qquad\qquad\qquad\qquad \theta \ge 0, \end{aligned}$$

where e is the vector of ones. For θ large enough this problem is always consistent; in particular, (s^0, θ^0) with $s^0 = 0$ and

$$\theta^0 = \max\{g_1(x^k), \ldots, g_m(x^k), 0\}$$

is a feasible point. Note that if x^k is feasible for (1.1), then $\theta^0 = 0$. The constant M is a positive scalar sufficiently large so that if (1.2) has an optimal solution δ^*, then $(s,\theta) = (\delta^*, 0)$ is the optimal solution of (8.1). Given the initial feasible point (s^0, θ^0) as above we can generate a sequence of approximate solutions (s^j, θ^j) to (8.1) by

$$\tag{8.2} s^{j+1} = s^j + \rho_j\, P_j\, \xi^j,$$

$$\tag{8.3} \theta^{j+1} = \theta^j + \rho_j \gamma^j,$$

where P_j is an $n \times p$ matrix of rank p (P_j depends on k), $\rho_j \in (0,1]$, and (ξ^j, γ^j) is the solution of

$$\tag{8.4} \begin{aligned} &\min_{(\xi,\gamma)} \nabla f(x^k)^{\mathrm{t}}(s^j + P_j\xi) + \tfrac{1}{2}(s^j + P_j\xi)^{\mathrm{t}} B^k (s^j + P_j\xi) + M\,(\theta^j + \gamma) \\ &\text{subject to } \nabla g(x^k)^{\mathrm{t}}(s^j + P_j\xi) + g(x^k) - e\,(\theta^j + \gamma) \le 0, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \theta^j + \gamma \ge 0. \end{aligned}$$

Observe that (8.4) is (8.1) restricted to the affine space $\{s^j + P_j\xi :\ \xi \in \mathcal{R}^p\}$.

This method of solving (8.1) allows a variety of implementations. For example, if the matrices P_j are suitably chosen and $\rho_j = 1$, then it becomes an active set method. In our implementation we use the O3D interior point algorithm (see [1]), where the number of columns of P_j is three, so that the problem (8.4) is rather easily solved. Note that if x^k is feasible, then $\theta^0 = 0$ and the dependence on γ in (8.4) is removed or, in general, if (1.2) has any feasible solution, the value of θ^j will be zero for j sufficiently large.

We will use the approximate solution (s^J, θ^J) for a given $J \ge 1$ to generate a step at iteration k of our algorithm by means of the formulas

$$\tag{8.5} \delta^k = s^J,$$

$$\tag{8.6} \bar{q}^k = -(\nabla g(x^k)^{\mathrm{t}} \delta^k + g(x^k) + z^k - e\,\theta^J).$$

Note that this definition of $\bar{q}^k$ differs from that of (3.1). The purpose of the added term involving θ^J is to ensure that z^{k+1} remains nonnegative. From the constraints in (8.4),

$$z^{k+1} = z^k + \alpha\,\bar{q}^k \ge 0$$

for $\alpha \in [0,1]$. In order for this step to be useful it must have the same type of properties that are proved for the step determined by the exact solution of (1.2) in section 5. The following results provide the basic properties of this step with respect to the change in feasibility and the decrease in the merit function.

For these results we need the first-order necessary and complementary slackness conditions to (8.4) for a solution (ξ^j, γ^j). They are

$$(P_j)^{\mathrm{t}}\left[B^k P_j \xi^j + B^k s^j + \nabla f(x^k) + \nabla g(x^k)\,\mu^j\right] = 0, \tag{8.7}$$

$$M - e^{\mathrm{t}}\mu^j - \nu^j = 0, \tag{8.8}$$

$$(\mu^j)^{\mathrm{t}}\left[\nabla g(x^k)^{\mathrm{t}} P_j \xi^j + g(x^k) + \nabla g(x^k)^{\mathrm{t}} s^j - e(\theta^j + \gamma^j)\right] = 0, \tag{8.9}$$

$$\nu^j\ \left[\theta^j + \gamma^j\right] = 0 \tag{8.10}$$

for nonnegative multipliers $\mu^j \in \mathcal{R}^m$ and $\nu^j \in \mathcal{R}^1$.

PROPOSITION 8.1. *Let $(\delta^k, \bar{q}^k)$ be defined by (8.5) and (8.6) for a given J. Then*

$$\nabla r(x^k, z^k)^{\mathrm{t}}(\delta^k, \bar{q}^k) = -2r(x^k, z^k) + \bar{c}(x^k, z^k)^{\mathrm{t}} e\,\theta^J.$$

Proof. This differs from the proof of Proposition 5.6 only in the presence of the term involving θ^J in $\bar{q}^k$. □

This shows that for θ^J sufficiently small, $(\delta^k, \bar{q}^k)$ is a descent direction for r. We next prove a similar result for the merit functions. We begin with a lemma that gives the heart of the induction proof.

LEMMA 8.2. *Let (s^j, θ^j) be defined by (8.2) and (8.3). Then for any $j \geq 0$,*

$$\nabla f(x^k)^{\mathrm{t}} s^j \leq -\tfrac{1}{2}(s^j)^{\mathrm{t}}\, B_k\, s^j + M\,(\theta^0 - \theta^j).$$

Proof. The proof is by induction. It certainly holds for $j = 0$. Assume that it is true for $j \geq 0$. Thus

$$\triangle \equiv \nabla f(x^k)^{\mathrm{t}} s^{j+1} = \nabla f(x^k)^{\mathrm{t}} s^j + \rho_j\, \nabla f(x^k)^{\mathrm{t}} P_j \xi^j.$$

Since $\rho_j \leq 1$ it follows from the induction assumption, the positive definiteness of B^k, and (8.7) that

$$\begin{aligned}\triangle \leq\ & -\tfrac{1}{2}(s^j)^{\mathrm{t}}\, B^k\, s^j + M\,(\theta^0 - \theta^j)\\ & -\tfrac{1}{2}\rho_j^2(\xi^j)^{\mathrm{t}}(P_j)^{\mathrm{t}} B^k P_j \xi^j - \rho_j (s^j)^{\mathrm{t}} B^k P_j \xi^j\\ & -\rho_j(\mu^j)^{\mathrm{t}} \nabla g(x^k)^{\mathrm{t}} P_j \xi^j.\end{aligned}$$

Then, from (8.9), the first constraint in (8.4), and the nonnegativity of μ^j,

$$\begin{aligned}\triangle &\leq -\tfrac{1}{2}(s^{j+1})^{\mathrm{t}} B^k s^{j+1} + M\,(\theta^0 - \theta^j)\\ &\quad +\rho_j(\mu^j)^{\mathrm{t}}\left[\nabla g(x^k)^{\mathrm{t}} s^j + g(x^k) - e(\theta^j + \gamma^j)\right]\\ &\leq -\tfrac{1}{2}(s^{j+1})^{\mathrm{t}} B^k s^{j+1} + M\,(\theta^0 - \theta^j) - \rho_j(\mu^j)^{\mathrm{t}} e\,\gamma^j.\end{aligned}$$

From (8.8) and (8.3)

$$\begin{aligned}\triangle &\leq -\tfrac{1}{2}(s^{j+1})^{\mathrm{t}} B^k s^{j+1} + M\,(\theta^0 - \theta^j) - \rho_j\,(M - \nu^j)\gamma^j\\ &= -\tfrac{1}{2}(s^{j+1})^{\mathrm{t}} B^k s^{j+1} + M\,(\theta^0 - \theta^{j+1}) + \rho_j \nu_j \gamma_j.\end{aligned}$$

If $\theta^{j+1} = \gamma^j + \theta^j > 0$, i.e., if s^{j+1} is not a feasible point for (1.2), then from (8.10) we have that $\nu^j = 0$ and the last term is equal to 0. Otherwise, $\gamma^j = -\theta^j \leq 0$, and since $\nu^j \geq 0$, the last term is nonpositive. In either case, the result follows. □

Proposition 8.3. *There exist positive constants $\sigma_j, j = 1, \ldots, 5$, such that for $(x^k, z^k) \in \mathcal{G}$ and $(\delta^k, \bar{q}^k)$ given by* (8.5) *and* (8.6) *for some integer J we have*

$$\begin{aligned}\nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, \bar{q}^k) \leq -\sigma_1 \left\|\delta^k\right\|^2 + \sigma_2 \left\|\nabla g(x^k)^{\mathsf{t}}\delta^k + \bar{q}^k\right\| \\ -\frac{1}{d}\left\|\nabla g(x^k)^{\mathsf{t}}\delta^k + \bar{q}^k\right\| \left[\sigma_3 \left\|\nabla g(x^k)^{\mathsf{t}}\delta^k + \bar{q}^k\right\| - \sigma_4\theta^J\right] + \sigma_5\,\theta^J.\end{aligned}$$

Proof. From (3.3) and (3.4) we have

$$\begin{aligned}\triangle &\equiv \nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, \bar{q}^k) \\ &= \nabla f(x^k)^{\mathsf{t}}\delta^k + (\bar{\lambda}^k)^{\mathsf{t}}(\nabla g(x^k)^{\mathsf{t}}\delta^k + \bar{q}^k) \\ &\quad + \frac{2}{d}\bar{c}(x^k, z^k)^{\mathsf{t}}\bar{A}_k^{-1}(\nabla g(x^k)^{\mathsf{t}}\delta^k + \bar{q}^k).\end{aligned}$$

Using the previous lemma and the definition of $\bar{q}^k$ we obtain

$$\begin{aligned}\triangle \leq -\frac{1}{2}(\delta^k)^{\mathsf{t}}B^k\delta^k + M(\theta^0 - \theta^J) + (\bar{\lambda}^k)^{\mathsf{t}}(\nabla g(x^k)^{\mathsf{t}}\delta^k + \bar{q}^k) \\ -\frac{2}{d}(\nabla g(x^k)^{\mathsf{t}}\delta^k + \bar{q}^k - e\theta^J)^{\mathsf{t}}\bar{A}_k^{-1}(\nabla g(x^k)^{\mathsf{t}}\delta^k + \bar{q}^k).\end{aligned}$$

Noting that for some positive constant χ,

$$\theta^0 \leq \chi \left\|g(x^k) + z^k\right\| \leq \chi \left\|\nabla g(x^k)^{\mathsf{t}}\delta^k + \bar{q}^k - e\theta^J\right\|,$$

and using the positive definiteness of B^k and $\bar{A}_k$, we obtain the desired result. □

We now consider the possibility of solving the quadratic subproblem by using the iterative technique suggested above and discuss the implications for global convergence. There are several possible scenarios. In the ideal situation, for each k the subproblems are consistent and are solved completely. Then the theory of the preceding three sections holds and the algorithm is globally convergent. The more realistic cases allow for early termination of the algorithm and the possibility of inconsistency of the quadratic programs.

If we assume that the subproblems are all consistent then, at each major iteration, the iterations on the subproblem can be continued until feasibility is reached, i.e., until the value of θ is zero. Thereafter, the iterations can be terminated at any iteration J and the resulting step $(\delta^k, \bar{q}^k)$ computed via (8.5) and (8.6). Since $\theta^J = 0$, $\bar{q}^k$ becomes the standard step q^k and Proposition 8.1 shows that the resulting step will be a descent step for the measure of infeasibility $r(x, z)$. Moreover, Proposition 8.3 shows that

$$\begin{aligned}\nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, \bar{q}^k) \leq -\sigma_1 \left\|\delta^k\right\|^2 + \sigma_2 \left\|\nabla g(x^k)^{\mathsf{t}}\delta^k + \bar{q}^k\right\| \\ -\frac{\sigma_3}{d}\left\|\nabla g(x^k)^{\mathsf{t}}\delta^k + \bar{q}^k\right\|^2.\end{aligned} \tag{8.11}$$

We would like to use this result to obtain a proposition of the form of Proposition 5.7 for this truncated step. However, because of the second term on the right-hand side, the above inequality does not have the proper form to apply Lemma 5.4. A weaker form of Proposition 5.7 can be proven, i.e., one in which the result holds outside of a ball of radius ϵ around the solution.

Proposition 8.4. *Let $\epsilon > 0$ be given. Let $(x^k, z^k) \in \mathcal{G} - \mathcal{S}_\epsilon$ and suppose that $(\delta^k, \bar{q}^k)$ is given by* (8.5)–(8.6) *for some J with $\theta^J = 0$. Then there exist a $d(\epsilon) > 0$ and a positive constant κ such that*

$$\nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, \bar{q}^k) \leq -\kappa \left\|(\delta^k, \bar{q}^k)\right\|^2$$

for all $d \leq d(\epsilon)$.

Proof. Since $\theta^J = 0$ we have $\bar{q}^k = q^k$. From the compactness of $\mathcal{C}_0 - \mathcal{S}_\epsilon$ and the fact that $\delta^k = 0$ and $(x^k, z^k) \in \mathcal{C}_0$ imply that $(x^k, z^k) \in \mathcal{S}$, we have that

$$\omega_\epsilon = \min\left\{\left\|\delta^k\right\| : (x^k, z^k) \in \mathcal{C}_0 - \mathcal{S}_\epsilon,\ B^k \in \mathcal{B}\right\}$$

is positive. Let $\eta_\epsilon > 0$ be such that $(x^k, z^k) \in \mathcal{C}_{\eta_\epsilon} - \mathcal{S}_\epsilon$ implies

$$\left\|\delta^k\right\| \geq \frac{\omega_\epsilon}{2} \tag{8.12}$$

and let

$$\gamma_\epsilon = \min\left\{\eta_\epsilon, \frac{\sigma_1 (\omega_\epsilon)^2}{8\,\sigma_2}\right\}. \tag{8.13}$$

Then for $(x^k, z^k) \in \mathcal{C}_{\gamma_\epsilon} - \mathcal{S}_\epsilon$, (8.12) holds and

$$\sigma_2 \left\|\nabla g(x^k)^{\mathrm{t}}\delta^k + q^k\right\| \leq \frac{\sigma_1}{2}(\omega_\epsilon/2)^2 \leq \frac{\sigma_1}{2}\left\|\delta^k\right\|^2.$$

Thus, for these (x^k, z^k), from (8.11)

$$\nabla\psi_d^k(x^k, z^k)^{\mathrm{t}}(\delta^k, \bar{q}^k) \leq -\frac{\sigma_1}{2}\left\|\delta^k\right\|^2 - \frac{\sigma_3}{d}\left\|\nabla g(x^k)^{\mathrm{t}}\delta^k + q^k\right\|^2. \tag{8.14}$$

On the other hand, if $(x^k, z^k) \notin \mathcal{C}_{\gamma_\epsilon} \cup \mathcal{S}_\epsilon$, then for

$$d < \frac{\sigma_3\gamma_\epsilon}{2\,\sigma_2}$$

we have, using (8.13),

$$\begin{aligned}\nabla\psi_d^k(x^k, z^k)^{\mathrm{t}}(\delta^k, \bar{q}^k) &\leq -\frac{\sigma_1}{2}\left\|\delta^k\right\|^2 + \frac{1}{d}\left[\frac{d\,\sigma_2}{\gamma_\epsilon} - \sigma_3\right]\left\|\nabla g(x^k)^{\mathrm{t}}\delta^k + q^k\right\|^2 \\ &\leq -\frac{\sigma_1}{2}\left\|\delta^k\right\|^2 - \frac{\sigma_3}{2\,d}\left\|\nabla g(x^k)^{\mathrm{t}}\delta^k + q^k\right\|^2.\end{aligned}$$

The result follows from Lemma 5.4 using this inequality together with (8.14). □

This result shows that the desired inequality is obtained for small d, except in a neighborhood of the first-order solutions. That is, a uniform d cannot be used for the entire algorithm. In practice this is not an added burden since the appropriate value of d for the theoretical convergence of the preceding sections cannot be identified a priori; in practice a heuristic must be used occasionally to adjust the value of the parameter (see [2]). In any case, we see that the step obtained by an early termination of this type of iterative algorithm for solving the quadratic subproblem is a satisfactory one in many respects and can be incorporated into a globally convergent algorithm, for example, by solving the quadratic program to greater accuracy as a solution is approached. Since rapid local convergence almost always requires such a strategy, this is, in fact, good computational practice.

The situation in which many of the subproblems are inconsistent is more difficult to handle. If the inconsistencies are mild, e.g., if the iterate is close to feasibility, then θ^J can be made small for sufficiently large J and Propositions 8.1 and 8.3 can be used in much the same way as the previous case. If the optimal value of θ is large, the generated step may not be a descent step for either $r(x, z)$ or for the approximate merit function. In this case, the heuristic procedure used in our algorithm is to take the step (δ^k, q^k) with q^k defined as in (3.1), i.e., not depending on θ. This step is a descent direction for $r(x, z)$; however, it is possible that the new z, $z^{k+1} = z^k + \alpha\, q^k$ will become negative. Our strategy here is to reset the negative values of z^{k+1} to a small positive value. Although there is little theoretical justification for this approach, it seems to work well in practice. (See [13].)

9. Discussion.

We have demonstrated global convergence for a basic version of an SQP algorithm for solving large-scale problems, and we have extended our analysis to the case of approximately solving the subproblems. An actual implementation of this algorithm, however, involves further extensions and modifications that expand the range of applicability of the algorithm. This is especially true in the large-scale case where issues of efficiency are paramount. In this section we briefly consider a few of these modifications and their implications on the theory. For a more complete discussion of these issues, see [2] and [13].

9.1. q-superlinear convergence.

An important consideration in the use of the merit function approach to the implementation of an SQP algorithm is the issue of the final rate of convergence. As is shown in [5] the rate of convergence is dependent on the choice of the matrices B^k. A discussion of actual strategies for selecting B^k is beyond the scope of this paper, but we show that if q-superlinear convergence is possible, then the use of our merit functions will not interfere with the process; i.e., a steplength of $\alpha = 1$ will ultimately be acceptable to both merit functions. Recall that obtaining a fast rate of convergence generally requires a strong solution and that the quadratic subproblems can be solved exactly near that solution; thus we make these assumptions here.

First, since the quadratic program will identify the correct active constraints near the solution, the value of δ in that area will be given as the solution to

$$\begin{aligned} &\min_{\delta}\ \nabla f(x)^{\mathrm{t}}\delta + \tfrac{1}{2}\delta^{\mathrm{t}}B\delta \\ &\text{subject to}\ \ \nabla g_a(x)^{\mathrm{t}}\delta + g_a(x) = 0, \end{aligned} \tag{9.1}$$

where (g_a, g_u) denotes the partition into active and inactive constraints at x^*. If we denote by P_a the projection onto the space orthogonal to the gradients of the active constraints at x^*, then the characterization of the q-superlinear convergence of the sequence $\{x^k\}$ generated by the SQP algorithm is [7]

$$\lim_{k\to\infty} \frac{\left\|P_a(H_{xx}L(x^k, \mu^k) - B^k)\delta^k\right\|}{\|\delta^k\|} = 0, \tag{9.2}$$

where (δ^k, μ^k) is the optimal solution-multiplier pair for the quadratic program (9.1).

To prove the result we need a further, mild assumption, namely, "tangential convergence" of the iterates x^k. To explain, let $Q_a = I - P_a$. Then $\{x^k\}$ is said to converge tangentially if

$$\frac{\left\|Q_a\delta^k\right\|}{\|\delta^k\|} \to 0. \tag{9.3}$$

In [3] it is shown that tangential convergence implies q-superlinear convergence, and in [4] we argue that the converse nearly always holds, especially in the nonconvex case. Our computational experience and that of others support this conclusion.

PROPOSITION 9.1. *Let the hypotheses of Proposition* 5.5 *hold and assume that the sequence* $\{x^k\}$ *generated by the algorithm converges to* x^* *tangentially and q-superlinearly. Let* $\sigma \in (0, \frac{1}{2})$. *Then there exists a* $\bar{d} > 0$ *such that for each* $d \in (0, \bar{d})$ *there is a positive integer* $J(d)$ *satisfying*

$$\psi_d^k(x^k + \delta^k, z^k + q^k) - \psi_d^k(x^k, z^k) \leq \sigma \nabla \psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k)$$

for $k \geq J(d)$.

Proof. Let

$$\triangle \equiv \left[\psi_d^k(x^k + \delta^k, z^k + q^k) - \psi_d^k(x^k, z^k) - \sigma \nabla \psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k)\right].$$

Using Taylor series,

$$\begin{aligned}\triangle = (1-\sigma)\nabla\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k) + \tfrac{1}{2}(\delta^k, q^k)^{\mathsf{t}} H\psi_d^k(x^k, z^k)(\delta^k, q^k)\\ +O\left(\left\|(\delta^k, q^k)\right\|^3\right).\end{aligned}$$

For positive constants ξ_1 and ξ_2 we have, using essentially the same steps given in the proof of Proposition 5.5,

$$\begin{aligned}\psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k) \leq -(\delta^k)^{\mathsf{t}} B^k \delta^k + \xi_1 \left\|(\delta^k, q^k)\right\| \left\|\nabla g(x^k)^{\mathsf{t}}\delta^k + q^k\right\|\\ -\frac{2}{d}\left[\nabla g(x^k)^{\mathsf{t}}\delta^k + q^k\right]\left(\bar{A}_k\right)^{-1}\left[\nabla g(x^k)^{\mathsf{t}}\delta^k + q^k\right]\end{aligned}$$

and, upon expanding $H\psi_d^k(x^k, z^k)$ by differentiating (3.3) and (3.4),

$$\begin{aligned}(\delta^k, q^k)^{\mathsf{t}} H\psi_d^k(x^k, z^k)(\delta^k, q^k) \leq (\delta^k)^{\mathsf{t}} H_{xx}L(x^k, z^k)\delta^k\\ + \left[\nabla g(x^k)^{\mathsf{t}}\delta^k + q^k\right]^{\mathsf{t}}\left(\bar{A}_k\right)^{-1}\left[\nabla g(x^k)^{\mathsf{t}}\delta^k + q^k\right]\\ +\frac{\xi_2}{2}\left\|(\delta^k, q^k)\right\|^2 \left\|\nabla g(x^k)^{\mathsf{t}}\delta^k + q^k\right\|.\end{aligned}$$

The positive definiteness of B^k and the fact that $P_a + Q_a$ is the identity matrix yield

$$-\delta^{k\,\mathsf{t}} B^k \delta^k \leq -\left(\frac{1}{2} - \sigma\right)\rho_1 \left\|(\delta^k, q^k)\right\|^2 + \frac{(P_a + Q_a)}{2}(\delta^k)^{\mathsf{t}} B^k \delta^k$$

and

$$H_{xx}L(x^k, z^k) = (P_a + Q_a)\, H_{xx}L(x^k, z^k).$$

Thus

$$\begin{aligned}\triangle \leq -\left(\frac{1}{2} - \sigma\right)\rho_1 \left\|\delta^k\right\|^2 + (1-\sigma)\xi_1 \left\|(\delta^k, q^k)\right\| \left\|\nabla g(x^k)^{\mathsf{t}}\delta^k + q^k\right\|\\ -\frac{(1-2\sigma)}{d}\left[\nabla g(x^k)^{\mathsf{t}}\delta^k + q^k\right]^{\mathsf{t}}\left(\bar{A}_k\right)^{-1}\left[\nabla g(x^k)^{\mathsf{t}}\delta^k + q^k\right]\\ +O\left(\left\|(\delta^k, q^k)\right\|^3\right) + \frac{1}{2}\left\|\delta^k\right\| \left\|P_a(H_{xx}L(x^k, z^k) - B^k)\delta^k\right\|\\ +\frac{1}{2}\left\|Q_a\delta^k\right\| \left\|(H_{xx}L(x^k, z^k) - B^k)\delta^k\right\|.\end{aligned}$$

Now using (9.2) and (9.3) and the type of argument used in the proof of Proposition 5.6 we see that $\triangle \leq 0$ for d sufficiently small and k large. Since

$$\psi_d^k(x^k + \delta^k, z^k + q^k) - \psi_d^k(x^k, z^k) = \triangle + \sigma\, \psi_d^k(x^k, z^k)^{\mathsf{t}}(\delta^k, q^k),$$

the result follows. □

A similar result holds for $\psi_d(x, z)$.

9.2. The dynamic adjustment of *d*. We have assumed in the above theory that the parameter d has been chosen in advance to be small enough so that the results of section 5 hold. Of course, this is unreasonable in practice, and thus we have developed heuristic procedures to adjust d dynamically. It follows from Proposition 5.5 that if (x^k, z^k) is not feasible, then the penalty parameter d can be made small enough to ensure that the step is a direction of descent for ψ_d^k. In our implementation, if, at some nonfeasible point, a good decrease in ψ_d^k is not achieved, we decrease d by an appropriate factor. This can be done without fundamentally altering the convergence theory. However, using too small of a value of d often causes the algorithm to slow dramatically, since it tends to force the iterates to stay close to the constraints. To avoid this, it is desirable to incorporate a means for allowing an increase in d when the steps are good descent steps for ψ_d^k. Since increasing d can theoretically cause the iterates to cycle, care has to be taken in implementing such a procedure. In [2] and [13] we have used a heuristic strategy for increasing d that avoids cycling and that has been effective in our numerical experiments. However, a theoretical proof of global convergence when this process is implemented is lacking.

9.3. A trust region approach. In the implemented version of our algorithm the O3D solver is employed in the following way. At iteration k in the main algorithm, the quadratic program (1.2) is formed and the iteration procedure described above is begun. The algorithm is stopped when a prescribed tolerance for an optimality condition is satisfied *or* when a *trust region constraint* of the form

$$\left\| s^j \right\| \leq \tau_k$$

is violated. If the final iterate is optimal (or very nearly optimal) for (1.2), then, of course, the theory described in the preceding sections applies. The significance of this trust region approach is in the case where the last iterate is not optimal. The discussion in section 8 motivates the use of this step despite its nonoptimality. The trust region constraint can be implemented in such a way that it will become inactive near the solution, and the optimal solution of the quadratic program will thus be computed. The trust region parameter, τ_k, is adjusted in a manner similar in spirit to that used in most trust region methods; that is, the decision to increase or decrease τ_k is based on a comparison of the predicted and actual reductions of the merit function. In our implementation we use either ψ_d or ψ_d^k depending on the current status of the point $(x^k$,$z^k)$. For details of this procedure, as well as results for numerical experiments, see [2].

9.4. Final remarks. The two tests in step 5 of the algorithm require the evaluation of $\nabla\psi_d(x, z)$, which is quite expensive. Propositions 5.6 and 5.8 assure us that close enough to feasibility, the tests will be automatically satisfied. In our computational experience we have never encountered a situation where the second test was necessary, and we therefore do not perform it at all. Moreover, we require only that ψ_d be reduced, not that the Goldstein–Armijo condition be satisfied, thus avoiding all calculations of $\nabla\psi_d(x, z)$.

As noted in section 4 we need to adopt a procedure that will keep $\bar{A}(x, z)$ nonsingular. We observed that there was an inexpensive procedure to keep $z_k > 0$ for all k. This can be done by choosing $z_0 > 0$ and modifying the update so that

$$z^{k+1} = z^k + \alpha\gamma q^k,$$

where $\gamma < 1$. Such a modification will not be necessary in the neighborhood of a strong solution and, far from the solution, this modification does not affect the theory. In practice we simply proceed until $\bar{A}$ becomes ill conditioned and then increase z as appropriate.

The algorithm in this paper does not depend explicitly on the approximations λ^k to the optimal multiplier λ^*. However, since B^k is an approximation to the Hessian of the Lagrangian, λ^k enters into the calculation of B^k and, because B^k determines the final rate of convergence, it is of interest to construct a sequence $\{\lambda^k\}$ that is a good approximation to λ^*. If we are solving the subproblems exactly, we generally use

$$\lambda^{k+1} = \lambda^k + \alpha(\mu^k - \lambda^k),$$

where μ^k is the multiplier for (1.2). If the μ^k are chosen by a consistent method, e.g., as the minimum norm multiplier, then it follows that if $x^k \to x^*$ then $\lambda^k \to \lambda^*$. However, when the quadratic subproblems are solved only approximately, these multiplier estimates are usually poor and we use the least squares multipliers, $\bar{\lambda}(x, z)$, instead.

Other ways of handling infeasible quadratic subproblems have been developed and tested by one of the authors [13]. The procedures there perturb the linearized constraints to ensure feasibility of the subproblems. This has the effect of better balancing the necessity for feasibility with the need to become optimal. Excellent results have been obtained on problems with a large number of nonlinear constraints.

REFERENCES

[1] P. T. Boggs, P. D. Domich, and J. E. Rogers, *An interior-point method for general large scale quadratic programming problems*, Ann. Oper. Res., 62 (1996), pp. 419–437.

[2] P. T. Boggs, A. J. Kearsley, and J. W. Tolle, *A practical algorithm for general large scale nonlinear optimization problems*, SIAM J. Optim., 9 (1999), pp 755–778.

[3] P. T. Boggs and J. W. Tolle, *A strategy for global convergence in a sequential quadratic programming algorithm*, SIAM J. Numer. Anal., 26 (1989), pp. 600–623.

[4] P. T. Boggs and J. W. Tolle, *Convergence properties of a class of rank-two updates*, SIAM J. Optim., 4 (1994), pp. 262–287.

[5] P. T. Boggs and J. W. Tolle, *Sequential quadratic programming*, in Acta Numerica, 1995, Cambridge University Press, Cambridge, 1995, pp. 1–51.

[6] P. T. Boggs, J. W. Tolle, and A. J. Kearsley, *On the convergence of a trust region SQP algorithm for nonlinearly constrained optimization problems*, in Proceedings of the 17th IFIP TC7 Conference on System Modeling and Optimization, J. Dolezal, ed., Chapman and Hall, London, 1995, pp. 1–14.

[7] P. T. Boggs, J. W. Tolle, and P. Wang, *On the local convergence of quasi-Newton methods for constrained optimization*, SIAM J. Control Optim., 20 (1982), pp. 161–171.

[8] R. H. Byrd, R. A. Tapia, and Y. Zhang, *An SQP augmented Lagrangian BFGS algorithm for constrained optimization*, SIAM J. Optim., 2 (1992), pp. 210–241.

[9] A. R. Conn, N. I. M. Gould, and P. T. Toint, *Lancelot: A Fortran Package for Large-Scale Nonlinear Optimization*, Springer Ser. Comput. Math. 17, Springer-Verlag, Heidelberg, New York, 1992.

[10] A. El-Bakry, R. A. Tapia, T. Tsuchyia, and Y. Zhang, *On the formulation and theory of the primal-dual Newton interior-point method for nonlinear programming*, J. Optim. Theory Appl., 89 (1996), pp. 507–541.

[11] R. Franke and E. Arnold, *Applying new numerical algorithms to the solution of discrete-time optimal control problems,* in Computer-Intensive Methods in Control and Signal Processing: The Curse of Dimensionality, K. Warwick and M. Kárný, eds., Birkhäuser Verlag, Basel, 1997, pp. 105–118.

[12] P. Gill, W. Murray, and M. Saunders, *SNOPT: An SQP Algorithm for Large Scale Constrained Optimization*, Preprint NA97-2, University of California, San Diego, 1997.

[13] A. J. Kearsley, *The Use of Optimization Techniques in the Solution of Partial Differential Equations from Science and Engineering*, Ph.D. thesis, Rice University, Houston, TX, 1996.

[14] W. Murray and F. J. Prieto, *A sequential quadratic programming algorithm using an incomplete solution of the subproblem*, SIAM J. Optim., 5 (1995), pp. 590–640.

[15] B. A. Murtagh and M. A. Saunders, *MINOS* 5.4 *User's Guide*, SOL, Department of Operations Research SOL 83-20R, Stanford University, Stanford, CA, revised 1995.

[16] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGraw-Hill, New York, 1995.

[17] J. Nocedal, *Theory of algorithms for unconstrained optimization*, in Acta Numerica, 1992, Cambridge University Press, Cambridge, 1992, pp. 199–242.

SIAM J. OPTIM.
Vol. 9, No. 4, pp. 863–876

NONSYMMETRIC SEARCH DIRECTIONS FOR SEMIDEFINITE PROGRAMMING[*]

NATHAN BRIXIUS[†], FLORIAN A. POTRA[‡], AND RONGQIN SHENG[§]

To John Dennis on the occasion of his 60th birthday.

Abstract. Two nonsymmetric search directions for semidefinite programming, the XZ and ZX search directions, are proposed. They are derived from a nonsymmetric formulation of the semidefinite programming problem. The XZ direction corresponds to the direct linearization of the central path equation $XZ = \nu I$, while the ZX direction corresponds to $ZX = \nu I$. The XZ and ZX directions are well defined if both X and Z are positive definite matrices, where X may be nonsymmetric. We present an algorithm using the XZ and ZX directions alternately following the Mehrotra predictor-corrector framework. Numerical results show that the XZ/ZX algorithm, in many cases, requires less CPU time than the XZ+ZX method of Alizadeh, Overton, and Haeberly [SIAM J. Optim., 8 (1998), pp. 746–768] while achieving similar accuracy.

Key words. semidefinite programming, nonsymmetric, search direction, interior-point algorithm, high accuracy

AMS subject classifications. 65K05, 90C25, 90C30

PII. S1052623498333883

1. Introduction.

The semidefinite programming (SDP) problem has the standard form

$$(P) \quad \min\{C \bullet X : A_i \bullet X = b_i,\ i = 1, \dots, m,\ X \in \mathcal{S}^n_+\}, \tag{1.1}$$

and its associated dual problem is

$$(D) \quad \max\left\{b^T y : \sum_{i=1}^m y_i A_i + Z = C,\ (y, Z) \in \mathsf{R}^m \times \mathcal{S}^n_+\right\}, \tag{1.2}$$

where $C \in \mathcal{S}^n, A_i \in \mathcal{S}^n, i = 1, \dots, m, b = (b_1, \dots, b_m)^T \in \mathsf{R}^m$ are given data. Here $\mathcal{S}^n$ denotes the set of all $n \times n$ symmetric matrices and $\mathcal{S}^n_+$ the set of all $n \times n$ symmetric positive semidefinite matrices. $G \bullet H$ is the trace of $G^T H$. For simplicity we assume that $A_i,\ i = 1, \dots, m$, are linearly independent.

Under the assumption that both (1.1) and (1.2) have finite solutions and their optimal values are equal, X^* and (y^*, Z^*) are solutions of (1.1) and (1.2) if and only if they are solutions of the following nonlinear system:

$$A_i \bullet X = b_i,\ i = 1, \dots, m, \tag{1.3a}$$

[*] Received by the editors February 9, 1998; accepted for publication (in revised form) February 15, 1999; published electronically September 24, 1999. This work was supported in part by the Department of Energy under contract W-31-109-Eng-38 and by the National Science Foundation under grant DMS-9706894. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/siopt/9-4/33388.html

[†] Department of Computer Science, University of Iowa, Iowa City, IA 52242 (nathan-brixius@uiowa.edu).

[‡] Department of Mathematics and Statistics, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250 (potra@math.umbc.edu).

[§] Lucent Technologies, 2443 Warrenville Road, Lisle, IL 60532 (rsheng@lucent.com).

$$\sum_{i=1}^{m} y_i A_i + Z = C, \tag{1.3b}$$

$$XZ = 0, \;\; X, \; Z \in \mathcal{S}_+^n. \tag{1.3c}$$

Most primal-dual interior-point methods for semidefinite programming can be interpreted as Newton-type algorithms for solving the nonlinear system (1.3). The search directions used by those interior-point algorithms are associated with different ways of linearizing the central path equation

$$XZ = \nu I, \tag{1.4}$$

where $\nu \geq 0$ is the central path parameter.

To ensure that the iterates X^k and Z^k are symmetric matrices, symmetric reformulations of central path equation (1.4) have been developed. Alizadeh, Haeberly, and Overton [2] considered instead of (1.4) the symmetric equation

$$XZ + ZX = 2\nu I. \tag{1.5}$$

Zhang [12] proposed a generalized symmetrization of the form

$$\frac{1}{2}[P^{-1}XZP + (P^{-1}XZP)^T] = \nu I, \tag{1.6}$$

where P can be any nonsingular matrix. Recently, Monteiro and Tsuchiya [6] considered the symmetric central path equations

$$Z^{1/2}XZ^{1/2} = \nu I, \qquad X^{1/2}ZX^{1/2} = \nu I. \tag{1.7}$$

Linearization of the above symmetric central path equations leads to different search directions. The most commonly used directions are the XZ+ZX or Alizadeh–Haeberly–Overton (AHO) direction [2], the Helmberg–Kojima–Moteiro (HKM) direction [3, 4, 5], and the Nesterov–Todd (NT) direction [8], obtained from (1.6) by taking P equal to I, $Z^{1/2}$, and $[Z^{1/2}(Z^{1/2}XZ^{1/2})^{-1/2}Z^{1/2}]^{1/2}$, respectively. Among these directions, AHO has been observed to achieve the highest accuracy. We also mention that Monteiro and Zanjácomo [7] and Toh [10] recently reported other search directions that can attain high accuracy.

All the above-mentioned search directions involve the linearization of a specific symmetric central path equation. In this paper, we show that the nonsymmetric central path equation (1.4) can be directly used without any symmetrization and that the resulting nonsymmetric search direction can be used in interior-point algorithms. Our approach is based on the following nonsymmetric formulation of SDP whose solution set contains that of (1.3):

$$A_i \bullet X = b_i, \; i = 1, \ldots, m, \tag{1.8a}$$

$$\sum_{i=1}^{m} y_i A_i + Z = C, \tag{1.8b}$$

$$XZ = 0, \;\; 0 \preceq X \in \mathsf{R}^{n\times n}, \; Z \in \mathcal{S}_+^n. \tag{1.8c}$$

In (1.8) the notation $0 \preceq X \in \mathsf{R}^{n\times n}$ means that X is positive semidefinite but not necessarily symmetric. In section 2, we will prove that if (X^*, y^*, Z^*) is a solution of (1.8), then $(\mathbf{sym}(X^*), y^*, Z^*)$ is a solution of (1.3), where we define the operator **sym** by

$$\mathbf{sym}(G) = \frac{1}{2}(G + G^T) \quad \text{for any real square matrix } G.$$

The same result holds if (1.8c) is replaced by

$$ZX = 0, \;\; 0 \preceq X \in \mathsf{R}^{n\times n}, \; Z \in \mathcal{S}^n_+. \tag{1.9}$$

The *XZ search direction* $(\Delta X, \Delta y, \Delta Z)$ is defined as the solution of the following linear system:

$$A_i \bullet \Delta X = b_i - A_i \bullet X, \;\; i = 1, \dots, m, \tag{1.10a}$$

$$\sum_{i=1}^{m} \Delta y_i A_i + \Delta Z = C - \sum_{i=1}^{m} y_i A_i - Z, \tag{1.10b}$$

$$X \Delta Z + \Delta X Z = \sigma\mu I - XZ, \tag{1.10c}$$

where $\mu = X \bullet Z/n$ and $\sigma \in [0, 1]$ is a centering parameter. Thus, the XZ direction can be viewed as the result of the direct linearization of the central path equation $XZ = \nu I$.

Correspondingly the *ZX search direction* is the solution of the linear system (1.10) with (1.10c) replaced by

$$Z\Delta X + \Delta Z X = \sigma\mu I - ZX.$$

We will show that the XZ and ZX directions exist provided X and Z are positive definite. Extensive numerical experiments show that interior-point methods based on the XZ or ZX direction alone can obtain neither the high accuracy of the AHO method nor the efficiency of the HKM method. On the other hand, if these directions are used alternately, then both the accuracy and speed of convergence are improved. Such a method is called an XZ/ZX method. Our numerical experiments show that the XZ/ZX method integrated in the Mehrotra predictor-corrector framework is competitive with the corresponding AHO method. The two methods have similar accuracy. Although our method usually takes about three more iterations, the CPU time, as well as the number of floating-point operations, is less in many cases. This is because our algorithm avoids the Lyapunov equations that the AHO method has to solve at each iteration.

The following notation and terminology are used throughout the paper:
R^p: the p-dimensional Euclidean space;
R^p_+: the nonnegative orthant of R^p;
R^p_{++}: the positive orthant of R^p;
$\mathsf{R}^{p\times q}$: the set of all $p \times q$ matrices with real entries;
$\mathcal{S}^p$: the set of all $p \times p$ symmetric matrices;
$\mathcal{S}^p_+$: the set of all $p \times p$ symmetric positive semidefinite matrices;
$\mathcal{S}^p_{++}$: the set of all $p \times p$ symmetric positive matrices;
$M \succeq 0$: M is positive semidefinite;
$M \succ 0$: M is positive definite;
$\lambda_i(M), \; i = 1, \dots, n$: the eigenvalues of $M \in \mathcal{S}^n$;
$\lambda_{\max}(M), \; \lambda_{\min}(M)$: the largest, smallest, eigenvalue of $M \in \mathcal{S}^n$;
$G \bullet H \equiv \mathrm{Tr}(G^T H)$;

$\|\cdot\|$: Euclidean norm of a vector and the corresponding norm of a matrix, i.e.,
$\|y\| \equiv \sqrt{\sum_{i=1}^{p} y_i^2}, \quad \|M\| \equiv \max\{\|My\| : \|y\| = 1\}$;
$\|M\|_F \equiv \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{q}[M]_{ij}^2},\ M \in \mathsf{R}^{p\times q}$: Frobenius norm of a matrix;
$\mathbf{sym}(M) \equiv (M + M^T)/2,\ M \in \mathsf{R}^{p\times p}$.

2. On the nonsymmetric formulation of SDP.

2. On the nonsymmetric formulation of SDP. The following result is well known.

LEMMA 2.1. *Let $X, Z \in \mathcal{S}_+^n$. Then $X \bullet Z = 0$ if and only if $XZ = ZX = 0$.*

The next lemma shows that (1.8c) can be replaced by (1.9).

LEMMA 2.2. *Let $0 \preceq X \in \mathsf{R}^{n\times n}, Z \in \mathcal{S}_+^n$. Then $XZ = 0$ if and only if $ZX = 0$.*

Proof. ($\Rightarrow$). $XZ = 0$ implies $(X + X^T) \bullet Z = 2X^T \bullet Z = 0$. Then from Lemma 2.1, we obtain $(X + X^T)Z = 0$, which yields $X^T Z = -XZ = 0$ and hence $ZX = (X^T Z)^T = 0$.

($\Leftarrow$). This is similar. □

THEOREM 2.3.

(a) *Every solution of* (1.3) *is also a solution of* (1.8).

(b) *If (X^*, y^*, Z^*) is a solution of* (1.8), *then* $(\mathbf{sym}(X^*), y^*, Z^*)$ *is a solution of* (1.3).

Proof. Part (a) follows directly from the definition of (1.8). To prove (b), we need to show only that $(\mathbf{sym}(X^*), y^*, Z^*)$ satisfies (1.3a) and (1.3c). Since $A_i, i = 1, \ldots, m$, are symmetric, we have

$$A_i \bullet \mathbf{sym}(X^*) = A_i \bullet X^* = b_i,\ i = 1, \ldots, m.$$

From $X^*Z^* = 0$ and Lemma 2.2 we obtain $Z^*X^* = 0$. Therefore,

$$\mathbf{sym}(X^*)Z^* = \frac{1}{2}[X^*Z^* + (Z^*X^*)^T] = 0. \qquad \square$$

3. The XZ and ZX search directions. The linear system (1.10) for the XZ search direction can be written in the following matrix form:

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z\otimes I & 0 & I\otimes X \end{pmatrix}\begin{pmatrix} \mathbf{vec}(\Delta X) \\ \Delta y \\ \mathbf{vec}(\Delta Z)\end{pmatrix} = \begin{pmatrix} \mathbf{vec}(R_d) \\ r \\ \mathbf{vec}(R_c)\end{pmatrix}, \tag{3.1}$$

where

$$A^T = [\mathbf{vec}(A_1), \mathbf{vec}(A_2), ..., \mathbf{vec}(A_m)],$$

$$r_i = b_i - A_i \bullet X,\ i = 1, \ldots, m, \tag{3.2a}$$

$$R_d = C - \sum_{i=1}^{m} y_i A_i - Z, \tag{3.2b}$$

$$r^T = [r_1, r_2, ..., r_m], \tag{3.2c}$$

$$R_c = \sigma\mu I - XZ. \tag{3.2d}$$

Here $\otimes$ denotes the Kronecker product. For any $n \times n$ matrix M, $\mathbf{vec}(M)$ denotes the vector obtained by stacking the columns of M, that is,

$$\mathbf{vec}(M) = (m_{11}, m_{21}, \ldots, m_{1n}, \ldots, m_{nn})^T.$$

The linear system (1.10) can be solved by the following procedure:

- Compute Δy by solving the linear system

  $$M\Delta y = h, \tag{3.3}$$

  where

  $$M = A(Z^{-1} \otimes X)A^T$$

  and

  $$h = r + A[\mathbf{vec}(XR_dZ^{-1}) - \mathbf{vec}(R_cZ^{-1})].$$

- Compute $\Delta Z,\ \Delta X$ as follows:

  $$\begin{aligned}\Delta Z &= R_d - \sum_{i=1}^{m} \Delta y_i A_i,\\ \Delta X &= R_cZ^{-1} - X\Delta Z Z^{-1}.\end{aligned}$$

LEMMA 3.1. *If $X \in \mathsf{R}^{n\times n}$ and $Z \in \mathcal{S}^n$ are positive definite, then the linear system* (1.10) *has a unique solution* $(\Delta X, \Delta y, \Delta Z) \in \mathsf{R}^{n\times n} \times \mathsf{R}^m \times \mathcal{S}^n$.

Proof. If the solution $(\Delta X, \Delta y, \Delta Z)$ of (1.10) exists, then the symmetry of ΔZ is automatic from (1.10b). Therefore, it is sufficient to prove that the Schur matrix $A(Z^{-1} \otimes X)A^T$ is nonsingular. From the symmetry of Z^{-1}, we have

$$\begin{aligned}&A(Z^{-1} \otimes X)A^T + [A(Z^{-1} \otimes X)A^T]^T\\ &= A[Z^{-1} \otimes (X + X^T)]A^T\ .\end{aligned}$$

The right-hand side of the above equation is positive definite because A has full rank and both Z and $X + X^T$ are positive definite. Therefore, $A(Z^{-1} \otimes X)A^T$ is positive definite and hence nonsingular. □

Remark 3.2. In the Schur complement equation (3.3) the Schur matrix M and the right side h can be computed by

$$m_{i,j} = A_i \bullet (XA_jZ^{-1}) \tag{3.4}$$

and

$$h_i = r_i + A_i \bullet [(XR_dZ^{-1}) - R_cZ^{-1}].$$

The methods for computing M for the HKM and XZ directions are quite similar. However, for the XZ direction, since X is not necessarily symmetric, neither is M. The formulas given in [10] for computing M when the matrix X is symmetric adapt easily to the XZ direction.

Let us consider the complexity of the computation of the XZ direction. If the matrices A_i are not sparse, then the major computational effort consists of forming the Schur matrix M. If formula (3.4) is used, the XZ direction can be computed in $4mn^3 + 2m^2n^2 + O(\max\{m,n\}^3)$ flops, since $2m$ matrix multiplications and m^2 inner products are involved. Therefore, the complexity of computing the XZ direction by using formula (3.4) is

$$4mn^3 + 2m^2n^2 + O(\max\{m,n\}^3).$$

Table 1

Computational results for varying classes of SDP. Ten random instances of each class were tested. Note that the infeasibility of all problems is reduced to a level less than 10^{-12} *except for the last problem, where* 10^{-11} *is attained.*

	Average accuracy achieved by $\lvert\mathrm{mean}(\log_{10}(X \bullet Z))\rvert$				Average CPU time (min.) to attain the accuracy			
	AHO	XZ/ZX	HKM	NT	AHO	XZ/ZX	HKM	NT
Random $n = 100$ $m = 50$	8.79	8.98	7.10	7.14	4.82	4.24	3.23	3.25
Random $n = 50$ $m = 100$	9.55	9.18	7.74	7.65	3.81	4.18	2.55	2.32
Random $n = 100$ $m = 100$	8.63	8.60	7.35	7.05	13.50	13.75	9.43	8.68
Norm min. $n = 100$ $m = 30$	12.91	12.53	9.57	9.39	1.76	1.25	0.99	1.24
Cheby. Poly. $n = 100$ $m = 31$	13.84	13.02	10.65	9.03	2.51	1.81	1.46	1.55
Maxcut $n = 200$ $m = 200$	11.11	10.01	7.86	7.51	64.50	26.16	21.21	22.28
ETP $n = 110$ $m = 55$	9.35	7.40	8.17	8.18	4.59	3.84	3.09	3.19
Log. Cheby. $n = 300$ $n = 50$	10.12	5.71	9.28	9.31	18.29	11.44	9.70	9.64

Remark 3.3. The complexity of computing most commonly used search directions for SDP is of the form

$$\alpha mn^3 + \beta m^2 n^2 + O(\max\{m, n\}^3), \tag{3.5}$$

where α and β are two positive constants (see [7, 10]). We note that the third term in (3.5) cannot be neglected because sometimes it may contribute significantly to the complexity, especially when extra matrix factorizations are used. We also note that the computation of the XZ direction needs the least number of matrix factorizations. This feature is also shared by the HKM direction.

Remark 3.4. Theoretically, the complexity of computing the XZ direction can be reduced to

$$3mn^3 + 2m^2 n^2 + O(\max\{m, n\}^3),$$

by using the Cholesky factorization $Z = LL^T$ and the following formula, which is equivalent to (3.4):

$$m_{i,j} = (L^{-1} A_i) \bullet (L^{-1} A_j X^T). \tag{3.6}$$

Since L^{-1} is triangular, the computation of $L^{-1} A_i$, $i = 1, \ldots, m$, takes $mn^3 + O(\max\{m, n\}^3)$ flops. After $L^{-1} A_i$, $i = 1, \ldots, m$, is obtained, the computation of $L^{-1} A_j X^T$ involves m matrix multiplications and thus needs $2mn^3 + O(\max\{m, n\}^3)$ flops. Finally, m^2 inner products are needed, thus accounting for $2m^2 n^2 + O(\max\{m, n\}^3)$

Table 2

Average results for the number of iterations and the flops per iteration used to achieve the accuracy listed in Table 1.

	Average number of iterations to achieve the accuracy				Average Mflops per iteration			
	AHO	XZ/ZX	HKM	NT	AHO	XZ/ZX	HKM	NT
Random $n = 100$ $m = 50$	13.1	16.0	15.7	15.5	618	299	275	300
Random $n = 50$ $m = 100$	13.8	16.7	16.1	14.5	188	111	86	89
Random $n = 100$ $m = 100$	13.0	16.4	16.5	15.2	1253	652	553	578
Norm min. $n = 100$ $m = 30$	13.9	16.2	14.5	16.0	399	223	191	255
Cheby. Poly. $n = 100$ $m = 31$	13.9	16.1	15.1	14.7	1848	947	902	988
Maxcut $n = 200$ $m = 200$	15.6	16.2	15.4	15.4	12680	3419	3405	3616
ETP $n = 110$ $m = 55$	22.6	35.0	31.6	30.5	86	43.4	45.3	66.9
Log. Cheby. $n = 300$ $m = 50$	18.8	21.8	21.0	20.4	2.71	2.16	1.45	1.48

flops. Therefore, the computation of the XZ direction with formula (3.6) takes $3mn^3 + 2m^2n^2 + O(\max\{m,n\}^3)$ flops. However, in our MATLAB implementation, we use (3.4) instead of (3.6) because the CPU time often increases when (3.6) is applied. This is due to the fact that MATLAB is an interpreted language; code that tries to exploit the structure of L in computing $L^{-1}A_i,\ i = 1,\dots,m$, is slow in comparison with the built-in functions provided by MATLAB. A similar observation was made by Toh [10]. Nevertheless, (3.6) may be useful in other computational environments, e.g., in an SDP code written in a compiled language.

Remark 3.5. Alternatively, the complexity of computing the XZ direction can be reduced to

$$4mn^3 + m^2n^2 + O(\max\{m,n\}^3)$$

by first symmetrizing XA_jZ^{-1}:

$$m_{i,j} = A_i \bullet \mathbf{sym}(XA_jZ^{-1}). \tag{3.7}$$

Again, we note that in a computational environment such as MATLAB, using (3.7) is not likely to improve performance significantly.

Remark 3.6. The computation of the ZX direction is similar to that of the XZ direction. Actually, $(\Delta X, \Delta y, \Delta Z)$ is an XZ direction at (X, y, Z) if and only if $(\Delta X^T, \Delta y, \Delta Z)$ is a ZX direction at (X^T, y, Z).

This fact can be easily observed by taking the transpose of both sides of (1.10c) and by rewriting (1.10a) as $A_i \bullet \Delta X^T = b_i - A_i \bullet X^T$.

Table 3

Computational results for the control LMI and truss topology design problems, $\gamma = 0.98$. The starting point is $(X^0, y^0, Z^0) = \rho(I, 0, I)$. "Infeasibility" indicates the maximum of the relative primal and dual infeasibilities.

Problem	ρ	$\lvert \log_{10}(X \bullet Z) \rvert$		CPU seconds		Iterations		Infeasibility	
		AHO	XZ/ZX	AHO	XZ/ZX	AHO	XZ/ZX	AHO	XZ/ZX
truss1	10^3	11.36	10.93	3.0	1.9	16	16	3.4e-13	6.0e-13
truss2	10^3	11.59	9.19	144.7	122.9	17	18	1.0e-13	5.8e-13
truss3	10^3	8.57	8.80	22.1	17.3	18	19	8.8e-13	5.1e-13
truss4	10^3	11.32	11.33	6.0	4.5	16	18	4.9e-13	8.1e-13
truss5	10^3	11.08	10.39	2619	2465	19	21	1.9e-11	2.3e-11
truss6	10^3	4.70	5.43	2442	5738	18	24	3.2e-08	5.1e-07
truss7	10^3	3.74	3.78	721	1384	18	24	2.5e-13	5.3e-13
truss8	10^3	11.81	9.95	30813	27613	21	24	8.0e-10	9.7e-11
hinf1	10^2	3.80	8.42	3.7	4.6	12	21	1.1e-08	1.4e-09
hinf2	10^2	9.93	6.34	5.2	4.1	16	18	3.8e-07	1.7e-09
hinf3	10^2	10.22	7.38	5.8	3.8	17	19	6.2e-06	6.4e-07
hinf4	10^3	2.03	6.40	4.1	4.5	13	20	1.6e-07	6.4e-09
hinf5	10^3	10.06	4.17	5.5	4.3	18	19	2.3e-04	1.8e-05
hinf6	10^3	1.24	5.53	4.8	7.9	15	35	2.6e-05	1.1e-04
hinf7	10^3	6.22	4.98	4.8	4.9	16	22	1.5e-06	6.0e-06
hinf8	10^4	7.12	4.56	5.4	5.2	18	23	8.5e-07	5.4e-08
hinf9	10^3	9.53	9.45	5.0	3.9	17	19	1.7e-08	4.6e-06
hinf10	10^3	2.60	2.78	13.8	11.1	22	24	3.2e-07	3.4e-08
hinf11	10^6	5.85	3.45	24.7	21.6	21	23	5.2e-09	1.9e-07
hinf12	10^3	4.75	1.85	111.3	68.8	49	38	4.3e-05	1.5e-06
hinf13	10^4	0.74	1.92	139.2	150.3	22	33	2.8e-05	4.9e-07
hinf14	10^6	3.61	1.92	138.4	116.7	21	24	1.0e-08	5.9e-07
hinf15	10^2	1.15	0.43	385.3	271.6	37	35	3.3e-04	6.0e-05
hinf37	10^3	9.53	9.45	5.0	4.0	17	19	1.7e-08	4.6e-06

4. The XZ/ZX method. We call the algorithm described below an XZ/ZX method because it uses the XZ and ZX search directions alternately. It follows the Mehrotra predictor-corrector algorithmic framework of Todd, Toh, and Tütüncü [9].

Algorithm 4.1.

Select a starting point $(X^0, y^0, Z^0) \in \mathsf{R}^{n \times n} \times \mathsf{R}^n \times \mathcal{S}^n$ such that X^0 and Z^0 are positive definite. Choose an exponent ω and a constant $\gamma \in (0, 1)$.

Repeat for $k = 0, 1, 2, \ldots$:
[For simplicity, let $(X, y, Z) = (X^k, y^k, Z^k)$ and $(X^+, y^+, Z^+) = (X^{k+1}, y^{k+1}, Z^{k+1})$.]

(Predictor step)

- *Compute the predictor direction $(\delta X, \delta y, \delta Z)$ by solving the linear system (1.10) with $\sigma = 0$.*
- *Determine the parameter σ:*

$$\sigma := \left(\frac{(X + \psi \delta X) \bullet (Z + \phi \delta Z)}{X \bullet Z} \right)^{\omega}, \tag{4.1}$$

where

Table 4

*Computational results for the control LMI and truss topology design problems, $\gamma = 0.99$. The starting point is $(X^0, y^0, Z^0) = \rho(I, 0, I)$. "Infeasibility" indicates the maximum of the relative primal and dual infeasibilities. A * indicates that $\log_{10}(X \bullet Z) > 0$.*

Problem	ρ	$\lvert \log_{10}(X \bullet Z) \rvert$		CPU seconds		Iterations		Infeasibility	
		AHO	XZ/ZX	AHO	XZ/ZX	AHO	XZ/ZX	AHO	XZ/ZX
truss1	10^3	12.19	10.83	3.1	1.7	15	14	1.236e-11	4.155e-11
truss2	10^3	11.57	8.83	151.7	104.4	17	16	1.832e-10	1.652e-10
truss3	10^3	8.82	8.81	19.4	15.3	16	18	5.661e-11	3.381e-11
truss4	10^3	12.22	10.40	5.8	3.5	15	14	2.892e-11	5.675e-11
truss5	10^3	8.46	8.12	2165.7	1857.7	19	20	5.001e-11	6.523e-11
truss6	10^3	4.01	3.82	2060.3	6880.3	16	30	2.202e-11	4.170e-6
truss7	10^3	5.01	3.73	690.7	976.6	17	17	1.821e-11	8.247e-6
truss8	10^3	11.21	10.17	22310.5	15752.7	21	23	1.9e-10	1.012e-10
hinf1	10^3	4.84	4.37	4.2	3.1	13	14	6.8e-08	4.5e-09
hinf2	10^3	9.72	5.33	5.1	4.4	16	19	1.2e-08	5.9e-10
hinf3	10^3	10.65	6.49	5.1	3.6	16	16	5.3e-06	5.4e-07
hinf4	10^3	10.62	6.55	5.7	4.4	18	19	5.2e-08	1.2e-08
hinf5	10^3	5.52	5.35	1.3	4.4	4	19	4.6e+02	3.6e-04
hinf6	10^3	7.35	4.68	9.1	7.0	28	31	2.6e-04	2.3e-04
hinf7	10^3	7.79	3.43	4.4	5.1	14	23	5.4e-06	8.8e-06
hinf8	10^3	8.71	5.21	6.1	4.8	19	21	7.2e-06	5.3e-07
hinf9	10^3	10.94	9.89	5.4	5.8	17	26	8.1e-09	5.9e-05
hinf10	10^3	2.44	2.22	14.7	14.0	23	31	1.1e-06	6.7e-08
hinf11	10^3	2.53	3.20	22.2	23.0	18	25	4.2e-07	8.1e-07
hinf12	10^3	4.77	*	95.6	81.7	44	50	4.8e-05	3.3e-02
hinf13	10^3	0.43	0.68	68.0	98.9	18	32	6.6e-04	1.0e-05
hinf14	10^3	3.08	2.06	118.2	198.6	19	42	4.8e-08	1.5e-07
hinf15	10^3	0.82	*	275.6	191.2	29	27	2.1e-04	8.5e-05
hinf37	10^3	10.94	9.89	5.4	5.9	17	26	8.1e-09	5.9e-05

$$\text{(4.2a)} \qquad \psi := \frac{-\gamma}{\min\left(-\gamma,\ \lambda_{\min}(\mathbf{sym}(X)^{-1}\mathbf{sym}(\delta X)\right)},$$

$$\text{(4.2b)} \qquad \phi := \frac{-\gamma}{\min\left(-\gamma,\ \lambda_{\min}(Z^{-1}\delta Z)\right)}.$$

(Corrector step)

- *Compute the corrector direction $(\Delta X, \Delta y, \Delta Z)$ by solving linear system* (1.10) *with σ defined by* (4.1) *and the right side of* (1.10c) *modified as*

$$\sigma\mu I - XZ - \delta X \delta Z.$$

- *Compute ψ and ϕ from* (4.2) *with $\delta X, \delta Z$ replaced by $\Delta X, \Delta Z$.*
- *Update $(X^+, y^+, Z^+) = (X^T, y, Z) + (\psi \Delta X^T, \phi \Delta y, \phi \Delta Z)$.*

In our numerical implementation, we choose $\gamma = 0.99$ and set ω equal to 2 for the AHO method and 1 for others.

Remark 4.2. In Algorithm 4.1, through the updating

$$X^{k+1} := [X^k + \psi_k \Delta X^k]^T,$$

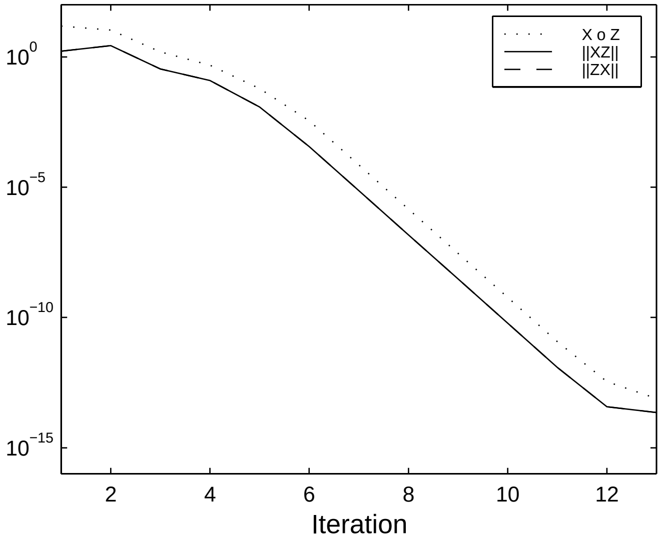


FIG. 1. *Convergence of AHO direction on a matrix norm minimization problem.*

we actually use the XZ and ZX directions alternately. In view of Remark 3.5, we see that Algorithm 4.1 is equivalent to an algorithm using the XZ and ZX directions alternately with the iteration sequence $\{(\tilde{X}^k,\ y^k,\ Z^k)\}$, where $\tilde{X}^k = X^k$ for $k = 2p-1$ and $\tilde{X}^k = (X^k)^T$ for $k = 2p,\ p \geq 1$. This property can be verified by a simple linear algebra manipulation.

5. Numerical results. We thank Toh, Todd, and Tütüncü for making their MATLAB code SDPT3 [11] available to us. We used their code for running the Mehrotra algorithm using the AHO, HKM, and NT search directions. We first tested the following problems:

1. random SDP problem with $n = 100,\ m = 50$,
2. random SDP problem with $n = 50,\ m = 100$,
3. random SDP problem with $n = 100,\ m = 100$,
4. the matrix norm minimization problem with $n = 100,\ m = 20$,
5. the problem of computing the Chebyshev polynomial of a matrix with $n = 100,\ m = 31$,
6. the Max-Cut problem with $n = 200,\ m = 200$,
7. the Educational Testing Problem (ETP) with $n = 110,\ m = 55$, and
8. the logarithmic Chebyshev approximation problem with $n = 300, m = 50$.

All the problems are taken from [9] and [11]. The reader is referred to [9] and [11] for details on the problems and the computation of the AHO, HKM, and NT search directions. We performed our numerical experiment using MATLAB 5.0. The computations were carried out on an IBM RS/6000 SP system at Argonne National Laboratory.

We tested 10 random instances for each problem. We stopped the computation when either no progress was made (due to numerical instability) or the number of iterations reached 50. The average results are given in Tables 1 and 2.

From the results displayed in the two tables we observe the following:

- The XZ/ZX method and the AHO method achieve higher accuracy than the

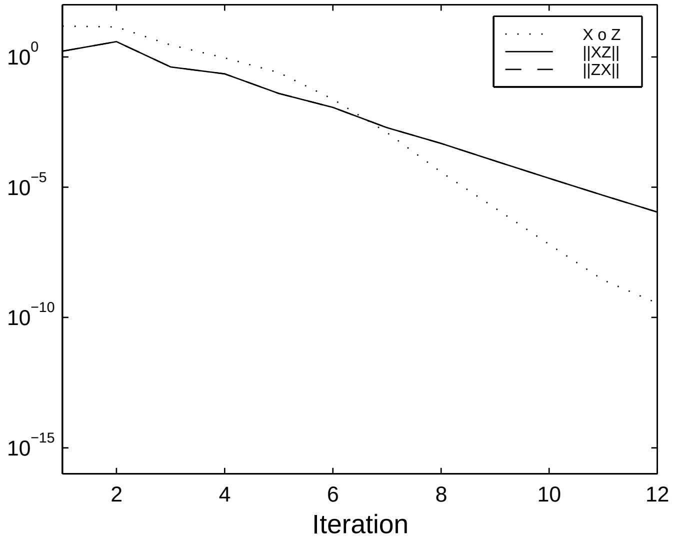


FIG. 2. *Convergence of HKM direction on a matrix norm minimization problem.*

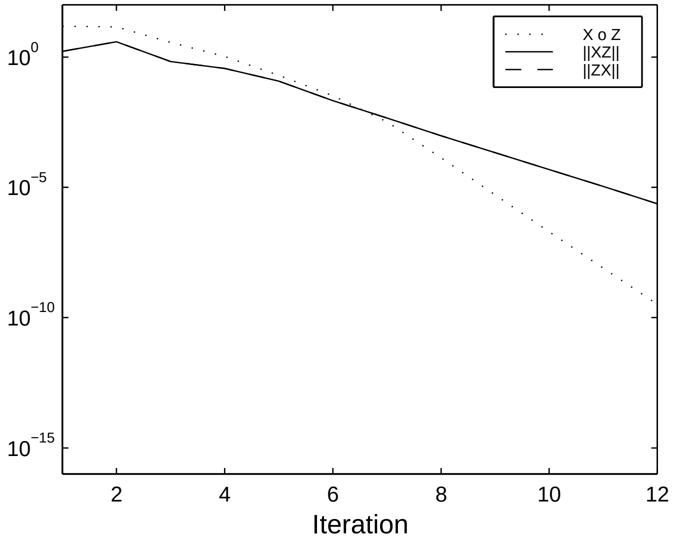


FIG. 3. *Convergence of NT direction on a matrix norm minimization problem.*

other methods for most problems.

- The XZ/ZX method is not able to achieve high accuracy on the ETP or logarithmic Chebyshev problems.
- In many cases the XZ/ZX method is faster than the AHO method.
- The XZ/ZX method takes about three more iterations than the AHO method with the exception of the ETP problem, where the XZ/ZX method takes significantly more iterations.
- With the exception of the ETP problem the XZ/ZX method requires signifi-

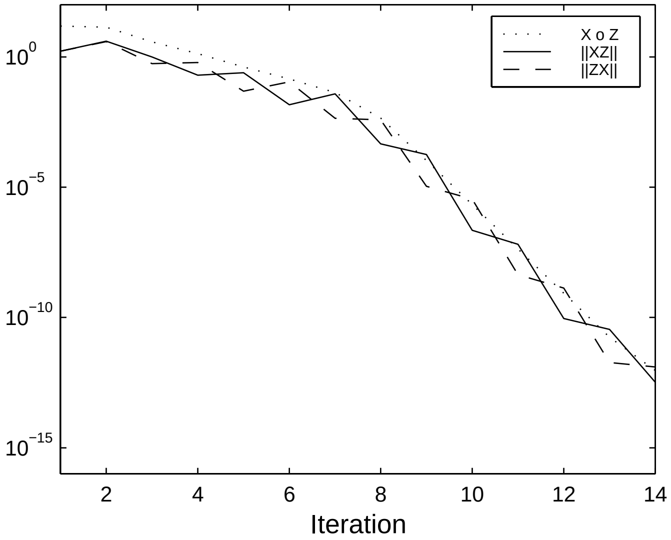


FIG. 4. *Convergence of XZ/ZX direction on a matrix norm minimization problem.*

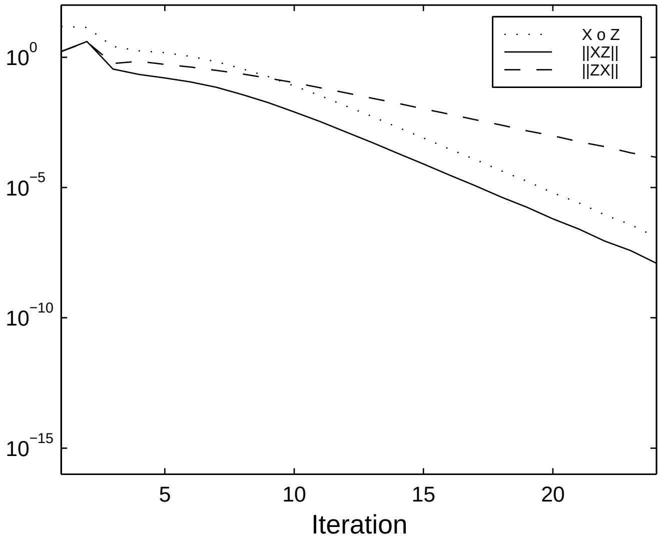


FIG. 5. *Convergence of XZ direction on a matrix norm minimization problem.*

cantly fewer flops per iteration than the AHO method and only slightly more flops than the HKM method, which requires the fewest flops per iteration of the methods tested.

The problems tested in Tables 1 and 2 were randomly generated and we used the starting points suggested in the MATLAB software SDPT3. For these problems, we chose $\gamma = 0.99$. We also tested 8 problems from truss topology design and 16 control linear matrix inequality problems. These problems have been used for the benchmarks of the MATLAB software SDPpack [1], which utilizes the AHO method, and are available online at http://cs.nyu.edu/cs/faculty/overton/sdppack/v0.9-beta/testproblems/.

The computational results for these problems are given in Tables 3 and 4. In Table 3, we chose $\gamma = 0.98$, and in Table 4 we were more aggressive and chose $\gamma = 0.99$. Here we also measured the relative infeasibility of our solution, which is given by

$$\max\{\|r\|/\|b\|, \|R_d\|_F/\|C\|_F\}.$$

For some of the test problems (e.g., hinf13 and hinf15), neither algorithm is capable of finding a solution of sufficient accuracy when $\gamma = 0.99$. The starting point we used is $(X^0, y^0, Z^0) = \rho(I, 0, I)$, where $\rho > 0$. Our results again show that the XZ/ZX method is competitive with the AHO method.

In addition, we conducted experiments to determine why the XZ/ZX method is so much more effective than the XZ method. To this end, we used SDPT3 to solve a matrix norm minimization problem of size $n = 100$, $m = 31$ using the XZ, XZ/ZX, AHO, NT, and HKM methods. Figures 1–5 represent the quantities $X \bullet Z$, $\|XZ\|_F$, and $\|ZX\|_F$ after each iteration of each algorithm. Of course, $\|XZ\|_F = \|ZX\|_F$ for the latter three directions since the iterates X^k and Z^k are symmetric.

Based on the graphs, we divide the five search directions into two distinct classes. The graphs of the HKM method, NT method, and XZ method appear quite similar. For these directions, after an initial period of 5 to 10 iterations, the duality gap $X \bullet Z$ decreases much faster than the quantities $\|XZ\|_F$ and $\|ZX\|_F$. The XZ method behaves much as we might expect; the quantity $\|XZ\|_F$ is in all cases smaller than $\|ZX\|_F$, which seems reasonable since the XZ method makes no attempt to decrease $\|ZX\|_F$.

The second class consists of the AHO and XZ/ZX methods. Figures 1 and 4 show that all three measured quantities decrease at roughly the same rate throughout both algorithms. However, the alternation between XZ and ZX directions in the XZ/ZX method produces a zigzag pattern. Even-numbered iterations force $\|XZ\|_F$ to decrease sharply, while odd-numbered iterations force $\|ZX\|_F$ to decrease sharply.

Acknowledgment. The authors thank Steve Wright for insightful discussions.

REFERENCES

[1] F. Alizadeh, J.-P. A. Haeberly, M. V. Nayakkankuppam, M. L. Overton, and S. Schmieta, *SDPpack User's Guide,* Technical report, New York University, June 1997; also available online from http://www.cs.nyu.edu/faculty/overton/sdppack/sdppack.html.

[2] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results*, SIAM J. Optim., 8 (1998), pp. 746–768.

[3] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. Optim., 6 (1996), pp. 342–361.

[4] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices*, SIAM J. Optim., 7 (1997), pp. 86–125.

[5] R. D. C. Monteiro, *Primal-dual path-following algorithms for semidefinite programming*, SIAM J. Optim., 7 (1997), pp. 663–678.

[6] R. D. C. Monteiro and T. Tsuchiya, *Polynomiality of primal-dual algorithms for semidefinite linear complementarity problems based on the Kojima-Shindoh-Hara family of directions*, Math. Program., 84 (1999), pp. 39–53.

[7] R. D. C. Monteiro and P. Zanjácomo, *Implementation of primal-dual methods for semidefinite programming based on Monteiro and Tsuchiya Newton directions and their variants*,

Working paper, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1997.

[8] Yu. E. Nesterov and M. J. Todd, *Primal-dual interior-point methods for self-scaled cones*, SIAM J. Optim., 8 (1998), pp. 324–364.

[9] M. J. Todd, K. C. Toh, and R. H. Tütüncü, *On the Nesterov-Todd direction in semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 769–796.

[10] K. C. Toh, *Search directions for primal-dual interior point methods in semidefinite programming*, SIAM J. Optim., submitted.

[11] K. C. Toh, M. J. Todd, and R. H. Tütüncü, *SDPT3—a Matlab software package for semidefinite programming*, Technical report 1177, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1998.

[12] Y. Zhang, *On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 365–386.

SIAM J. OPTIM.
Vol. 9, No. 4, pp. 877–900

AN INTERIOR POINT ALGORITHM FOR LARGE-SCALE NONLINEAR PROGRAMMING*

RICHARD H. BYRD[†], MARY E. HRIBAR[‡], AND JORGE NOCEDAL[§]

Dedicated to John Dennis, who has made crucial contributions to optimization, and has helped us greatly in our careers

Abstract. The design and implementation of a new algorithm for solving large nonlinear programming problems is described. It follows a barrier approach that employs sequential quadratic programming and trust regions to solve the subproblems occurring in the iteration. Both primal and primal-dual versions of the algorithm are developed, and their performance is illustrated in a set of numerical tests.

Key words. constrained optimization, interior point method, large-scale optimization, nonlinear programming, primal method, primal-dual method, sequential quadratic programming, barrier method, trust region method

AMS subject classifications. 65K10, 49N, 49M

PII. S1052623497325107

1. Introduction.

In this paper we discuss the design, implementation, and performance of an interior point method for solving the nonlinearly constrained optimization problem

$$\begin{aligned} &\min f(x) \\ &\text{subject to } h(x) = 0, \\ &\qquad\qquad g(x) \leq 0, \end{aligned} \tag{1.1}$$

where $f : \mathbf{R}^n \to \mathbf{R}$, $h : \mathbf{R}^n \to \mathbf{R}^t$, and $g : \mathbf{R}^n \to \mathbf{R}^m$ are smooth functions. We are particularly interested in the case when (1.1) is not a convex program and when the number of variables n is large. We assume in this paper that first and second derivatives of the objective function and constraints are available, but our strategy can be extended to make use of quasi-Newton approximations.

Interior point methods provide an alternative to active set methods for the treatment of inequality constraints. Our algorithm, which is based on the framework proposed by Byrd, Gilbert, and Nocedal [7], incorporates within the interior point method two powerful tools for solving nonlinear problems: sequential quadratic programming (SQP) and trust region techniques. SQP ideas are used to efficiently handle nonlinearities in the constraints. Trust region strategies allow the algorithm to treat convex and nonconvex problems uniformly, permit the direct use of second derivative information, and provide a safeguard in the presence of nearly dependent constraint gradients.

*Received by the editors July 30, 1997; accepted for publication (in revised form) January 25, 1999; published electronically September 24, 1999.

http://www.siam.org/journals/siopt/9-4/32510.html

[†] Computer Science Department, University of Colorado, Boulder, CO 80309 (richard@cs.colorado.edu). This author was supported by ARO grant DAAH04-94-0228 and AFOSR grant F49620-94-1-0101.

[‡] CAAM Department, Rice University, Houston, TX 77005 (marybeth@caam.rice.edu). This author was supported by Department of Energy grant DE-FG02-87ER25047-A004.

[§] ECE Department, Northwestern University, Evanston, IL 60208 (nocedal@ece.nwu.edu). This author was supported by National Science Foundation grant CCR-9625613 and by Department of Energy grant DE-FG02-87ER25047-A004.

Of crucial importance in the new algorithm are the formulation and solution of the equality constrained barrier subproblems that determine the steps of the algorithm. The formulation of the subproblems gives the iteration primal or primal-dual characteristics and ensures that the slack variables remain safely positive. The technique used to solve the subproblems has a great impact on the efficiency and robustness of the algorithm; we use an adaptation of the trust region method of Byrd [6] and Omojokun [32] which has proved to be effective for solving large equality constrained problems [29].

Our numerical results suggest that the new algorithm holds much promise: it appears to be robust and efficient (in terms of function evaluations), and it can make effective use of second derivative information. The test results also indicate that the primal-dual version of the algorithm is superior to the primal version. The new algorithm has a solid theoretical foundation, since it follows the principles of the globally convergent primal method developed in [7]. In particular, the approximate solution strategies for the subproblems in the algorithm are chosen to satisfy the explicit conditions for global convergence stated in that paper.

There has been much research in using interior point methods for nonlinear programming; most of it concerns line search methods. The special case when the problem is a *convex program* can be handled by line search methods that are direct extensions of interior point methods for linear programming (see, e.g., [1]). In the convex case, the step generated by the solution of the primal-dual equations can be shown to be a descent direction for several merit functions, and this allows one to establish global convergence results. Other research [18, 42] has focused on the *local behavior* of interior point line search methods for nonlinear programming. Conditions have been given that guarantee superlinear and quadratic rates of convergence. These algorithms can also be viewed as a direct extension of linear programming methods in that they do not make provisions for the case when the problem is nonconvex.

Several line search algorithms designed for *nonconvex* problems have recently been proposed [41, 20, 15, 21, 2, 33]. An important feature of many of these methods is a strategy for modifying the KKT system used in the computation of the search direction. This modification, which is usually based on a matrix factorization algorithm, ensures that the search direction is a descent direction for the merit function. These approaches are interesting, but there is not yet enough experience to fully evaluate their efficacy in general-purpose codes.

The use of trust region strategies in interior point methods for linear and nonlinear problems is not new [5, 31]. Coleman and Li [13, 12] proposed a primal method for bound constrained nonlinear optimization; see also [17]. Plantenga [34] developed an algorithm for general nonlinear programming that has some features in common with our algorithm; the main differences lie in his treatment of the trust region, in the purely primal nature of his step, and in the fact that his algorithm reverts to an active set method near the solution.

The algorithm proposed in this paper makes use of sequential quadratic programming techniques [3, 19, 23, 22] and in this sense is related to the line search algorithm of Yamashita [41]. But the way in which our algorithm combines trust region strategies, interior point approaches, and sequential quadratic programming techniques leads to an iteration that is different from those proposed in the literature.

2. The new algorithm.

The algorithm is a barrier method in which the subproblems are solved approximately by an SQP iteration with trust regions. Each barrier subproblem is of the form

$$
\begin{aligned}
&\min_{x,s} f(x) - \mu \sum_{i=1}^{m} \ln s_i \\
&\text{subject to } h(x) = 0, \\
&\qquad\qquad\quad g(x) + s = 0,
\end{aligned}
\tag{2.1}
$$

where $\mu > 0$ is the *barrier parameter* and where the slack variable s is assumed to be positive. By letting μ converge to zero, the sequence of solutions to (2.1) should normally converge to a stationary point of the original nonlinear program (1.1). As in some interior point methods for linear programming [40], our algorithm does not require feasibility of the iterates with respect to the inequality constraints in (1.1) but only forces the slack variables in (2.1) to remain positive.

To characterize the solution of the barrier problem (2.1) we introduce its Lagrangian,

$$
\mathcal{L}(x, s, \lambda_h, \lambda_g) = f(x) - \mu \sum_{i=1}^{m} \ln s_i + \lambda_h^T h(x) + \lambda_g^T (g(x) + s),
\tag{2.2}
$$

where λ_h and λ_g are the Lagrange multipliers. Rather than solving each barrier subproblem (2.1) accurately, we will be content with an approximate solution $(\hat{x}, \hat{s})$ satisfying $E(\hat{x}, \hat{s}; \mu) \le \epsilon_\mu$, where E measures the optimality conditions of the barrier problem and is defined by

$$
\begin{aligned}
E(x, s; \mu) = \max\big(&\|\nabla f(x) + A_h(x)\lambda_h + A_g(x)\lambda_g\|_\infty, \|S\lambda_g - \mu e\|_\infty, \|h(x)\|_\infty, \\
&\|g(x) + s\|_\infty\big).
\end{aligned}
\tag{2.3}
$$

Here $e = [1, \ldots, 1]^T$, $S = \text{diag}(s^1, \ldots, s^m)$, with superscripts indicating components of a vector, and

$$
A_h(x) = [\nabla h^1(x), \ldots, \nabla h^t(x)], \qquad A_g(x) = [\nabla g^1(x), \ldots, \nabla g^m(x)]
$$

are the matrices of constraint gradients. Throughout the paper we will assume that A_h has full column rank. In the definition of the optimality measure E, the vectors λ_h, λ_g are least squares multiplier estimates (to be discussed later) and thus are functions of x, s, and μ. We will show later (see (3.7)–(3.10)) that the terms in (2.3) correspond to each of the equations of the so-called perturbed KKT system upon which our primal-dual algorithm is based. The tolerance ϵ_μ, which determines the accuracy in the solution of the barrier problems, is decreased from one barrier problem to the next and must converge to zero. In this paper we will use the simple strategy of reducing both ϵ_μ and μ by a constant factor $\theta \in (0, 1)$. We test for optimality for the nonlinear program (1.1) by means of $E(x, s; 0)$.

Algorithm I. Barrier algorithm for solving the nonlinear problem (1.1).

```
Choose an initial value for the barrier parameter $\mu > 0$, and select
the parameters $\epsilon_\mu > 0$, $\theta \in (0, 1)$, and the final stop tolerance $\epsilon_{\text{TOL}}$.
Choose the starting point $x$ and $s > 0$, and evaluate the objective
function, constraints, and their derivatives at $x$.
**Repeat** until $E(x, s; 0) \le \epsilon_{\text{TOL}}$:
        1. Apply an SQP method with trust regions, starting from $(x, s)$,
            to find an approximate solution $(x^+, s^+)$ of the barrier
            problem (2.1) satisfying $E(x^+, s^+; \mu) \leq \epsilon_\mu$.
        2. Set $\mu \leftarrow \theta\mu,\ \epsilon_\mu \leftarrow \theta\epsilon_\mu,\ x \leftarrow x^+,\ s \leftarrow s^+$.
    end
```

To obtain a rapidly convergent algorithm, it is necessary to carefully control the rate at which the barrier parameter μ and the convergence tolerance ϵ_μ are decreased [18, 42]. This question has been studied, in the context of our algorithm, in [8].

Most of the work of Algorithm I lies clearly in step 1, in the approximate solution of an equality constrained problem with an implicit lower bound on the slack variables. The challenge is to perform this step efficiently, even when μ is small, while forcing the slack variables to remain positive. To do this we apply an adaptation of the equality constrained SQP iteration with trust regions proposed by Byrd [6] and Omojokun [32] and developed by Lalee, Nocedal, and Plantenga [29] for large-scale equality constrained optimization. We follow an SQP approach because, in our view, it is effective for solving equality constrained problems, even when the problem is ill-conditioned and the constraints are highly nonlinear (see also [3, 22, 19, 23]), and we choose to use trust region strategies to globalize the SQP iteration because they facilitate the use of second derivative information when the problem is nonconvex.

However, our numerical experience shows that a straightforward application of this SQP method to the barrier problem leads to inefficient steps that tend to violate the positivity of the slack variables and that are thus frequently cut short by the trust region constraint. The novelty of our approach lies in the formulation of the quadratic model in the SQP iteration and in the definition of the (scaled) trust region. These are designed to produce steps that have some of the properties of primal-dual iterations and that avoid approaching the boundary of the feasible region too soon.

In order to describe our approach more precisely, it is instructive to briefly review the basic principles of SQP for equality constrained optimization with trust regions [3, 9, 10, 29, 38]. Every iteration of such an SQP method begins by constructing a quadratic model of the Lagrangian function. A step d of the algorithm is computed by minimizing the quadratic model, subject to satisfying a linear approximation to the constraints and subject to a trust region bound on this step. If the step d gives a sufficient reduction in the chosen merit function, then it is accepted; otherwise the step is rejected, the trust region is reduced, and a new step is computed.

Let us apply these ideas to the barrier problem (2.1), in order to compute a step $d = (d_x, d_s)$ from the current iterate (x_k, s_k). To economize space we will often write vectors with x- and s-components as

$$\begin{pmatrix} d_x \\ d_s \end{pmatrix} = (d_x, d_s).$$

After computing Lagrange multiplier estimates (λ_h, λ_g), we formulate the subproblem

$$\min_{d_x, d_s} \ \nabla f(x_k)^T d_x + \frac{1}{2} d_x^T \nabla_{xx}^2 \mathcal{L}(x_k, s_k, \lambda_h, \lambda_g) d_x - \mu e^T S_k^{-1} d_s + \frac{1}{2} d_s^T \Sigma_k d_s \tag{2.4}$$

$$\text{subject to } A_h(x_k)^T d_x + h(x_k) = r_h, \tag{2.5}$$

$$A_g(x_k)^T d_x + d_s + g(x_k) + s_k = r_g, \tag{2.6}$$

$$(d_x, d_s) \in T_k. \tag{2.7}$$

Here Σ_k is an $m \times m$ positive definite diagonal matrix that represents either the Hessian of the Lagrangian (2.2) with respect to s or an approximation to it. As we will see in the next section, the choice of Σ_k is of crucial importance because it determines whether the iteration has primal or primal-dual characteristics. Ideally, we would like our step to satisfy (2.5)–(2.6) with $r = (r_h, r_g) = 0$, i.e., to satisfy the linearized constraints. However, this may be inconsistent with (2.7), so we choose the residual vector r to be the smallest vector such that (2.5)–(2.7) are consistent (with some margin). This computation is done by solving the preliminary subproblem in which we compute the *normal step*, described in section 3.2. The closed and bounded set T_k defines the region around x_k where the quadratic model (2.4) and the linearized constraints (2.5)–(2.6) can be trusted to be good approximations to the problem, and it also ensures the feasibility of the slack variables. This trust region also guarantees that (2.4)–(2.7) has a finite solution even when $\nabla^2_{xx}\mathcal{L}(x_k, s_k, \lambda_h, \lambda_g)$ is not positive definite. The precise form of the trust region T_k requires careful consideration and will be described in the next section.

We compute a step $d = (d_x, d_s)$ by approximately minimizing the quadratic model (2.4) subject to the constraints (2.5)–(2.7), as will be described in section 3.2. We then determine if the step is acceptable according to the reduction obtained in the following merit function:

$$\phi(x, s; \nu) = f(x) - \mu \sum_{i=1}^{m} \ln s_i + \nu \left\| \begin{bmatrix} h(x) \\ g(x) + s \end{bmatrix} \right\|_2, \tag{2.8}$$

where $\nu > 0$ is a penalty parameter. This nondifferentiable merit function has been successfully used in the SQP algorithm of Byrd [6] and Omojokun [32] and has been analyzed in the context of interior point methods in [7]. We summarize this SQP trust region approach as follows.

Algorithm II. SQP trust region algorithm for the barrier problem (2.1).

```
Input parameters $\mu > 0$ and $\epsilon_\mu > 0$ and values $k$, $x_k$, and $s_k > 0$;
set trust region $T_k$; compute Lagrange multipliers $\lambda_h$ and $\lambda_g$.
Repeat until $E(x_k, s_k; \mu) \le \epsilon_\mu$
        Compute $d = (d_x, d_s)$ by approximately solving (2.4)–(2.7).
        If the step $d$ provides sufficient decrease in $\phi$
            then set $x_{k+1} = x_k + d_x$, $s_{k+1} = s_k + d_s$,
                compute new Lagrange multiplier estimates $\lambda_h$ and $\lambda_g$,
                and possibly enlarge the trust region;
            else set $x_{k+1} = x_k$, $s_{k+1} = s_k$, and shrink the trust region.
        Set $k := k + 1$.
end
```

Algorithm II is called at each execution of step 1 of Algorithm I. The iterates of Algorithm II are indexed by (x_k, s_k), where the index k runs continuously during Algorithm I. In the next section we present a full description of Algorithm II, which forms the core of the new interior point algorithm.

3. Algorithm for solving the barrier problem. Many details of the SQP trust region method outlined in Algorithm II need to be developed. We first give a precise description of the subproblem (2.4)–(2.7), including the choice of the diagonal matrix Σ_k which gives rise to primal or primal-dual iterations. Furthermore, we define the right-hand vectors (r_h, r_g), the form of the trust region constraint T_k, and the choice of Lagrange multiplier estimates. Once a complete description of the subproblem (2.4)–(2.7) has been given, we will present our procedure for finding an approximate solution of it. We will conclude this section with a discussion of various other details of implementation of the new algorithm.

3.1. Formulation of the subproblem. Let us begin by considering the quadratic model (2.4). We have mentioned that SQP methods choose the Hessian of this model to be the Hessian of the Lagrangian of the problem under consideration, or an approximation to it. Since the problem being solved by Algorithm II is the barrier problem (2.1), which has a separable objective function in the variables x and s, its Hessian consists of two blocks. As indicated in (2.4), we choose the Hessian of the quadratic model with respect to d_x to be $\nabla^2_{xx}\mathcal{L}(x_k, s_k, \lambda_h, \lambda_g)$ (which we abbreviate as $\nabla^2_{xx}\mathcal{L}_k$) but consider several choices for the Hessian Σ_k of the model with respect to d_s. The first choice is to define $\Sigma_k = \nabla^2_{ss}\mathcal{L}_k$, which gives

$$\Sigma_k = \mu S_k^{-2}. \tag{3.1}$$

The general algorithm studied in Byrd, Gilbert, and Nocedal [7] defines Σ_k in this manner.

To study the effect of Σ_k in the step computation, let us analyze the simple case when the matrix $\nabla^2_{xx}\mathcal{L}_k$ is positive definite on the null space of the constraint gradients, when the residual (r_h, r_g) is zero, and when the step generated by (2.4)–(2.7) lies strictly inside the trust region. In this case the subproblem (2.4)–(2.6) has a unique solution $d = (d_x, d_s)$ which satisfies the linear system

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}_k & 0 & A_h(x_k) & A_g(x_k) \\ 0 & \Sigma_k & 0 & I \\ A_h^T(x_k) & 0 & 0 & 0 \\ A_g^T(x_k) & I & 0 & 0 \end{bmatrix} \begin{bmatrix} d_x \\ d_s \\ \lambda_h^+ \\ \lambda_g^+ \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) \\ \mu S_k^{-1} e \\ -h(x_k) \\ -g(x_k) - s_k \end{bmatrix}. \tag{3.2}$$

If Σ_k is defined by (3.1), we call this approach a *primal method.* In this case, it is easy to verify (see, e.g., [18, 40, 7]) that the system (3.2) is equivalent to a Newton iteration on the KKT conditions of the barrier problem (2.1), which are given by

$$\nabla f(x) + A_h(x)\lambda_h + A_g(x)\lambda_g = 0, \tag{3.3}$$

$$-\mu S^{-1} e + \lambda_g = 0, \tag{3.4}$$

$$h(x) = 0, \tag{3.5}$$

$$g(x) + s = 0. \tag{3.6}$$

Several authors, including Jarre and S. Wright [28], M. Wright [39], and Conn, Gould, and Toint [16] have given arguments suggesting that the primal search direction will often cause the slack variables to become negative, and that it can be inefficient. Although those papers consider a different formulation of the problem, it is easy to see [27] that the arguments apply in our case.

Research in linear programming [40] has shown that a more effective interior point method is obtained by considering the *perturbed KKT system*

$$\nabla f(x) + A_h(x)\lambda_h + A_g(x)\lambda_g = 0, \tag{3.7}$$

$$S\lambda_g - \mu e = 0, \tag{3.8}$$

$$h(x) = 0, \tag{3.9}$$

$$g(x) + s = 0, \tag{3.10}$$

which is obtained by multiplying (3.4) by S. Although (3.4)–(3.6) and (3.8)–(3.10) have the same solutions, applying Newton's method to them will produce different iterates. It is well known, and also easy to verify, that a Newton step on (3.8)–(3.10) is given by the solution to (3.2), with

$$\Sigma_k = S_k^{-1}\Lambda_g. \tag{3.11}$$

Here Λ_g =diag$(\lambda_g^1, \ldots, \lambda_g^m)$ contains the Lagrange multiplier estimates corresponding to the inequality constraints. The system (3.2) with Σ_k defined by (3.11) is called the primal-dual system. This choice of Σ_k may be viewed as an approximation to $\nabla^2_{ss}\mathcal{L}_k$ since, by (3.4), at the solution (x, s, λ) of the barrier problem the equation $\mu S^{-1} = \Lambda_g$ is satisfied. Substituting this equation in (3.1) gives (3.11).

The system (3.7)–(3.10) has the advantage that the derivatives of (3.8) are bounded as any slack variables approach zero, which is not the case with (3.4). In fact, analysis of the primal-dual step, as well as computational experience with linear programs, has shown that it overcomes the drawbacks of the primal step: it does not tend to violate the constraints on the slacks, and it usually makes excellent progress toward the solution (see, e.g., [28, 39, 40, 37]). These observations suggest that the primal-dual model in which Σ_k is given by (3.11) is likely to perform better than the primal choice (3.1). Of course, these arguments do not apply directly to our algorithm which solves the SQP subproblem inexactly and whose trust region constraint may be active. Nevertheless, as the iterates approach a solution point, the algorithm will resemble more and more an interior point method in which a Newton step on some form of the KKT conditions of the barrier problem is taken at each step.

Lagrange multiplier estimates are needed both in the primal-dual choice (3.11) of Σ_k and in the Hessian $\nabla^2\mathcal{L}_{xx}(x_k, s_k, \lambda_h, \lambda_g)$. To complete our description of the quadratic model (2.4) we must discuss how these multipliers are computed.

Lagrange multipliers. Since the method we will use for finding an approximate solution to the subproblem (2.4)–(2.7) does not always provide Lagrange multiplier estimates as a side computation, we will obtain them using a least squares approach. As is done in some SQP methods [19, 3], which compute least squares estimates based on the stationarity conditions at the current iterate, we will choose the vector $\lambda = (\lambda_h, \lambda_g)$ that minimizes the Euclidean norm of (3.7)–(3.8). This gives the formula

$$\lambda_k = \begin{bmatrix} \lambda_h \\ \lambda_g \end{bmatrix} = \lambda^{LS}(x_k, s_k, \mu) = \left(\hat{A}_k^T \hat{A}_k\right)^{-1} \hat{A}_k^T \begin{bmatrix} -\nabla f(x_k) \\ \mu e \end{bmatrix}, \tag{3.12}$$

where

$$\hat{A}_k = \begin{bmatrix} A_h(x_k) & A_g(x_k) \\ 0 & S_k \end{bmatrix}. \tag{3.13}$$

The computation of (3.12) will be performed by solving an augmented system, instead of factoring $\hat{A}_k^T \hat{A}_k$, as will be discussed in section 3.4.

We should note that the multiplier estimates λ_g obtained in this manner may not always be positive, and it may be questionable to use them in this case in the primal-dual choice of Σ_k given by (3.11). In particular, since the Hessian of the barrier term $-\mu \sum \ln s_i$ is known to be positive definite, it seems undesirable to create an indefinite approximation Σ_k to it. On the other hand, one could argue that trust region methods can handle indefinite approximations and therefore that the multipliers need not be modified. We cannot see a compelling argument in favor of either strategy. In primal-dual interior point methods for linear programming, the initial Lagrange multiplier estimate is chosen to be positive, and in subsequent iterations a backtracking line search ensures that all new multiplier estimates remain safely positive (see, e.g., [40]). Here we follow a different approach, not enforcing the positivity of the multipliers λ_g but ensuring that the quadratic model remains convex in the slack variables. To do so, in the primal-dual version of the algorithm we define the ith diagonal element of Σ_k as

$$\sigma_k^i = \begin{cases} \lambda_g^i/s^i & \text{if } \lambda_g^i > 0, \\ \mu/(s^i)^2 & \text{otherwise.} \end{cases} \tag{3.14}$$

This means, in particular, that when a multiplier λ_g^i given by (3.12) is negative, the corresponding entry in the primal-dual matrix Σ_k coincides with the corresponding entry in the primal Hessian.

To avoid an abrupt change in Σ_k when μ is decreased, we modify the definition of λ_k slightly in the primal-dual version of the algorithm. If (x_k, s_k) is the starting point for a new barrier subproblem (i.e., the input in Algorithm II), then in the formula (3.14) λ_g is the multiplier from the last iterate of the previous barrier problem.

Thus the definition of the multipliers is

$$\lambda_k = \begin{cases} \lambda^{LS}(x_k, s_k, \mu) & \text{in primal version,} \\ \lambda^{LS}(x_k, s_k, \bar{\mu}) & \text{in primal-dual version,} \end{cases} \tag{3.15}$$

where $\bar{\mu}$ is the value of the barrier parameter used in the computation of (x_k, s_k). As mentioned earlier, other strategies for computing multiplier estimates can be used, and we do not yet know which choice might be preferable in practice.

This approach could just barely be considered a primal-dual method, as other primal-dual methods treat the multipliers λ_h, λ_g as independent variables. In that respect our approach is much closer to those SQP methods where the multipliers have a subordinate role, being estimated as a function of the primal variables, and not appearing explicitly in the merit function.

The trust region. Algorithm II stipulates that the step (d_x, d_s) must be restricted to a set T_k, called the trust region. We will define T_k to accomplish two goals. First, it should restrict the step to a region where the quadratic model (2.4) is a good approximation of the Lagrangian (2.2) and where the linear equations (2.5)–(2.6) are good approximations to the constraints. This is the basic philosophy of trust regions and is normally achieved by imposing a bound of the form $\|(d_x, d_s)\| \leq \Delta_k$, where the trust region radius Δ_k is updated at every iteration according to how successful the step has been.

We will impose such a bound on the step, but the shape of the trust region must also take into account other requirements of Algorithm II. Since the slack variables should not approach zero prematurely, we introduce the scaling S_k^{-1} that penalizes steps d_s near the boundary of the feasible region. This scaled trust region will be defined as

$$\|(d_x, S_k^{-1} d_s)\|_2 \leq \Delta_k \tag{3.16}$$

and we will allow Δ_k to be greater than 1. The second objective of our trust region is to ensure that the slack variables remain positive. For this purpose we impose the well-known [40, 37] fraction to the boundary rule

$$s_k + d_s \geq (1 - \tau) s_k, \tag{3.17}$$

where $\tau \in (0,1)$; in our tests we use $\tau = 0.995$. Combining this inequality, which can be rephrased as $d_s \geq -\tau s_k$, with (3.16) we obtain the final form of the trust region,

$$\|(d_x, S_k^{-1} d_s)\|_2 \leq \Delta_k \quad \text{and} \quad d_s \geq -\tau s_k. \tag{3.18}$$

We have experimented with other forms of the trust region, in particular with box-shaped trust regions defined by an ℓ_∞ norm, but so far (3.18) appears to be the most appropriate for our algorithm.

Now that the quadratic model (2.4) and the trust region (2.7) have been defined, it remains only to specify the choice of the residual vector $r = (r_h, r_g)$ in (2.5)–(2.6). This vector will be determined during the course of solving the subproblem, as discussed next.

3.2. Solution of the quadratic subproblem. We will use the decomposition proposed by Byrd [6] and Omojokun [32] to find an approximate solution of the subproblem (2.4)–(2.7). In this approach the step d is a combination of a *normal step* that attempts to satisfy the linear constraints (2.5)–(2.6) as well as possible and a *tangential step* that lies on the tangent space of the constraints and that tries to achieve optimality. The efficiency of the new algorithm depends, to a great extent, on how these two components of the step are computed.

Throughout this section we omit the iteration subscript and write s_k as s, $A_h(x_k)$ as A_h, etc.

Normal step. It is clear [38] that restricting the size of the step d by means of the trust region bounds (3.18) may preclude d from satisfying the linearized constraints (2.5)–(2.6) with $r = 0$. To find a value of r that makes the quadratic subproblem feasible, we first compute the normal step v that lies well within the trust region and that approximately satisfies (2.5)–(2.6), in the least squares sense. To do this, we choose a parameter $\zeta \in (0,1)$ (in our code we use the value $\zeta = 0.8$) and formulate the following subproblem in the variable $v = (v_x, v_s)$

$$\begin{aligned} &\min_v \|A_h^T v_x + h\|_2^2 + \|A_g^T v_x + v_s + g + s\|_2^2 \\ &\quad \text{subject to} \quad \|(v_x, S^{-1} v_s)\|_2 \leq \zeta\Delta, \\ &\qquad\qquad\qquad\qquad v_s \geq -\tau s/2. \end{aligned} \tag{3.19}$$

To simplify the constraints we define

$$\tilde{v} = (v_x, \tilde{v}_s) = (v_x, S^{-1} v_s).$$

Performing this transformation, recalling the definition (3.13) of $\hat{A}$, squaring and expanding the quadratic objective, and ignoring constant terms, we obtain

$$\min_{\tilde{v}} \ m(\tilde{v}) \equiv 2 \begin{bmatrix} h^T & (g+s)^T \end{bmatrix} \hat{A}^T \begin{bmatrix} v_x \\ \tilde{v}_s \end{bmatrix} + \begin{bmatrix} v_x^T & \tilde{v}_s^T \end{bmatrix} \hat{A}\hat{A}^T \begin{bmatrix} v_x \\ \tilde{v}_s \end{bmatrix} \tag{3.20}$$

$$\text{subject to} \quad \|\tilde{v}\|_2 \leq \zeta\Delta, \tag{3.21}$$

$$\tilde{v}_s \geq -\tau/2. \tag{3.22}$$

We compute an approximate solution of this problem by means of an adaptation of the *dogleg method* [35], which provides a relatively inexpensive solution that is good enough to allow our algorithm to be robust and rapidly convergent. Like the dogleg method, it provides at least as much decrease on (3.19) as a truncated steepest descent step, and it equals the unconstrained minimizer of (3.19) if that vector satisfies the constraints of the subproblem. This first property, together with the fact that it lies in the range space of $\hat{A}$, implies that the normal step satisfies the conditions for global convergence given in [7].

We first calculate the Cauchy point $\tilde{v}^{CP}$ for problem (3.20)–(3.21), which is obtained by minimizing the quadratic (3.20) along the steepest descent direction, starting from $\tilde{v} = 0$. A simple computation shows that

$$\tilde{v}^{CP} = \begin{bmatrix} v_x^{CP} \\ \tilde{v}_s^{CP} \end{bmatrix} = -\alpha \hat{A} \begin{bmatrix} h \\ g+s \end{bmatrix}, \tag{3.23}$$

where α is given by

$$\alpha = \frac{\left\| \hat{A} \begin{bmatrix} h \\ g+s \end{bmatrix} \right\|_2^2}{\begin{bmatrix} h^T & g^T + s^T \end{bmatrix} (\hat{A}^T \hat{A})^2 \begin{bmatrix} h \\ g+s \end{bmatrix}}.$$

Note that this computation is inexpensive, requiring only matrix-vector multiplications and no matrix factorizations.

We then compute the Newton step $\tilde{v}^N$, which in our case is defined as the minimum norm minimizer of (3.20). It is given by

$$\tilde{v}^N = \begin{bmatrix} v_x^N \\ \tilde{v}_s^N \end{bmatrix} = -\hat{A}(\hat{A}^T \hat{A})^{-1} \begin{bmatrix} h \\ g+s \end{bmatrix}. \tag{3.24}$$

The computation of $\tilde{v}^N$ will be done by solving an augmented system, instead of factoring $\hat{A}^T \hat{A}$, as will be discussed in section 3.4.

The Cauchy and Newton steps define the dogleg path, which consists of the two line segments from $\tilde{v} = 0$ to $\tilde{v} = \tilde{v}^{CP}$ and from $\tilde{v} = \tilde{v}^{CP}$ to $\tilde{v} = \tilde{v}^N$. We compute the normal step by minimizing $m(\tilde{v})$ subject to (3.21) and (3.22) along this path and along the Newton direction, as described below.

Dogleg procedure.

```
    Compute $\tilde{v}^{CP}$ and $\tilde{v}^N$.
    $\theta_1 = \max\{\theta \in (0,1] | \theta\tilde{v}^N$ is feasible$\}$
    **If** $\theta_1 = 1$ **then**
        $\tilde{v} = \tilde{v}^N$
    **Else**
        $\theta_2 = \max\{\theta \in (0,1] | (1-\theta)\tilde{v}^{CP} + \theta\tilde{v}^N$ is feasible for (3.21) and (3.22)$\}$
        **If** no such value $\theta_2$ exists **then**
            $\theta_3 = \max\{\theta \in (0,1] | \theta\tilde{v}^{CP}$ is feasible$\}$
            $\tilde{v}^{DL} = \theta_3\tilde{v}^{CP}$
        **Else**
            $\tilde{v}^{DL} = (1-\theta_2)\tilde{v}^{CP} + \theta_2\tilde{v}^N$
        **Endif**
        **If** $m(\tilde{v}^{DL}) < m(\theta_1\tilde{v}^N)$ **then**
            $\tilde{v} = \tilde{v}^{DL}$
        **Else**
            $\tilde{v} = \theta_1\tilde{v}^N$
        **Endif**
    **Endif**
    $v = (v_x, S\tilde{v}_s)$
```

Since the model function m is convex, it decreases along the dogleg path, and thus the dogleg point $\tilde{v}^{DL}$ minimizes m along that path, subject to (3.21) and (3.22). Note that even if $\tilde{v}^{CP}$ and $\tilde{v}^{N}$ are infeasible, the line from $\tilde{v}^{CP}$ to $\tilde{v}^{N}$ may still contain a feasible segment. Also, to try to achieve a greater reduction in the model function, we compare the dogleg step with the Newton step truncated to the feasible region and choose whichever of these two points gives a lower value of m. Finally, we obtain the normal step by transforming $\tilde{v}$ into the original space of variables.

For future reference we note that the step $\tilde{v}$ lies in the range space of $\hat{A}$; see (3.23) and (3.24).

An alternative to the dogleg method is to compute the normal step by means of Steihaug's implementation of the conjugate gradient method [36]. This is described in detail in [27] (see also [29]), and it is certainly a viable option. We prefer the dogleg method in this study because it allows us to compute the normal step using a direct linear algebra solver, thereby avoiding the difficulties that can arise when applying the conjugate gradient method to ill-conditioned systems. In addition, the matrix factorization performed during the computation of the Lagrange multipliers can be saved and used to compute the normal step, giving significant savings in computation. We will return to this in section 3.4.

Tangential problem. Once the normal step v is computed, we define the vectors r_h and r_g in (2.5)–(2.6) as the residuals in the normal step computation, i.e.,

$$r_h = A_h^T v_x + h, \qquad r_g = A_g^T v_x + v_s + g + s.$$

The subproblem (2.4)–(2.7) therefore takes the form

$$\min \nabla f^T d_x - \mu e^T S^{-1} d_s + \frac{1}{2}(d_x^T \nabla_{xx}^2 \mathcal{L} d_x + d_s^T \Sigma d_s) \tag{3.25}$$

$$\text{subject to} \quad A_h^T d_x = A_h^T v_x, \tag{3.26}$$

$$A_g^T d_x + d_s = A_g^T v_x + v_s, \tag{3.27}$$

$$\|(d_x, S^{-1} d_s)\|_2 \le \Delta, \tag{3.28}$$

$$d_s \ge -\tau s. \tag{3.29}$$

We will devote much attention to this subproblem, whose solution represents the most complex and time-consuming part of the new algorithm.

Let us motivate our choice of the residual vectors r_h and r_g. First, the constraints (3.26)–(3.29) are now feasible since $d = v$ clearly satisfies them (recall that $\zeta < 1$ in (3.19)). Second, we are demanding that the total step d makes as much progress toward satisfying the constraints (3.26)–(3.27) as the normal step v.

To find an approximate solution of (3.25)–(3.29), we write $d = v + w$, where v is the normal step and w, which is to be determined, is tangent to the (scaled) constraint gradients. Introducing the same change of variables as in the normal step computation, we define

$$\tilde{d} = \begin{pmatrix} \tilde{d}_x \\ \tilde{d}_s \end{pmatrix} = \begin{pmatrix} d_x \\ S^{-1} d_s \end{pmatrix} = \begin{pmatrix} v_x \\ \tilde{v}_s \end{pmatrix} + \begin{pmatrix} w_x \\ \tilde{w}_s \end{pmatrix} = \tilde{v} + \tilde{w}. \tag{3.30}$$

Using this and defining

$$G = \begin{bmatrix} \nabla_{xx}^2 \mathcal{L} & 0 \\ 0 & S \Sigma S \end{bmatrix}, \tag{3.31}$$

the objective of (3.25) can be expressed as

$$q(\tilde v+\tilde w) \;\equiv\; (\nabla f^T,\ -\mu e^T)(\tilde v+\tilde w)+\frac{1}{2}(\tilde v+\tilde w)^T G(\tilde v+\tilde w). \tag{3.32}$$

The constraint (3.28) can be rewritten as

$$\|\tilde d\|_2^2=\|\tilde v+\tilde w\|_2^2\le \Delta^2. \tag{3.33}$$

We have noted in section 3.2 that the (scaled) normal step $\tilde v$ lies in the range space of $\hat A$, and we will require that w satisfies $\hat A^T\tilde w=0$. Thus $\tilde w^T\tilde v=0$, and (3.28) can be expressed as

$$\|\tilde w\|_2^2\le \Delta^2-\|\tilde v\|_2^2.$$

Using this, (3.32), and the definitions (3.30), we can rewrite (3.25)–(3.29) as

$$\min_{\tilde w}\ q(\tilde v+\tilde w)\equiv q(\tilde v)+\nabla f^T w_x-\mu e^T\tilde w_s+(G\tilde v)^T\tilde w+\frac{1}{2}(\tilde w^T G\tilde w) \tag{3.34}$$

$$\text{subject to}\quad A_h^T w_x=0, \tag{3.35}$$

$$A_g^T w_x+S\tilde w_s=0, \tag{3.36}$$

$$\|\tilde w\|_2^2\le \Delta^2-\|\tilde v\|_2^2, \tag{3.37}$$

$$\tilde w_s\ge -\tau e-\tilde v_s. \tag{3.38}$$

We call this the *tangential* subproblem. Clearly this subproblem can be very expensive to solve. However, the shape of the feasible region for this problem resembles a trust region in that the boundaries of the feasible region are never close to the origin ($\tilde w=0$) in the scaled coordinates. So it is reasonable to expect that an adaptation of a method for computing an approximate solution of a trust region problem, such as the conjugate gradient (CG) iteration proposed by Steihaug, will be efficient in this context. We will follow this approach and apply the CG method to the quadratic objective (3.34) while forcing the CG iterates to satisfy the constraints (3.35)–(3.36). To take into account the trust region and the possibility of indefiniteness in the model, we will terminate the CG iteration using the stopping tests of Steihaug [36]. We will also precondition the CG iteration.

Rather than simply presenting this CG iteration, we will now describe in detail the steps that lead to it, and we will motivate our preconditioning strategy.

Since $\tilde w$ is assumed to lie in the null space of $\hat A^T$, it can be expressed as

$$\tilde w=\tilde Z u\equiv\begin{pmatrix} Z_x\\ \tilde Z_s\end{pmatrix}u \tag{3.39}$$

for some vector $u\in\mathbf{R}^{n-t}$, where $\tilde Z$ is a basis for the null space of $\hat A^T$. The constraints (3.35)–(3.36) can be written as $\hat A^T\tilde w=0$ and are therefore satisfied by any $\tilde w$ of the form (3.39). Therefore the tangential problem (3.34)–(3.38) can be stated as

$$\min_u\ q(\tilde v+\tilde Z u) \tag{3.40}$$

$$\text{subject to}\quad \|\tilde Z u\|_2^2\le \Delta^2-\|\tilde v\|_2^2,$$

$$\tilde Z_s u\ge -\tau e-\tilde v_s. \tag{3.41}$$

We will precondition the CG iteration because, if we were to apply unpreconditioned CG for minimizing (3.40), a poor choice of Z could cause the CG iteration to be very slow. To see this, note that the Hessian of (3.40) is

$$\tilde{Z}^T G \tilde{Z},$$

and a poor choice of Z could make this matrix unnecessarily ill-conditioned. Such a poor choice of null space basis could occur, for example, when using the easily computable basis

$$\tilde{Z} = \begin{bmatrix} \hat{A}_1^{-1} \hat{A}_2 \\ -I \end{bmatrix}$$

based on the basic–nonbasic partition $\hat{A}^T = [\hat{A}_1 \ \hat{A}_2]$. This problem can be avoided by preconditioning the CG iteration for minimizing (3.40) by the matrix

$$\tilde{Z}^T \tilde{Z}, \tag{3.42}$$

in which case the rate of convergence is governed by the spectrum of

$$(\tilde{Z}^T \tilde{Z})^{-\frac{1}{2}} \tilde{Z}^T G \tilde{Z} (\tilde{Z}^T \tilde{Z})^{-\frac{1}{2}}. \tag{3.43}$$

Since the matrix $\tilde{Z}(\tilde{Z}^T \tilde{Z})^{-\frac{1}{2}}$ has orthonormal columns, the behavior of the CG iteration will now be identical to that obtained when $\tilde{Z}$ is a basis with orthonormal columns. Note also from (3.4) that $\mu S^{-1} \approx \Lambda_g$ near the solution of the barrier problem, and thus by (3.11) $S \Sigma S$ is close to μI. From (3.31) we see that (3.43) does become increasingly ill-conditioned as $\mu \to 0$, but this ill-conditioning does not greatly degrade the performance of the CG method since it results in one tight cluster of small eigenvalues. The numerical tests described in section 4 confirm that the solution by the CG method does not become significantly more difficult as μ tends to zero.

The CG iteration computes estimates of the minimizer of (3.40) by the recursion (see, e.g., [19])

$$u^+ = u + \alpha \delta, \tag{3.44}$$

where the parameter α is chosen to minimize the quadratic objective q along the direction δ. Since the gradient of q with respect to u is $\tilde{Z}^T \nabla q(\tilde{v} + \tilde{Z} u)$, and since our preconditioner is given by (3.42), the conjugate directions δ are recurred by

$$\delta^+ = -(\tilde{Z}^T \tilde{Z})^{-1} \tilde{Z}^T \nabla q(\tilde{v} + \tilde{Z} u) + \beta \delta, \tag{3.45}$$

where the parameter β is initially zero and is chosen at subsequent steps to maintain conjugacy.

However, because of the computational cost of manipulations with the preconditioner (3.42), it is preferable to perform the CG iteration in the full space rather than the reduced space. More specifically, by applying the transformation (3.39) to (3.44)–(3.45), we obtain the following iteration in the variable $\tilde{w}$ of problem (3.34):

$$\tilde{w}^+ = \tilde{w} + \alpha p \qquad (p \equiv \tilde{Z} \delta), \tag{3.46}$$

$$p^+ = -\tilde{Z}(\tilde{Z}^T \tilde{Z})^{-1} \tilde{Z}^T \nabla q(\tilde{v} + \tilde{w}) + \beta p. \tag{3.47}$$

We have therefore obtained a CG iteration to minimize the objective (3.34) of the tangential subproblem that, by construction, satisfies the constraints (3.35)–(3.36). Note that the matrix $\tilde{Z}(\tilde{Z}^T\tilde{Z})^{-1}\tilde{Z}^T$ is actually the orthogonal projection onto the null space of $\hat{A}^T$ and thus can be expressed as

$$P = \tilde{Z}(\tilde{Z}^T\tilde{Z})^{-1}\tilde{Z}^T = I - \hat{A}(\hat{A}^T\hat{A})^{-1}\hat{A}^T. \tag{3.48}$$

We compute projections of the form Pr by solving an augmented system whose coefficient matrix coincides with that used in the normal step and Lagrange multiplier computations, as will be discussed in section 3.4. The resulting iteration is equivalent to the preconditioned CG iteration in the null space, described above, but allows us to totally bypass the computation of the null space matrix Z. The computation of the projected residual Pr corresponds to the preconditioning step in the null space iteration.

Because of the trust region constraint (3.37), and due to the possibility of indefiniteness in the quadratic model, we use Steihaug's stopping tests in the iteration (3.46)–(3.47): we terminate if the projected gradient of q is smaller than a prescribed tolerance, if the direction p^+ is one of negative curvature, or if the iterates violate the trust region norm constraint (3.37). We include an additional step truncation to satisfy the bound constraint (3.38).

PCG PROCEDURE. PROJECTED CG METHOD FOR THE TANGENTIAL SUBPROBLEM (3.34)–(3.38).

```
Set $\tilde{w} = 0$, $r = (r_x, r_s) = (\nabla f, -\mu e) + G\tilde{v}$, $g = Pr$, $p = -g$, tol $=$
$0.01\sqrt{g^T r}$.
**Repeat** at most $2(n-t)$ times, or until a stopping test is satisfied.
    If $p^T Gp \le 0$
        then $\tilde{w}^+ = \tilde{w} + \theta p$, where $\theta > 0$ is such that $\|\tilde{w}^+\|_2 = \Delta$;  STOP
    $\alpha = r^T g / p^T Gp$
    $\tilde{w}^+ = \tilde{w} + \alpha p$
    If $\|\tilde{w}^+\|_2 > \Delta$
        then $\tilde{w}^+ = \tilde{w} + \theta p$, where $\theta > 0$ is such that $\|\tilde{w}^+\|_2 = \Delta$;  STOP
    $r^+ = r + \alpha Gp$
    $g^+ = Pr^+$
    If $(g^+)^T r^+ <$ tol,  STOP
    $\beta = (r^+)^T g^+ / r^T g$
    $p^+ = -g^+ + \beta p$
    $\tilde{w} \leftarrow \tilde{w}^+$,   $r \leftarrow r^+$,   $p \leftarrow p^+$
**End repeat**
If $\tilde{w}^+$ does not satisfy the slack variable bound (3.38), restore the
last feasible iterate $\tilde{w}$ and the direction $p$ computed at that point.
Set $\tilde{w}^+ = \tilde{w} + \theta p$, where $\theta > 0$ is is the largest value such that $\tilde{w} + \theta p$
is feasible. Set $w = (w_x, w_s) = (\tilde{w}_x^+, S\tilde{w}_s^+)$.
```

Note that during the **Repeat** loop we test only whether the trust region norm constraint (3.37) is satisfied and ignore the slack variable bound (3.38). The reason for this is that it can be shown [36] that the norm of the iterates $\|\tilde{w}\|_2$ increases during the conjugate gradient iteration, so that once an iterate violates (3.37), all subsequent iterates will also violate this constraint. It is therefore sensible to stop iterating when (3.37) is violated. However, the slack bounds (3.38) could be crossed several times, so we do not check feasibility with respect to the bound until we have gone as far as possible subject to the norm constraint. Thus, at the end of the **Repeat** loop the point $\tilde{w}^+$ may not satisfy the slack variable bounds (3.38). In this case we select the last intersection point of the path generated by the iterates $\tilde{w}$ with the bounds (3.38). This strategy has the potential of being wasteful, because we could generate a series of iterates that violate the slack variable bounds and never return to the feasible region. To control this cost we include a limit of $2(n-t)$ CG iterations in the tangential step computation. In the tests described in section 4, the infeasible CG steps accounted for about 2% of the total, and our strategy appears to pay off because, in our experience, when the iterates did return to the feasible region they usually generated a much better step than the one obtained when the bounds were first encountered.

In section 3.4 we will show how the projection Pr^+ can be computed by solving an augmented system whose coefficient matrix is the same as that needed in the normal step and Lagrange multiplier computations.

3.3. Merit function, trust region, and second-order correction. The merit function $\phi(x, s; \nu)$, defined by (2.8), is used to determine whether the total step $d = v + w$ is acceptable and also provides information on how to update the trust region radius Δ. The penalty parameter ν (not to be confused with the barrier parameter μ) balances the relative contribution of the objective function and constraints, and needs to be selected at every iteration so that the step d and the merit function ϕ are compatible. By this we mean that if the trust region is sufficiently small, then the step d must give a reduction in ϕ.

We approximate the change in the merit function due to the step d by the *predicted reduction* defined as

$$\text{pred}(d) = -q(\tilde{v} + \tilde{w}) + \nu\text{vpred}, \tag{3.49}$$

where q is the objective in the tangential subproblem (3.34) and vpred is the reduction provided by the normal step,

$$\text{vpred} = \left\| \begin{bmatrix} h \\ g+s \end{bmatrix} \right\| - \left\| \begin{bmatrix} h \\ g+s \end{bmatrix} + \hat{A}^T \tilde{v} \right\|. \tag{3.50}$$

The definition (3.49) is motivated and analyzed in [7] and is similar to the measures used in other trust region algorithms for constrained optimization. We demand that ν be large enough that pred(d) be positive and proportional to vpred, i.e.,

$$\text{pred}(d) \geq \rho\nu\text{vpred}, \tag{3.51}$$

where $0 < \rho < 1$ (in our code we use the value $\rho = 0.3$).

We see from (3.49) that we can enforce inequality (3.51) by choosing the penalty parameter ν so that

$$\nu \geq \frac{q(\tilde{v} + \tilde{w})}{(1-\rho)\text{vpred}}. \tag{3.52}$$

As has been argued in [7], if $m(\tilde{v}) = 0$, then $\tilde{v} = 0$, which implies $q(\tilde{v} + \tilde{w}) \leq 0$, and so (3.51) is satisfied for any value of ν. In this case ν can be defined as its value in the previous iteration of Algorithm II, ν^-. Thus we update ν as follows.

Penalty parameter procedure.

```
    **If** $m(\tilde{v}) = 0$ **then**
        $\nu = \nu^-$
    Else
        $\nu = \max\left\{\nu^-, \frac{q(\tilde{v}+\tilde{w})}{(1-\rho)\text{vpred}}\right\}$.
    End
```

This procedure is applied while the barrier parameter μ is fixed. Thus, for a fixed barrier problem the penalty parameter ν is monotonically increasing as the iterations progress, which is an important property for the global convergence analysis of the algorithm [7]. If the value of the barrier parameter was just changed at the beginning of the current iteration, the value of ν^- to be used in the penalty parameter procedure is reset to a default initial value.

Now that the merit function has been completely specified, let us consider how to use it to determine if a step d is to be accepted by Algorithm II. As is common in trust region methods, we compute the *actual reduction* in the merit function,

$$\text{ared}(d) = \phi(x, s; \nu) - \phi(x + d_x, s + d_s; \nu), \tag{3.53}$$

and accept d only if it gives a sufficient reduction in ϕ, in the sense that

$$\gamma \equiv \frac{\text{ared}(d)}{\text{pred}(d)} \geq \eta, \tag{3.54}$$

where $0 < \eta < 1$ (in our code we use $\eta = 10^{-8}$). Using essentially the same argument as in [7] it can be shown that (3.54) will be satisfied if the trust region radius Δ is sufficiently small.

If a step is accepted, then the trust region is increased as follows:

$$\Delta^+ = \begin{cases} \max\{7\|d\|, \Delta\} & \text{if} \quad \gamma \geq 0.9, \\ \max\{2\|d\|, \Delta\} & \text{if} \quad 0.3 \leq \gamma < 0.9, \\ \Delta & \text{if} \quad \eta \leq \gamma < 0.3. \end{cases} \tag{3.55}$$

When a step is rejected, the new trust region radius is at most one-half, but not less than one-tenth, of the length of the step. To determine the exact fraction of contraction in Δ we use linear or quadratic interpolation; the details are given in [34]. We also adjust Δ when the barrier parameter μ is reduced using the rule $\Delta \leftarrow \max(5\Delta, 1)$.

In order to achieve fast convergence, it is important that near the solution the trust region be inactive so that the algorithm can take full Newton steps. However, because of the nondifferentiability of the merit function, it can occur that a step that approaches the solution point does not satisfy (3.54) and is rejected. (This is sometimes referred to as the Maratos effect; see, e.g., [30, 11].) Since this problem is caused by an increase in the norm of the constraints due to their nonlinearity, one way to rectify the situation is to add a *second order correction* step y when (3.54) fails. (See section 14.4 in [19].) This is a Newton-like step on the constraints and amounts to computing (3.24) at the point $x + d$. In our implementation the second order correction is applied only when the normal component is small relative to the tangential component of the step.

Procedure SOC. Second order correction.

```
    If $\|\tilde{v}\| \leq 0.1\|\tilde{w}\|$ then
        $y = \hat{A}\left(\hat{A}^T\hat{A}\right)^{-1}\begin{bmatrix} h(x+d_x) \\ g(x+d_x)+s+d_s \end{bmatrix}$
    Else
        $y = 0$
    End
```

The total step of Algorithm II, when a second order correction is needed, is given by $d + y$.

3.4. Solution of linear systems. The algorithm requires the solution of three linear systems per iteration. They occur in the computation of the Lagrange multiplier estimates (3.12), in the Newton component (3.24) of the normal step, and in the projection Pr^+ required by the PCG procedure, where P is defined by (3.48). We now show that these three systems can be solved using only one matrix factorization.

Note that the normal step (3.24) requires the solution of a system of the form

$$\hat{A}^T \hat{A} x = b,$$

where $\hat{A}$ is defined by (3.13). We compute the solution by solving the *augmented system*

$$\begin{bmatrix} I & \hat{A} \\ \hat{A}^T & 0 \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} = \begin{bmatrix} 0 \\ -b \end{bmatrix}. \tag{3.56}$$

Similarly, the computation $g = Pr$, where P is expressed in terms of $\hat{A}$ (3.48), can be performed by solving

$$\begin{bmatrix} I & \hat{A} \\ \hat{A}^T & 0 \end{bmatrix} \begin{bmatrix} g \\ l \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}. \tag{3.57}$$

Moreover, if we solve the system (3.57) with r replaced by $(-\nabla f, \mu e)^T$, then, by (3.12), the vector l contains the least squares multiplier estimates.

We use routine MA27 [25] to factor the coefficient matrix in (3.56) and (3.57). We prefer working with this augmented system, rather than factoring the normal equations matrix $\hat{A}^T \hat{A}$, because our numerical experience and the analysis given by Gould, Hribar, and Nocedal [24] indicates that it is usually more accurate. Our code includes an option for detecting errors in the solution of the linear systems and applying iterative refinement, when necessary. A detailed description of this procedure is given in [24].

3.5. Full description of the new interior point method. Having gone over all the details of our approach we can now present a complete description of the new algorithm for solving the nonlinear programming problem (1.1). We will refer to this algorithm as NITRO (Nonlinear Interior point Trust Region Optimizer). There are primal and primal-dual versions of the algorithm, depending on how Σ_k, (3.1) and (3.11), and the Lagrange multipliers λ_k (3.15) are defined.

The stopping conditions for each barrier subproblem, and for the entire algorithm, are based on the function $E(x, s; \mu)$, which is defined by (2.3), where $(\lambda_h, \lambda_g) = \lambda^{\text{LS}}(x, s, \mu)$ is defined by (3.12).

```
Algorithm III. Complete NITRO algorithm.
Choose a value for the parameters $\eta > 0$, $\tau \in (0,1)$, $\theta \in (0,1)$, and
$\zeta \in (0,1)$, and select the stopping tolerances $\epsilon_\mu$ and $\epsilon_{\text{TOL}}$. Choose
an initial value for $\mu$, $x_0$, $s_0 > 0$ and $\Delta_0$. Set $k = 0$.
Repeat until $E(x_k, s_k; 0) \le \epsilon_{\text{TOL}}$:
    Repeat until $E(x_k, s_k; \mu) \le \epsilon_\mu$:
        Compute the normal step $v_k = (v_x, v_s)$ by the dogleg procedure,
        described in section 3.2.
        Compute Lagrange multipliers from (3.15).
        Compute $\nabla^2_{xx}\mathcal{L}(x_k, s_k, \lambda_h, \lambda_g)$ and $\Sigma_k$, using (3.1) or (3.14).
        Compute the tangential step $w_k$ by the PCG procedure.
        Compute the total step $d_k = v_k + w_k$.
        Update $\nu_k$ by penalty parameter procedure in section 3.3.
        Compute $\text{pred}_k(d_k)$ by (3.49) and $\text{ared}_k(d_k)$ by (3.53).
        If $\text{ared}_k(d_k) \geq \eta\text{pred}_k(d_k)$
            Then set $x_{k+1} = x_k + d_x$, $s_{k+1} = s_k + d_s$, and update $\Delta_{k+1}$
            by (3.55).
        Else perform Procedure SOC to obtain $y_k = (y_x, y_s)$.
            If $y_k \neq 0$, if $\text{ared}_k(d_k + y_k) \geq \eta\text{pred}_k(d_k)$,
              and if $s_k + d_s + y_s \geq (1 - \tau)s_k$
            then set $x_{k+1} = x_k + d_x + y_x$, $s_{k+1} = s_k + d_s + y_s$,
              and $\Delta_{k+1} = \Delta_k$.
            else set $x_{k+1} = x_k$, $s_{k+1} = s_k$, $\Delta_{k+1} \in [0.1\Delta_k, 0.5\Delta_k]$.
            Endif
        Endif
        Set $k \leftarrow k + 1$.
    End
    $\mu \leftarrow \theta\mu$ , $\epsilon_\mu \leftarrow \theta\epsilon_\mu$.
    Reset $\nu_{k-1}$ and $\Delta_k$.
End
```

In our code we assign the following values to the parameters in the algorithm: $\eta = 10^{-8}$, $\tau = 0.995$, $\theta = 0.2$, $\zeta = 0.8$, and $\epsilon_{\text{TOL}} = 10^{-7}$. We use the following initial values: $\epsilon_\mu = 0.1$, $\mu = 0.1$, $\nu_0 = 1$, and $\Delta_0 = 1$.

Byrd, Gilbert, and Nocedal [7] present a global convergence analysis for an algorithm that is very similar to the one just given. Perhaps the only significant difference is that [7] studies only the primal method, where Σ_k is given by (3.1); here we are interested also in the primal-dual formulation. We expect, however, that the results of [7] can be extended without great difficulty to the primal-dual case.

4. Numerical tests. We have tested our algorithm on a set of problems from the CUTE collection [4], whose characteristics are described in Table 1. For each problem, we give the number of variables and the total number of constraints, including equalities and general inequalities (but not bounds on the variables). We also state what kinds of conditions are imposed on the variables (fixed, free, bounds). For example, in problem CORKSCRW some variables are fixed, some are free, and some contain bounds. We also specify what kind of general constraints occur in the problem (equalities, inequalities, linear, nonlinear) and the characteristics of the objective function. The problem set has been chosen for its variety: it contains problems with negative curvature (e.g., OPTMASS), problems with ill-conditioned matrices of constraint gradients (e.g., HAGER4), problems containing only simple bounds (OBSTCLAE, TORSION1), problems with highly nonlinear equality constraints, and problems with a large number of variables and constraints. On the other hand, our test set is small enough to allow us to know each problem well and analyze each run in detail.

In Table 2 we present the results for the primal-dual version of our new algorithm, NITRO. For comparison we also solved the problems with LANCELOT [14] using second derivatives and all its default settings. The runs of NITRO were terminated when $E(x_k, s_k; 0) \leq 10^{-7}$, and LANCELOT was stopped when the projected gradient

Table 1
The main test problem set.

Problem	# of var	# of constr	Variable types	Constraint types	Objective
CORKSCRW	456	350	free, bounded, fixed	linear eq, nonlin ineq	nonlinear
COSHFUN	61	20	free	nonlin ineq	linear
DIXCHLNV	100	50	bounded	nonlin eq	nonlinear
GAUSSELM	14	11	free, bounded, fixed	linear ineq, nonlin eq	linear
HAGER4	2001	1000	free, bounded, fixed	linear eq	nonlinear
HIMMELBK	24	14	bounded	linear eq, nonlin eq	linear
NGONE	100	1273	bounded, fixed	linear ineq, nonlin ineq	nonlinear
OBSTCLAE	1024	0	bounded, fixed		nonlinear
OPTCNTRL	32	20	free, bounded, fixed	linear eq, nonlin eq	nonlinear
OPTMASS	1210	1005	free, fixed	linear eq, nonlin ineq	nonlinear
ORTHREGF	1205	400	free, bounded	nonlin eq	nonlinear
READING1	202	100	bounded, fixed	nonlin eq	nonlinear
SVANBERG	500	500	bounded	nonlin ineq	nonlinear
TORSION1	484	0	bounded, fixed		nonlinear

Table 2
Number of function evaluations, number of CG iterations, and CPU time for the new primal-dual interior point method (NITRO) and LANCELOT (LAN). An asterisk (*) indicates that the method did not meet the stopping test in 10,000 *iterations.*

	# of	# of	f evals		CG iters		Time	
Problem	var	constr	NITRO	LAN	NITRO	LAN	NITRO	LAN
CORKSCRW	456	350	61	171	430	114780	53.78	657.94
COSHFUN	61	20	40	149	1316	3421	4.22	5.83
DIXCHLNV	100	50	19	1445	83	1431	14.46	153.97
GAUSSELM	14	11	52	28	115	112	0.79	0.25
HAGER4	2001	1000	18	14	281	2291	37.34	99.65
HIMMELBK	24	14	33	154	89	1533	4.15	8.18
NGONE	100	1273	256	3997	1821	129963	1027.51	1446.09
OBSTCLAE	1024	0	26	5	6184	366	566.39	12.98
OPTCNTRL	32	20	47	25	165	65	1.44	0.3
OPTMASS	1210	1005	39	*	151	*	24.79	*
ORTHREGF	1205	400	30	192	78	315	57.09	48.18
READING1	202	100	40	720	130	13981	130.89	74.13
SVANBERG	500	500	35	82	5067	3908	2720.19	120.96
TORSION1	484	0	19	8	2174	66	58.39	1.11

and constraint violations were less than 10^{-7}; the termination criteria for these two methods are therefore very similar. In all these problems the two codes approached the same solution point. Since both algorithms use the conjugate gradient method to compute the step, we also report in Table 2 the total number of CG iterations needed for convergence. All runs were performed on a SPARCstation 20 with 32 MB of main memory, using a Fortran-77 compiler and double precision; the CPU time reported is in seconds. An asterisk indicates that the stopping test was not satisfied after 10,000 iterations. The results of NITRO reported in Table 2 are highly encouraging, particularly the number of function evaluations.

In Table 3 we compare the primal version of NITRO using (3.1) and the primal-dual version using (3.11). The column under the header "%full steps" denotes the percentage of steps that did not encounter the trust region (3.18). We see that the primal-dual version (pd) outperforms the primal version (p), and its step tends to be constrained by the trust region less often.

To observe whether the tangential subproblem becomes very difficult to solve

Table 3

Primal dual vs. primal options of the new interior point method. The number of function evaluations and percentage of full steps are given. An asterisk (*) indicates that the stopping test was not satisfied in 10,000 *iterations.*

	# of	# of	NITRO (pd)		NITRO (p)	
Problem	var	constr	f evals	%full steps	f evals	%full steps
CORKSCRW	456	350	61	40	78	58
COSHFUN	61	20	40	83	472	6
DIXCHLNV	100	50	19	79	18	78
GAUSSELM	14	11	52	27	62	27
HAGER4	2001	1000	18	78	21	62
HIMMELBK	24	14	33	79	62	36
NGONE	100	1273	256	6	200	18
OBSTCLAE	1024	0	26	77	60	82
OPTCNTRL	32	20	47	92	51	73
OPTMASS	1210	1005	39	59	67	60
ORTHREGF	1205	400	30	30	31	30
READING1	202	100	40	78	33	33
SVANBERG	500	500	35	71	61	72
TORSION1	484	0	19	79	41	78

Table 4

Analysis of the last step computed by NITRO. Total number of CG iterations divided by the dimension of the linear system, $n-t$, and the type of step taken.

	# of	# of	NITRO (pd)	
Problem	var	constr	CG iter	Step type
CORKSCRW	456	350	0.03	full
COSHFUN	61	20	2.0	CG limit
DIXCHLNV	100	50	0.1	full
GAUSSELM	14	11	0.4	full
HAGER4	2001	1000	0.1	full
HIMMELBK	24	14	0.3	full
NGONE	100	1273	0.08	hit tr
OBSTCLAE	1024	0	2.0	CG limit
OPTCNTRL	32	20	0.3	full
OPTMASS	1210	1005	0.0	full
ORTHREGF	1205	400	0.006	full
READING1	202	100	0.03	full
SVANBERG	500	500	1.5	full
TORSION1	484	0	1.8	full

as the barrier parameter approaches zero, we report in Table 4 the number of CG iterations required in the step computation during the *last* iteration of the interior point algorithm. At this stage the barrier parameter μ is of order 10^{-7}. Table 4 gives the number of CG iterations relative to the dimension $n-t$ of the linear system to be solved. (Recall that the code imposes a limit of 2 on this ratio.) We also report if the step was inside the trust region (full), if it encountered the trust region (hit tr), or if the number of CG iterations reached the permissible limit of $2(n-t)$. These results, as well as an examination of the complete runs, indicate that the subproblems do not become particularly hard to solve as the problem approaches the solution. This is due to the preconditioning described before the statement of PCG procedure.

To test the robustness of the new interior point method, we chose some of the commonly used problems from the Hock and Schittkowski collection [26], as programmed in CUTE. The results are given in Table 5 and include all the problems that we tested. Since these problems contain a very small number of variables, we do

Table 5

The number of function evaluations for NITRO and LANCELOT to solve a subset of the Hock and Schittkowski test collection. An asterisk (*) indicates a failure to obtain a solution within 10,000 *iterations. A double asterisk indicates that LANCELOT computed a point that was not a local minimum.*

Problem	NITRO	LAN
HS2	18	7
HS3	12	5
HS4	11	2
HS7	8	18
HS10	17	19
HS11	14	19
HS13	40	81
HS14	14	13
HS16	15	16
HS17	27	20
HS19	47	36
HS20	18	23
HS22	15	11
HS24	19	8
HS26	16	39
HS28	3	4
HS31	13	13
HS32	19	9
HS33	28	12
HS39	19	21
HS46	16	29
HS51	3	3
HS52	3	8
HS53	8	8
HS63	13	14
HS64	43	53
HS65	20	28
HS70	35	29
HS71	16	16
HS72	44	94
HS73	29	18
HS74	15	28
HS75	107	141**
HS77	17	22
HS78	5	12
HS79	6	10
HS80	13	15
HS81	13	17
HS83	36	26
HS84	20	60
HS85	1658	17
HS86	16	18
HS93	14	6
HS95	156	8
HS96	196	8
HS97	45	19
HS98	53	18
HS99	*	70
HS100	20	46
HS104	34	62
HS105	34	15
HS106	221	*
HS107	15	40
HS108	49	24
HS109	*	*
HS111	15	47
HS112	14	44
HS113	17	81
HS114	33	763
HS116	71	*
HS117	40	50
HS118	28	17
HS119	31	29

not report CPU time. NITRO failed for problems HS99 and HS109. In problem HS99, the code terminated very close to a solution because the trust region was too small. In problem HS109, the routine MA27 failed to factor the augmented systems in (3.56) and (3.57) because they were determined to be very close to singular. LANCELOT failed for four problems. In HS75, the code completed without reporting any errors. However, the point that was returned failed to satisfy the stopping test. In problems HS106, HS109, and HS116, LANCELOT was unable to compute a solution in 10,000 iterations.

It is reassuring to observe that NITRO failed on very few problems. Nevertheless, its performance is not as good as that of LANCELOT on these small problems, and it appears that our strategy for decreasing the barrier parameter is overly conservative. We suspect that by decreasing it more rapidly, and in a carefully controlled manner [8], the number of function evaluations will be reduced significantly. We should also mention that we do not yet have a complete understanding of the behavior of NITRO on some of the problems on which it took a large number of iterations.

5. Final remarks.

We have presented an interior point method for solving large nonlinear programming problems. Rather than trying to mimic primal-dual interior point methods for linear programming, we have taken the approach of developing a fairly standard SQP trust region method and introduced in it some of the key features of primal-dual iterations. No attempt was made to obtain a rapidly convergent method: the barrier parameter was decreased at a linear rate, forcing the iterates of the algorithm to converge linearly. We have, however, given careful attention to the treatment of nonconvexity and to the exploitation of sparsity through the use of the conjugate gradient method and the sparse Cholesky code MA28, and we have designed many features to make the algorithm robust on general problems. This approach appears to have paid off in that the algorithm has proved to be capable of solving a wide range of problems, even when ill-conditioning and nonconvexity are present. Our tests seem to indicate that our code is competitive on large problems with a production code such as LANCELOT. We have also shown that the preconditioning of the tangential subproblem has, to a large extent, removed the effects of the ill-conditioning inherent in interior point methods and that the CG iteration does not have particular difficulties in computing the tangential component of the step as the iterates approach the solution.

The algorithm presented here is not as rapidly convergent as it can be. We are currently developing [8] various mechanisms to accelerate the iteration; these include the use of higher order corrections and rules for decreasing the barrier parameter at a superlinear rate. We should also note that the technique for refining the solution of linear systems referred to at the end of section 3.4 is very conservative (in that it demands very tight accuracy) and leads to high execution times on some problems. More efficient techniques for refining the solution of linear systems are the subject of current investigation [24].

Acknowledgment. We would like to thank Guanghui Liu for help in the preparation of this article.

REFERENCES

[1] K. M. Anstreicher and J.-P. Vial, *On the convergence of an infeasible primal-dual interior-point method for convex programming*, Optim. Methods Softw., 3 (1994), pp. 273–283.

[2] M. Argaez, *Exact and Inexact Newton Linesearch Interior-Point Algorithms for Nonlinear Programming Problems*, Technical Report TR97-13, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1997.

[3] P. T. Boggs and J. W. Tolle, *Sequential quadratic programming*, Acta Numer., 4 (1996), pp. 1–51.

[4] I. Bongartz, N. I. M. Gould, A. R. Conn, and Ph. L. Toint, *CUTE: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[5] J. F. Bonnans and C. Pola, *A trust region interior point algorithm for linearly constrained optimization*, SIAM J. Optim., 7 (1997), pp. 717–731.

[6] R. H. Byrd, *Robust trust region methods for constrained optimization*, talk presented at the Third SIAM Conference on Optimization, Houston, TX, 1987.

[7] R. H. Byrd, J. C. Gilbert, and J. Nocedal, *A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming*, Technical Report OTC 96/02, Optimization Technology Center, Northwestern University, Evanston, IL, 1996.

[8] R. H. Byrd, G. Liu, and J. Nocedal, *On the local behavior of an interior-point algorithm for nonlinear programming*, in Numerical Analysis 1997, D.F. Griffiths and D.J. Higham, eds., Addison–Wesley Longman, Reading, MA, 1997.

[9] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *A trust region algorithm for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 24 (1987), pp. 1152–1170.

[10] M. R. Celis, J. E. Dennis, and R. A. Tapia, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization 1984, Proceedings SIAM Conference on Numerical Optimization, Boulder, CO, June 12–14, 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, PA, 1985, pp. 71–82.

[11] R. Chamberlain, C. Lemarechal, H. C. Pedersen, and M. J. D. Powell, *The watchdog technique for forcing convergence in algorithms for constrained optimization*, Math. Programming, 16 (1982), pp. 1–17.

[12] T. F. Coleman and Y. Li, *On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds*, Math. Programming, 67 (1994), pp. 189–224.

[13] T. F. Coleman and Y. Li, *An interior trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optim., 6 (1996), pp. 418–445.

[14] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *LANCELOT: A FORTRAN Package for Large-Scale Nonlinear Optimization (Release A)*, Springer Ser. Comput. Math. 17, Springer, New York, 1992.

[15] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *A Primal-Dual Algorithm for Minimizing a Non-Convex Function Subject to Bound and Linear Equality Constraints*, Technical Report RC 20639, IBM T.J. Watson Research Center, Yorktown Heights, NY, 1997.

[16] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *A note on using alternative second-order models for the subproblems arising in barrier function methods for minimization*, Numer. Math., 68 (1994), pp. 17–33.

[17] J. E. Dennis, M. Heinkenschloss, and L. N. Vicente, *Trust-region interior-point SQP algorithms for a class of nonlinear programming problems*, SIAM J. Control Optim., 36 (1998), pp. 1750–1794.

[18] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang, *On the formulation and theory of the Newton interior-point method for nonlinear programming*, J. Optim. Theory Appl., 89 (1996), pp. 507–545.

[19] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley, New York, 1990.

[20] A. Forsgren and P. E. Gill, *Primal-dual interior methods for nonconvex nonlinear programming*, SIAM J. Optim., 8 (1998), pp. 1132–1152.

[21] D. M. Gay, M. L. Overton, and M. H. Wright, *A primal-dual interior method for nonconvex nonlinear programming*, in Advances in Nonlinear Programming, Y. Yuan, ed., Kluwer Academic Publishers, Dordrecht, Netherlands, 1998, pp. 31–56.

[22] P. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, New York, 1981.

[23] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization*, Technical Report NA 97-2, Department of Mathematics, University of California, San Diego, 1997; SIAM J. Optim., submitted.

[24] N. I. M. Gould, M. E. Hribar, and J. Nocedal, *On the Solution of Equality Constrained Quadratic Programming Problems Arising in Optimization*, Technical Report OTC 98/06, Optimization Technology Center, Northwestern University, Evanston, IL, 1998; SIAM J. Sci. Comput., submitted.

[25] Harwell Subroutine Library, *A Catalogue of Subroutines (Release* 12*)*, AEA Technology, Harwell, Oxfordshire, England, 1995.

[26] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Econom. and Math. Systems 187, Springer, New York, 1981.

[27] M. E. Hribar, *Large-Scale Constrained Optimization*, Ph.D. thesis, EECS Department, Northwestern University, Evanston, IL, 1996.

[28] F. Jarre and S. J. Wright, *On the Role of the Objective Function in Barrier Methods*, Technical Report MCS-P485-1294, MCS Division, Argonne National Laboratory, Argonne, IL, 1994.

[29] M. Lalee, J. Nocedal, and T. Plantenga, *On the implementation of an algorithm for large-scale equality constrained optimization*, SIAM J. Optim., 8 (1998), pp. 682–706.

[30] N. Maratos, *Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization Problems*, Ph.D. thesis, University of London, 1978.

[31] R. D. C. Monteiro and Y. Wang, *Trust region affine scaling algorithms for linearly constrained convex and concave programs*, Math. Programming, 80 (1998), pp. 283–313.

[32] E. Omojokun, *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*, Ph.D. thesis, University of Colorado, Boulder, CO, 1989.

[33] Z. Parada, *A Modified Augmented Lagrangian Merit Function and q-Superlinear Characterization Results for Primal-Dual Quasi-Newton Interior-Point Methods for Nonlinear Programming*, Technical Report TR97-12, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1997.

[34] T. Plantenga, *A trust region method for nonlinear programming based on primal interior-point techniques*, SIAM J. Sci. Comput., 20 (1998), pp. 282–305.

[35] M. J. D. Powell, *A hybrid method for nonlinear equations*, in Numerical Methods for Nonlinear Algebraic Equations, P. Rabinowitz, ed., Gordon & Breach, London, 1970, pp. 87–114.

[36] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[37] R. J. Vanderbei, *Linear Programming*, Kluwer, Dordrecht, the Netherlands, 1996.

[38] A. Vardi, *A trust region algorithm for equality constrained minimization: Convergence properties and implementation*, Math. Programming, 22 (1985), pp. 575–591.

[39] M. H. Wright, *Why a pure primal Newton barrier step may be infeasible*, SIAM J. Optim., 5 (1995), pp. 1–12.

[40] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, PA, 1997.

[41] H. Yamashita, *A Globally Convergent Primal-Dual Interior-Point Method for Constrained Optimization*, Technical Report, Mathematical Systems Institute, Inc., Tokyo, Japan, 1994.

[42] H. Yamashita and H. Yabe, *Superlinear and quadratic convergence of some primal-dual interior point methods for constrained optimization*, Math. Programming, 75 (1996), pp. 377–397.

SIAM J. OPTIM.
Vol. 9, No. 4, pp. 901–923

NONLINEAR OPTIMIZATION, QUADRATURE, AND INTERPOLATION*

H. CHENG[†], V. ROKHLIN[†], AND N. YARVIN[†]

To John Dennis on the occasion of his 60th birthday.

Abstract. We present a nonlinear optimization procedure for the design of generalized Gaussian quadratures for a fairly broad class of functions. While some of the components of the algorithm have been published previously, we introduce an improved procedure for the determination of an acceptable initial point for the continuation scheme that stabilizes the Newton-type process used to find the quadratures. The resulting procedure never failed when applied to Chebyshev systems (for which the existence and uniqueness of generalized Gaussian quadratures are well known); it also worked for many non-Chebyshev systems, for which the generalized Gaussian quadratures are not guaranteed to exist. The performance of the algorithm is illustrated with several numerical examples; some of the presented quadratures integrate efficiently large classes of singular functions.

Key words. nonlinear optimization, quadratures, singular integrands, interpolation

AMS subject classifications. 49N99, 65D30, 65D32, 65D05

PII. S1052623498349796

1. Introduction.

Quadrature formulae constitute one of the most developed areas of computational mathematics. They are used both as a stand-alone numerical tool for the evaluation of integrals and as an analytical apparatus for the design of interpolation schemes, finite element schemes, etc. Most of the quadrature formulae (at least for functions on $\mathbb{R}^1$) currently in use can be separated into three groups:

1. Gaussian quadratures are the optimal tool for the evaluation of integrals of the form

$$\int_a^b \omega(t) \cdot P(t)dt, \tag{1.1}$$

where P is a polynomial of t (or a function well approximated by a polynomial) and ω is a (more or less) arbitrary nonnegative function $[a,b] \to \mathbb{R}$. Gaussian quadratures are extremely efficient, mathematically elegant, and easy to obtain (see, for example, [3]); whenever applicable, they tend to be the numerical tool of choice.

2. Interpolatory quadrature formulae (Newton–Cotes, etc.) are based on approximating the integrand by some standard function (usually a polynomial) and integrating the latter. These schemes have the advantage that they (usually) do not prescribe the locations of the nodes; they tend to become numerically unstable for high orders.

3. Miscellaneous special-purpose quadratures ("product integration rules," nonstandard Richardson extrapolation, etc.) are normally used when the situation precludes the use of more straightforward techniques.

There appears to exist a class of situations where classical approaches fail to produce rapidly convergent schemes. Specifically, suppose that we wish to integrate

[*] Received by the editors December 18, 1998; accepted for publication (in revised form) January 27, 1999; published electronically September 24, 1999. This work was supported in part by DARPA/AFOSR under contract F49620/97/1/0011, ONR under grant N00014-96-1-0188, and AFOSR under contracts F49620-97-C-0052 and F49620-95-C-0075.

http://www.siam.org/journals/siopt/9-4/34979.html

[†] Department of Computer Science, Yale University, New Haven, CT 06520 (cheng-hongwei@cs.yale.edu, rokhlin-vladimir@cs.yale.edu, yarvin-norman@cs.yale.edu).

functions of the form

$$\sum_{k=0}^{n} \phi_k(x) \cdot s_k(x), \tag{1.2}$$

where ϕ_k are smooth functions (or polynomials) mapping $[0,1] \to \mathbb{R}$, and the functions $s_k : [0,1] \to \mathbb{R}$ are known a priori and have singularities at $x = 0$. In many situations of interest, the functions s_k have *different* singularities at $x = 0$, and the functions ϕ_k *are not known*; it is known only that the integrand has the form (1.2) and its values at points on the interval $[0,1]$ can be evaluated. While efficient quadratures for functions of the form (1.2) would have obvious applications in the solution of integral equations, in numerical complex analysis, and in several other areas, the authors have failed to find such an apparatus in the literature.

It has been known for about 100 years that Gaussian quadratures admit a drastic generalization, replacing polynomials with fairly general systems of functions (see [11, 12, 2, 8, 6, 7]). The constructions found in [11, 12, 2, 8, 6, 7] do not easily yield numerical algorithms for the design of such quadrature formulae; algorithms of this type were designed (in some cases) in [10, 15], where the resulting quadrature rules are referred to as generalized Gaussian quadratures. The approach is based on the observations that the nodes and weights of Gaussian quadratures satisfy systems of nonlinear equations, that these equations have unique solutions, and that when polynomials are replaced with other systems of functions, similar systems of equations are easily constructed. While for functions of the form (1.2) the resulting equations are nonlinear, overdetermined, and nonunique, in the least squares sense they have unique solutions under surprisingly general conditions (see [10, 15]); Newton-type methods converge in this environment, provided a good initial approximation can be found.

As often happens, in the absence of a good initial approximation the Newton process fails to converge. To some extent, this problem is remedied by the use of continuation techniques, which turn out to be almost always available when designing quadratures for integrands (1.2). However, yet another problem is frequently encountered: although *mathematically* the solution of the nonlinear problem is unique for all values of the continuation parameter, *numerically* it is not unique at all. Once the (numerical) rank of the Jacobian of an intermediate problem is sufficiently low, the continuation process breaks down; attempts to use globalized search techniques have not been successful.

The final step in the design of a robust scheme for the construction of generalized Gaussian quadratures is described in section 3.3. It finds an initial approximation for which the Jacobian of the system being solved has an acceptably low condition number. While the reasoning behind this step is partly heuristic, in our experience it works remarkably well. It never failed for a Chebyshev system (see section 2.1); furthermore, it worked for most of the non-Chebyshev systems we tried it on. For a more detailed discussion of our numerical experience, see section 5, where we also present quadratures for functions with *almost* general power singularities at one end (or both ends) of the interval of integration and for functions with several other types of singularities.

The paper is structured as follows. Section 2 contains mathematical and numerical preliminaries. In section 3 we build the numerical apparatus to be used in section 4 to construct the procedure for the determination of nodes and weights of generalized Gaussian quadratures. Section 5 contains several examples of quadratures we have obtained. Finally, in section 6 we outline several possible extensions of this work.

2. Mathematical and numerical preliminaries.

2.1. Chebyshev systems.

DEFINITION 2.1. *A sequence of functions $\phi_1, \ldots, \phi_n$ will be referred to as a* Chebyshev system *on the interval $[a, b]$ if each of them is continuous and the determinant*

$$\begin{vmatrix} \phi_1(x_1) & \cdots & \phi_1(x_n) \\ \vdots & & \vdots \\ \phi_n(x_1) & \cdots & \phi_n(x_n) \end{vmatrix} \tag{2.1}$$

is nonzero for any sequence of points $x_1, \ldots, x_n$ such that $a \le x_1 < x_2 < \cdots < x_n \le b$. An alternate definition of a Chebyshev system is that any linear combination of the functions with nonzero coefficients must have no more than n zeros.

A related definition is that of an extended Chebyshev system.

DEFINITION 2.2. *Given a set of functions $\phi_1, \ldots, \phi_n$ which are continuously differentiable on an interval $[a, b]$, and given a sequence of points $x_1, \ldots, x_n$ such that $a \le x_1 \le x_2 \le \cdots \le x_n \le b$, let the sequence $m_1, \ldots, m_n$ be defined by the formulae*

(2.2)
$$\begin{aligned} m_1 &= 0, & & \\ m_j &= 0 & & \text{if } j > 1 \text{ and } x_j \ne x_{j-1}, \\ m_j &= j - 1 & & \text{if } j > 1 \text{ and } x_j = x_{j-1} = \cdots = x_1, \\ m_j &= k & & \text{if } j > k + 1 \text{ and } x_j = x_{j-1} = \cdots = x_{j-k} \ne x_{j-k-1}. \end{aligned}$$

Let the matrix $C(x_1, \ldots, x_n) = [c_{ij}]$ be defined by the formula

$$c_{ij} = \frac{d^{m_j}\phi_i}{dx^{m_j}}(x_j), \tag{2.3}$$

in which $\frac{d^0\phi_i}{dx^0}(x_j)$ is taken to be the function value $\phi_i(x_j)$. Then $\phi_1, \ldots, \phi_n$ will be referred to as an extended Chebyshev system *on $[a, b]$ if the determinant $|C(x_1, \ldots, x_n)|$ is nonzero for all such sequences x_i.*

Remark 2.1. It is obvious from Definition 2.2 that an extended Chebyshev system is a special case of the Chebyshev system. The additional constraint is that the successive points x_i at which the function is sampled to form the matrix may be identical; in that case, for each duplicated point, the first corresponding column contains the function values, the second column contains the first derivatives of the functions, the third column contains the second derivatives of the functions, and so forth; this matrix also must be nonsingular.

Examples of Chebyshev and extended Chebyshev systems include the following (additional examples can be found in [7]).

Example 2.1. The powers $1, x, x^2, \ldots, x^n$ form an extended Chebyshev system on the interval $(-\infty, \infty)$.

Example 2.2. The exponentials $e^{-\lambda_1 x}, e^{-\lambda_2 x}, \ldots, e^{-\lambda_n x}$ form an extended Chebyshev system for any distinct $\lambda_1, \ldots, \lambda_n > 0$ on the interval $[0, \infty)$.

Example 2.3. The functions $1, \cos x, \sin x, \cos 2x, \sin 2x, \ldots, \cos nx, \sin nx$ form a Chebyshev system on the interval $[0, 2\pi)$.

2.2. Generalized Gaussian quadratures. The quadrature rules considered in this paper are expressions of the form

$$\sum_{j=1}^{n} w_j \cdot \phi(x_j), \tag{2.4}$$

where the points $x_j \in \mathbb{R}$ and coefficients $w_j \in \mathbb{R}$ are referred to as the nodes and weights of the quadrature, respectively. They serve as approximations to integrals of the form

$$\int_a^b \phi(x) \cdot \omega(x) dx, \tag{2.5}$$

where ω has the form

$$\omega(x) = \tilde{\omega}(x) + \sum_{j=1}^{m} \mu_j \cdot \delta(x - \chi_j) \tag{2.6}$$

with m a nonnegative integer, $\tilde{\omega} : [a, b] \to \mathbb{R}$ an integrable nonnegative function, $\chi_1, \chi_2, \ldots, \chi_m$ points on the interval $[a, b]$, $\mu_1, \mu_2, \ldots, \mu_m$ positive real coefficients, and δ the Dirac δ-function on $\mathbb{R}$.

Remark 2.2. Obviously, (2.6) defines ω to be a linear combination of a nonnegative function with a finite collection of δ-functions with positive coefficients. In a mild abuse of terminology, throughout this paper we will be referring to ω as a nonnegative function.

Quadratures are typically chosen so that the quadrature (2.4) is exact for some set of functions, commonly polynomials of a fixed order. Of these, the classical Gaussian quadrature rules consist of n nodes and integrate polynomials of order $2n-1$ exactly. In [10], the notion of a Gaussian quadrature was generalized as follows.

DEFINITION 2.3. *A quadrature formula will be referred to as* Gaussian *with respect to a set of* $2n$ *functions* $\phi_1, \ldots, \phi_{2n} : [a, b] \to \mathbb{R}$ *and a weight function* $\omega : [a, b] \to \mathbb{R}^+$ *if it consists of* n *weights and nodes and integrates the functions* ϕ_i *exactly with the weight function* ω *for all* $i = 1, \ldots, 2n$. *The weights and nodes of a Gaussian quadrature will be referred to as Gaussian weights and nodes, respectively.*

The following theorem appears to be due to Markov [11, 12]; proofs of it can also be found in [8] and [7] (in a somewhat different form).

THEOREM 2.4. *Suppose that the functions* $\phi_1, \ldots, \phi_{2n} : [a, b] \to \mathbb{R}$ *form a Chebyshev system on* $[a, b]$. *Suppose in addition that* $\omega : [a, b] \to \mathbb{R}$ *is defined by* (2.6) *and that either*

$$\int_a^b \tilde{\omega}(x) dx > 0 \tag{2.7}$$

or $m \geq n$ *(or both). Then there exists a unique Gaussian quadrature for* $\phi_1, \ldots, \phi_{2n}$ *on* $[a, b]$ *with respect to the weight function* ω. *The weights of this quadrature are positive.*

2.3. Quadrature and interpolation. As is well known, when Gaussian nodes on the interval $[-1, 1]$ are used for interpolation (for example, via the Lagrange formula), the resulting procedure is numerically stable. Furthermore, the precision obtained via Gaussian (Lagrange) interpolation is almost as high as that obtained via

Chebyshev interpolation (see, for example, [4]). Generally, given a weight function ω, the nodes of Gaussian quadratures corresponding to ω lead to interpolation formulae that are stable in an appropriately chosen norm. In this subsection, we formalize this fact for both Gaussian and many generalized Gaussian quadratures. The analytical tool of this subsection is the following obvious theorem.

THEOREM 2.5. *Suppose that the function $\omega : [a,b] \to \mathbb{R}$ is nonnegative and the functions $\phi_1, \phi_2, \ldots, \phi_n : [a,b] \to \mathbb{R}$ are orthonormal with respect to the weight function ω, i.e.,*

$$\int_a^b \omega(x) \cdot \phi_j(x) \cdot \phi_i(x) dx = \delta_{ij} \tag{2.8}$$

for all $i, j = 1, 2, \ldots, n$. (δ_{ij} denotes Kroneker's δ-function.) Suppose further that the n-point quadrature rule $x_1, x_2, \ldots, x_n$, $w_1, w_2, \ldots, w_n$ is such that $w_i > 0$ for all $1 \le i \le n$. Finally, suppose that

$$\sum_{k=1}^{n} w_k \cdot \phi_i(x_k) \cdot \phi_j(x_k) = \delta_{ij} \tag{2.9}$$

for all $i, j = 1, 2, \ldots, n$. Then the $n \times n$-matrix A defined by the formula

$$A_{ij} = \sqrt{w}_j \cdot \phi_i(x_j) \tag{2.10}$$

is orthogonal.

Suppose now that we would like to construct an interpolation formula on the interval $[a,b]$ for functions of the form

$$f(x) = \sum_{i=1}^{n} \alpha_i \cdot \phi_i(x) \tag{2.11}$$

with $\alpha_1, \alpha_2, \ldots, \alpha_n$ arbitrary real coefficients. In other words, suppose that we are given the values $f_1, f_2, \ldots, f_n$ of a function f at a collection of points $x_1, x_2, \ldots, x_n$ and that it is known that f is defined by the formula (2.11), but the coefficients $\alpha_1, \alpha_2, \ldots, \alpha_n$ are not known; we would like to be able to evaluate f at arbitrary points on $[a,b]$. The obvious way to do so is to observe that the values $f_1, f_2, \ldots, f_n$ are linear functions of the coefficients $\alpha_1, \alpha_2, \ldots, \alpha_n$ (due to (2.11)); evaluating (2.11) at the points $x_1, x_2, \ldots, x_n$, we obtain the system of equations

$$f_j = \sum_{i=1}^{n} \alpha_i \cdot \phi_i(x_j), \tag{2.12}$$

with $j = 1, 2, \ldots, n$. Defining the $n \times n$-matrix B by the formula

$$b_{j,i} = \phi_i(x_j), \tag{2.13}$$

we rewrite (2.12) in the form

$$F = B\alpha, \tag{2.14}$$

with the vectors $\alpha, F \in \mathbb{R}^n$ defined by the formulae

$$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n), \tag{2.15}$$

$$F = (f_1, f_2, \dots, f_n). \tag{2.16}$$

Now, as long as the matrix B is nonsingular, we can evaluate the coefficients $\alpha_1, \alpha_2, \dots, \alpha_n$ via the formula

$$\alpha = B^{-1}F \tag{2.17}$$

and use (2.11) to evaluate f at arbitrary points on $[a, b]$. Of course, in actual numerical calculations, it is not sufficient for B to be invertible; its condition number must not be too high. The following observation is the principal purpose of this subsection.

Observation 2.1. Under the conditions of Theorem 2.5,

$$A = D \circ B \tag{2.18}$$

with D the diagonal matrix defined by the formula

$$D_{i,i} = \sqrt{w_i} \tag{2.19}$$

and

$$\alpha = A^* D F \tag{2.20}$$

(due to the combination of (2.17) with (2.18)). In other words, given the table of values $f_1, f_2, \dots, f_n$ of the function f at the nodes $x_1, x_2, \dots, x_n$, one obtains the coefficients of the expansion (2.11) by applying to the vector F the product of two matrices; the first of these matrices is orthogonal and the second is diagonal; the diagonal elements of the latter are square roots of (positive) weights of the n-point quadrature formula exact for all pairwise products of the functions $\phi_1, \phi_2, \dots, \phi_n$.

Remark 2.3. While at first glance the above observation appears to be very limited in its scope (since it relies on the quadrature formula being *exact* for all pairwise products of the functions $\phi_1, \phi_2, \dots, \phi_n$), in reality it means that whenever the nodes of a generalized Gaussian quadrature formula are used as interpolation nodes, the resulting interpolation formula tends to be stable. The reason for this happy coincidence is the fact that the matrix A (see (2.10) above) need not be orthogonal for the stability of the interpolation formula; it needs only to be well conditioned. Thus, as long as the quadrature formula is reasonably accurate for all pairwise products of the functions $\phi_1, \phi_2, \dots, \phi_n$, the matrix A is close to being orthogonal; therefore, the condition number of A is close to unity, and the interpolation based on the nodes $x_1, x_2, \dots, x_n$ is stable.

2.4. Convergence of Newton's method.

In this section, we observe that the nodes and the weights of a Gaussian quadrature satisfy a simple system of nonlinear equations. We then prove that the Newton method for this system of equations is always quadratically convergent, provided the functions to be integrated constitute an extended Chebyshev system.

Given a set of functions $\phi_1, \dots, \phi_{2n}$ and a weight function ω, the Gaussian quadrature is defined by the system of equations

$$\sum_{j=1}^{n} w_j \cdot \phi_1(x_j) = \int_a^b \phi_1(x) \cdot \omega(x) dx,$$
$$\sum_{j=1}^{n} w_j \cdot \phi_2(x_j) = \int_a^b \phi_2(x) \cdot \omega(x) dx,$$

$$\vdots$$

$$\sum_{j=1}^{n} w_j \cdot \phi_{2n}(x_j) = \int_a^b \phi_{2n}(x) \cdot \omega(x)dx, \tag{2.21}$$

(see Definition 2.3). Let the left-hand sides of these equations be denoted by f_1 through f_{2n}. Then each f_i is a function of the weights $w_1, \dots, w_n$ and nodes $x_1, \dots, x_n$ of the quadrature. Its partial derivatives are given by the obvious formulae

$$\frac{\partial f_k}{\partial w_i} = \phi_k(x_i), \tag{2.22}$$

$$\frac{\partial f_k}{\partial x_i} = w_i \cdot \phi_k'(x_i). \tag{2.23}$$

Thus, the Jacobian matrix of the system (2.21) is

$$J(x_1, \dots, x_n, w_1, \dots, w_n) = \begin{pmatrix} \phi_1(x_1) & \cdots & \phi_1(x_n) & w_1\phi_1'(x_1) & \cdots & w_n\phi_1'(x_n) \\ \vdots & & \vdots & \vdots & & \vdots \\ \phi_{2n}(x_1) & \cdots & \phi_{2n}(x_n) & w_1\phi_{2n}'(x_1) & \cdots & w_n\phi_{2n}'(x_n) \end{pmatrix}. \tag{2.24}$$

LEMMA 2.6. *Suppose that the functions $\phi_1, \dots, \phi_{2n}$ form an extended Chebyshev system. Let the Gaussian quadrature for these functions be denoted by $\hat{w}_i$ and $\hat{x}_i$. Then the determinant of J is nonzero at the point which constitutes the Gaussian quadrature; in other words, $|J(\hat{x}_1, \dots, \hat{x}_n, \hat{w}_1, \dots, \hat{w}_n)| \neq 0$.*

Proof. It is immediately obvious from (2.24) that

$$\begin{aligned} &|J(\hat{x}_1, \dots, \hat{x}_n, \hat{w}_1, \dots, \hat{w}_n)| \\ &= \hat{w}_1 \cdot \hat{w}_2 \cdot \ \cdots \ \cdot \hat{w}_{n-1} \cdot \hat{w}_n \cdot \begin{vmatrix} \phi_1(\hat{x}_1) & \cdots & \phi_1(\hat{x}_n) & \phi_1'(\hat{x}_1) & \cdots & \phi_1'(\hat{x}_n) \\ \vdots & & \vdots & \vdots & & \vdots \\ \phi_{2n}(\hat{x}_1) & \cdots & \phi_{2n}(\hat{x}_n) & \phi_{2n}'(\hat{x}_1) & \cdots & \phi_{2n}'(\hat{x}_n) \end{vmatrix}. \end{aligned} \tag{2.25}$$

If $\phi_1, \dots, \phi_{2n}$ form an extended Chebyshev system, then by Theorem 2.4 the weights $\hat{w}_1, \dots, \hat{w}_n$ of the Gaussian quadrature are positive. In addition, by the definition of an extended Chebyshev system, the determinant in the right-hand side of (2.25) is nonzero. Thus

$$|J(\hat{x}_1, \dots, \hat{x}_n, \hat{w}_1, \dots, \hat{w}_n)| \neq 0. \qquad \square \tag{2.26}$$

Using the inverse function theorem, we immediately obtain the following corollary.

COROLLARY 2.7. *Under the conditions of Lemma* 2.6, *the Gaussian weights and nodes depend continuously on the weight function.*

2.5. Singular value decomposition. The singular value decomposition (SVD) is a ubiquitous tool in numerical analysis, given for the case of real matrices by the following lemma (see, for instance, [13] for more details).

LEMMA 2.8. *For any $n \times m$ real matrix A, there exist, for some integer p, an $n \times p$ real matrix U with orthonormal columns, an $m \times p$ real matrix V with* *orthonormal columns, and a $p \times p$ real diagonal matrix $S = [s_{ij}]$ whose diagonal entries are nonnegative, such that $A = U \cdot S \cdot V^*$ and $s_{ii} \geq s_{i+1,i+1}$ for all $i = 1, \ldots, p-1$.*

The diagonal entries s_{ii} of S are called singular values; the columns of the matrix V are called right singular vectors; the columns of the matrix U are called left singular vectors.

2.6. SVD of a sequence of functions. A similar decomposition exists (see [5, 16]) if the columns of the matrix A are replaced with functions, as follows.

THEOREM 2.9. *Suppose that the functions $\phi_1, \phi_2, \ldots, \phi_n : [a,b] \to \mathbb{R}$ are square integrable. Then there exist a finite orthonormal sequence of functions $u_1, u_2, \ldots, u_p : [a,b] \to \mathbb{R}$, an $n \times p$ matrix $V = [v_{ij}]$ with orthonormal columns, and a sequence $s_1 \geq s_2 \geq \cdots \geq s_p > 0 \in \mathbb{R}$, for some integer p, such that*

$$\phi_j(x) = \sum_{i=1}^{p} u_i(x) s_i v_{ij} \tag{2.27}$$

for all $x \in [a,b]$ and all $j = 1, \ldots, n$.

The sequence $\{s_i\}$ is uniquely determined by K.

By analogy to the finite-dimensional case, we refer to this factorization as the SVD. We refer to the functions $\{u_i\}$ as singular functions, to the columns of the matrix V as singular vectors, and to the numbers $\{s_i\}$ as singular values.

A popular application of the SVD is for the purpose of "compressing" data. Specifically, it often happens that while the total number n of functions is large, almost all of the coefficients s_j in the decomposition (2.27) are negligibly small. In such cases, (2.27) is truncated after a small number (say, p_0) of terms, and the resulting expansion

$$\phi_j(x) = \sum_{i=1}^{p_0} u_i(x) \cdot s_i \cdot v_{ij} \tag{2.28}$$

is viewed as a compact representation of the original family of functions $\phi_1, \phi_2, \ldots, \phi_n$.

The following theorem states that given a sequence of functions on the interval $[a,b]$, their decomposition of the form (2.28), and a quadrature formula with positive weights on the interval $[a,b]$, the accuracy of the quadrature for the functions $\phi_1, \phi_2, \ldots, \phi_n$ is determined by its accuracy for the singular functions u_j, corresponding to nontrivial singular values. Its proof is an exercise in elementary linear algebra and is omitted.

THEOREM 2.10. *Suppose that under the conditions of Theorem 2.9, ϵ is a positive real number, $1 < p_0 < n$ is an integer, and*

$$\sum_{i=p_0+1}^{n} s_i^2 < \frac{\epsilon^2}{4}. \tag{2.29}$$

Suppose further that the m-point quadrature formula $\{x_i, w_i\}$ integrates the functions u_i exactly, i.e.,

$$\sum_{j=1}^{m} w_j \cdot u_i(x_j) = \int_a^b u_i(x)\, dx \tag{2.30}$$

for all $i = 1, 2, \ldots, p_0$, and that all the weights $w_1, \ldots, w_m$ are positive. Then for each $i = 1, 2, \ldots, n$,

$$\left| \sum_{j=1}^{m} w_j \cdot \phi_i(x_j) - \int_a^b \phi_i(x)\ dx \right| < \epsilon \cdot ||\phi_i||_{L^2}. \tag{2.31}$$

3. Numerical apparatus.

3.1. Continuation method. For Newton's method to converge, the starting point provided to it must be close to the desired solution. One scheme for generating such starting points is the continuation method, described below.

Suppose that in addition to the function $F : \mathbb{R}^n \to \mathbb{R}^n$ whose zero is to be found, another function $G : [0,1] \times \mathbb{R}^n \to \mathbb{R}^n$ is available which possesses the following properties:

- For any $x \in \mathbb{R}^n$,

$$G(1, x) = F(x). \tag{3.1}$$

- The solution of the equation

$$G(0, x) = 0 \tag{3.2}$$

  is known.
- For all $t \in [0,1]$, the equation

$$G(t, x) = 0 \tag{3.3}$$

  has a unique solution x at which the conditions for Newton's method to converge are satisfied.
- The solution x is a continuous function of t.

If these conditions are met, an algorithm for the solution of the equation

$$F(x) = 0 \tag{3.4}$$

is as follows. Let the points t_i, for $i = 1, \ldots, m$, be defined by the formula $t_i = i/m$. Solve in succession the equations

$$\begin{aligned} G(t_1, x) &= 0, \\ G(t_2, x) &= 0, \\ &\vdots \\ G(t_m, x) &= 0 \end{aligned} \tag{3.5}$$

using Newton's method, with the starting point for Newton's method for each equation taken to be the solution of the preceding equation. Due to (3.1), the solution x of the final equation $G(t_m, x) = 0$ is identical to the solution of (3.4); obviously, for sufficiently large m, Newton's method is guaranteed to converge at each step.

Remark 3.1. In practice, it is desirable to choose the smallest m for which the above algorithm will work, in order to reduce the computational cost of the scheme. On the other hand, the largest step $(t_i - t_{i-1})$ for which the Newton method will converge commonly varies as a function of t. Thus the algorithm described in this paper uses an adaptive version of the scheme.

3.2. Continuation scheme. The continuation scheme used is as follows. Let the weight functions $\omega : [0,1] \times [a,b] \to \mathbb{R}^+$ be defined by the formula

$$\omega(\alpha, x) = \alpha\omega_1(x) + (1-\alpha)\sum_{j=1}^{n} \delta(x - c_j), \tag{3.6}$$

where ω_1 is the weight function for which a Gaussian quadrature is desired, δ denotes the Dirac delta function, and the points $c_j \in [a,b]$ are arbitrary distinct points. These weight functions have the following properties:

- With $\alpha = 1$, the weight function is equal to the desired weight function ω_1, due to (3.6).
- With $\alpha = 0$, the Gaussian weights and nodes are

$$w_j = 1, \tag{3.7}$$

$$x_j = c_j \tag{3.8}$$

  for $j = 1, \dots, n$, whatever the functions ϕ_i are (since $\omega(0,x) = 0$, unless $x = c_j$ for some $j \in [1,n]$).
- The quadrature weights and nodes depend continuously on α (by Corollary 2.7).

The intermediate problems that the continuation method solves are the Gaussian quadratures relative to the weight functions $\omega(\alpha, *)$. The scheme starts by setting $\alpha = 0$ then increases α in an adaptive manner until $\alpha = 1$, as follows. A current step size is maintained, by which α is incremented after each successful termination of Newton's method. After each unsuccessful termination of Newton's method, the step size is halved and the algorithm restarts from the point yielded by the last successful termination. After a certain number of successful steps, the current step size is doubled. (Experimentally, the current problem was found to be well suited to an aggressive mode of adaption: in the authors' implementation, the initial value of the step size was chosen to be 0.5, and the step size was doubled after two successful terminations of Newton's method.)

3.3. Starting points. The choice of the points c_j was left indefinite above. In exact arithmetic, and applied to a Chebyshev system, the algorithm would converge for any choice of distinct points (see Lemma 2.6). However, the number of steps of the continuation method, and thus the speed of execution, is affected by the choice. More important, the numerical stability of the scheme might be compromised due to poor conditioning of the matrix J (see (2.24)). Indeed, while Lemma 2.6 guarantees that the matrix J is nonsingular, it says nothing about its condition number. In addition, we will be applying the algorithm to cases where the conditions of Lemma 2.6 are not satisfied. For these reasons, the following method of choosing the starting points was adopted. The method seeks to create a matrix J that is well conditioned. It is a pivoted Gram–Schmidt orthogonalization, altered to operate on pairs of vectors:

1. Choose a set of points $x_1, x_2, \dots, x_m$ on the interval of integration $[a,b]$, such that each of the functions $\phi_1, \phi_2, \dots, \phi_n$, and each of their derivatives, can be interpolated on $[a,b]$ in a well-conditioned manner from values at these points.

2. Create a matrix $\tilde{J}$, of the same form as (2.24), where the points $\{x_j\}$ which determine the columns are the points chosen in step 1. (This matrix thus has $2m$ columns.)

3. Perform the following sequence of operations n times:

(a) Choose the point x_j for which the two columns corresponding to x_j have the largest size. (The issue of what "size" to use is discussed below.)

(b) Orthogonalize the remaining columns to both of those two columns.

The points x_j chosen in step 3(a) are then the starting points c_j used in the continuation method.

The algorithm as specified above is for exact arithmetic. As with Gram–Schmidt, the algorithm is numerically unstable but can be stabilized by an additional reorthogonalization: after step 3(a), reorthogonalize the two new pivot columns to all of the previously chosen pivot columns.

Remark 3.2. The "size of two columns" that was used for step 3(a) was the sum of the norms of the columns, after the second column had been orthogonalized to the first. This poses the obvious danger that one of the two columns chosen might have a small norm, which was covered up by a large norm of its companion. This would render it unsuitable for pivoting; this danger was never realized in our numerical experiments, but if it were, the obvious remedy would be to attempt to change the definition of the size. The authors have not investigated this issue in detail.

3.4. Nested Legendre discretizations of finite sequences of functions.

In this paper, we will be confronted with finite sequences of functions $\phi_1, \phi_2, \dots \phi_n$ on the interval $[a, b]$ possessing the following properties:

- The total number n of functions ϕ_i is reasonably large (e.g., $10,000$).
- The rank of the set $\phi_1, \phi_2, \dots \phi_n$, is low (e.g., 40) to high precision.
- Each of the functions $\phi_1, \phi_2, \dots \phi_n$ is analytic on the interval $[a, b]$, except at a finite (small) number of points; $\phi_i \in L^1[a, b]$ for all $i = 1, 2, \dots, n$.

Now, if we wish to handle (interpolate, integrate, differentiate, etc.) numerically functions of the form

$$\psi(x) = \sum_{i=1}^{n} \alpha_i \cdot \phi_i, \tag{3.9}$$

often it is not convenient to represent them by collections of coefficients $\alpha_1, \alpha_2, \dots \alpha_n$. Indeed, if the functions $\phi_1, \phi_2, \dots \phi_n$ are linearly dependent, the number of coefficients α_i necessary to represent them in the form (3.9) might be grossly excessive, compared to the actual complexity of the function to be represented. Furthermore, the coefficients α_i by themselves provide no mechanism for the integration, interpolation, etc., of functions of the form (3.9); each time such procedures have to be performed, one has to recompute the original functions $\phi_1, \phi_2, \dots \phi_n$. Since the latter is often expensive or impossible, it is desirable to have a purely numerical procedure for representing sums of the form (3.9). Preferably, the scheme should use no information about the functions ϕ_i, except for their values at a finite (preferably not very large) collection of points on $[a, b]$.

When the functions ϕ_i are smooth, a widely used tool for representing them is Chebyshev interpolation: a sufficiently large integer m is chosen, and the functions $\phi_1, \phi_2, \dots \phi_n$ are tabulated at m Chebyshev nodes on $[a, b]$ and obtained at all other points on $[a, b]$ via standard interpolation procedures. While Chebyshev nodes are an extremely good choice, they are not the only one; for example, Gaussian (Legendre) nodes are almost as efficient as the Chebyshev ones when the functions are to be interpolated and are twice as efficient when the functions are to be integrated (see, for example, [4]). When the behavior of the functions ϕ_i is very nonuniform over the interval $[a, b]$, Chebyshev (Gaussian, etc.) interpolation becomes inefficient; for singular functions it is liable to fail completely. In such cases, adaptive Chebyshev interpolation is used, whereby the interval is subdivided into a collection of subintervals so that on each subinterval, all of the functions ϕ_i are accurately approximated by a Chebyshev expansion of low order; the subdivisions are performed automatically. When some (or all) of the functions ϕ_i have singularities on the interval $[a, b]$, schemes of this type cluster the subintervals near each singularity until the subinterval nearest to the singularity is so small as to be ignorable for the purposes of the calculations to be performed.

In the first stage of the algorithm that we use, we build a nested Chebyshev discretization of the interval $[a, b]$ for each of the functions ϕ_i. In the second stage, all such discretizations are merged to obtain a single discretization by which all of the functions ϕ_i are adequately represented. In the third stage, n *Legendre* nodes are constructed on each of the obtained intervals.

Stage 1.

1. Choose the precision ϵ and some reasonably large m. (In actual computations, we use $m = 16$.)

2. Construct the m Chebyshev nodes $x_1^{[a,b]}$, $x_2^{[a,b]}$, …, $x_m^{[a,b]}$ on the interval $[a, b]$. Evaluate ϕ at the nodes $x_1^{[a,b]}$, $x_2^{[a,b]}$, …, $x_m^{[a,b]}$, obtaining the values $\phi_1^{[a,b]}$, $\phi_2^{[a,b]}$, …, $\phi_m^{[a,b]}$.

3. Subdivide the interval $[a, b]$ into the subintervals $[a, (a + b)/2]$, $[(a + b)/2, b]$. Construct the Chebyshev nodes $x_1^{[a,(a+b)/2]}$, $x_2^{[a,(a+b)/2]}$, …, $x_m^{[a,(a+b)/2]}$ on the interval $[a, (a + b)/2]$ and the Chebyshev nodes $x_1^{[(a+b)/2,b]}$, $x_2^{[(a+b)/2,b]}$, …, $x_m^{[(a+b)/2,b]}$ on the interval $[(a+b)/2, b]$. Evaluate the function ϕ at the nodes $x_1^{[a,(a+b)/2]}$, $x_2^{[a,(a+b)/2]}$, …, $x_m^{[a,(a+b)/2]}$, $x_1^{[(a+b)/2,b]}$, $x_2^{[(a+b)/2,b]}$, …, $x_m^{[(a+b)/2,b]}$, obtaining the values $\phi_1^{[a,(a+b)/2]}$, $\phi_2^{[a,(a+b)/2]}$, …, $\phi_m^{[a,(a+b)/2]}$, $\phi_1^{[(a+b)/2,b]}$, $\phi_2^{[(a+b)/2,b]}$, …, $\phi_m^{[(a+b)/2,b]}$, respectively.

4. Interpolate the values of the function ϕ from the nodes $x_1^{[a,b]}$, $x_2^{[a,b]}$, …, $x_m^{[a,b]}$ on the interval $[a, b]$ to the nodes $x_1^{[a,(a+b)/2]}$, $x_2^{[a,(a+b)/2]}$, …, $x_m^{[a,(a+b)/2]}$, $x_1^{[(a+b)/2,b]}$, $x_2^{[(a+b)/2,b]}$, …, $x_m^{[(a+b)/2,b]}$ on the intervals $[a, (a + b)/2]$, $[(a + b)/2, b]$. If the interpolated values agree to the precision ϵ with the values $\phi_1^{[a,(a+b)/2]}$, $\phi_2^{[a,(a+b)/2]}$, …, $\phi_m^{[a,(a+b)/2]}$, $\phi_1^{[(a+b)/2,b]}$, $\phi_2^{[(a+b)/2,b]}$, …, $\phi_m^{[(a+b)/2,b]}$ calculated directly in step 2 above, the algorithm concludes that the function ϕ is adequately resolved by the m Chebyshev nodes on the interval $[a, b]$; otherwise, the procedure is repeated recursively for each of the subintervals $[a, (a + b)/2]$, $[(a + b)/2, b]$.

Stage 2. Store the ends (left and right) of all subintervals in all subdivisions in a single array a. Sort the elements of a; remove multiple elements in a. The resulting array of points on the interval $[a, b]$ (including the points a, b) is the array of ends of subintervals of the final subdivision.

Stage 3. Construct an m-point Legendre discretization of each of the subintervals obtained in Stage 2 above.

Remark 3.3. In the algorithm above, we use Chebyshev discretizations in Stage 1 to construct the subdivision of the interval $[a, b]$; in subsequent calculations we use Legendre discretizations. The reason for this choice is that the interpolations in Stage 1 are carried out more efficiently with *Chebyshev* discretizations, via the discrete cosine transform and related tools; the Legendre discretizations used subsequently lead to linear interpolation schemes that preserve inner products (see the following subsection).

Remark 3.4. The scheme of this subsection is a fairly reliable apparatus for the automatic discretization of sets of (more or less) arbitrary user-specified functions. While it is very easy to construct counterexamples in which the algorithm will fail to resolve some (or all) of the input functions, this problem has never been encountered in our practice.

3.5. Approximation of SVD of a sequence of functions. This section describes a numerical procedure for computing an approximation to the SVD of a sequence of functions.

The algorithm uses quadratures possessing the following property.

Definition 3.1. *We will say that the combination of a quadrature and an interpolation scheme* preserves inner products *on an interval* $[a, b]$ *if it possesses the following properties:*

- *The nodes of the quadrature are identical to the nodes of the interpolation scheme.*
- *The function that is output by the interpolation scheme depends in a linear fashion on the values input to the interpolation scheme.*
- *The quadrature integrates exactly any product of two interpolated functions; that is, for any two functions* $f, g : [a, b] \to \mathbb{R}$ *produced by the interpolation scheme, the integral*

$$\int_a^b f(x) \cdot g(x) dx \tag{3.10}$$

  is computed exactly by the quadrature.

Quadratures and interpolation schemes possessing this property include the following.

Example 3.1. The combination of a (classical) Gaussian quadrature at Legendre nodes and polynomial interpolation at the same nodes preserves inner products, since polynomial interpolation on n nodes produces an interpolating polynomial of order $n-1$, the product of two such polynomials is a polynomial of order $2n-2$, and a Gaussian quadrature integrates exactly all polynomials up to order $2n-1$.

Example 3.2. If an interval is broken into several subintervals, and a quadrature and interpolation scheme preserving inner products is used on each subinterval, then the arrangement as a whole preserves inner products on the original interval. (This follows directly from the definition.)

Example 3.3. The combination of the trapezoidal rule on the interval $[0, 2\pi]$ and Fourier interpolation (using the interpolation functions $1, \cos x, \sin x, \cos 2x, \sin 2x, \ldots, \cos nx, \sin nx$) preserves inner products.

The algorithm described below takes as input a sequence of functions $\phi_1, \phi_2, \ldots, \phi_n : [a, b] \to \mathbb{R}$. It uses as a tool a quadrature and a linear interpolation scheme on the interval $[a, b]$ preserving inner products; the weights and nodes of this quadrature will be denoted by $w_1, \ldots, w_n \in \mathbb{R}$ and $x_1, \ldots, x_n \in [a, b]$, respectively. As will be shown below, the accuracy of the algorithm is then determined by the accuracy to which the interpolation scheme approximates the functions $\phi_1, \phi_2, \ldots, \phi_n$.

The output of the algorithm is a sequence of functions $u_1, \ldots, u_p : [a, b] \to \mathbb{R}$, a sequence of vectors $v_1, \ldots, v_p \in \mathbb{R}^n$, and a sequence of singular values $s_1, \ldots, s_p \in \mathbb{R}$, forming an approximation to the singular value decomposition of $\phi_1, \phi_2, \ldots, \phi_n$.

Description of the algorithm.

1. Construct the $n \times m$ matrix $A = [a_{ij}]$ defined by the formula

$$a_{ij} = \phi_j(x_i) \cdot \sqrt{w_i}. \tag{3.11}$$

2. Compute the SVD of A, to produce the factorization

$$A = U \circ S \circ V^*, \tag{3.12}$$

where $U = [u_{ij}]$ is an $n \times p$ matrix with orthonormal columns, $V = [v_{ij}]$ is an $m \times p$ matrix with orthonormal columns, and S is a $p \times p$ diagonal matrix whose jth diagonal entry is s_j.

3. Construct the $n \times p$ values $u_k(x_i)$ defined by the formula

$$u_k(x_i) = u_{ik}/\sqrt{w_i}. \tag{3.13}$$

4. For any desired point $x \in [a, b]$, evaluate the functions $u_k : [a, b] \to \mathbb{R}$ using the interpolation scheme on $[a, b]$.

The proof of the following theorem can be found (in a considerably more general form) in [15].

Theorem 3.2. *Suppose that the combination of the quadrature and interpolation scheme with weights and nodes $w_1, \dots, w_n \in \mathbb{R}$ and $x_1, \dots, x_n \in [a, b]$, respectively, preserves inner products on $[a, b]$. For any sequence of functions $\phi_1, \phi_2, \dots, \phi_n : [a, b] \to \mathbb{R}$, let $u_i : [a, b] \to \mathbb{R}$, $v_{ij} \in \mathbb{R}$, and $s_i \in \mathbb{R}$ be defined in* (3.11)–(3.13) *for all $i = 1, \dots, p$. Then*

1. *The functions u_i are orthonormal, i.e.,*

$$\int_a^b u_i(x) u_k(x) dx = \delta_{ik} \tag{3.14}$$

for all $i, k = 1, \dots, p$, with δ_{ik} the Kronecker symbol.

2. *The columns of V are orthonormal, i.e.,*

$$\sum_{j=1}^{n} v_{ij} v_{kj} dx = \delta_{ik} \tag{3.15}$$

for all $i, k = 1, \dots, p$.

3. *The sequence of functions $\tilde{\phi}_1, \tilde{\phi}_2, \dots, \tilde{\phi}_n : [a, b] \to \mathbb{R}$ defined by the formula*

$$\tilde{\phi}_k(x) = \sum_{j=1}^{p} s_j u_j(x) v_{jk} \tag{3.16}$$

is identical to the sequence of functions produced by sampling the functions $\phi_1, \phi_2, \dots, \phi_n$ at the points $\{x_i\}$ then interpolating with the interpolation scheme on $[a, b]$.

4. Numerical algorithm.

This section describes a numerical algorithm for the evaluation of nodes and weights of generalized Gaussian quadratures. The algorithm's input is a sequence of functions $\phi_1, \dots, \phi_{2n} : [a, b] \to \mathbb{R}$ and the precision ϵ to which the quadratures are to be calculated; its output is the weights and nodes of the quadrature. The functions ϕ_i are supplied by the user in the form of a subroutine, with input parameters (x, i) and output parameter $\phi_i(x)$. The algorithm uses the components described in the preceding section.

1. The interval $[a, b]$ is discretized via the scheme described in subsection 3.4, so that all functions ϕ_1, $\phi_2, \dots,$ ϕ_n are represented to the precision ϵ.

2. All of the functions ϕ_1, $\phi_2, \dots,$ ϕ_n are tabulated at the nodes of the discretization obtained in step 1 above, and the SVD is obtained of the sequence of functions

Table 1
16-*node quadrature for functions of the form* (5.1) *with* $\alpha \in [-0.6, 1]$, $N = 4$, *and precision* 10^{-15}.

x_i	w_i
0.1646476245461994E-18	0.2477997131959177E-17
0.2004881755033198E-13	0.1863311166024058E-12
0.4902407997203263E-10	0.3215991324579055E-09
0.1396853977847601E-07	0.6788563189534853E-07
0.9715236454504147E-06	0.3586206403622012E-05
0.2502196135803993E-04	0.7130636866829449E-04
0.3120851149673110E-03	0.6951436010759356E-03
0.2264576163994000E-02	0.3979838127986921E-02
0.1086917746927712E-01	0.1515746778330600E-01
0.3777218640280392E-01	0.4182483334409624E-01
0.1013279037973986E+00	0.8854031057518543E-01
0.2196196157836697E+00	0.1490380907486389E+00
0.3972680999338400E+00	0.2028312538451011E+00
0.6135562966157080E+00	0.2216836945000430E+00
0.8216868417553706E+00	0.1844567448110479E+00
0.9636466562372551E+00	0.9171766188102896E-01

$\phi_1, \phi_2, \ldots, \phi_n$ via the scheme described in subsection 3.5; we will be denoting the obtained singular values by $\lambda_1, \lambda_2, \ldots$.

3. Denoting by k the positive integer number such that $\lambda_{2 \cdot k+1} \leq \epsilon \leq \lambda_{2 \cdot k-1}$, we observe that any quadrature formula with positive coefficients that integrates the obtained singular functions $u_1, u_2, \ldots u_{2 \cdot k}$ exactly will integrate all of the functions $\phi_1, \phi_2, \ldots, \phi_n$ with precision ϵ (see Theorem 2.10). The remainder of the algorithm is devoted to constructing a k-point quadrature formula that will integrate the functions $u_1, u_2, \ldots u_{2 \cdot k}$ exactly.

4. The scheme of subsection 3.3 is used to find the starting nodes $x_1^0, x_2^0, \ldots, x_k^0$ for the continuation process of subsection 3.2.

5. An adaptive version of the continuation method of subsection 3.2 is used to obtain the k-point quadrature for the functions $u_1, u_2, \ldots, u_{2 \cdot k}$; on each step, the Newton algorithm described in subsection 2.4 is used to solve the system (2.21) defining the nodes and roots of the quadrature formula.

Remark 4.1. We would like to reiterate that the quadrature formulae produced by the procedure of this section do not integrate the user-specified functions $\phi_1, \phi_2, \ldots, \phi_n$ exactly; instead, they produce approximations to the integrals. Needless to say, the two are indistinguishable as long as the chosen precision ϵ is less than the machine precision.

5. Numerical examples. A variety of quadratures was generated via the algorithm of this paper; several of these are presented below to illustrate its performance. In Examples 5.1 and 5.2, the calculations were performed in extended precision (Fortran `REAL*16`) arithmetic, to ensure full double precision in the obtained result. In Example 5.3, the calculations were performed in double precision, since the accuracy of the quadrature listed in Table 5 is only nine digits.

Example 5.1. An obvious problem of interest is the integration on an interval of functions that have a singularity at one end of that interval (or at both ends); of particular interest are power and logarithmic singularities. Many techniques have been proposed for dealing with such problems (see, for example, [1]). While some of these approaches are quite effective for some of the singularities, they have the drawback that each deals only with one particular singularity. In this example, we

TABLE 2

8-node quadrature for functions of the form (5.1) *with* $\alpha \in [-0.6, 1]$, $N = 4$, *and precision* 10^{-7}.

x_i	w_i
0.1312034302206730E-07	0.1393140646786704E-06
0.2793817088002595E-04	0.1549484313499085E-03
0.2038371172070937E-02	0.6673805929140874E-02
0.2702722219647910E-01	0.5430869272244519E-01
0.1343993651970034E+00	0.1694172186704161E+00
0.3682213359901025E+00	0.2898751155944595E+00
0.6792045461791814E+00	0.3076390470455203E+00
0.9309603731369270E+00	0.1719310626051804E+00

TABLE 3

19-node quadrature for functions of the form (5.1) *with* $\alpha \in [-0.6, 1]$, $N = 9$, *and precision* 10^{-15}.

x_i	w_i
0.1846942465536925E-18	0.2756403589261532E-17
0.1989380701597045E-13	0.1824804592695847E-12
0.4312593909743526E-10	0.2777592139982985E-09
0.1092964737770428E-07	0.5186860611615611E-07
0.6810397860708155E-06	0.2442433440466041E-05
0.1588655973896037E-04	0.4380009969129837E-04
0.1818339165855430E-03	0.3906506115636250E-03
0.1227551979000820E-02	0.2077051291912717E-02
0.5556316902145769E-02	0.7461053476901383E-02
0.1847419717287859E-01	0.1978838865640943E-01
0.4825255045366560E-01	0.4136988974623410E-01
0.1041307630444531E+00	0.7157248041035670E-01
0.1928680775398894E+00	0.1060884317057585E+00
0.3153775090195431E+00	0.1377804712043467E+00
0.4647713088385197E+00	0.1585409276263068E+00
0.6264814981191495E+00	0.1614751848557232E+00
0.7804757620006211E+00	0.1428196856993585E+00
0.9050563637732498E+00	0.1031243266706421E+00
0.9813553783808000E+00	0.4746516336480648E-01

present quadrature rules for the integration of functions of the form

$$\sum_{k=0}^{n} \left(\gamma_k \cdot log(x) + \sum_{j=1}^{m} \beta_{k,j} \cdot x^{\alpha_j} \right) \cdot P_k(x), \tag{5.1}$$

where P_k denotes the (normalized) orthogonal polynomial of order k on the interval $[0, 1]$; $\beta_{k,j}$, γ_k are arbitrary real numbers; and α_j are arbitrary real numbers on the interval $[-0.6, 1]$.

To design such quadratures, we choose a reasonably large natural m; construct m Legendre nodes $\alpha_1, \alpha_2, \ldots, \alpha_m$ on the interval $[-0.6, 1]$, and use all functions of the forms

$$P_k(x) \cdot x^{\alpha_j}, \tag{5.2}$$

$$P_k(x) \cdot log(x) \tag{5.3}$$

as input functions ϕ_i for the algorithm of the preceding section. The result is a set of quadratures for functions of the forms (5.2), (5.3). A somewhat involved analytical calculation shows that for sufficiently large m, the obtained quadratures will work for

TABLE 4
26-node quadrature for functions of the form (5.1) *with* $\alpha \in [-0.6, 1]$, $N = 19$, *and precision* 10^{-15}.

x_i	w_i
0.2852686209735951E-20	0.4390385492743041E-19
0.4655349788609637E-15	0.4445881189691443E-14
0.1432147899313873E-11	0.9689649973398580E-11
0.4915792345704672E-09	0.2471786670704959E-08
0.3986884553883893E-07	0.1527652265503579E-06
0.1168849078081257E-05	0.3470933550491954E-05
0.1630549221175312E-04	0.3803166416108812E-04
0.1307331567674635E-03	0.2422240257088061E-03
0.6884061227847875E-03	0.1022568448159836E-02
0.2620448293548410E-02	0.3143745934305781E-02
0.7740029188833982E-02	0.7549238041954824E-02
0.1872452403074940E-01	0.1495112040361046E-01
0.3869460001276389E-01	0.2548756008178511E-01
0.7058074961479188E-01	0.3865021281644121E-01
0.1165353335503884E+00	0.5342389042306681E-01
0.1775282580420220E+00	0.6849323863305738E-01
0.2531447462199369E+00	0.8243302008328313E-01
0.3415558481256653E+00	0.9386320384208941E-01
0.4396281348394975E+00	0.1015733726852001E+00
0.5431447278197111E+00	0.1046214551363520E+00
0.6471126706707170E+00	0.1024074963963311E+00
0.7461308154896283E+00	0.9472049436813551E-01
0.8347900655356778E+00	0.8175595131244442E-01
0.9080759999882411E+00	0.6410309004863602E-01
0.9617441758037388E+00	0.4270384642243640E-01
0.9926478556999123E+00	0.1881261305258270E-01

all functions of the form (5.1), and our numerical experiments show that $m = 100$ ensures full double precision accuracy for all $\alpha_j \in [-0.6, 1]$.

In Tables 1–5, we list quadrature nodes and weights for $n = 4, 9, 19, 29$. In Tables 1, 3, 4, and 5, the number of nodes is chosen to guarantee 15-digit accuracy. In Table 2, the number of nodes is chosen to guarantee 7 digits.

Example 5.2. The quadrature rules in this example are very similar to those in Example 5.1 except here we construct quadrature rules for functions singular at *both* ends of the interval where they are to be integrated. Specifically, integrands have the form

(5.4)
$$\sum_{k=0}^{n} \left(\sum_{j=1}^{m} (a_{k,j} \cdot (1+x)^{\alpha_j} + b_{k,j} \cdot (1-x)^{\alpha_j}) + c_k \cdot log(1+x) + d_k \cdot log(1-x) \right) \cdot P_k(x),$$

where P_k denotes the (normalized) orthogonal polynomial of order k on the interval $[-1, 1]$; $a_{k,j}$, $b_{k,j}$, c_k, d_k are arbitrary real numbers; and α_j are arbitrary real numbers on the interval $[-0.1, 1]$. Quadrature nodes and weights for $n = 4, 9, 19, 39$ are listed in Tables 6, 7, 8, and 9, respectively; in all cases, the precision is 10^{-15}.

Example 5.3. In this example, we construct a direct generalization of quadratures constructed in Example 5.1, permitting the integrands to have power and logarithmic singularities *at arbitrary points on the closed half-line to the left of the interval of*

TABLE 5
36-node quadrature for functions of the form (5.1) *with* $\alpha \in [-0.6, 1]$*,* $N = 39$*, and precision* 10^{-15}.

x_i	w_i
0.1174238417413926E-19	0.1769042596381234E-18
0.1422439193737780E-14	0.1318732300270049E-13
0.3350676698582048E-11	0.2181187238172082E-10
0.8987762100979194E-09	0.4306047388907762E-08
0.5804062676082615E-07	0.2097251047066944E-06
0.1381879982602796E-05	0.3830347070073085E-05
0.1599014834456195E-04	0.3447814965093908E-04
0.1086072834052024E-03	0.1843012333973045E-03
0.4939690780979653E-03	0.6658876227138618E-03
0.1653457719227906E-02	0.1785581170381193E-02
0.4371083474213578E-02	0.3817614649487054E-02
0.9635942477742897E-02	0.6885390581283880E-02
0.1847241513238332E-01	0.1094085630140653E-01
0.3179190367214565E-01	0.1581728538518057E-01
0.5030636405050507E-01	0.2129142636454853E-01
0.7449442868952319E-01	0.2712481569656370E-01
0.1045979502135202E+00	0.3308456773919071E-01
0.1406326475828715E+00	0.3895216905306892E-01
0.1824044449022998E+00	0.4452688339606666E-01
0.2295280679235570E+00	0.4962723403902098E-01
0.2814468220422235E+00	0.5409202169130247E-01
0.3374533767644982E+00	0.5778135262022458E-01
0.3967116179369689E+00	0.6057773920656186E-01
0.4582796041927400E+00	0.6238718893653459E-01
0.5211335571597729E+00	0.6314016307782669E-01
0.5841926980689389E+00	0.6279229386348975E-01
0.6463446423449487E+00	0.6132477029316637E-01
0.7064709858680002E+00	0.5874432665998542E-01
0.7634726623238107E+00	0.5508279084756487E-01
0.8162946187294954E+00	0.5039617034177984E-01
0.8639493438008133E+00	0.4476327290202123E-01
0.9055387898384755E+00	0.3828387702474601E-01
0.9402742542357631E+00	0.3107648956468336E-01
0.9674938463383342E+00	0.2327578565976658E-01
0.9866773942995437E+00	0.1503024417658587E-01
0.9974613070359063E+00	0.6508977351752366E-02

integration. Specifically, integrands have the form

$$\sum_{k=0}^{n} \left(\gamma_k \cdot log(x+h) + \sum_{j=1}^{m} \beta_{k,j} \cdot (x+h)^{\alpha_j} \right) \cdot P_k(x), \tag{5.5}$$

where P_k denotes the (normalized) orthogonal polynomial of order k on the interval $[0, 1]$; $\beta_{k,j}$, γ_k are arbitrary real numbers; α_j are arbitrary real numbers on the interval $[-0.65, 1]$; and h is an arbitrary positive real number. In this case, the calculations were conducted in double precision; the 38-node quadrature formula for $n = 19$ is given in Table 10; its precision is 10^{-9}.

Several observations can be made from Tables 1–8 and from the more detailed numerical experiments we have conducted:

- The algorithm of this paper is always effective for Chebyshev systems; it almost always works for non-Chebyshev systems.
- The scheme does not lose very many digits compared with the machine precision; when the calculations are performed in double precision, the quadratures

TABLE 6
22-node quadrature for functions of the form (5.5) *with* $\alpha \in [-0.1, 1]$*,* $N = 4$*, and precision* 10^{-15}.

$\pm x_i$	w_i
0.1666008119316040E+00	0.3286464553329054E+00
0.4736467937561296E+00	0.2782402062916909E+00
0.7129463900017805E+00	0.1977249261400840E+00
0.8687173264995090E+00	0.1158087624474726E+00
0.9515411665787298E+00	0.5425992604604305E-01
0.9862971262509680E+00	0.1943874113675287E-01
0.9972429072629104E+00	0.4979788483749470E-02
0.9996464539418006E+00	0.8238003428108275E-03
0.9999757993153293E+00	0.7462712208720397E-04
0.9999993605804343E+00	0.2746237603563529E-05
0.9999999970230195E+00	0.2041880191195951E-07

TABLE 7
27-node quadrature for functions of the form (5.5) *with* $\alpha \in [-0.1, 1]$*,* $N = 9$*, and precision* 10^{-15}.

$\pm x_i$	w_i
0.0000000000000000E+00	0.1969765126094452E+00
0.1953889665467211E+00	0.1922287111905558E+00
0.3814298736462841E+00	0.1784269782500965E+00
0.5496484616443740E+00	0.1568677485350913E+00
0.6932613279607421E+00	0.1296176364576521E+00
0.8078808016610349E+00	0.9937321489137896E-01
0.8920478424190657E+00	0.6925317917837661E-01
0.9475053154471952E+00	0.4247396818782292E-01
0.9790448975739819E+00	0.2179872525134398E-01
0.9936444652327659E+00	0.8672220251831163E-02
0.9986936386311707E+00	0.2388475528070173E-02
0.9998477986092101E+00	0.3837648653769931E-03
0.9999927156219827E+00	0.2671422777541431E-04
0.9999999335937359E+00	0.4068798910349743E-06

can be obtained to 11 or 12 digits; the accuracy of quadratures in Tables 1–9 is full double precision; we used extended precision arithmetic in Fortran to obtain them.

- The algorithm of this paper is not very efficient. For example, the quadrature formula in Table 1 took about two minutes of CPU time on UltraSPARC 2; the quadrature in Table 8 took about two hours of CPU time. Of course, extended precision on the UltraSPARC is quite inefficient; in double precision, Table 8 took about four minutes to construct. In any event, the quadratures of the type presented in this paper need not be constructed "on the fly"; the nodes and weights can be precomputed and stored. From this point of view, the CPU time requirements of our algorithm are not excessive. Still, its CPU time requirements grow as n^3 for large n, making it unsuitable for the construction of quadratures of very high order.

6. Generalizations and conclusions.

We have constructed a scheme for the design of generalized Gaussian quadratures for fairly broad classes of functions. The results presented here should be viewed as somewhat experimental, since while the algorithm appears to work under quite general conditions, we can *prove* only that it *has to* work for Chebyshev systems.

Several possible extensions of the work suggest themselves:

1. Quadratures of the type designed in this paper can be used within compound

Table 8

33-*node quadrature for functions of the form* (5.5) *with* $\alpha \in [-0.1, 1]$, $N = 19$, *and precision* 10^{-15}.

$\pm x_i$	w_i
0.0000000000000000E+00	0.1802406542699465E+00
0.1789856568226836E+00	0.1764865559769247E+00
0.3505713663705831E+00	0.1655482040246752E+00
0.5079970396268890E+00	0.1483733690643724E+00
0.6457344058749438E+00	0.1264620956535221E+00
0.7599840782344723E+00	0.1017484935648103E+00
0.8490304782768580E+00	0.7643386171408831E-01
0.9134021329241244E+00	0.5276203409291129E-01
0.9557717316319267E+00	0.3272086426808218E-01
0.9805181730564275E+00	0.1766845539228831E-01
0.9929045523533901E+00	0.7963812531655223E-02
0.9979798758935006E+00	0.2833884283485953E-02
0.9995837651123616E+00	0.7387521680930171E-03
0.9999445617386989E+00	0.1267394032662049E-03
0.9999960165362139E+00	0.1207609748958691E-04
0.9999998889650372E+00	0.4709227238502033E-06
0.9999999994557687E+00	0.3706639850258617E-08

Table 9

45-*node quadrature for functions of the form* (5.5) *with* $\alpha \in [-0.1, 1]$, $N = 39$, *and precision* 10^{-15}.

$\pm x_i$	w_i
0.0000000000000000E+00	0.1138212938786054E+00
0.1135283181390291E+00	0.1129431358863252E+00
0.2253080046824045E+00	0.1103317059272695E+00
0.3336364252858657E+00	0.1060558645237672E+00
0.4369024052356911E+00	0.1002294986469973E+00
0.5336306707891807E+00	0.9301028558331059E-01
0.6225248777667337E+00	0.8459812566475355E-01
0.7025089656717720E+00	0.7523338442881639E-01
0.7727667118189729E+00	0.6519506433099722E-01
0.8327794264993337E+00	0.5479889055074179E-01
0.8823615451977041E+00	0.4439489209928996E-01
0.9216930322777481E+00	0.3436308131973152E-01
0.9513451962287941E+00	0.2510376733595393E-01
0.9722913641056944E+00	0.1701539437521317E-01
0.9858845322639776E+00	0.1044852849794223E-01
0.9937724959340503E+00	0.5626146436355554E-02
0.9977200386244100E+00	0.2543352365327656E-02
0.9993454278943935E+00	0.9118380718941661E-03
0.9998636273258416E+00	0.2403706487446808E-03
0.9999815974719829E+00	0.4181929949775085E-04
0.9999986596740707E+00	0.4045883666118617E-05
0.9999999622133619E+00	0.1599158044436823E-06
0.9999999998137450E+00	0.1268296767711113E-08

quadrature rules, not unlike the classical Gaussian quadratures. In particular, they can be substituted for Gaussian quadratures in the scheme described in subsection 3.4. If the functions to be integrated have (for example) power singularities at the left end of the interval of integration, the quadrature rules in Example 5.1 will eliminate the bunching of nodes near the left end of the interval. In this respect, of particular interest are quadratures of the type found in Example 5.3, since their use will eliminate the bunching of quadrature nodes near the ends of the interval for integrands with

Table 10

38-node quadrature for functions of the form (5.5) *with* $\alpha \in [-0.65, 1]$, $N = 19$, *and precision* 10^{-9}.

x_i	w_i
0.7629165866352161E-18	0.4643955333268610E-17
0.3799719398931375E-16	0.1132690565299208E-15
0.5684549949701512E-15	0.1423549582265871E-14
0.6085909916179373E-14	0.1371876219104025E-13
0.5277191865393953E-13	0.1094397021531007E-12
0.3900442913791902E-12	0.7534990994077416E-12
0.2535538557277294E-11	0.4603432835276850E-11
0.1481755662897140E-10	0.2545533729683496E-10
0.7911595380511587E-10	0.1293022088581050E-09
0.3907746000477183E-09	0.6102781198001779E-09
0.1803070816493823E-08	0.2700678436986190E-08
0.7833265344260583E-08	0.1128792193586090E-07
0.3224897189563689E-07	0.4482855569803782E-07
0.1264894823726299E-06	0.1700035548631482E-06
0.4747932260937661E-06	0.6182057321480894E-06
0.1711978528765632E-05	0.2163108715557027E-05
0.5948052018171647E-05	0.7302447810573277E-05
0.1995877304286260E-04	0.2382492261847977E-04
0.6475274273537152E-04	0.7511062044871306E-04
0.2029004100170709E-03	0.2279609908900293E-03
0.6109309950274235E-03	0.6592765068003472E-03
0.1747449285439932E-02	0.1781666222619331E-02
0.4661579935095226E-02	0.4378093849756735E-02
0.1135932523990354E-01	0.9537600800370288E-02
0.2491532030262493E-01	0.1820679046524441E-01
0.4902801284057732E-01	0.3060746663786768E-01
0.8713816071641225E-01	0.4600643316091537E-01
0.1415514175271372E+00	0.6292513465068938E-01
0.2128806314974303E+00	0.7951989233968431E-01
0.2998564528132552E+00	0.9391761648476182E-01
0.3994239415560721E+00	0.1044517799613406E+00
0.5070313867113639E+00	0.1098153664961849E+00
0.6170411438386144E+00	0.1091553255900476E+00
0.7232121752054713E+00	0.1021230666276667E+00
0.8192137516286219E+00	0.8888680524875885E-01
0.8991333728333283E+00	0.7010796674100402E-01
0.9579443204807173E+00	0.4688508195206744E-01
0.9919093183441774E+00	0.2069742637648333E-01

power singularities *anywhere on* $\mathbb{R}$ *outside the interval of integration.* Furthermore, one does not have to replace classical Gaussian quadratures with ours on all of the subintervals of a compound rule; it is sufficient to do so only on those subintervals near the ends of the interval of integration. In other situations, different special-purpose generalized Gaussian quadratures might be used. Such adaptive compound rules have been constructed; a paper describing them is in preparation.

2. While our numerical experiments indicate that the scheme of this paper works under very general conditions, we have been able to prove only that it has to work for Chebyshev systems (see subsection 2.1). This discrepancy seems to indicate that it might be profitable to investigate generalizations of Theorem 2.4 to sets of functions other than Chebyshev systems.

3. By combining Observation 2.1 and Remark 2.3 with results in sections 3 and 4, it is fairly straightforward to construct algorithms for the efficient interpolation of fairly large classes of singular functions. For example, the nodes $x_1, x_2, \ldots, x_{36}$ in

Table 5 lead to a stable interpolation formula on the interval $[0,1]$ for all functions of the form

$$\sum_{k=0}^{n} P_k(x) \cdot \sum_{j=1}^{m} \beta_{k,j} \cdot x^{\alpha_j} \tag{6.1}$$

with $-0.3 \leq \alpha_j \leq 1$, $0 \leq k \leq 19$, and the precision of interpolation 10^{-15}. Interpolation schemes of this type are currently under vigorous investigation and will be reported in the near future.

4. In many situations (especially in the numerical solution of partial differential equations), it is desirable to have "quadrature" formulae that, in addition to evaluating integrals, would evaluate certain pseudodifferential operators, i.e., derivative, Hilbert transform, derivative of the Hilbert transform, etc. Clearly, such quadratures cannot have positive weights, except for the Hilbert transform. Several such quadratures have been constructed numerically, and the appropriate theory appears to be fairly straightforward; this work will be reported at a later date.

5. While the theory of Gaussian quadratures in one dimension is extremely simple and is well understood, no similar theory exists in higher dimensions, except for a few scattered results (see, for example, [9, 14]). The approach of this paper is quite different from the classical Gaussian quadratures, and it appears possible to generalize it (at least formally) to higher dimensions. While the advantages of such a construction would be significant, our investigation of it is at a very early stage. If successful, it will be reported at a later date.

REFERENCES

[1] R. Bulirsch and J. Stoer, *Fehlerabschätzungen und Extrapolation mit Rationalen Funktionen bei Verfahren vom Richardson-Typus*, Numer. Math., 6 (1964), pp. 413–427.

[2] F. Gantmacher and M. Krein, *Oscillation Matrices and Kernels and Small Oscillations of Mechanical Systems*, 2nd ed., Gosudarstv. Izdat. Tehn-Teor. Lit., Moscow, 1950 (in Russian).

[3] W. Gautschi, *On generating orthogonal polynomials*, SIAM J. Sci. and Statist. Comput., 3 (1982), pp. 289–317.

[4] D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, CBMS-NSF Regional Conf. Ser. Appl. Math. 26, SIAM, Philadelphia, PA, 1977.

[5] T. Hrycak and V. Rokhlin, *An improved fast multipole algorithm for potential fields*, SIAM J. Sci. Comput., 19 (1998), pp. 1804–1826.

[6] S. Karlin, *The existence of eigenvalues for integral operators*, Trans. Amer. Math. Soc., 113 (1964), pp. 1–17.

[7] S. Karlin and W. J. Studden, *Tchebycheff Systems with Applications in Analysis and Statistics*, Wiley-Interscience, New York, 1966.

[8] M. G. Krein, *The Ideas of P.L. Chebyshev and A.A. Markov in the Theory of Limiting Values of Integrals*, Amer. Math. Soc. Transl. Ser. 2, 12, AMS, Providence, RI, 1959, pp. 1–122.

[9] J. N. Lyness and D. Jespersen, *Moderate degree symmetric quadrature rules for the triangle*, J. Inst. Math. Appl., 15 (1975), pp. 19–32.

[10] J. Ma, V. Rokhlin, and S. Wandzura, *Generalized Gaussian quadratures rules for systems of arbitrary functions*, SIAM J. Numer. Anal., 33 (1996), pp. 971–996.

[11] A. A. Markov, *On the Limiting Values of Integrals in Connection with Interpolation*, Zap. Imp. Akad. Nauk. Fiz.-Mat. Otd. (8) 6 (1898) (in Russian).

[12] A. A. Markov, *Selected Papers on Continued Fractions and the Theory of Functions Deviating Least from Zero*, OGIZ, Moscow, Leningrad, 1948 (in Russian).

[13] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, 2nd ed., Springer-Verlag, Berlin, New York, 1993.

[14] S. Wandzura and H. Xiao, *Quadrature Rules on Triangles in* $\mathbb{R}^2$, Technical report YALEU/DCS/RR-1168, Yale University, New Haven, CT, 1998.

[15] N. Yarvin and V. Rokhlin, *Generalized Gaussian quadratures and singular value decompositions of integral operators*, SIAM J. Sci. Comput., 20 (1999), pp. 699–718.

[16] N. Yarvin and V. Rokhlin, *An improved fast multipole algorithm for potential fields on the line*, SIAM J. Numer. Anal., 36 (1999), pp. 629–666.

SIAM J. Optim.
Vol. 9, No. 4, pp. 924–947

TWO-STEP ALGORITHMS FOR NONLINEAR OPTIMIZATION WITH STRUCTURED APPLICATIONS*

ANDREW R. CONN†, LUÍS N. VICENTE‡, and CHANDU VISWESWARIAH§

To John Dennis on the occasion of his 60th birthday.

Abstract. In this paper we propose extensions to trust-region algorithms in which the classical step is augmented with a second step that we insist yields a decrease in the value of the objective function. The classical convergence theory for trust-region algorithms is adapted to this class of two-step algorithms.

The algorithms can be applied to any problem with variable(s) whose contribution to the objective function is a known functional form. In the nonlinear programming package LANCELOT, they have been applied to update slack variables and variables introduced to solve minimax problems, leading to enhanced optimization efficiency. Extensive numerical results are presented to show the effectiveness of these techniques.

Key words. trust regions, line searches, two-step algorithms, spacer steps, slack variables, LANCELOT, minimax problems, expensive function evaluations, circuit optimization

AMS subject classifications. 49M37, 90C06, 90C30

PII. S1052623498334396

1. Introduction.

In nonlinear optimization problems with expensive function and gradient evaluations, it is desirable to extract as much improvement as possible at each iteration of an algorithm. When the objective function contains a subset of variables that occurs in a predictable functional form, a second, computationally relatively inexpensive, update can be applied to these variables following a classical optimization step. The additional step provides a further reduction in the objective function and can lead to superior optimization efficiency. The two-step algorithms have been successfully applied to the updating of slack variables and to a particular formulation of minimax problems, as is indicated by numerical results on a variety of problems. In these instances a subset of variables (slack variables and variables introduced to solve minimax problems) appears in a fixed, known algebraic form in the objective function. However, since it can be applied to any problem where a subset of the variables can be optimized relatively cheaply compared with the cost of evaluating the entire function (for example if some terms require simulation and other independent terms are available analytically), their applicability is really rather broad. We propose modifications to existing nonlinear optimization algorithms. An alternative approach, when feasible, is to reformulate the original problem by eliminating a subset of variables and then to apply the algorithms in the remaining variables (see, for example, Golub and Pereyra [17]).

*Received by the editors February 15, 1998; accepted for publication (in revised form) June 9, 1999; published electronically September 24, 1999.

http://www.siam.org/journals/siopt/9-4/33439.html

†Department of Mathematical Sciences, IBM T.J. Watson Research Center, Route 134 and Taconic, Room 33-206, Yorktown Heights, NY 10598 (arconn@watson.ibm.com).

‡Departamento de Matemática, Universidade de Coimbra, 3000 Coimbra, Portugal (lvicente@mat.uc.pt). This work was begun when this author was visiting the IBM T.J. Watson Research Center at Yorktown Heights and was supported in part by Centro de Matemática da Universidade de Coimbra, Instituto de Telecomunicações, FCT, Praxis XXI 2/2.1/MAT/346/94, and IBM Portugal.

§Computer Architecture and Design Automation, IBM T.J. Watson Research Center, Route 134 and Taconic, Room 33-156, Yorktown Heights, NY 10598.

This paper deals with two-step algorithms where the second step is required to yield a decrease in the value of the objective function. The analysis given here covers the global convergence of two-step trust-region algorithms and it is presented for the unconstrained minimization problem

$$\text{minimize} \quad f(y), \tag{1.1}$$

where $y \in \mathbb{R}^p$ and $f : \mathbb{R}^p \longrightarrow \mathbb{R}$ is a twice continuously differentiable function. For both trust regions and line searches, one can consider two versions of the two-step algorithms, one called greedy and the other called conservative. The greedy version exploits as much as possible the decrease obtained by the second step, whereas the conservative approach calculates the second step only after the first step has been confirmed to satisfy the traditional criteria required for global convergence. We point out that the conservative two-step line-search algorithm is not new and can be found in the books by Bertsekas [1, section 1.3.1] and Luenberger [19, section 7.10], where the second step is called a spacer step. A description of the greedy and conservative two-step line-search algorithms can be found in [11].

In trust regions, if the second step is guaranteed to decrease the value of the objective function, global convergence of the type $\liminf_{k \longrightarrow +\infty} \|\nabla f(y_k)\| = 0$ is immediately attained. Further, in the cases where the first step would be rejected, the sum of the first and second steps has a better chance of being accepted (see Remark 3.1). To obtain $\lim_{k \longrightarrow +\infty} \|\nabla f(y_k)\| = 0$, either the norm of the second step has to be controlled by the trust region (see condition (3.9)) or the decrease on the objective function attained by the second step has to be of the order of magnitude of the norm of this step (see condition (3.8)).

The update of the slack variables referred to above motivated the study of the local rate of convergence of a two-step Newton's method. We show that a second Newton step in some of the variables retains the q-quadratic rate of convergence of the traditional Newton's method.

This paper is structured as follows. In section 2 we introduce the two-step trust-region algorithms, and in section 3 we analyze their global convergence properties. The local rate of the two-step Newton's method is studied in section 4. The application of the two-step ideas to update slack variables and variables introduced for the solution of minimax problems is described in section 5. Section 6 presents the numerical results obtained with LANCELOT using these updates for analytic problems and dynamic-simulation-based and analytic static-timing-based circuit optimization problems. Finally, some conclusions are drawn in section 7.

2. Two-step trust-region algorithms.

We first consider the trust-region framework presented in the paper by Moré [20] for unconstrained minimization. The (classical) trust-region algorithm builds a quadratic model of the form

$$m_k(y_k + s) \;=\; f(y_k) + \nabla f(y_k)^T s + \frac{1}{2} s^T H_k s$$

at the current point y_k, where H_k is an approximation to $\nabla^2 f(y_k)$ (note that $m_k(y_k) = f(y_k)$). Then a step s_k is computed by approximately solving the trust-region subproblem

$$\begin{array}{rl} \text{minimize} & m_k(y_k + s) \\ \text{subject to} & \|s\| \leq \Delta_k, \end{array} \tag{2.1}$$

where Δ_k is called the trust-region radius and $\|\cdot\|$ is an arbitrary norm. The new point $y_{k+1} = y_k + s_k$ is tested for acceptance. If the actual reduction $f(y_k) - f(y_k + s_k)$ is larger than a given fraction of the predicted reduction $m_k(y_k) - m_k(y_k + s_k)$, then the step s_k and the new point y_{k+1} are accepted. In this situation, the quadratic model $m_k(y_k + s)$ is considered to be a good approximation to the function $f(y)$ in the region $\|y_k - y\| \leq \Delta_k$. The trust radius may be increased. Otherwise, the quadratic model $m_k(y_k + s)$ is considered not to be a good approximation to the function $f(y)$ in the region $\|y - y_k\| \leq \Delta_k$. In this case, the new point y_{k+1} is rejected, and a new trust-region subproblem of the form (2.1) is solved for a smaller value of the trust radius. This simple trust-region algorithm is described below.

Algorithm 2.1 (trust-region algorithm).

1. Given y_0, the value $f(y_0)$, the gradient $\nabla f(y_0)$ and an approximation H_0 to the Hessian of f at y_0, and the initial trust-region radius Δ_0. Set $k = 0$. Choose γ and α in $(0, 1)$.
2. Compute a step s_k based on the trust-region problem (2.1).
3. Compute

$$\rho_k \;=\; \frac{f(y_k) - f(y_k + s_k)}{m_k(y_k) - m_k(y_k + s_k)}.$$

4. In the case where

$$\rho_k > \alpha,$$

set

$$y_{k+1} = y_k + s_k,$$

compute H_{k+1}, and select Δ_{k+1} satisfying $\Delta_{k+1} \;\geq\; \Delta_k$.
Otherwise, set

$$y_{k+1} = y_k\;, \qquad H_{k+1} = H_k, \qquad \text{and} \qquad \Delta_{k+1} = \gamma\Delta_k.$$

5. Increment k by one and go to step 2.

The mechanism used to update the trust radius that is described in Algorithm 2.1 is simple and suffices to prove convergence results. In practice, with the goal of improving optimization efficiency, one uses updating schemes that are more complex, involving several subcases according to the value of ρ_k.

We propose in this paper a modification of this trust-region algorithm. We are motivated by a situation where it is desirable to update slack variables and variables introduced to solve minimax problems, at every iteration of the trust-region algorithm [7] implemented in LANCELOT [9]. See section 5 for more details on practical applications.

The two-step trust-region algorithm is quite easy to describe. Suppose that after computing a step $\bar{s}_k$ based on the trust-region subproblem (2.1) we know some properties of the function $f(y)$ that enable us to compute a new step $\hat{s}_k$ for which we can guarantee that $f(y_k + \bar{s}_k + \hat{s}_k) < f(y_k + \bar{s}_k)$. In this situation we would certainly like to have $y_{k+1} = y_k + \bar{s}_k + \hat{s}_k$ and to test whether this new point should be accepted. This modification requires a careful redefinition of the actual and predicted reductions given for Algorithm 2.1. The new actual and predicted reductions that we propose are

$$(2.2)\ \ ared(y_k, \bar{s}_k, \hat{s}_k) \ = \ f(y_k) - f(y_k + \bar{s}_k + \hat{s}_k),$$

$$(2.3)\ \ pred(y_k, \bar{s}_k, \hat{s}_k) \ = \ m_k(y_k) - m_k(y_k + \bar{s}_k) + f(y_k + \bar{s}_k) - f(y_k + \bar{s}_k + \hat{s}_k).$$

The new predicted reduction is the predicted reduction obtained by the first step plus the (actual) reduction obtained by the second step. The choice $pred(y_k, \bar{s}_k, \hat{s}_k) = m_k(y_k) - m_k(y_k + \bar{s}_k + \hat{s}_k)$ is not appropriate since the second step $\hat{s}_k$ is not computed using the model $m_k(y_k + s)$.

The two-step trust-region algorithm is given below.

Algorithm 2.2 (two-step trust-region algorithm, greedy).

1. Same as in Algorithm 2.1.
2. Compute a step $\bar{s}_k$ based on the trust-region problem (2.1).
3. If possible, find another step $\hat{s}_k$ such that

$$f(y_k + \bar{s}_k + \hat{s}_k) \ < \ f(y_k + \bar{s}_k).$$

   Otherwise, set $\hat{s}_k = 0$.

4. Compute

$$\hat{\rho}_k \ = \ \frac{ared(y_k, \bar{s}_k, \hat{s}_k)}{pred(y_k, \bar{s}_k, \hat{s}_k)}.$$

5. In the case where

$$\hat{\rho}_k > \alpha,$$

   set

$$y_{k+1} = y_k + \bar{s}_k + \hat{s}_k,$$

   compute H_{k+1}, and select Δ_{k+1} satisfying $\Delta_{k+1} \ \geq \ \Delta_k$.
   Otherwise, set

$$y_{k+1} = y_k\,, \qquad H_{k+1} = H_k, \qquad \text{and} \qquad \Delta_{k+1} = \gamma\Delta_k.$$

6. Increment k by one and go to step 2.

The two-step trust-region Algorithm 2.2 evaluates the new point $y_k + \bar{s}_k + \hat{s}_k$ for acceptance after both steps $\bar{s}_k$ and $\hat{s}_k$ have been computed. We call this version "greedy" because it tries to take as much advantage as possible of the decrease obtained by the second step $\hat{s}_k$. Note that although the function f is evaluated twice in Algorithm 2.2, the reevaluation is often computationally inexpensive. The context in which we are particularly interested involves relatively expensive evaluations at $y_k + \bar{s}_k$ and evaluations at $y_k + \bar{s}_k + \hat{s}_k$ involving only a subset of the variables that are cheap to compute (see section 5).

We could also consider a two-step trust-region algorithm, where first an acceptable step $\bar{s}_k$ is determined and only afterward a second step $\hat{s}_k$ is computed. This algorithm is outlined below.

Algorithm 2.3 (two-step trust-region algorithm, conservative).

1. Same as in Algorithm 2.1.
2. Repeat
   (a) Compute a step $\bar{s}_k$ based on the trust-region problem (2.1).

```
        (b) Compute
                $\rho_k \;=\; \frac{f(y_k) - f(y_k + \bar{s}_k)}{m_k(y_k) - m_k(y_k + \bar{s}_k)}.$
        (c) If $\rho_k > \alpha$, then set
                $\bar{y}_k = y_k + \bar{s}_k,$
            compute $\Delta_{k+1}$ satisfying $\Delta_{k+1} \;\geq\; \Delta_k$, and set accepted = true.
            If $\rho_k \leq \alpha$, set $\Delta_k = \gamma \Delta_k$ and accepted = false.
      Until accepted.
  3. If possible, find another step $\hat{s}_k$ such that
                $f(\bar{y}_k + \hat{s}_k) \;<\; f(\bar{y}_k).$
      Otherwise, set $\hat{s}_k = 0$.
  4. Set $y_{k+1} = \bar{y}_k + \hat{s}_k$.
  5. Update $H_k$. Increment $k$ by one and go to step 2.
```

The same comments about the function evaluations apply to Algorithm 2.3 after the computation of a successful step $\bar{s}_k$. However, in the case of Algorithm 2.3, the function f has to be evaluated twice only in iterations corresponding to successful first steps $\bar{s}_k$.

3. Global convergence of the two-step trust-region algorithms. We analyze first the two-step trust-region Algorithm 2.2, i.e., the greedy version. The analysis for the conservative Algorithm 2.3 is similar.

In this section we make the assumption that $\{H_k\}$ is a bounded sequence. So, there exists a $\sigma > 0$ for which

$$\|H_k\| \;\leq\; \sigma \quad \text{for all } k. \tag{3.1}$$

We require the step $\bar{s}_k$ to satisfy a fraction of Cauchy decrease on the trust-region problem (2.1). In other words, we ask $\bar{s}_k$ to satisfy

$$f(y_k) - m_k(y_k + \bar{s}_k) \;\geq\; \beta\,(m_k(y_k) - m_k(y_k + c_k)) \tag{3.2}$$

for $\beta \in (0, 1]$. The step c_k is called the Cauchy step, and it is defined as the solution of the scalar problem in the unknown η

$$\begin{aligned} \text{minimize} \quad & m_k(y_k + s) \\ \text{subject to} \quad & \|s\| \leq \Delta_k\,, \\ & s \;=\; \eta \nabla f(y_k)\,,\; \eta \in \mathbb{R}. \end{aligned}$$

There is a variety of algorithms that compute steps satisfying this condition (see [3], [22], [23], [25], and [26]).

PROPOSITION 3.1. *If $\bar{s}_k$ satisfies a fraction of Cauchy decrease, then*

$$f(y_k) - m_k(y_k + \bar{s}_k) \;\geq\; \frac{\beta}{2} \|\nabla f(y_k)\| \,\min\left\{\Delta_k, \frac{\|\nabla f(y_k)\|}{\sigma}\right\}, \tag{3.3}$$

where β and σ are as in (3.2) *and* (3.1)*, respectively.*

Proof. For the proof see Powell [24, Theorem 4] or Moré [20, Lemma 4.8]. □

If we use this proposition and the fact that $f(y_k+\bar{s}_k) > f(y_k+\bar{s}_k+\hat{s}_k)$, we obtain

$$\begin{aligned} pred(y_k, \bar{s}_k, \hat{s}_k) &= f(y_k) - m_k(y_k + \bar{s}_k) + f(y_k + \bar{s}_k) - f(y_k + \bar{s}_k + \hat{s}_k) \\ &\geq \frac{\beta}{2}\|\nabla f(y_k)\| \min\left\{\Delta_k, \frac{\|\nabla f(y_k)\|}{\sigma}\right\} + f(y_k + \bar{s}_k) - f(y_k + \bar{s}_k + \hat{s}_k) \\ &\geq \frac{\beta}{2}\|\nabla f(y_k)\| \min\left\{\Delta_k, \frac{\|\nabla f(y_k)\|}{\sigma}\right\}. \end{aligned} \tag{3.4}$$

This inequality is crucial to prove global convergence of the two-step algorithm. In particular, if the iteration k is successful, then

$$\begin{aligned} ared(y_k, \bar{s}_k, \hat{s}_k) &= f(y_k) - f(y_k + \bar{s}_k + \hat{s}_k) \\ &\geq \tfrac{\alpha\beta}{2}\|\nabla f(y_k)\| \min\left\{\Delta_k, \tfrac{\|\nabla f(y_k)\|}{\sigma}\right\}. \end{aligned} \tag{3.5}$$

We are ready to prove the first convergence result.

THEOREM 3.1. *Consider a sequence $\{y_k\}$ generated by Algorithm* 2.2 *where $\bar{s}_k$ satisfies* (3.2). *If f is continuously differentiable and bounded below on*

$$\mathcal{L}(y_0) \;=\; \{y : f(y) \leq f(y_0)\},$$

and $\{H_k\}$ is a bounded sequence, then

$$\liminf_{k \longrightarrow +\infty} \;\|\nabla f(y_k)\| \;=\; 0. \tag{3.6}$$

So, if the sequence $\{y_k\}$ is bounded, there exists at least one limit point y_* for which $\nabla f(y_*) = 0$.

Proof. The proof is similar to the proof given in [20, Theorem 4.10].

Assume by contradiction that $\{\|\nabla f(y_k)\|\}$ is bounded away from zero, i.e., that there exists an $\epsilon > 0$ such that $\|\nabla f(y_k)\| \geq \epsilon$ for all k. As in [20, Theorem 4.10], we make direct use of (3.5) and of the rules that update the trust radius, to obtain

$$\sum_{k=0}^{+\infty} \Delta_k \;<\; +\infty,$$

and so $\lim_{k \longrightarrow +\infty} \Delta_k = 0$.

The next step is to show that $\lim_{k \longrightarrow +\infty} |\hat{\rho}_k - 1| = 0$. Note that from the definitions (2.2) and (2.3), we have

$$\begin{aligned} ared(y_k, \bar{s}_k, \hat{s}_k) &- pred(y_k, \bar{s}_k, \hat{s}_k) \\ &= f(y_k) - f(y_k + \bar{s}_k) + \nabla f(y_k)^T \bar{s}_k + \tfrac{1}{2}\bar{s}_k^T H_k \bar{s}_k, \end{aligned} \tag{3.7}$$

which in turn, by using a Taylor series expansion and $\|\bar{s}_k\| \leq \Delta_k$, implies

$$|ared(y_k, \bar{s}_k, \hat{s}_k) - pred(y_k, \bar{s}_k, \hat{s}_k)| \;\leq\; o(\Delta_k).$$

This inequality and (3.4) show that $|\hat{\rho}_k - 1|$ converges to zero. The rest of the proof follows a classical argument in trust regions: if $\hat{\rho}_k$ converges to one, the rules that update the trust radius show that Δ_k cannot converge to zero. So, a contradiction is attained and the proof is completed. □

The result of Theorem 3.1 does not require the step $\hat{s}_k$ to be $\mathcal{O}(\Delta_k)$, which may seem surprising. This result shows the appropriateness of the definitions given in (2.2) and (2.3) for the actual and predicted reductions. These definitions allow us to obtain the conditions (3.5) and (3.7) that are crucial to establish (3.6).

Remark 3.1. It is also important to note that the definitions (2.2) and (2.3) can improve the acceptability of a step. In fact, we have

$$\hat{\rho}_k \;=\; \frac{t_k + \rho_k}{t_k + 1} \;\equiv\; \hat{\rho}_k(t_k),$$

where $t_k = \frac{f(y_k+\bar{s}_k)-f(y_k+\bar{s}_k+\hat{s}_k)}{m_k(y_k)-m_k(y_k+\bar{s}_k)}$ and $\rho_k \;=\; \frac{f(y_k)-f(y_k+\bar{s}_k)}{m_k(y_k)-m_k(y_k+\bar{s}_k)}$, as before. We now note that $\hat{\rho}_k(0) = \rho_k$ and the function $\hat{\rho}_k(t_k)$ is strictly increasing if $\rho_k < 1$. In other words, in cases where a standard trust-region algorithm rejects a step the modified criterion is always better than the usual one. Further, it can be noted that $\hat{\rho}_k - 1 = \frac{\rho_k - 1}{t_k+1}$, which indicates that all successful iterations of the standard algorithm will also be successful in the modified two-step algorithm. In particular, $\hat{\rho}_k > 1$ whenever $\rho_k > 1$.

The next step in the analysis is to prove that, with additional conditions on the second step, $\lim_{k \longrightarrow +\infty} \; \|\nabla f(y_k)\| = 0$.

THEOREM 3.2. *Consider a sequence* $\{y_k\}$ *generated by Algorithm* 2.2 *where* $\bar{s}_k$ *satisfies* (3.2). *Assume that* f *is continuously differentiable and bounded below on* $\mathcal{L}(y_0)$ *and that* $\{H_k\}$ *is a bounded sequence. If* ∇f *is uniformly continuous on* $\mathcal{L}(y_0)$ *and if either*

$$f(y_k + \bar{s}_k) - f(y_k + \bar{s}_k + \hat{s}_k) \;\geq\; c_1 \|\hat{s}_k\| \tag{3.8}$$

or

$$\|\hat{s}_k\| \leq c_2 \Delta_k, \tag{3.9}$$

where c_1 *and* c_2 *are positive constants independent of* k*, then*

$$\lim_{k \longrightarrow +\infty} \; \|\nabla f(y_k)\| \;=\; 0. \tag{3.10}$$

So, if the sequence $\{y_k\}$ *is bounded, every limit point* y_* *satisfies* $\nabla f(y_*) = 0$.

Proof. The proof is similar to the proof given in [20, Theorem 4.14]. See also Thomas [27].

We show the result by contradiction. Assume, therefore, that there exists an $\epsilon_1 \in (0,1)$ and a subsequence indexed by $\{m_i\}$ of successful iterates such that, for all m_i in this subsequence, $\|\nabla f(y_{m_i})\| \geq \epsilon_1$. Theorem 3.1 guarantees the existence of another subsequence indexed by $\{l_i\}$ such that $\|\nabla f(y_k)\| \geq \epsilon_2$, for $m_i \leq k < l_i$ and $\|\nabla f(y_{l_i})\| < \epsilon_2$ (where $\{m_i\}$ is, without loss of generality, the subsequence previously mentioned). Here ϵ_2 is any real number chosen to be in $(0, \epsilon_1)$. Since $\{f(y_k) - f(y_{k+1})\}$ converges to zero, for k sufficiently large corresponding to successful iterations $m_i \leq k < l_i$

$$f(y_k) - f(y_{k+1}) \;\geq\; \kappa_1 \, \Delta_k + c_1 \|\hat{s}_k\| \tag{3.11}$$

holds if (3.8) is satisfied, and

$$f(y_k) - f(y_{k+1}) \;\geq\; \kappa_1 \, \Delta_k \tag{3.12}$$

holds otherwise with $\kappa_1 = \frac{\alpha \beta \epsilon_2}{2}$.

We consider the cases (3.8) and (3.9) separately. In both cases we make use of

$$\|y_{m_i} - y_{l_i}\| \le \sum_{k=m_i}^{l_i-1} \|y_k - y_{k+1}\|,$$

$$f(y_{m_i}) - f(y_{l_i}) = \sum_{k=m_i}^{l_i-1} [f(y_k) - f(y_{k+1})].$$

In the sums $\sum_{k=m_i}^{l_i-1}$ we consider only indices corresponding to successful iterations.

If (3.8) holds, then we use (3.11) to obtain

$$\begin{aligned}\textstyle\sum_{k=m_i}^{l_i-1}[f(y_k) - f(y_{k+1})] &\ge \textstyle\sum_{k=m_i}^{l_i-1}[\kappa_1\Delta_k + c_1\|\hat{s}_k\|] \\ &\ge \min\{\kappa_1, c_1\}\textstyle\sum_{k=m_i}^{l_i-1}[\|\bar{s}_k\| + \|\hat{s}_k\|] \\ &\ge \min\{\kappa_1, c_1\}\textstyle\sum_{k=m_i}^{l_i-1}\|y_k - y_{k+1}\|.\end{aligned}$$

If (3.9) holds, then we appeal to (3.12) and write

$$\begin{aligned}\textstyle\sum_{k=m_i}^{l_i-1}[f(y_k) - f(y_{k+1})] &\ge \textstyle\sum_{k=m_i}^{l_i-1}\kappa_1\Delta_k \\ &\ge \tfrac{\kappa_1}{2}\min\{1, \tfrac{1}{c_2}\}\textstyle\sum_{k=m_i}^{l_i-1}[\|\bar{s}_k\| + \|\hat{s}_k\|] \\ &\ge \tfrac{\kappa_1}{2}\min\{1, \tfrac{1}{c_2}\}\textstyle\sum_{k=m_i}^{l_i-1}\|y_k - y_{k+1}\|.\end{aligned}$$

In either case we obtain

$$\|y_{m_i} - y_{l_i}\| \;\le\; \kappa_2\left(f(y_{m_i}) - f(y_{l_i})\right),$$

and since the right-hand side of this inequality goes to zero, so does the left-hand side $\|y_{m_i} - y_{l_i}\|$. Since the gradient of f is uniformly continuous, we have for i sufficiently large that

$$\epsilon_1 \;\le\; \|\nabla f(y_{m_i})\| \;\le\; \|\nabla f(y_{m_i}) - \nabla f(y_{l_i})\| + \|\nabla f(y_{l_i})\| \;\le\; 2\epsilon_2.$$

Since ϵ_2 can be any number in $(0, \epsilon_1)$ this inequality contradicts the supposition. □

In the theorem above we required the norm of the step $\hat{s}_k$ to be either $\mathcal{O}(\Delta_k)$ or $\mathcal{O}\left(f(y_k + \bar{s}_k) - f(y_k + \bar{s}_k + \hat{s}_k)\right)$. The former condition can be enforced in step 2 of Algorithm 2.2, although this might not be beneficial and could lead to an inferior decrease.

We can obtain global convergence to a point that also satisfies the necessary second-order conditions for optimality. For this purpose, we require the step $\bar{s}_k$ to satisfy a fraction of optimal decrease for the trust-region problem (2.1). In other words, we ask $\bar{s}_k$ to satisfy

$$f(y_k) - m_k(y_k + \bar{s}_k) \;\ge\; \beta\left(f(y_k) - m_k(y_k + s_k^*)\right), \tag{3.13}$$

where $\beta \in (0, 1]$, and s_k^* is an optimal solution of (2.1). (This condition can be weakened in several ways [20].) A step $\bar{s}_k$ satisfying a fraction of optimal decrease can be computed by using the algorithms proposed in [22] and [25] in the case where the trust-region norm is Euclidean. The global convergence result is the following.

THEOREM 3.3. *Consider a sequence $\{y_k\}$ generated by Algorithm 2.2, where $H_k = \nabla^2 f(y_k)$ and $\bar{s}_k$ satisfies (3.13). If $\mathcal{L}(y_0)$ is compact and f is twice continuously differentiable on $\mathcal{L}(y_0)$, then there exists at least one limit point y_* for which $\nabla f(y_*) = 0$ and $\nabla^2 f(y_*)$ is positive semidefinite.*

Proof. The proof is basically the same as the proof of Theorem 4.7 in [22]. □

To obtain stronger global convergence results to second-order points, for instance, the results in Theorems 4.11 and 4.13 in [22] (see also [21, Theorem 4.17, c and d]), other conditions are required, such as $\|\hat{s}_k\|$ being of $\mathcal{O}(\Delta_k)$.

The next results show that the second step can preserve the nice local properties of the behavior of the trust radius that are typical in trust-region algorithms.

THEOREM 3.4. *Let $\{y_k\}$ be a sequence generated by Algorithm 2.2 where $\bar{s}_k$ satisfies (3.2) and $H_k = \nabla^2 f(y_k)$. In addition, assume that the step $\hat{s}_k$ satisfies either condition (3.8) or condition (3.9). If f is twice continuously differentiable and bounded below on $\mathcal{L}(y_0)$ and $\{y_k\}$ has a limit point y_* such that $H_* = \nabla^2 f(y_*)$ is positive definite, then $\{y_k\}$ converges to y_*, all iterations are eventually successful, and $\{\Delta_k\}$ is bounded away from zero.*

Proof. From Theorem 3.2 we can guarantee that $\lim_{k \longrightarrow +\infty} \|\nabla f(y_k)\| = 0$. Therefore, the proof is basically the same as the proof of Theorem 4.19 in [20]. □

An alternative to this result, where we do not impose conditions (3.8) or (3.9) on the second step, is given below. However, we need to assume that $\{y_k\}$ converges to y_*.

THEOREM 3.5. *Let $\{y_k\}$ be a sequence generated by Algorithm 2.2, where $\bar{s}_k$ satisfies (3.2) and $H_k = \nabla^2 f(y_k)$. If f is twice continuously differentiable on $\mathcal{L}(y_0)$ and $\{y_k\}$ converges to a point y_* such that $H_* = \nabla^2 f(y_*)$ is positive definite, then all iterations are eventually successful and $\{\Delta_k\}$ is bounded away from zero.*

Proof. The first step $\bar{s}_k$ yields a decrease in the quadratic model:

$$m_k(y_k) - m_k(y_k + \bar{s}_k) \;=\; -\nabla f(y_k)^T \bar{s}_k - \frac{1}{2}\bar{s}_k^T H_k \bar{s}_k \;\geq\; 0.$$

Thus, the assumptions made on H_k and H_* guarantee

$$\|\bar{s}_k\| \;\leq\; c_3 \|\nabla f(y_k)\| \tag{3.14}$$

for sufficiently large k, which in turn, by using (3.4), implies

$$pred(y_k, \bar{s}_k, \hat{s}_k) \;\geq\; c_4 \|\bar{s}_k\|^2. \tag{3.15}$$

(The constants c_3 and c_4 are independent of k.)

A Taylor series expansion for the expression (3.7) gives

$$|ared(y_k, \bar{s}_k, \hat{s}_k) - pred(y_k, \bar{s}_k, \hat{s}_k)| \;\leq\; o(\|\bar{s}_k\|^2)\,. \tag{3.16}$$

The fact that $\{y_k\}$ converges and the result $\liminf_{k \longrightarrow +\infty} \|\nabla f(y_k)\| = 0$ of Theorem 3.1 together imply $\lim_{k \longrightarrow +\infty} \|\nabla f(y_k)\| = 0$. Thus, from (3.14) we get $\lim_{k \longrightarrow +\infty} \|\bar{s}_k\| = 0$.

The proof is terminated with a typical argument in trust regions. From (3.15), (3.16), and $\lim_{k \longrightarrow +\infty} \|\bar{s}_k\| = 0$, we obtain the limit

$$\lim_{k \longrightarrow +\infty} \left| \frac{ared(y_k, \bar{s}_k, \hat{s}_k)}{pred(y_k, \bar{s}_k, \hat{s}_k)} - 1 \right| \;=\; 0,$$

which shows, by appealing to the rules that update the trust radius, that all iterations are eventually successful and the trust radius is uniformly bounded away from zero. □

The global convergence analysis for Algorithm 2.3 is identical to the analysis given above for Algorithm 2.2. We point out that Algorithm 2.3 is well defined since at a nonstationary point it is always possible to find an acceptable first step. Also, for every k,

$$\begin{aligned} f(y_k) - f(y_{k+1}) &= f(y_k) - f(y_k + \bar{s}_k) + f(y_k + \bar{s}_k) - f(y_{k+1}) \\ &\geq \frac{\alpha\beta}{2}\|\nabla f(y_k)\| \min\left\{\Delta_k, \frac{\|\nabla f(y_k)\|}{\sigma}\right\} + f(y_k + \bar{s}_k) - f(y_{k+1}) \\ &\geq \frac{\alpha\beta}{2}\|\nabla f(y_k)\| \min\left\{\Delta_k, \frac{\|\nabla f(y_k)\|}{\sigma}\right\}. \end{aligned}$$

Thus, the results given in Theorems 3.1–3.5 hold for Algorithm 2.3. The lim inf–type result (3.6) is obtained under the classical assumptions for trust-region algorithms for unconstrained optimization. To obtain the lim-type result (3.10) one of the two conditions (3.8) and (3.9) is required.

In the case of the applications considered in section 5, the decrease obtained by the second step $\hat{s}_k$ is always guaranteed to satisfy

$$f(y_k + \bar{s}_k) - f(y_k + \bar{s}_k + \hat{s}_k) \;\geq\; c_5\|\hat{s}_k\|^2. \tag{3.17}$$

Moreover, the objective function strictly decreases along the segment between the points $y_k + \bar{s}_k$ and $y_k + \bar{s}_k + \hat{s}_k$. In this case we can modify step 3 of Algorithms 2.2 and 2.3 in such a way that we meet the requirements of Theorem 3.2. This modification is given below. It is easy to verify that $\hat{s}_k \neq 0$ satisfies $f(y_k + \bar{s}_k + \hat{s}_k) < f(y_k + \bar{s}_k)$ and either (3.8) or (3.9).

Algorithm 3.1 (step 3 for Algorithms 2.2 and 2.3, quadratic decrease case).

3. Compute a step $\hat{s}_k$ such that

$$f(y_k + \bar{s}_k) - f(y_k + \bar{s}_k + \hat{s}_k) \;\geq\; c_5\|\hat{s}_k\|^2\,.$$

If $\|\hat{s}_k\| < \nu$, then scale $\hat{s}_k$ by $\min\{1, \frac{c_2\Delta_k}{\nu}\}$ so that $\|\hat{s}_k\| \leq c_2\Delta_k$ and $\hat{s}_k$ is not enlarged.
(Otherwise (3.8) holds with $c_1 = \nu c_5$.)

The positive parameters ν and c_2 should be set a priori in step 1 of Algorithms 2.2 and 2.3.

Of course, we would like to prove the result of Theorem 3.2 for the case where the condition (3.8) is replaced by the condition (3.17). However, such a result is unlikely to be true.

4. Local rate of convergence of a two-step Newton's method.

In the next section we are interested in two-step algorithms where the second step is calculated as a Newton-type step in some of the variables. In this section we investigate the local rate of convergence for an algorithm where each step is composed of two Newton steps, the second being computed only for a subset of the variables. For this purpose let

$$y \;=\; \begin{pmatrix} x \\ u \end{pmatrix}.$$

Suppose the first step $\bar{s}_k$ is a full Newton step, i.e., $\bar{s}_k = -\nabla^2 f(y_k)^{-1} \nabla f(y_k)$. Also, let

$$\bar{y}_k = \begin{pmatrix} \bar{x}_k \\ \bar{u}_k \end{pmatrix} = y_k + \bar{s}_k.$$

At the intermediate point $\bar{y}_k$, a Newton step is applied in the variables u with $x = \bar{x}_k$ fixed. This two-step Newton's method is described below.

ALGORITHM 4.1 (two-step Newton's method).

```
1. Choose $y_0$.
2. For $k = 1, 2, \ldots$ do
    2.1 Compute $\bar{s}_k = -\nabla^2 f(y_k)^{-1} \nabla f(y_k)$ and set $\bar{y}_k = y_k + \bar{s}_k$.
    2.2 Compute $\hat{s}_k = ( \begin{matrix} 0 \\ -\nabla^2_{uu} f(\bar{y}_k)^{-1} \nabla_u f(\bar{y}_k) \end{matrix} )$ and set $s_k = \bar{s}_k + \hat{s}_k$.
    2.3 Set $y_{k+1} = y_k + s_k$.
```

The proof of the local convergence rate of the two-step Newton's method requires a few modifications from the standard proof of Newton's method [12, Theorem 5.2.1]. Recall that that proof of Newton's method is by induction.

COROLLARY 4.1. *Let f be twice continuously differentiable in an open set D where the second partial derivatives are Lipschitz continuous. If $\{y_k\}$ is a sequence generated by Algorithm* 4.1 *converging to a point $y_* \in D$ for which $\nabla f(y_*) = 0$ and $\nabla^2 f(y_*)$ is positive definite, then $\{y_k\}$ converges with a q-quadratic rate.*

Proof. If y_k is sufficiently close to y_*, the perturbation result [12, Theorem 3.1.4] can be used to prove the nonsingularity of the Hessian matrix $\nabla^2 f(y_k)$. Furthermore,

$$\|\bar{y}_k - y_*\| \leq c_6 \|y_k - y_*\|^2. \tag{4.1}$$

Now we show that $\nabla^2_{uu} f(\bar{y}_k)$ is also nonsingular. First we point out that $\nabla^2_{uu} f(y)$ is Lipschitz continuous on D and $\nabla^2_{uu} f(y_*)$ is positive definite. Thus, inequality (4.1) and the perturbation lemma cited above together imply the nonsingularity of $\nabla^2_{uu} f(\bar{y}_k)$. Hence the method is locally well defined, and the second step yields

$$\|y_{k+1} - \bar{y}_k\| = \|\hat{s}_k\| = \|\nabla^2_{uu} f(\bar{y}_k)^{-1} \left(\nabla_u f(\bar{y}_k) - \nabla_u f(y_*) \right) \| \leq c_7 \|\bar{y}_k - y_*\|, \tag{4.2}$$

since $\nabla_u f(y)$ is Lipschitz continuous near y_*. Now we use inequalities (4.1) and (4.2) and write

$$\begin{aligned} \|y_{k+1} - y_*\| &\leq \|y_{k+1} - \bar{y}_k\| + \|\bar{y}_k - y_*\| \\ &\leq (c_7 + 1) \|\bar{y}_k - y_*\| \\ &\leq c_6 (c_7 + 1) \|y_k - y_*\|^2. \end{aligned}$$

This last inequality establishes the q-quadratic rate of convergence. □

5. Applications. We begin by considering updating the slack variables in LANCELOT. Suppose the problem we are trying to solve has the form

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && c_i(x) \geq 0\,, \quad i = 1, \ldots, m, \end{aligned} \tag{5.1}$$

where $x \in \mathbb{R}^n$ and n and m are positive integers. The technique implemented in the LANCELOT package [9] is the augmented Lagrangian algorithm proposed by Conn, Gould, and Toint in [8]. For the application of the augmented Lagrangian algorithm this problem is reformulated as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & c_i(x) - u_i = 0\,, \quad i = 1, \ldots, m, \\ & u_i \geq 0\,, \quad i = 1, \ldots, m, \end{array}$$

by adding the slack variables u_i, $i = 1, \ldots, m$. This algorithm considers the following augmented Lagrangian merit function:

$$\Phi(x, u, \lambda, S, \mu) \;=\; f(x) + \sum_{i=1}^{m} \lambda_i (c_i(x) - u_i) + \frac{1}{2\mu} \sum_{i=1}^{m} s_{ii} (c_i(x) - u_i)^2,$$

where λ_i is an estimate for the Lagrange multiplier associated with the ith constraint, μ is a (positive) penalty parameter, s_{ii} is a (positive) scaling factor that is associated with the ith constraint, and $S = [s_{ij}]$ with $s_{ij} = 0$ for $i \neq j$.

LANCELOT [7], [9] solves a sequence of minimization problems with simple bounds of the form

$$\begin{array}{ll} \text{minimize} & \Phi(x, u, \lambda, S, \mu) \\ \text{subject to} & u_i \geq 0\,, \quad i = 1, \ldots, m, \end{array} \tag{5.2}$$

for fixed values of μ, s_{ii}, and λ_i, $i = 1, \ldots, m$. The two-step trust-region framework and analysis described in this paper for unconstrained minimization problems can be extended in an entirely straightforward way to a number of algorithms for minimization problems with simple bounds, in particular to the algorithms [7] used by LANCELOT to solve problem (5.2).

If x is fixed, the function $\Phi(x, u, \lambda, S, \mu)$ is quadratic in the slack variables u. Let us denote this quadratic by $q(u;\, x)$:

$$q(u;\, x) \;=\; \Phi(x, u, \lambda, S, \mu) = d(x) + e(x)^T u + \frac{1}{2} u^T F u,$$

where $d(x)$ and $e(x)$ depend on x but F is constant. (The dependency on λ_i, s_{ii}, and μ is not important since these are constants fixed before the minimization process is started.)

The key idea is to update these slack variables at every iteration k of the trust-region algorithm [7] that is used in LANCELOT to solve problem (5.2). The trust-region algorithm computes, at the current point y_k, a first step $\bar{s}_k$. Now, at the new point $y_k + \bar{s}_k$ we compute the step $\hat{s}_k$ by updating the slack variables u. So, we have

$$y_k \;=\; \begin{pmatrix} x_k \\ u_k \end{pmatrix}, \quad \bar{s}_k \;=\; \begin{pmatrix} (\bar{s}_k)_x \\ (\bar{s}_k)_u \end{pmatrix}, \quad \hat{s}_k \;=\; \begin{pmatrix} 0 \\ \Delta u_k \end{pmatrix},$$

$$f(y_k + \bar{s}_k) \;=\; q(\bar{u}_k;\, \bar{x}_k)\,, \quad f(y_k + \bar{s}_k + \hat{s}_k) \;=\; q(\bar{u}_k + \Delta u_k;\, \bar{x}_k)\,,$$

where

$$\bar{x}_k \;=\; x_k + (\bar{s}_k)_x\,, \qquad \bar{u}_k \;=\; u_k + (\bar{s}_k)_u\,.$$

(Here f represents the objective function of sections 1–4.) Note that the second step $\hat{s}_k$ is exclusively in the components associated with slack variables. This step is computed as $u_{k+1} - \bar{u}_k$, where u_{k+1} is the optimal solution of

$$\begin{array}{rl} \text{minimize} & q(u;\, \bar{x}_k) \\ \text{subject to} & u_i \geq 0\,, \quad i = 1, \ldots, m. \end{array} \tag{5.3}$$

Due to the simple form of this quadratic, the solution is explicit:

$$(u_{k+1})_i \;=\; \max\left\{0,\, \frac{\mu\lambda_i}{s_{ii}} + c_i(\bar{x}_k)\right\}\,, \quad i = 1, \ldots, m. \tag{5.4}$$

It is important to remark that these updates require no further function or gradient evaluations. They have also been considered in the codes NPSOL [16] and SNOPT [15] to update slack variables after the application of a line search to the augmented Lagrangian merit function and prior to the solution of the next quadratic programming problem. Other ways of dealing with slack variables have been studied in the literature (see Gould [18] and the references therein).

For the study of the impact of the slack variable update on the global convergence of the trust-region algorithm, the step in these variables is required only to decrease the quadratic $q(u;\, \bar{x}_k)$ from $\bar{u}_k$ to $\bar{u}_k + \Delta u_k$. In such a case, we can always guarantee that the decrease in the objective function is larger than $\|\hat{s}_k\|^2$, that is, that (3.17) holds. This result is shown in the following proposition. We drop $\bar{x}_k$ from $q(\,\cdot\,;\, \bar{x}_k)$ to simplify the notation.

PROPOSITION 5.1. *There exists a positive constant c_5 such that, whenever $q(\bar{u}_k + \Delta u_k) < q(\bar{u}_k)$, we have*

$$q(\bar{u}_k) - q(\bar{u}_k + \Delta u_k) \;\geq\; c_5 \|\Delta u_k\|^2.$$

Proof. First we note a few properties of the quadratic $q(u)$. Simple algebraic manipulations lead to

$$q(\bar{u}_k) - q(\bar{u}_k + \Delta u_k) \;=\; -\left(F(\bar{u}_k + \Delta u_k) + e(\bar{x}_k)\right)^T \Delta u_k + \frac{1}{2} {\Delta u_k}^T F \Delta u_k. \tag{5.5}$$

Also, since $q(u)$ is convex,

$$q(\bar{u}_k) - q(\bar{u}_k + \Delta u_k) \;\geq\; \left|\nabla q(\bar{u}_k + \Delta u_k)^T \Delta u_k\right|. \tag{5.6}$$

Let c be a positive constant such that $c < \frac{\lambda_{min}(F)}{2}$, where $\lambda_{min}(F)$ is the smallest eigenvalue of F. Now we consider two cases:

1. $\left|\nabla q\left(\bar{u}_k + \Delta u_k\right)^T \Delta u_k\right| \geq c\|\Delta u_k\|^2$. In this case we use (5.6) to obtain

$$q(\bar{u}_k) - q(\bar{u}_k + \Delta u_k) \;\geq\; c\|\Delta u_k\|^2.$$

2. $\left|\nabla q\left(\bar{u}_k + \Delta u_k\right)^T \Delta u_k\right| < c\|\Delta u_k\|^2$. In this case we appeal to (5.5) and

$$\nabla q\left(\bar{u}_k + \Delta u_k\right) = F(\bar{u}_k + \Delta u_k) + e(\bar{x}_k)$$

to get

$$\begin{aligned} q(\bar{u}_k) - q(\bar{u}_k + \Delta u_k) &= -\left(F(\bar{u}_k + \Delta u_k) + e(\bar{x}_k)\right)^T \Delta u_k + \frac{1}{2} {\Delta u_k}^T F \Delta u_k \\ &\geq \left(\frac{\lambda_{min}(F)}{2} - c\right) \|\Delta u_k\|^2. \end{aligned}$$

The proof is completed by setting $c_5 = \min\{c, \frac{\lambda_{min}(F)}{2} - c\}$. □

Another example of the application of two-step algorithms arises in one approach to the solution of minimax problems. Consider the following minimax problem:

$$\min_{x} \max_{i=1,\ldots,m} f_i(x), \tag{5.7}$$

where each f_i is a real-valued function defined in $\mathbb{R}^n$. One way of solving this minimax problem is to reformulate it as a nonlinear programming problem by adding an artificial variable z. See [18] for more details. This leads to

$$\begin{array}{ll} \text{minimize} & z \\ \text{subject to} & z - f_i(x) - u_i = 0, \quad i = 1, \ldots, m, \\ & u_i \geq 0, \quad i = 1, \ldots, m, \end{array} \tag{5.8}$$

where the slack variables have also been introduced. If LANCELOT is used to solve this nonlinear programming problem, then the augmented Lagrangian algorithm requires the solution of a sequence of problems with simple bounds of the type

$$\begin{array}{ll} \text{minimize} & \Phi(x, z, u, \lambda, S, \mu) \\ \text{subject to} & u_i \geq 0, \quad i = 1, \ldots, m, \end{array} \tag{5.9}$$

where

$$\Phi(x, z, u, \lambda, S, \mu) \;=\; z + \sum_{i=1}^{m} \lambda_i (z - f_i(x) - u_i) + \frac{1}{2\mu} \sum_{i=1}^{m} s_{ii} (z - f_i(x) - u_i)^2.$$

In this situation the function $\Phi(x, z, u, \lambda, S, \mu)$ is quadratic in the variables u and z for fixed values of x. (Again, λ, S, and μ are constants and not variables for problem (5.9).) The application of the two-step trust-region algorithm follows in a similar way. The Hessian of the quadratic is positive semidefinite with the following form:

$$F \;=\; \frac{1}{\mu} \begin{pmatrix} s_{11} & 0 & \cdot & \cdot & \cdot & 0 & -s_{11} \\ 0 & \cdot & & & & & \cdot \\ \cdot & & \cdot & & & & \cdot \\ \cdot & & & \cdot & & & \cdot \\ \cdot & & & & \cdot & & \cdot \\ 0 & & & & & s_{nn} & -s_{nn} \\ -s_{11} & \cdot & \cdot & \cdot & \cdot & -s_{nn} & \sum_{i=1}^{m} s_{ii} \end{pmatrix},$$

where the last row and the last column correspond to the variable z. The solution of the quadratic program

$$\begin{array}{ll} \text{minimize} & q(z, u; \bar{x}_k) \\ \text{subject to} & u_i \geq 0\,, \quad i = 1, \ldots, m, \end{array} \tag{5.10}$$

is given by

$$(u_{k+1})_i \;=\; \max\left\{0, \frac{\mu \lambda_i}{s_{ii}} - f_i(\bar{x}_k) + z_{k+1}\right\}, \quad i = 1, \ldots, m, \tag{5.11}$$

where z_{k+1} is the solution of the equation

$$-\frac{1}{\mu}\sum_{i=1}^{m} s_{ii} \max\left\{0,\, \frac{\mu\lambda_i}{s_{ii}} - f_i(\bar{x}_k) + z\right\} \;+\; \frac{1}{\mu}\left(\sum_{i=1}^{m} s_{ii}\right) z \;=\; b \tag{5.12}$$

with right-hand side

$$b = -1 - \sum_{i=1}^{m}\left(\lambda_i - \frac{s_{ii}}{\mu} f_i(\bar{x}_k)\right). \tag{5.13}$$

Equation (5.12) is solved easily with $\mathcal{O}(m)$ floating point operations and comparisons, showing that the solution of the quadratic program (5.10) is a relatively inexpensive calculation.

There are several nonlinear optimization problems in which some subset of the problem variables occur linearly, for example, arrival times in static-timing-based circuit optimization problems [6]. Such problems can also benefit from two-step updating.

6. Numerical tests.

6.1. Analytic problems.

We modified LANCELOT (release A) [9] to include the slack variable update (5.4) and the slack and minimax variable updates (5.11)–(5.13). These updates were incorporated in LANCELOT using a greedy two-step modification of the trust-region algorithm [7] for minimization problems with simple bounds that is implemented in the subroutine SBMIN. (The greedy two-step trust-region algorithm for unconstrained minimization problems is Algorithm 2.2.) We tested the following versions of LANCELOT:

1. LANCELOT (release A) with the default parameter configuration SPEC.SPC file, except that we increased the maximum number of iterations to 4000.
2. Version 1 with the slack and minimax variable updates (5.4) and (5.11)–(5.13) incorporated in SBMIN using a greedy two-step trust-region algorithm.
3. The same as version 2 but with no update of the variable z for minimax problems, i.e., z fixed in (5.11)–(5.13).

We compared the numerical performance of these three versions on a set of problems[1] from the CUTE collection [2]. This set of problems is listed in Table 6.1 and, in the case of minimax formulations, in Table 6.2, where we mention the number of variables (including slacks and, where applicable, the minimax variable z), the number of slack variables, and the number of equality and inequality constraints (excluding simple bounds on the variables). Note that the minimax problems were reformulated as nonlinear programming problems by the introduction of an additional minimax variable z as shown above (5.8).

The computational results are presented in Tables 6.3, 6.4, and 6.5. All tests were conducted on an IBM Risc/System 6000 model 390 workstation. In Table 6.3 we compare the results of versions 1 and 2 for problems that are not minimax problems. In Table 6.4 we present the results of versions 1 and 2 for minimax problems. In Table

[1] Although CUTE contains more than 56 problems with general constraints the majority of these are equality constrained problems. We excluded all problems that took more than 4000 iterations with both versions 1 and 2. We included the rest, with the exception of some problems that are too easy, making a total of 56 problems of which 30 are minimax problems and 26 are nonminimax problems.

Table 6.1
Nonminimax problems from the CUTE collection used in testing.

Problem name	Variables	Slacks	Constraints
CAR2	209	30	146
CORE1	83	18	59
CORE2	157	26	134
CORKSCRW	106	70	10
CSFI1	7	2	4
CSFI2	7	2	4
HADAMARD	769	512	648
HS32	4	1	1
HS67	17	14	14
HS85	26	21	21
HS109	13	8	4
NET1	67	19	57
NET2	181	37	160
ORBIT2	298	30	207
PRODPL0	69	9	29
PRODPL1	69	9	29
SSEBNLN	218	24	96
SWOPF	97	14	92
TFI1F	3	101	101
TFI2F	3	101	101
TFI3F	3	101	101
VANDERM1	10	9	19
VANDERM2	10	9	19
VANDERM3	10	9	19
VANDERM4	5	4	9
ZIGZAG	74	10	50

6.5 we include the results of versions 1 and 3 for minimax problems. In Tables 6.4 and 6.5 we include the majority of the minimax problems but not all. (See section 6.3 for numerical results on the remaining problems.) In these tables we report the value of the flag `INFORM`, the number of iterations, the total CPU time, and the determined values (a single value if they are both the same) of the objective function. The values of `INFORM` have the following meanings:

`INFORM = 0` for normal return, meaning that the norm of the projected gradient of the augmented Lagrangian function has become smaller than 10^{-5}.

`INFORM = 1` for cases where the maximum number of iterations (4000) has been reached.

`INFORM = 3` for cases where the norm of the step has become too small.

Our conclusion based on these sets of problems is that the version with the slack and minimax variable updates exhibits superior numerical behavior. In fact, this version required an average of 15% fewer iterations than the version without these updates. (The problems HS109, HAIFAM, and POLAK6 were excluded from this calculation, mainly because the comparison was extraordinarily favorable in the first two cases and worse in the last.) Comparing Tables 6.4 and 6.5, updating the minimax variable z in addition to two-step updates on just the slacks is seen to yield a significant benefit. However, there are some minimax problems where the two-step algorithm performs poorly and this situation is analyzed in detail in section 6.3.

6.2. Circuit optimization problems. We have built extensive experience with circuit optimization problems, where—due to expensive function evaluations, modest numerical noise levels, and practical stopping criteria—the implementation is designed to terminate before many "asymptotic" iterations are taken. The algorithms described

TABLE 6.2
Minimax problems from the CUTE collection used in testing.

Problem name	Variables	Slacks	Constraints
CB2	6	3	3
CB3	6	3	3
CHACONN1	6	3	3
CHACONN2	6	3	3
CONGIGMZ	8	5	5
COSHFUN	81	20	20
DEMYMALO	6	3	3
GIGOMEZ1	6	3	3
GOFFIN	101	50	50
HAIFAL	9301	8958	8958
HAIFAM	249	150	150
HALDMADS	48	42	42
KIWCRESC	5	2	2
MADSEN	9	6	6
MAKELA1	5	2	2
MAKELA2	6	3	3
MAKELA3	41	20	20
MAKELA4	61	40	40
MIFFLIN1	5	2	2
MIFFLIN2	5	2	2
MINMAXBD	25	20	20
POLAK1	5	2	2
POLAK2	13	2	2
POLAK3	22	10	10
POLAK4	6	3	3
POLAK5	5	2	2
POLAK6	9	4	4
SPIRAL	5	2	2
SPRALX	5	2	2
WOMFLET	6	3	3

in this paper have been used in a dynamic-simulation-based circuit optimization tool called JiffyTune (see [4], [5], and [10]). JiffyTune optimizes transistor and wire sizes of digital integrated circuits to meet delay, power, and area goals. It is based on fast circuit simulation and time-domain sensitivity computation in SPECS (see [13] and [28]). To optimize multiple path delays through a high-performance circuit, the tuning is often formulated as a minimax problem or a minimization problem with nonlinear inequality constraints.

We remark that many of the analytic problems (especially the minimax problems) are rather small and involve inexpensive function evaluations. Moreover, it is clear that two-step updating is unlikely to be helpful asymptotically in these situations. Consequently we also report numerical results with circuit optimization problems which are indicative of problems with expensive function evaluations, where termination (because of inherent noise and practical considerations) is encouraged to occur before any significant asymptotic behavior. The numerical results are presented in Table 6.6. As in version 1, the second step consisted of the slack and minimax variable updates (5.4) and (5.11)–(5.13). However, the gradient and constraint tolerances used were 10^{-3} and 10^{-5}, respectively, with some safeguards related to an expected level of numerical noise. We can clearly observe from Table 6.6 that the two-step algorithm leads to better final objective function values. In practical applications where a simple function evaluation takes more than 10 minutes of CPU time, the effectiveness of such a simple addition is indeed significant. (There are situations where the greedy

Table 6.3

Comparison between versions 1 and 2 for nonminimax problems (LANCELOT without and with two-step updating).

Problem name	Inform	Iterations	Total CPU	Obj. function
CAR2	0/0	80/67	15.2/12.3	2.67
CORE1	0/0	953/983	7.41/17	91.1
CORE2	0/0	1048/1086	25.6/25.7	72.9
CORKSCRW	0/0	41/42	0.55/0.54	1.16
CSFI1	0/0	112/127	0.11/0.11	-49.1
CSFI2	0/0	78/83	0.07/0.07	55
HADAMARD	0/0	1709/548	2290/276	1.14/1
HS32	0/0	5/5	0.01/0.01	1
HS67	0/0	33/21	0.08/0.07	-1.16e+03
HS85	1/0	4000/3734	27.1/23.6	-1.85/-2.22
HS109	3/3	1578/753	7.58/3.11	5.36e+03
NET1	3/0	69/60	0.57/0.54	9.41e+05
NET2	3/0	95/69	3.53/2.92	1.19e+06
ORBIT2	0/3	615/612	3020/2750	312
PRODPL0	3/0	36/26	0.29/0.23	58.8
PRODPL1	0/0	56/32	0.55/0.51	35.7
SSEBNLN	0/0	51/51	1.46/1.47	1e+12
SWOPF	0/0	204/136	7.68/5.51	0.0679
TFI1	0/0	26/24	0.4/0.25	5.33
TFI2	0/0	25/45	0.33/0.41	0.649
TFI3	0/0	23/34	0.38/0.38	4.3
VANDERM1	0/0	13/13	0.05/0.08	0
VANDERM2	0/0	13/13	0.08/0.07	0
VANDERM3	0/0	14/16	0.07/0.08	0
VANDERM4	0/0	81/82	0.1/0.1	0
ZIGZAG	0/0	35/31	0.54/0.43	1.8

Table 6.4

Comparison between versions 1 and 2 for minimax problems (LANCELOT without and with two-step updating).

Problem name	Inform	Iterations	Total CPU	Obj. function
CB2	0/0	17/11	0.03/0.01	1.95
CB3	0/0	14/10	0.05/0.02	2
CHACONN1	0/0	12/8	0.02/0.04	1.95
CHACONN2	0/0	13/10	0.01/0.02	2
CONGIGMZ	0/0	32/19	0.04/0.05	28
COSHFUN	0/0	127/69	1.31/1.06	-0.773
DEMYMALO	0/0	24/17	0.03/0.03	-3
GIGOMEZ1	0/0	27/19	0.04/0.02	-3
GOFFIN	0/0	14/4	1.03/0.67	0
HAIFAM	1/0	4000/136	1140/85.1	-45
HALDMADS	0/0	48/73	0.49/0.72	0.0001
KIWCRESC	0/0	19/14	0.02/0.02	0
MADSEN	0/0	29/18	0.05/0.04	0.616
MAKELA1	0/0	17/18	0.04/0.02	-1.41
MAKELA2	0/0	21/9	0.05/0	7.2
MAKELA4	0/0	6/4	0.09/0.08	0
MIFFLIN1	0/0	11/7	0.03/0.01	-1
MIFFLIN2	0/0	37/32	0.04/0.05	-1
POLAK1	0/0	35/19	0.04/0.02	2.72
POLAK2	0/0	40/24	0.09/0.07	54.6
POLAK5	0/0	28/20	0.07/0.04	50
POLAK6	0/0	124/149	0.24/0.23	-44
SPIRAL	0/0	85/93	0.1/0.07	0
SPRALX	0/0	87/93	0.13/0.08	0

Table 6.5

Comparison of versions 1 *and* 3 *for minimax problems (LANCELOT without and with two-step updating only on slacks).*

Problem name	Inform	Iterations	Total CPU	Obj. function
CB2	0/0	17/17	0.03/0.03	1.95
CB3	0/0	14/16	0.05/0.03	2
CHACONN1	0/0	12/10	0.02/0.03	1.95
CHACONN2	0/0	13/13	0.01/0.04	2
CONGIGMZ	0/0	32/25	0.04/0.1	28
COSHFUN	0/0	127/92	1.31/1.08	-0.773
DEMYMALO	0/0	24/18	0.03/0.03	-3
GIGOMEZ1	0/0	27/20	0.04/0.02	-3
GOFFIN	0/0	14/8	1.03/0.66	0
HAIFAM	1/3	4000/609	1140/76.7	-45
HALDMADS	0/0	48/46	0.49/0.54	0.0001
KIWCRESC	0/0	19/18	0.02/0.03	0
MADSEN	0/0	29/23	0.05/0.05	0.616
MAKELA1	0/0	17/19	0.04/0.02	-1.41
MAKELA2	0/0	21/24	0.05/0.03	7.2
MAKELA4	0/0	6/6	0.09/0.11	0
MIFFLIN1	0/0	11/11	0.03/0.03	-1
MIFFLIN2	0/0	37/37	0.04/0.03	-1
POLAK1	0/0	35/32	0.04/0.06	2.72
POLAK2	0/0	40/15	0.09/0.04	54.6
POLAK5	0/0	28/28	0.07/0.01	50
POLAK6	0/0	124/332	0.24/0.48	-44
SPIRAL	0/0	85/85	0.1/0.07	0
SPRALX	0/0	87/87	0.13/0.09	0

two-step trust-region algorithm is able to take advantage of the decrease given by the slack and minimax variable updates and, by doing so, this algorithm can accept steps that otherwise would have been rejected; see Remark 3.1.)

We also applied the algorithms of this paper to analytic static-timing-based circuit optimization problems (see Table 6.7), where the advantage of the two-step approach is increasingly apparent for larger problems.

6.3. Further experiments with minimax problems. In this section we consider those minimax problems in our test set for which the two-step algorithm not only does not improve numerically the results obtained in the one-step case, but also makes them considerably worse (see the first part of Table 6.8). We analyze the reasons for the failure of the two-step updating on some minimax problems and discuss a few ways to enforce better numerical behavior.

We consider the general minimax problem (5.7). Our aim is to show that for some types of minimax problems the second step has a tendency to make the Hessian of Φ ill-conditioned. Let us assume that $\lambda_i = 0$ and $s_{ii} = 1$ for all $i = 1, \ldots, m$ (as happens by default for the first LANCELOT major iteration). Under these circumstances, we have

$$\Phi(x, z, u, \mu) \;=\; z + \frac{1}{2\mu} \sum_{i=1}^{m} (z - f_i(x) - u_i)^2.$$

By using the notation $g_i(x, z, u) = z - f_i(x) - u_i$, we have the following expressions for the elements of the gradient of Φ:

Table 6.6

LANCELOT without and with two-step updating for dynamic-simulation-based circuit optimization problems. Ineq. is the number of inequality constraints.

Problem name	Variables	Ineq.	Iterations	Total CPU	Obj. function
Nonminimax:					
IOmuxpower	102	42	21/29	7230/9220	-15100/-16000
durham2	13	2	17/17	93.5/93.5	472
chen2	2	1	14/14	91/91.2	4290
IOmux	101	41	60/61	18000/17700	-16200/-15900
Nov01power	5	1	37/54	24.5/35.6	273/268
lau2	5	1	33/32	47.9/46.3	158
Nov01	8	4	29/33	22.1/27.3	193/181
coulman_cold	33	17	22/22	69.5/68.3	271/262
clkgen	22	5	25/5	35/10.8	1.98/1.82
coulman_hot	33	17	16/32	46.2/100	283/253
davies3	16	1	30/30	368/368	254
coulman_delay	33	17	26/24	72.6/73.5	116/111
Minimax:					
bultmann_latch	39	13	17/18	41.8/46.8	95.9/84.6
stall1	30	5	23/19	3350/3050	156/86.8
coulman_cold_minmax	34	17	61/80	184/229	69.4/66.9
coulman_hot_minmax	34	17	66/44	197/134	74.4/75.1
fleischer	110	5	53/61	267/330	-458/-505
mod5	51	10	17/51	11200/33100	98.9/19
northrop_xor	18	8	67/64	78.3/77.7	-34.1/-30.2
coulman_delay_minmax	34	17	100/100	290/306	67.4/70.5

Table 6.7

LANCELOT without and with two-step updating for analytic (minimax) static-timing-based circuit optimization problems. Ineq. is the number of inequality constraints.

Problem name	Variables	Ineq.	Iterations	Total CPU	Obj. function
Symmetric 3	37	$2^4 - 1$	39/40	0.12/0.15	7.7
Symmetric 4	77	$2^5 - 1$	69/60	0.63/0.6	10.2
Symmetric 5	157	$2^6 - 1$	97/81	2.09/1.64	12.7
Symmetric 6	317	$2^7 - 1$	140/118	9.38/7.14	15.2
Symmetric 7	637	$2^8 - 1$	270/183	44.3/35.3	17.6
Symmetric 8	1277	$2^9 - 1$	385/340	247/221	19.9
Symmetric 9	2557	$2^{10} - 1$	901/639	1920/1300	22.1
Nonsymmetric 3	37	$2^4 - 1$	44/27	0.18/0.16	12.4
Nonsymmetric 4	77	$2^5 - 1$	58/37	0.57/0.31	16
Nonsymmetric 5	157	$2^6 - 1$	78/45	1.84/0.91	19.7
Nonsymmetric 6	317	$2^7 - 1$	75/54	5.89/3.3	23.6
Nonsymmetric 7	637	$2^8 - 1$	96/50	30.9/9.02	27.7
Nonsymmetric 8	1277	$2^9 - 1$	92/53	63.6/31.6	31.7
Nonsymmetric 9	2557	$2^{10} - 1$	130/63	300/95	35.7

$$
\begin{aligned}
\nabla_{x_j}\Phi &= -\frac{1}{\mu}\sum_{i=1}^{m}\nabla_{x_j} f_i(x) g_i(x,z,u), \quad j = 1,\dots,n,\\
\nabla_z\Phi &= 1 + \frac{1}{\mu}\sum_{i=1}^{m} g_i(x,z,u),\\
\nabla_{u_i}\Phi &= -\frac{1}{\mu} g_i(x,z,u), \ i = 1,\dots,m.
\end{aligned}
$$

Table 6.8

First part: comparison of versions 1 *and* 2 *for minimax problems (LANCELOT without and with two-step updating). Second part: comparison of versions* 4 *and* 5 *for minimax problems (LANCELOT without and with two-step updating).*

Problem name	Inform	Iterations	Total CPU	Obj. function
HAIFAL	0/1	679/4000	872346.06/366146.81	-12.8/-12.7828
MAKELA3	0/0	66/2816	0.26/5.84	0
MINMAXBD	0/0	267/952	1.34/3.59	116
POLAK3	0/0	71/125	0.4/0.8	5.93
POLAK4	3/1	14/4000	0.04/3.23	0
WOMFLET	0/0	63/150	0.07/0.13	0
HAIFAL	0/0	287/41	61603.1/8480.99	-12.8
MAKELA3	0/0	20/48	0.09/0.22	0
MINMAXBD	0/0	47/43	0.25/0.22	116
POLAK3	0/0	44/14	0.22/0.18	5.93
POLAK4	3/3	31/15	0.04/0.04	0
WOMFLET	0/0	26/32	0.03/0.04	6.05/0

Similarly the elements of the Hessian matrix of Φ are given by

$$\begin{array}{rr} \nabla^2_{x_j x_k}\Phi = -\frac{1}{\mu}\sum_{i=1}^m [\nabla^2_{x_j x_k} f_i(x) g_i(x,z,u) - \nabla_{x_j} f_i(x) \nabla_{x_k} f_i(x)], & \nabla^2_{zz}\Phi = \frac{m}{\mu}, \\ \nabla^2_{u_i u_l}\Phi = \frac{\delta_{il}}{\mu}, & \nabla^2_{z u_i}\Phi = -\frac{1}{\mu}, \\ \nabla^2_{x_j z}\Phi = -\frac{1}{\mu}\sum_{i=1}^m \nabla_{x_j} f_i(x), & \nabla^2_{u_i x_j}\Phi = \frac{1}{\mu}\nabla_{x_j} f_i(x), \end{array}$$

for $i, l = 1, \ldots, m$ and $j, k = 1, \ldots, n$. If the magnitudes of the products $\nabla^2_{x_j x_k} f_i(x) g_i(x,z,u)$ are small compared to those of the products $\nabla_{x_j} f_i(x) \nabla_{x_k} f_i(x)$, then the Hessian of Φ is given approximately by

$$\frac{1}{\mu}\left(\begin{array}{ccccccc} \sum_i a_{i1}a_{i1} & \ldots & \sum_i a_{i1}a_{in} & -\sum_i a_{i1} & a_{11} & \ldots & a_{m1} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_i a_{in}a_{i1} & \ldots & \sum_i a_{in}a_{in} & -\sum_i a_{in} & a_{1n} & \ldots & a_{mn} \\ -\sum_i a_{i1} & \ldots & -\sum_i a_{in} & m & -1 & \ldots & -1 \\ a_{11} & \ldots & a_{1n} & -1 & 1 & & \\ \vdots & \ddots & \vdots & \vdots & & \ddots & \\ a_{m1} & \ldots & a_{mn} & -1 & & & 1 \end{array}\right),$$

where a_{ij} denotes $\nabla_{x_j} f_i(x)$ and the indices i in the sums go from 1 to m. This matrix is clearly singular. In fact, the $(n+1)$st row is the negative sum of the last m rows. Moreover, any of the first n rows is a linear combination of the last m rows. As a result of these observations, the Hessian (and the projected Hessian) of Φ is ill-conditioned if

$$\left|\frac{1}{\mu}\sum_{i=1}^m \nabla_{x_j} f_i(x) \nabla_{x_k} f_i(x)\right| \gg \left|\frac{1}{\mu}\sum_{i=1}^m \nabla^2_{x_j x_k} f_i(x) g_i(x,z,u)\right| \tag{6.1}$$

happens for “many” indices j and k. This is the key point in this analysis: the second step has a tendency to produce iterates that worsen property (6.1) because it produces a decrease on the values of $g_i(x,z,u)$ for some indices i. The Hessian of Φ might very well be ill-conditioned if no second steps are applied, but there is no doubt (and the numerical results are evidence of this claim) that the second step for some problems worsens the situation by making the Hessian of Φ more ill-conditioned.

In the presence of nonzero Lagrange multipliers λ_i, $i = 1, \ldots, m$, the formulae for the gradient and the Hessian of Φ are the same with $g_i(x, z, u)$ substituted by $g_i(x, z, u) + \mu\lambda_i$, and similar conclusions could be drawn.

The second step may produce very bad results on some minimax problems because it points toward the set $\{(x, z, u) : g_i(x, z, u) = 0 \text{ for some } i\}$ (where the Hessian of the augmented Lagrangian is ill-conditioned), and this effect influences negatively the calculation of the first step at the next iteration. Given this undesirable feature of the Hessian of Φ at points close to this set, one possible improvement to the two-step algorithm is to make sure that the calculation of the first step is accurate (in the LANCELOT context this could be achieved by choosing a smaller tolerance for the stopping criterion of the conjugate-gradient technique). Another possible improvement is to reduce the ill-conditioning of the Hessian of Φ (for instance by increasing the value of the penalty parameter μ as can be seen in examples with a few variables). Indeed, these modifications improve the bad numerical results presented before: in the second part of Table 6.8 we compare the results obtained by the following modifications of versions 1 and 2:

4. Version 1 with an initial value for the penalty parameter μ of 100 (the default value is 0.1).
5. Version 2 with an initial value for the penalty parameter μ of 100 and a tolerance of 10^{-12} in the stopping criterion for conjugate gradients.

The study of strategies that can make two-step updating more effective for minimax problems in general is the subject for future research.

7. Concluding remarks.

In this paper we presented and analyzed a framework under which classical algorithms for nonlinear optimization can be modified to allow second computationally efficient steps that are not generated in the conventional way but that are guaranteed to yield decrease in the objective function. We gave as examples of the two-step algorithms the update of slack variables in LANCELOT and the update of variables introduced to solve minimax problems. However, we emphasize that the two-step algorithms can be very generally applied, for example, whenever the functions defining the problem are in a known functional form in some of the variables.

We considered trust-region algorithms for which we proposed a greedy and a conservative two-step algorithm. We analyzed the convergence properties of the trust-region two-step algorithms (see [11] for line-search two-step algorithms), deriving the conditions under which they attain global convergence. We also showed that a two-step Newton's method (for which the second step is computed only for a subset of the variables) has a q-quadratic rate of convergence.

The greedy two-step algorithms are designed to exploit as much as possible the decrease attained by the second step. The trust-region framework allowed us to design a greedy two-step trust-region algorithm that is particularly well tailored to achieve this purpose.

Finally, we included numerical evidence that this technique is effective, particularly for problems with expensive function evaluations. The two-step algorithms have already found practical applications in optimization of high-performance custom microprocessor integrated circuits.

Acknowledgments. We are grateful to N. I. M. Gould (Rutherford Appleton Laboratory) for his comments and suggestions on an earlier version of this paper, which led to many improvements. We are also grateful to K. Scheinberg (IBM T.J.

Watson Research Center) for help with the numerical results and explanation in section 6.3. We would like to thank I. M. Elfadel (IBM T.J. Watson Research Center) for providing the analytic static-timing-based optimization circuit problems. Finally, we are grateful to the referees for their useful comments and suggestions.

REFERENCES

[1] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Computer Science and Applied Mathematics, Academic Press, New York, 1982.

[2] I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *CUTE: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[3] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *Approximate solution of the trust-region problem by minimization over two-dimensional subspaces*, Math. Programming, 40 (1988), pp. 247–263.

[4] A. R. Conn, P. K. Coulman, R. A. Haring, G. L. Morrill, and C. Visweswariah, *Optimization of custom MOS circuits by transistor sizing*, in Proceedings, IEEE International Conference on Computer-Aided Design, San Jose, CA, 1996, pp. 174–180.

[5] A. R. Conn, P. K. Coulman, R. A. Haring, G. L. Morrill, C. Visweswariah, and C. W. Wu, *JiffyTune: Circuit optimization using time-domain sensitivities*, IEEE Trans. CAD of ICs and Systems, 17 (1998), pp. 1292–1309.

[6] A. R. Conn, I. M. Elfadel, W. W. Molzen, Jr., P. R. O'Brien, P. N. Strenski, C. Visweswariah, and C. B. Whan, *Gradient-based optimization of custom circuits using a static-timing formulation*, in Proceedings, 1999 Design Automation Conference, New Orleans, LA, 1999, pp. 452–459.

[7] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460.

[8] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.

[9] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1992.

[10] A. R. Conn, R. A. Haring, C. Visweswariah, and C. W. Wu, *Circuit optimization via adjoint Lagrangians*, in Proceedings, IEEE International Conference on Computer-Aided Design, 1997, San Jose, CA, pp. 281–288.

[11] A. R. Conn, L. N. Vicente, and C. Visweswariah, *Two-Step Algorithms for Nonlinear Optimization with Structured Applications*, Research Report RC 21198(94689), IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY, 1998.

[12] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[13] P. Feldmann, T. V. Nguyen, S. W. Director, and R. A. Rohrer, *Sensitivity computation in piecewise approximate circuit simulation*, IEEE Trans. CAD of ICs and Systems, 10 (1991), pp. 171–183.

[14] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley, Chichester, 1987.

[15] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization*, Report NA 97-2, Department of Mathematics, University of California, San Diego, 1997; SIAM J. Optim., submitted.

[16] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's Guide for NPSOL* 5.0*: A Fortran Package for Nonlinear Programming*, Technical Report SOL 86-1, System Optimization Laboratory, Stanford University, Stanford, CA, 1998.

[17] G. H. Golub and V. Pereyra, *The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate*, SIAM J. Numer. Anal., 10 (1973), pp. 413–432.

[18] N. I. M. Gould, *On solving three classes of nonlinear programming problems via simple differentiable penalty functions*, J. Optim. Theory Appl., 56 (1988), pp. 89–126.

[19] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison–Wesley, Reading, MA, 1989.

[20] J. J. Moré, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming: The State of the Art, A. Bachem, M. Grotschel, and B. Korte, eds., Springer-Verlag, New York, 1983, pp. 258–287.

[21] J. J. Moré, *Generalizations of the trust-region problem*, Optim. Methods Softw., 2 (1993),

pp. 189–209.

[22] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[23] M. J. D. Powell, *A new algorithm for unconstrained optimization*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, 1970, pp. 31–66.

[24] M. J. D. Powell, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1975, pp. 1–27.

[25] D. C. Sorensen, *Minimization of a large-scale quadratic function subject to a spherical constraint*, SIAM J. Optim., 7 (1997) 141–161.

[26] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[27] S. W. Thomas, *Sequential Estimation Techniques for Quasi-Newton Algorithms*, Ph.D. thesis, Cornell University, Ithaca, NY, 1975.

[28] C. Visweswariah and R. A. Rohrer, *Piecewise approximate circuit simulation*, IEEE Trans. CAD of ICs and Systems, 10 (1991), pp. 861–870.

SIAM J. OPTIM.
Vol. 9, No. 4, pp. 948–964

WHERE BEST TO HOLD A DRUM FAST[*]

STEVEN J. COX[†] AND PAUL X. UHLIG[‡]

To John Dennis, progenitor, advocate, and friend, on his 60th birthday

Abstract. If we are allowed to fasten, say, one half of a drum's boundary, which half produces the lowest or highest bass note? The answer is a natural limit of solutions to a family of extremal Robin problems for the least eigenvalue of the Laplacian. We produce explicit extremizers when the drum is a disk while for general shapes we establish existence and necessary conditions, and we build and test a pair of numerical methods.

Key words. membrane, eigenvalue, Robin, mixed, optimal design

AMS subject classifications. 35P15, 65K05, 73D30

PII. S1052623497326083

1. Introduction.

We consider the fundamental mode of vibration of a drumhead that is fastened along part of its boundary and free on the remainder. More precisely, we study the least eigenvalue of

$$\begin{aligned} -\Delta u &= \xi u \quad \text{in} \quad \Omega, \\ u &= 0 \quad \text{on} \quad \Gamma, \\ \frac{\partial u}{\partial n} &= 0 \quad \text{on} \quad \partial\Omega \setminus \Gamma, \end{aligned}$$

where Ω is a smooth, open, bounded, connected planar set and Γ is a measurable subset of its boundary. We denote this least eigenvalue by $\xi_1(\Gamma)$ and seek its extremes as Γ varies over subsets of $\partial\Omega$ of prescribed measure. Closely related questions for one-dimensional continua have been raised in the engineering literature; see, e.g., Mroz and Rozvany [14] and Chuang and Hou [5].

We begin the analysis of our model problem by expressing the two boundary conditions in the single equation

$$1_\Gamma u + (1 - 1_\Gamma)\partial u/\partial n = 0 \quad \text{on} \quad \partial\Omega, \tag{1.1}$$

where 1_Γ denotes the characteristic function of Γ. With an eye toward a convenient variational characterization of $\xi_1(\Gamma)$ we note that (1.1) is not a boundary condition of the third (or Robin) type. To achieve this the coefficient of $\partial u/\partial n$ must be constant. Before blindly dividing through by $1 - 1_\Gamma$ we introduce a simple regularization. In particular, we arrive at (1.1) in the limit as $\varepsilon \to 0$ in

$$1_\Gamma u + (1 + \varepsilon - 1_\Gamma)\partial u/\partial n = 0 \quad \text{on} \quad \partial\Omega$$

or, equivalently,

$$\varepsilon^{-1} 1_\Gamma u + \partial u/\partial n = 0 \quad \text{on} \quad \partial\Omega. \tag{1.2}$$

[*] Received by the editors August 19, 1997; accepted for publication (in revised form) December 2, 1997; published electronically September 24, 1999. This work was supported by NSF grant DMS 9258312 and a fellowship from the Humboldt Foundation.

http://www.siam.org/journals/siopt/9-4/32608.html

[†] Department of Computational and Applied Mathematics, Rice University, 6100 Main St., Houston, TX 77005 (cox@rice.edu).

[‡] Mathematics Department, St. Mary's University, San Antonio, TX 78228 (mathpaul @vax.stmarytx.edu).

Physically, the drumhead remains free on $\partial\Omega \setminus \Gamma$ while on Γ it is elastically supported by a fastener of stiffness $1/\varepsilon$. We denote by $\xi_1^\varepsilon(\Gamma)$ the least eigenvalue of $-\Delta$ subject to (1.2). This boundary condition is indeed of the third type and so we may record the weak formulation

$$\int_\Omega \nabla u \cdot \nabla v \, dx + \varepsilon^{-1} \int_{\partial\Omega} 1_\Gamma uv \, ds = \xi \int_\Omega uv \, dx \qquad \forall\, v \in H^1(\Omega) \tag{1.3}$$

and the associated variational characterization

$$\xi_1^\varepsilon(\Gamma) = \inf_{u \in H_1^1(\Omega)} \int_\Omega |\nabla u|^2 \, dx + \varepsilon^{-1} \int_{\partial\Omega} 1_\Gamma u^2 \, ds, \tag{1.4}$$

where $H_1^1(\Omega)$ is the class of $H^1(\Omega)$ functions with $L^2(\Omega)$ norm one. The advantage of the chosen regularization lies in the fact that in both (1.3) and (1.4), the underlying function space *does not* vary with Γ.

We now fix a number $\gamma \in (0,1)$ (the Dirichlet fraction) and formulate the optimal design problems whose solutions will determine the range of $\xi_1^\varepsilon(\Gamma)$ as Γ varies over those subsets of $\partial\Omega$ of size $\gamma|\partial\Omega|$. In particular, we study

$$\inf_{1_\Gamma \in ad_\gamma(\partial\Omega)} \xi_1^\varepsilon(\Gamma) \quad \text{and} \quad \sup_{1_\Gamma \in ad_\gamma(\partial\Omega)} \xi_1^\varepsilon(\Gamma),$$

where

$$ad_\gamma(\partial\Omega) \equiv \{1_\Gamma : \Gamma \subset \partial\Omega,\ |\Gamma| = \gamma|\partial\Omega|\}$$

and $|\Gamma|$ denotes the one-dimensional Hausdorff measure of Γ. Generally speaking, we shall see that minimal designs favor a connected Γ while maximal designs tend to fragment Γ. Accordingly, in section 2, we establish existence of minimizers and (relaxed) maximizers by showing that ξ_1^ε is weak* continuous on the weak* closure of $ad_\gamma(\partial\Omega)$. In section 3 we characterize minimizers via first order necessary conditions and provide an explicit minimal design for the disk. In section 4 analogous first order conditions lead to the uniqueness of the maximizer and its characterization in terms of the normal derivative of the first eigenfunction of the pure Dirichlet problem. In section 5 we construct distinct approaches to the numerical minimization and maximization of ξ_1^ε. We test these methods on elliptical and L-shaped drums in section 6.

Although stated in the context of the planar Laplacian, our arguments apply, without change, to second order self-adjoint elliptic equations on smooth bounded domains in an arbitrary number of dimensions. Although isoperimetric inequalities for mixed and Robin problems have received considerable attention (see, e.g., Bandle [1]) the paper of Buttazzo [4] appears to be the first and only to consider an extremal Robin problem on a fixed domain.

On completion of this work we learned that Denzler [10] had been simultaneously pursuing the same set of questions. Via methods quite distinct from those invoked here he showed that ξ_1 attains its minimum on $ad_\gamma(\partial\Omega)$ and that the supremum of ξ_1 is $\lambda_1(\Omega)$, the least Dirichlet eigenvalue.

2. Existence. We shall denote by $L(\partial\Omega, [0,1])$ those measurable functions on $\partial\Omega$ that take values in the interval $[0,1]$. With respect to the weak* topology on $L^\infty(\partial\Omega)$ Friedland [12] has shown the following.

Proposition 2.1. *The weak* closure of* $ad_\gamma(\partial\Omega)$ *is*

$$ad_\gamma^*(\partial\Omega) \equiv \left\{ \theta \in L(\partial\Omega, [0,1]) : \int_{\partial\Omega} \theta(x) \, ds = \gamma|\partial\Omega| \right\}.$$

In addition, $ad_\gamma(\partial\Omega)$ is the set of extreme points of $ad^*_\gamma(\partial\Omega)$.

For $\theta \in ad^*_\gamma(\partial\Omega)$ we denote by $\xi^\varepsilon_1(\theta)$ the first eigenvalue of $-\Delta$ subject to

$$\varepsilon^{-1}\theta u + \partial u/\partial n = 0 \quad \text{on} \quad \partial\Omega. \tag{2.1}$$

The analogous variational characterization

$$\xi^\varepsilon_1(\theta) = \inf_{u\in H^1_1(\Omega)} \mathcal{R}_\varepsilon(u,\theta), \qquad \text{where} \quad \mathcal{R}_\varepsilon(u,\theta) \equiv \int_\Omega |\nabla u|^2\,dx + \varepsilon^{-1}\int_{\partial\Omega}\theta u^2\,ds, \tag{2.2}$$

leads immediately to

$$0 < \xi^\varepsilon_1(\theta) \le \lambda_1(\Omega) \qquad \forall\,\theta \in ad^*_\gamma(\partial\Omega) \quad \text{and} \quad \forall\,\varepsilon > 0, \tag{2.3}$$

where $\lambda_1(\Omega)$ is the first eigenvalue of $-\Delta$ subject to Dirichlet conditions over the *entire* boundary. As $\mathcal{R}_\varepsilon(u,\theta) = \mathcal{R}_\varepsilon(|u|,\theta)$ it follows from (2.2) that $\xi^\varepsilon_1(\theta)$ is simple and may be associated with a nonnegative eigenfunction.

Proposition 2.2. *The mapping $\theta \mapsto \xi^\varepsilon_1(\theta)$ is continuous with respect to the weak* topology on $L(\partial\Omega,[0,1])$.*

Proof. Suppose $\theta_n \overset{*}{\rightharpoonup} \theta$ and that u_n is the positive first eigenfunction, associated with θ_n, normalized such that

$$\int_\Omega u_n^2\,dx = 1 \quad \text{and} \quad \int_\Omega |\nabla u_n|^2\,dx + \varepsilon^{-1}\int_{\partial\Omega}\theta_n u_n^2\,ds = \xi^\varepsilon_1(\theta_n). \tag{2.4}$$

From (2.3) and (2.4) it follows that $\{u_n\}_n$ is bounded in $H^1(\Omega)$ and hence that $u_n \rightharpoonup u$ in $H^1(\Omega)$ and $u_n \to u$ in $L^2(\Omega)$ and the traces $u_n|_{\partial\Omega} \to u|_{\partial\Omega}$ in $L^2(\partial\Omega)$. In addition $\xi^\varepsilon_1(\theta_n) \to \xi$. These observations permit us to pass to the limit in the weak form

$$\int_\Omega \nabla u_n\cdot\nabla v\,dx + \varepsilon^{-1}\int_{\partial\Omega}\theta_n u_n v\,ds = \xi^\varepsilon_1(\theta_n)\int_\Omega u_n v\,dx$$

and so conclude that ξ and u constitute an eigenpair for θ. As u is positive it follows that $\xi = \xi^\varepsilon_1(\theta)$. □

As $ad^*_\gamma(\partial\Omega)$ is weak* compact Corollary 2.3 now follows.

Corollary 2.3.

$$\inf_{1_\Gamma\in ad_\gamma(\partial\Omega)} \xi^\varepsilon_1(\Gamma) = \min_{\theta\in ad^*_\gamma(\partial\Omega)} \xi^\varepsilon_1(\theta)$$

and

$$\sup_{1_\Gamma\in ad_\gamma(\partial\Omega)} \xi^\varepsilon_1(\Gamma) = \max_{\theta\in ad^*_\gamma(\partial\Omega)} \xi^\varepsilon_1(\theta).$$

Our interest is in characterizing those θ at which ξ^ε_1 attains its extremes. A number of previous studies have produced lower and upper bounds for $\xi^\varepsilon_1(\theta)$.

Regarding the latter, such bounds are typically achieved by replacing θ with a constant and Ω with a disk. Pólya and Szegő accomplish this for starlike Ω via the method of similar level lines; see Bandle [1, Thm. III.3.21]. Hersch uses conformal transplantation and so requires that Ω merely be simply connected. More precisely, he demonstrates (see [1, Thm. III.3.17]) that

$$\xi^\varepsilon_1(\theta,\Omega) \le \xi^\varepsilon_1(\gamma|\partial\Omega|/|\partial D_\Omega|, D_\Omega) \qquad \forall\,\theta \in ad^*_\gamma(\partial\Omega), \tag{2.5}$$

where D_Ω is the disk with radius equal to the conformal radius of Ω. Of course, when Ω is itself a disk this result states that $\theta \equiv \gamma$ is maximal.

The construction of useful lower bounds is considerably more difficult. All attempts to bound $\xi_1^\varepsilon(\theta)$ from below apply *only* to the case of constant θ. We cite Philippin [15], Bossel [3], and Sperb [17].

3. Minimizing ξ_1^ε.

We show that $\theta \mapsto \xi_1^\varepsilon(\theta)$ possesses a classical, i.e., $ad_\gamma(\partial\Omega)$, minimizer. We compute it in the case of the disk while in the general case we produce pointwise optimality conditions.

Returning to (2.2) we recognize that $\theta \mapsto \xi_1^\varepsilon(\theta)$ is an infimum of affine functions of θ. The following proposition results.

PROPOSITION 3.1. *$\theta \mapsto \xi_1^\varepsilon(\theta)$ is concave on $ad_\gamma^*(\partial\Omega)$.*

If we now recall (see, e.g., Bauer [2]) that a bounded concave function on a compact convex set attains its minimum at an extreme point, we arrive at the following.

COROLLARY 3.2. *$\theta \mapsto \xi_1^\varepsilon(\theta)$ attains its minimum on $ad_\gamma(\partial\Omega)$.*

We now produce an explicit minimizer in the case that Ω is a disk, D. This is accomplished through circular symmetrization, defined as follows.

Given $v \in H^1(D)$ we take $u(r,t) = v(x)$, where $x = r(\cos t, \sin t)$ and $-\pi < t \le \pi$. Now, at each r we replace $t \mapsto u(r,t)$ with its symmetrically increasing rearrangement

$$u^\vee(r,t) = \inf\{c : t \in \{s : u(r,s) \le c\}^*\},$$

where A^* is simply the interval $(-|A|/2, |A|/2)$. We then take $v^\vee(x) \equiv u^\vee(r,t)$ to be the circular (increasing about $t = 0$) rearrangement of v. The corresponding symmetrically decreasing rearrangement is

$$u^\wedge(r,t) = u^\vee(r, \pi - |t|).$$

As a simple example we note that if $1_\Gamma \in ad_\gamma(\partial D)$, then

$$1_\Gamma^\wedge(t) = 1_{\Gamma^*} = \begin{cases} 1 & \text{if } |t| \le \gamma\pi, \\ 0 & \text{otherwise.} \end{cases}$$

We now recall (see, e.g., Cox and Kawohl [9]) that circular rearrangement cannot increase the Dirichlet integral and that $u^\vee$ and $1_\Gamma^\wedge$ are oppositely ordered. As a result,

$$\mathcal{R}_\varepsilon(v, 1_\Gamma) \ge \mathcal{R}_\varepsilon(v^\vee, 1_{\Gamma^*}) \qquad \forall\,(v, 1_\Gamma) \in H_1^1(D) \times ad_\gamma(\partial D)$$

and so we arrive at the following proposition.

PROPOSITION 3.3. *$1_\Gamma \mapsto \xi_1^\varepsilon(1_\Gamma)$ attains its minimum at 1_{Γ^*}.*

As 1_{Γ^*} is clearly independent of ε we proceed to let ε approach 0. Our preliminary result does not require the domain to be a disk.

LEMMA 3.4. *If $\Gamma \subset \partial\Omega$, then $\xi_1^\varepsilon(1_\Gamma) \to \xi_1(1_\Gamma)$ as $\varepsilon \to 0$.*

Proof. Let $u_\varepsilon \in H_1^1(\Omega)$ denote the eigenfunction associated with $\xi_1^\varepsilon(1_\Gamma)$. Now, recalling (2.3), we find

$$\int_\Omega |\nabla u_\varepsilon|^2\,dx + \varepsilon^{-1}\int_\Gamma u_\varepsilon^2\,dx = \xi_1^\varepsilon(1_\Gamma) \le \lambda_1(\Omega). \tag{3.1}$$

As a result, $\{u_\varepsilon\}_{\varepsilon>0}$ is clearly bounded in $H^1(\Omega)$ and, moreover,

$$\int_\Gamma u_\varepsilon^2\,dx = O(\varepsilon).$$

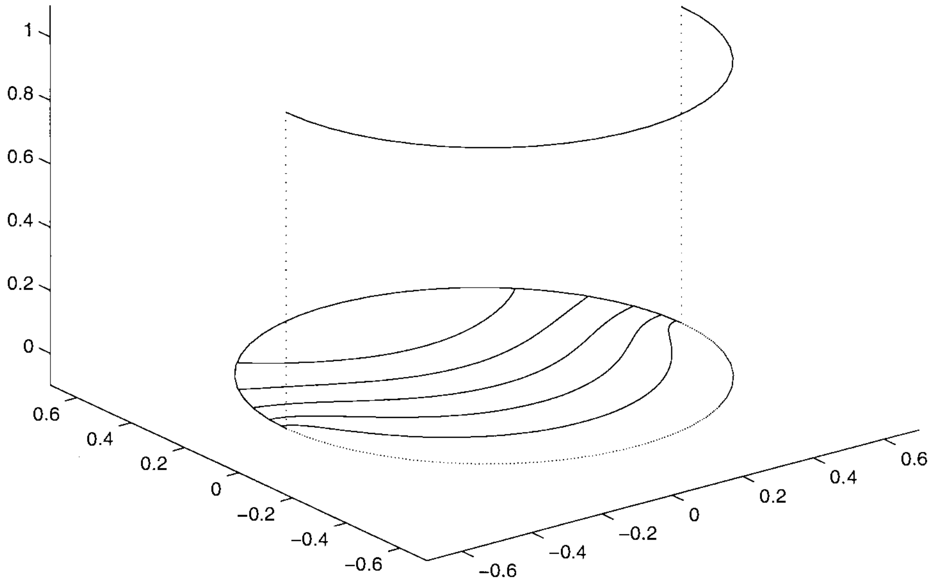


FIG. 1. *Minimal fastening of the disk.*

Hence (a subsequence of) u_ε converges weakly in $H^1(\Omega)$ to some $u_0 \in H^1_1(\Omega, \Gamma)$, those functions in $H^1_1(\Omega)$ with vanishing trace on Γ. We now show that u_0 is the eigenfunction associated with $\xi_1(1_\Gamma)$. Taking the limit inferior throughout (3.1) gives

$$\int_\Omega |\nabla u_0|^2\,dx \le \liminf_{\varepsilon\to 0} \xi_1^\varepsilon(\Gamma).$$

Now if there exists a $u \in H^1_1(\Omega, \Gamma)$ and a $\delta > 0$ for which

$$\int_\Omega |\nabla u|^2\,dx \le \int_\Omega |\nabla u_0|^2\,dx - \delta,$$

then (3.1) implies $\mathcal{R}_\varepsilon(u, 1_\Gamma) < \xi_1^\varepsilon(\Gamma)$ for some ε, contrary to Rayleigh's principle. Hence,

$$\xi_1(\Gamma) = \int_\Omega |\nabla u_0|^2\,dx \le \liminf_{\varepsilon\to 0} \xi_1^\varepsilon(\Gamma).$$

The simple observation $\xi_1^\varepsilon(\Gamma) \le \xi_1(\Gamma)$ completes the argument. □

COROLLARY 3.5. $1_\Gamma \mapsto \xi_1(1_\Gamma)$ *attains its minimum at* 1_{Γ^*}.

In Figure 1 we have plotted 1_{Γ^*} for $\gamma = 1/2$ on the disk of unit diameter along with the contours of the associated first eigenfunction, computed by the `pdeeig` routine in MATLAB [13] via a piecewise linear approximation on 259328 triangles. The computed value of $\xi_1(1_{\Gamma^*})$ is 4.86.

As the eigenvalue problem for such a design does not yield to separation of variables we return to the question posed at the close of the last section, namely, can one bound $\xi_1(1_{\Gamma^*})$ from below? Even in this simplest of all possible geometries our best analytical bound requires the majority of the boundary to be Dirichlet. More precisely, if Ω is the disk of radius R and $\gamma > 1/2$, then

$$\xi_1(1_{\Gamma^*}) \ge \frac{2\gamma - 1}{2R^2} j_0^2,$$

where j_0 is the first zero of the Bessel function J_0. This follows from Bandle's generalization of a result of Nehari; see [1, Thm. III.3.9].

We now return to a general domain and denote by $\check{\theta}^\varepsilon$ the minimizer of ξ_1^ε over $ad_\gamma^*(\partial\Omega)$. We take $\check{u}^\varepsilon \in H_1^1(\Omega)$ to be the positive eigenfunction associated with $\check{\theta}^\varepsilon$ and record

$$\xi_1^\varepsilon(\check{\theta}^\varepsilon) = \mathcal{R}_\varepsilon(\check{u}^\varepsilon, \check{\theta}^\varepsilon) = \min_{\theta\in ad_\gamma^*(\partial\Omega)} \min_{u\in H_1^1(\Omega)} \mathcal{R}_\varepsilon(u,\theta) = \min_{u\in H_1^1(\Omega)} \min_{\theta\in ad_\gamma^*(\partial\Omega)} \mathcal{R}_\varepsilon(u,\theta).$$

In other words,

$$\mathcal{R}_\varepsilon(\check{u}^\varepsilon, \check{\theta}^\varepsilon) = \min_{u\in H_1^1(\Omega)} \mathcal{R}_\varepsilon(u, \check{\theta}^\varepsilon) \quad \text{and} \quad \mathcal{R}_\varepsilon(\check{u}^\varepsilon, \check{\theta}^\varepsilon) = \min_{\theta\in ad_\gamma^*(\partial\Omega)} \mathcal{R}_\varepsilon(\check{u}^\varepsilon, \theta).$$

The former simply states that $\check{u}^\varepsilon$ is an eigenfunction corresponding to $\check{\theta}^\varepsilon$. The latter, however, informs us that

$$\int_{\partial\Omega} \check{\theta}^\varepsilon |\check{u}^\varepsilon|^2\, ds = \min_{\theta\in ad_\gamma^*(\partial\Omega)} \int_{\partial\Omega} \theta |\check{u}^\varepsilon|^2\, ds. \tag{3.2}$$

We remove the integral constraint on $\check{\theta}^\varepsilon$ at the cost of a Lagrange multiplier. More precisely, from the Lagrange multiplier rule, [6, Thm. 6.1.1], we deduce that (3.2) implies the existence of $\nu_1 \ge 0$ and $|\nu_1| + |\nu_2| > 0$ such that

$$\int_{\partial\Omega} \check{\theta}^\varepsilon (\nu_1 |\check{u}^\varepsilon|^2 + \nu_2)\, ds = \min_{\theta\in L(\partial\Omega,[0,1])} \int_{\partial\Omega} \theta (\nu_1 |\check{u}^\varepsilon|^2 + \nu_2)\, ds. \tag{3.3}$$

From $\nu_1 |\check{u}^\varepsilon|^2 \ge 0$ we deduce from (3.3) that $\nu_2 \le 0$.

If $\nu_2 = 0$, then (3.3) implies that $\check{\theta}^\varepsilon \check{u}^\varepsilon$ must vanish on the full boundary. Now, the boundary condition (2.1) implies that $\check{u}^\varepsilon$ is a Neumann eigenfunction. As $\check{u}^\varepsilon$ does not change sign it can only be the constant eigenfunction. Now $\check{\theta}^\varepsilon \check{u}^\varepsilon = 0$ implies that $\check{\theta}^\varepsilon$ is identically zero, contrary to its integral constraint. Therefore, $\nu_2 < 0$.

Now, if $\nu_1 = 0$, then as $\nu_2 < 0$, (3.3) implies that $\check{\theta}^\varepsilon$ is identically one, contrary to its integral constraint. Therefore, $\nu_1 > 0$.

With $\nu^2 \equiv -\nu_2/\nu_1$ we deduce from (3.3) the following pointwise necessary conditions:

$$\check{\theta}^\varepsilon(x) = 0 \Rightarrow \check{u}^\varepsilon(x) \ge \nu, \tag{3.4}$$

$$0 < \check{\theta}^\varepsilon(x) < 1 \Rightarrow \check{u}^\varepsilon(x) = \nu, \tag{3.5}$$

$$\check{\theta}^\varepsilon(x) = 1 \Rightarrow \check{u}^\varepsilon(x) \le \nu. \tag{3.6}$$

Recalling that $\check{\theta}^\varepsilon$ may be assumed a member of $ad_\gamma(\partial\Omega)$, it follows that $\check{\theta}^\varepsilon$ jumps across a level set of the trace of its corresponding eigenfunction, $\check{u}^\varepsilon$.

4. Maximizing ξ_1^ε.

Recalling (2.5) we begin with a simple proof of the fact that constant θ is maximal for the disk. Noting only that u_γ, the eigenfunction corresponding to $\theta \equiv \gamma$ on the disk, is radial we find

$$\xi_1^\varepsilon(\theta) \le \mathcal{R}_\varepsilon(u_\gamma, \theta) = \mathcal{R}_\varepsilon(u_\gamma, \gamma) = \xi_1^\varepsilon(\gamma) \qquad \forall\, \theta \in ad_\gamma^*(\partial D). \tag{4.1}$$

With regard to general Ω we shall see that where the maximizing θ is neither zero nor one the trace of its corresponding eigenfunction is, like u_γ, constant. In addition, we establish uniqueness of the maximizer and show that when it lies everywhere between zero and one it is (to lowest order in ε) proportional to the normal derivative of the first Dirichlet eigenfunction on Ω.

The first step is the derivation of pointwise conditions analogous to (3.4)–(3.6). These shall stem from knowledge of the gradient of $\theta \mapsto \xi_1^\varepsilon(\theta)$.

PROPOSITION 4.1. *$\theta \mapsto \xi_1^\varepsilon(\theta)$ is smooth and*

$$\langle \partial \xi_1^\varepsilon(\theta), \psi \rangle = \varepsilon^{-1} \int_{\partial\Omega} \psi u^2 \, ds,$$

where $u \in H_1^1(\Omega)$ is the nonnegative eigenfunction associated with θ.

Proof. The gradient of a simple eigenvalue of a self-adjoint operator is the gradient of the Rayleigh quotient evaluated at the corresponding eigenfunction. See Cox [8] for details. □

If $\hat{\theta}^\varepsilon$ maximizes ξ_1^ε over $ad_\gamma^*(\partial\Omega)$, then $\partial\xi_1^\varepsilon(\hat{\theta}^\varepsilon) \in N_{ad_\gamma^*(\partial\Omega)}(\hat{\theta}^\varepsilon)$, the cone of normals to $ad_\gamma^*(\partial\Omega)$ at $\hat{\theta}^\varepsilon$. As $ad_\gamma^*(\partial\Omega)$ is convex this means that

$$\langle \partial\xi_1^\varepsilon(\hat{\theta}^\varepsilon), \hat{\theta}^\varepsilon \rangle = \max_{\theta \in ad_\gamma^*(\partial\Omega)} \langle \partial\xi_1^\varepsilon(\hat{\theta}^\varepsilon), \theta \rangle,$$

that is,

$$\int_{\partial\Omega} \hat{\theta}^\varepsilon |\hat{u}^\varepsilon|^2 \, ds = \max_{\theta \in ad_\gamma^*(\partial\Omega)} \int_{\partial\Omega} \theta |\hat{u}^\varepsilon|^2 \, ds, \tag{4.2}$$

where $\hat{u}^\varepsilon$ is the positive eigenfunction corresponding to $\hat{\theta}^\varepsilon$. As above, to shed the integral constraint we invoke the Lagrange multiplier rule of Clarke. This gives a $\nu_1 \le 0$ and ν_2 for which $|\nu_1| + |\nu_2| > 0$ and

$$\int_{\partial\Omega} \hat{\theta}^\varepsilon (\nu_1 |\hat{u}^\varepsilon|^2 + \nu_2) \, ds = \max_{\theta \in L(\partial\Omega, [0,1])} \int_{\partial\Omega} \theta (\nu_1 |\hat{u}^\varepsilon|^2 + \nu_2) \, ds. \tag{4.3}$$

From $\nu_1 |\hat{u}^\varepsilon|^2 \le 0$ we deduce from (4.3) that $\nu_2 > 0$. Similarly, $\nu_1 < 0$. With $\nu^2 \equiv -\nu_2/\nu_1$ we arrive at the pointwise necessary conditions

$$\hat{\theta}^\varepsilon(x) = 0 \Rightarrow \hat{u}^\varepsilon(x) \le \nu, \tag{4.4}$$

$$0 < \hat{\theta}^\varepsilon(x) < 1 \Rightarrow \hat{u}^\varepsilon(x) = \nu, \tag{4.5}$$

$$\hat{\theta}^\varepsilon(x) = 1 \Rightarrow \hat{u}^\varepsilon(x) \ge \nu. \tag{4.6}$$

From Proposition 3.1 we note that these conditions are also sufficient.

A further consequence of (4.2) is that $(\hat{u}^\varepsilon, \hat{\theta}^\varepsilon)$ is a saddle point of $\mathcal{R}_\varepsilon$, i.e.,

$$\mathcal{R}_\varepsilon(\hat{u}^\varepsilon, \theta) \le \mathcal{R}_\varepsilon(\hat{u}^\varepsilon, \hat{\theta}^\varepsilon) \le \mathcal{R}_\varepsilon(u, \hat{\theta}^\varepsilon) \qquad \forall\, (u, \theta) \in H_1^1(\Omega) \times ad_\gamma^*(\partial\Omega).$$

From this observation comes the following proposition.

PROPOSITION 4.2. *$\hat{\theta}^\varepsilon$ is unique.*

Proof. Suppose that θ_1 and θ_2 are both maximizers of $\theta \mapsto \xi_1^\varepsilon(\theta)$ and that u_1 and u_2 are the respective first eigenfunctions. We find

$$\mathcal{R}_\varepsilon(u_1, \theta_2) \le \mathcal{R}_\varepsilon(u_1, \theta_1) \le \mathcal{R}_\varepsilon(u_2, \theta_1),$$
$$\mathcal{R}_\varepsilon(u_2, \theta_1) \le \mathcal{R}_\varepsilon(u_2, \theta_2) \le \mathcal{R}_\varepsilon(u_1, \theta_2).$$

However, as $\mathcal{R}_\varepsilon(u_1, \theta_1) = \mathcal{R}_\varepsilon(u_2, \theta_2)$ we find that u_1 and u_2 are both eigenfunctions for θ_1 and hence $u_1 = u_2$. Recalling the respective weak forms we find

$$\int_{\partial\Omega} (\theta_1 - \theta_2) u_1 v \, ds = 0 \qquad \forall\, v \in H^1(\Omega),$$

and hence $\theta_1 = \theta_2$ on the support of $u_1|_{\partial\Omega}$, the trace of u_1. Off of the support of $u_1|_{\partial\Omega}$ it follows from (4.4) that $\theta_1 = \theta_2 = 0$. □

From uniqueness we are able to ascertain symmetry. In particular, if Ω is symmetric with respect to a line ℓ we may reflect $\hat\theta^\varepsilon$ across ℓ to $\hat\theta^\varepsilon_\ell$. By simply reflecting the associated $\hat u^\varepsilon$ it follows that $\xi_1^\varepsilon(\hat\theta^\varepsilon) = \xi_1^\varepsilon(\hat\theta^\varepsilon_\ell)$ and hence, by uniqueness, that $\hat\theta^\varepsilon = \hat\theta^\varepsilon_\ell$. We have proven the following.

PROPOSITION 4.3. *$\hat\theta^\varepsilon$ is symmetric about every line of symmetry of Ω.*

This leads to a third proof of (4.1).

PROPOSITION 4.4. *If Ω is a disk, then $\hat\theta^\varepsilon \equiv \gamma$. Disks are the only (smooth) sets with a constant maximizer.*

Proof. Full symmetry implies that $\hat\theta^\varepsilon$ must be constant. The only admissible constant is γ. Given a constant maximizer, it follows from (4.5) that $\check u^\varepsilon$ is identically ν on $\partial\Omega$. From the boundary condition (2.1) we then find that $\partial\check u^\varepsilon/\partial n = -\nu\gamma/\varepsilon$ on $\partial\Omega$. Serrin [16, Thm. 2] has shown that a disk is the *only* C^2 domain on which one may solve $(\Delta + \xi)u = 0$ subject to constant Dirichlet and Neumann data. □

If $\Omega = D_a$ is a disk of radius a, then $u(r) = J_0(\sqrt{\xi} r)$ is a radial solution of $-\Delta u = \xi u$. The best eigenvalue, $\xi_1^\varepsilon(\gamma)$, is therefore the least positive ξ for which

$$\gamma u(a) + \varepsilon u'(a) = 0.$$

It follows immediately then that $\xi_1^\varepsilon(\gamma) \to \lambda_1(D_a)$ as $\varepsilon \to 0$, where $\lambda_1(D_a)$ is the least positive root of $\lambda \mapsto J_0(\sqrt{\lambda} a)$, i.e., the first Dirichlet eigenvalue of D_a. This approach to the Dirichlet eigenvalue holds, in fact, for every domain Ω.

PROPOSITION 4.5. *If $\hat\theta^\varepsilon$ maximizes $\theta \mapsto \xi_1^\varepsilon(\theta)$ over $ad_\gamma(\partial\Omega)$, then $\xi_1^\varepsilon(\hat\theta^\varepsilon) \to \lambda_1(\Omega)$ as $\varepsilon \to 0$.*

Proof. As $\xi_1^\varepsilon(\gamma) \le \xi_1^\varepsilon(\hat\theta^\varepsilon) \le \lambda_1(\Omega)$ it suffices to show that

$$\lambda_1(\Omega) \le \liminf_{\varepsilon \to 0} \xi_1^\varepsilon(\gamma). \tag{4.7}$$

Let us denote by $u_1^\varepsilon \in H^1_1(\Omega)$ the positive eigenfunction corresponding to $\xi_1^\varepsilon(\gamma)$. As $\|u_1^\varepsilon\|_2 = 1$ and $\|\nabla u_1^\varepsilon\|_2^2 \le \lambda_1(\Omega)$ it follows that there exists a $u_1 \in H^1_1(\Omega)$ for which $u_1^\varepsilon \rightharpoonup u_1$ in $H^1(\Omega)$ as $\varepsilon \to 0$. Given the normalization of u_1^ε we find that

$$\gamma \int_{\partial\Omega} |u_1^\varepsilon|^2 \, ds = \varepsilon \int_\Omega |\nabla u_1^\varepsilon|^2 \, dx + \varepsilon \xi_1^\varepsilon(\gamma) \to 0$$

as $\varepsilon \to 0$, i.e., $u_1^\varepsilon|_{\partial\Omega} \to 0$ in $L^2(\partial\Omega)$. As $u_1^\varepsilon|_{\partial\Omega} \to u_1|_{\partial\Omega}$ in $L^2(\partial\Omega)$ it follows that $u_1 \in H^1_0(\Omega)$. Now, given the weak lower semicontinuity of $u \mapsto \|\nabla u\|_2^2$ and the nonnegativity of the boundary term, we find

$$\int_\Omega |\nabla u_1|^2 \, dx \le \liminf_{\varepsilon \to 0} \int_\Omega |\nabla u_1^\varepsilon|^2 \, dx + \frac{\gamma}{\varepsilon} \int_{\partial\Omega} |u_1^\varepsilon|^2 \, ds = \liminf_{\varepsilon \to 0} \xi_1^\varepsilon(\gamma).$$

As $u_1 \in H^1_0(\Omega)$ and $\|u_1^\varepsilon\|_2 = 1$ it follows from Rayleigh's principle that the left-hand side is larger than $\lambda_1(\Omega)$. This establishes (4.7). □

This proposition addresses the limiting behavior of the eigenvalue but says nothing about the limiting optimal design. We shall now show that if the limiting design takes values strictly between 0 and 1, then it is proportional to the normal derivative of the first Dirichlet eigenfunction.

We begin at the necessary condition (4.5) and note that for constant ν and $\xi < \lambda_1(\Omega)$ one may solve

$$-\Delta u = \xi u \quad \text{in} \quad \Omega, \qquad u = \nu \quad \text{on} \quad \partial\Omega$$

in terms of the Dirichlet eigenfunctions, $\{\phi_j\}$, and Dirichlet eigenvalues, $\{\lambda_j\}$, of Ω. In particular,

$$u = \nu + \nu\xi \sum_{j=1}^{\infty} \frac{\langle \phi_j, 1\rangle}{\lambda_j - \xi}\phi_j.$$

The Robin condition (2.1) now suggests

$$\theta = -\frac{\varepsilon}{\nu}\frac{\partial u}{\partial n} = -\varepsilon\xi \sum_{j=1}^{\infty} \frac{\langle \phi_j, 1\rangle}{\lambda_j - \xi}\frac{\partial \phi_j}{\partial n}. \tag{4.8}$$

Integrating this expression over $\partial\Omega$ we find

$$\gamma|\partial\Omega| = \int_{\partial\Omega} \theta\, ds = -\varepsilon\xi \sum_{j=1}^{\infty} \frac{\langle \phi_j, 1\rangle}{\lambda_j - \xi} \int_{\partial\Omega} \frac{\partial \phi_j}{\partial n}\, ds = \varepsilon\xi \sum_{j=1}^{\infty} \frac{\langle \phi_j, 1\rangle^2}{\lambda_j - \xi}\lambda_j. \tag{4.9}$$

We view this as an equation for ξ. As the right side is continuous and strictly increasing from 0 (at $\xi = 0$) to ∞ (at $\xi = \lambda_1(\Omega)$) there exists a unique solution, ξ_1^ε, depending smoothly on ε. Expressing ξ_1^ε as a power series, identification of like powers in (4.9) brings

$$\xi_1^\varepsilon = \lambda_1(\Omega) - \frac{\lambda_1^2(\Omega)\langle \phi_1, 1\rangle^2}{\gamma|\partial\Omega|}\varepsilon + O(\varepsilon^2). \tag{4.10}$$

Substituting this into (4.8) we arrive at

$$\theta^\varepsilon = \gamma\frac{\partial \phi_1}{\partial n}\Big/\overline{\frac{\partial \phi_1}{\partial n}} + O(\varepsilon) \quad \text{where} \quad \overline{\frac{\partial \phi_1}{\partial n}} = \frac{1}{|\partial\Omega|}\int_{\partial\Omega} \frac{\partial \phi_1}{\partial n}\, ds. \tag{4.11}$$

Hence, if $\hat{\theta}^\varepsilon$ takes values strictly between 0 and 1 it must necessarily be of this form. Moreover, as the necessary conditions are also sufficient, whenever the above derivation produces an admissible design this design is maximal. Regarding the admissibility of θ^ε we note that, by construction, it is nonnegative and has the correct average. It remains only to check whether it is bounded above by 1. One scenario in which this bound is ensured is when Ω is smooth (in which case $\phi_1 \in C^1(\overline{\Omega})$) and ε and γ are sufficiently small. Finally, we remark that (4.10) provides a nice refinement of Proposition 4.5 in that it expresses, in terms of the Dirichlet fraction, γ, the rate at which $\xi_1^\varepsilon(\hat{\theta}^\varepsilon)$ approaches $\lambda_1(\Omega)$.

5. Algorithms. We confine the design, θ, and the eigenfunction, u, to finite-dimensional spaces and so arrive at optimization problems amenable to a computer.

We write $\partial\Omega$ as the closure of the disjoint union of m open edges, $\{\Gamma_j\}_{j=1}^m$, and then restrict θ to

$$\theta(s) = \sum_{j=1}^m \Theta_j 1_{\Gamma_j}(s),$$

where $\Theta \in \mathbb{R}^m$ satisfies the box constraints

$$0 \le \Theta_j \le 1, \quad j = 1, \ldots, m, \tag{5.1}$$

and the integral constraint

$$\sum_{j=1}^m \Theta_j |\Gamma_j| = \gamma |\partial\Omega|. \tag{5.2}$$

To compute ξ_1^ε at such a θ we restrict our search to eigenvectors of the form

$$u(x) = \sum_{i=1}^p U_i T_i(x),$$

where $p < \infty$ and the T_i comprise a so-called Galerkin basis for a p-dimensional subspace of $H^1(\Omega)$. On substituting this expansion into the weak form (1.3) with v running through the T_i we arrive at the $p \times p$ eigensystem

$$(K + \varepsilon^{-1} Q(\Theta))U = \Xi M U, \tag{5.3}$$

where K and M are independent of Θ while

$$Q_{ij}(\Theta) = \int_{\partial\Omega} \theta T_i T_j \, ds = \sum_{k=1}^m \Theta_k \int_{\Gamma_k} T_i T_j \, ds. \tag{5.4}$$

Let us denote the least eigenvalue of (5.3) by $\Xi_1^\varepsilon(\Theta)$. As this approximation procedure respects the symmetry of the original problem we retain a variational characterization,

$$\Xi_1^\varepsilon(\Theta) = \min_{\langle MU,U\rangle=1} \mathcal{R}_\varepsilon(U,\Theta), \qquad \mathcal{R}_\varepsilon(U,\Theta) \equiv \langle (K + \varepsilon^{-1} Q(\Theta))U, U\rangle. \tag{5.5}$$

As $\Theta \mapsto Q(\Theta)$ is linear it follows from (5.5) that $\Theta \mapsto \Xi_1^\varepsilon(\Theta)$ is concave. Now, denoting by AD_γ^* those $\Theta \in \mathbb{R}^m$ satisfying (5.1) and (5.2), we may pose the finite-dimensional optimization problems

$$\min_{\Theta \in AD_\gamma^*} \Xi_1^\varepsilon(\Theta) \quad \text{and} \quad \max_{\Theta \in AD_\gamma^*} \Xi_1^\varepsilon(\Theta).$$

As AD_γ^* is compact and convex and Ξ_1^ε is bounded and concave it follows that $\Theta \mapsto \Xi_1^\varepsilon(\Theta)$ attains its minimum at an extreme point of AD_γ^*, i.e., on AD_γ, those $\Theta \in AD_\gamma^*$ each component of which is either zero or one.

Let us now turn to the gradient of $\Theta \mapsto \Xi_1^\varepsilon(\Theta)$. For well-chosen basis functions, e.g., piecewise linear hats, it can be shown that $\Xi_1^\varepsilon(\Theta) \to \xi_1^\varepsilon(\theta)$ as m and p approach ∞. In particular, $\Xi_1^\varepsilon(\Theta)$ is simple for sufficiently large m and p. As a result we may apply the finite-dimensional analogue of Proposition 4.1,

$$\frac{\partial \Xi_1^\varepsilon(\Theta)}{\partial \Theta_k} = \frac{1}{\varepsilon} \left\langle \frac{\partial Q(\Theta)}{\partial \Theta_k} U_1^\varepsilon, U_1^\varepsilon \right\rangle, \tag{5.6}$$

where the associated eigenvector, U_1^ε, is normalized according to $\langle MU_1^\varepsilon, U_1^\varepsilon\rangle = 1$. The implementation of (5.6), in particular the application of $\partial Q(\Theta)/\partial\Theta_k$, requires a careful accounting of the assembly of Q. Recalling (5.4) we find

$$\frac{\partial Q_{ij}(\Theta)}{\partial \Theta_k} = \int_{\Gamma_k} T_i T_j \, ds.$$

To begin, let us evaluate these integrals under the assumption that Γ_k is the interval $[a,b]$ and that this interval is partitioned by the first components of the grid points $x_i = (s_i, 0)$, i.e.,

$$a = s_1 < s_2 < \cdots < s_{n-1} < s_n = b.$$

We also suppose that $T_i(x_j) = \delta_{ij}$ and that T_i is piecewise linear. As a result

$$\int_{\Gamma_k} T_i T_j \, ds = \frac{1}{3}\begin{cases} |s_1 - s_2| & \text{if } i = j = 1, \\ |s_{i-1} - s_i| + |s_i - s_{i+1}| & \text{if } 1 < i = j < n, \\ |s_{n-1} - s_n| & \text{if } i = j = n, \\ |s_i - s_j|/2 & \text{if } |i - j| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Substituting the above into (5.6) we find

$$\frac{\partial \Xi_1^\varepsilon(\Theta)}{\partial\Theta_k} = \frac{1}{3\varepsilon}\sum_{i=1}^{n-1}\left\{(U_1^\varepsilon)_i^2 + (U_1^\varepsilon)_i(U_1^\varepsilon)_{i+1} + (U_1^\varepsilon)_{i+1}^2\right\}|s_{i+1} - s_i|.$$

In the general case, i.e., where the T_i remain piecewise linear although Γ_k may be a planar segment whose edges and grid points are ordered by a black-box grid generator (as in MATLAB's PDE toolbox), the gradient takes the form

$$\text{(5.7)} \quad \frac{\partial \Xi_1^\varepsilon(\Theta)}{\partial\Theta_k} = \frac{1}{3\varepsilon}\sum_{i\in\mathcal{I}_k}\langle U_1^\varepsilon\rangle_i |\omega_i|, \qquad \langle U_1^\varepsilon\rangle_i \equiv (U_1^\varepsilon)_{\omega_i^+}^2 + (U_1^\varepsilon)_{\omega_i^+}(U_1^\varepsilon)_{\omega_i^-} + (U_1^\varepsilon)_{\omega_i^-}^2,$$

where $\mathcal{I}_k$ is the set of indices of mesh edges ω_i contained in Γ_k and $\omega_i^\pm$ are the indices of the grid points constituting the endpoints of ω_i. From here it is a simple matter to derive the finite-dimensional analogues of our pointwise optimality conditions. In particular, if each Γ_k corresponds to a single mesh edge and $\check{\Theta}^\varepsilon \in AD_\gamma$ is a classical minimizer of Ξ_1^ε and $\check{U}_1^\varepsilon$ its associated eigenvector, then there exists a ν such that

$$\text{(5.8)} \quad \begin{aligned} \check{\Theta}_k^\varepsilon = 0 &\Rightarrow \langle \check{U}_1^\varepsilon\rangle_k > \nu, \\ \check{\Theta}_k^\varepsilon = 1 &\Rightarrow \langle \check{U}_1^\varepsilon\rangle_k < \nu. \end{aligned}$$

These conditions are reminiscent of those that arise in Krein's problem of the optimal distribution of mass; see, e.g., Cox [7]. As such we apply the simple alternating search strategy of [7] to our minimum problem. More precisely, given $\Theta^{(j)} \in AD_\gamma$,

(I) compute $U^{(j)}$, the minimizer of $U \mapsto \mathcal{R}_\varepsilon(U, \Theta^{(j)})$ subject to $\langle MU, U\rangle = 1$.

(II) compute $\Theta^{(j+1)}$, the minimizer of $\Theta \mapsto \mathcal{R}_\varepsilon(U^{(j)}, \Theta)$ subject to $\Theta \in AD_\gamma$.

(III) if $\Theta^{(j+1)} \neq \Theta^{(j)}$, then set $j = j + 1$ and go to (I).

The implementation of (I) simply requires the solution of (5.3) with $\Theta = \Theta^{(j)}$. The optimality conditions (5.8) animate the implementation of (II). More precisely, we compute $\mathcal{J} \equiv \{k : \langle U^{(j)}\rangle_k < \nu\}$, where ν is chosen in such a way that

$$\sum_{k\in\mathcal{J}} |\Gamma_k| = \gamma|\partial\Omega|,$$

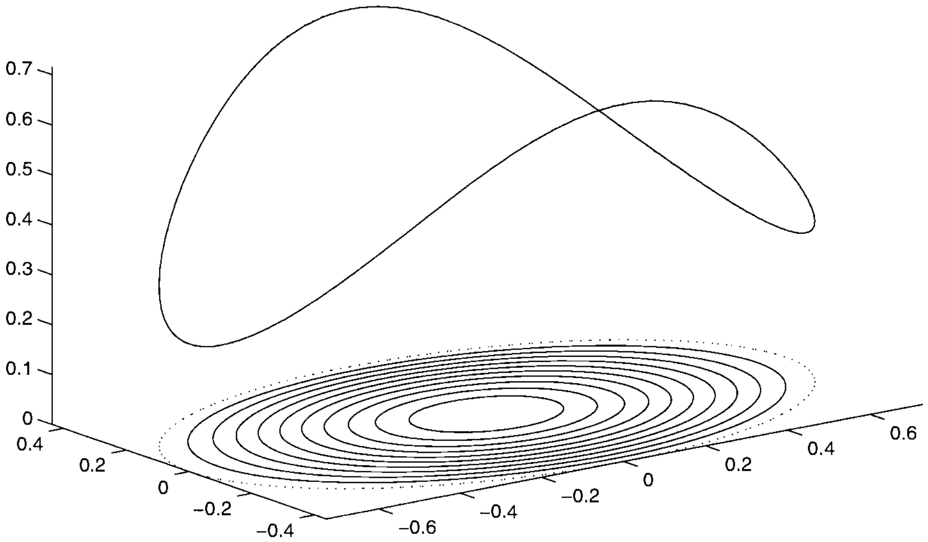


FIG. 2. *The limiting maximal fastener,* Φ.

and then define

$$\Theta_k^{(j+1)} = \begin{cases} 1 & \text{if } k \in \mathcal{J}, \\ 0 & \text{otherwise.} \end{cases}$$

This completes our description of the minimization algorithm.

With respect to the maximization problem, recalling that we have a smooth, concave function subject only to box and linear constraints, we may invoke any of a number of standard optimization packages.

6. Numerical results. For the maximization of Ξ_1^ε we used the `constr` function found in MATLAB's optimization toolbox. The assembly of (5.3) and the computation of Ξ_1^ε and U_1^ε were carried out by the `pdeeig` function found in MATLAB's PDE toolbox. Given U_1^ε we coded the gradient computation (5.7) ourselves. We present here the results of our computations for two representative domains.

In the first case we consider the drumhead whose boundary is the ellipse

$$\frac{x^2}{25} + \frac{y^2}{9} = \frac{1}{16}.$$

Recalling the discussion at the close of section 4 we expect the maximizer, $\hat{\Theta}^\varepsilon$, as $\varepsilon \to 0$, to coincide with

$$\Phi \equiv \gamma \frac{\partial \phi_1}{\partial n} \Big/ \overline{\frac{\partial \phi_1}{\partial n}},$$

the product of γ and the normalized normal derivative of the first Dirichlet eigenfunction of the ellipse. For the purpose of illustration, in Figure 2 we have plotted the underlying ellipse, the contours of the associated ϕ_1, and the graph of its corresponding Φ, with $\gamma = 1/2$. The eigenfunction was computed at the $p = 96545$ vertices of 191488 triangles. The boundary was partitioned into $m = 100$ edges and the associated Dirichlet eigenvalue was 20.45. Next, we set $\varepsilon = 10^r$, let r range from

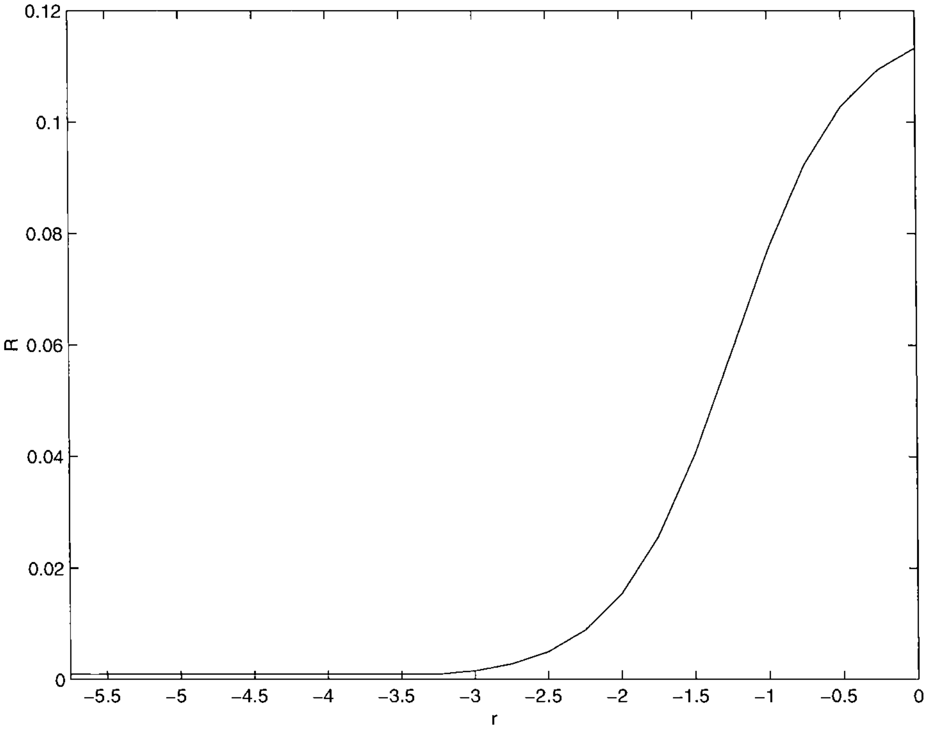


FIG. 3. $\|\Phi - \hat{\Theta}^{\varepsilon}\|_{\infty}$ *as* $\varepsilon \to 0$.

0 to −6, and denote by $\hat{\Theta}^{10^r}$ the maximizer returned by `constr` on the grid quoted above using the default stopping criteria. We measured the pointwise distance from $\hat{\Theta}^{10^r}$ to Φ via

$$R(r) \equiv \|\Phi - \hat{\Theta}^{10^r}\|_{\infty} \equiv \max_k |\Phi_k - \hat{\Theta}_k^{10^r}|$$

and have recorded its graph in Figure 3. That no improvement is seen for $\varepsilon < 10^{-3}$ is most likely due to the fact that our computed Φ is itself accurate only to 10^{-2}.

As a nonconvex example, we pursue the maximizer over the L-shaped region familiar to users of MATLAB. It is well known (see, e.g., Fox, Henrici, and Moler [11]) that the gradient of the first Dirichlet eigenfunction is not bounded in a neighborhood of the reentrant corner. As a result, we may not expect (4.11) to hold along the entire boundary. In Figure 4 we have plotted $\hat{\Theta}^{\varepsilon}$, the maximizer returned by `constr` along with the level sets of its corresponding eigenfunction. Working over a grid of $p = 49665$ vertices, 97792 triangles, and $m = 192$ boundary segments with $\varepsilon = 10^{-3}$ and $\gamma = 1/2$ we found $\xi_1^{\varepsilon}(\hat{\Theta}^{\varepsilon}) \approx 9.59$. Note that the level sets indeed resemble those of the first Dirichlet eigenfunction and that $\hat{\Theta}^{\varepsilon}$ behaves like a clipped version of its normal derivative.

Finally, we wish to present numerical results for the minimization problem. As above, we concentrate on the ellipse and the L. With respect to the former we offer in Figures 5 and 6, respectively, the initial iterate supplied to, and final iterate delivered by, the alternating search minimization algorithm presented at the close of the previous section. The domain was approximated by 13374 triangles with $p = 7288$ vertices.

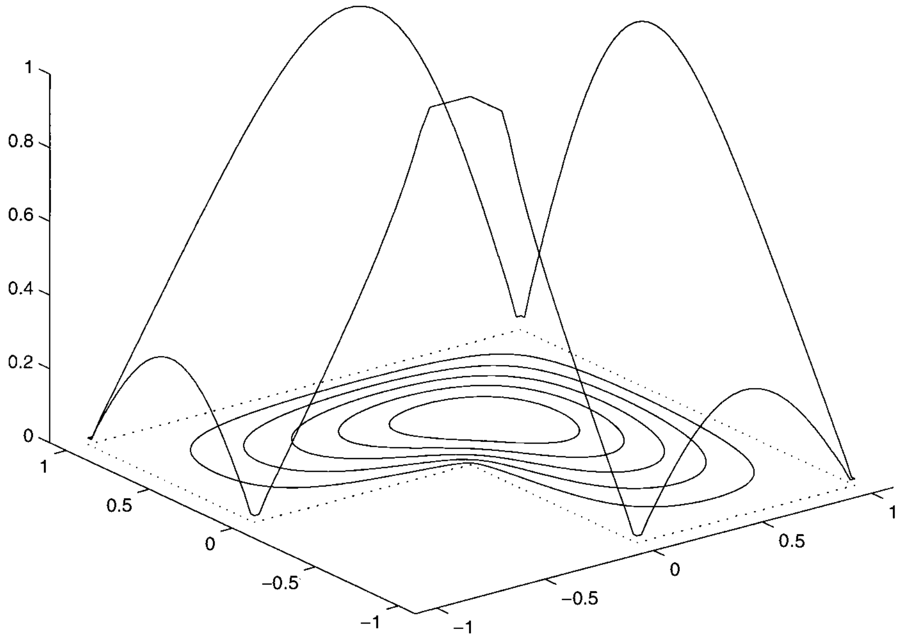


FIG. 4. *Maximal fastening of the L.*

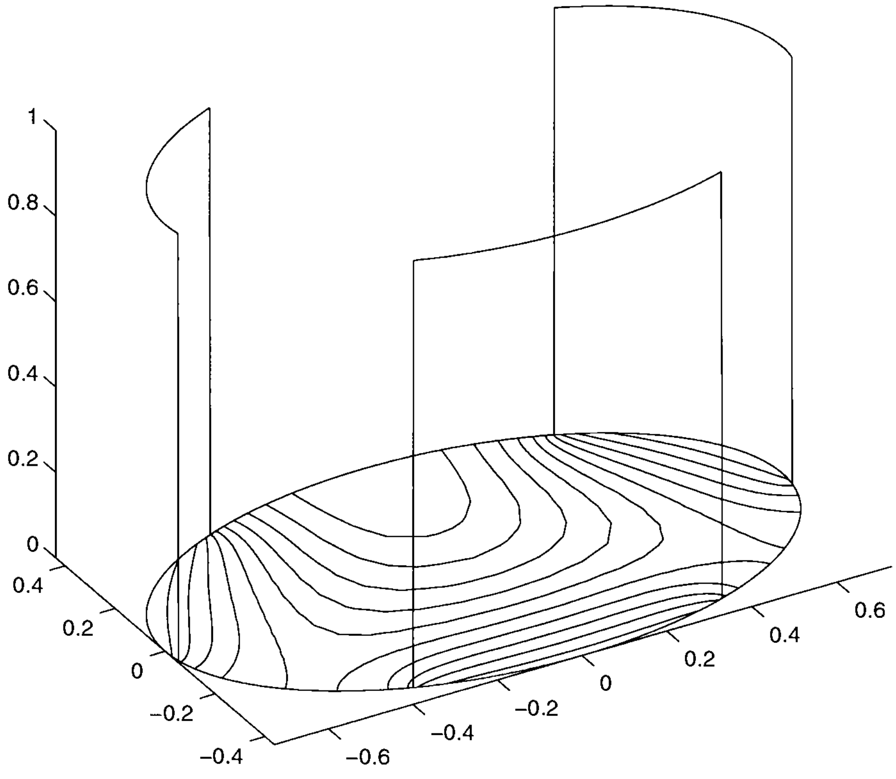


FIG. 5. *Initial iterate.*

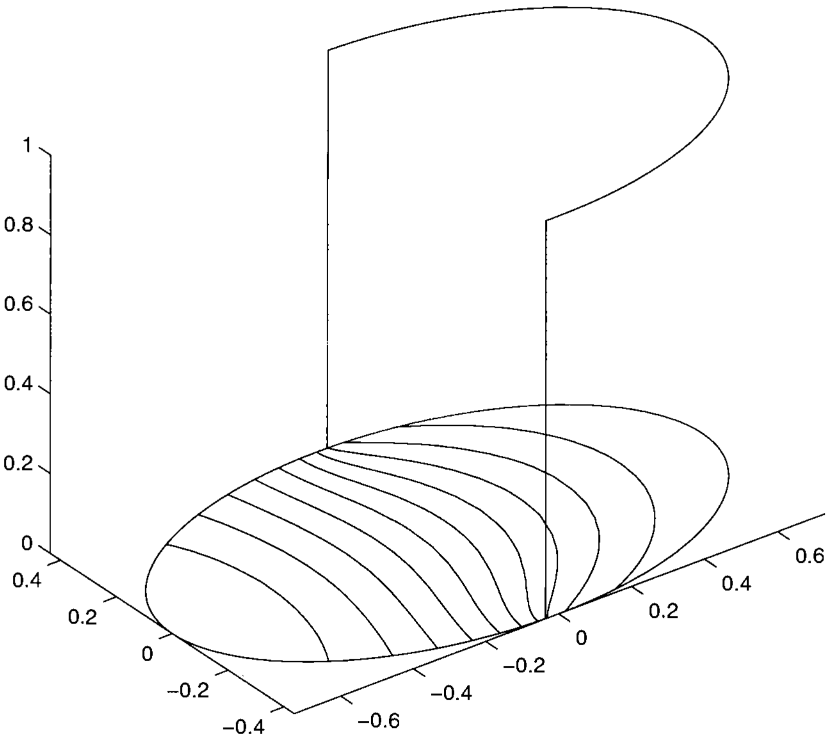


FIG. 6. *Final iterate.*

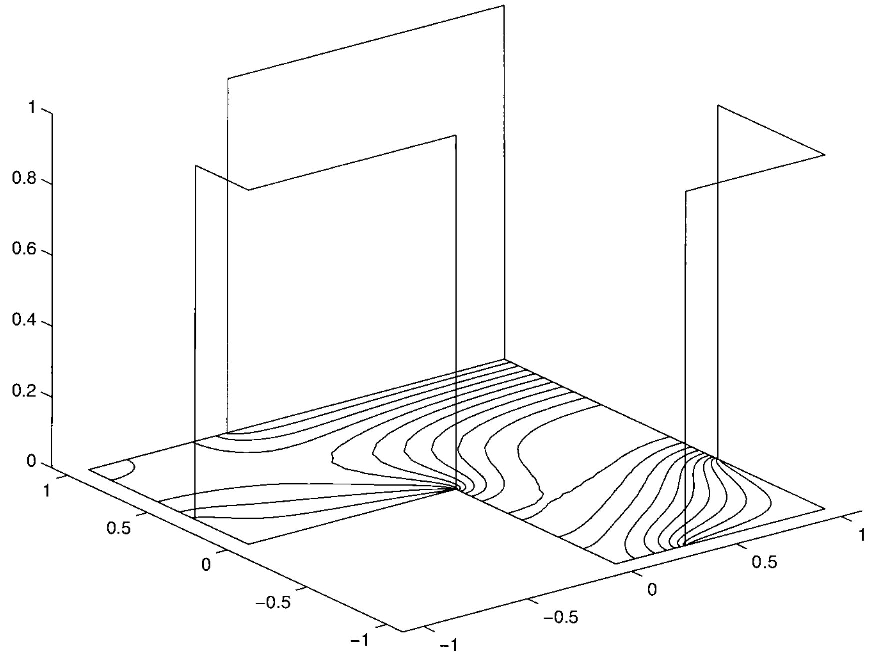


FIG. 7. *Initial iterate.*

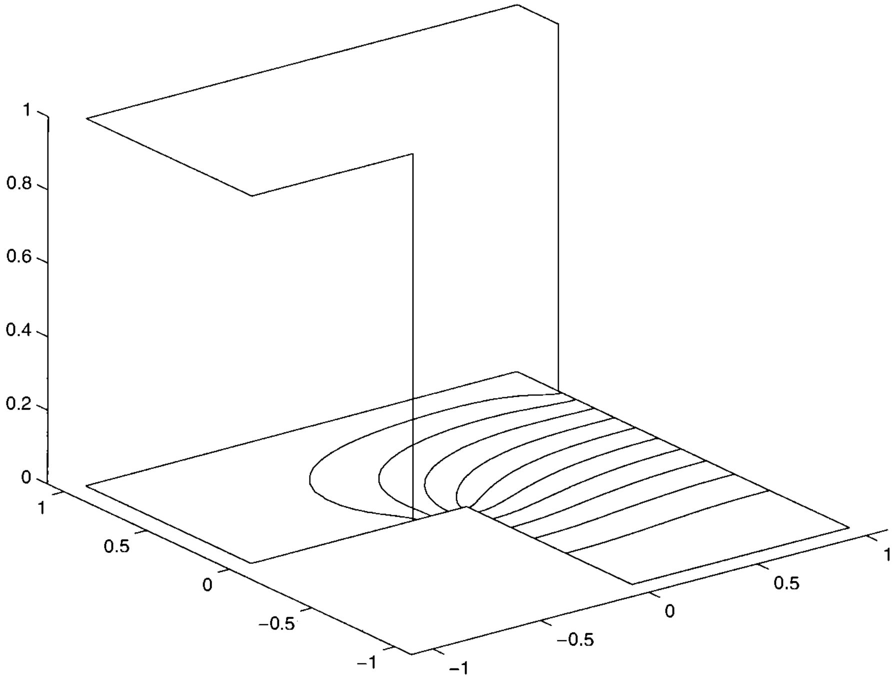


Fig. 8. *Final iterate.*

Its boundary was partitioned into $m = 1200$ edges. With $\gamma = 1/2$ and $\varepsilon = 0.1$ the algorithm came to rest in 69 iterations. The eigenvalue, 6.68, of the initial iterate was diminished to 3.07. In both cases we have also plotted the contours of the associated eigenfunction.

The initial and final iterates, along with the contours of their associated eigenfunctions, for the L-shaped drum are depicted in Figures 7 and 8. In this case the domain was approximated by 18238 triangles with $p = 9936$ edges. Its boundary was partitioned into $m = 1632$ edges. With $\gamma = 1/2$ and $\varepsilon = 0.01$ the algorithm came to rest in 31 iterations and reduced the eigenvalue of the initial iterate, 4.08, to 0.88. We note that the final iterate pulled the Dirichlet data away from the reentrant corner and wrapped it around the outer corner. The resulting eigenvalue is indeed less than 1.09, the eigenvalue of the L with Dirichlet data on the three legs above the diagonal $x = y$.

REFERENCES

[1] C. Bandle, *Isoperimetric Inequalities and Applications*, Pitman, Boston, 1980.

[2] H. Bauer, *Sur le prolongement des formes linéaires positives dans un espace vectorial ordonné*, C.R. Acad. Sci. Paris, 244 (1957), pp. 289–292.

[3] M.-H. Bossel, *Membranes elastiquement liées inhomogènes ou sur une surface: Une nouvelle extension de theoréme isoperimetrique de Rayleigh-Faber-Krahn*, Z. Angew. Math. Phys., 39 (1988), pp. 733–742.

[4] G. Buttazzo, *Thin insulating layers: The optimization point of view*, in Material Instabilities in Continuum Mechanics and Related Mathematical Problems, J.M. Ball, ed., Oxford University Press, Oxford, UK, 1988, pp. 11–19.

[5] C.H. Chuang and G.J.W. Hou, *Eigenvalue sensitivity analysis of planar frames with variable joint and support locations*, AIAA J., 30 (1992), pp. 2138–2147.

[6] F. Clarke, *Optimization and Nonsmooth Analysis*, 2nd ed., Classics Appl. Math. 5, SIAM, Philadelphia, PA, 1990.

[7] S.J. Cox, *The two–phase drum with the deepest bass note*, Japan J. Indust. Appl. Math., 8 (1991), pp. 345–355.

[8] S.J. Cox, *The generalized gradient at a multiple eigenvalue*, J. Funct. Anal., 33 (1995), pp. 30–40.

[9] S.J. Cox and B. Kawohl, *Circular symmetrization and extremal Robin conditions*, Z. Angew. Math. Phys., 50 (1999), pp. 301–311.

[10] J. Denzler, *Windows of given area with minimal heat diffusion*, Trans. Amer. Math. Soc., 351 (1999), pp. 569–580.

[11] L. Fox, P. Henrici, and C. Moler, *Approximations and bounds for eigenvalues of elliptic operators*, SIAM J. Numer. Anal., 4 (1967), pp. 89–102.

[12] S. Friedland, *Extremal eigenvalue problems defined for certain classes of functions*, Arch. Rational Mech. Anal., 67 (1977), pp. 73–81.

[13] Matlab User's Guide, The Math Works Inc., Natick, MA, 1996.

[14] Z. Mroz and G.I.N. Rozvany, *Optimal design of structures with variable support conditions*, J. Optim. Theory Appl., 15 (1975), pp. 85–101.

[15] G.A. Philippin, *Some remarks on the elastically supported membrane*, Z. Angew. Math. Phys., 29 (1978), pp. 306–314.

[16] J. Serrin, *A symmetry problem in potential theory*, Arch. Rational Mech. Anal., 43 (1971), pp. 304–318.

[17] R. Sperb, *An isoperimetric inequality for the first eigenvalue of the Laplacian under Robin boundary conditions*, in General Inequalities 6, W. Walter, ed., Birkhauser, Basel, 1992, pp. 361–367.

SIAM J. Optim.
Vol. 9, No. 4, pp. 965–990

A GLOBAL CONVERGENCE THEORY FOR DENNIS, EL-ALEM, AND MACIEL'S CLASS OF TRUST-REGION ALGORITHMS FOR CONSTRAINED OPTIMIZATION WITHOUT ASSUMING REGULARITY[*]

MAHMOUD EL-ALEM[†]

To John Dennis on the occasion of his 60th birthday.

Abstract. This work presents a convergence theory for Dennis, El-Alem, and Maciel's class of trust-region-based algorithms for solving the smooth nonlinear programming problem with equality constraints. The results are proved under very mild conditions on the quasi-normal and tangential components of the trial steps. The Lagrange multiplier estimates and the Hessian estimates are assumed to be bounded. No regularity assumption is made. In particular, linear independence of the gradients of the constraints is not assumed. The theory proves global convergence for the class. In particular, it shows that a subsequence of the iteration sequence satisfies one of four types of Mayer–Bliss stationary conditions in the limit.

Key words. nonlinear programming, equality constrained problems, constrained optimization, global convergence, regularity assumption, augmented Lagrangian, Mayer–Bliss points, stationary points, quasi-normal step, trust region

AMS subject classifications. 65K05, 49D37

PII. S1052623497331762

1. Introduction.

Over the last two decades, trust-region algorithms have enjoyed a good reputation on the basis of their remarkable numerical reliability in conjunction with a sound and complete convergence theory. They have proven to be very effective and robust techniques for solving unconstrained and equality constrained optimization problems.

The first trust-region algorithm was given by Levenberg [27] and later was rederived by Marquardt [30]. The algorithm was designed for solving nonlinear least-squares problems. Powell [39] derived from the Levenberg–Marquardt method the first trust-region algorithm for solving the unconstrained minimization problem. Detailed discussion of the Levenberg–Marquardt method can be found in Moré [35], and discussion of the trust-region method for solving the unconstrained optimization problem can be found in Dennis and Schnabel [14], Fletcher [23], and Shultz, Schnabel, and Byrd [42].

Since the mid 1980s, many authors have considered extending the trust-region idea to the following equality constrained optimization problem:

$$(\mathrm{EQ}) \equiv \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & C(x) = 0. \end{cases}$$

The functions $f : \Re^n \to \Re$ and $C : \Re^n \to \Re^m$ are at least twice continuously differentiable, where $m < n$.

[*] Received by the editors December 23, 1997; accepted for publication (in revised form) March 8, 1999; published electronically September 24, 1999. This research was done while the author was visiting the Center for Research on Parallel Computations, Rice University, Houston, TX. The author was supported by NSF Cooperative Agreement CCR-9120008 as a visiting member of the Center for Research on Parallel Computation.

http://www.siam.org/journals/siopt/9-4/33176.html

[†] Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt (elalem@alex.eun.eg, mmelalem@hotmail.com).

Most trust-region algorithms for solving problem (EQ) try to combine the trust-region idea with the successive quadratic programming (SQP) method. In general, the SQP method iteratively minimizes a quadratic model of the Lagrangian function

$$\ell(x,\lambda) = f(x) + \lambda^T C(x), \tag{1.1}$$

where λ is the Lagrange multiplier vector, subject to a linear approximation of the constraints. At each iteration k, the SQP method obtains a step s_k^{QP} and an associated Lagrange multiplier step $\Delta\lambda_k^{QP}$ by solving the following quadratic programming subproblem:

$$\begin{array}{ll} \text{minimize} & \nabla_x \ell(x_k,\lambda_k)^T s + \frac{1}{2} s^T H_k s \\ \text{subject to} & C(x_k) + \nabla C(x_k)^T s = 0, \end{array}$$

where H_k is the Hessian of the Lagrangian function (1.1) at (x_k, λ_k) or an approximation to it.

If a trust-region constraint is simply added to the quadratic programming subproblem, the resulting trust-region subproblem may be infeasible because the trust-region constraint and the hyperplane $C(x_k)+\nabla C(x_k)^T s = 0$ may have no intersecting points. In other words, the two constrained sets may be disjoint. Even if they intersect, there is no guarantee that when the trust-region radius δ_k is decreased, the above subproblem remains feasible. Note that the global convergence of the trust-region methods is based on being able to reduce δ_k until the model trust-region subproblem accurately represents the actual problem.

To avoid possible infeasibility in the subproblem, different approaches have been proposed. The first approach is to relax the linear constraints in such a way that the resulting feasible set is nonempty. In particular, the hyperplane $C(x_k)+\nabla C(x_k)^T s = 0$ is replaced by the relaxed hyperplane $\nu_k C(x_k) + \nabla C(x_k)^T s = 0$, where $\nu_k \in [0,1]$. This approach was first suggested by Miele, Huang, and Heideman [33] in the context of a line-search globalization strategy for solving problem (EQ) (see also Miele, Cragg, and Levy [32] and Miele, Levy, and Cragg [34]). It was later used to obtain a feasible trust-region subproblem by Vardi [43], Byrd, Schnabel, and Schultz [10], and El-Hallabi [21].

A major difficulty with this approach lies in the problem of choosing ν_k so that a feasible trust-region subproblem is guaranteed. This difficulty makes this approach impractical.

The second approach for resolving this infeasibility was proposed by Celis, Dennis, and Tapia [12]. They replaced the linear constraints by the quadratic constraint $\|C(x_k) + \nabla C(x_k)^T s\|_2^2 \le \theta_k$, where θ_k is a given parameter chosen to ensure that the resulting trust-region subproblem is always feasible. This approach was used by El-Alem [17] and Powell and Yuan [41]. The parameter θ_k is also chosen to ensure a sufficient decrease in the quadratic model of the linearized constraints. This decrease is at least a fraction of the decrease obtained by the Cauchy step, which is defined to be the minimizer of $\|C(x_k) + \nabla C(x_k)^T s\|_2^2$ inside the trust region in the steepest descent direction.

In Celis, Dennis, and Tapia [12] and El-Alem [17], the parameter θ_k was taken to be

$$\theta_k = (1-\hat{\nu})\|C(x_k)\|_2^2 + \hat{\nu}\|C(x_k) + \nabla C(x_k)^T s_k^{cp}\|_2^2$$

for some fixed $\hat{\nu} \in (0,1)$, where s_k^{cp} is the Cauchy step. In Powell and Yuan [41], the choice of θ_k was

$$\theta_k = \{\min \|C(x_k) + \nabla C(x_k)^T s\|_2^2 \ : \ \underline{\nu}\ \delta_k \le \|s\|_2 \le \bar{\nu}\delta_k\},$$

where $0 < \underline{\nu} \le \bar{\nu} \le 1$ and δ_k is the trust-region radius.

A major disadvantage with this approach lies in the fact that the resulting trust-region subproblem has two quadratic constraints, so that there is no efficient algorithm for finding a good approximation to the solution of this subproblem. Although Williamson [45] has attempted to produce an efficient algorithm by computing an inexact solution of the subproblem and others have suggested algorithms to solve special cases of this subproblem (see El-Alem and Tapia [20], Yuan [46], [47], and Zhang [51]), the results are not, in general, satisfactory. This approach will remain impractical until an efficient way of solving the trust-region subproblem is discovered.

The reduced Hessian technique is another approach to overcoming the difficulty of having an infeasible trust-region subproblem. This approach was suggested by Byrd [9] and Omojokun [37]. In this approach, the trial step s_k is decomposed into two components: the tangential component s_k^t and the normal component s_k^n. The step s_k^n is computed by solving the following trust-region subproblem:

$$\begin{array}{ll} \text{minimize} & \|C(x_k) + \nabla C(x_k)^T s^n\|_2^2 \\ \text{subject to} & \|s^n\|_2 \le \nu\delta_k \end{array}$$

for some $\nu \in (0,1)$. The tangential component s_k^t is then obtained by solving another trust-region subproblem. Let Z_k be a matrix that forms an orthonormal basis for the null space of $\nabla C(x_k)^T$ and let $s_k^t = Z_k \bar{s}_k^t$. The step $\bar{s}_k^t$ is computed by solving the following trust-region subproblem:

$$\begin{array}{ll} \text{minimize} & [Z_k^T(\nabla_x \ell(x_k,\lambda_k) + H_k s_k^n)]^T \bar{s}^t + \frac{1}{2}(\bar{s}^t)^T Z_k^T H_k Z_k \bar{s}^t \\ \text{subject to} & \|Z_k \bar{s}^t\|_2^2 \le \delta_k^2 - \|s_k^n\|_2^2. \end{array}$$

The trial step s_k has the form $s_k = s_k^n + Z_k \bar{s}_k^t$.

This approach has been used by many authors. See, for example, Alexandrov [1], [2], Alexandrov and Dennis [3], Biegler, Nocedal, and Schmid [4], Dennis and Vicente [15], El-Alem [18], [19], Lalee [25], Lalee, Nocedal, and Plantenga [26], Maciel [28], Plantenga [38], Vicente [44], and Zhang and Zhu [50].

One of the advantages of this approach is that the two trust-region subproblems are similar to the trust-region subproblem for the unconstrained case.

Dennis, El-Alem, and Maciel [13] have considered a general class of trust-region-based algorithms for solving problem (EQ). In their algorithms, the two components of the trial step are not necessarily orthogonal. We present this class of algorithms in the next section.

In unconstrained optimization, the use of a trust region has made it possible to make strong guarantees of convergence. In particular, to guarantee global convergence, it is not necessary to require that the Hessian approximation be positive definite or even well conditioned, but only that it be uniformly bounded. To ensure global convergence, the step is required only to satisfy the fraction of Cauchy decrease condition; that is, the step must produce at least a fraction of the decrease obtained by the Cauchy step.

Powell [40] proved a powerful theorem. He showed that if the sequence of iterates generated by the algorithm satisfies the fraction of Cauchy decrease condition and if the sequence of Hessian approximations is bounded, then

$$\liminf_{k\to\infty} \|\nabla f(x_k)\|_2 = 0.$$

Powell's theorem does not prove convergence to a solution of the unconstrained problem. It proves only that a subsequence of the sequence of gradients of the objective function converges to zero. The strength of this result, however, comes from the weak assumptions imposed on the sequence of local models. Detailed discussion about the convergence results of trust-region algorithms for unconstrained optimization can be found in Carter [11], Moré [36], and Shultz, Schnabel, and Byrd [42].

Many authors have established global convergence results for algorithms that have been suggested for solving problem (EQ). El-Alem [17] and Powell and Yuan [41] have proved global convergence for variants of the Celis, Dennis, and Tapia trust-region algorithm by showing that

$$\liminf_{k\to\infty} \{\|Z_k^T \nabla f(x_k)\|_2 + \|C(x_k)\|_2\} = 0.$$

Analogous to Powell's theorem for the unconstrained case, Dennis, El-Alem, and Maciel [13] proved for their class of algorithms that

$$\liminf_{k\to\infty} \{\|W_k^T \nabla f(x_k)\|_2 + \|C(x_k)\|_2\} = 0,$$

where W_k is a matrix that forms a basis (not necessarily orthogonal) for the null space of $\nabla C(x_k)^T$.

In Dennis, El-Alem, and Maciel's class of algorithms, the local model of the problem is generally taken to be a linear model of the constraints and a quadratic model of the Lagrangian function. The information in the local model depends on the Lagrange multiplier estimates as well as the second-order information. Analogous to Powell's theorem, Dennis, El-Alem, and Maciel only require the boundedness of the sequences of model Lagrange multipliers and Hessians. The results of Dennis, El-Alem, and Maciel were proved under very mild conditions on the quasi-normal and tangential components of the trial steps. However, their results were proved under the linear independence assumption.

In this paper, we reduce Dennis, El-Alem, and Maciel's assumptions even further and still obtain similar global convergence results. In our theory, the linear independence assumption on the gradients of the constraints is not made. Our theory is so general that it holds for any algorithm that uses the augmented Lagrangian as a merit function, the El-Alem scheme for updating the penalty parameter [17], and bounded Lagrange multiplier and Hessian estimates. Similar results for related algorithms in an abstract setting were established by Burke [6], [8].

The following notation is used throughout the rest of the paper. The sequence of points generated by the algorithm is denoted by $\{(x_k, \lambda_k)\}$. We abbreviate $f(x_k)$ as f_k, $\ell(x_k, \lambda_k)$ as ℓ_k, and so on. However, $f(x)$ is not abbreviated when emphasizing the dependence of f on x. We use the same symbol 0 to denote the real number zero, the zero vector, and the zero matrix. Finally, all norms used in this paper are l_2-norms.

The organization of the paper is as follows. In section 2, we present in detail all the components of the general class of trust-region-based algorithms suggested by Dennis, El-Alem, and Maciel [13]. An overall summary of the class is presented at the end of this section. In section 3, we state the assumptions under which we prove global convergence. The main results of this paper show that the algorithm generates a sequence of iterates that has a subsequence that asymptotically satisfies one of four types of stationary conditions for problem (EQ). In section 4, we identify these conditions, state their definitions, and demonstrate some of their properties. In section 5, we state our main global convergence results. Our convergence theory is presented in sections 6–8. Finally, section 9 contains concluding remarks.

2. General trust-region-based algorithms.

In this section, we present the class of algorithms suggested by Dennis, El-Alem, and Maciel [13] for solving problem (EQ). This is a general class of trust-region-based algorithms. The basic idea of the trust-region algorithms is as follows. Approximate the problem by a model trust-region subproblem. The trial step is obtained by solving this subproblem. Test for accepting or rejecting the trial step and update the trust-region radius accordingly. If the step is rejected, decrease the radius of the trust region and compute another one using the new value of the trust-region radius. To test the trial steps, a merit function must be employed. Such a merit function often involves a parameter, usually called the penalty parameter. This parameter is updated using an updating scheme. More details about the trust-region method for constrained optimization can be found in Dennis, El-Alem, and Maciel [13].

In any trust-region algorithm for solving problem (EQ), there are four important issues to be considered. At each iteration k, we must first compute a trial step, and we address this issue in section 2.1. Once the step is computed, we will need a criteria for accepting the trial step. Section 2.2 is devoted to this subject. To test the step, the penalty parameter needs to be updated. We address this issue in section 2.3. Finally, we need a procedure for updating the trust-region radius; it is presented in section 2.4.

An overall summary of the algorithm is presented in section 2.5.

2.1. Computing a trial step.

We do not present a particular way to compute the trial steps. Instead, we present some conditions the steps must satisfy. Let s_k be decomposed into two components: the tangential component s_k^t and the quasi-normal component s_k^n. The trial step will then have the form $s_k = s_k^t + s_k^n$. Observe that the two components of the trial step are not necessarily orthogonal.

If $C_k \neq 0$, then the quasi-normal component s_k^n of the trial step is required to produce at least a fraction of the decrease in the quadratic model of the linearized constraints obtained by the Cauchy step. The Cauchy step s_k^{cp} is the step that solves the following problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|\nabla C_k^T s + C_k\|^2 \\ \text{subject to} & \|s\| \leq \tau\delta_k, \\ & s = -n_k^{cp}\nabla C_k C_k, \qquad n_k^{cp} > 0. \end{array}$$

So, the quasi-normal component s_k^n is chosen such that it satisfies, for some $r_1 \in (0, 1]$,

$$\|C_k\|^2 - \|C_k + \nabla C_k^T s^n\|^2 \geq r_1\{\|C_k\|^2 - \|C_k + \nabla C_k^T s_k^{cp}\|^2\}. \tag{2.1}$$

Dennis, El-Alem, and Maciel require that the quasi-normal component s_k^n of the trial step satisfy, at every iteration k,

$$\|s_k^n\| \leq K\|C_k\|, \tag{2.2}$$

where K is a positive constant. This condition is needed to obtain Dennis, El-Alem, and Maciel's global convergence results. It says that when the current point is close to the feasible region, the normal step must be short. It can be viewed as a relaxation to the orthogonality condition of s_k^n and s_k^t.

Condition (2.2) on the normal step is inappropriate when the regularity assumption is not made. It can contradict the Cauchy decrease condition (2.1) imposed on s_k^n. Let us consider the following example. If $C(x) = x^2$, then the Cauchy step $s_k^{cp} = -\frac{x_k}{2}$, which decreases the quadratic model by x_k^4. However, any step satisfying (2.2) satisfies $\|s_k^n\| \leq Kx_k^2$ and will decrease the quadratic model by at most $2|C_k^T \nabla C_k^T s_k^n| \leq 4Kx_k^5$. This is arbitrarily poor compared to the decrease obtained by the Cauchy step when x_k is small. For this reason, we modify condition (2.2) to be

$$\|s_k^n\| \leq K \|s_k^{mn}\|, \tag{2.3}$$

where s_k^{mn} is the minimum-norm solution of

$$\begin{array}{ll} \text{minimize} & \|\nabla C_k^T s + C_k\|^2 \\ \text{subject to} & \|s\| \leq \tau \delta_k \end{array} \tag{2.4}$$

for some $\tau \in (0,1)$, where δ_k is the trust-region radius.

Condition (2.3) deals with the above example with no difficulty. This condition is equivalent to condition (2.2) whenever ∇C_k has full column rank and allows the full SQP step to be taken when it is inside the trust region. We are indebted to Richard Byrd for informing us of the importance of using condition (2.3) instead of (2.2) when the regularity assumption is not made. We note here that the Cauchy step defined above satisfies condition (2.3) for some $K > 0$.

As stated in section 5.1 of [13], we do not suggest choosing K and enforcing condition (2.3). Rather, we suggest that (2.3) results naturally from any reasonable algorithm for computing a step s_k^n.

Now we use the quasi-normal component to choose a linear manifold $\mathcal{M}_k$, parallel to the null space of the constraints. Let $\mathcal{M}_k = \{s : \nabla C_k^T s = \nabla C_k^T s_k^n\}$. We select the tangential component from $\mathcal{M}_k$. Observe that the intersection of $\mathcal{M}_k$ and the set $\{s = s^t + s_k^n : \|s\| \leq \delta_k\}$ is not empty.

On the manifold $\mathcal{M}_k$, we consider the quadratic model $q_k(s)$ of the Lagrangian function (1.1) given by

$$q_k(s) = \ell_k + \nabla_x \ell_k^T s + \frac{1}{2} s^T H_k s. \tag{2.5}$$

Let W_k be a matrix whose columns form a basis for the null space of ∇C_k^T. Then, when $W_k^T \nabla q_k(s_k^n) \neq 0$, the tangential component s_k^t is taken to be any step that satisfies the fraction of Cauchy decrease condition from s_k^n on $q_k(s)$ reduced to $\mathcal{M}_k$. That is, the trial step

$$s_k = s_k^t + s_k^n \in \mathcal{G}_k \cap \mathcal{M}_k, \tag{2.6}$$

where

$$\mathcal{G}_k = \{s = s^t + s_k^n : \|s\| \leq \delta_k,\ q_k(s_k^n) - q_k(s) \geq r_2[q_k(s_k^n) - q_k(s_k^n - \mathsf{t}_k^{cp} W_k W_k^T \nabla q_k(s_k^n))]\}.$$

The constant $r_2 \in (0,1]$ and t_k^{cp} is given by

$$\mathsf{t}_k^{cp} = \begin{cases} \dfrac{\|W_k^T \nabla q_k(s_k^n)\|^2}{\nabla q_k(s_k^n)^T W_k \bar{H}_k W_k^T \nabla q_k(s_k^n)} & \text{if } \dfrac{\|W_k^T \nabla q_k(s_k^n)\|^2 \|W_k W_k^T \nabla q_k(s_k^n)\|}{\nabla q_k(s_k^n)^T W_k \bar{H}_k W_k^T \nabla q_k(s_k^n)} \leq \bar{\delta}_k \\ & \text{and } \nabla q_k(s_k^n)^T W_k \bar{H}_k W_k^T \nabla q_k(s_k^n) > 0, \\ \dfrac{\bar{\delta}_k}{\|W_k W_k^T \nabla q_k(s_k^n)\|} & \text{otherwise,} \end{cases}$$

where $\bar{H}_k = W_k^T H_k W_k$ is the reduced Hessian matrix and $\bar{\delta}_k$ is the maximum length of the step allowed inside the set $\mathcal{M}_k \cap \{s = s^t + s_k^n : \|s\| \leq \delta_k\}$ in the negative reduced gradient direction $-W_k^T \nabla q_k(s_k^n)$.

Once the trial step is computed, an estimate for the Lagrange multiplier λ_{k+1} is needed to determine whether the computed trial step will be accepted. At this point, we will not present a particular way to estimate the Lagrange multiplier. Instead, we impose a condition on the estimates of the Lagrange multiplier that is needed to prove global convergence. The sequence $\{\lambda_k\}$ of Lagrange multiplier estimates is required to be bounded. So, any approximation to the Lagrange multiplier vector that produces a bounded sequence can be used. For example, setting λ_k to a fixed vector (or even the zero vector) for all k is valid. In section 9, we suggest two practical ways to estimate λ_k that produce bounded sequences of multipliers.

2.2. Testing the trial steps. Let s_k be a trial step computed by the algorithm and let λ_{k+1} be an estimate of the Lagrange multiplier vector. We test whether the point $(x_k + s_k, \lambda_{k+1})$ will be taken as a next iterate. In order to do this, a merit function is needed. We use, as a merit function, the augmented Lagrangian

$$\Phi(x, \lambda; \rho) = f(x) + \lambda^T C(x) + \rho \|C(x)\|^2, \tag{2.7}$$

where ρ is the penalty parameter. Many authors have used (2.7) as a merit function. See, for example, Gill, Murray, and Wright [24].

The actual reduction in the merit function in moving from (x_k, λ_k) to $(x_k + s_k, \lambda_{k+1})$ is defined to be

$$Ared_k = \Phi(x_k, \lambda_k; \rho_k) - \Phi(x_k + s_k, \lambda_{k+1}; \rho_k).$$

This can be written as

$$\begin{aligned} Ared_k = {} & \ell(x_k, \lambda_k) - \ell(x_k + s_k, \lambda_k) - \Delta\lambda_k^T C(x_k + s_k) \\ & + \rho_k [\, \|C_k\|^2 - \|C(x_k + s_k)\|^2], \end{aligned} \tag{2.8}$$

where $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$. The predicted reduction has the form

$$\begin{aligned} Pred_k = {} & -\nabla_x \ell_k^T s_k - \frac{1}{2} s_k{}^T H_k s_k - \Delta\lambda_k^T [C_k + \nabla C_k^T s_k] \\ & + \rho_k [\, \|C_k\|^2 - \|C_k + \nabla C_k^T s_k\|^2]. \end{aligned} \tag{2.9}$$

The acceptable step should be the step that produces a decrease in the merit function. To test for this, the predicted reduction has to be forced to be greater than zero by increasing the value of the penalty parameter if necessary. This takes us to the following section.

2.3. Updating the penalty parameter. To update the penalty parameter, Dennis, El-Alem, and Maciel [13] used a scheme proposed in [17]. This scheme ensures that the merit function is predicted to be decreased at each iteration by at least a fraction of the Cauchy decrease in the quadratic model of the linearized constraint. This indicates compatibility with the fraction of Cauchy decrease condition imposed on the quasi-normal component of the trial steps.

It is noteworthy that, since no regularity is assumed, there is no guarantee that when $\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2 = 0$, we have $Pred_k \geq 0$. Therefore, it could happen that $\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2 = 0$ and $Pred_k < 0$. In this case, the algorithm should be terminated because it is an infeasible stationary point of the constraints, as we will show in section 4. We write our way of updating the penalty parameter in algorithmic form as follows.

Algorithm 2.1. Updating the Penalty Parameter.

Step 1. Initialization

Set $\rho_{-1} = 1$ *and choose a small constant* $\hat{\rho} > 0$.

Step 2. At the current iterate x_k, after s_k has been chosen:

set $\rho_k = \rho_{k-1}$.

(a) If $Pred_k < 0$ *and* $\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2 = 0$*,* **then** *terminate.*

(b) If $Pred_k < \frac{\rho_k}{2}[\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2]$, **then** *set*

$$\rho_k = \frac{2[q_k(s_k) - q_k(0) + \Delta\lambda_k^T(C_k + \nabla C_k^T s_k)]}{\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2} + \hat{\rho}. \tag{2.10}$$

The initial choice of the penalty parameter ρ_{-1} is arbitrary. However, it should be chosen such that it is consistent with the scale of the problem. Here, for convenience, we take $\rho_{-1} = 1$.

The termination at Step 2(a) of the above algorithm can occur only at an infeasible Mayer–Bliss point, which is a good reason to stop. See section 4.

An immediate consequence of the above algorithm is that, at the current iteration, either the point x_k is an infeasible stationary point of the constraints (see section 4) and the algorithm terminates at Step 2(a) of the above algorithm or

$$Pred_k \geq \frac{\rho_k}{2}[\|C_k\|^2 - \|C_k + \nabla C_k^T s_k\|^2]. \tag{2.11}$$

2.4. Updating the trust-region radius. After computing a trial step and updating the penalty parameter, we test the step and accept it only if the actual reduction is greater than some fraction of the predicted reduction. That is, we accept the step s_k if $\frac{Ared_k}{Pred_k} \geq \eta_1$, where $\eta_1 \in (0, 1)$. Otherwise, we reject the step and decrease the radius of the trust region by setting $\delta_k = \alpha_1\|s_k\|$, where $\alpha_1 \in (0, 1)$.

The strategy that we follow for updating the trust-region radius is based on the standard rules for the unconstrained case (see, for example, Shultz, Schnabel, and Byrd [42]). However, for our global convergence theory, we use a modification due to Zhang, Kim, and Lasdon [49] (see also El-Hallabi and Tapia [22] and Dennis, El-Alem, and Maciel [13]). This modification is of no importance in practice; it is merely an analytic formality. At the beginning we set constants $\delta_{\max} \geq \delta_{\min}$, and each time we find an acceptable step, we start the next iteration with a trust-region radius greater than or equal to $\delta_{\min}$. In short, δ_k can be reduced below $\delta_{\min}$ while seeking an acceptable step, but $\delta_{\min} \leq \delta_{k+1}$ must hold at the beginning of the next iteration after finding an acceptable step. We must also have, for all k, $\delta_k \leq \delta_{\max}$. We present the method of updating the trust-region radius used by Dennis, El-Alem, and Maciel [13] in Step 5 of Algorithm 2.2 below.

After accepting the step and updating the trust-region radius, the Hessian matrix H_k must be updated. Our theory requires the sequence $\{H_k\}$ of approximate Hessians to be bounded. Thus, the exact Hessians or any approximation scheme that produces a bounded sequence of Hessians can be used. For instance, setting $H_k = 0$ for all k is valid.

Since we do not specify a particular way of computing W_k, it is required that $\{\|W_k\|\}$ be bounded and the sequence of smallest singular values of the matrices W_k , $k = 1, 2, \ldots$, be bounded away from zero.

2.5. Summary of the algorithm. We present a summary of the Dennis, El-Alem, and Maciel class of trust-region-based algorithms for solving problem (EQ).

ALGORITHM 2.2. THE TRUST-REGION ALGORITHM.

```
Step 0. (* Initialization *)
        Given $x_0$, $\lambda_0$, compute $W_0$.
        Choose $\alpha_1$, $\alpha_2$, $\eta_1$, $\eta_2$, $\hat{\rho}$, $\delta_{\min}$, $\delta_0$, and $\delta_{\max}$, such that $0 < \alpha_1 < 1 < \alpha_2$,
        $0 < \eta_1 < \eta_2 < 1$, $\hat{\rho} > 0$, and $\delta_{\min} \leq \delta_0 \leq \delta_{\max}$.
        Set $\rho_{-1} = 1$ and $k = 0$.
Step 1. (* Compute a trial step *)
      If $x_k$ is feasible, then
          (a) Find a step $s_k^t$ that satisfies a fraction of Cauchy decrease condition
              on the quadratic model $q_k(s)$ of the Lagrangian around $x_k$. (* See
              relation (2.6). *)
          (b) Set $s_k = s_k^t$.
      else        (* $C(x_k) \neq 0$ *)
          (a) Compute a quasi-normal step $s_k^n$ that satisfies condition (2.3) and a
              fraction of Cauchy decrease condition on the quadratic model of the
              linearized constraints. (* See inequality (2.1). *)
          (b) If $W_k^T \nabla q(s_k^n) = 0$, then set $s_k^t = 0$.
              else (* $W_k^T \nabla q(s_k^n) \neq 0$ *)
                  Find $s_k^t$ that satisfies a fraction of Cauchy decrease condition
                  on the quadratic model $q_k(s_k^n + s)$ from $s_k^n$. (* See relation
                  (2.6). *)
              End if
          (c) Set $s_k = s_k^n + s_k^t$.
      End if
Step 2. (* Update $\lambda_k$ *)
          Choose an estimate $\lambda_{k+1}$ of the Lagrange multiplier vector.
              Set $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$.
Step 3. (* Update the penalty parameter *)
          Update $\rho_{k-1}$ to obtain $\rho_k$ by using Algorithm 2.1.
Step 4. (* Evaluating the step and updating the trust-region radius *)
      If $\frac{Ared_k}{Pred_k} < \eta_1$ then
              Reduce the trust-region radius: $\delta_k \leftarrow \alpha_1 \|s_k\|$.
              Go to Step 2.
      Else if $\eta_1 \leq \frac{Ared_k}{Pred_k} < \eta_2$ then
              Accept the step: $x_{k+1} = x_k + s_k$.
              Set the trust-region radius: $\delta_{k+1} = \max\{\delta_k, \delta_{\min}\}$.
      Else  (* $\frac{Ared_k}{Pred_k} \geq \eta_2$ *)
              Accept the step: $x_{k+1} = x_k + s_k$.
              Increase the trust-region radius: $\delta_{k+1} = \min\{\delta_{\max}, \max\{\delta_{\min}, \alpha_2 \delta_k\}\}$.
      End if
Step 5.  (* Update $H_k$ and $k$ *)
              Update $H_k$.
          Set $k \leftarrow k + 1$, and go to Step 1.
```

In a practical implementation of the algorithm, a stopping criterion should be added. See section 9 for more details.

3. General assumptions. Let Ω be a convex subset of $\Re^n$ that contains all of x_k and $x_k + s_k$ for all trial steps s_k examined in the course of the algorithm. On the set Ω, we assume:

A1. f and C are twice continuously differentiable for all $x \in \Omega$.

A2. $f(x), \nabla f(x),\ \nabla^2 f(x),\ C(x),\ \nabla C(x),$ and $\nabla^2 C_i(x)$ for $i = 1, \ldots, m$ are uniformly bounded in Ω.

A3. The sequence of Lagrange multiplier vectors $\{\lambda_k\}$ is bounded.

A4. If approximations to the Hessian matrices are used, then we require that the matrices $H_k, k = 1, 2, \ldots,$ be uniformly bounded in norms.

A5. The sequence $\{\|W_k\|\}$, is bounded and the sequence of smallest singular values of the matrices $W_k, k = 1, 2, \ldots$ is bounded away from zero.

The above are the assumptions under which we prove global convergence. Observe that they do not include the assumption of the linear independence of the gradients of the constraints, an assumption commonly used by many researchers.

An immediate consequence of the above assumptions is the existence of positive constants b and b_1, such that for all k,

$$\|\nabla C_k C_k\| \leq b \tag{3.1}$$

and

$$\|W_k^T H_k\| \leq b_1. \tag{3.2}$$

4. Stationary points. In this section, we give definitions to four types of stationary points, show some of their properties, and show some relations between them. The terminology used in this section follows Burke [7], [8] and Yuan [48].

DEFINITION 4.1 (first-order point). *A point $x_\star \in \Re^n$ is called a first-order point of problem* (EQ) *if it satisfies*

$$W(x)^T \nabla f(x) = 0, \tag{4.1}$$

$$C(x) = 0. \tag{4.2}$$

Equations (4.1) and (4.2) are called the first-order conditions. If $x_\star$ solves (4.1), then this implies the existence of $\lambda_\star$ such that $x_\star$ and $\lambda_\star$ satisfy $\nabla f(x) + \nabla C(x)\lambda = 0$.

DEFINITION 4.2 (Mayer–Bliss point). *A point $x_\star \in \Re^n$ is called a feasible Mayer-Bliss point or simply a Mayer–Bliss point of problem* (EQ) *if there exist a constant $\gamma_\star \in \Re$ and a Lagrange multiplier vector $\lambda_\star \in \Re^m$ such that $(\gamma_\star, \lambda_\star) \neq (0, 0)$ and $x_\star$, $\gamma_\star$, and $\lambda_\star$ satisfy the following conditions:*

$$\gamma \nabla f(x) + \nabla C(x)\lambda = 0, \tag{4.3}$$

$$C(x) = 0. \tag{4.4}$$

Equations (4.3) and (4.4) are called the feasible Mayer–Bliss conditions. See Mayer [31] and Bliss [5].

The feasible Mayer–Bliss conditions are the same as the well-known Fritz John conditions for general nonlinear programming when they are applied to problem (EQ). See Mangasarian [29].

If $(x_\star, \gamma_\star, \lambda_\star)$ is a feasible Mayer–Bliss point and $\gamma_\star \neq 0$, then $(x_\star, \frac{\lambda_\star}{\gamma_\star})$ is a first-order point. Conversely, if $(x_\star, \lambda_\star)$ is a first-order point then it is a feasible Mayer–Bliss point with $\gamma_\star = 1$.

DEFINITION 4.3 (infeasible Mayer-Bliss point). *A point $x_\star \in \Re^n$ is called an infeasible Mayer–Bliss point if $x_\star$ satisfies the following conditions:*

$$\nabla C(x) C(x) = 0, \tag{4.5}$$

$$C(x) \neq 0. \tag{4.6}$$

Equations (4.5) and (4.6) are called the infeasible Mayer–Bliss conditions.

If $x_\star$ is an infeasible Mayer–Bliss point, then there exist a constant $\gamma_\star \in \Re$ and a Lagrange multiplier vector $\lambda_\star \in \Re^m$ such that $(\gamma_\star, \lambda_\star) \neq (0,0)$ and $x_\star$, $\gamma_\star$, and $\lambda_\star$ satisfy the following conditions:

$$\begin{aligned} \gamma \nabla f(x) + \nabla C(x)\lambda &= 0, \\ \nabla C(x) C(x) &= 0. \end{aligned}$$

If, in addition, $\gamma_\star \neq 0$, then $(x_\star, \frac{\lambda_\star}{\gamma_\star})$ is an infeasible first-order point. The definition of such a point is given in the following definition, which is added for completeness.

DEFINITION 4.4 (infeasible first-order point). *An infeasible point $x_\star \in \Re^n$ is called an infeasible first-order point of problem* (EQ) *if it satisfies*

$$W(x)^T \nabla f(x) = 0, \tag{4.7}$$

$$\nabla C(x) C(x) = 0. \tag{4.8}$$

Equations (4.7) and (4.8) are called the infeasible first-order conditions for problem (EQ).

DEFINITION 4.5 (stationary conditions). *The conditions stated in any of Definitions* 4.1–4.4 *are called stationary conditions of problem* (EQ) *and the point that satisfies any of these stationary conditions is called a stationary point.*

The following are noteworthy:

(a) Mayer–Bliss points are the union of first-order points and feasible nonregular points;

(b) infeasible first-order points are stationary points of $\|C(x)\|^2$ that are stationary with respect to

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & C(x) = C(x_\star); \end{array}$$

(c) infeasible Mayer–Bliss points are infeasible stationary points of $\|C(x)\|^2$.

The following lemma gives a condition for an infeasible iterate x_k generated by the algorithm to be a Mayer–Bliss point.

LEMMA 4.1. *If at a point x_k generated by the algorithm, $\|C_k\| \neq 0$ and*

$$\text{minimum}_{\|s\| \le \delta_k} \quad \|C_k + \nabla C_k^T s\|^2 = \|C_k\|^2, \tag{4.9}$$

then it is an infeasible Mayer–Bliss point.

Proof. If (4.9) holds, then $s_k = 0$ is an unconstrained minimizer. Let $\bar{q}(s) = \|C_k + \nabla C_k^T s\|^2$. Because $\bar{q}(s)$ is convex, the local minimizer is a global one. Also, $\nabla_s \bar{q}(s_k) = 0$ implies that the minimizer satisfies $\nabla C_k C_k = 0$. Hence (4.5) holds. This completes the proof. □

In the above lemma, it is easy to see that at the point x_k, the matrix ∇C_k does not have full column rank.

It is noteworthy that if the algorithm generates a point x_k which is an infeasible Mayer–Bliss point and $Pred_k < 0$, then the algorithm may not be able to move away from this point. In this case, the algorithm terminates at Step 2(a) of Algorithm 2.1. However, we will proceed with the analysis assuming that this will not occur.

The following lemma gives conditions for the sequence of iterates generated by the algorithm to have a subsequence that satisfies the feasible Mayer–Bliss conditions in the limit. A similar lemma for a different algorithm was given by Yuan [48]. By satisfying the feasible Mayer–Bliss conditions in the limit (asymptotically), we mean the existence of λ and γ such that the left-hand sides of (4.3) and (4.4) converge to zero.

Even though the following lemma is not used in our analysis, we include it because it identifies a situation that may happen to a subsequence of the iteration sequence. Namely, a sequence of steps $\{s_{k_j}\}$ succeeded in driving the sequence $\{\|C_{k_j}\|\}$ to zero in the limit. At the same time, $\|s_k\| \le \|C_k\|$ holds for all $k \in \{k_j\}$. As a result of that, the sequence $\{\|s_{k_j}\|\}$ is driven to zero in the limit by the sequence $\{\|C_{k_j}\|\}$.

Lemma 4.2. *If there exists a subsequence of infeasible iterates $\{k_j\}$ such that $\lim_{k_j\to\infty} \|C_{k_j}\| = 0$ and*

$$(4.10) \qquad \lim_{k_j\to\infty} \left\{ \text{minimum}_{s\in\Re^n} \quad \frac{\|C_{k_j} + \nabla C_{k_j}^T s\|^2}{\|C_{k_j}\|^2} : \|s\| \le \|C_{k_j}\| \right\} = 1,$$

then it satisfies the feasible Mayer–Bliss conditions in the limit.

Proof. The above limit is equivalent to

$$(4.11) \qquad \lim_{k_j\to\infty} \left\{ \text{minimum}_{\|d\|\le 1} \quad \|U_{k_j} + \nabla C_{k_j}^T d\|^2 \right\} = 1,$$

where U_{k_j} is a unit vector in the direction of C_{k_j} and $d = \frac{s}{\|C_{k_j}\|}$. Let $\bar{d}_{k_j}$ be a solution to the minimization problem inside the above limit. Then there exists a nonnegative parameter μ_{k_j} such that

$$(4.12) \qquad \nabla C_{k_j} U_{k_j} + \nabla C_{k_j} \nabla C_{k_j}^T \bar{d}_{k_j} + \mu_{k_j} \bar{d}_{k_j} = 0$$

and

$$(4.13) \qquad \mu_{k_j}(\|\bar{d}_{k_j}\|^2 - 1) = 0.$$

From (4.11), we have

$$(4.14) \qquad \lim_{k_j\to\infty} \left\{ 2\bar{d}_{k_j}{}^T \nabla C_{k_j} U_{k_j} + \bar{d}_{k_j}{}^T \nabla C_{k_j} \nabla C_{k_j}^T \bar{d}_{k_j} \right\} = 0.$$

If $\lim_{k_j\to\infty} \bar{d}_{k_j} = 0$, then from (4.12), we have $\lim_{k_j\to\infty} \nabla C_{k_j} U_{k_j} = 0$. Otherwise, multiply (4.12) from the right by $2\bar{d}_{k_j}{}^T$, subtract from (4.14), and use (4.13). We thus obtain, as $k_j \to \infty$, $\nabla C_{k_j}^T \bar{d}_{k_j} \to 0$ and $\mu_{k_j} \to 0$. This yields

$$(4.15) \qquad \lim_{k_j\to\infty} \nabla C_{k_j} U_{k_j} = 0.$$

Hence, in both cases, (4.3) holds in the limit with $\gamma = 0$. This shows that the lemma is true. □

Notice that the result we obtained in the above lemma is independent of the sequence $\{\|W_{k_j}^T \nabla f_{k_j}\|\}$. In other words, it may be the case that the sequence of steps $\{\|s_{k_j}\|\}$ converges to zero, while the sequence $\{\|W_{k_j}^T \nabla f_{k_j}\|\}$ is bounded away from zero.

From Definition 4.2, we can easily see that, for any subsequence $\{k_j\}$ of the iteration sequence that asymptotically satisfies the feasible Mayer–Bliss conditions that are not first-order conditions, the corresponding sequence of smallest singular values of the matrices ∇C_k for all $k \in \{k_j\}$ is not bounded away from zero.

The following lemma gives conditions for a subsequence of the iteration sequence to satisfy the feasible Mayer–Bliss conditions in the limit.

LEMMA 4.3. *If there exists a subsequence of iterates* $\{k_j\}$ *such that* $\{\|C_{k_j}\|\}$ *converges to zero and the sequence of smallest singular values of* $\{\nabla C_{k_j}\}$ *converges to zero, then it satisfies the feasible Mayer–Bliss conditions in the limit.*

Proof. Take $\{\lambda_{k_j}\}$ to be the sequence of the right singular vectors that correspond to the smallest singular values of ∇C_k for all $k \in \{k_i\}$. In the limit, an equation similar to (4.15) holds. This implies that (4.3) holds in the limit with $\gamma = 0$. This completes the proof. □

Throughout the rest of the paper, we use $\{\sigma_{k_i}\}$ to denote the sequence of smallest singular values of ∇C_k for all $k \in \{k_i\}$.

In the rest of the paper, we present our convergence results. We start with the following section, which summarizes our global convergence results.

5. Main result. In this section, we state the main result of our convergence analysis in order to understand the motivation for the lemmas presented in the next two sections.

THEOREM 5.1. *If* A1–A5 *hold, then the sequence of iterates generated by Algorithm* 2.2 *has a subsequence that satisfies one of the Mayer–Bliss stationary conditions of problem* (EQ) *in the limit.*

The statement of the above theorem means that, asymptotically, the iteration sequence satisfies either the infeasible Mayer–Bliss conditions that are not infeasible first-order conditions, the feasible Mayer–Bliss conditions that are not first-order conditions, the infeasible first-order conditions, or the first-order conditions.

The above theorem summarizes the main results of this paper; its proof is presented in section 8. The proof needs some intermediate lemmas, which are presented in the following two sections.

6. Intermediate results. In this section, we present some technical lemmas needed in the proof of our main global convergence results.

The following two lemmas use the fact that the steps s_k^n and s_k^t satisfy the fraction of Cauchy decrease condition. They express in a manageable form the pair of fraction of Cauchy decrease conditions imposed on the trial steps.

LEMMA 6.1. *Assume* A1–A2. *Then there exists a positive constant* K_1 *independent of the iterates such that the quasi-normal component of the trial step* s_k^n *satisfies*

$$\|C_k\|^2 - \|C_k + \nabla C_k^T s_k^n\|^2 \geq K_1 \|\nabla C_k C_k\| \min\{\|\nabla C_k C_k\|, \delta_k\}. \tag{6.1}$$

Proof. If $\nabla C_k C_k = 0$, then $\nabla C_k^T s_k^n = 0$ and (6.1) holds immediately, since the right-hand side is zero.

Assume that $\|\nabla C_k C_k\| > 0$. In this case, the proof follows from the fact that the step s_k^n satisfies the fraction of Cauchy decrease condition and from using assumption A2. For a proof, see Powell [40] or Moré [36]. □

From (2.6) and the above lemma, we have for all k

$$Pred_k \geq \frac{K_1 \rho_k}{2} \|\nabla C_k C_k\| \min\{\|\nabla C_k C_k\|, \delta_k\}. \tag{6.2}$$

LEMMA 6.2. *Assume* A1–A5. *Then there exists a positive constant* K_2 *independent of the iterates such that*

$$q_k(s_k^n) - q_k(s_k) \geq K_2 \|W_k^T \nabla q_k(s_k^n)\| \min\{\|W_k^T \nabla q_k(s_k^n)\|, \delta_k\}. \tag{6.3}$$

Proof. The proof is similar to the proof of the above lemma. □

The following two lemmas give upper bounds on the difference between the actual reduction and the predicted reduction.

LEMMA 6.3. *Assume* A1–A4. *Then there exists a positive constant K_3 independent of k such that*

$$|Ared_k - Pred_k| \le K_3 \; [\; \|s_k\|^2 + \rho_k\|s_k\|^3 + \rho_k\|s_k\|^2\|C_k\| \;]. \tag{6.4}$$

Proof. From (2.8), (2.9), and the Cauchy–Schwarz inequality, we have

$$\begin{aligned} |Ared_k - Pred_k| \le{}& |\ell(x_k,\lambda_k) + \nabla\ell(x_k,\lambda_k)^T s_k + \frac{1}{2}s_k^T H_k s_k - \ell(x_k+s_k,\lambda_k)| \\ &+|\Delta\lambda_k^T[C_k + \nabla C_k^T s_k - C(x_k+s_k)]\;| \\ &+\rho_k|\|C_k+\nabla C_k^T s_k\|^2 - \|C(x_k+s_k)\|^2| \\ \le{}& \frac{1}{2}|s_k^T[H_k - \nabla^2\ell(x_k+\xi_1 s_k)]s_k| + \frac{1}{2}|s_k^T\nabla^2 C(x_k+\xi_2 s_k)\Delta\lambda_k s_k| \\ &+\rho_k|s_k^T[\nabla C_k\nabla C_k^T - \nabla C(x_k+\xi_3 s_k)\nabla C(x_k+\xi_3 s_k)^T]s_k| \\ &+\rho_k|s_k^T\nabla^2 C(x_k+\xi_3 s_k)C(x_k+\xi_3 s_k)s_k| \end{aligned}$$

for some $\xi_1,\xi_2,\xi_3 \in (0,1)$. By using assumptions A1–A4, the proof follows. □

LEMMA 6.4. *Assume* A1–A4. *Then there exists a positive constant K_4 independent of k such that*

$$|Ared_k - Pred_k| \le K_4\rho_k\|s_k\|^2. \tag{6.5}$$

Proof. The proof follows directly from the above lemma and the fact that $\rho_k \ge 1$ and $\|s_k\|$ and $\|C_k\|$ are uniformly bounded. □

The following lemma shows that if at any iteration k, the point x_k is not a stationary point of the constraints, then the algorithm cannot loop infinitely without finding an acceptable step. To state the lemma, we need to introduce one more notation. The ith trial iterate of iteration k is denoted by k^i. We note here that the notation k^i does not refer to an element of a subsequence. In fact, k and i are two independent indices, and s_{k^i} is not a member of the sequence $\{s_k\}$ unless it is acceptable.

LEMMA 6.5. *Assume* A1–A4. *If $\|\nabla C_k C_k\| > 0$, then an acceptable step is found after finitely many trials; i.e., the condition $Ared_{k^j}/Pred_{k^j} \ge \eta_1$ will be satisfied for some finite j.*

Proof. Since $\|\nabla C_k C_k\| > 0$, then using (6.2) and (6.5), we have

$$\left|\frac{Ared_k}{Pred_k} - 1\right| = \frac{|Ared_k - Pred_k|}{Pred_k} \le \frac{2K_4\delta_k^2}{K_1\|\nabla C_k C_k\|\min\{\|\nabla C_k C_k\|,\delta_k\}}.$$

Now, as the trial step s_{k^j} gets rejected, δ_{k^j} becomes small, and eventually we will have

$$\left|\frac{Ared_{k^j}}{Pred_{k^j}} - 1\right| \le \frac{2K_4\delta_{k^j}}{K_1\|\nabla C_k C_k\|}.$$

This inequality implies that after a finite number of trials (i.e., for j finite), the acceptance rule will be met, and this completes the proof. □

LEMMA 6.6. *Assume* A1–A5. *There exists a constant $K_5 > 0$ such that for any indices j and k, where ρ_{k^j} is increased at the jth trial iterate of the kth iteration,*

$$\rho_{k^j}\{\|C_k\| - \|C_k + \nabla C_k^T s_{k^j}\|\} \leq K_5 \max\{\|C_k\|, \|s_{k^j}^{mn}\|\},$$

where $s_{k^j}^{mn}$ is the minimum-norm solution of (2.4) *with $\delta_k = \delta_{k^j}$.*

Proof. If ρ_{k^j} is increased at the jth trial step of the kth iteration, then it is updated by (2.10). Hence,

$$\begin{aligned}
\frac{\rho_{k^j}}{2}[\|C_k\|^2 - \|C_k + \nabla C_k^T s_{k^j}\|^2] &= [q_k(s_{k^j}) - q_k(0)] + \Delta\lambda_{k^j}{}^T(C_k + \nabla C_k^T s_{k^j}) \\
&\quad + \frac{\hat{\rho}}{2}[\|C_k\|^2 - \|C_k + \nabla C_k^T s_{k^j}\|^2] \\
&= [q_k(s_{k^j}) - q_k(s_{k^j}^n)] + [q_k(s_{k^j}^n) - q_k(0)] \\
&\quad + \Delta\lambda_{k^j}^T(C_k + \nabla C_k^T s_{k^j}^n) \\
&\quad + \frac{\hat{\rho}}{2}[-2(\nabla C_k C_k)^T s_{k^j}^n - \|\nabla C_k^T s_{k^j}^n\|^2] \\
&\leq [q_k(s_{k^j}) - q_k(s_{k^j}^n)] + \|\nabla \ell_k\|\|s_{k^j}^n\| + \frac{1}{2}\|H_k\|\|s_{k^j}^n\|^2 \\
&\quad + \|\Delta\lambda_{k^j}\|\|C_k + \nabla C_k^T s_{k^j}^n\| + \hat{\rho}[\|\nabla C_k C_k\|\|s_{k^j}^n\| \\
&\quad + \|\nabla C_k^T\|^2\|s_{k^j}^n\|^2].
\end{aligned}$$

The rest of the proof follows by applying (6.3), (2.3), and the fact that $\delta_k \leq \delta_{max}$ to the right-hand side, followed by the use of the general assumptions. □

From (6.1) and the above lemma, we can write, at any trial iteration k^j at which the penalty parameter is increased,

$$\rho_{k^j}\|\nabla C_k C_k\| \min\{\|\nabla C_k C_k\|, \delta_{k^j}\} \leq \frac{K_5}{K_1}\max\{\|C_k\|, \|s_{k^j}^{mn}\|\}. \tag{6.6}$$

LEMMA 6.7. *Assume* A1–A4. *If the jth trial step of a given iteration k satisfies*

$$\|s_{k^j}\| \leq \min\left\{\frac{(1-\eta_1)K_1}{4K_4}, 1\right\}\|\nabla C_k C_k\|, \tag{6.7}$$

then the step must be accepted.

Proof. The proof is by contradiction. Suppose that (6.7) holds but the step s_{k^j} is rejected. Then, we have

$$(1-\eta_1) < \frac{|Ared_{k^j} - Pred_{k^j}|}{Pred_{k^j}}.$$

Substituting from (6.2) and (6.5) and using (6.7), we have

$$(1-\eta_1) < \frac{2K_4\|s_{k^j}\|^2}{K_1\|\nabla C_k C_k\|\|s_{k^j}\|} \leq \frac{1}{2}(1-\eta_1).$$

This gives a contradiction and implies that the step must be accepted. This completes the proof of the lemma. □

The following lemma is a consequence of the above lemma.

LEMMA 6.8. *Assume* A1–A4. *All trial iterates j of any iteration k generated by the algorithm satisfy*

$$\delta_{k^j} \geq \min\left\{\frac{\delta_{\min}}{b}, \alpha_1 \frac{(1-\eta_1)K_1}{4K_4}, \alpha_1\right\} \|\nabla C_k C_k\|, \tag{6.8}$$

where b is as in (3.1) *and α is as in Step* 5 *of Algorithm* 2.2.

Proof. Consider any iterate k^j. If the previous step was accepted, i.e., if $j = 1$, then $\delta_{k^j} = \delta_{k^1} \geq \delta_{\min}$. Using (3.1), we can write

$$\delta_{k^j} \geq \frac{\delta_{\min}}{b} \|\nabla C_k C_k\|.$$

Therefore, (6.8) holds in this case.

Now assume that $j > 1$. That is, there exists at least one rejected trial step. Hence, we must have

$$\|s_{k^{j-1}}\| > \min\left\{\frac{(1-\eta_1)K_1}{4K_4}, 1\right\} \|\nabla C_k C_k\|;$$

otherwise, we get a contradiction with Lemma 6.7. From the method of updating the trust region, we have

$$\delta_{k^j} = \alpha_1 \|s_{k^{j-1}}\| > \alpha_1 \min\left\{\frac{(1-\eta_1)K_1}{4K_4}, 1\right\} \|\nabla C_k C_k\|.$$

Hence the lemma is proved. □

The following lemma is used in proving that the sequence $\{\|\nabla C_k C_k\|\}$ converges to zero. It says that as long as $\{\|\nabla C_k C_k\|\}$ is bounded away from zero, the sequence of trust-region radii $\{\delta_{k^j}\}$ is bounded away from zero.

LEMMA 6.9. *Assume* A1–A4. *Given $\varepsilon_0 > 0$, there exists $K_6 > 0$ such that if $\|\nabla C_k C_k\| \geq \varepsilon_0$, then for all trial iterates j of any iteration k,*

$$\delta_{k^j} > K_6.$$

Proof. Using $\|\nabla C_k C_k\| \geq \varepsilon_0$ in (6.8) and taking

$$K_6 = \varepsilon_0 \min\left\{\frac{\delta_{\min}}{b}, \alpha_1 \frac{(1-\eta_1)K_1}{4K_4}, \alpha_1\right\}, \tag{6.9}$$

the proof follows directly from the above lemma. □

From (6.6) and Lemma 6.8 and using the general assumptions, we have for all k^j at which the penalty parameter is increased

$$\rho_{k^j} \|\nabla C_k C_k\|^2 \leq K_7, \tag{6.10}$$

where K_7 is a positive constant that does not depend on j or k. This inequality is used in studying the convergence of the sequence $\{\|\nabla C_k C_k\|\}$. This is the subject of the following section.

7. Stationary points of the constraints. The following lemma proves that for the iteration sequence generated by Algorithm 2.2, if $\{\rho_k\}$ is unbounded, then the sequence $\{\|\nabla C_k C_k\|\}$ is not bounded away from zero.

LEMMA 7.1. *Assume* A1–A5. *If $\{\rho_k\}$ is unbounded, then the sequence of iterates generated by the algorithm satisfies*

$$\lim_{k_i \to \infty} \|\nabla C_{k_i} C_{k_i}\| = 0, \tag{7.1}$$

where $\{k_i\}$ is the subsequence of the iteration sequence at which the penalty parameter is increased.

Proof. The proof follows directly from the assumption that $\{\rho_k\}$ is unbounded and from (6.10). □

If, in addition to the assumptions of the above lemma, we have $\limsup_{k_i \to \infty} \|C_{k_i}\| > 0$, then the sequence $\{k_i\}$ has a subsequence that satisfies the infeasible Mayer–Bliss conditions in the limit.

The following lemma proves a stronger result when $\lim_{k_i \to \infty} \|C_{k_i}\| = 0$, where $\{k_i\}$ is the subsequence of the iteration sequence at which the penalty parameter is increased.

LEMMA 7.2. *Assume* A1–A5. *If $\{\rho_k\}$ is unbounded and $\lim_{k_i \to \infty} \|C_{k_i}\| = 0$, where $\{k_i\}$ is the sequence of iterates at which the penalty parameter is increased, then the iteration sequence satisfies*

$$\lim_{k \to \infty} \|\nabla C_k C_k\| = 0. \tag{7.2}$$

Proof. Suppose that $\limsup_{k \to \infty} \|\nabla C_k C_k\| \geq \varepsilon > 0$. This implies the existence of an infinite subsequence of indices $\{k_j\}$ indexing iterates that satisfy $\|\nabla C_k C_k\| \geq \frac{\varepsilon}{2}$ for all $k \in \{k_j\}$.

From Lemma 7.1, $\lim_{k_i \to \infty} \|\nabla C_{k_i} C_{k_i}\| = 0$, where $\{k_i\}$ is the subsequence of the iteration sequence at which the penalty parameter is increased. Therefore, for k sufficiently large, there are no common elements between the two sequences $\{k_i\}$ and $\{k_j\}$. For all $\hat{k} \in \{k_j\}$, using (6.2) and Lemma 6.9, we have

$$\frac{Ared_{\hat{k}}}{\rho_{\hat{k}}} \geq \eta_1 \frac{Pred_{\hat{k}}}{\rho_{\hat{k}}} \geq \eta_1 \frac{\varepsilon K_1}{4} \min\left[\frac{\varepsilon}{2}, \delta_{\hat{k}}\right] \geq \eta_1 \frac{\varepsilon K_1}{4} \min\left[\frac{\varepsilon}{2}, \bar{K}_6\right],$$

where $\bar{K}_6$ is the same as K_6 in (6.9) with ε_0 replaced by $\frac{\varepsilon}{2}$. Hence, we have

$$\frac{\ell_{\hat{k}} - \ell_{\hat{k}+1}}{\rho_{\hat{k}}} + \|C_{\hat{k}}\|^2 - \|C_{\hat{k}+1}\|^2 \geq \eta_1 \frac{\varepsilon K_1}{4} \min\left[\frac{\varepsilon}{2}, \bar{K}_6\right] > 0. \tag{7.3}$$

On the other hand, for all acceptable steps generated by the algorithm, we have

$$\frac{\ell_k - \ell_{k+1}}{\rho_k} + \|C_k\|^2 - \|C_{k+1}\|^2 \geq 0. \tag{7.4}$$

Let $k_{\hat{i}}$ and $k_{\hat{i}+1}$ be two consecutive elements of the sequence $\{k_i\}$ such that there exists an iterate $k \in \{k_j\}$ between $k_{\hat{i}}$ and $k_{\hat{i}+1}$. From (7.3) and (7.4), we can write

$$\sum_{k=k_{\hat{i}}}^{k_{\hat{i}+1}-1} \frac{\{\ell_k - \ell_{k+1}\}}{\rho_k} + \|C_{k_{\hat{i}}}\|^2 - \|C_{k_{\hat{i}+1}}\|^2 \geq \eta_1 \frac{\varepsilon K_1}{4} \min\left[\frac{\varepsilon}{2}, \bar{K}_6\right] > 0.$$

Because the value of the penalty parameter is the same for all iterates $k_{\hat{i}}, \ldots, k_{\hat{i}+1} - 1$, we have

$$\frac{\ell_{k_{\hat{i}}} - \ell_{k_{\hat{i}+1}}}{\rho_{k_{\hat{i}}}} + \|C_{k_{\hat{i}}}\|^2 - \|C_{k_{\hat{i}+1}}\|^2 \geq \eta_1 \frac{\varepsilon K_1}{4} \min\left[\frac{\varepsilon}{2}, \bar{K}_6\right].$$

But because $|\ell_k|$ is bounded and $\rho_k \to \infty$ as $k \to \infty$, we can write, for $k_{\hat{i}}$ sufficiently large,

$$\|C_{k_{\hat{i}}}\|^2 - \|C_{k_{\hat{i}+1}}\|^2 \geq \eta_1 \frac{\varepsilon K_1}{8} \min\left[\frac{\varepsilon}{2}, \bar{K}_6\right] > 0.$$

This contradicts the assumption that $\lim_{k_i \to \infty} \|C_{k_i}\| = 0$. The supposition is wrong. This proves the lemma. □

When $\{\rho_k\}$ is bounded, we have the following result.

Lemma 7.3. *Assume* A1–A4. *If $\{\rho_k\}$ is bounded, then the sequence of iterates generated by the algorithm satisfies*

$$\lim_{k \to \infty} \|\nabla C_k C_k\| = 0. \tag{7.5}$$

Proof. The proof is by contradiction. Suppose that $\limsup_{k \to \infty} \|\nabla C_k C_k\| \geq \varepsilon_0 > 0$. This implies the existence of an infinite subsequence of indices $\{k_j\}$ indexing iterates that satisfy $\|\nabla C_k C_k\| \geq \frac{\varepsilon_0}{2}$ for all $k \in \{k_j\}$.

Since $\{\rho_k\}$ is bounded, there exist an integer $\bar{k}$ and a positive constant $\bar{\rho}$ such that for all $k \geq \bar{k}$, $\rho_k = \bar{\rho}$. Using the general assumptions, this fact implies that $\{\Phi_k\}$ is bounded.

From (6.2), we have for all acceptable steps generated by the algorithm

$$\Phi_k - \Phi_{k+1} = Ared_k \geq \eta_1 Pred_k \geq 0.$$

From (6.2) and Lemma 6.9, we have for all $k_j \geq \bar{k}$

$$Pred_{k_j} \geq \frac{K_1 \bar{\rho} \varepsilon_0}{4} \min\left\{\frac{\varepsilon_0}{2}, \hat{K}_6\right\} > 0, \tag{7.6}$$

where $\hat{K}_6$ is the same as K_6 in (6.9) with ε_0 replaced by $\frac{\varepsilon_0}{2}$. Using the fact that the steps indexed by any member of the sequence $\{k_j\}$ are acceptable, we have

$$\Phi_{k_j} - \Phi_{k_j+1} = Ared_{k_j} \geq \eta_1 Pred_{k_j} \geq \eta_1 \frac{K_1 \bar{\rho} \varepsilon_0}{4} \min\left\{\frac{\varepsilon_0}{2}, \hat{K}_6\right\} > 0. \tag{7.7}$$

Since $\{\Phi_k\}$ is bounded below, a contradiction arises if we let k_j go to infinity. This proves the lemma. □

Theorem 7.4. *Assume* A1–A5. *If* $\limsup_{k \to \infty} \|C_k\| > 0$, *then the iteration sequence has a subsequence that satisfies the infeasible Mayer–Bliss conditions in the limit.*

Proof. Consider first the case when $\{\rho_k\}$ is unbounded. From Lemma 7.1, we have $\lim_{k_i \to \infty} \|\nabla C_{k_i} C_{k_i}\| = 0$, where $\{k_i\}$ is the sequence of iterates at which the penalty parameter is increased.

If $\limsup_{k_i \to \infty} \|C_{k_i}\| > 0$, then there exists a subsequence of the sequence $\{k_i\}$ that satisfies the infeasible Mayer–Bliss conditions in the limit.

Assume that $\lim_{k_i \to \infty} \|C_{k_i}\| = 0$. From Lemma 7.2, we have $\lim_{k \to \infty} \|\nabla C_k C_k\| = 0$. On the other hand, because $\limsup_{k \to \infty} \|C_k\| > 0$, there exists a subsequence of the iteration sequence that satisfies the infeasible Mayer–Bliss conditions in the limit.

Now, consider the case when $\{\rho_k\}$ is bounded. From Lemma 7.3, we have $\lim_{k\to\infty}\|\nabla C_k C_k\| = 0$. This limit and the assumption that $\limsup_{k\to\infty}\|C_k\| > 0$ imply the existence of a subsequence of the iteration sequence that satisfies the infeasible Mayer–Bliss conditions in the limit. This completes the proof. □

8. Stationary conditions. In this section, we answer the following questions. Does the iteration sequence have a subsequence that satisfies the Mayer–Bliss conditions in the limit? If yes, can we identify it? Does the iteration sequence have a subsequence that satisfies the first-order conditions in the limit? If yes, can we identify it? To answer these questions, we need the following three technical lemmas.

The following lemma gives a lower bound on the predicted decrease in the merit function produced by the trial step.

LEMMA 8.1. *Assume* A1–A5. *The predicted decrease in the merit function satisfies*

$$\begin{aligned} Pred_{k^i} \geq{}& K_2\|W_k^T\nabla q_k(s^n_{k^i})\|\min\{\|W_k^T\nabla q_k(s^n_{k^i})\|\ ,\ \delta_{k^i}\}\\ &- K_8\max\{\|C_k\|, \|s^{mn}_{k^i}\|\} + \rho_k[\|C_k\|^2 - \|C_k + \nabla C_k^T s_{k^i}\|^2], \end{aligned}$$

where K_2 *is as in Lemma* 6.2 *and* K_8 *is a positive constant independent of* k *and* i.

Proof. We have

$$\begin{aligned} q_k(0) - q_k(s^n_{k^i}) &= -\nabla_x\ell_k^T s^n_{k^i} - \frac{1}{2}{s^n_{k^i}}^T H_k s^n_{k^i}\\ &\geq -\|\nabla_x\ell_k\|\,\|s^n_{k^i}\| - \frac{1}{2}\|H_k\|\,\|s^n_{k^i}\|^2\\ &= -(\|\nabla_x\ell_k\| + \frac{1}{2}\|H_k\|\,\|s^n_{k^i}\|)\,\|s^n_{k^i}\|. \end{aligned}$$

Using (2.3) and the fact that $\|s^n_{k^i}\| < \delta_{\max}$, we can write

$$q_k(0) - q_k(s^n_{k^i}) \geq -K\left(\|\nabla_x\ell_k\| + \frac{1}{2}\|H_k\|\,\delta_{\max}\right)\ \|s^{mn}_{k^i}\|.$$

Using the facts that λ_k and $\Delta\lambda_k$ are bounded, $\|C_k + \nabla C_k^T s_{k^i}\| \leq \|C_k\|$, and the general assumptions, there exists a positive constant K_8 such that

$$q_k(0) - q_k(s^n_{k^i}) - {\Delta\lambda_{k^i}}^T(C_k + \nabla C_k^T s_{k^i}) \geq -K_8\max\{\|C_k\|, \|s^{mn}_{k^i}\|\}. \tag{8.1}$$

Now, we have

$$\begin{aligned} Pred_{k^i} &= q_k(0) - q_k(s_{k^i}) - {\Delta\lambda_{k^i}}^T(C_k + \nabla C_k^T s_{k^i}) + \rho[\|C_k\|^2 - \|C_k + \nabla C_k^T s_{k^i}\|^2]\\ &= (q_k(s^n_{k^i}) - q_k(s_{k^i}))\\ &\quad + (q_k(0) - q_k(s^n_{k^i})) - {\Delta\lambda_{k^i}}^T(C_k + \nabla C_k^T s_{k^i})\\ &\quad + \rho[\|C_k\|^2 - \|C_k + \nabla C_k^T s_{k^i}\|^2]. \end{aligned}$$

Substituting (6.3) and (8.1) in this inequality, we obtain the desired result. □

LEMMA 8.2. *Assume* A1–A5. *If at a given trial iteration* k^i, $\|W_k^T\nabla f_k\| \geq \varepsilon_0$ *and* $\max\{\|C_k\|, \|s^{mn}_{k^i}\|\} \leq \beta\delta_{k^i}$, *where* ε_0 *is a positive constant and* β *is given by*

$$0 < \beta \leq \min\left\{\frac{\varepsilon_0}{2b_1K\delta_{\max}}, \frac{K_2\varepsilon_0}{4K_8}\min\left\{\frac{\varepsilon_0}{2\delta_{\max}}, 1\right\}\right\},$$

where K is as in (2.3)*, b_1 is as in* (3.2)*, K_2 is as in* (6.3)*, and K_8 is as in Lemma* 8.1*, then there exists a positive constant K_9 that depends on ε_0 but does not depend on k or i, such that*

$$Pred_{k^i} \geq K_9\delta_{k^i} + \rho_{k^i}\{\|C_k\|^2 - \|C_k + \nabla C_k^T s_{k^i}\|^2\}. \tag{8.2}$$

Proof. Since

$$\begin{aligned}\|W_k^T \nabla q_k(s^n_{k^i})\| &= \|W_k^T(\nabla_x \ell_k + H_k s^n_{k^i})\| \geq \|W_k^T \nabla_x \ell_k\| - \|W_k^T H_k s^n_{k^i}\| \\ &\geq \varepsilon_0 - b_1 K \|s^{mn}_{k^i}\| \geq \varepsilon_0 - b_1 K \beta \delta_{k^i},\end{aligned}$$

and since $\beta \leq \frac{\varepsilon_0}{2b_1 K \delta_{\max}}$, it follows that

$$\|W_k^T \nabla q_k(s^n_{k^i})\| \geq \frac{1}{2}\varepsilon_0.$$

From Lemma 8.1, the above inequality, and the assumption that $\max\{\|C_k\|, \|s^{mn}_{k^i}\|\} \leq \beta\delta_{k^i}$, we have

$$Pred_{k^i} \geq \frac{K_2\varepsilon_0}{2}\min\left\{\frac{\varepsilon_0}{2}, \delta_{k^i}\right\} - K_8\beta\delta_{k^i} + \rho[\|C_k\|^2 - \|C_k + \nabla C_k^T s_{k^i}\|^2].$$

Thus

$$Pred_{k^i} \geq \frac{K_2\varepsilon_0}{4}\min\left\{\frac{\varepsilon_0}{2\delta_{\max}}, 1\right\}\delta_{k^i} + \rho[\|C_k\|^2 - \|C_k + \nabla C_k^T s_{k^i}\|^2].$$

The result follows if we take $K_9 = \frac{K_2\varepsilon_0}{4}\min\{\frac{\varepsilon_0}{2\delta_{\max}}, 1\}$. □

The above lemma shows that at any trial iteration k^i with $\|W_k^T \nabla f_k\| \geq \varepsilon_0$, if it satisfies $\max\{\|C_k\|, \|s^{mn}_{k^i}\|\} \leq \beta\delta_{k^i}$, then the penalty parameter is not increased at this trial iterate.

The following lemma bounds $\|s^{mn}_{k^i}\|$ by $\|C_k\|$ and $\|C_k\|$ by $\|\nabla C_k C_k\|$ for any iteration where the smallest singular value of ∇C_k is not zero.

LEMMA 8.3. *Assume* A1 *and* A2*. If there exists a subsequence $\{k_i\}$ of the iteration sequence such that the sequence of smallest singular values $\{\sigma_{k_i}\}$ is bounded away from zero, then all trial iterates j of any iteration $k \in \{k_i\}$ satisfy*

$$\|s^{mn}_{k^j}\| \leq K_{10}\|C_k\|, \tag{8.3}$$

and for any $k \in \{k_i\}$

$$\|C_k\| \leq K_{11}\|\nabla C_k C_k\|, \tag{8.4}$$

where K_{10} and K_{11} are two positive constants that do not depend on k or j.

Proof. The proof of (8.3) is similar to the proof of Lemma 7.1 of Dennis, El-Alem, and Maciel [13]. The proof of (8.4) follows from the fact that for all $k \in \{k_i\}$, $\|C_k\| \leq \|(\nabla C_k^T \nabla C_k)^{-1}\nabla C_k^T\|\|\nabla C_k C_k\|$, followed by the use of the assumptions. □

From the above two lemmas, if for the subsequence $\{k_i\}$ of the iteration sequence at which the penalty parameter is increased, $\{\sigma_{k_i}\}$ is bounded away from zero, and $\|W_k^T \nabla f_k\| \geq \varepsilon_0$ for all $k \in \{k_i\}$, then

$$\|C_k\| > \beta_1\delta_k \tag{8.5}$$

holds for all $k \in \{k_i\}$, where $\beta_1 = \frac{\beta}{\max\{1, K_{10}\}}$, β is as in Lemma 8.2, and K_{10} is as in (8.3).

From (6.6), (8.3), and (8.4), if $\{k_i^j\}$ is the sequence of iterates at which the penalty parameter is increased and $\{\sigma_{k_i}\}$ is bounded away from zero, then we have for all $k \in \{k_i^j\}$,

$$\rho_k^j \|C_k\| \le K_{12}, \tag{8.6}$$

where K_{12} is a positive constant independent of k.

The following theorem studies the behavior of the iteration sequence when $\{\|C_k\|\}$ converges to zero and $\{\rho_k\}$ is unbounded.

THEOREM 8.4. *Assume* A1–A5. *Assume also that* $\{\rho_k\}$ *is unbounded and* $\{\|C_k\|\}$ *converges to zero. The iteration sequence at which* ρ_k *is increased has a subsequence that satisfies the feasible Mayer–Bliss conditions in the limit.*

Proof. Let $\{k_j\}$ be the iteration sequence at which the penalty parameter is increased. Since $\lim_{k_j \to \infty} \|C_{k_j}\| = 0$, if a subsequence of σ_{k_j} converges to zero, then by Lemma 4.3, the corresponding subsequence of iterates satisfies the feasible Mayer–Bliss conditions in the limit and the proof ends here.

Consider the case where $\{\sigma_{k_j}\}$ is bounded away from zero. Suppose that, for all $k \in \{k_j\}$,

$$\|W_k^T \nabla f_k\| \ge \varepsilon_0 > 0. \tag{8.7}$$

From (8.5), there exist some trial iterates i of k for all $k \in \{k_j\}$, such that $\|C_k\| > \beta_1 \delta_{k^i}$. But because $\lim_{k_j \to \infty} \|C_{k_j}\| = 0$, we have

$$\lim_{k_j \to \infty} \delta_{k_j^i} = 0. \tag{8.8}$$

The rest of the proof is by contradiction. From the method of updating the trust-region radius, $\delta_{k_j^1} \ge \delta_{\min}$. Therefore, the superscript $i \ne 1$ in (8.8). Because $\delta_{k_j^1} \ge \delta_{\min}$, $\|C_k\| > \beta_1 \delta_{k^i}$, and both of $\delta_{k_j^i}$ and C_{k_j} are converging to zero, then for k_j sufficiently large, there must be an $m > 1$ such that $\|C_{k_j}\| > \beta_1 \delta_{k_j^m}$ and $\|C_{k_j}\| \le \beta_1 \delta_{k_j^{m-1}}$, where β_1 is as in (8.5). Using $\delta_{k_j^m} = \alpha_1 \|s_{k_j^{m-1}}\|$ and (8.6), we have

$$\rho_{k_j^{m-1}} \|s_{k_j^{m-1}}\| \le \rho_{k_j^m} \frac{\delta_{k_j^m}}{\alpha_1} \le \rho_{k_j^m} \frac{\|C_{k_j}\|}{\alpha_1 \beta_1} \le \frac{K_{12}}{\alpha_1 \beta_1}.$$

From Lemma 6.3 and the above inequality, we have

$$\begin{aligned} \left| Ared_{k_j^{m-1}} - Pred_{k_j^{m-1}} \right| &\le K_3 [1 + (1 + \beta_1)\ \rho_{k_j^{m-1}}\ \|s_{k_j^{m-1}}\|\ ]\ \|s_{k_j^{m-1}}\|\ \delta_{k_j^{m-1}} \\ &\le K_3\ \left[\ 1 + (1 + \beta_1) \frac{K_{12}}{\alpha_1 \beta_1}\ \right]\ \|s_{k_j^{m-1}}\|\ \delta_{k_j^{m-1}}. \end{aligned}$$

Also, $\|C_{k_j}\| \le \beta_1 \delta_{k_j^{m-1}}$ implies that $\max\{\|C_{k_j}\|, \|s^{mn}_{k_j^{m-1}}\|\} \le \beta \delta_{k_j^{m-1}}$. Hence, from Lemma 8.2, we have

$$Pred_{k_j^{m-1}} \ge K_9 \delta_{k_j^{m-1}}.$$

Therefore, since $s_{k_j^{m-1}}$ was a rejected step,

$$(1 - \eta_1) < \frac{|Ared_{k_j^{m-1}} - Pred_{k_j^{m-1}}|}{Pred_{k_j^{m-1}}} \le \frac{K_3 [1 + (1 + \beta_1) \frac{K_{12}}{\alpha_1 \beta_1}] \|s_{k_j^{m-1}}\|}{K_9}.$$

Hence, $\|s_{k_j^{m-1}}\| > \frac{K_9(1-\eta_1)}{K_3[1+(1+\beta_1)\frac{K_{12}}{\alpha_1\beta_1}]}$ and we obtain

$$\delta_{k_j^m} \geq \alpha_1 \|s_{k_j^{m-1}}\| \geq \frac{\alpha_1 K_9(1-\eta_1)}{K_3[1+(1+\beta_1)\frac{K_{12}}{\alpha_1\beta_1}]}.$$

This means that $\delta_{k_j^m}$ is bounded below. Hence $\{\|C_{k_j}\|\}$ is bounded away from zero. This contradicts the assumption that $\{\|C_k\|\}$ converges to zero and means that for k_j sufficiently large there is no m such that $\|C_{k_j}\| > \beta_1 \delta_{k_j^m}$ holds. Hence, all trial iterates i of k_j satisfy $\|C_{k_j}\| \leq \beta_1 \delta_{k_j^i}$. But this contradicts the fact that k_j is an iterate at which $\rho_{k_j^i}$, for some trial i, is increased. This contradiction implies that the supposition (8.7) was wrong and completes the proof of the theorem. □

From the above lemma, we conclude that, if along the subsequence of the iteration sequence at which ρ_k is increased, the corresponding subsequence of σ_k converges to zero, then it has a subsequence that asymptotically satisfies the feasible Mayer–Bliss conditions. Otherwise, it has a subsequence that satisfies the first-order conditions in the limit.

When $\{\rho_k\}$ is bounded, there must exist a positive integer $\bar{k}$ and a positive constant $\bar{\rho}$ such that for all $k \geq \bar{k}$, $\rho_k = \bar{\rho}$. Without loss of generality we will take $\rho_k = \bar{\rho}$ for all k, whenever we assume that $\{\rho_k\}$ is bounded.

The following theorem studies the asymptotic behavior of the iteration sequence when $\{\rho_k\}$ is bounded.

Theorem 8.5. *Assume* A1–A5. *Assume also that the sequence* $\{\rho_k\}$ *is bounded and* $\lim_{k\to\infty} \|C_k\| = 0$. *Then the iteration sequence has a subsequence that satisfies the feasible Mayer–Bliss conditions in the limit.*

Proof. Suppose that there exists an infinite subsequence $\{k_j\}$ of the iteration sequence such that $\{\sigma_{k_j}\}$ is not bounded away from zero. Then, from Lemma 4.3, there exists a subsequence that satisfies the feasible Mayer–Bliss conditions in the limit.

Let us assume that $\{\sigma_k\}$ is bounded away from zero and suppose that for all k, $\|W_k^T \nabla f_k\| \geq \varepsilon_0 > 0$. Define $\delta' = \min\{\frac{K_9(1-\eta_1)}{K_4\bar{\rho}}, \delta_{\min}\}$. For any k, either $\delta_k \geq \delta'$ or for some trial iterate i, $\delta_{k^i} \in [\alpha_1\delta', \delta']$. Since C_k is converging to zero, we know that in either case the conditions of Lemma 8.2 are satisfied by k^i, if k is large enough, and $Pred_{k^i} \geq K_9 \delta_{k^i}$. Now, if $\delta_{k^i} \in [\alpha_1\delta', \delta']$, then

$$\frac{|Ared_{k^i} - Pred_{k^i}|}{Pred_{k^i}} \leq \frac{K_4\bar{\rho}\delta'}{K_9} \leq (1-\eta_1).$$

Thus the trial step is accepted and $\delta_k = \delta_{k^i}$. This implies that $\delta_k \geq \alpha_1\delta'$ for all k sufficiently large. This means that δ_k is bounded below away from zero.

On the other hand, we have

$$\Phi_k - \Phi_{k+1} = Ared_k \geq \eta_1 Pred_k \geq \eta_1 K_9 \delta_k.$$

If we take the limit as $k \to \infty$, we obtain

$$\lim_{k\to\infty} \delta_k = 0. \tag{8.9}$$

This shows a contradiction. Therefore, the supposition is wrong and we have

$$\lim_{k_j\to\infty} \|W_{k_j}^T \nabla f_{k_j}\| \;=\; 0.$$

This completes the proof of the theorem. □

Let us again state and then prove our main global convergence result, Theorem 5.1.

THEOREM 5.1. *Assume* A1–A5. *Then the sequence of iterates generated by Algorithm* 2.2 *has a subsequence that satisfies one of the Mayer–Bliss stationary conditions of problem* (EQ) *in the limit.*

Proof. The proof follows immediately from Theorems 7.4, 8.4, and 8.5. □

9. Summary and concluding remarks.

We have established a global convergence theory for the class of trust-region-based algorithms suggested by Dennis, El-Alem, and Maciel [13]. This class of algorithm is characterized by generating steps such that their quasi-normal components satisfy a fraction of Cauchy decrease condition on the quadratic model of the linearized constraints. Furthermore, their tangential components satisfy a fraction of Cauchy decrease condition on the quadratic model of the Lagrangian function associated with the problem reduced to the tangent space of the constraints. The augmented Lagrangian is used as a merit function. To update the penalty parameter, a scheme proposed in [16] was used.

Because the two components of the trial step are not necessarily orthogonal, an additional condition on the length of the normal component is needed to prove global convergence. Dennis, El-Alem, and Maciel [13] suggested the condition $\|s_k^n\| \leq K\|C_k\|$. In this paper, we used $\|s_k^n\| \leq K\|s_k^{mn}\|$, where s_k^{mn} is the minimum-norm solution that minimizes $\|C_k + \nabla C_k^T s\|$ inside the trust region δ_k. This condition is equivalent to the above condition whenever ∇C_k has full column rank and allows the full SQP step to be taken when it is inside the trust region.

As pointed out in section 2.3, if at a given iteration k the algorithm generates an infeasible point with $\|C_k\|^2 - \|C_k + \nabla C_k s_k\|^2 = 0$, then it may not be able to move away from that point. We pointed out in Lemma 4.1 that in this case the point is necessarily an infeasible Mayer–Bliss point. Probably, if a good estimate of the Lagrange multiplier vector is used every iteration, or at least at this point, then the algorithm moves away from such points. Avoiding Mayer–Bliss points that are not first-order points is an important issue for algorithms that are designed to handle the lack of linear independence in the gradients of the constraints. This issue indeed deserves to be studied.

The main feature of the global convergence theory presented in this paper is that the gradients of the constraints are allowed to be linearly dependent. We showed that under the general assumptions of section 3, and without the regularity assumption, the iteration sequence has a subsequence that asymptotically satisfies one of four types of stationary conditions. In particular, it asymptotically satisfies either the infeasible Mayer–Bliss conditions that are not infeasible first-order conditions, the feasible Mayer–Bliss conditions that are not first-order conditions, the infeasible first-order conditions, or the first-order conditions.

In a practical implementation of the algorithm, a stopping criterion should be added. To stop their algorithm, Dennis, El-Alem, and Maciel used the condition "if $\|W_k^T \nabla f_k\| + \|\nabla C_k C_k\| \leq \varepsilon_{tol}$, for some $\varepsilon_{tol} > 0$, then terminate." But, because we do not assume that the columns of ∇C_k are linearly independent, the iteration sequence may have no subsequence that asymptotically satisfies the first-order conditions. In other words, it may be the case that no iterate k generated by the algorithm satisfies the above condition. Therefore, other termination conditions should be added. A reasonable stopping rule would be to test for the three kinds of stationary conditions (see section 4).

Our theory requires that the sequence of Lagrange multipliers $\{\lambda_k\}$ be bounded. This means that any update formula that produces bounded multipliers $\{\lambda_k\}$ can be used. However, because we do not make any assumptions on the gradient of the constraints, most formulas for updating the Lagrange multiplier may be undefined or produce an unbounded sequence of multipliers. We suggest computing λ_k as the solution of the following trust-region subproblem:

$$\begin{aligned} &\text{minimize } \|\nabla C_k \lambda + \nabla f_k\| \\ &\text{subject to} \quad \|\lambda\| \le \delta_k. \end{aligned} \tag{9.1}$$

It is clear that the Lagrange multipliers are well defined and satisfy at every iteration k, $\|\lambda_k\| \le \delta_{\max}$.

Another way of enforcing boundedness on the multipliers is to replace the matrix ∇C_k with another matrix $\widehat{\nabla C_k}$ such that for all k, $(\widehat{\nabla C_k})^+$ is uniformly bounded, where $(\widehat{\nabla C_k})^+$ denotes the pseudoinverse matrix of $\widehat{\nabla C_k}$. Then λ_k is obtained as the minimum-norm solution to the following minimization problem:

$$\text{minimize } \|\widehat{\nabla C_k}\lambda + \nabla f_k\|. \tag{9.2}$$

As an example, if we factor ∇C_k using the SVD decomposition, we obtain $\nabla C_k = U_k \Sigma_k V_k^T$. Now for all k, we construct another matrix $\widehat{\Sigma}_k$ from Σ_k by setting to zero all singular values less than a prespecified small constant. Let $\widehat{\nabla C_k} = U_k \widehat{\Sigma}_k V_k^T$. We have for all k that $(\widehat{\nabla C_k})^+$ is uniformly bounded. Therefore, using the global assumptions (see section 4), problem (9.2) will produce a bounded sequence of Lagrange multipliers.

There are many interesting questions that have not been properly discussed in the literature. The problem of the unboundedness of the multipliers and the Hessian matrices is one question that deserves to be investigated.

Another important question is related to the assumptions stated in section 3 of this paper. In the general assumptions, it is assumed that Ω is a convex set that contains x_k and $x_k + s_k$ for all trial steps and that the objective function, the constraints, and the first and the second derivatives of the constraints are bounded in Ω. These are assumptions on the behavior of the algorithm, and not assumptions on the problem. No reasonable, sufficient condition, under which these algorithmic assumptions hold, has appeared in the literature. Of course, a sufficient condition is that Ω is compact. But this is again a condition on the algorithm and not on the problem. In unconstrained optimization, the usual reasonable condition is "bounded level set." But in the equality constrained case, the difficulty is serious. It is not clear what should be bounded. This important question of equality constrained optimization deserves to be investigated.

Acknowledgments. The author thanks Rice University for the congenial scientific atmosphere that it provided. The author is also greatly indebted to Richard Byrd, two anonymous referees, and the associate editor for their careful and insightful reading of an earlier version of this paper.

REFERENCES

[1] N. Alexandrov, *Multi-level Algorithms for Nonlinear Equations and Equality Constrained Optimization*, Tech. Rep. 93-20, Department of Computational and Applied Mathematics, Rice University, Houston, TX, May 1993.

[2] N. Alexandrov, *Multi-level Algorithms for Nonlinear Equations and Equality Constrained Optimization*, Ph.D. thesis, Rice University, Houston, TX, 1993.

[3] N. Alexandrov and J. E. Dennis, Jr., *Multi-Level Algorithms for Nonlinear Optimization*, Tech. Rep. 94-24, Department of Computational and Applied Mathematics, Rice University, Houston, TX, June 1994.

[4] L. T. Biegler, J. Nocedal, and C. Schmid, *A reduced Hessian method for large-scale constrained optimization*, SIAM J. Optim., 5 (1995), pp. 314–347.

[5] G. A. Bliss, *Normality and abnormality in the calculus of variations*, Trans. Amer. Math. Soc., 43 (1938), pp. 365–376.

[6] J. V. Burke, *A sequential quadratic programming method for potentially infeasible mathematical programs*, J. Math. Anal. Appl., 139 (1989), pp. 319–351.

[7] J. V. Burke, *An exact panelization viewpoint of constrained optimization*, SIAM J. Control Optim., 29 (1991), pp. 968–998.

[8] J. V. Burke, *A robust trust region method for constrained nonlinear programming problems*, SIAM J. Optim., 2 (1992), pp. 325–347.

[9] R. H. Byrd, *Robust Trust Region Methods for Nonlinearly Constrained Optimization*, presented at SIAM Conference on Optimization, Houston, TX, 1987.

[10] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *A trust region algorithm for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 24 (1987), pp. 1152–1170.

[11] R. Carter, *Multi-Model Algorithms for Optimization*, Ph.D. thesis, Rice University, Houston, TX, 1986.

[12] M. R. Celis, J. E. Dennis, and R. A. Tapia, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization 1984, SIAM, Philadelphia, PA, 1985.

[13] J. E. Dennis, Jr., M. M. El-Alem, and M. C. Maciel, *A global convergence theory for general trust-region-based algorithms for equality constrained optimization*, SIAM J. Optim., 7 (1997), pp. 177–207.

[14] J. E. Dennis, Jr., and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983; Russian edition: Mir, Moscow, O. Burdakov, trans., 1988.

[15] J. E. Dennis, Jr., and L. N. Vicente, *On the convergence theory of trust-region-based algorithms for equality-constrained optimization*, SIAM J. Optim., 7 (1997), pp. 927–950.

[16] M. M. El-Alem, *A Global Convergence Theory for a Class of Trust Region Algorithms for Constrained Optimization*, Ph.D. thesis, Rice University, Houston, TX, 1988.

[17] M. M. El-Alem, *A global convergence theory for the Celis–Dennis–Tapia trust-region algorithm for constrained optimization*, SIAM J. Numer. Anal., 28 (1991), pp. 266–290.

[18] M. M. El-Alem, *A robust trust-region algorithm with a nonmonotonic penalty parameter scheme for constrained optimization*, SIAM J. Optim., 5 (1995), pp. 348–378.

[19] M. M. El-Alem, *Convergence to a second-order point of a trust-region algorithm with a nonmonotonic penalty parameter for constrained optimization*, J. Optim. Theory Appl., 91 (1996), pp. 61–79.

[20] M. M. El-Alem and R. A. Tapia, *Numerical experience with a polyhedral norm CDT trust-region algorithm*, J. Optim. Theory Appl., 85 (1995), pp. 575–592.

[21] M. El-Hallabi, *A Global Convergence Theory for Arbitrary Norm Trust-Region Algorithms for Equality Constrained Optimization*, Tech. Rep. TR93-60, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1993.

[22] M. El-Hallabi and R. Tapia, *A Global Convergence Theory for Arbitrary Norm Trust-Region Method for Nonlinear Eequations*, Tech. Rep. TR93-41, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1993.

[23] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, New York, 1987.

[24] P. E. Gill, W. Murray, and M. Wright, *Some theoretical properties of an augmented lagrangian merit function*, in Advances in Optimization and Parallel Computing, Elsevier, New York, 1992, pp. 127–143.

[25] M. Lalee, *Algorithms for Nonlinear Optimization*, Ph.D. thesis, Northwestern University, Evanston, IL, 1993.

[26] M. Lalee, J. Nocedal, and Plantenga, *On the Implementation of an Algorithm for Large-Scale Equality Constrained Optimization*, Tech. Rep. 93, EECS Department, Northwestern University, Evanston, IL, Oct. 1993.

[27] K. Levenberg, *A method for the solution of certain problems in least squares*, Quart. Appl. Math., 2 (1944), pp. 164–168.

[28] M. C. Maciel, *A Global Convergence Theory for a General Class of Trust Region Algorithm for Equality Constrained Optimization*, Ph.D. thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1992.

[29] O. L. Mangasarian, *Nonlinear Programming*, McGraw-Hill, New York, 1969.

[30] D. W. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters*, J.

SIAM, 11 (1963), pp. 431–441.

[31] A. Mayer, *Begrudung der lagrange'schen multiplicatorenmethode in der variationsrechnung*, Math. Annal., 26 (1886), pp. 74–82.

[32] A. Miele, E. E. Cragg, and A. V. Levy, *Use of the augmented penalty function in mathematical programming problems: Part* 2, J. Optim. Theory Appl., 8 (1971), pp. 131–153.

[33] A. Miele, H. Y. Huang, and J. C. Heideman, *Sequential gradient-restoration algorithm for the minimization of constrained functions: Ordinary and conjugate gradient versions*, J. Optim. Theory Appl., 4 (1969), pp. 213–243.

[34] A. Miele, A. V. Levy, and E. E. Cragg, *Modifications and extensions of the conjugate gradient-restoration algorithm for mathematical programming problems*, J. Optim. Theory Appl., 7 (1971), pp. 450–472.

[35] J. J. Moré, *The Levenberg-Marquardt algorithm: Implementation and theory*, in Numerical Analysis, Lecture Notes in Math. 630, G. A. Watson, ed., Springer-Verlag, Berlin, New York, 1977, pp. 105–116.

[36] J. J. Moré, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming. The State of the Art, A. Bachem, M. Grotschel, and B. Korte, eds., Springer-Verlag, New York, 1983, pp. 258–287.

[37] E. O. Omojokun, *Trust Region Strategies for Optimization with Nonlinear Equality and Inequality Constraints*, Ph.D. thesis, University of Colorado, Boulder, 1989.

[38] T. Plantenga, *Large-Scale Nonlinear Constrained Optimization Using Trust Regions*, Ph.D. thesis, Northwestern University, Evanston, IL, 1995.

[39] M. J. D. Powell, *A new algorithm for unconstrained optimization*, in Nonlinear Programming, J. Rosen, O. Mangasarian, and K. Ritter, eds., Academic Press, New York, 1970, pp. 31–65.

[40] M. J. D. Powell, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. Mangasarian, R. Meyer, and S. Robinson, eds., Academic Press, New York, 1975, pp. 1–27.

[41] M. J. D. Powell and Y. Yuan, *A trust region algorithm for equality constrained optimization*, Math. Programming, 49 (1991), pp. 189–211.

[42] G. A. Shultz, R. B. Schnabel, and R. H. Byrd, *A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties*, SIAM J. Numer. Anal., 22 (1985), pp. 47–67.

[43] A. Vardi, *A trust region algorithm for equality constrained minimization: Convergence properties and implementation*, SIAM J. Numer. Anal., 22 (1985), pp. 575–591.

[44] L. N. Vicente, *Trust-Region Interior-Point Algorithms for a Class of Nonlinear Programming Problems*, Ph.D. thesis, Rice University, Houston, TX, 1996.

[45] K. A. Williamson, *A Robust Trust Region Algorithm for Nonlinear Programming*, Ph.D. thesis, Rice University, Houston, TX, 1990.

[46] Y. Yuan, *A Dual Algorithm for Minimizing a Quadratic Function with Two Quadratic Constraints*, Report DAMTP-NA3, University of Cambridge, Cambridge, UK, 1988.

[47] Y. Yuan, *On a subproblem of trust-region algorithm for constrained optimization*, Math. Programming, 47 (1990), pp. 53–63.

[48] Y. Yuan, *On the convergence of a new trust region algorithm*, Numer. Math., 70 (1995), pp. 515–539.

[49] J. Zhang, N. Kim, and L. Lasdon, *An improved successive linear programming algorithm*, Management Sci., 31 (1985), pp. 1312–1331.

[50] J. Z. Zhang and D. T. Zhu, *Projected quasi-Newton algorithm with trust region for constrained optimization*, J. Optim. Theory Appl., 67 (1990), pp. 369–393.

[51] Y. Zhang, *Computing a Celis-Dennis-Tapia trust-region step for equality constrained optimization*, Math. Programming, 55 (1992), pp. 109–124.

SIAM J. Optim.
Vol. 9, No. 4, pp. 991–1009

EXPRESSING COMPLEMENTARITY PROBLEMS IN AN ALGEBRAIC MODELING LANGUAGE AND COMMUNICATING THEM TO SOLVERS*

MICHAEL C. FERRIS[†], ROBERT FOURER[‡], AND DAVID M. GAY[§]

Dedicated to John Dennis on his 60th birthday, in appreciation of his many contributions to the discipline of nonlinear optimization

Abstract. Diverse problems in optimization, engineering, and economics have natural formulations in terms of complementarity conditions, which state (in their simplest form) that either a certain nonnegative variable must be zero or a corresponding inequality must hold with equality, or both. A variety of algorithms has been devised for solving problems expressed in terms of complementarity conditions. It is thus attractive to consider extending algebraic modeling languages, which are widely used for sending ordinary equations and inequality constraints to solvers, so that they can express complementarity problems directly. We describe an extension to the AMPL modeling language that can express the most common complementarity conditions in a concise and flexible way, through the introduction of a single new "complements" operator. We present details of an efficient implementation that incorporates an augmented presolve phase to simplify complementarity problems, and that converts complementarity conditions to a canonical form convenient for solvers.

Key words. complementarity, algebraic modeling languages, optimization

AMS subject classifications. 49J40, 65K10, 90C33

PII. S105262349833338X

1. Introduction.

After equations and inequalities, complementarity conditions are among the most common kinds of constraints formulated in terms of decision variables. In their simplest form, they state that either a certain nonnegative variable must be zero or a corresponding inequality must hold with equality, or both.

Complementarity conditions play a key role in the theory of convex optimization, being the natural form for optimality conditions in inequality-constrained problems. They also arise in a variety of applications from engineering to economics. As a result, various algorithms have been proposed to solve *complementarity problems* whose constraints consist partly or entirely of complementarity conditions. Several of these algorithms have been developed into large-scale, robust implementations of solvers for complementarity problems.

The work described in this paper is concerned not with the details of any particular algorithm for complementarity problems but with the broader concern of helping people communicate such problems to a variety of solvers. We consider specifically the possibilities for extending algebraic modeling languages, which are widely used in communicating equality and inequality constraints, so as to express linear and nonlinear complementarity conditions. We show how the introduction of a "complements" operator enables a modeling language to express a variety of these conditions clearly

[*] Received by the editors January 29, 1998; accepted for publication (in revised form) January 15, 1999; published electronically September 24, 1999.

http://www.siam.org/journals/siopt/9-4/33338.html

[†] Computer Sciences Department, University of Wisconsin, Madison, WI 53706 (ferris@cs.wisc.edu). The research of this author was supported in part by National Science Foundation grant CCR-9619765.

[‡] Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208-3119 (4er@iems.nwu.edu). The research of this author was supported in part by National Science Foundation grants DMI-9414487 and DMI-9800077.

[§] Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 (dmg@research.bell-labs.com).

and concisely for human modelers while remaining amenable to efficient translation to forms required by solvers.

As a practical illustration of this approach, we describe its implementation in the AMPL modeling language. We touch upon a number of practical concerns, such as the extension of the presolve phase for simplifying problems and the design of a canonical form for communicating problems to solvers.

Relevant background in complementarity and in modeling languages is summarized in section 2 below. The kinds of complementarity conditions that we aim to represent are surveyed in section 3, along with a critique of previous representations for complementarity in modeling languages. Our new AMPL representation is then presented in section 4 and is evaluated with respect to specific design criteria.

The remainder of this paper addresses further aspects of our complementarity enhancements to the AMPL design, including extension of the presolve phase (section 5), canonical forms for communication with solvers (section 6), and extensions of related constraint notations and representations (section 7). All of these features have been implemented as part of a recent release of the AMPL software; a demonstration version, including a link to the PATH solver, can be tried out through a Web interface, as explained in our concluding remarks in section 8.

2. Background.

The significance of our topic stems from the existence of applications and algorithms for complementarity problems, together with modeling languages capable of expressing such problems. We begin by briefly reviewing each of these areas.

2.1. Applications.

Complementarity relations arise in a variety of engineering and economics applications [17, 18, 26], most commonly to express an equilibrium of quantities such as forces or prices.

One standard application in engineering arises in contact mechanics, where complementarity expresses the fact that friction occurs only when two bodies are in contact. Other applications are found in structural mechanics, structural design, traffic equilibrium, and optimal control [18].

Interest among economists in solving complementarity problems is due in part to increased use of computational general equilibrium models [34], where complementarity is used to express Walras's law, and to the equivalence of various games to complementarity problems [10].

Some generalizations of nonlinear programming, such as multilevel optimization—in which auxiliary objectives are to be minimized—may be reformulated as problems with complementarity conditions [1, 2, 3, 14]. There is a growing literature on these and other mathematical programming problems with equilibrium constraints, or MPECs [28, 29].

2.2. Solvers.

The demands of applications have motivated a variety of algorithms for complementarity problems [4]. Modelers currently have a choice of robust and efficient implementations, such as MILES [33] and PATH [12, 16].

Recent research in this area can be divided into two general algorithmic approaches [4]. One approach transforms complementarity problems so that they can be solved using existing methods for differentiable optimization or equation solving. The other generalizes existing methods—including Newton-type methods, path search methods, projection and proximal methods, and interior-point methods—to apply to complementarity problems of certain kinds. In particular, many standard techniques have been extended to deal with the special forms of nonsmoothness that naturally appear when formulating complementarity problems. No comprehensive survey of algorithms for complementarity problems is currently available, but extensive references to algorithms can be found in [17, 18, 26].

2.3. Modeling languages. Constructing problem descriptions suitable for solvers is a substantial task that can easily consume more time and expense than finding problem solutions. Modeling languages have become a popular means of streamlining this task. They allow people to work with general models expressed in a natural and convenient form while leaving for the language processor the work of translating models and communicating problem instances to solvers.

We are concerned in particular with *algebraic* modeling languages, which describe expressions, equations, and inequalities by use of familiar algebraic terms and operators. As an example, a collection of inequality constraints defined by

$$\sum_{r\in\mathcal{R}} T_{ru} \;\geq\; q_c^0 \prod_{j\in\mathcal{M}} (P_{ju}/p_j^0)^{e_{cj}} \quad \text{for all} \quad c\in\mathcal{C}, u\in\mathcal{U}$$

could be transcribed to the AMPL language [22, 23] as

```
subject to ineq1 {c in C, u in U}:
   sum {r in R} T[r,u] >=
      q0[c] * prod {j in M} (P[j,u] / p0[j]) ** e[c,j];
```

or, using somewhat more mnemonic identifiers, as

```
subject to CrudeSupply {cr in CRUDES, u in USERS}:
   sum {r in REFIN} Trans[r,u] >=
      q0[cr] * prod {co in COMOD} (P[co,u] / p0[co]) ** esub[cr,co];
```

Other AMPL statements define the index sets, numerical data, and variables that appear in such an expression, as seen in the illustration of an AMPL complementarity problem in Figure 1 of section 4. Algebraic languages, such as AMPL, AIMMS [5], GAMS [6, 8], and LINGO [35], are currently the most popular type of modeling language for describing linear and nonlinear optimization problems.

With the specification of the objective omitted, algebraic modeling languages are equally useful for describing problems of finding feasible solutions to systems of equality and inequality constraints. We thus approach complementarity conditions as an additional kind of constraint to which modeling languages may be extended. The design of any such extension involves many tradeoffs between the goal of making the language natural and convenient for people and the requirement that the language be processed with reasonable efficiency by a computer system. We have previously described the tradeoffs involved in various extensions to AMPL [19, 21]; similar considerations have influenced our extensions for complementarity, as we explain next.

3. Design issues. To motivate our choice of a modeling language representation for complementarity conditions, we first describe the variety of conditions that we want the language to be able to represent. We then take a critical look at representations that have been used previously in the GAMS and AMPL languages.

3.1. Forms of complementarity. A few fundamental forms account for almost all of the complementarity conditions that people want to use in models. The simplest of these forms can be written in terms of a variable x_j and an associated function $g_j(x)$, where x is a vector of variables that contains x_j.

The *classical* form of complementarity condition is the one described at the beginning of this paper. It requires that

$$\begin{aligned} &\text{either } x_j = 0 \quad \text{and} \quad g_j(x) \geq 0 \\ &\quad\;\; \text{or } x_j > 0 \quad \text{and} \quad g_j(x) = 0. \end{aligned} \tag{1}$$

This condition can be viewed as consisting of the inequalities $x_j \geq 0$ and $g_j(x) \geq 0$, together with the complementarity restriction that at least one of these must hold with equality. The complementarity restriction can be written equivalently as

$$x_j \cdot g_j(x) = 0$$

or as the nonsmooth equation

$$\min(x_j, g_j(x)) = 0.$$

If conditions of this sort are imposed for every $j \in \mathcal{J}$, then they may also be written jointly as $x \geq 0$, $g(x) \geq 0$, and $x^T g(x) = 0$.

The more general *mixed* complementarity condition on a bounded variable $\ell_j \leq x_j \leq u_j$ and a function $g_j(x)$ states that

$$\begin{aligned} &\text{either } x_j = \ell_j \quad \text{and} \quad g_j(x) \geq 0 \\ &\quad\;\; \text{or } x_j = u_j \quad \text{and} \quad g_j(x) \leq 0 \\ &\quad\;\; \text{or } \ell_j < x_j < u_j \quad \text{and} \quad g_j(x) = 0. \end{aligned} \tag{2}$$

This form generalizes the classical complementarity condition, which is the special case in which $\ell_j = 0$ and $u_j = \infty$. It can be expressed equivalently as the variational inequality problem of finding $x_j \in [\ell_j, u_j]$ such that

$$(y_j - x_j) \cdot g_j(x) \geq 0 \quad \text{for all} \quad y_j \in [\ell_j, u_j],$$

or, where there is such a condition for each $j \in \mathcal{J}$, as the joint problem of finding $x \in [\ell, u]$ such that $(y - x)^T g(x) \geq 0$ for all $y \in [\ell, u]$. A mixed complementarity condition can be split into two of the classical conditions, but only through the addition of auxiliary variables. Thus it is desirable for a modeling language to represent mixed complementarity directly, rather than requiring that all mixed conditions be transformed to classical ones. The greater simplicity of classical complementarity (as in (1)) argues that it should also be represented directly, however, rather than having to be written as a special case of the mixed form with an infinite bound.

For completeness, our collection of fundamental complementarity conditions also includes the trivial case

$$x_j \text{ “free”} \quad \text{and} \quad g_j(x) = 0, \tag{3}$$

which can be seen to be another special case of mixed complementarity, with $\ell_j = -\infty$ and $u_j = +\infty$.

The above forms may be extended by substituting a function $f_j(x)$ for the individual variable x_j. Thus a *generalized classical* complementarity condition can be written

$$\begin{aligned} &\text{either } f_j(x) = 0 \quad \text{and} \quad g_j(x) \geq 0 \\ &\quad\;\; \text{or } f_j(x) > 0 \quad \text{and} \quad g_j(x) = 0, \end{aligned} \tag{4}$$

and a *generalized mixed* complementarity condition has the form

$$
\begin{aligned}
&\text{either } f_j(x) = \ell_j \quad \text{and} \quad g_j(x) \geq 0\\
&\quad\text{or } f_j(x) = u_j \quad \text{and} \quad g_j(x) \leq 0\\
&\quad\text{or } \ell_j < f_j(x) < u_j \quad \text{and} \quad g_j(x) = 0.
\end{aligned} \tag{5}
$$

Complementarity conditions in these forms can be transformed to the simpler forms (1) and (2), but only by adding a variable and a defining equation. Thus it is desirable that a modeling language be able to directly represent these forms as well.

The above forms allow a modeler to express not only models that are well formed, solvable, and stable but also models that are poorly specified or badly behaved. For $g_j(x)$ as simple a function as $1 - x_j$, the complementarity condition (1) is equivalent to specifying that x_j can take only the values zero and one. This gives some indication of the difficulties associated with solving complementarity problems; the "tightness" requirement is combinatorial in nature and the solution set of a complementarity problem need not be convex or even connected.

It is possible to avoid undesirably hard cases by placing some restrictions on the functions involved. Just as there are classes of well-behaved nonlinear optimization problems that involve convex functions, for complementarity problems there is a corresponding notion of a monotone function g_j, which satisfies

$$(y - x)^T (g_j(y) - g_j(x)) \geq 0$$

for all x and y [26, 32]. Current modeling languages largely avoid such restrictions, however, in the interest of keeping their design simple and general. Especially in working with nonlinear problems, a modeler is expected to be aware that solvers frequently have difficulties if the model is poorly specified or if the initial point is far from a solution. Some assistance may be provided by routines that test functions for desirable properties, but they are typically incorporated into individual solvers or related analysis tools such as MProbe [9].

3.2. Modeling language representations. The GAMS modeling language [6, 8] was the first (to our knowledge) to provide for specification of complementarity problems [15]. As explained in [34], GAMS does not express complementarity through any modification to its constraint syntax, but rather by an extension to its model-defining statement. The list of constraints in its `model` statement is generalized to allow the specification of complementary constraint-variable pairs, as in the following example from `pies.gms` in MCPLIB [13]:

```
model pies / delc.c, delo.o, delct.ct, delot.ot, dellt.lt, delht.ht,
   dembal.p, cmbal.cv, ombal.ov, lmbal.lv, hmbal.hv, ruse.mu /;
```

The specification `delc.c`, for example, indicates that the constraints `delc`,

```
delc(creg,ctyp) ..
   ccost(creg,ctyp) + sum(R, cruse(R,creg,ctyp) * mu(R)) =g= cv(creg);
```

are complementary to the variables `c` having lower bounds 0,

```
positive variables
   c(creg,ctyp), ...
```

and having upper bounds assigned from a data table,

```
c.up(creg,ctyp) = cmax(creg,ctyp);
```

From the fact that `c` has finite lower and upper bounds, GAMS infers that a certain mixed complementarity condition is intended; from the expression in the `delc` constraint statement, GAMS determines what we have been calling the function $g_j(x)$. Thus explicit conditions of the form (2) need not be added to the model. Analogous inferences allow variables that have only one finite bound to induce classical complementary conditions of the form (1).

These simple conventions provide sufficient expressiveness to describe a considerable variety of applications, as evidenced by the over 50 GAMS complementarity models collected in MCPLIB. The design of these extensions also promotes the reuse of equations previously declared, thus helping modelers to transform existing models into the complementarity framework.

Nevertheless, several aspects of the GAMS approach remain problematical. A full description of any one complementarity condition tends to be spread over several sections of the GAMS model, as seen in the example above. Generalized complementarity conditions can only be represented via transformations to simpler forms. Finally, and most seriously, the sense of the inequality in a complementary constraint is determined from the bounds on the corresponding variable, not by the inequality actually written in the statement of the constraint. As a result, both mixed and classical complementarity conditions may be interpreted by the GAMS processor in ways that are counterintuitive to modelers.

For the mixed case, the function $g_j(x)$ in (2) must be specified by means of a GAMS constraint declaration, even though it is not subject to any inequality. For example, although the `delc` statement above appears to define `=g=` ($\geq$) constraints, the implied complementarity condition allows the left-hand side of `delc(creg,ctyp)` to be less than the right-hand side when the corresponding variable `c(creg,ctyp)` is at upper bound. (The GAMS result listing marks a constraint as "redefined" if it is violated by the solution in this way.)

For the classical case, it is up to the modeler to correctly state the inequality on $g_j(x)$ in (1). As another example (also from `pies.gms`), the nonnegative variables `ct(creg,users)` are defined as being complementary to the constraints

```
delct(creg,users) ..
   ctcost(creg,users) + cv(creg) =g= p("C",users);
```

Because the `ct` variables have finite lower bounds but not finite upper bounds, the relational operator in this case *must* be `=g=`. The mathematically equivalent constraint

```
delct(creg,users) ..
   p("C",users) =l= ctcost(creg,users) + cv(creg);
```

is rejected as an error, because the relational operator `=l=` ($\leq$) is not compatible with complementary variables having only a finite lower bound. This distinction is hard to impress upon modelers, who see the above statements as two ways of saying the same thing.

A similar complementarity representation has been implemented in [11, Chap. 2] for the AMPL modeling language [22, 23], although with some differences in the nature of the extension. Complementarity is indicated by writing a constraint in the equivalent multiplicative form $x_j \cdot g_j(x) = 0$, with bounds on the variable x_j specified in the declaration for the variable. Thus no new syntax is added to any part of the AMPL language (a key requirement of the design in [11]) and the existing AMPL translator can process the model and create a problem file in its usual format. Detection of complementarity conditions is left to the solver, or more accurately to the AMPL driver (or interface routines) for the solver. The driver examines the expression tree for each constraint to determine the variable x_j and then generates an appropriate complementarity constraint for the solver, depending on which bounds of x_j are finite, using much the same logic as the GAMS implementation.

This design also has many of the same drawbacks as the GAMS one. Most seriously, constraints that appear in the model are not necessarily enforced by the complementarity solver. The conditions actually enforced must be inferred from information that is partly in one place (a constraint) and partly in another (a variable's bounds). Generalized complementarity conditions must be handled by transformation to a simpler form.

4. A new representation.

In creating a new form for complementarity conditions, we have sought to address the drawbacks of previous designs while preserving the existing strengths of the AMPL language. We begin this section by describing the representation that we ended up choosing. We then consider the extent to which our representation satisfies a range of design criteria.

Of particular importance for our discussion is the variety of arithmetic constraint expressions that AMPL recognizes. They can be summarized as

$$\begin{array}{l} \textit{expr}_1 \texttt{ <= } \textit{expr}_2, \\ \textit{expr}_1 \texttt{ >= } \textit{expr}_2, \\ \textit{expr}_1 \texttt{ = } \textit{expr}_2, \\ \textit{const}_1 \texttt{ <= } \textit{expr} \texttt{ <= } \textit{const}_2, \\ \textit{const}_1 \texttt{ >= } \textit{expr} \texttt{ >= } \textit{const}_2, \end{array}$$

where *expr* is any valid arithmetic expression, possibly involving variables (linearly or nonlinearly), and *const* is an arithmetic expression that does not contain variables.

Illustrations in this section are taken from `pies.mod`, the previous GAMS example's AMPL counterpart, which is shown in Figures 1 and 2. Additional AMPL complementarity models and corresponding data files can be found in MCPLIB [13] and at http://www.ampl.com/ampl/NEW/COMPLEMENT/.

4.1. Design specifics.

The key to our design is the realization that the different complementarity forms (1), (2), and (3) have the same general structure. In each case, a variable is complementary, in some sense, to a function of variables; and in each case, exactly two inequalities are involved (counting one equality as two inequalities). The function can be defined by a modeling language expression, and the inequalities are corresponding modeling language constraints. The same observations apply to the generalized forms (4) and (5), except that the variable is replaced by a second function.

These observations suggest that all of the fundamental complementarity conditions identified in section 3.1 can be represented by AMPL expressions of the form

$$\textit{item}_1 \texttt{ complements } \textit{item}_2$$

The keyword `complements` is a new operator. Its operands *item*$_1$ and *item*$_2$ may be AMPL arithmetic expressions, or they may be AMPL arithmetic constraints of any of the types listed above, provided that together they contain exactly two inequalities. A solution satisfies such an expression if it satisfies the constraints among the operands to `complements` and also the appropriate kind of complementarity between the operands.

If a constraint of this new kind has two single inequality operands, as in

```
subject to delct {c in CREG, u in USERS}:
```

```
set COMOD := {"C","L","H"};   # coal and light and heavy oil

set R;       # resources
set CREG;    # coal producing regions
set OREG;    # crude oil producing regions
set CTYP;    # increments of coal production
set OTYP;    # increments of oil production
set REFIN;   # refineries
set USERS;   # consumption regions

param rmax {R};               # maximum resource usage
param cmax {CREG,CTYP};       # coal prod. limits
param omax {OREG,OTYP};       # oil prod. limits
param rcost {REFIN};          # refining cost
param q0 {COMOD};             # base demand for commodities
param p0 {COMOD};             # base prices for commodities
param demand {COMOD,USERS};   # computed at optimality
param output {REFIN,COMOD};   # % output of light/heavy oil
param esub {COMOD,COMOD};     # cross-elasticities of substitution
param cruse {R,CREG,CTYP};    # resource use in coal prod
param oruse {R,OREG,OTYP};    # resource use in oil prod
param ccost {CREG,CTYP};      # coal production cost
param ocost {OREG,OTYP};      # oil production cost
param ctcost {CREG,USERS};    # coal transportation costs
param otcost {OREG,REFIN};    # crude oil transportation costs
param ltcost {REFIN,USERS};   # light oil transportation costs
param htcost {REFIN,USERS};   # heavy oil transportation costs

var C {CREG, CTYP};      # coal production
var O {OREG, OTYP};      # oil production
var Ct {CREG, USERS};    # coal transportation levels
var To {OREG, REFIN};    # crude oil transportation levels
var Lt {REFIN, USERS};   # light transportation levels
var Ht {REFIN, USERS};   # heavy transportation levels
var P {COMOD, USERS};    # commodity prices
var Mu {R};              # dual to ruse: marginal utility
var Cv {CREG};           # dual to cmbal
var Ov {OREG};           # dual to ombal
var Lv {REFIN};          # dual to lmbal
var Hv {REFIN};          # dual to hmbal
```

FIG. 1. *An AMPL model of a complementarity problem, part* 1 : *Declarations of sets, numerical data, and decision variables.*

```
      0 <= Ct[c,u]  complements  ctcost[c,u] + Cv[c] >= P["C",u];
```

then it specifies a classical complementarity condition. Both inequalities must hold, at least one with equality.

If a constraint of this kind has instead one double inequality operand and one expression operand, as in

```
   subject to delc {c in CREG, t in CTYP}:
      0 <= C[c,t] <= cmax[c,t]  complements
      ccost[c,t] + sum {res in R} cruse[res,c,t] * Mu[res] - Cv[c];
```

```
subj to delc {c in CREG, t in CTYP}:
  0 <= C[c,t] <= cmax[c,t] complements
  ccost[c,t] + (sum {res in R} cruse[res,c,t] * Mu[res]) - Cv[c];

subj to delo {o in OREG, t in OTYP}:
  0 <= O[o,t] <= omax[o,t] complements
  ocost[o,t] + (sum {res in R} oruse[res,o,t] * Mu[res]) - Ov[o];

subj to delct {c in CREG, u in USERS}:
  0 <= Ct[c,u] complements
  ctcost[c,u] + Cv[c] >= P["C",u];

subj to delot {o in OREG, r in REFIN}:
  0 <= To[o,r] complements
  otcost[o,r] + rcost[r] + Ov[o] >=
     output[r,"L"] * Lv[r] + output[r,"H"] * Hv[r];

subj to dellt {r in REFIN, u in USERS}:
  0 <= Lt[r,u] complements
  ltcost[r,u] + Lv[r] >= P["L",u];

subj to delht {r in REFIN, u in USERS}:
  0 <= Ht[r,u] complements
  htcost[r,u] + Hv[r] >= P["H",u];

subj to dembal {co in COMOD, u in USERS}:  # excess supply of product
  .1 <= P[co,u] complements
  (if co = "C" then sum {c in CREG} Ct[c,u]) +
  (if co = "L" then sum {r in REFIN} Lt[r,u]) +
  (if co = "H" then sum {r in REFIN} Ht[r,u]) >=
     q0[co] * prod {cc in COMOD} (P[cc,u]/p0[cc])**esub[co,cc];

subj to cmbal {c in CREG}:                 # coal material balance
  Cv[c] complements
  sum {t in CTYP} C[c,t] = sum {u in USERS} Ct[c,u];

subj to ombal {o in OREG}:                 # oil material balance
  Ov[o] complements
  sum {t in OTYP} O[o,t] = sum {r in REFIN} To[o,r];

subj to lmbal {r in REFIN}:                # light material balance
  Lv[r] complements
  sum {o in OREG} To[o,r] * output[r,"L"] = sum {u in USERS} Lt[r,u];

subj to hmbal {r in REFIN}:                # heavy material balance
  Hv[r] complements
  sum {o in OREG} To[o,r] * output[r,"H"] = sum {u in USERS} Ht[r,u];

subj to ruse {res in R}:                   # resource use constraints
  0 <= Mu[res] complements
  rmax[res] >=
     sum {c in CREG, t in CTYP} C[c,t] * cruse[res,c,t] +
     sum {o in OREG, t in OTYP} O[o,t] * oruse[res,o,t];
```

FIG. 2. *An AMPL model of a complementarity problem, part* 2*: Declarations of complementarity conditions.*

then it specifies a mixed complementarity condition. Again both inequalities must hold, but the nature of the complementarity is somewhat different. Either the lower inequality holds with equality and the expression is nonnegative, or the upper inequality holds with equality and the expression is nonpositive, or neither inequality holds with equality and the expression is zero.

A single equality constraint may take the place of the double inequality:

```
subject to cmbal {c in CREG}:
   Cv[c]  complements
   sum {t in CTYP} C[c,t] = sum u in USERS Ct[c,u];
```

This form of constraint merely imposes the equality. It has no effect on the expression operand, which could just as well be omitted (along with the `complements` operator). It does have some value in exhibiting “square” models (like `pies.mod`), where each constraint is paired with a different complementary variable; squareness is required by some solvers, as discussed further in section 6.2.

Although the first operand to `complements` in the above examples involves only a single variable, our definitions make no mention of this fact, and indeed it is not a requirement of our representation. So long as the total number of inequality operators is two, our representation allows each operand to be any arithmetic expression or constraint.

4.2. Design criteria. Our representation’s introduction of the `complements` operator is valuable in several respects. Its presence clearly distinguishes complementarity constraints from other types, and its operands contain all of the information necessary to define a complementarity condition. The definition of an AMPL complementarity constraint (or indexed collection of constraints) thus appears all in one place, rather than being divided among different statements, as in earlier designs.

There is also no question as to which kind of complementarity is intended by our representation, since the classical and mixed forms are readily distinguished by the position of the inequalities relative to the `complements` operator. Nor is there any question as to which two inequalities are implied, since both appear explicitly as operands to `complements`. Earlier designs’ practice of inferring such information from the number of finite bounds is avoided entirely. At the same time, the interpretation of existing AMPL constraint forms—ones that do not contain the `complements` operator—is left unchanged, and existing models are unaffected.

The incorporation of existing AMPL expression and constraint forms into the new representation is also valuable. By allowing operands to `complements` to be any AMPL arithmetic expressions and constraints, subject only to the two-inequality restriction, we keep our language rules simple to apply and easy to remember. The constraint `delct` above, for example, may be written in any obviously equivalent fashion, such as

```
subject to delct {c in CREG, u in USERS}:
   Ct[c,u] >= 0 complements ctcost[c,u] + Cv[c] >= P["C",u];
```

or

```
subject to delct {c in CREG, u in USERS}:
   0 >= P["C",u] - ctcost[c,u] - Cv[c] complements Ct[c,u] >= 0;
```

We also make no special distinction for inequalities that happen to be bounds on individual variables. As a result, the generalized complementarity forms (4) and (5) are specified as easily as their more restricted counterparts (1) and (2), without any of the transformations required by earlier designs.

Our representation does require the modeler or reader to remember the rules for deriving a complementarity condition from `complements` and its operands. In this respect, `complements` is a primitive operator, like `+` or `<=`, whose meaning must be furnished from a user's knowledge of the modeling language. The alternative would be to write out the complementarity requirement more explicitly, perhaps in forms (1) or (2) or their generalizations. As an example, the constraint `delc` introduced above might be declared equivalently in the following form motivated by (2):

```
var Cdual {c in CREG, t in CTYP}
   = ccost[c,t] + sum {res in R} cruse[res,c,t] * Mu[res];

subject to delc {c in CREG, t in CTYP}:
   C[c,t] = 0 and Cdual[c,t] >= Cv[c]  or
   C[c,t] = cmax[c,t] and Cdual[c,t] <= Cv[c]  or
   0 <= C[c,t] <= cmax[c,t] and Cdual[c,t] = 0;
```

This representation clearly states the entire complementarity condition using only existing AMPL operators. However, its adoption would require that AMPL be extended to allow variables in the operands to boolean operators (such as `and` and `or`). Such an extension would introduce a great variety of constraint types unrelated to complementarity, making complementarity constraints much harder for the modeler (and AMPL processor) to recognize. We could further modify the design to preserve recognizability, but only by introducing complex new rules on the use of variables with boolean operators. In light of this and similar examples, we have decided that the drawbacks of having a primitive `complements` operator are greatly outweighed by the advantages.

5. Extending presolve. Often it is worthwhile to simplify an optimization problem before sending it to a solver. Brearley, Mitra, and Williams [7] describe a set of simplification techniques based on iteratively tightening the bounds on variables and constraint expressions. These "presolve" techniques have been found to work well for linear programs and are provided as an option by many commercial linear programming solvers.

The AMPL modeling language processor also incorporates a primal presolve phase [20] that applies the ideas of [7] to linear constraints. (Nonlinearities are handled, but in a naive way. Because AMPL may send several objectives to the solver, we have not yet exploited the opportunities described in [7] to use dual information.) An integrated presolver is useful to a modeling language system in several respects. By identifying constraints involving only one variable, the presolver makes it irrelevant whether one states bounds on a variable in the variable's declaration or in a separate constraint declaration. Presolving sometimes results in a significantly smaller problem to convey to the solver, reducing communication time. Presolving on the modeling language side can also benefit any solver that does not have its own presolve phase.

Complementarity constraints introduce new information that we can exploit in AMPL's presolve routines. For instance, given a constraint of the form

$expr_1$ `>= 0  complements` $expr_2$ `>= 0`,

if we can deduce a positive lower bound on $expr_1$, then we can infer that it is strictly positive for all feasible points, and we can replace the constraint by

$expr_2$ `= 0`.

Similarly, for a constraint of the form

$const_1$ `<=` $expr_2$ `<=` $const_3$ `complements` $expr_4$,

there are various possibilities. If we can deduce that, say, $expr_2 < const_3$ for all feasible points, then we can replace the constraint by

$const_1$ `<=` $expr_2$ `complements` $expr_4$ `>= 0`.

If we can deduce that always $const_1 < expr_2 < const_3$, then we can replace the constraint by

$expr_4$ `= 0`.

Conversely, if we can deduce that $expr_4 < 0$, then we can replace the constraint by

$expr_2$ `=` $const_3$,

and so forth.

Each of these deductions can be triggered by presolve's manipulations of variable and constraint bounds. There are many combinations to be considered, but they are straightforward to enumerate and fast to check. As a simple illustration, consider the model

```
var x1;
var x2;
var x3;
subj to f1:   x1 >= 0  complements  x1 + 2*x2 + 3*x3 >= 1;
subj to f2:   x2 >= 0  complements  x2 - x3 >= -1;
subj to f3:   x3 >= 0  complements  x1 + x2 >= -1;
```

proposed by Munson [30] and called `munson1.mod` in MCPLIB [13]. The first inequalities in the complementarity constraints imply that all the variables are nonnegative. Then the second constraint in `f3` must always be slack, which implies that `x3` $= 0$, whence the second constraint in `f2` must always be slack, which implies that `x2` $= 0$. The second constraint in `f1` now reduces to `x1` ≥ 1, which implies that the first inequality in `f1` must always be slack, which implies `x1` $= 1$, and the presolver completely determines the solution. Our results in section 6 identify two larger test problems for which presolve's simplifications are significant.

Presolve folds together all bounds on a variable, whether specified in an ordinary `var` or `subj to` declaration or as an operand to `complements`. The user has the option of turning off most of presolve's logic, in which case separate bounds on some variables may remain separate. However, regardless of the presolve setting, AMPL detects when bounds given in a `var` declaration are redundant due to the same or tighter bounds being given as an argument to `complements` in a subsequent constraint. For example, to enhance the definition of variable `Ct` in Figure 1, we could add bounds,

```
var Ct {CREG, USERS} >= 0;
```

even though Figure 2 defines the same bounds in a subsequent complementarity constraint:

```
subj to delct {c in CREG, u in USERS}:
   0 <= Ct[c,u] complements ctcost[c,u] + Cv[c] >= P["C",u];
```

The redundant bounds do not make any difference to the form of the problem seen by the solver.

6. Communicating problems to solvers.

Since a modeling language is designed for the convenience of human modelers, the language processing software must do a certain amount of transformation to put problems into the forms required by efficient solvers. We describe in this section the transformations performed by AMPL's language processor to yield a canonical complementarity form useful for a variety of solvers. We then briefly comment on specific drivers for the PATH solver and for solvers written in MATLAB.

6.1. Transformation to canonical form.

To simplify the task of presenting complementarity problems to solvers, we have arranged for the AMPL processor to transform general complementarity constraints to the form

$$\ell_1 \leq \mathit{expr} \leq u_1 \quad \texttt{complements} \quad \ell_2 \leq \mathit{variable} \leq u_2, \tag{6}$$

where the complemented *variable*s are all distinct, and exactly two of the constants ℓ_1, u_1, ℓ_2, u_2 are finite. Ignoring the infinite bounds, this representation clearly describes a classical or mixed complementarity condition by the rules previously given.

This canonical form has the advantage of allowing the left operand and right operand of `complements` to be communicated to the solver as an ordinary constraint and an ordinary variable, respectively, as described in [24]. The complementarity extension can then be implemented by sending the solver only one new array, `cvar`, which pairs constraints with variables. Specifically, if the `i`th constraint seen by the solver has arisen by a complementarity relationship of the form (6) with the `j`th variable, then `cvar[i]` is set to `j`. Otherwise, the `i`th constraint has not arisen from any complementarity relation, and `cvar[i]` is set to an index that does not correspond to any variable.

The form of the transformation to (6) is straightforward, though it sometimes involves adding a new variable and an equality constraint defining the new variable. An expression complementing a general double-inequality constraint, for example, is transformed by

$$\begin{aligned} &\mathit{expr}_1 \ \texttt{complements} \ \ell \leq \mathit{expr}_2 \leq u \\ \Longrightarrow\ &-\infty \leq \mathit{expr}_1 \leq +\infty \ \texttt{complements} \ \ell \leq z \leq u, \ \ z = \mathit{expr}_2, \end{aligned}$$

where z is the new variable. In the common case of a bounded variable v complementing a single inequality, it is unnecessary to introduce a new variable and equality constraint, as long as v has not already been used as the canonical *variable* in another complementarity constraint. For example,

$$\begin{aligned} &v \geq 0 \ \texttt{complements} \ \mathit{expr} \geq 0 \\ \Longrightarrow\ &0 \leq \mathit{expr} \leq +\infty \ \texttt{complements} \ 0 \leq v \leq +\infty. \end{aligned}$$

However, if v is used in two such constraints, then it can serve as the canonical variable for the first, but a new variable w must be introduced as the canonical variable of the second:

$$\begin{aligned} &v \geq 0 \ \texttt{complements} \ \mathit{expr}_1 \geq 0, \ \ v \geq 0 \ \texttt{complements} \ \mathit{expr}_2 \geq 0 \\ \Longrightarrow\ &0 \leq \mathit{expr}_1 \leq +\infty \ \texttt{complements} \ 0 \leq v \leq +\infty, \\ &0 \leq \mathit{expr}_2 \leq +\infty \ \texttt{complements} \ 0 \leq w \leq +\infty, \ \ w = v. \end{aligned}$$

Other cases are similarly straightforward. All of AMPL's transformations to canonical form preserve the property of monotonicity described in section 3.1, ensuring that the complementarity problem sent to a solver will tend to be as well behaved as the problem originally formulated by the modeler.

6.2. Interface to PATH. Some current solvers, such as PATH [12, 16], want to see only complementarity conditions. If a problem is "square" in the sense that the number of variables equals the number of equality constraints plus the number of canonical complementarity conditions (6), then it is straightforward to create an equivalent pure complementarity problem in which all constraints have the form (6).

First, each finite bound on an unassociated variable (one not yet associated with a canonical complementarity constraint) is removed from the variable and added as a separate inequality constraint instead. Then each equality constraint can be converted to a complementarity condition by the transformation

$$\begin{aligned} & \mathit{expr} = \mathit{const} \\ \Longrightarrow\ & \mathit{const} \le \mathit{expr} \le \mathit{const} \ \ \texttt{complements} \ \ -\infty \le v \le +\infty, \end{aligned}$$

where v is any unassociated variable. The squareness of the problem ensures that every equality can be covered by a different variable in this way. Finally, a new variable is associated with each of the inequality constraints (including the aforementioned constraints created from variable bounds), after which the inequalities can also be converted to complementarity conditions. For example,

$$\begin{aligned} & \mathit{expr} \ge \mathit{const} \\ \Longrightarrow\ & \mathit{const} \le \mathit{expr} \le +\infty \ \ \texttt{complements} \ \ 0 \le z \le +\infty, \end{aligned}$$

where z is the new variable. The other inequality forms are handled similarly.

We have implemented an interface (or driver) that, when compiled with PATH, produces an AMPL solver `path` for square complementarity problems. It can be used in the AMPL command environment in the same way as other solvers:

```
ampl:  model pies.mod;
ampl:  data pies.dat;
ampl:  option solver path;
ampl:  solve;
PATH 3.0:  Solution found.
14 iterations (1 for crash); 28 pivots.
30 function, 16 gradient evaluations.
```

The driver reads a problem in the canonical form (6) and applies the manipulations described above to produce a problem consisting entirely of complementarity conditions, as the PATH solver requires. Instructions and C source for this driver are freely available from ftp://netlib.bell-labs.com/netlib/ampl/solvers/path.

Table 1 shows the results of running `path` on some AMPL problems from MCPLIB [13]. Certain problems are supplied with several starting guesses, as distinguished in the *start* column. Results are given both with ("yes") and without ("no") deduction of bounds by AMPL's presolver, in the two cases (*choi* and *pies*) where presolving makes a difference. The columns headed *nv*, *ncc*, and *nsc* give the numbers of variables, complementarity constraints (6), and side constraints seen by the solver (before it makes the previously described manipulations). The *nfunc* and *ngrad* columns report the numbers of function and gradient (Jacobian) evaluations.

6.3. Interface to MATLAB. Often it is convenient to use MATLAB [25, 31] implementations to experiment with algorithms. The examples associated with [24] include source for MATLAB *mex* functions that provide various information about op-

Table 1
Tests of the `path` *solver for AMPL.*

Problem	Start	Presolve	*nv*	*ncc*	*nsc*	*Iters*	*Pivots*	*nfunc*	*ngrad*
bertsek	1	—	15	10	5	5	13	25	6
bertsek	2	—	15	10	5	4	5	10	6
bertsek	3	—	15	10	5	6	12	14	8
bertsek	4	—	15	10	5	5	13	25	6
bertsek	5	—	15	10	5	4	3	10	6
bertsek	6	—	15	10	5	5	13	25	6
choi	1	no	14	13	2	4	7	10	6
choi	1	yes	13	13	0	4	3	10	6
ehl_def	1	—	101	100	1	5	6	12	7
ehl_kost	1	—	101	100	1	5	6	12	7
josephy	1	—	4	4	0	8	18	29	10
josephy	2	—	4	4	0	10	16	27	12
josephy	3	—	4	4	0	16	22	34	18
josephy	4	—	4	4	0	5	4	13	7
josephy	5	—	4	4	0	3	2	8	5
josephy	6	—	4	4	0	10	31	26	12
josephy	7	—	4	4	0	10	16	25	12
josephy	8	—	4	4	0	2	1	6	4
kojshin	1	—	4	4	0	10	21	26	12
kojshin	2	—	4	4	0	13	34	68	16
kojshin	3	—	4	4	0	16	36	34	18
kojshin	4	—	4	4	0	1	0	4	3
kojshin	5	—	4	4	0	5	6	12	7
kojshin	6	—	4	4	0	15	29	39	17
kojshin	7	—	4	4	0	10	27	25	12
kojshin	8	—	4	4	0	4	5	10	6
nash	1	—	10	10	0	6	5	14	8
nash	2	—	10	10	0	6	5	14	8
nash	3	—	10	10	0	5	4	12	7
nash	4	—	10	10	0	3	2	8	5
obstacle	1	—	2500	2500	0	7	1	14	9
pies	1	no	42	34	16	14	150	30	16
pies	1	yes	42	34	8	14	28	30	16

timization problems expressed in AMPL, such as dimensions, bounds, starting guesses, and function, gradient (or Jacobian matrix), and Lagrangian Hessian values. To encourage experiments with complementarity algorithms, we have extended these *mex* functions to also make available the `cvar` array of complementarity relations (as defined in section 6.1).

7. Related notation. The complementarity extensions to AMPL constraints necessitate corresponding extensions to notation for referring to constraints. This section briefly describes extensions to the "dot suffix" notation for constraint-related quantities, and to "synonyms" for constraint names.

7.1. Suffixes. As an aid to evaluating and understanding computed solutions, it is convenient to have notation for quantities such as lower and upper bounds, slack values (distances from bounds), and reduced costs associated with variables and constraints. The AMPL language admits various .*suffix* notations to denote these quantities. In particular, AMPL puts each constraint into the canonical form $\ell \leq$ *body* $\leq u$, in which ℓ and u are constants (possibly $-\infty$ and $+\infty$), with $\ell = u$ for equality constraints, after which the most frequently used .*suffix* options can be defined as shown in Table 2.

For dealing with complementarity constraints, we extend the .*suffix* notations

Table 2
Most frequently used suffixes for a constraint `Foo`*, in terms of the canonical form* $\ell \le body \le u$.

Notation	Meaning
`Foo.body`	$body$
`Foo.lb`	ℓ
`Foo.ub`	u
`Foo.lslack`	$body - \ell$
`Foo.uslack`	$u - body$
`Foo.slack`	$\min(\texttt{Foo.lslack}, \texttt{Foo.uslack})$

in several ways. A complementarity constraint `Goo` may be viewed as consisting of a "left" and "right" constraint, `Goo.L` and `Goo.R`, with a complementarity condition between them. To indicate quantities associated with `Goo`'s left and right constraints, we introduce the notations `Goo.L`*suf* and `Goo.R`*suf*, where *suf* is any suffix permitted for an ordinary constraint. For showing how close a complementarity condition is to holding, we also introduce the notation `Goo.slack`, whose meaning depends on `Goo`'s nature. If `Goo.L` and `Goo.R` involve one explicit inequality each, then

$$\texttt{Goo.slack} = \min(\texttt{Goo.Lslack}, \texttt{Goo.Rslack}).$$

Otherwise `Goo` has one of the forms

```
Goo.Lbody complements ℓ <= Goo.Rbody <= u,
ℓ <= Goo.Lbody <= u complements Goo.Rbody.
```

In the former case,

$$\texttt{Goo.slack} = \begin{cases} \min(\texttt{Goo.Lbody}, \texttt{Goo.Rbody} - \ell) & \text{if } \texttt{Goo.Rbody} \le \ell, \\ \min(-\texttt{Goo.Lbody}, u - \texttt{Goo.Rbody}) & \text{if } \texttt{Goo.Rbody} \ge u, \\ -|\texttt{Goo.Lbody}| & \text{otherwise;} \end{cases}$$

the latter case is defined analogously. Clearly `Goo.slack` is zerowhen the complementarity condition is satisfied. If `Goo.L` and `Goo.R` involve one inequality each, `Goo.slack` can be positive (if both constraints are strictly satisfied) or negative (if at least one is violated), so its sign conveys some information. In the other cases `Goo.slack` is always nonpositive.

7.2. Synonyms. Models are usually most conveniently described in terms of several kinds of differently named (and indexed) constraints, as seen in Figure 2. But sometimes it is helpful to address the variables and constraints with a uniform notation. For this purpose, AMPL offers generic *synonyms* for constraints (as well as variables and objectives). The synonym `_con[`i`]` denotes the ith constraint as the modeler sees constraints (before presolve) for $i = 1, \dots,$ `_ncons`, and `_scon[`i`]` denotes the ith constraint that the solver sees (after presolve) for $i = 1, \dots,$ `_sncons`. The notations `_conname[`i`]` and `_sconname[`i`]` denote the corresponding names of these constraints.

After considering several possibilities, we have found it most convenient to introduce separate synonyms for complementarity constraints, reserving `_con` and other existing synonyms for "ordinary" constraints (including each of the pair of constraints involved in a complementarity constraint declaration). The new synonyms are `_ccon[`i`]` for the ith complementarity constraint before presolve and `_cconname[`i`]` for its name, both for $i = 1, \dots,$ `_nccons`. We also define `_scvar[`i`]` as the index of the complementing variable associated with constraint i in the canonical form (6) sent to the solver. As an example of the use of these synonyms, one can see the extent to which the current solution satisfies the constraints of a complementarity problem by issuing the AMPL command

```
display max {i in 1.._nccons} abs(_ccon[i].slack),
        min {i in 1.._ncons} _con[i].slack;
```

to show the maximum complementarity violation and, over all constraints, the maximum constraint violation (negative values of `.slack` indicating violations).

8. Concluding remarks.

Modeling languages make it easy for people to go from a familiar mathematical formulation to the solution of a specific problem instance without worrying about computer programming details such as the data structures that solvers require. Thus modelers can focus on choosing the right model instead of worrying over lower-level aspects of implementation. Modeling languages have hitherto been used mainly for expressing conventional linear and nonlinear programs. The present work describes an extension to a wider class, including complementarity problems and mathematical programming problems with equilibrium constraints.

We hope that experience with and reaction to the present work will guide us in designing other useful extensions. One obvious possibility concerns expressing bilevel and multilevel optimization problems. These can be transformed to complementarity problems of the kind we have addressed, but only by means of a cumbersome conversion that requires one to write hand-coded derivative expressions in the complementarity constraints. It might be possible instead to introduce a simple extension that allows a constraint to reference the values of lower-level objectives.

An implementation of AMPL that includes the new complementarity extensions can be accessed through Web interfaces at either of the following sites:

http://www.ampl.com/ampl/TRYAMPL/
http://www.mcs.anl.gov/neos/Server/server-solvers.html

Although they differ in details, both of these interfaces accept AMPL models, data, and commands for execution on a remote computer, with PATH as one option for the choice of solver. Both then generate a Web page showing the results. Thus it is not necessary to have AMPL running locally to experiment with the new complementarity features.

Acknowledgments. We thank Brian Kernighan, Margaret Wright, and the anonymous referees for helpful comments on the manuscript.

REFERENCES

[1] J.F. Bard, *An algorithm for solving the general bilevel programming problem*, Math. Oper. Res., 8 (1983), pp. 260–272.

[2] J.F. Bard, *Optimality conditions for the bilevel programming problem*, Naval Res. Logist., 31 (1984), pp. 13–26.

[3] J.F. Bard, *Convex two-level optimization*, Math. Programming, 40 (1988), pp. 15–27.

[4] S.C. Billups, S.P. Dirkse, and M.C. Ferris, *A comparison of large scale mixed complementarity problem solvers*, Comput. Optim. Appl., 7 (1997), pp. 3–25.

[5] J.J. Bisschop and R. Entriken, *AIMMS: The Modeling System*, Paragon Decision Technology, Haarlem, the Netherlands, 1993.

[6] J. Bisschop and A. Meeraus, *On the development of a general algebraic modeling system in a strategic planning environment*, Math. Programming Stud., 20 (1982), pp. 1–29.

[7] A.L. Brearley, G. Mitra, and H.P. Williams, *Analysis of mathematical programming problems prior to applying the simplex method*, Math. Programming, 8 (1975), pp. 54–83.

[8] A. Brooke, D. Kendrick, and A. Meeraus, *GAMS: A User's Guide, Release* 2.25, Scientific Press/Duxbury Press, San Francisco, CA, 1992.

[9] J.W. Chinneck, *MProbe: Software for exploring nonlinear models*, Ann. Oper Res. special issue on modeling languages, to appear; also available online from http://www.sce.carleton.ca/faculty/chinneck/mprobe.html.

[10] R.W. Cottle, J.-S. Pang, and R.E. Stone, *The Linear Complementarity Problem*, Academic Press, New York, 1992.

[11] S.P. Dirkse, *Robust Solution of Mixed Complementarity Problems*, Mathematical Programming Technical Report 94-12, Computer Sciences Department, University of Wisconsin, Madison, 1994; also available online from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/94-12.ps.Z.

[12] S.P. Dirkse and M.C. Ferris, *The PATH solver: A non-monotone stabilization scheme for mixed complementarity problems*, Optim. Methods Softw., 5 (1995), pp. 123–156.

[13] S.P. Dirkse and M.C. Ferris, *MCPLIB: A collection of nonlinear mixed complementarity problems*, Optim. Methods Softw., 5 (1995), pp. 319–345; also available online from ftp://ftp.cs.wisc.edu/math-prog/mcplib/.

[14] S.P. Dirkse and M.C. Ferris, *Modeling and solution environments for MPEC: GAMS & MATLAB*, in Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, M. Fukishima and L. Qi, eds., Kluwer Academic Publishers, Norwell, MA, 1998; also available online from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/97-09.ps.Z.

[15] S.P. Dirkse, M.C. Ferris, P.V. Preckel, and T.F. Rutherford, *The GAMS Callable Program Library for Variational and Complementarity Solvers*, Mathematical Programming Technical Report 94-07, Computer Sciences Department, University of Wisconsin, Madison, 1994; also available online from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/94-07.ps.Z.

[16] M.C. Ferris and T.S. Munson, *Interfaces to PATH* 3.0: *Design, implementation and usage*, Comput. Optim. Appl., 12 (1999), pp. 207–227. Also available online from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/97-12.ps.Z

[17] M.C. Ferris and J.-S. Pang, *Complementarity and Variational Problems: State of the Art*, SIAM, Philadelphia, 1997.

[18] M.C. Ferris and J.-S. Pang, *Engineering and economic applications of complementarity problems*, SIAM Rev., 39 (1997), pp. 669–713.

[19] R. Fourer, *Extending a general-purpose algebraic modeling language to combinatorial optimization: A logic programming approach*, in Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search: Interfaces in Computer Science and Operations Research, D.L. Woodruff, ed., Kluwer Academic Publishers, Norwell, MA, 1998, pp. 31–74.

[20] R. Fourer and D.M. Gay, *Experience with a primal presolve algorithm*, in Large Scale Optimization: State of the Art, W.W. Hager, D.W. Hearn, and P.M. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA, 1994, pp. 135–154.

[21] R. Fourer and D.M. Gay, *Expressing special structures in an algebraic modeling language for mathematical programming*, ORSA J. Comput., 7 (1995), pp. 166–190.

[22] R. Fourer, D.M. Gay, and B.W. Kernighan, *A modeling language for mathematical programming*, Management Sci., 36 (1990), pp. 519–554.

[23] R. Fourer, D.M. Gay, and B.W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Scientific Press/Duxbury Press, San Francisco, CA, 1993.

[24] D.M. Gay, *Hooking Your Solver to AMPL*, Technical Report 97-4-06, Computing Sciences Research Center, Bell Laboratories, Lucent Technologies, 1997; also available online from http://www.ampl.com/ampl/REFS/hooking2.ps.gz.

[25] D.C. Hanselman and B.C. Littlefield, *Mastering MATLAB* 5: *A Comprehensive Tutorial and Reference*, Prentice-Hall, Upper Saddle River, NJ, 1997.

[26] P.T. Harker and J.-S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 161–220.

[27] M.M. Kostreva, *Elasto-hydrodynamic lubrication: A non-linear complementarity problem*, Internat. J. Numer. Methods Fluids, 4 (1984), pp. 377–397.

[28] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, Cambridge, UK, 1996.

[29] Z.-Q. Luo, J.-S. Pang, D. Ralph, and S.-Q. Wu, *Exact penalization and stationarity conditions of mathematical programs with equilibrium constraints*, Math. Programming, 75 (1996), pp. 19–76.

[30] T.S. Munson, *Private communication*, December 1997.

[31] D. Redfern and C. Campbell, *MATLAB Handbook*, Springer-Verlag, New York, 1998.

[32] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[33] T.F. Rutherford, *MILES: A Mixed Inequality and Nonlinear Equation Solver*, Working paper, Dept. of Economics, University of Colorado, 1993; also available online from http://robles.colorado.edu/~tomruth/milesdoc/milesdoc.htm.

[34] T.F. Rutherford, *Extensions of GAMS for complementarity problems arising in applied economic analysis*, J. Econom. Dynam. Control, 19 (1995), pp. 1299–1324.

[35] L. Schrage, *Optimization Modeling with LINGO*, LINDO Systems, Chicago, IL, 1998.

SIAM J. OPTIM.
Vol. 9, No. 4, pp. 1010–1040

ON THE COMPLEXITY OF SOLVING FEASIBLE LINEAR PROGRAMS SPECIFIED WITH APPROXIMATE DATA*

SHARON FILIPOWSKI[†]

This paper is dedicated with respect and admiration to John Dennis on the occasion of his 60th birthday

Abstract. The problem of solving linear programs specified with approximate data is considered. Algorithms are given for linear programs having both general inequality and nonnegativity constraints and for linear programs having only general inequality constraints.

Given approximate data for the actual (unknown) instance, the algorithms use knowledge that the instance in question is primal feasible to reduce the data precision necessary to give an approximation to the solution set of the actual instance when the actual instance has an optimal solution. In some cases, problem instances that would otherwise require perfect precision to solve can now be solved with approximate data because of the knowledge of primal feasibility.

The algorithms are computationally efficient. Furthermore, the algorithms require not much more data accuracy than the minimum amount necessary to give an approximate solution of a desired accuracy when the actual instance has an optimal solution. This work aids in the development of a computational complexity theory that uses approximate data and knowledge.

Key words. complexity of linear programming, approximate data, approximate solutions, condition measures, knowledge

AMS subject classifications. 90C05, 90C60

PII. S1052623494268467

1. Introduction.

In traditional complexity theory based on the Turing machine model of computation, all data are assumed to be rational and exact. Because of the existence of real numbers and because of experimental and round-off errors, these assumptions are not always appropriate. Furthermore, in traditional complexity theory, the efficiency of an algorithm in solving a particular problem instance is measured in terms of the bit length of the problem instance. Therefore, no attention is paid to the intrinsic difficulty of solving the particular problem instance (see Smale [6]).

Renegar [4, 5] developed a complexity theory that allows the problem data to consist of real numbers and approximate data while still maintaining finite precision computations. This new complexity theory uses the Turing machine as the underlying model of computation; however, the efficiency of an algorithm in solving a particular problem instance is measured not in terms of the bit length of the problem instance in question but in terms of a *condition measure*. This condition measure reflects the intrinsic difficulty of the problem instance to be solved and is similar to traditional condition numbers for systems of linear equations.

Renegar [5] and Vera [7, 8, 9] have obtained many interesting results within this new framework. Their results range from providing an algorithm that efficiently determines if a system of linear inequalities specified with approximate data is feasible or infeasible to providing an algorithm that efficiently solves convex quadratic programs specified with approximate data.

*Received by the editors May 27, 1994; accepted for publication (in revised form) August 3, 1998; published electronically September 24, 1999.

http://www.siam.org/journals/siopt/9-4/26846.html

[†]The Boeing Company, P.O. Box 3707, MS 7L-20, Seattle, WA 98124-2207 (sharon.k.filipowski@boeing.com).

Our work presented in [1, 2] took a step forward by allowing for the use of *knowledge* in this new complexity theory. An example of knowledge, and one that is considered in [1], is the knowledge that the actual system of linear inequalities is feasible before computations begin. Furthermore, as another example of knowledge, it is assumed in [2] that certain of the constraint matrix coefficients of the actual linear program are known to be equal to zero before computations begin. In this paper, we assume it is known that the actual linear program is primal feasible before computations begin. The use of knowledge is discussed in more detail in the remainder of this section as well as in section 4.

We now discuss what it means for an algorithm to *solve efficiently* a feasible linear program specified with approximate data. To do this, we first discuss *solving* linear programs of the following form:

$$\begin{aligned} &\max c^T x \\ &\qquad Ax \le b, \\ &\qquad\ x \ge 0, \end{aligned}$$

where $A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m,\ c \in \mathbb{R}^n$, and $x \in \mathbb{R}^n$, assuming it is *known* that the actual (unknown) instance is *primal feasible* before computations begin.

Let $d = (A, b, c) \in \mathbb{R}^{mn+m+n}$ denote the data vector of the actual instance. We assume that a rational approximation $\bar{d} = (\bar{A}, \bar{b}, \bar{c})$ to the actual instance is given, along with a rational error bound $\bar{\delta}$. Thus we have *approximate data* $(\bar{d}, \bar{\delta})$ satisfying $\|d - \bar{d}\| < \bar{\delta}$, where the norm used here (and throughout this paper) is the infinity norm for vectors. That is, $\|d\| \equiv \max_i(|d_i|)$.

Because of the knowledge of primal feasibility, the actual instance either has an optimal solution or is unbounded. First, assume that the actual instance has an optimal solution. Because only an approximation to the data of the actual instance is available, we can in general provide only an approximation to the solution set of the instance in question. Therefore, given approximate data $(\bar{d}, \bar{\delta})$, an algorithm is said to *solve* the actual instance if it provides an $\bar{x} \in \mathbb{R}^n$ and an $\bar{\epsilon} \in (0, \infty)$ such that

$$\bar{x} \in \{\tilde{x} : \|x^* - \tilde{x}\| < \bar{\epsilon} \text{ for some } x^* \text{ that solves } \max c^T x \text{ such that } Ax \le b,\ x \ge 0\}.$$

We define $\bar{x}$ to be an $\bar{\epsilon}$-*approximate solution* to the actual instance. Second, if the actual instance is unbounded, an algorithm is said to *solve* the actual instance if, given approximate data $(\bar{d}, \bar{\delta})$, it responds that the instance in question is unbounded.

Finally, given approximate data $(\bar{d}, \bar{\delta})$ and the knowledge of primal feasibility, we want an algorithm to return correctly with one of the following statements:

- The actual linear program is unbounded.
- The actual linear program has an optimal solution and $\bar{x} \in \mathbb{R}^n$ and $\bar{\epsilon} \in (0, \infty)$ are guaranteed to satisfy $\bar{x} \in \{\tilde{x} : \|x^* - \tilde{x}\| < \bar{\epsilon}$ for some x^* that solves $\max c^T x$ such that $Ax \le b,\ x \ge 0\}$.
- Better data accuracy is needed. (In this case, the algorithm is not able either to respond that the actual linear program is unbounded or to provide an $\bar{\epsilon}$-approximate solution, for any $\bar{\epsilon} \in (0, \infty)$, with the given approximate data $(\bar{d}, \bar{\delta})$.)

Because the actual instance is unknown, for an algorithm to be able to provide $\bar{x}$ as an $\bar{\epsilon}$-approximate solution, given approximate data $(\bar{d}, \bar{\delta})$ and the knowledge of primal feasibility, it must be able to guarantee that $\bar{x}$ will serve as an $\bar{\epsilon}$-approximate solution to all primal feasible instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| < \bar{\delta}$. Similarly, for an algorithm to be able to reply that the actual instance is unbounded, given approximate data $(\bar{d}, \bar{\delta})$ and the knowledge of primal feasibility, it must be able to guarantee that all primal feasible instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| < \bar{\delta}$ are unbounded.

Because of the knowledge of primal feasibility, an algorithm does not need to determine that the actual instance is primal feasible before solving the instance in question. Therefore, it might be the case that an algorithm does not need as much data precision to solve the actual instance as it does when there is no knowledge. Furthermore, in some cases, problem instances that would otherwise require perfect precision to solve without the knowledge of primal feasibility can now be solved without perfect precision because of the knowledge. The use of primal feasibility knowledge is discussed in more detail in section 4.

We now briefly discuss what it means for an algorithm to be efficient in solving a feasible linear program specified with approximate data. We refer the reader to [5] and to the introductions in [1, 8] for a more thorough discussion about this new complexity theory.

An algorithm is said to be *efficient* if it is both *computationally efficient* and *data efficient*. An algorithm is said to be computationally efficient if it runs in polynomial-time in the bit length of the approximate data $(\bar{d}, \bar{\delta})$. An algorithm is said to be data efficient if it uses *nearly minimal data precision*. We now briefly discuss data efficiency.

Assume that the actual instance d has an optimal solution. For a solution accuracy $\epsilon \in (0, \infty)$, there is a minimum perturbation size necessary such that there does not exist a point that serves as an ϵ-approximate solution for all primal feasible instances $\tilde{d}$ satisfying $\|d - \tilde{d}\| < \delta$ for any δ strictly larger than this minimum perturbation size. Furthermore, assuming that the actual instance is unbounded, there is a minimum perturbation size necessary such that not all primal feasible instances $\tilde{d}$ satisfying $\|d - \tilde{d}\| < \delta$ are unbounded for any δ strictly larger than this minimum perturbation size. Assuming the knowledge of primal feasibility, denoted by pf, for each instance d and solution accuracy ϵ, denote this minimum perturbation size by $\delta_{pf}(d, \epsilon)$.

In defining what it means for an algorithm to be data efficient, we use a *condition measure*, as first discussed by Renegar [5]. The condition measure for a pair d and ϵ, assuming the knowledge of primal feasibility, is

$$C_{pf}(d, \epsilon) \equiv \begin{cases} \frac{\|d\|}{\delta_{pf}(d,\epsilon)} & \text{if } \delta(d, \epsilon) > 0, \\ \infty & \text{otherwise.} \end{cases}$$

Roughly, $\log[\mathrm{C}_{pf}(d, \epsilon)]$ relative bits of accuracy are necessary to solve the actual instance so that this condition measure reflects the intrinsic difficulty of solving the problem instance in question.

Finally, an algorithm is said to be data efficient if there exist polynomials $p(m, n)$, $q(m, n)$, $r(m, n)$, and $t(m, n)$ in the variables m and n that are independent of the actual instance and desired solution accuracy such that the algorithm is guaranteed either to respond that the actual instance is unbounded or to provide a $q(m, n)\epsilon$-approximate solution when provided with approximate data that has error bound satisfying

$$\frac{\bar{\delta}}{\|d\|} \leq \left(\frac{1}{C_{pf}(d, \epsilon)}\right)^{r(m,n)} \left(\frac{1}{p(m, n)^{t(m,n)}}\right).$$

Therefore, the algorithm is guaranteed either to respond that the actual instance is unbounded or to provide a $q(m, n)\epsilon$-approximate solution when provided with only linearly more, in terms of $\log[C_{pf}(d,\epsilon)]$, bits of accuracy than the minimum amount necessary. The factor of $q(m,n)$ is added to make the definition norm independent. (It is assumed that all polynomials in the variables m and n in this paper are greater than or equal to 1 for all $m,n \geq 1$.)

Renegar originally defined data efficiency with $r(m,n)$ being restricted to a constant and with $t(m,n)$ being restricted to a value of 1. However, he mentioned in [5] that a slightly weaker definition of data efficiency might be needed in general.

We present two algorithms in this paper. In section 2 we present a computationally efficient algorithm that solves linear programs of the following form:

$$\begin{aligned} (P)\ \ \max c^T x& \\ Ax &\leq b, \\ x &\geq 0, \end{aligned}$$

where $A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m,\ c \in \mathbb{R}^n$, and $x \in \mathbb{R}^n$. Furthermore, assuming it is known that the actual instance is primal feasible before computations begin, we show that the algorithm is data efficient when the actual instance has an optimal solution. We give an example after the statement of Theorem 2.7 that shows that the algorithm is not data efficient when the actual instance is unbounded. When the actual instance is unbounded, the algorithm is the same as Renegar's algorithm [5], where there is no knowledge so that the knowledge of primal feasibility has not been used.

In section 3 we present a computationally efficient algorithm that solves linear programs of the following form:

$$\begin{aligned} (P)\ \ \max c^T x& \\ Ax &\leq b, \end{aligned}$$

where $A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m,\ c \in \mathbb{R}^n$, and $x \in \mathbb{R}^n$. Similar to the previous case, assuming it is known that the actual instance is primal feasible before computations begin, we show that the algorithm is data efficient when the actual instance has an optimal solution. Furthermore, we give an example after the statement of Theorem 3.10 that shows that the algorithm is not data efficient when the actual instance is unbounded. As before, in this case the algorithm is the same as Vera's algorithm [9], in which there is no knowledge, so the knowledge of primal feasibility has not been used.

In section 4, we discuss the use of the knowledge of primal feasibility. In particular, we give examples to show how use of the knowledge can lessen the data accuracy necessary to solve the problem instance in question. In addition, we give an example to show that in some cases, problem instances that would require perfect precision to solve without the knowledge of primal feasibility can now be solved without perfect precision because of the use of the knowledge. Furthermore, we give a brief comparison of our algorithm presented in section 2 with Renegar's algorithm [5], where knowledge is not considered.

Finally, in contrast to traditional complexity theory based on the Turing machine model of computation, transformations of the constraints of a linear program do not exist in this new theory such that one algorithm provides a fully efficient algorithm for all forms of a linear program. (For example, see Renegar [5].) This is the reason that two different algorithms are needed for the two different linear programs mentioned above. However, the algorithm and the analysis for linear programs with only general inequality constraints use the algorithm and analysis for linear programs with both general inequality and nonnegativity constraints. Therefore, an effort has been made to create an algorithm that can be used for all forms of a linear program, so that this new theory can share some of the beneficial properties of traditional complexity theory. This is discussed in more detail in section 3.

2. Linear programming: $\max\{\, c^T x : Ax \le b,\ x \ge 0\}$.

2.1. The algorithm. We first consider *solving* linear programs of the following form:

$$\begin{aligned}(P)\max c^T x&\\ Ax &\le b,\\ x &\ge 0,\end{aligned}$$

where $A \in \mathbb{R}^{m\times n},\ \ b \in \mathbb{R}^m,\ c \in \mathbb{R}^n$, and $x \in \mathbb{R}^n$, assuming it is *known* that the actual instance is *primal feasible* before computations begin. The dual of this linear program can be written

$$\begin{aligned}(D)\min b^T y&\\ A^T y &\ge c,\\ y &\ge 0,\end{aligned}$$

where $y \in \mathbb{R}^m$. The algorithm and a sketch of the algorithm will be given after some definitions are made and after some previous results that will be used by the algorithm are stated.

For a particular instance $d = (A, b, c)$, let $\text{Feas}(d)$ denote the feasible region for the instance d, let $\text{DualFeas}(d)$ denote the feasible region for the dual of the instance d, and let $\text{Opt}(d)$ denote the set of primal optimal solutions for the instance d. Also, if d has an optimal solution, let $k(d)$ denote the optimal objective function value for the instance d. That is,

$$\text{Feas}(d) \equiv \{x \ :\ Ax \le b,\ x \ge 0\},$$

$$\text{DualFeas}(d) \equiv \{y \ :\ A^T y \ge c,\ y \ge 0\},$$

$$\text{Opt}(d) \equiv \{x^* \ :\ x^* \in \text{Feas}(d) \text{ and } c^T x^* \ge c^T x \text{ for all } x \in \text{Feas}(d)\},$$

and

$$k(d) \equiv \max\{c^T x \ :\ Ax \le b,\ x \ge 0\}.$$

Let e denote the vector of all ones, with the dimension being clear from the context. Given approximate data $(\bar{d}, \bar{\delta})$, define

$$\bar{d}^+ \equiv (\bar{A} - \bar{\delta} e e^T,\ \bar{b} + \bar{\delta} e,\ \bar{c} + \bar{\delta} e),$$

$$\bar{d}^- \equiv (\bar{A} + \bar{\delta} e e^T,\ \bar{b} - \bar{\delta} e,\ \bar{c} - \bar{\delta} e).$$

Furthermore, let $e_i \in \mathbb{R}^n$ denote the ith unit vector in $\mathbb{R}^n$, for $i = 1, \ldots, n$. For an instance $\tilde{d}$ whose feasible region is bounded, let $\text{cen}_\infty(\text{Feas}(\tilde{d})) \in \mathbb{R}^n$ denote the *infinity center*, defined in terms of the infinity norm, of the feasible region for $\tilde{d}$. That is, for $i = 1, \ldots, n$,

$$(\text{cen}_\infty(\text{Feas}(\tilde{d})))_i \equiv \frac{1}{2}(v_i^+(\tilde{d}) + v_i^-(\tilde{d})),$$

where

$$v_i^+(\tilde{d}) \equiv \max\{\ e_i^T x\ :\ \tilde{A}x \le \tilde{b},\ x \ge 0\ \}$$

and

$$v_i^-(\tilde{d}) \equiv \min\{\ e_i^T x\ :\ \tilde{A}x \le \tilde{b},\ x \ge 0\ \}.$$

Also, if the feasible region for the instance $\tilde{d}$ is bounded, let $\text{rad}_\infty(\text{Feas}(\tilde{d})) \in \mathbb{R}$ denote the *infinity radius*, defined again in terms of the infinity norm, of the feasible region for the instance $\tilde{d}$. That is,

$$\text{rad}_\infty(\text{Feas}(\tilde{d})) \equiv \max_{1\le i\le n} \left\{ \frac{1}{2}(v_i^+(\tilde{d}) - v_i^-(\tilde{d})) \right\}.$$

Finally, if the feasible region for an instance $\tilde{d}$ is unbounded, $\text{cen}_\infty(\text{Feas}(\tilde{d}))$ is undefined, and we let $\text{rad}_\infty(\text{Feas}(\tilde{d})) \equiv \infty$.

The algorithm uses the following three lemmas, due to Renegar [5].

LEMMA 2.1 (see [5, equation (4.2)]). $\text{Feas}(\bar{d}^-) \subseteq \text{Feas}(\tilde{d}) \subseteq \text{Feas}(\bar{d}^+)$ *for all* $\tilde{d}$ *satisfying* $\|\tilde{d} - \bar{d}\| \le \bar{\delta}$.

Because of the symmetry of the primal and dual linear programs considered, a similar result holds for the feasible regions of the dual linear programs.

LEMMA 2.2 (see [5, equation (4.3)]). $\text{DualFeas}(\bar{d}^+) \subseteq \text{DualFeas}(\tilde{d}) \subseteq \text{DualFeas}(\bar{d}^-)$ *for all* $\tilde{d}$ *satisfying* $\|\tilde{d} - \bar{d}\| \le \bar{\delta}$.

The following lemma states that given approximate data $(\bar{d}, \bar{\delta})$, if $\bar{d}^-$ has an optimal solution, a portion of the feasible region of the instance $\bar{d}^+$ contains all optimal solutions for all instances $\tilde{d}$ satisfying $\|\bar{d} - \tilde{d}\| \le \bar{\delta}$.

LEMMA 2.3 (see [5, equation (4.4)]). *Given approximate data* $(\bar{d}, \bar{\delta})$, *assume that* $\bar{d}^-$ *has an optimal solution. Then* $\|\bar{d} - \tilde{d}\| \le \bar{\delta}$ *and* $\tilde{x} \in \text{Opt}(\tilde{d})$ *imply that* $\tilde{x} \in \text{Feas}(\bar{d}^+) \cap \{x : (\bar{c} + \bar{\delta}e)^T x \ge k(\bar{d}^-)\}$.

We now give a sketch of the algorithm. Given approximate data $(\bar{d}, \bar{\delta})$ and the knowledge of primal feasibility, the algorithm first checks if the actual instance is unbounded. Because of the knowledge of primal feasibility, it is enough to check if the actual instance is dual infeasible. Using Lemma 2.2, this checking can be done by deciding if $\bar{d}^-$ is dual infeasible (i.e., deciding if $(\bar{A} + \bar{\delta}ee^T)^T y \ge (\bar{c} - \bar{\delta}e),\ y \ge 0$, is infeasible). If $\bar{d}^-$ is dual infeasible, so that all instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \le \bar{\delta}$ are dual infeasible, the algorithm can reply that the actual instance is unbounded.

If the algorithm does not stop on the first step, the algorithm then checks if the actual instance has an optimal solution. Because of the knowledge of primal feasibility, it is enough to check if the actual instance is dual feasible. Again, using Lemma 2.2, this checking can be accomplished by deciding if $\bar{d}^+$ is dual feasible (i.e., deciding if $(\bar{A} - \bar{\delta}ee^T)^T y \ge (\bar{c} + \bar{\delta}e),\ y \ge 0$, is feasible). If $\bar{d}^+$ is dual feasible, the actual instance is guaranteed to have an optimal solution, and the algorithm can continue. However, if $\bar{d}^+$ is dual infeasible, so that not all instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \le \bar{\delta}$ are dual feasible, the algorithm must stop. In this case, the algorithm does not have enough accuracy to be able to determine that the actual instance has an optimal solution, and, hence, to provide an approximate solution of any accuracy. This will follow from Lemma 2.10.

If it has been determined that the actual instance has an optimal solution, the algorithm then tries to provide an $\bar{\epsilon}$-approximate solution to the actual instance for some $\bar{\epsilon} \in (0, \infty)$. It does this by first using Lemmas 2.5 and 2.6 to check if the origin is an optimal solution for the actual instance. If this is the case, the algorithm can stop with the origin as an $\bar{\epsilon}$-approximate solution for all $\bar{\epsilon} \in (0, \infty)$. This step is needed to make the algorithm data efficient when the actual instance has an optimal solution; this is discussed in more detail in section 2.2.

If it has not been determined that the origin is an optimal solution for the actual instance, it is then checked if all instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$ have an optimal solution. Because it has already been determined that all instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$ are dual feasible, all such instances will have an optimal solution if they all are primal feasible. Therefore, using Lemma 2.1, this checking can be done by deciding if $\bar{d}^-$ is primal feasible (i.e., deciding if $(\bar{A} + \bar{\delta}ee^T)x \leq (\bar{b} - \bar{\delta}e),\ x \geq 0$, is feasible). If $\bar{d}^-$ is primal feasible, so that all instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$ are primal feasible, the algorithm continues as in Renegar's algorithm [5]: the algorithm calculates the infinity radius of the portion of the feasible region for $\bar{d}^+$ that contains all optimal solutions for all instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$ (Lemma 2.3). If the radius of this region is finite, the infinity center is calculated and the algorithm stops and provides the infinity center as an $\bar{\epsilon}$-approximate solution, where $\bar{\epsilon}$ is any number larger than the calculated radius.

If all instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$ are not primal feasible, the algorithm checks if there exists an $\bar{\epsilon}$-approximate solution to all feasible points and, hence, to all optimal solutions, for all primal feasible instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$. Because the feasible region of $\bar{d}^+$ contains all such feasible points (Lemma 2.1), if the radius of the feasible region is finite, the infinity center is calculated and the algorithm stops and provides the infinity center as an $\bar{\epsilon}$-approximate solution, where $\bar{\epsilon}$ is any number larger than the calculated radius.

Finally, the algorithm might not have enough data accuracy either to determine that the actual instance is unbounded or to provide an $\bar{\epsilon}$-approximate solution, for any $\bar{\epsilon} \in (0, \infty)$, with the given approximate data $(\bar{d}, \bar{\delta})$.

The algorithm is given below.

ALGORITHM 2.4.

(0) *The algorithm assumes that* $(\bar{d}, \bar{\delta})$ *is given and that* d *is known to be primal feasible before computations begin.*

(1) *Check if* $\bar{d}^-$ *is dual infeasible. If so,* **STOP**; *the actual instance is unbounded.*

(2) *Check if* $\bar{d}^+$ *is dual feasible. If not, GOTO* (6).

(3) *Check if* $0 \in \mathrm{Opt}(\tilde{d})$ *for all* $\tilde{d}$ *satisfying both* $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$ *and* $\mathrm{Feas}(\tilde{d}) \neq \emptyset$, *using Lemmas* 2.5 *and* 2.6. *If so,* **STOP**; $\bar{x} = 0$ *serves as an* $\bar{\epsilon}$*-approximate solution for all* $\bar{\epsilon} \in (0, \infty)$.

(4) *Check if* $\bar{d}^-$ *is primal feasible. If so, check if* $\mathrm{rad}_{\infty}(\mathrm{Feas}(\bar{d}^+) \cap \{\, x : (\bar{c} + \bar{\delta}e)^T x \geq k(\bar{d}^-)\}) < \infty$. *If so,* **STOP**; $\bar{x} = \mathrm{cen}_\infty(\mathrm{Feas}(\bar{d}^+) \cap \{\, x : (\bar{c} + \bar{\delta}e)^T x \geq k(\bar{d}^-)\})$ *is an* $\bar{\epsilon}$*-approximate solution for all* $\bar{\epsilon} > \mathrm{rad}_\infty(\mathrm{Feas}(\bar{d}^+) \cap \{\, x : (\bar{c} + \bar{\delta}e)^T x \geq k(\bar{d}^-)\})$.

(5) *Check if* $\mathrm{rad}_\infty(\mathrm{Feas}(\bar{d}^+)) < \infty$. *If so,* **STOP**; $\bar{x} = \mathrm{cen}_\infty(\mathrm{Feas}(\bar{d}^+))$ *is an* $\bar{\epsilon}$*-approximate solution for all* $\bar{\epsilon} > \mathrm{rad}_\infty(\mathrm{Feas}(\bar{d}^+))$.

(6) *"Better data accuracy is needed."*

The following two lemmas present conditions that determine if the origin is an optimal solution for all primal feasible instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$. The first lemma is from [1].

LEMMA 2.5 (see [1, Lemma 2.3]). *Consider the following* m *linear programs in the variables* $x \in \mathbb{R}^n$ *and* $\Delta b \in \mathbb{R}^m$ *for* $i = 1, \ldots, m$:

$$\begin{aligned} &\min e_i^T(\bar{b} + \Delta b) \\ &\quad (\bar{A} - \bar{\delta}ee^T)x \leq \bar{b} + \Delta b, \end{aligned}$$

$$
\begin{aligned}
x &\geq 0,\\
\Delta b &\leq \bar{\delta} e,\\
-\Delta b &\leq \bar{\delta} e.
\end{aligned}
$$

Let $\Delta b_i^* \in \mathbb{R}^m$ solve the ith linear program for $i = 1, \dots, m$. Then $e_i^T(\bar{b} + \Delta b_i^*) \geq 0$ for all i if and only if $0 \in$ Feas$(\tilde{d})$ for all primal feasible instances $\tilde{d}$ satisfying $\|\bar{d} - \tilde{d}\| \leq \bar{\delta}$.

Proof. Assume that $e_i^T(\bar{b} + \Delta b_i^*) < 0$ for some i and let $\hat{d} = (\bar{A} - \bar{\delta} e e^T, \bar{b} + \Delta b_i^*, c)$. Then $\|\bar{d} - \hat{d}\| \leq \bar{\delta}$, Feas$(\hat{d}) \neq \emptyset$, and $0 \notin$ Feas$(\hat{d})$.

Next, assume that there exists an instance $\hat{d} = (\hat{A}, \hat{b}, \hat{c})$ that satisfies $\|\bar{d} - \hat{d}\| \leq \bar{\delta}$, Feas$(\hat{d}) \neq \emptyset$, and $\hat{b}_i < 0$ for some i (i.e., $0 \notin$ Feas$(\hat{d})$). Using Lemma 2.1, the instance $\tilde{d} = (\bar{A} - \bar{\delta} e e^T, \hat{b}, \hat{c})$ is primal feasible. Thus the ith linear program has optimal value $e_i^T(\bar{b} + \Delta b_i^*) \leq e_i^T \hat{b} < 0$. □

LEMMA 2.6. *Assume that the actual instance d is primal feasible, that $\|d - \bar{d}\| < \bar{\delta}$, and that the origin is a feasible point for all primal feasible instances $\tilde{d}$ satisfying $\|\bar{d} - \tilde{d}\| \leq \bar{\delta}$. Then $(\bar{c} + \bar{\delta} e) \leq 0$ if and only if the origin is an optimal solution for all primal feasible instances $\tilde{d}$ satisfying $\|\bar{d} - \tilde{d}\| \leq \bar{\delta}$.*

Proof. First assume that $(\bar{c} + \bar{\delta} e) \leq 0$. Therefore, for all primal feasible instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$, we have $\tilde{x} \in$ Feas$(\tilde{d})$, implying that $\tilde{c}^T \tilde{x} \leq 0$. Thus, the origin is an optimal solution for all primal feasible instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$.

Next assume that the origin is an optimal solution for all primal feasible instances $\tilde{d}$ satisfying $\|\bar{d} - \tilde{d}\| \leq \bar{\delta}$. Because of the knowledge of primal feasibility, the actual instance is primal feasible so that the origin is feasible for the actual instance d. Because the approximate data $(\bar{d}, \bar{\delta})$ satisfies $\|d - \bar{d}\| < \bar{\delta}$, we have $(\bar{A} - \bar{\delta} e e^T) < A$ and $\bar{b} + \bar{\delta} e > b$ so that the feasible region for the instance $(\bar{A} - \bar{\delta} e e^T,\ \bar{b} + \bar{\delta} e,\ \tilde{c})$ must be full-dimensional for any $\tilde{c}$ satisfying $\|\tilde{c} - \bar{c}\| \leq \bar{\delta}$. Therefore, because the origin is an optimal solution for any instance $\tilde{d} = (\bar{A} - \bar{\delta} e e^T,\ \bar{b} + \bar{\delta} e,\ \tilde{c})$, where $\tilde{c}$ satisfies $\|\tilde{c} - \bar{c}\| \leq \bar{\delta}$, we have $(\bar{c} + \bar{\delta} e) \leq 0$. □

2.2. Efficiency of the algorithm. The algorithm is computationally efficient because it relies just on linear programming (i.e., we assume that the algorithm uses a polynomial-time linear programming algorithm to solve all linear programs). The remainder of this section is devoted to showing that the algorithm is data efficient when the actual instance has an optimal solution. After the statement of Theorem 2.7, we give an example to show that the algorithm is not data efficient when the actual instance is unbounded.

Before proving that the algorithm is data efficient when the actual instance has an optimal solution, we state some definitions that will be used throughout the paper. For a particular *primal feasible* instance d, let $\delta_p^{'}(d)$ denote the distance between the instance d and the set of primal infeasible instances. That is,

$$\delta_p^{'}(d) \equiv \sup\{\delta \ :\ \|d - \tilde{d}\| < \delta \text{ implies that } \mathrm{Feas}(\tilde{d}) \neq \emptyset\}.$$

Similarly, if d is dual feasible, let $\delta_d^{'}(d)$ denote the distance between the instance d and the set of dual infeasible instances. Otherwise, if d is dual infeasible, let $\delta_d^{'}(d)$ denote the distance between the instance d and the set of dual feasible instances. That is,

$$\delta_d^{'}(d) \equiv \begin{cases} \sup\{\delta \ :\ \|d - \tilde{d}\| < \delta \text{ implies that } \mathrm{DualFeas}(\tilde{d}) \neq \emptyset\} & \text{if } d \text{ is dual feasible,}\\ \sup\{\delta \ :\ \|d - \tilde{d}\| < \delta \text{ implies that } \mathrm{DualFeas}(\tilde{d}) = \emptyset\} & \text{otherwise.} \end{cases}$$

Also, let

$$\delta^{'}(d) \equiv \min\{\delta_p^{'}(d), \delta_d^{'}(d)\}.$$

Furthermore, let $\delta^0_{pf}(d)$ denote the distance between the instance d and the set of primal feasible instances that do not have the origin as an optimal solution. If $0 \in \text{Opt}(d)$, then

$$\delta^0_{pf}(d) \equiv \sup\{\delta : \|d - \tilde{d}\| < \delta \text{ and } \text{Feas}(\tilde{d}) \neq \emptyset \text{ imply that } 0 \in \text{Opt}(\tilde{d})\}.$$

Furthermore, if $0 \notin \text{Opt}(d)$, then $\delta^0_{pf}(d) = 0$.

Using the knowledge that the actual instance is primal feasible, if the actual instance has an optimal solution, given a solution accuracy $\epsilon \in (0, \infty)$, the minimum perturbation size necessary such that there does not exist a point that serves as an ϵ-approximate solution for all primal feasible instances $\tilde{d}$ satisfying $\|d - \tilde{d}\| < \delta$ for any δ strictly larger than this minimum perturbation size is denoted by $\delta_{pf}(d, \epsilon)$ and can be written

$$\delta_{pf}(d, \epsilon) \equiv \sup\{\delta : \text{there exists } \bar{x} \in \mathbb{R}^n \text{ such that } \|d - \tilde{d}\| < \delta \text{ and } \text{Feas}(\tilde{d}) \neq \emptyset \text{ imply that there exists an } \tilde{x} \in \text{Opt}(\tilde{d}) \text{ satisfying } \|\bar{x} - \tilde{x}\| < \epsilon\}.$$

Also, if the actual instance is unbounded, the minimum perturbation size necessary such that not all primal feasible instances $\tilde{d}$ satisfying $\|\tilde{d} - d\| < \delta$ are unbounded for any δ strictly larger than this minimum perturbation size is denoted by $\delta_{pf}(d, \epsilon)$ and can be written

$$\delta_{pf}(d, \epsilon) \equiv \sup\{\delta : \|d - \tilde{d}\| < \delta \text{ and } \text{Feas}(\tilde{d}) \neq \emptyset \text{ imply that } \tilde{d} \text{ is unbounded}\}.$$

Notice that $\delta_{pf}(d, \epsilon)$ is independent of ϵ.

The data efficiency of the algorithm, when the actual instance has an optimal solution, follows from the following theorem. As mentioned in the introduction, when the actual instance is unbounded, we give an example to show that the algorithm is not data efficient. Furthermore, when the actual instance is unbounded, the algorithm is the same as Renegar's algorithm [5], in which there is no knowledge, so the knowledge of primal feasibility has not been used.

Theorem 2.7. *Assume it is known that the actual instance d is primal feasible before computations begin. There exist polynomials $p_1(m, n)$ and $t_1(m, n)$ in the variables m and n that are independent of the actual instance and desired solution accuracy such that, given $\epsilon \in (0, 1]$, Algorithm* 2.4 *is guaranteed to provide a $2m\sqrt{n}\epsilon$-approximate solution when the actual instance has an optimal solution and when provided with approximate data that has error bound satisfying*

$$\frac{\bar{\delta}}{\|d\|} \leq \left(\frac{\delta_{pf}(d, \epsilon)}{\|d\|}\right)^{25n} \left(\frac{1}{p_1(m, n)^{t_1(m,n)}}\right).$$

(Because the analysis in this paper relies on the analysis in [1], the desired solution accuracy ϵ is required to satisfy $\epsilon \in (0, 1]$. We comment on this again when we present Proposition 2.11.)

Before proving Theorem 2.7, we give an example to show why the algorithm with only steps (0), (1), (2), (4), (5), and (6) is not data efficient. After that we show that the algorithm is not data efficient when the actual instance is unbounded.

Consider the following instance $d = (A, b, c)$, where $A = (1, 1)$, $b = (0)$, and $c = (-1, -1)^T$. That is,

$$\begin{aligned} \max & -x_1 - x_2 \\ & x_1 + x_2 \leq 0, \\ & x_1, x_2 \geq 0. \end{aligned}$$

For this instance, $\|d\| = 1$, $\delta'_p(d) = 0$, $\delta^0_{pf}(d) = 1$, and $\delta_{pf}(d, \epsilon) = 1$ for all $\epsilon \in (0, 1]$. In particular, the origin is an optimal solution for all primal feasible instances $\tilde{d}$ that satisfy $\|d - \tilde{d}\| < 1$, and there exists an unbounded instance $\tilde{d}$ such that $\|d - \tilde{d}\| = 1$.

Let $p(m, n)$, $q(m, n)$, $t(m, n)$, and $r(m, n)$ be any polynomials in the variables m and n. We will show that for small enough ϵ, the algorithm is not able either to respond that the actual instance is unbounded on step (1) or to provide a $q(m, n)\epsilon$-approximate solution on either step (4) or step (5) when provided with approximate data that has error bound satisfying

$$\bar{\delta} \leq \left(\frac{\delta_{pf}(d, \epsilon)}{\|d\|} \right)^{r(m,n)} \left(\frac{1}{p(m, n)^{t(m,n)}} \right) = \frac{1}{p(m, n)^{t(m,n)}}.$$

First, because d is dual feasible, the algorithm cannot stop on step (1). Moreover, because $\delta'_p(d) = 0$, the algorithm cannot stop on step (4) without perfect precision. Furthermore, with the above approximate data error bound

$$\bar{\delta} \leq \frac{1}{p(m, n)^{t(m,n)}},$$

let $\bar{d}^+$ be an instance that the algorithm might consider in step (5). Then, assuming that $\bar{\delta} > 0$, $\text{rad}_\infty(\text{Feas}(\bar{d}^+)) > 0$. Thus for $\epsilon < \frac{1}{q(m,n)} \text{rad}_\infty(\text{Feas}(\bar{d}^+))$, the algorithm will not stop on step (5).

Therefore, for this instance and others as well, steps (0), (1), (2), (4), (5), and (6) of the algorithm are not sufficient to guarantee data efficiency. Theorem 2.7 states that the addition of step (3) is enough to guarantee data efficiency when the actual instance has an optimal solution.

Finally, we now show that the algorithm is not data efficient when the actual instance is unbounded. Consider the following linear program and its dual:

$$\begin{aligned} (P)\ \max\ & x \\ -\gamma x &\leq -1, \\ x &\geq 0, \end{aligned}$$

and

$$\begin{aligned} (D)\ \min\ & -y \\ -\gamma y &\geq 1, \\ y &\geq 0, \end{aligned}$$

where $0 < \gamma < 1$. We have that $\|d\| = 1$ and that $\delta'_p(d) = \delta'_d(d) = \gamma$. Furthermore, we have that $\delta_{pf}(d, \epsilon) = 1$ because any primal feasible instance $\tilde{d}$ satisfying $\|d - \tilde{d}\| < 1$ is dual infeasible and because there exists a $\tilde{d}$ such that $\text{Opt}(\tilde{d}) \neq \emptyset$ and $\|d - \tilde{d}\| = 1$. However, the algorithm will require the approximate data error bound to satisfy

$$\bar{\delta} \leq \frac{\delta'_d(d)}{2} = \frac{\gamma}{2}$$

to be guaranteed to stop with the answer that the actual instance is unbounded. (The extra factor of 1/2 is needed because, when given approximate data $(\bar{d}, \bar{\delta})$, the algorithm might be considering instances that are as far as $2\bar{\delta}$ away from the actual instance.) Because $\gamma = \delta_d^{'}(d)$ can be made arbitrarily small, the algorithm is not data efficient.

The proof of Theorem 2.7 follows from the following two propositions and one lemma. In these propositions and the lemma, we assume that $\|d\| = 1$. It is not until the proof of Theorem 2.7 that we consider the general case. The first proposition gives conditions on the actual instance d and the approximate data error bound such that for a given $\epsilon \in (0, 1]$, the radius of the feasible region of an instance $\bar{d}^+$ considered by the algorithm is guaranteed to be smaller than $m\sqrt{n}\epsilon$. In particular, it gives conditions for which, given some $\epsilon \in (0, 1]$, the algorithm is guaranteed to stop on step (5) with an $m\sqrt{n}\epsilon$-approximate solution.

PROPOSITION 2.8. *Assume it is known that the actual instance d is primal feasible before computations begin. Furthermore, assume that d has an optimal solution and that* $\|d\| = 1$. *There exist polynomials* $p_2(m, n)$ *and* $t_2(m, n)$ *in the variables m and n that are independent of the actual instance and desired solution accuracy such that if* $\epsilon \in (0, 1]$, $\delta_{pf}^0(d) \le \delta_{pf}(d, \epsilon)/2$, *and*

$$\max\{\delta_p^{'}(d), \|d - \tilde{d}\|\} \le \frac{(\delta_{pf}(d, \epsilon))^{6n}}{p_2(m, n)^{t_2(m,n)}},$$

then $\mathrm{rad}_\infty(\mathrm{Feas}(\tilde{d})) < m\sqrt{n}\epsilon$.

The second proposition is an adaptation of a proposition by Renegar (Proposition 4.2 in [5]). It gives conditions on the actual instance d and the approximate data error bound such that, given an $\epsilon \in (0, 1]$, the algorithm is guaranteed to provide a 2ϵ-approximate solution on step (4).

PROPOSITION 2.9. *Assume it is known that the actual instance d is primal feasible before computations begin. Furthermore, assume that d has an optimal solution and that* $\|d\| = 1$. *There exists a polynomial* $p_3(m, n)$ *in the variables m and n that is independent of the actual instance and desired solution accuracy such that if* $\epsilon \in (0, 1]$, $\delta_{pf}^0(d) \le \delta_{pf}(d, \epsilon)/2$, *and* $\delta_p^{'}(d) \ge (\delta_{pf}(d, \epsilon))^{6n}/p_2(m, n)^{t_2(m,n)}$, *where* $p_2(m, n)$ *and* $t_2(m, n)$ *are from Proposition* 2.8, *then Algorithm* 2.4 *is guaranteed to provide a* 2ϵ-*approximate solution on step* (4) *when provided with approximate data that has error bound satisfying*

$$\bar{\delta} \le \frac{(\delta_{pf}(d, \epsilon))^{7n}(\delta^{'}(d))^3}{p_3(m, n)^{t_2(m,n)}}.$$

Finally, the lemma shows that $\delta_{pf}(d, \epsilon) \le \delta_d^{'}(d)$ for all $\epsilon \in (0, \infty)$ if d is dual feasible. In particular, to be able to provide an ϵ-approximate solution of any accuracy when the actual instance has an optimal solution, the algorithm must have enough data accuracy such that only dual feasible instances are considered. It is not the case that $\delta_{pf}(d, \epsilon) \le \delta_d^{'}(d)$ for all $\epsilon \in (0, \infty)$ if the actual instance is dual infeasible, as shown in the second example after the statement of Theorem 2.7.

LEMMA 2.10. *Assume it is known that the actual instance d is primal feasible before computations begin. Furthermore, assume that d is dual feasible and that* $\epsilon \in (0, \infty)$. *Then*

$$\delta_{pf}(d, \epsilon) \le \delta_d^{'}(d).$$

We now prove Theorem 2.7 using Propositions 2.8 and 2.9 and Lemma 2.10.

Proof of Theorem 2.7. Consider the polynomials $p_1(m, n) = 4p_3(m, n)p_2(m, n)$ and $t_1(m, n) = 3t_2(m, n)$, where $p_2(m, n)$ and $t_2(m, n)$ are from Proposition 2.8 and where $p_3(m,n)$ is from Proposition 2.9, and assume that

$$\frac{\bar{\delta}}{\|d\|} \le \left(\frac{\delta_{pf}(d,\epsilon)}{\|d\|}\right)^{25n}\left(\frac{1}{p_1(m,n)^{t_1(m,n)}}\right). \tag{2.1}$$

It is assumed that the actual instance has an optimal solution. Notice that using Lemmas 2.2 and 2.10, the algorithm is guaranteed to determine that the actual instance is dual feasible on step (2) when provided with approximate data error bound $\bar{\delta}$ satisfying (2.1).

First, assume that $\delta_{pf}(d,\epsilon)/2 < \delta^0_{pf}(d)$. Using Lemmas 2.5 and 2.6, the algorithm is guaranteed to provide the origin as an $\bar{\epsilon}$-approximate solution on step (3) for all $\bar{\epsilon} \in (0,\infty)$ when provided with approximate data that has error bound satisfying

$$\bar{\delta} \le \frac{\delta^0_{pf}(d)}{2}.$$

(The extra factor of 1/2 is needed because, when given approximate data $(\bar{d},\bar{\delta})$, the algorithm might be considering instances that are as far as $2\bar{\delta}$ away from the actual instance.) Therefore, using the facts that $\delta_{pf}(d,\epsilon) \le \|d\|$ and that $\delta_{pf}(d,\epsilon)/2 < \delta^0_{pf}(d)$, the algorithm is guaranteed to stop with the approximate data error bound satisfying (2.1) because

$$\frac{\bar{\delta}}{\|d\|} \le \left(\frac{\delta_{pf}(d,\epsilon)}{\|d\|}\right)^{25n}\left(\frac{1}{p_1(m,n)^{t_1(m,n)}}\right) \le \frac{\delta_{pf}(d,\epsilon)}{4\|d\|} < \frac{\delta^0_{pf}(d)}{2\|d\|}.$$

Finally, assume that $\delta^0_{pf}(d) \le \delta_{pf}(d,\epsilon)/2$. Within this case, first assume that $\|d\| = 1$. We consider the more general case at the end of the proof. Furthermore, within this case, consider two subcases. First assume that

$$\frac{(\delta_{pf}(d,\epsilon))^{6n}}{p_2(m,n)^{t_2(m,n)}} < \delta'_p(d). \tag{2.2}$$

In this case, an algorithm does not necessarily need to determine that the actual instance is primal feasible before it can provide an ϵ-approximate solution; however, it cannot allow for much more inaccuracy in the approximate data than the distance to the set of primal infeasible instances. For this case, we now show that the algorithm is guaranteed to provide a 2ϵ-approximate solution on step (4) with the approximate data error bound satisfying (2.1).

Within this case, first assume that $\delta'(d) = \delta'_d(d) \le \delta'_p(d)$. Because $\delta_{pf}(d,\epsilon) \le 1$ (i.e., $\|d\| \le 1$) and $\delta_{pf}(d,\epsilon) \le \delta'_d(d)$ (Lemma 2.10),

$$\bar{\delta} \le \frac{(\delta_{pf}(d,\epsilon))^{25n}}{p_1(m,n)^{t_1(m,n)}} \le \frac{(\delta_{pf}(d,\epsilon))^{7n}(\delta_{pf}(d,\epsilon))^3}{p_3(m,n)^{t_2(m,n)}} \le \frac{(\delta_{pf}(d,\epsilon))^{7n}(\delta'(d))^3}{p_3(m,n)^{t_2(m,n)}}.$$

Therefore, because of (2.2) and because $\delta^0_{pf}(d) \le \delta_{pf}(d,\epsilon)/2$, the algorithm is guaranteed to provide a 2ϵ-approximate solution on step (4), using Proposition 2.9.

Otherwise, within this case, assume that $\delta'(d) = \delta'_p(d) < \delta'_d(d)$. Because of (2.2),

$$\bar{\delta} \le \frac{(\delta_{pf}(d,\epsilon))^{25n}}{p_1(m,n)^{t_1(m,n)}} = \left(\frac{(\delta_{pf}(d,\epsilon))^{7n}}{(4p_3(m,n))^{3t_2(m,n)}}\right)\left(\frac{(\delta_{pf}(d,\epsilon))^{6n}}{p_2(m,n)^{t_2(m,n)}}\right)^3 < \frac{(\delta_{pf}(d,\epsilon))^{7n}(\delta'(d))^3}{p_3(m,n)^{t_2(m,n)}}.$$

Again, because of (2.2) and the fact that $\delta^0_{pf}(d) \leq \delta_{pf}(d,\epsilon)/2$, the algorithm is guaranteed to provide a 2ϵ-approximate solution on step (4), using Proposition 2.9.

Otherwise, assume that

$$\delta_p^{'}(d) \leq \frac{(\delta_{pf}(d,\epsilon))^{6n}}{p_2(m,n)^{t_2(m,n)}}.$$

As in the first case, an algorithm also does not need to determine that the actual instance is primal feasible before being able to provide an ϵ-approximate solution. However, in this case, an algorithm might be able to provide an ϵ-approximate solution with a lot less accuracy in the approximate data error bound than the distance to the set of primal infeasible instances. In particular, an algorithm might be able to provide an ϵ-approximate solution without perfect precision even if the distance between the instance in question and the set of primal infeasible instances is zero. For this case, we show that the algorithm is guaranteed to provide an $m\sqrt{n}\epsilon$-approximate solution on step (5).

Because $\delta_{pf}(d,\epsilon) \leq 1$,

$$\bar{\delta} \leq \frac{(\delta_{pf}(d,\epsilon))^{25n}}{p_1(m,n)^{t_1(m,n)}} \leq \frac{(\delta_{pf}(d,\epsilon))^{6n}}{2p_2(m,n)^{t_2(m,n)}}.$$

Furthermore, because $\delta^0_{pf}(d) \leq \delta_{pf}(d,\epsilon)/2$ as well, the algorithm is guaranteed to provide an $m\sqrt{n}\epsilon$-approximate solution on step (5) using Proposition 2.8. (Again, the extra factor of 1/2 is needed because, given approximate data $(\bar{d},\bar{\delta})$, the algorithm might be considering instances as far as $2\bar{\delta}$ away from the actual instance.)

Now, assume that $\|d\| \neq 1$. We can assume that $\|d\| \neq 0$, because $\bar{\delta} > 0$ implies that $\delta_{pf}(d,\epsilon) > 0$. However, $\|d\| = 0$ implies that $d = 0$, so that with an arbitrarily small perturbation of d an unbounded instance exists so that $\delta_{pf}(d,\epsilon) = 0$ for all $\epsilon \in (0,1]$. Finally, consider the scaled instance

$$\hat{d} = \frac{d}{\|d\|}.$$

Because Opt(d) is invariant under positive scaling of the data for any instance d, we can assume that the algorithm is attempting to solve the instance $\hat{d}$. In particular, if an approximate data error bound $\bar{\delta}$ can be given such that the algorithm would be guaranteed to provide a $2m\sqrt{n}\epsilon$-approximate solution to the scaled instance $\hat{d}$, this approximate data error bound would be small enough to guarantee that the algorithm is going to stop with a $2m\sqrt{n}\epsilon$-approximate solution to the actual instance.

We have that

$$\delta_{pf}(\hat{d},\epsilon) = \frac{\delta_{pf}(d,\epsilon)}{\|d\|}$$

(i.e., write every instance $\tilde{d}$ around $\hat{d}$ as a scaled instance). Similarly, given the approximate data $(\bar{d},\bar{\delta})$, a box of size $\bar{\delta}$ around the actual approximate data instance $\bar{d}$ corresponds to a box of size $\bar{\delta}/\|d\|$ around the scaled instance $\bar{d}/\|d\|$. As a result, once

$$\frac{\bar{\delta}}{\|d\|} \leq \left(\frac{\delta_{pf}(d,\epsilon)}{\|d\|}\right)^{25n}\left(\frac{1}{p_1(m,n)^{t_1(m,n)}}\right) = \left(\frac{1}{C_{pf}(d,\epsilon)}\right)^{25n}\left(\frac{1}{p_1(m,n)^{t_1(m,n)}}\right),$$

the algorithm is guaranteed to provide a $2m\sqrt{n}$-approximate solution to the scaled instance $\hat{d}$ and, hence, to the actual instance, which proves the result of the theorem. □

In the remainder of this section, we prove Propositions 2.8 and 2.9 and Lemma 2.10. Proposition 2.8 can be proved with the following two propositions and one lemma. The first proposition considers just the feasible regions of the instances in question. For a desired solution accuracy $\epsilon \in (0,1]$, it gives conditions in terms of the actual instance, desired solution accuracy, and approximate data error bound size such that the algorithm is guaranteed to provide an $m\sqrt{n}\epsilon$-approximate solution on step (5) of the algorithm. This is Proposition 3.3 in [1] and is the part of the analysis where the requirement $\epsilon \in (0,1]$ comes from.

PROPOSITION 2.11 (see [1, Proposition 3.3]). *Let $d = (A, b)$ be the data of a feasible system of linear inequalities of the form $\{x : Ax \leq b,\ x \geq 0\}$. Assume that $\|d\| = 1$ and let*

$$\bar{\delta}_{pf}(d,\epsilon) \equiv \sup\{\delta : \textit{there exists } \bar{x} \in \mathbb{R}^n \textit{ such that } \|d - \tilde{d}\| < \delta \textit{ and } \mathrm{Feas}(\tilde{d}) \neq \emptyset \textit{ imply}$$
$$\textit{that there exists } \tilde{x} \in \mathrm{Feas}(\tilde{d}) \textit{ satisfying } \|\bar{x} - \tilde{x}\| < \epsilon\}.$$

There exist polynomials $p_4(m,n)$ and $t_2(m,n)$ in the variables m and n that are independent of the actual instance and desired solution accuracy such that if $\epsilon \in (0,1]$ and

$$\max\{\delta_p^{'}(d), \|d - \tilde{d}\|\} \leq \frac{(\bar{\delta}_{pf}(d,\epsilon))^{5n}\epsilon}{p_4(m,n)^{t_2(m,n)}},$$

then $\mathrm{rad}_\infty(\mathrm{Feas}(\tilde{d})) < m\sqrt{n}\epsilon$.

The next proposition adjusts the previous proposition so that providing an approximate solution of a desired accuracy to the optimal solution set of the actual instance instead of just the feasible region of the instance in question is considered so that it can be used to prove Proposition 2.8.

PROPOSITION 2.12. *Assume it is known that the actual instance $d = (A, b, c)$ is primal feasible before computations begin. Furthermore, assume that d has an optimal solution and that $\|d\| = 1$. There exist polynomials $p_4(m,n)$ and $t_2(m,n)$ in the variables m and n independent of the actual instance and desired solution accuracy such that if $\epsilon \in (0,1]$ and*

$$\max\{\delta_p^{'}(d), \|d - \tilde{d}\|\} \leq \frac{(\delta_{pf}(d,\epsilon))^{5n}\epsilon}{p_4(m,n)^{t_2(m,n)}},$$

then $\mathrm{rad}_\infty(\mathrm{Feas}(\tilde{d})) < m\sqrt{n}\epsilon$.

Proof. Let $p_4(m,n)$ and $t_2(m,n)$ be the polynomials in Proposition 2.11. If $\bar{x} \in \mathbb{R}^n$ is an ϵ-approximate solution to the optimal solution set of an instance d, then it is an ϵ-approximate solution to the feasible region of the instance d. Therefore, $\delta_{pf}(d,\epsilon) \leq \bar{\delta}_{pf}(d,\epsilon)$, so that

$$\max\{\delta_p^{'}(d), \|d - \tilde{d}\|\} \leq \frac{(\bar{\delta}_{pf}(d,\epsilon))^{5n}\epsilon}{p_4(m,n)^{t_2(m,n)}}$$

as well. Assuming that $\|(A, b)\| = 1$, the assumptions of Proposition 2.11 hold, and thus the result of this proposition holds as well.

Therefore, assume that $\|(A,b)\| < 1$. We will show that

$$\max\{\delta_p^{'}(d), \|d-\tilde{d}\|\} \leq \frac{(\delta_{pf}(d,\epsilon))^{5n}\epsilon}{p_4(m,n)^{t_2(m,n)}}$$

is sufficient as well.

Consider the scaled instance

$$\hat{d} \equiv \frac{d}{\|(A,b)\|} = (\hat{A}, \hat{b}, \hat{c}).$$

We can assume that $\|(A,b)\| \neq 0$; otherwise, $\delta_{pf}(d,\epsilon) = 0$ for all $\epsilon \in (0,1]$, so that $\bar{\delta} > 0$ cannot satisfy the assumption of the proposition.

Because the feasible region of a linear program is invariant under positive scaling of the data, if we can show that

$$\text{rad}_\infty\left(\text{Feas}\left(\frac{\tilde{d}}{\|(A,b)\|}\right)\right) < m\sqrt{n}\epsilon$$

for all considered $\tilde{d}$, the result of the proposition follows. Furthermore, because $\|(\hat{A},\hat{b})\| = 1$, Proposition 2.11 can be applied to $(\hat{A},\hat{b})$.

We have that

$$\delta_p^{'}(\hat{d}) = \frac{\delta_p^{'}(d)}{\|(A,b)\|}$$

and

$$\delta_{pf}(\hat{d},\epsilon) = \frac{\delta_{pf}(d,\epsilon)}{\|(A,b)\|}.$$

As a result, because

$$\delta_p{'}(d) \leq \frac{(\delta_{pf}(d,\epsilon))^{5n}\epsilon}{p_4(m,n)^{t_2(m,n)}},$$

we have that

$$\begin{aligned}\delta_p^{'}(\hat{d}) &= \frac{\delta_p^{'}(d)}{\|(A,b)\|}\\ &\leq \left(\frac{1}{\|(A,b)\|}\right)\frac{(\delta_{pf}(d,\epsilon))^{5n}\epsilon}{p_4(m,n)^{t_2(m,n)}}\\ &= \left(\frac{1}{\|(A,b)\|}\right)\frac{(\|(A,b)\|\delta_{pf}(\hat{d},\epsilon))^{5n}\epsilon}{p_4(m,n)^{t_2(m,n)}}.\end{aligned}$$

Because $\|(A,b)\| < 1$, we have that

$$\delta_p^{'}(\hat{d}) \leq \frac{(\delta_{pf}(\hat{d},\epsilon))^{5n}\epsilon}{p_4(m,n)^{t_2(m,n)}}.$$

Therefore, because $\|(\hat{A},\hat{b})\| = 1$, Proposition 2.11 can be used to come to the conclusion that

$$\|\hat{d}-\tilde{d}\| \leq \frac{(\delta_{pf}(\hat{d},\epsilon))^{5n}\epsilon}{p_4(m,n)^{t_2(m,n)}} = \left(\frac{\delta_{pf}(d,\epsilon)}{\|(A,b)\|}\right)^{5n}\frac{\epsilon}{p_4(m,n)^{t_2(m,n)}}$$

implies that $\mathrm{rad}_\infty(\mathrm{Feas}(\tilde{d})) < m\sqrt{n}\epsilon$. However, a box around d of size

$$\frac{(\delta_{pf}(d,\epsilon))^{5n}\epsilon}{p_4(m,n)^{t_2(m,n)}}$$

corresponds to a box around $\hat{d}$ of size

$$\frac{1}{\|(A,b)\|}\left(\frac{(\delta_{pf}(d,\epsilon))^{5n}\epsilon}{p_4(m,n)^{t_2(m,n)}}\right).$$

Because $\|(A,b)\| < 1$, the result of the proposition follows. □

To use Proposition 2.12 to prove Proposition 2.8, we need to be able to express the need for accuracy in the approximate data to be at least ϵ in terms of $\delta_{pf}(d,\epsilon)$. In particular, we need a polynomial $k(m,n)$ such that $\delta_{pf}(d,\epsilon)/k(m,n) \leq \epsilon$ for all instances considered in Proposition 2.8. The following lemma gives this polynomial.

LEMMA 2.13. *Assume it is known that the actual instance d is primal feasible before computations begin. Furthermore, assume that d has an optimal solution, that $\|d\| = 1$, and that $\delta^0_{pf}(d) \leq \delta_{pf}(d,\epsilon)/2$. Then*

$$\frac{\delta_{pf}(d,\epsilon)}{8n} \leq \epsilon$$

for all $\epsilon \in (0,\infty)$.

Proof. Assume that $\delta_{pf}(d,\epsilon) > 0$; otherwise, the lemma holds already. First assume that $0 \notin \mathrm{Opt}(d)$. Let $\bar{x} \in \mathrm{Opt}(d)$ be an extreme point. By assumption, $\bar{x} \neq 0$. Either $\{\bar{x}\} = \mathrm{Opt}(d)$ or, by an arbitrarily small perturbation of the objective function of d, there exists an instance whose solution set consists solely of the point $\bar{x}$.

Let $\bar{f} \in \mathbb{R}^n$ be defined by

$$\bar{f}_i \equiv \begin{cases} 1 & \text{if } \bar{x}_i > 0, \\ 0 & \text{otherwise,} \end{cases}$$

and consider the instance $\hat{d} = (A, b + 2\epsilon A\bar{f}, c)$. Thus $\|d - \hat{d}\| \leq 2n\epsilon$, and it can be shown that $\hat{x} = \bar{x} + 2\epsilon\bar{f}$ is an optimal extreme point for $\hat{d}$. Again, either $\{\hat{x}\} = \{\bar{x} + 2\epsilon\bar{f}\} = \mathrm{Opt}(\hat{d})$ or, by an arbitrarily small perturbation of the objective function, one can create an instance whose solution set consists solely of the point $\hat{x}$. Because $\|\bar{x} - (\bar{x} + 2\epsilon\bar{f})\| = 2\epsilon$, we have $\|d - \hat{d}\| \geq \delta_{pf}(d,\epsilon)$, so that $\delta_{pf}(d,\epsilon) \leq 2n\epsilon$.

Now assume that $0 \in \mathrm{Opt}(d)$. Either $\{0\} = \mathrm{Opt}(d)$ or, by an arbitrarily small perturbation of the objective function of d, there exists an instance whose solution set consists solely of the origin.

Let $\tilde{d}$ satisfy $\|d - \tilde{d}\| \leq \frac{3}{4}\delta_{pf}(d,\epsilon)$, $\mathrm{Opt}(\tilde{d}) \neq \emptyset$, and $0 \notin \mathrm{Opt}(\tilde{d})$. By the assumptions of the lemma, such an instance exists. Let $\bar{x} \in \mathrm{Opt}(\tilde{d})$ be an extreme point. Define $\bar{f}$ as above and consider the instance $\hat{d} = (\tilde{A}, \tilde{b} + 2\epsilon\tilde{A}\bar{f}, \tilde{c})$. Again, $\|\tilde{d} - \hat{d}\| \leq 2n\epsilon$, and $\hat{x} = \bar{x} + 2\epsilon\bar{f}$ is the optimal extreme point for $\hat{d}$ or an instance arbitrarily close to $\hat{d}$. Therefore,

$$\begin{aligned} \delta_{pf}(d,\epsilon) &\leq \|d - \hat{d}\| \\ &\leq \|d - \tilde{d}\| + \|\tilde{d} - \hat{d}\| \\ &\leq \frac{3}{4}\delta_{pf}(d,\epsilon) + 2n\epsilon \end{aligned}$$

so that $\delta_{pf}(d,\epsilon) \le 8n\epsilon$. □

We now use Proposition 2.12 and Lemma 2.13 to prove Proposition 2.8.

Proof of Proposition 2.8. Consider the polynomials $p_2(m,n) = 8np_4(m,n)$ and $t_2(m,n)$, where $p_4(m,n)$ and $t_2(m,n)$ are from Proposition 2.12. Assume that

$$\max\{\|d-\tilde{d}\|, \delta_p^{'}(d)\} \le \frac{(\delta_{pf}(d,\epsilon))^{6n}}{p_2(m,n)^{t_2(m,n)}} = \frac{(\delta_{pf}(d,\epsilon))^{6n}}{((8n)p_4(m,n))^{t_2(m,n)}}, \tag{2.3}$$

as in the assumption of Proposition 2.8. Using the result of Lemma 2.13, that $\delta_{pf}(d,\epsilon) \le 1$, and (2.3), we obtain that

$$\max\{\|d-\tilde{d}\|, \delta_p^{'}(d)\} \le \left(\frac{(\delta_{pf}(d,\epsilon))^{5n}}{p_4(m,n)^{t_2(m,n)}}\right)\left(\frac{\delta_{pf}(d,\epsilon)}{8n}\right) \le \frac{(\delta_{pf}(d,\epsilon))^{5n}\epsilon}{p_4(m,n)^{t_2(m,n)}}.$$

Thus, using the result of Proposition 2.12, Proposition 2.8 is proved. □

We now prove Proposition 2.9. The proof is similar to a proof by Renegar in [5] and relies on the following proposition (Proposition 2.14), also by Renegar. Also, if $\bar{d}^-$ has an optimal solution, let $\bar{\epsilon} = \mathrm{rad}_\infty(\mathrm{Feas}(\bar{d}^+) \cap \{x : (\bar{c} + \bar{\delta}e)^T x \ge k(\bar{d}^-)\})$, with $\bar{\epsilon}$ possibly being equal to ∞, just as in the algorithm.

Proposition 2.14 (see [5, Proposition 4.6]). *Assume it is known that the actual instance d is primal feasible before computations begin. Furthermore, assume that $\|d\| = 1$. There exist constants $K_5 \in \mathbb{R}_+$ and $K_6 \in \mathbb{R}_+$ that are independent of the problem instance and desired solution accuracy and such that if $\bar{\delta}$ and $\Delta\delta$ are positive numbers satisfying $\bar{\delta} + \Delta\delta \le K_5\delta^{'}(d)$, then for all $\epsilon \in (0,\infty)$,*

$$\epsilon < \bar{\epsilon} - K_6\left(\frac{\bar{\delta}}{\Delta\delta}\right)\left(\frac{1}{\delta^{'}(d)}\right)^3 \text{ implies that } 2\bar{\delta} + \Delta\delta \ge \delta_{pf}(d,\epsilon).$$

Proof of Proposition 2.9. Let $p_3(m,n) = 16nK_6p_2(m,n)/(\min\{1/2, K_5\})$, where $p_2(m,n)$ is from Proposition 2.8 and where K_5 and K_6 are from Proposition 2.14. Also assume that

$$\bar{\delta} \le \frac{(\delta_{pf}(d,\epsilon))^{7n}(\delta^{'}(d))^3}{p_3(m,n)^{t_2(m,n)}}, \tag{2.4}$$

where $t_2(m,n)$ is from Proposition 2.8.

First, (2.4) and $\bar{\delta} > 0$ imply that

$$\delta_{pf}(d,\epsilon) > 0. \tag{2.5}$$

Define

$$\Delta\delta \equiv \min\left\{\frac{1}{2}, K_5\right\}\frac{(\delta_{pf}(d,\epsilon))^{6n}}{p_2(m,n)^{t_2(m,n)}} - 2\bar{\delta}, \tag{2.6}$$

where K_5 is from Proposition 2.14. Note that (2.5), (2.6), and $\delta_{pf}(d,\epsilon) \le 1$ imply that

$$2\bar{\delta} + \Delta\delta < \frac{(\delta_{pf}(d,\epsilon))^{6n}}{p_2(m,n)^{t_2(m,n)}} \le \delta_{pf}(d,\epsilon). \tag{2.7}$$

Also because $\delta_{pf}(d,\epsilon) \le 1$, $\delta^{'}(d) \le 1$, and because of (2.4),

$$\bar{\delta} \le \frac{1}{4}\min\left\{\frac{1}{2}, K_5\right\}\frac{(\delta_{pf}(d,\epsilon))^{6n}}{p_2(m,n)^{t_2(m,n)}}$$

so that, using (2.6),

$$\Delta\delta \geq \frac{1}{2}\min\left\{\frac{1}{2}, K_5\right\}\frac{(\delta_{pf}(d,\epsilon))^{6n}}{p_2(m,n)^{t_2(m,n)}}. \tag{2.8}$$

Thus, because $\delta_{pf}(d,\epsilon) > 0$ and $K_5 \in \mathbb{R}_+$, we have that $\Delta\delta > 0$. Furthermore, using the fact that $\bar{\delta} > 0$, $\delta_{pf}(d,\epsilon) \leq \delta_d^{'}(d)$ (Lemma 2.10), (2.6), and

$$\frac{(\delta_{pf}(d,\epsilon))^{6n}}{p_2(m,n)^{t_2(m,n)}} \leq \delta_p^{'}(d),$$

we have

$$\bar{\delta} + \Delta\delta \leq K_5\frac{(\delta_{pf}(d,\epsilon))^{6n}}{p_2(m,n)^{t_2(m,n)}} \leq K_5\delta^{'}(d). \tag{2.9}$$

Therefore, we can use the results of Proposition 2.14 along with (2.7) to get that

$$\epsilon \geq \bar{\epsilon} - K_6\left(\frac{\bar{\delta}}{\Delta\delta}\right)\left(\frac{1}{\delta^{'}(d)}\right)^3. \tag{2.10}$$

Using (2.4), (2.8), the fact that $\delta_{pf}(d,\epsilon) \leq 1$, and the fact that $\delta_{pf}(d,\epsilon) \leq 8n\epsilon$ (Lemma 2.13), we have

$$\epsilon \geq \bar{\epsilon} - \epsilon.$$

Therefore, the results of the proposition hold. □

We finally prove Lemma 2.10.

Proof of Lemma 2.10. Assume that $\delta_{pf}(d,\epsilon) > 0$; otherwise, the lemma follows already. Furthermore, assume that $\delta_{pf}(d,\epsilon) > \delta_d^{'}(d)$. We will show that there exists an instance $\hat{d}$ that satisfies $\|d - \hat{d}\| < \delta_{pf}(d,\epsilon)$ and is unbounded, thus deriving a contradiction to the definition of $\delta_{pf}(d,\epsilon)$.

Because $\delta_{pf}(d,\epsilon) > \delta_d^{'}(d)$, there exists an instance $\tilde{d}$ that satisfies $\|d - \tilde{d}\| < \delta_{pf}(d,\epsilon)$ and is dual infeasible. Either $\tilde{d}$ is primal infeasible or it is unbounded. If it is unbounded, we have arrived at a contradiction (i.e., let $\hat{d} = \tilde{d}$). Thus assume that $\tilde{d}$ is primal infeasible. We now show that with an arbitrarily small perturbation of $\tilde{d}$ there exists an unbounded linear program, thus deriving a contradiction.

Because $\tilde{d} = (\tilde{A}, \tilde{b}, \tilde{c})$ is dual infeasible, it can be shown, using Farkas' lemma, that the following system is feasible:

$$\begin{aligned} \tilde{A}x &\leq 0, \\ x &\geq 0, \\ \tilde{c}^T x &> 0. \end{aligned}$$

Let x be any solution to the above system. Because $\tilde{d}$ is primal infeasible, x cannot satisfy $\tilde{A}x < 0$. However, because $x \neq 0$ (since $\tilde{c}^T x > 0$) with an arbitrarily small perturbation of the entries of $\tilde{A}$ (call the new matrix $\hat{A}$), the following system is feasible:

$$\begin{aligned} \hat{A}x &< 0, \\ x &\geq 0, \\ \tilde{c}^T x &> 0. \end{aligned}$$

Therefore, the instance $\hat{d} = (\hat{A}, \tilde{b}, \tilde{c})$ is unbounded. □

3. Linear programming: max{ $c^Tx : Ax \leq b$}.

3.1. The algorithm. We now consider solving linear programs of the following form:

$$\begin{aligned}(P)\max c^Tx&\\ Ax &\leq b,\end{aligned}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and $x \in \mathbb{R}^n$, assuming it is *known* that the actual instance is *primal feasible* before computations begin. The dual of this linear program can be written

$$\begin{aligned}(D)\min b^Ty&\\ A^Ty &= c,\\ y &\geq 0,\end{aligned}$$

where $y \in \mathbb{R}^m$.

As discussed in the introduction, we have made an effort to create an algorithm that can be used for all forms of a linear program. Because the combination of general inequality and nonnegativity constraints provides many useful properties in a linear program (Lemmas 2.1, 2.2, and 2.3) that other forms of the constraints in general do not provide, we have made an effort for the algorithm and analysis for linear programs with only general inequality constraints to use the algorithm and analysis for linear programs with both general inequality and nonnegativity constraints.

We now give a brief sketch of the algorithm. Given approximate data $(\bar{d}, \bar{\delta})$, the algorithm determines whether the actual instance is unbounded, the actual instance has an optimal solution, or better data accuracy is needed. If it has been determined that the actual instance has an optimal solution, the feasible regions and, hence, optimal solution sets of all primal feasible instances $\tilde{d}$ satisfying $\|\tilde{d}-\bar{d}\| \leq \bar{\delta}$ are *translated* toward the nonnegative orthant. Then, methods from Algorithm 2.4, where linear programs with both general inequality and nonnegativity constraints are considered, are used to try to provide an $\bar{\epsilon}$-approximate solution to all primal feasible translated instances, for some $\bar{\epsilon} \in (0, \infty)$. If such an $\bar{\epsilon}$-approximate solution is found, it is *translated back* to provide an $\bar{\epsilon}$-approximate solution to the actual instance. The details of the algorithm are discussed more thoroughly in the remainder of this section.

As before, for a particular instance d, let

$$\text{Feas}(d) \equiv \{x \ : \ Ax \leq b\},$$

let

$$\text{DualFeas}(d) \equiv \{y \ : \ A^Ty = c, \ y \geq 0\},$$

and let

$$\text{Opt}(d) \equiv \{x^* \ : \ x^* \in \text{Feas}(d) \text{ and } c^Tx^* \geq c^Tx \text{ for all } x \in \text{Feas}(d)\}.$$

Also, if d has an optimal solution, let

$$k(d) \equiv \max\{c^Tx : Ax \leq b\}.$$

In addition, for a particular *primal feasible* instance d, let $\delta_p^{'}(d)$ denote the distance between the instance d and the set of primal infeasible instances. If d is dual feasible, let $\delta'_d(d)$ denote the distance between the instance d and the set of dual infeasible instances, and if d is dual infeasible, let $\delta'_d(d)$ denote the distance between the instance d and the set of dual feasible instances. Finally, let $\delta'(d) \equiv \min\{\delta'_p(d),\ \delta'_d(d)\}$.

We now give a detailed sketch of the algorithm. As in Algorithm 2.4, it is first checked if the actual instance is unbounded. Because of the knowledge of primal feasibility, it is enough to check that the actual instance is dual infeasible. Therefore, it is first checked if all instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$ are dual infeasible. If all such instances are dual infeasible, it can be concluded that the actual instance is unbounded.

Because of the combination of equality and nonnegativity constraints present in the dual linear programs, the dual infeasibility of all instances $\tilde{d}$ satisfying $\|\tilde{d}-\bar{d}\| \leq \bar{\delta}$ can be checked by solving just one linear program, as shown by Vera [8]. We present his result in the following theorem.

THEOREM 3.1 (see [8, Proposition 3.12]). *There exists a computationally efficient algorithm that will ask for better data accuracy if the actual instance d is dual feasible or will correctly determine the dual infeasibility of d when provided with approximate data that has error bound satisfying*

$$\bar{\delta} \leq \frac{\delta'_d(d)}{2}.$$

If it has not been determined that the actual instance is unbounded, it is then checked if the actual instance has an optimal solution. Again, because of the knowledge of primal feasibility, it is enough to check if the actual instance is dual feasible. Therefore, it is then checked if all instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$ are dual feasible. If all such instances are not dual feasible, the algorithm must stop because it does not have enough data accuracy to provide an $\bar{\epsilon}$-approximate solution for any $\bar{\epsilon} \in (0, \infty)$, as will be shown in Lemma 3.12.

As discussed above, Vera has shown that the dual infeasibility of all instances considered given the approximate data $(\bar{d}, \bar{\delta})$ can be determined by checking the dual infeasibility of just one linear program. This result does not extend to the case of checking the dual feasibility of all instances considered, given the approximate data; however, Freund and Vera [3] have shown that there exists a computationally efficient algorithm such that the dual feasibility of all instances $\tilde{d}$ considered given the approximate data can be determined using minimal precision. We state this result in the following theorem.

THEOREM 3.2 (see [3, Remark 3.1]). *There exists a computationally efficient algorithm that will ask for better data accuracy if the actual instance d is dual infeasible or will correctly determine the dual feasibility of d when provided with approximate data that has error bound satisfying*

$$\bar{\delta} \leq \frac{\delta'_d(d)}{2}.$$

As mentioned earlier, if it has been determined that the actual instance has an optimal solution, the feasible regions and, hence, optimal solution sets of all primal feasible instances $\tilde{d}$ satisfying $\|\bar{d}-\tilde{d}\| \leq \bar{\delta}$ are *translated* toward the positive orthant. The algorithm then uses Algorithm 2.4, where the linear programs considered have both general inequality and nonnegativity constraints, to try to provide an $\bar{\epsilon}$-approximate solution to all the primal feasible translated instances for some $\bar{\epsilon} \in (0, \infty)$. If such an $\bar{\epsilon}$-approximate solution is found, it is *translated back* to provide an $\bar{\epsilon}$-approximate solution to the actual instance. We now explain the details of the translation and how Algorithm 2.4 is used.

For an instance $\tilde{d}$ and a translation size $\gamma \in \mathbb{R}_+$, let

$$\tilde{d}_\gamma \equiv (\tilde{A}, \tilde{b} + \gamma \tilde{A} e, \tilde{c})$$

be the associated translated instance. Using this definition, we have the following lemma that relates the optimal solution set of an original instance $\tilde{d}$ to the optimal solution set of the associated translated instance $\tilde{d}_\gamma$.

Lemma 3.3. *Let $\gamma \in \mathbb{R}$. Then $\tilde{x} \in \text{Opt}(\tilde{d})$ if and only if $\tilde{x} + \gamma e \in \text{Opt}(\tilde{d}_\gamma)$.*

Proof. We first show that $\tilde{x} \in \text{Feas}(\tilde{d})$ if and only if $\tilde{x} + \gamma e \in \text{Feas}(\tilde{d}_\gamma)$. Let $\tilde{x} \in \text{Feas}(\tilde{d})$. Then $\tilde{A}(\tilde{x} + \gamma e) \leq \tilde{b} + \gamma \tilde{A} e$ so that $\tilde{x} + \gamma e \in \text{Feas}(\tilde{d}_\gamma)$. Similarly, let $\tilde{z} \in \text{Feas}(\tilde{d}_\gamma)$. Then $\tilde{A}(\tilde{z} - \gamma e) \leq \tilde{b} + \gamma \tilde{A} e - \gamma \tilde{A} e = \tilde{b}$ so that $\tilde{z} - \gamma e \in \text{Feas}(\tilde{d})$.

Let $\tilde{x} \in \text{Opt}(\tilde{d})$ so that $\tilde{x} + \gamma e \in \text{Feas}(\tilde{d}_\gamma)$. Assume there exists a $\tilde{z} \in \text{Feas}(\tilde{d}_\gamma)$ satisfying $\tilde{c}^T \tilde{z} > \tilde{c}^T(\tilde{x} + \gamma e)$. We then have $\tilde{c}^T(\tilde{z} - \gamma e) > \tilde{c}^T \tilde{x}$, deriving a contradiction to the claim that $\tilde{x} \in \text{Opt}(\tilde{d})$. The other direction follows in a similar way. □

Before we state another lemma, recall the following definitions. If

$$\bar{x} \in \{\tilde{x} \; : \; \|\tilde{x} - x^*\| < \epsilon \text{ for some } x^* \in \text{Opt}(d)\},$$

then $\bar{x}$ is called an *ϵ-approximate solution* to the instance d. Similarly, if

$$\bar{z}_\gamma \in \{\tilde{z}_\gamma \; : \; \|\tilde{z}_\gamma - z^*_\gamma\| < \epsilon \text{ for some } z^*_\gamma \in \text{Opt}(d_\gamma)\},$$

then $\bar{z}_\gamma$ is called an *ϵ-approximate solution* to the instance d_γ.

Using these definitions, we have the following lemma that relates an ϵ-approximate solution to original instances to an ϵ-approximate solution to translated instances.

Lemma 3.4. *Given a vector $\bar{x}$ and a scalar γ, $\bar{x}$ is an ϵ-approximate solution to $\tilde{d}$ if and only if $\bar{z}_\gamma = \bar{x} + \gamma e$ is an ϵ-approximate solution to $\tilde{d}_\gamma$ for all data instances $\tilde{d} = (\tilde{A}, \tilde{b}, \tilde{c})$ satisfying both $\|d - \tilde{d}\| \leq \delta$ and* Feas*$(\tilde{d}) \neq \emptyset$, where $\tilde{d}_\gamma = (\tilde{A}, \tilde{b} + \gamma \tilde{A} e, \tilde{c})$.*

Proof. Assume that $\bar{x}$ is an ϵ-approximate solution to all $\tilde{d}$ satisfying both $\|d - \tilde{d}\| \leq \delta$ and $\text{Feas}(\tilde{d}) \neq \emptyset$, so that for all such instances $\tilde{d}$, there exists an $\tilde{x} \in \text{Opt}(\tilde{d})$ that satisfies $\|\bar{x} - \tilde{x}\| < \epsilon$. Using Lemma 3.3, $\tilde{z}_\gamma = \tilde{x} + \gamma e \in \text{Opt}(\tilde{d}_\gamma)$. Furthermore, $\|(\bar{x} + \gamma e) - (\tilde{x} + \gamma e)\| < \epsilon$, so that $\bar{z}_\gamma = \bar{x} + \gamma e$ is an ϵ-approximate solution to all $\tilde{d}_\gamma$, where $\tilde{d}$ satisfies both $\|d - \tilde{d}\| \leq \delta$ and $\text{Feas}(\tilde{d}) \neq \emptyset$.

The other direction follows in a similar way. □

We now discuss how Lemma 3.4 is used by the algorithm. Let $\Delta A_1 \in \mathbb{R}^{m \times n}$, $\Delta b_1 \in \mathbb{R}^m$, and $\Delta c_1 \in \mathbb{R}^n$. Also, for an instance $\tilde{d} = (\tilde{A}, \tilde{b}, \tilde{c})$, a translation size γ, and a perturbation size δ, let

$$S_1(\tilde{d}, \gamma, \delta) \equiv \{(\tilde{A} + \Delta A_1, \tilde{b} + \gamma \tilde{A} e + \gamma \Delta A_1 e + \Delta b_1, \tilde{c} + \Delta c_1) : \|(\Delta A_1, \Delta b_1, \Delta c_1)\| < \delta\}.$$

Given approximate data $(\bar{d}, \bar{\delta})$, assume that it has been determined that the actual instance has an optimal solution. Then, to try to provide an $\bar{\epsilon}$-approximate solution to the actual instance d for some $\bar{\epsilon} \in (0, \infty)$, the algorithm uses Algorithm 2.4 to try to provide an $\bar{\epsilon}$-approximate solution $\bar{z}_\gamma \in \mathbb{R}^n$ to all primal feasible translated instances $\tilde{d}_\gamma \in S_1(\bar{d}, \gamma, \bar{\delta})$, for some translation size $\gamma \in \mathbb{R}_+$. If the algorithm can do this, then, using Lemma 3.4, it can provide $\bar{x} = \bar{z}_\gamma - \gamma e$ as an $\bar{\epsilon}$-approximate solution to the actual instance.

To be able to use Algorithm 2.4 and $S_1(\bar{d}, \gamma, \bar{\delta})$, the translation size γ must be large enough such that either all instances $\tilde{d}_\gamma \in S_1(\bar{d}, \gamma, \bar{\delta})$ have an optimal solution and all optimal solutions are contained in the nonnegative orthant, or such that all feasible points for all primal feasible instances $\tilde{d}_\gamma \in S_1(\bar{d}, \gamma, \bar{\delta})$ are contained in the nonnegative orthant. If either case holds, nonnegativity constraints can be added to the translated instances without changing the optimal solution sets, and possibly even the feasible regions, so that Algorithm 2.4 can be used to try to provide an $\bar{\epsilon}$-approximate solution to all primal feasible instances $\tilde{d}_\gamma \in S_1(\bar{d}, \gamma, \bar{\delta})$ for some $\bar{\epsilon} \in (0, \infty)$. Checking if at least one of these cases holds can be done using extensions of Lemmas 2.1 and 2.3. To extend these lemmas, we need the following definitions. Given approximate data $(\bar{d}, \bar{\delta})$ and a translation size $\gamma \in \mathbb{R}_+$, let

$$\tilde{\delta} = \bar{\delta}(1 + n\gamma),$$

$$\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta}) \equiv (\bar{A} - \bar{\delta}ee^T,\ \bar{b} + \gamma\bar{A}e + \tilde{\delta}e,\ \bar{c} + \bar{\delta}e),$$

and

$$\bar{d}_\gamma^-(\bar{\delta}, \tilde{\delta}) \equiv (\bar{A} + \bar{\delta}ee^T,\ \bar{b} + \gamma\bar{A}e - \tilde{\delta}e,\ \bar{c} - \bar{\delta}e).$$

We have the following lemmas, the proofs of which are omitted because they are similar to the proofs of Lemmas 2.1 and 2.3.

LEMMA 3.5. $\mathrm{Feas}(\bar{d}_\gamma^-(\bar{\delta}, \tilde{\delta})) \cap \{z : z \geq 0\} \subseteq \mathrm{Feas}(\tilde{d}_\gamma) \cap \{z : z \geq 0\} \subseteq \mathrm{Feas}(\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})) \cap \{z : z \geq 0\}$ *for all* $\tilde{d}_\gamma \in S_1(\bar{d}, \gamma, \bar{\delta})$.

LEMMA 3.6. *Given approximate data* $(\bar{d}, \bar{\delta})$, *assume that* $Opt(\bar{d}_\gamma^-(\bar{\delta}, \tilde{\delta})) \cap \{z : z \geq 0\} \neq \emptyset$. *Then,* $\tilde{d}_\gamma \in S_1(\bar{d}, \gamma, \bar{\delta})$ *and* $\tilde{z}_\gamma \in \mathrm{Opt}(\tilde{d}_\gamma) \cap \{z : z \geq 0\}$ *imply that* $\tilde{z}_\gamma \in \mathrm{Feas}(\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})) \cap \{z : z \geq 0\} \cap \{z : (\bar{c} + \bar{\delta}e)^T z \geq k(\bar{d}_\gamma^-(\bar{\delta}, \tilde{\delta}))\}$.

To try to provide an $\bar{\epsilon}$-approximate solution to all primal feasible instances $\tilde{d}_\gamma \in S_1(\bar{d}, \gamma, \bar{\delta})$, the algorithm first checks if $\bar{d}_\gamma^-(\bar{\delta}, \tilde{\delta})$ has an optimal solution in the nonnegative orthant. If this is the case, using Lemma 3.5, all translated instances $\tilde{d}_\gamma \in S_1(\bar{d}, \gamma, \bar{\delta})$ are primal feasible. Furthermore, because the translation has not changed the dual feasible regions, all such translated instances have an optimal solution. Therefore, it is then checked, using Lemma 3.6, if the portion of the feasible region for $\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})$, which contains all optimal solutions in the nonnegative orthant for all primal feasible instances $\tilde{d}_\gamma \in S_1(\bar{d}, \gamma, \bar{\delta})$, is contained in the positive orthant. If this is the case, all optimal solutions for all instances $\tilde{d}_\gamma \in S_1(\bar{d}, \gamma, \bar{\delta})$ are contained in this region. Therefore, it is enough to check if this region is bounded. If it is, the algorithm can provide the infinity center of this region as an $\bar{\epsilon}$-approximate solution for any $\bar{\epsilon}$ strictly larger than the calculated radius.

If it has been determined that $\bar{d}_\gamma^-(\bar{\delta}, \tilde{\delta})$ does not have an optimal solution in the nonnegative orthant, it is checked if all feasible points for all primal feasible instances $\tilde{d}_\gamma \in S_1(\bar{d}, \gamma, \bar{\delta})$ are contained in the nonnegative orthant. This is done by deciding if the feasible region for $\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})$ is contained in the positive orthant. If this is the case, using Lemma 3.5, all feasible points for all primal feasible instances $\tilde{d}_\gamma \in S_1(\bar{d}, \gamma, \bar{\delta})$ are contained in the feasible region of $\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})$ and, hence, in the nonnegative orthant. Thus, it is then enough to check if the feasible region for $\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})$ is bounded. If this is true, the algorithm can provide its infinity center as an $\bar{\epsilon}$-approximate solution for any $\bar{\epsilon}$ strictly larger than the calculated radius.

Finally, the algorithm might not have enough data accuracy either to determine that the actual instance is unbounded or to provide an $\bar{\epsilon}$-approximate solution, for any $\bar{\epsilon} \in (0, \infty)$, with the given approximate data $(\bar{d}, \bar{\delta})$.

Before we state the algorithm, we make some remarks and state two lemmas that will be used to prove that the algorithm is data efficient when the actual instance has an optimal solution. The instances $\bar{d}^{+}_{\gamma}(\bar{\delta},\tilde{\delta})$ and $\bar{d}^{-}_{\gamma}(\bar{\delta},\tilde{\delta})$ are not contained in $S_1(\bar{d},\gamma,\bar{\delta})$, in contrast to $\bar{d}^{-}$ and $\bar{d}^{+}$ satisfying $\|\bar{d}-\bar{d}^{-}\| \leq \bar{\delta}$ and $\|\bar{d}-\bar{d}^{+}\| \leq \bar{\delta}$, when linear programs with both general inequality and nonnegativity constraints were considered. Therefore, the algorithm will not be trying to provide an $\bar{\epsilon}$-approximate solution to all primal feasible instances $\tilde{d}_{\gamma} \in S_1(\bar{d},\gamma,\bar{\delta})$ only, but instead to all primal feasible instances in the following slightly enlarged set $S_2(\bar{d},\gamma,\bar{\delta})$, where for an instance $\tilde{d}$, a translation size γ, and a perturbation size δ,

$$S_2(\tilde{d},\gamma,\delta) \equiv \{(\tilde{A}+\Delta A_2, \tilde{b}+\gamma\tilde{A}e+\Delta b_2, \tilde{c}+\Delta c_2) : \|(\Delta A_2,\Delta c_2)\| < \delta,\ \|\Delta b_2\| < \delta(1+n\gamma)\}.$$

We have the following two lemmas that will be used when proving that the algorithm is data efficient when the actual instance has an optimal solution. The first lemma shows how well $S_2(\tilde{d},\gamma,\delta)$ approximates $S_1(\tilde{d},\gamma,\delta)$ for an instance $\tilde{d}$, a translation size γ, and a perturbation size δ. Furthermore, assuming that the actual instance has an optimal solution, the second lemma gives a bound on the additional precision needed to solve the actual instance due to using the translation.

Lemma 3.7. *Let $\gamma,\delta \in \mathbb{R}_+$. Then*

$$S_1(\tilde{d},\gamma,\delta) \subseteq S_2(\tilde{d},\gamma,\delta).$$

Furthermore, assume that $\delta_2(1+2n\gamma) \leq \delta_1$ for $\delta_1,\delta_2 \in \mathbb{R}_+$. Then

$$S_2(\tilde{d},\gamma,\delta_2) \subseteq S_1(\tilde{d},\gamma,\delta_1).$$

Proof. Let $(\tilde{A}+\Delta A_1, \tilde{b}+\gamma\tilde{A}e+\gamma\Delta A_1 e+\Delta b_1, \tilde{c}+\Delta c_1) \in S_1(\tilde{d},\gamma,\delta)$ so that $\|(\Delta A_1,\Delta b_1,\Delta c_1)\| < \delta$. Because $\|\Delta b_1+\gamma\Delta A_1 e\| < \delta(1+n\gamma)$, the first claim follows.

Let $(\tilde{A}+\Delta A_2, \tilde{b}+\gamma\tilde{A}e+\Delta b_2, \tilde{c}+\Delta c_2) \in S_2(\tilde{d},\gamma,\delta_2)$ so that $\|(\Delta A_2,\Delta c_2)\| < \delta_2$ and $\|\Delta b_2\| < \delta_2(1+n\gamma)$. To show that $(\tilde{A}+\Delta A_2, \tilde{b}+\gamma\tilde{A}e+\Delta b_2, \tilde{c}+\Delta c_2) \in S_1(\tilde{d},\gamma,\delta_1)$ for $\delta_1 \geq \delta_2(1+2n\gamma)$, we need to show that there exists $(\Delta A_1,\Delta b_1,\Delta c_1)$ that satisfy both

$$\|(\Delta A_1,\Delta b_1,\Delta c_1)\| < \delta_1$$

and

$$\begin{aligned}\tilde{A}+\Delta A_1 &= \tilde{A}+\Delta A_2,\\ \tilde{b}+\gamma\tilde{A}e+\gamma\Delta A_1 e+\Delta b_1 &= \tilde{b}+\gamma\tilde{A}e+\Delta b_2,\\ \tilde{c}+\Delta c_1 &= \tilde{c}+\Delta c_2\end{aligned}$$

for some $\delta_1 \geq \delta_2(1+2n\gamma)$. Thus we have $\Delta A_1 = \Delta A_2$ and $\Delta c_1 = \Delta c_2$. Furthermore, we have $\Delta b_1 = \Delta b_2 - \gamma\Delta A_2 e$. Because $\|\Delta b_2 - \gamma\Delta A_2 e\| < \delta_2(1+2n\gamma)$, this is always possible with $\delta_1 \geq \delta_2(1+2n\gamma)$. □

Lemma 3.8. *Assume it is known that the actual instance d is primal feasible before computations begin. Furthermore, assume that $\gamma \in \mathbb{R}_+$ and that d has an optimal solution. Then $\delta_{pf}(d_\gamma,\epsilon) \geq \frac{\delta_{pf}(d,\epsilon)}{1+2n\gamma}$.*

Proof. If $\delta_{pf}(d,\epsilon) = 0$, then the result follows immediately. Therefore, let us assume that $\delta_{pf}(d,\epsilon) > 0$. Using Lemma 3.3, there exists an ϵ-approximate solution to all instances $\tilde{d}_\gamma \in S_1(d, \gamma, \delta_{pf}(d, \epsilon))$. Using Lemma 3.7, all $\tilde{d}_\gamma \in S_2(d, \gamma, \frac{\delta_{pf}(d,\epsilon)}{1+2n\gamma})$ have the same ϵ-approximate solution so that

$$\delta_{pf}(d_\gamma, \epsilon) \geq \frac{\delta_{pf}(d, \epsilon)}{1 + 2n\gamma}. \qquad \square$$

The algorithm is given below.

Algorithm 3.9.

(0) *The algorithm assumes that* $(\bar{d}, \bar{\delta})$ *is given and that it is known that the actual instance* d *is primal feasible before computations begin.*

(1) *Check if all instances* $\tilde{d}$ *satisfying* $\|\bar{d} - \tilde{d}\| \leq \bar{\delta}$ *are dual infeasible using Theorem* 3.1. *If so,* **STOP**; *the actual linear program is unbounded.*

(2) *Check if all instances* $\tilde{d}$ *satisfying* $\|\bar{d} - \tilde{d}\| \leq \bar{\delta}$ *are dual feasible using Theorem* 3.2. *If not, GOTO* (5).

(3) *Let* $\bar{d}_\gamma = (\bar{A}, \bar{b} + \gamma \bar{A} e, \bar{c})$, *where* $\gamma = 6\lceil \frac{1}{\bar{\delta}^{1/102n}} \rceil$ *and where* $\lceil a \rceil$ *is the smallest integer such that* $\lceil a \rceil \geq a$. *Check if* $\mathrm{Opt}(\bar{d}_\gamma^-(\bar{\delta}, \tilde{\delta})) \cap \{z : z \geq 0\} \neq \emptyset$. *If so, check if* $\mathrm{Feas}(\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})) \cap \{z : (\bar{c} + \bar{\delta} e)^T z \geq k(\bar{d}_\gamma^-(\bar{\delta}, \tilde{\delta}))\} \cap \{z : z > 0\} = \mathrm{Feas}(\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})) \cap \{z : (\bar{c} + \bar{\delta} e)^T z \geq k(\bar{d}_\gamma^-(\bar{\delta}, \tilde{\delta}))\}$. *If so, check if* $\mathrm{rad}_\infty(\mathrm{Feas}(\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})) \cap \{z : (\bar{c} + \bar{\delta} e)^T z \geq k(\bar{d}_\gamma^-(\bar{\delta}, \tilde{\delta}))\}) < \infty$. *If so,* **STOP**; *let* $\bar{z}_\gamma = \mathrm{cen}_\infty(\mathrm{Feas}(\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})) \cap \{z : (\bar{c} + \bar{\delta} e)^T z \geq k(\bar{d}_\gamma^-(\bar{\delta}, \tilde{\delta}))\})$. *Then* $\bar{x} = \bar{z}_\gamma - \gamma e$ *serves as an* $\bar{\epsilon}$*-approximate solution for all* $\bar{\epsilon} > \mathrm{rad}_\infty(\mathrm{Feas}(\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})) \cap \{z : (\bar{c} + \bar{\delta} e)^T z \geq k(\bar{d}_\gamma^-(\bar{\delta}, \tilde{\delta}))\})$.

(4) *Check if* $\mathrm{Feas}(\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})) \cap \{z : z > 0\} = \mathrm{Feas}(\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta}))$. *If so, check if* $\mathrm{rad}_\infty(\mathrm{Feas}(\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta}))) < \infty$. *If so,* **STOP**; *let* $\bar{z}_\gamma = \mathrm{cen}_\infty(\mathrm{Feas}(\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})))$. *Then* $\bar{x} = \bar{z}_\gamma - \gamma e$ *serves as an* $\bar{\epsilon}$*-approximate solution for all* $\bar{\epsilon} > \mathrm{rad}_\infty(\mathrm{Feas}(\bar{d}_\gamma^+(\bar{\delta}, \tilde{\delta})))$.

(5) *"Better data accuracy is needed."*

3.2. Efficiency of the algorithm. The algorithm is computationally efficient because it relies just on linear programming (i.e., we assume that all linear programs are solved using a polynomial-time linear program algorithm). The remainder of the section is devoted to showing that the algorithm is data efficient when the actual instance has an optimal solution. After the statement of Theorem 3.10, we give an example to show that the algorithm is not data efficient when the actual instance is unbounded.

Because of the knowledge of primal feasibility, if the actual instance has an optimal solution, the minimum perturbation size necessary such that there does not exist an ϵ-approximate solution to all primal feasible instances $\tilde{d}$ satisfying $\|d - \tilde{d}\| < \delta$ for any δ strictly larger than this minimal perturbation size is denoted by $\delta_{pf}(d, \epsilon)$ and can be written as follows:

$$\begin{aligned}\delta_{pf}(d, \epsilon) \equiv \sup\{\delta : {}& \text{there exists } \bar{x} \in \mathbb{R}^n \text{ such that } \|d - \tilde{d}\| < \delta \text{ and } \mathrm{Feas}(\tilde{d}) \neq \emptyset \\ & \text{imply that there exists } \tilde{x} \in \mathrm{Opt}(\tilde{d}) \text{ satisfying } \|\bar{x} - \tilde{x}\| < \epsilon\}.\end{aligned}$$

Also, assuming that the actual instance is unbounded, the minimum perturbation size necessary such that not all primal feasible instances $\tilde{d}$ satisfying $\|d - \tilde{d}\| < \delta$ are unbounded for any δ strictly larger than this minimal perturbation size is also denoted by $\delta_{pf}(d, \epsilon)$ and can be written as follows:

$$\delta_{pf}(d, \epsilon) \equiv \sup\{\delta : \|d - \tilde{d}\| < \delta \text{ and } \mathrm{Feas}(\tilde{d}) \neq \emptyset \text{ imply that } \tilde{d} \text{ is unbounded}\}.$$

Note that $\delta_{pf}(d,\epsilon)$ is independent of ϵ.

The data efficiency when the actual instance has an optimal solution follows from the following theorem. As mentioned in the introduction, we give an example after the statement of the theorem that shows that the algorithm is not data efficient when the actual instance is unbounded. Furthermore, if the actual instance is unbounded, the algorithm is the same as Vera's algorithm [9], so that the knowledge of primal feasibility is not being used.

Theorem 3.10. *Assume it is known that the actual instance d is primal feasible before computations begin. There exist polynomials $p_6(m,n)$ and $t_3(m,n)$ in the variables m and n independent of the actual instance and desired solution accuracy such that for $\epsilon \in (0,1]$, Algorithm* 3.9 *is guaranteed to provide a $2m\sqrt{n}\epsilon$-approximate solution when the actual instance has an optimal solution and when provided with approximate data that has error bound satisfying*

$$\frac{\bar{\delta}}{\|d\|} \le \left(\frac{\delta_{pf}(d,\epsilon)}{\|d\|}\right)^{102n} \left(\frac{1}{p_6(m,n)^{t_3(m,n)}}\right).$$

We now give an example to show that the algorithm is not data efficient when the actual instance is unbounded. It is similar to the example given when symmetric linear programs were considered. Consider the following linear program and its dual:

$$\begin{aligned} (P)\ \max\ & x \\ -\gamma x &\le -1, \\ -x &\le 0, \end{aligned}$$

and

$$\begin{aligned} (D)\ \min\ & -y_1 \\ -\gamma y_1 - y_2 &= 1, \\ y_1, y_2 &\ge 0, \end{aligned}$$

where $0 < \gamma < 1$. We have that $\|d\| = 1$ and that $\delta_p^{'}(d) = \delta_d^{'}(d) = \gamma$. Furthermore, we have that $\delta_{pf}(d,\epsilon) = 1$ for all $\epsilon \in (0,1]$ because any primal feasible instance $\tilde{d}$ satisfying $\|d - \tilde{d}\| < 1$ is dual infeasible and because there exists an instance $\hat{d}$ that has an optimal solution and such that $\|d - \hat{d}\| = 1$. However, the algorithm will require the approximate data error bound to satisfy

$$\bar{\delta} \le \frac{\delta_d^{'}(d)}{2} = \frac{\gamma}{2}$$

to be guaranteed to stop with the answer that the actual instance is unbounded. (The extra factor of 1/2 is needed because, when given approximate data $(\bar{d}, \bar{\delta})$, the algorithm might be considering instances that are as far as $2\bar{\delta}$ away from the actual instance.) Because $\gamma = \delta_d^{'}(d)$ can be made arbitrarily small, the algorithm is not data efficient.

The proof of Theorem 3.10 uses the following lemmas. As in the previous section, we assume that $\|d\| = 1$ when proving the lemmas. It is not until the proof of Theorem 3.10 that we consider the more general case.

Assuming that the actual instance d has an optimal solution and that $\delta_{pf}(d,\epsilon) > 0$ for a given $\epsilon \in (0,\infty)$, the following lemma will be used to bound the size of an optimal solution to d. This bound will be used to determine a sufficient translation size such that for given approximate data $(\bar{d}, \bar{\delta})$, all optimal solutions for all primal feasible translated instances $\tilde{d}_\gamma \in S_2(\bar{d}, \gamma, \bar{\delta})$ are contained in the positive orthant.

LEMMA 3.11. *Assume it is known that the actual instance d is primal feasible before computations begin. Furthermore, assume that d has an optimal solution, that $\epsilon \in (0, \infty)$, and that $\delta_{pf}(d, \epsilon) > 0$. There exists an $x^{'} \in \mathrm{Opt}(d)$ such that*

$$\|x^{'}\| \leq \frac{2\epsilon\|d\|}{\delta_{pf}(d, \epsilon)} - 2\epsilon.$$

Proof. From the definition of $\delta_{pf}(d, \epsilon)$, there exists an $\bar{x}$ such that $\|\tilde{d} - d\| < \delta_{pf}(d, \epsilon)$ and $\mathrm{Feas}(\tilde{d}) \neq \emptyset$ imply that there exists an $\tilde{x} \in \mathrm{Opt}(\tilde{d})$ satisfying $\|\tilde{x} - \bar{x}\| < \epsilon$.

Because $\delta_{pf}(d, \epsilon) > 0$, d must have a bounded optimal solution set. As a result, for an extreme point optimal solution $x^{'}$ of d, there exists an arbitrarily small perturbation of the objective function that creates an instance $d^{'} \equiv (A, b, c^{'})$ that has the unique optimal solution $x^{'}$. (If d already has a unique optimal solution, no perturbation is needed.) Clearly, $\|x^{'} - \bar{x}\| < \epsilon$.

We now show that $x^{'}$ satisfies

$$\|x^{'}\| \leq \frac{2\epsilon\|d\|}{\delta_{pf}(d, \epsilon)} - 2\epsilon.$$

Assume that $x^{'} \neq 0$. Otherwise, if $x^{'} = 0$, the bound is satisfied. Consider the instance $d^{''} = (A^{'}, b, c^{'})$ that is obtained by replacing A with

$$A^{'} \equiv \left(\frac{\|x^{'}\|}{\|x^{'}\| + 2\epsilon}\right) A.$$

It can be shown that

$$x^{''} \equiv \left(\frac{\|x^{'}\| + 2\epsilon}{\|x^{'}\|}\right) x^{'}$$

is the unique optimal solution for $d^{''}$. Because $\|x^{'} - x^{''}\| = 2\epsilon$ and $\|x^{'} - \bar{x}\| < \epsilon$, we have $\|x^{''} - \bar{x}\| \geq \epsilon$. Because $\|x^{''} - \bar{x}\| \geq \epsilon$, we have that

$$\begin{aligned} \delta_{pf}(d, \epsilon) &\leq \|d^{''} - d\| \\ &\leq \left(\frac{2\epsilon}{2\epsilon + \|x^{'}\|}\right) \|d\|, \end{aligned}$$

so that

$$\|x^{'}\| \leq \frac{2\epsilon\|d\|}{\delta_{pf}(d, \epsilon)} - 2\epsilon. \qquad \square$$

This next lemma is similar to Lemma 2.10. It states that in order for an algorithm to be able to provide an $\bar{\epsilon}$-approximate solution for any $\bar{\epsilon} \in (0, \infty)$, it must have enough accuracy such that all instances considered by the algorithm are dual feasible.

LEMMA 3.12. *Assume it is known that the actual instance d is primal feasible before computations begin. Furthermore, assume that d has an optimal solution and that $\epsilon \in (0, \infty)$. Then*

$$\delta_{pf}(d, \epsilon) \leq \delta_d^{'}(d).$$

Proof. Assume that $\delta_{pf}(d,\epsilon) > 0$; otherwise, the lemma holds already. The proof is similar to the proof of Lemma 2.10 because $\tilde{d}$ dual infeasible implies that the system

$$\tilde{A}x \leq 0, \qquad \tilde{c}^T x > 0$$

is feasible. □

We finally prove Theorem 3.10.

Proof of Theorem 3.10. Assume that

$$\frac{\bar{\delta}}{\|d\|} \leq \left(\frac{\delta_{pf}(d,\epsilon)}{\|d\|}\right)^{102n} \left(\frac{1}{p_6(m,n)^{t_3(m,n)}}\right), \tag{3.1}$$

where $p_6(m,n) = 96np_1(m,n)$, $t_3(m,n) = 102nt_1(m,n)$, and $p_1(m,n)$ and $t_1(m,n)$ are from Theorem 2.7.

It is assumed that the actual instance has an optimal solution. Using Theorem 3.2, the algorithm is guaranteed to determine that the actual instance is dual feasible when provided with approximate data error bound $\bar{\delta}$ satisfying

$$\bar{\delta} \leq \frac{\delta_d^{'}(d)}{2}.$$

Because $\delta_{pf}(d,\epsilon) \leq \delta_d^{'}(d)$ (Lemma 3.12), the algorithm is guaranteed to determine that the actual instance is dual feasible with approximate data error bound $\bar{\delta}$ satisfying (3.1).

Now, assume that $\|d\| = 1$. We consider the more general case at the end of the proof. Because $\bar{\delta} > 0$ we have that $\delta_{pf}(d,\epsilon) > 0$ for all $\epsilon \in (0,1]$. As a result, we can use Lemma 3.11 to get that $\|d - \tilde{d}\| < \frac{\delta_{pf}(d,\epsilon)}{2}$ and $\tilde{d}$ primal feasible imply that there exists an $\tilde{x} \in \mathrm{Opt}(\tilde{d})$ satisfying

$$\|\tilde{x}\| \leq \frac{8}{\delta_{pf}(d,\epsilon)}.$$

Furthermore, we have that $\|d - \tilde{d}\| < \delta_{pf}(d,\epsilon)$ and $\tilde{d}$ primal feasible imply that $\mathrm{rad}_\infty(\mathrm{Opt}(\tilde{d})) < \epsilon$. Otherwise, with an arbitrarily small perturbation of the objective function of $\tilde{d}$, instances can be created that have optimal solution sets that are farther than ϵ apart. Therefore, $\|d - \tilde{d}\| < \frac{\delta_{pf}(d,\epsilon)}{2}$ and $\tilde{d}$ primal feasible imply that if $\tilde{x} \in \mathrm{Opt}(\tilde{d})$, then

$$\|\tilde{x}\| \leq \frac{10}{\delta_{pf}(d,\epsilon)}.$$

Using this bound and the definition of $\delta_{pf}(d,\epsilon)$, there exists an ϵ-approximate solution $\bar{x}$ for all primal feasible $\tilde{d}$ satisfying $\|d - \tilde{d}\| < \frac{\delta_{pf}(d,\epsilon)}{2}$ that satisfies

$$\|\bar{x}\| \leq \frac{11}{\delta_{pf}(d,\epsilon)}.$$

Finally, because

$$\bar{\delta} \leq \left(\frac{\delta_{pf}(d,\epsilon)}{2}\right)^{102n},$$

we have that

$$\gamma = 6\left\lceil \frac{1}{\bar{\delta}^{1/102n}} \right\rceil \geq \frac{6}{\bar{\delta}^{1/102n}} \geq \frac{12}{\delta_{pf}(d,\epsilon)}.$$

As a result, Lemma 3.3 implies that $\tilde{d}_\gamma \in S_1(d, \gamma, \frac{\delta_{pf}(d,\epsilon)}{2})$ and $\tilde{d}_\gamma$ primal feasible imply that $\tilde{z}_\gamma \in \mathrm{Opt}(\tilde{d}_\gamma)$ satisfies

$$\tilde{z}_\gamma \geq \frac{2}{\delta_{pf}(d,\epsilon)} e \geq 2e.$$

Furthermore, Lemma 3.4 implies that there exists an ϵ-approximate solution $\bar{z}_\gamma$ for all primal feasible instances $\tilde{d}_\gamma \in S_1(d, \gamma, \frac{\delta_{pf}(d,\epsilon)}{2})$ satisfying

$$\bar{z}_\gamma \geq \frac{1}{\delta_{pf}(d,\epsilon)} e \geq e,$$

where $e \in \mathbb{R}^n$ is the vector of all ones.

We now show that Algorithm 3.9 is guaranteed to provide a $2m\sqrt{n}\epsilon$-approximate solution with the given approximate data error bound. Consider an instance $\tilde{d}_\gamma \in S_1(d, \gamma, \frac{\delta_{pf}(d,\epsilon)}{2})$ and the instance $\tilde{d}^*_\gamma$ whose objective function vector is the same as the objective function vector of $\tilde{d}_\gamma$ and whose feasible region consists of the intersection of the feasible region for $\tilde{d}_\gamma$ and the nonnegative orthant (i.e., $\mathrm{Feas}(\tilde{d}^*_\gamma) = \mathrm{Feas}(\tilde{d}_\gamma) \cap \{z : z \geq 0\}$). Because $\tilde{d}_\gamma \in S_1(d, \gamma, \frac{\delta_{pf}(d,\epsilon)}{2})$, $\tilde{d}_\gamma$ primal feasible, and $\tilde{z}_\gamma \in \mathrm{Opt}(\tilde{d}_\gamma)$ imply that $\tilde{z}_\gamma \geq 2e$, if $\tilde{d}_\gamma$ is primal feasible, $\tilde{d}^*_\gamma$ has the same optimal solution set as $\tilde{d}_\gamma$. Furthermore, because there exists an ϵ-approximate solution $\bar{z}_\gamma$ satisfying $\bar{z}_\gamma \geq e$ to all primal feasible instances $\tilde{d}_\gamma \in S_1(d, \gamma, \frac{\delta_{pf}(d,\epsilon)}{2})$, there exists an ϵ-approximate solution $\bar{z}^*_\gamma \geq e$ to all primal feasible instances $\tilde{d}^*_\gamma$, where $\tilde{d}_\gamma \in S_1(d, \gamma, \frac{\delta_{pf}(d,\epsilon)}{2})$. In particular, $\bar{z}^*_\gamma = \bar{z}_\gamma$. Therefore, using Lemma 3.8, we have that

$$\delta_{pf}(d^*_\gamma, \epsilon) \geq \frac{\delta_{pf}(d,\epsilon)}{2}\left(\frac{1}{1+2n\gamma}\right).$$

As a result, using Theorem 2.7, once

$$\text{(3.2)} \qquad \frac{\tilde{\delta}}{\|d_\gamma\|} = \frac{\bar{\delta}(1+n\gamma)}{\|d_\gamma\|} \leq \left(\frac{\delta_{pf}(d,\epsilon)}{2\|d_\gamma\|(1+2n\gamma)}\right)^{25n} \frac{1}{p_1(m,n)^{t_1(m,n)}},$$

where $p_1(m,n)$ and $t_1(m,n)$ are from Theorem 2.7, Algorithm 2.4 is guaranteed to provide a $2m\sqrt{n}\epsilon$-approximate solution $\bar{z}_\gamma$ to d^*_γ and hence to d_γ. As a result, using Lemma 3.4, $\bar{x} = \bar{z}_\gamma - \gamma e$ is guaranteed to be a $2m\sqrt{n}\epsilon$-approximate solution to d. Using some algebra and the fact that $\|d_\gamma\| \leq (1+n\gamma)\|d\|$, we have that (3.1) implies (3.2).

Finally, the proof that the theorem holds when $\|d\| \neq 1$ is similar to the proof of this case in the proof of Theorem 2.7. □

4. Remarks. We now make some remarks about the use of the knowledge of primal feasibility when solving linear programs with both general inequality and nonnegativity constraints. Similar remarks hold when solving linear programs with just general inequality constraints.

We first discuss how the algorithm uses the knowledge of primal feasibility. Because of the knowledge of primal feasibility, the algorithm does not need to determine that the actual instance is primal feasible before attempting to solve the instance in question. This use of the knowledge is shown by the presence of steps (1), (3), and (5) in Algorithm 2.4. In particular, in steps (1) and (3), the algorithm has not already determined that all instances considered by the algorithm are primal feasible before attempting to respond that the actual instance is unbounded or to provide an approximate solution of some accuracy. Furthermore, in step (5), the algorithm knows that not all instances considered are primal feasible before attempting to provide an approximate solution of some accuracy.

When using the knowledge of primal feasibility, an algorithm will be able to solve some problem instances with less precision than what an algorithm that does not have or use the knowledge of primal feasibility would be able to do. Also, in some cases, problem instances that would require perfect precision to solve without the knowledge of primal feasibility can now be solved without perfect precision with the use of the knowledge.

For example, consider the following linear program that was discussed in section 2.2:

$$\begin{aligned} \max\ & -x_1 - x_2 \\ & x_1 + x_2 \le 0, \\ & x_1, x_2 \ge 0. \end{aligned}$$

Because an arbitrarily small perturbation of the data can give an instance that is primal infeasible, an algorithm that does not have or use the knowledge of primal feasibility would need perfect precision just to be able to determine that the actual instance is primal feasible before attempting to provide an approximate solution. However, for this instance, Algorithm 2.4 is able to provide an approximate solution in step (3) without perfect precision.

For another example, consider the linear program:

$$\begin{array}{lrcrcr} \max & x_1 & + & x_2 & & \\ & x_1 & & & \le & 1, \\ & -x_1 & & & \le & -1, \\ & & & x_2 & \le & 1, \\ & & & -x_2 & \le & -1, \\ & x_1, & & x_2 & \ge & 0. \end{array}$$

Again, an arbitrarily small perturbation of the data can give an instance that is primal infeasible so that an algorithm that does not have or use the knowledge of primal feasibility will need perfect precision to be able to provide an approximate solution. However, Algorithm 2.4 is able to provide an approximate solution in step (5) without perfect precision.

Furthermore, consider a slight variation of the previous example,

$$\begin{array}{lrcrcr} \max & x_1 & + & x_2 & & \\ & x_1 & & & \le & 1, \\ & -x_1 & & & \le & -(1-\xi), \\ & & & x_2 & \le & 1, \\ & & & -x_2 & \le & -(1-\xi), \\ & x_1, & & x_2 & \ge & 0, \end{array}$$

where ξ satisfies $0 < \xi \ll 1$. Then, an algorithm that does not have or use the knowledge of primal feasibility will need the approximate data error bound $\bar{\delta}$ to satisfy at least

$$\bar{\delta} \leq \frac{\xi}{2}$$

to be guaranteed to be able to determine that the actual instance is primal feasible before attempting to provide an approximate solution. However, with the knowledge of primal feasibility, Algorithm 2.4 does not need much accuracy to be able to provide an approximate solution in step (5).

We now discuss the complexity results when the actual instance has an optimal solution. In particular, we discuss Theorem 2.7. This theorem states that Algorithm 2.4 is guaranteed to provide a $2m\sqrt{n}\epsilon$-approximate solution when provided with approximate data that has error bound $\bar{\delta}$ satisfying

$$\frac{\bar{\delta}}{\|d\|} \leq \left(\frac{\delta_{pf}(d,\epsilon)}{\|d\|}\right)^{25n} \left(\frac{1}{p_1(m,n)^{t_1(m,n)}}\right)$$

for some problem instance and solution accuracy-independent polynomials $p_1(m,n)$ and $t_1(m,n)$ in the variables m and n.

Renegar's [5] algorithm that does not use the knowledge of primal feasibility is guaranteed to provide an ϵ-approximate solution when provided with approximate data that has error bound satisfying

$$\frac{\bar{\delta}}{\|d\|} \leq \left(\frac{\delta(d,\epsilon)}{\|d\|}\right)^{6} \left(\frac{1}{p(m,n)}\right)$$

for some problem instance and solution accuracy-independent polynomial $p(m,n)$ in the variables m and n, where

$$\delta(d,\epsilon) \equiv \sup \Big\{ \delta : \text{there exists } \bar{x} \in \mathbb{R}^n \text{ such that } \|d - \tilde{d}\| < \delta \text{ implies that there exists } \tilde{x} \in \operatorname{Opt}(\tilde{d}) \text{ satisfying } \|\bar{x} - \tilde{x}\| \leq \epsilon \Big\}.$$

Because $25n \geq 6$ for all $n \geq 1$, it first appears that the algorithm that uses the knowledge of primal feasibility has a worse complexity result; however, this is not the case. First, if all instances $\tilde{d}$ satisfying $\|\tilde{d} - \bar{d}\| \leq \bar{\delta}$, given the approximate data $(\bar{d}, \bar{\delta})$, are primal feasible, Algorithm 2.4 reduces to Renegar's algorithm in steps (1) and (5). In this case, the knowledge of primal feasibility has not been helpful in reducing the precision necessary for an algorithm to solve the instance in question.

Furthermore, it might be the case, as in the above examples, that

$$\delta(d,\epsilon) \ll \delta_{pf}(d,\epsilon).$$

Therefore, it might be the case that

$$\left(\frac{\delta(d,\epsilon)}{\|d\|}\right)^{6} \left(\frac{1}{p(m,n)}\right) < \left(\frac{\delta_{pf}(d,\epsilon)}{\|d\|}\right)^{25n} \left(\frac{1}{p_1(m,n)^{t_1(m,n)}}\right),$$

so that Algorithm 2.4 is guaranteed to solve the actual with less precision. This is the case for the instances discussed above and will clearly be the case when $\delta_{pf}(d,\epsilon) > 0$ while $\delta(d,\epsilon) = 0$.

Acknowledgments. Consideration of primal feasibility knowledge began while I was a Ph.D. student at Cornell University in the School of Operations Research and Industrial Engineering under the direction of James Renegar, whom I thank for many helpful discussions about this work. Also, most of the work on this paper was completed while I was a faculty member in the Department of Industrial and Manufacturing Systems Engineering at Iowa State University. Finally, I thank two referees for carefully reading this paper and for their helpful comments.

REFERENCES

[1] S. Filipowski, *On the complexity of solving feasible systems of linear inequalities specified with approximate data*, Math. Programming, 71 (1995), pp. 259–288.

[2] S. Filipowski, *On the complexity of solving sparse symmetric linear programs specified with approximate data*, Math. Oper. Res., 22 (1997), pp. 769–792.

[3] R. Freund and J. Vera, *Some characterizations and properties of the 'distance to ill-posedness' and the condition measure of a conic linear system*, Math. Programming, to appear.

[4] J. Renegar, *Some perturbation theory for linear programming*, Math. Programming Ser. A, 65 (1994), pp. 73–91.

[5] J. Renegar, *Incorporating condition measures into the complexity theory of linear programming*, SIAM J. Optim., 5 (1995), pp. 506–524.

[6] S. Smale, *Some remarks on the foundations of numerical analysis*, SIAM Rev., 32 (1990), pp. 211–220.

[7] J. Vera, *Ill-posedness in Mathematical Programming and Problem Solving with Approximate Data*, Ph.D. dissertation, Cornell University, Ithaca, NY, 1992.

[8] J. Vera, *Ill-posedness and the complexity of deciding existence of solutions to linear programs*, SIAM J. Optim., 6 (1996), pp 549–569.

[9] J. Vera, *Ill-posedness and the Computation of Solutions to Linear Programs with Approximate Data*, preprint, Department of Industrial and Systems Engineering, Catholic University of Chile, Santiago, Chile, 1992. Available from the author at jvera@ing.puc.cl.

SIAM J. Optim.
Vol. 9, No. 4, pp. 1041–1063

ON MODIFIED FACTORIZATIONS FOR LARGE-SCALE LINEARLY CONSTRAINED OPTIMIZATION*

NICHOLAS IAN MARK GOULD†

This paper is dedicated to John Dennis, Jr., as a token of my appreciation for his unswerving support for computational mathematical programming

Abstract. We consider the algebraic issues concerning the solution of general, large-scale, linearly constrained nonlinear optimization problems. Particular attention is given to suitable methods for solving the linear systems that occur at each iteration of such methods. The main issue addressed is how to ensure that a quadratic model of the objective function is positive definite in the null-space of the constraints while neither adversely affecting the convergence of Newton's method nor incurring a significant computational overhead. Numerical evidence to support the theoretical developments is provided.

Key words. large-scale problems, linearly constrained optimization, modified matrix factorizations, second-order optimality conditions

AMS subject classifications. 65F05, 65F10, 65F15, 65F50, 65K05, 90C30

PII. S1052623495290660

1. Introduction.

Newton-like line-search methods for unconstrained and linearly constrained optimization may be broadly divided into two categories on the basis of how they deal with nonconvexity. Some methods wholeheartedly embrace nonconvexity by calculating directions of negative curvature as a means of escape from regions of nonconvexity. Others prefer to pretend that the nonconvexity is not present by replacing the Newton model by a convex modification. Although the former approach has theoretical advantages (see, for example, [35]), the latter is often preferred because of its simplicity.

The prototypical example of convex modification is the modified Cholesky method of [25] for unconstrained minimization. Here the second derivatives $\nabla^2_{xx} f$ of the objective function $f(\boldsymbol{x})$ are replaced by a modification $\boldsymbol{B} = \nabla^2_{xx} f + \boldsymbol{D}$ whenever $\nabla^2_{xx} f$ is insufficiently positive definite. The diagonal perturbation $\boldsymbol{D}$ is chosen so that $\boldsymbol{B}$ is sufficiently positive definite, that is,

$$\boldsymbol{p}^T \boldsymbol{B} \boldsymbol{p} \geq \sigma \boldsymbol{p}^T \boldsymbol{p} \text{ for some constant } \sigma > 0 \text{ and all } \boldsymbol{p}. \tag{1.1}$$

Most significantly, the perturbation is determined during an attempted Cholesky factorization of $\nabla^2_{xx} f$. If $\nabla^2_{xx} f$ is itself sufficiently positive definite, $\boldsymbol{D}$ is zero. The cost of finding $\boldsymbol{B}$ is barely more than the cost of a Cholesky factorization, and the norm of the resulting $\boldsymbol{B}$ has a guaranteed bound. More recently, [38] developed a similar method with a better a priori bound, while extensions to large-scale unconstrained and bound constrained optimization, using sparse factorizations, have been proposed by [26], [7, Chapter 3], and [37]. A thorough survey of modified Newton methods for unconstrained optimization is given by [9].

In this paper, we are interested in extending these ideas to linearly constrained optimization. We shall not concern ourselves with inequality constraints but presume

*Received by the editors August 23, 1995; accepted for publication (in revised form) November 17, 1997; published electronically September 24, 1999.
http://www.siam.org/journals/siopt/9-4/29066.html

†Atlas Centre, Rutherford Appleton Laboratory, Oxfordshire OX11 0QX, England (n.gould@rl.ac.uk).

that these are being handled by active set or barrier methods (see, for instance, [27, section 5.2] and [17]). Thus we aim to solve a smooth linearly constrained, nonlinear optimization problem,

$$\underset{x\in\Re^n}{\text{minimize}} \quad f(\boldsymbol{x}), \tag{1.2}$$

subject to m general, linearly independent, linear equations

$$\boldsymbol{Ax} = \boldsymbol{b}. \tag{1.3}$$

We consider a general iteration in which we have a point $\boldsymbol{x}$ satisfying (1.3) and wish to determine an improvement $\boldsymbol{x}+\boldsymbol{p}$. We build a second-order model of the objective function and pick $\boldsymbol{p}$ as the solution of the equality constrained quadratic program

$$\underset{\boldsymbol{p}\in\Re^n}{\text{minimize}} \quad \tfrac{1}{2}\boldsymbol{p}^T\boldsymbol{Bp}+\boldsymbol{p}^T\boldsymbol{g} \quad \text{subject to} \quad \boldsymbol{Ap}=\boldsymbol{0}. \tag{1.4}$$

Here $\boldsymbol{g} \overset{\text{def}}{=} \nabla_x f$ is the gradient of f and $\boldsymbol{B}$ is a suitable symmetric approximation to the Hessian $\boldsymbol{H} \overset{\text{def}}{=} \nabla^2_{xx} f$. We wish to guarantee that $\boldsymbol{p}$ is bounded and thus we require that the model problem is bounded from below. We ensure this by requiring that $\boldsymbol{B}$ be *second-order sufficient*, that is, that

$$\begin{array}{c}\boldsymbol{p}^T\boldsymbol{Bp} \geq \sigma \boldsymbol{p}^T\boldsymbol{p} \;\text{ for some constant }\; \sigma > 0\\ \text{and all }\; \boldsymbol{p} \;\text{ satisfying }\; \boldsymbol{Ap}=\boldsymbol{0}.\end{array} \tag{1.5}$$

This condition is the natural generalization of (1.1) for the constrained case. To ensure rapid asymptotic convergence, we also require that $\boldsymbol{B}$ be equal to $\boldsymbol{H}$ whenever the latter is itself second-order sufficient.

We shall be concerned with general, large-scale problems so we will not consider methods based solely on dense factorizations. We presume that $\boldsymbol{H}$ and $\boldsymbol{A}$ are sparse, and we consider sparse direct methods. So long as $\boldsymbol{B}$ is second-order sufficient, the solution to (1.4) satisfies the sparse linear system

$$\begin{pmatrix} \boldsymbol{B} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{g} \\ \boldsymbol{0} \end{pmatrix}, \tag{1.6}$$

where $\boldsymbol{\lambda}$ are Lagrange multipliers. Note that the coefficient matrix,

$$\boldsymbol{K} \overset{\text{def}}{=} \begin{pmatrix} \boldsymbol{B} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix}, \tag{1.7}$$

of (1.6) is inevitably indefinite—it must have at least m positive and m negative eigenvalues (see [28]). Thus any matrix factorization of (1.7) must be capable of handling indefinite matrices. To be efficient, one would normally try to exploit the symmetry of $\boldsymbol{K}$ in the factorization. The natural generalization of the Cholesky (or more precisely $\boldsymbol{LDL}^T$) factorization in the symmetric, indefinite case is that first proposed by [5] and later improved by [4] and [18] in the dense case and [16] and [14] in the sparse case. Here a symmetric matrix $\boldsymbol{K}$ is decomposed as

$$\boldsymbol{K} = \boldsymbol{PLDL}^T\boldsymbol{P}^T, \tag{1.8}$$

where $\boldsymbol{P}$ is a permutation matrix, $\boldsymbol{L}$ is unit lower triangular, and $\boldsymbol{D}$ is block diagonal with blocks of size at most two. Each diagonal block corresponds to a pivoting operation. We shall refer to the blocks as 1 by 1 and 2 by 2 pivots.

Because we are particularly concerned with the large-scale case, it is the Duff–Reid variant that is of special interest. We note that the permutation matrices are used extensively in the factorization of sparse matrices to keep the fill-in—that is, the introduction of extra nonzeros in the factors—at an acceptable level. Unfortunately, the [30] implementation, MA27, [13] of the Duff–Reid variant sometimes proves inefficient when applied to matrices of the form (1.7) as the analysis phase treats the whole diagonal of $\boldsymbol{K}$ as if it contains nonzero entries. Thus a good predicted ordering supplied by the analyze phase is often replaced, for stability reasons, by a less satisfactory ordering when the factorization is performed, resulting in considerable extra work and fill-in. Ways of avoiding these difficulties, and of taking further advantage of the zero block in $\boldsymbol{K}$, have been suggested by [12] and form the basis for a recent Harwell Subroutine Library code MA47 [15].

In the special case when f is separable, $\boldsymbol{H}$ will be diagonal. In particular, when f is also strictly convex, $\boldsymbol{H}$ will be positive definite and a block elimination of $\boldsymbol{H}$ followed by a sparse Cholesky factorization of the (negative of the) Schur complement $\boldsymbol{A}\boldsymbol{H}^{-1}\boldsymbol{A}^T$ is feasible. Indeed, this approach is fundamental to many interior point methods for linear programming (see, for example, [34], [31], [32], or [6]). However, because such an approach is merely the restriction of a particular pivot order applied to (1.7), and because it is less appealing when $\boldsymbol{H}$ is not diagonal, Fourer and Mehrotra [22] have suggested methods for solving (1.7) using more general pivot sequences for linear programming problems, and Vanderbei and Carpenter [40] do the same for general problems.

If $\boldsymbol{B}$ is known a priori to be second-order sufficient, as for instance would be the case if $f(\boldsymbol{x})$ were strictly convex, we wholeheartedly recommend the use of MA27, MA47, or the procedure within loqo [40] to solve (1.6). When there is a chance that $\boldsymbol{B}$ may not be second-order sufficient, alternatives to blindly solving (1.6) must be sought. We note that it is always possible to determine a posteriori if $\boldsymbol{B}$ is second-order sufficient, as [28] showed that $\boldsymbol{B}$ is second-order sufficient if and only if $\boldsymbol{K}$ has precisely m negative and n positive eigenvalues. The inertia of $\boldsymbol{K}$ is trivially obtained by summing the inertia of the pivots.

Having set the scene, we may now describe our goals. We aim

(i) to determine a matrix $\boldsymbol{B} = \boldsymbol{H} + \boldsymbol{E}$ so that (1.5) is satisfied and such that the perturbation $\boldsymbol{E} = \mathbf{0}$ whenever $\boldsymbol{H}$ satisfies (1.5);

(ii) to obtain $\boldsymbol{E}$ without incurring undue overheads above those normally considered acceptable when calculating the search direction;

(iii) to ensure that $\|\boldsymbol{E}\|$ is bounded relative to $\max(\|\boldsymbol{A}\|, \|\boldsymbol{H}\|)$—provided that $\{\boldsymbol{x}\}$ remains bounded, this will ensure that $\boldsymbol{B}$ is uniformly bounded;

(iv) to use the sparsity and structure of (1.6) to derive a sparse factorization; and

(v) to limit numerical growth to acceptable limits to ensure a stable algorithm.

In this paper, we shall show how, to a certain extent, we may achieve these aims. The paper is organized as follows. In section 2 we describe Forsgren and Murray's modified factorization approach [21]. In sections 3 and 4, we describe techniques which are particularly attractive when the systems are large and sparse, and we indicate in section 5 how the ideas presented in this paper behave in practice, using a prototype code.

We shall denote the inertia of the generic symmetric matrix $\boldsymbol{M}$ as

$$\mathrm{In}\,(\boldsymbol{M}) = (m_+, m_-, m_0), \tag{1.9}$$

where m_+, m_-, and m_0 are, respectively, the numbers of positive, negative, and zero eigenvalues of $\boldsymbol{M}$. The n by n identity matrix will be written as $\boldsymbol{I}_n$, or $\boldsymbol{I}$ when the dimension is clear from the context. Finally, $\boldsymbol{e}_i$ will be the ith column of $\boldsymbol{I}$. We stress that throughout the paper, the matrix $\boldsymbol{B}$ denotes a second-order sufficient approximation to $\boldsymbol{H}$ which will actually be $\boldsymbol{H}$ whenever the latter is itself second-order sufficient.

2. Forsgren and Murray's sufficient pivoting conditions. As far as we are aware, the only serious attempt to generalize the modified Cholesky methods for unconstrained optimization to the general, large-scale, linearly constrained case is that by Forsgren and Murray [21]. All other methods we are aware of either are appropriate only for small-scale calculations because they disregard problem structure (see [19, section 11.1] or [27, section 5.1] for example) or implicitly assume that $n-m$ is sufficiently small that coping with dense matrices of order $n-m$ is practicable (see, for instance, [36]).

We say that the first n rows of $\boldsymbol{K}$ are $\boldsymbol{B}$-rows and the remaining m rows are $\boldsymbol{A}$-rows. Forsgren and Murray show that, if the pivots are restricted to be of certain types until all of the $\boldsymbol{A}$-rows of $\boldsymbol{K}$ have been eliminated, the remaining uneliminated (Schur-complement) matrix, $\boldsymbol{S}$, is sufficiently positive definite if and only if $\boldsymbol{B}$ is second-order sufficient. Until all $\boldsymbol{A}$-rows of $\boldsymbol{K}$ have been exhausted, Forsgren and Murray allow only the following types of pivots:

$\boldsymbol{b}_+$ pivots: strictly positive 1 by 1 pivots occurring in $\boldsymbol{B}$-rows of $\boldsymbol{K}$.

$\boldsymbol{a}_-$ pivots: strictly negative 1 by 1 pivots occurring in $\boldsymbol{A}$-rows of $\boldsymbol{K}$.

$\boldsymbol{ba}$ pivots: 2 by 2 pivots with a strictly negative determinant, one of whose rows is a $\boldsymbol{B}$-row and the other of whose rows is an $\boldsymbol{A}$-row of $\boldsymbol{K}$.

Forsgren and Murray further restrict the pivot so that the absolute value of its determinant is greater than a small positive constant so as to bound the elements in $\boldsymbol{L}$ and limit any growth in $\boldsymbol{S}$. The motivation behind this choice of pivot is simply that if i $\boldsymbol{A}$-rows have been eliminated, the factorized matrix has exactly i negative eigenvalues. Thus, when all $\boldsymbol{A}$-rows have been eliminated, the factorized matrix has precisely m negative eigenvalues and hence any further negative eigenvalues in $\boldsymbol{S}$ can occur only because $\boldsymbol{B}$ is not second-order sufficient.

Once $\boldsymbol{S}$ has been determined, Forsgren and Murray form a stable symmetric indefinite factorization. If the factors reveal that $\boldsymbol{S}$ is sufficiently positive definite, the (quasi-) Newton equations (1.6) are subsequently solved using the factorization. If $\boldsymbol{S}$ has insufficiently many positive eigenvalues, Forsgren and Murray show how both a direction of sufficient descent and a direction of negative curvature may be recovered from the factors, and they form a search arc as a linear or nonlinear combination of these two directions.

An obvious variation is, instead, to form a modified Cholesky factorization of $\boldsymbol{S}$. If no modification is performed, the true Hessian $\boldsymbol{H}$ must be second-order sufficient. Otherwise, a suitable perturbation $\boldsymbol{E}$ will have been produced. In either case, the Newton equations (1.6) are solved using the complete factorization.

The main difficulty with Forsgren and Murray's approach is that any restriction on the pivot order can disqualify potentially advantageous sparsity orderings. While it is always possible to choose a pivot according to the Forsgren–Murray recipe, the available choices all may lead to considerable fill-in. Nonetheless, we shall consider a number of variations of this scheme.

3. Methods using *ba* pivots.

In this section, we consider a scheme which uses a restricted version of [21]'s pivoting rules. Specifically, we consider what happens if we choose the first m pivots to be *ba* pivots. This choice of pivots is covered by Forsgren and Murray's analysis. We are particularly interested in these pivots because the fill-in is easy to predict and, most important, the stability of the method is determined entirely by $\boldsymbol{A}$. Hence, for linearly constrained problems, the same sequence of *ba* pivots may be used at each iteration.

Since we are primarily concerned with large problems, it is essential to try to ensure that the chosen permutation $\boldsymbol{P}$ introduces as little fill-in as possible. Notice that each *ba* pivot requires that we select a row and a column of $\boldsymbol{A}$ and that the selected column of $\boldsymbol{A}$ defines the row of $\boldsymbol{B}$ used.

Without loss of generality, we describe how the first *ba* pivot is determined. The same procedure then may be applied recursively to the Schur-complement of this pivot in $\boldsymbol{K}$ to determine *ba* pivots $2, \ldots, m$. Suppose that we consider the permutation

$$\boldsymbol{P}_1^T \boldsymbol{K} \boldsymbol{P}_1 = \left(\begin{array}{cc|cc} \beta & \alpha & \boldsymbol{b}_c^T & \boldsymbol{a}_c^T \\ \alpha & 0 & \boldsymbol{a}_r^T & \boldsymbol{0} \\ \hline \boldsymbol{b}_c & \boldsymbol{a}_r & \boldsymbol{B}_R & \boldsymbol{A}_R^T \\ \boldsymbol{a}_c & \boldsymbol{0} & \boldsymbol{A}_R & \boldsymbol{0} \end{array} \right), \tag{3.1}$$

where $\alpha \neq 0$ and β are scalars, $\boldsymbol{b}_c$ and $\boldsymbol{a}_r$ are $(n-1)$-vectors, $\boldsymbol{a}_c$ is an $(m-1)$-vector, and $\boldsymbol{B}_R$ and $\boldsymbol{A}_R$ are $n-1$ by $n-1$ and $m-1$ by $n-1$ matrices, respectively. Then a simple calculation reveals that the Schur-complement of the *ba* pivot in $\boldsymbol{P}_1^T \boldsymbol{K} \boldsymbol{P}_1$ is

$$\begin{aligned} \boldsymbol{S}_1 = & \begin{pmatrix} \boldsymbol{B}_R & \boldsymbol{A}_R^T \\ \boldsymbol{A}_R & \boldsymbol{0} \end{pmatrix} \\ & - \frac{1}{\alpha^2} \left[\alpha \begin{pmatrix} \boldsymbol{b}_c \\ \boldsymbol{a}_c \end{pmatrix} (\boldsymbol{a}_r^T \quad \boldsymbol{0}) + \alpha \begin{pmatrix} \boldsymbol{a}_r \\ \boldsymbol{0} \end{pmatrix} (\boldsymbol{b}_c^T \quad \boldsymbol{a}_c^T) - \beta \begin{pmatrix} \boldsymbol{a}_r \\ \boldsymbol{0} \end{pmatrix} (\boldsymbol{a}_r^T \quad \boldsymbol{0}) \right]. \end{aligned} \tag{3.2}$$

Notice that no fill-in occurs in the zero, bottom block of $\boldsymbol{S}_1$. Furthermore, suppose that we have picked α so that

$$|\alpha| \geq \upsilon \|\boldsymbol{a}_r\|_\infty \tag{3.3}$$

for some pivot tolerance $0 < \upsilon \leq 1$. Then it follows from (3.2) and (3.3) that the largest entry in the updated $\boldsymbol{A}$ can grow by a factor of at most $1 + 1/\upsilon$, while that in the updated $\boldsymbol{B}$ can grow by at most $(1 + 1/\upsilon)^2$. While these factors may be large, they do provide an upper bound on the overall growth factor after a sequence of *ba* pivots. Indeed, if we perform m *ba* pivots, then it is easy to show that

$$\|\boldsymbol{S}\| \leq (1 + \|\boldsymbol{A}_1^{-1} \boldsymbol{A}_2\|)^2 \|\boldsymbol{B}\|, \tag{3.4}$$

where $\boldsymbol{A} = (\boldsymbol{A}_1 \quad \boldsymbol{A}_2)\boldsymbol{P}$ (see [29]). Hence element growth may be controlled by repeated use of (3.3), and if one of the modified Cholesky methods cited in the introduction is subsequently employed to factorize $\boldsymbol{S}$, the perturbation matrix $\boldsymbol{E}$ will remain bounded in terms of the initial $\boldsymbol{A}$ and $\boldsymbol{B}$.

As the same permutation may be used at *every* iteration of the nonlinear programming algorithm, it is worth investing considerable effort in producing a good ordering.

We now follow [33] by picking the *ba* pivot to modify the least number of coefficients in the remaining $n+m-2$ order block of $\boldsymbol{P}_1^T\boldsymbol{K}\boldsymbol{P}_1$ as the Schur complement is formed. Thus we aim to minimize the number of nonzeros, n_s, in the matrix

$$\alpha\begin{pmatrix}\boldsymbol{b}_c\\ \boldsymbol{a}_c\end{pmatrix}(\boldsymbol{a}_r^T \quad \boldsymbol{0}) + \alpha\begin{pmatrix}\boldsymbol{a}_r\\ \boldsymbol{0}\end{pmatrix}(\boldsymbol{b}_c^T \quad \boldsymbol{a}_c^T) - \beta\begin{pmatrix}\boldsymbol{a}_r\\ \boldsymbol{0}\end{pmatrix}(\boldsymbol{a}_r^T \quad \boldsymbol{0}). \tag{3.5}$$

There are two cases to consider.

Following [12], we call a *ba* pivot a *tile* pivot if $\beta \neq 0$ and an *oxo* pivot when $\beta = 0$. We let $n_z(\boldsymbol{v})$ denote the number of nonzeros in the vector $\boldsymbol{v}$, and $n_o(\boldsymbol{v}, \boldsymbol{w})$ give the number of overlaps (the number of indices i for which both v_i and w_i are nonzero) between the vectors $\boldsymbol{v}$ and $\boldsymbol{w}$.

A simple computation reveals that if we choose an oxo pivot, the number of nonzeros in the matrix (3.5) is

$$n_s = 2n_z(\boldsymbol{a}_r)[n_z(\boldsymbol{a}_c) + n_z(\boldsymbol{b}_c)] - n_o(\boldsymbol{a}_r, \boldsymbol{b}_c)^2, \tag{3.6}$$

while a tile pivot yields

$$n_s \leq 2n_z(\boldsymbol{a}_r)[n_z(\boldsymbol{a}_c) + n_z(\boldsymbol{b}_c)] - n_o(\boldsymbol{a}_r, \boldsymbol{b}_c)^2 + [n_z(\boldsymbol{a}_r) - n_o(\boldsymbol{a}_r, \boldsymbol{b}_c)]^2. \tag{3.7}$$

(The inequality in (3.7) accounts for the possibility of exact cancellation between the terms in (3.5).) Thus if $\boldsymbol{A}$ has rows $\boldsymbol{a}_{r_i}$, $i = 1, \ldots, m$, and columns $\boldsymbol{a}_{c_j}$, $j = 1, \ldots, n$, and $\boldsymbol{B}$ has columns $\boldsymbol{b}_{c_j}$, $j = 1, \ldots, n$, one possibility is to pick the *ba* pivot for which

$$|a_{i,j}| \geq \upsilon \max_{1\leq l\leq n} |a_{i,l}| \tag{3.8}$$

and for which

$$\sigma^e_{i,j} = \begin{cases} 2(n_z(\boldsymbol{a}_{r_i}) - 1)(n_z(\boldsymbol{a}_{c_j}) + n_z(\boldsymbol{b}_{c_j}) - 1) - n_o(\boldsymbol{a}_{r_i}, \boldsymbol{b}_{c_i})^2 & \text{when} \quad b_{j,j} = 0, \\ \quad 2(n_z(\boldsymbol{a}_{r_i}) - 1)(n_z(\boldsymbol{a}_{c_j}) + n_z(\boldsymbol{b}_{c_j}) - 2) & \\ - (n_o(\boldsymbol{a}_{r_i}, \boldsymbol{b}_{c_i}) - 1)^2 + (n_z(\boldsymbol{a}_{r_i}) - n_o(\boldsymbol{a}_{r_i}, \boldsymbol{h}_{c_j}) - 2)^2 & \text{when} \quad b_{j,j} \neq 0 \end{cases} \tag{3.9}$$

is smallest. However, since computing $n_o(\boldsymbol{a}_{r_i}, \boldsymbol{b}_{c_j})$ may prove to be unacceptably expensive, we follow [12] and overestimate (3.6) and (3.7) by assuming that, except in the pivot rows, there are no overlaps and thus pick the pivot for which

$$\sigma^a_{i,j} = \begin{cases} 2(n_z(\boldsymbol{a}_{r_i}) - 1)(n_z(\boldsymbol{a}_{c_j}) + n_z(\boldsymbol{b}_{c_j}) - 1) & \text{when} \quad b_{j,j} = 0, \\ 2(n_z(\boldsymbol{a}_{r_i}) - 1)(n_z(\boldsymbol{a}_{c_j}) + n_z(\boldsymbol{b}_{c_j}) - 2) + (n_z(\boldsymbol{a}_{r_i}) - 1)^2 & \text{when} \quad b_{j,j} \neq 0 \end{cases} \tag{3.10}$$

is smallest. It is relatively straightforward to compute and update the nonzero counts required to use (3.10). Indeed, since $n_z(\boldsymbol{a}_{r_i})$ and $n_z(\boldsymbol{a}_{c_j}) + n_z(\boldsymbol{b}_{c_j})$ are, respectively, the row and column counts for the matrix

$$\begin{pmatrix}\boldsymbol{B}\\ \boldsymbol{A}\end{pmatrix}, \tag{3.11}$$

the schemes described by [11, section 9.2] are appropriate.

The main disadvantage of the schemes described in this section is that by restricting the pivot order, the fill-in within the Schur complement may prove unacceptable. This will be the case if $\boldsymbol{A}$ contains dense rows since then the Schur complement will almost certainly be completely dense.

A possible way of alleviating this difficulty is to allow all of the pivot types suggested by [21] (see section 2). A drawback is that by allowing b_+ and a_- pivots, we may generate nonzeros (fill-ins) in the "zero" block of (1.7) and thereafter the Markowitz costs (3.9) and (3.10) would be inappropriate. Appropriate Markowitz costs in this case have been suggested by [12]. Preference still should be given to pivots involving $\boldsymbol{A}$-rows if at all possible. A more serious complication is that $\boldsymbol{B}$ will contaminate $\boldsymbol{A}$ if these additional pivot types are allowed, and thus it may no longer be possible to use the same pivot order for a sequence of related problems.

Even if we allow all types of pivots suggested by Forsgren and Murray, there will certainly be cases where the Schur complement becomes unacceptably dense. In the next section, we consider methods that aim to avoid such difficulties.

4. Modified pivoting methods.

Suppose that $\boldsymbol{A}$ contains m_d rows with a large number of nonzeros and that the remaining $m_e \equiv m - m_d$ rows are sparse. Then it is likely that if any of the dense $\boldsymbol{A}$-rows is included in an early pivot, the remaining Schur complement will substantially fill in. It therefore makes sense to avoid pivoting on these rows until the closing stages of the elimination when the Schur complement may be treated as a dense matrix. However, the Forsgren–Murray pivoting rules may conspire to make this impossible.

Let us suppose that we have eliminated the sparse m_e rows of $\boldsymbol{A}$ using Forsgren and Murray's pivoting rules and that the remaining Schur complement $\boldsymbol{S}$ is relatively sparse excepting the m_d $\boldsymbol{A}$-rows. Thus, we may no longer use ba or a_- pivots and are restricted to using b pivots, that is, 1 by 1 pivots occurring in $\boldsymbol{B}$-rows of $\boldsymbol{S}$.

We may continue the factorization of $\boldsymbol{S}$ in two ways. First, we might pick a favorable pivoting sequence for 1 by 1 pivots from the $\boldsymbol{B}$-rows of $\boldsymbol{S}$ purely from a sparsity (fill-in) point of view. Such an approach implicitly assumes that the defined pivot sequence will be acceptable from a numerical viewpoint and is typical of the symbolic analysis phase of the sparse factorization of positive definite matrices (see, for example, [23] or [11]). Having determined the pivot sequence, a numerical (Cholesky or LDL^T) factorization stage proceeds either to completion or until an unacceptable numerical pivot is encountered. In our case, we view any pivot less than a small positive threshold as unacceptable and, slightly abusing notation, shall refer to this pivot as a b_- pivot. If a b_- pivot is encountered, a readjustment of the pivot order may allow the factorization to proceed further, but this is likely to introduce extra fill-in and merely delays us from facing up to an unacceptable pivot.

Second, we might use a combined analysis-factorization strategy, more typical of unsymmetric factorizations (again, see [11]), in which the pivot order is determined as the factorization proceeds and numerically unacceptable pivots moved down the pivot order. Ultimately, once again, if the b-rows/columns of $\boldsymbol{S}$ are insufficiently positive definite, this process will ultimately break down since all remaining b pivots will be b_- pivots. More fill-in may be predicted with this strategy than with the last, and, in the worst case, restrictions on the pivot order may produce an unacceptable level of fill-in within $\boldsymbol{B}$. Our preference is for the first (separate analysis and factorization phases) strategy because the second strategy is likely to prove a considerable overhead in optimization applications when many systems with the same structure are to be solved.

Thus our remaining concern is when the pivot we wish to use, or are forced to use, next is a b_- pivot. We shall refer to this as a *potential breakdown.* At this stage, we are no longer able to take Forsgren–Murray pivots. We now assume that the b_- pivot would be acceptable from the point of view of fill-in. We aim to investigate the consequences of attempting to use this pivot. Remember that our goal is ultimately only to modify $\boldsymbol{B}$ if it fails to be second-order sufficient.

4.1. Implicit modifications. In this section, we consider *always* modifying b_- pivots, but with the knowledge that we can reverse the effect of these modifications at a later stage.

4.1.1. Pseudo modification of b_- pivots. Suppose the uneliminated Schur-complement when we encounter potential breakdown is of the form

$$\begin{pmatrix} \beta_- & \boldsymbol{s}^T \\ \boldsymbol{s} & \boldsymbol{S} \end{pmatrix}, \tag{4.1}$$

where $\beta_- < \sigma_1$ is the candidate b_- pivot. Now suppose that

$$\beta_+ = \max(\sigma_2, ||\boldsymbol{s}||_\infty), \tag{4.2}$$

where $0 < \sigma_1 \leq \sigma_2$, and let

$$\Delta\beta \overset{\text{def}}{=} \beta_+ - \beta_-. \tag{4.3}$$

Then if we could replace β_- by β_+, the latter would be an acceptable pivot. This is precisely what we do, leaving the consequences for later. We call such a modification of $\boldsymbol{B}$ a *pseudo modification* since it is not yet clear that such a modification is actually required to guarantee that $\boldsymbol{B}$ is second-order sufficient.

We propose continuing such a strategy of replacing b_- pivots with acceptable b_+ pivots until the remaining Schur complement is sufficiently small that it may be treated by dense factorization methods. Thereafter, the Forsgren–Murray strategy may be applied to remove the remaining dense $\boldsymbol{A}$-rows, and a modified Cholesky factorization may then be applied to whatever remains. Thus the resulting (modified) Hessian matrix will be second-order sufficient. However, when replacing any b_- pivots with acceptable b_+ pivots, we may have unnecessarily altered elements and must now reverse any damage caused.

Stewart [39] suggested using pseudo modifications as an alternative to pivoting in Gaussian elimination, and he provided a satisfactory error analysis when a single modification is made. Such an analysis may, of course, be used recursively to cover the scheme suggested here. He comments that this strategy may be particularly beneficial for sparse problems, where altering the pivot sequence could lead to undesirable fill-in.

We will have formed a stable factorization of

$$\boldsymbol{N} \overset{\text{def}}{=} \begin{pmatrix} \boldsymbol{B} + \boldsymbol{\Delta B} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix}, \tag{4.4}$$

where $\boldsymbol{B} = \boldsymbol{H} + \boldsymbol{\Delta H}$, the diagonal matrix $\boldsymbol{\Delta B}$ corresponds to the m_- (say) modified b_- pivots, and the other diagonal matrix $\boldsymbol{\Delta H}$ corresponds to those diagonals changed using the dense modified Cholesky factorization. These later diagonal modifications are necessary to ensure that $\boldsymbol{B}$ is second-order sufficient, while it is not clear that the former modifications are so. Thus we investigate the consequences of removing these modifications.

4.1.2. Countering the effects of pseudo modifications. The system we wish to solve is (1.6), while we have a factorization of (4.4). Suppose that the ith pseudo modification ($1 \le i \le m_-$) occurred in column j_i of $\boldsymbol{H}$ and that the modification was $\Delta\beta_{j_i} > 0$. Let

$$\boldsymbol{\Delta B} = \boldsymbol{V}^T\boldsymbol{V}, \tag{4.5}$$

where $\boldsymbol{V}^T$ is the n by m_- matrix whose columns are $\boldsymbol{v}_i \overset{\text{def}}{=} \sqrt{\Delta\beta_{j_i}}\boldsymbol{e}_{j_i}$. Then we may write (1.6) as

$$\begin{pmatrix} \boldsymbol{B} + \boldsymbol{\Delta B} - \boldsymbol{V}^T\boldsymbol{V} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{g} \\ \boldsymbol{0} \end{pmatrix} \tag{4.6}$$

or equivalently as

$$\begin{pmatrix} \boldsymbol{B} + \boldsymbol{\Delta B} & \boldsymbol{A}^T & \boldsymbol{V}^T \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{V} & \boldsymbol{0} & \boldsymbol{I}_{m_-} \end{pmatrix} \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{\lambda} \\ \boldsymbol{s} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{g} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix} \tag{4.7}$$

for some auxiliary vector $\boldsymbol{s}$. A standard block-decomposition of (4.7) shows that we may determine the solution to (1.6) by solving, in order,

$$\begin{pmatrix} \boldsymbol{B} + \boldsymbol{\Delta B} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{\pi} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{g} \\ \boldsymbol{0} \end{pmatrix}, \tag{4.8}$$

$$\boldsymbol{G}\boldsymbol{s} = \boldsymbol{V}\boldsymbol{q}, \tag{4.9}$$

and

$$\begin{pmatrix} \boldsymbol{B} + \boldsymbol{\Delta B} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{g} - \boldsymbol{V}^T\boldsymbol{s} \\ \boldsymbol{0} \end{pmatrix}, \tag{4.10}$$

where

$$\boldsymbol{G} = \boldsymbol{I}_{m_-} - (\boldsymbol{V} \quad \boldsymbol{0}) \begin{pmatrix} \boldsymbol{B} + \boldsymbol{\Delta B} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{V}^T \\ \boldsymbol{0} \end{pmatrix}. \tag{4.11}$$

Thus to solve (1.6) via the stable factorization (4.4), we also need to form and factorize $\boldsymbol{G}$. This factorization also reveals whether the modification $\boldsymbol{\Delta B}$ is necessary. Thus we have the following proposition.

PROPOSITION 4.1. *$\boldsymbol{B}$ is second-order sufficient if and only if $\boldsymbol{G}$ is positive definite.*

Proof. The result follows from Sylvester's law of inertia (see, e.g., [8]) by considering different block decompositions of

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{B} + \boldsymbol{\Delta B} & \boldsymbol{A}^T & \boldsymbol{V}^T \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{V} & \boldsymbol{0} & \boldsymbol{I}_{m_-} \end{pmatrix}. \tag{4.12}$$

Pivoting on the first two blocks of $\boldsymbol{M}$ and using the definitions (4.4) and (4.11) reveals that

$$\text{In}\,(\boldsymbol{M}) = \text{In}\,(\boldsymbol{N}) + \text{In}\,(\boldsymbol{G}), \tag{4.13}$$

while pivoting on the last block and using the definition (1.7) gives that

$$\operatorname{In}(\boldsymbol{M}) = \operatorname{In}\left(\boldsymbol{I}_{m_-}\right) + \operatorname{In}(\boldsymbol{K}). \tag{4.14}$$

But, since $\boldsymbol{I}_{m_-}$ is clearly positive definite and $\boldsymbol{N}$ is second-order sufficient,

$$\operatorname{In}\left(\boldsymbol{I}_{m_-}\right) = (m_-, 0, 0) \quad \text{and} \quad \operatorname{In}(\boldsymbol{N}) = (n, m, 0). \tag{4.15}$$

Thus combining (4.13)–(4.15), we see that $\operatorname{In}(\boldsymbol{K}) = (n, m, 0)$ if and only if $\operatorname{In}(\boldsymbol{G}) = \operatorname{In}\left(\boldsymbol{G}^{-1}\right) = (m_-, 0, 0)$. The required result then follows since $\boldsymbol{B}$ is second-order sufficient if and only if $\operatorname{In}(\boldsymbol{K}) = (n, m, 0)$ (see [28]). □

This result suggests that $\boldsymbol{G}$ should be factorized using a modified Cholesky factorization. If no modification to $\boldsymbol{G}$ is made, $\boldsymbol{B}$ is second-order sufficient. Suppose, on the other hand, that the ith pseudo modification involved column j_i of $\boldsymbol{H}$ and that in the subsequent modified Cholesky factorization the ith diagonal of $\boldsymbol{G}$ was increased by $\Delta\gamma_i$. Then this is equivalent to *actually* modifying $\boldsymbol{B}$ by

$$\left(\frac{\Delta\gamma_i}{1+\Delta\gamma_i}\right) \boldsymbol{v}_i \boldsymbol{v}_i^T. \tag{4.16}$$

Thus modification of $\boldsymbol{G}$ gives an implicit modification of $\boldsymbol{B}$, and the actual modification is no larger than the pseudo modification.

We note that another possibility is to attempt a Cholesky factorization of $\boldsymbol{G}$ but to retain all the pseudo modifications if the factorization reveals that $\boldsymbol{G}$ is not sufficiently positive definite. This is equivalent to solving (4.8) and then setting $\boldsymbol{p} = \boldsymbol{q}$ and $\boldsymbol{\lambda} = \boldsymbol{\pi}$. However, instinctively we feel that it is better to remove as many pseudo modifications as possible, and thus we prefer to use the modified Cholesky factorization and (4.8)–(4.10).

4.1.3. The pseudo-modification algorithm. In summary, we propose the following algorithm:

1. Perform a symbolic/numerical analysis and factorization to obtain a good ordering for the complete numerical factorization.

(i) First, construct k_2 2 by 2 ba pivots, using the strategy outlined in section 3 (this involves processing the values of $\boldsymbol{A}$ but not $\boldsymbol{B}$). Stop once the resulting Schur-complement reaches a specified density, i.e., the proportion of its nonzero entries exceeds a given threshold.

(ii) Next, construct k_1 1 by 1 b pivots from the remaining Schur-complement using, for instance, the minimum degree ordering (see, for example, [23] or [11]). Stop once the resulting Schur-complement reaches a specified density.

(iii) The remaining Schur complement will be considered to be dense.

2. Perform the complete numerical factorization.

(i) Perform k_2 2 by 2 sparse eliminations, using the pivots specified in 1(i) above.

(ii) Perform k_1 1 by 1 sparse eliminations, using the pivots specified in 1(ii) above. Modify any b_- pivots encountered to ensure that they are sufficiently positive, using the scheme of, for example, [38]. Record any pseudo modifications made.

(iii) For the remaining dense block, factorize using the scheme of [21] until all the $\boldsymbol{A}$-rows have been eliminated. Thereafter, use a dense modified Cholesky factorization to eliminate the remaining $\boldsymbol{B}$-rows.

(iv) If any pseudo modifications were made in 2(ii) above, form the matrix $\boldsymbol{G}$. Perform a modified Cholesky factorization of $\boldsymbol{G}$.

3. Perform any solves, using the factors obtained in 2 above, by solving the sequence of equations (4.8)–(4.10).

One would normally anticipate performing only a single symbolic/numerical analysis and factorization per minimization, while many complete numerical factorizations and solves might be required. Thus, a good ordering will pay handsome dividends, and one might be prepared to expend considerable effort in step 1.

We should stress that (3.2) indicates that the Schur-complement of the $\boldsymbol{A}$-rows following the elimination of the *ba* pivots is independent of $\boldsymbol{B}$ and thus, since $\boldsymbol{A}$ is independent of $\boldsymbol{x}$, need be formed only once per minimization. This is the only numerical processing involved in the symbolic/numerical analysis and factorization phase.

Notice that the effectiveness of such a scheme depends upon the dimension of $\boldsymbol{G}$. Although the number of pseudo modifications will not be known until the numerical factorization phase, it may be possible to influence this by overriding the pivot selection outlined in section 3 to favor incorporating potentially small or negative elements within the initial *ba* pivots. For instance, if a diagonal of $\boldsymbol{B}$ is known to be negative, it may be worth trying to encourage this element to lie within a *ba* pivot so that it will not be available for pseudo modification in step 1(ii).

4.1.4. Generalizations. When $\boldsymbol{H}$ is strictly positive definite, no pseudo modifications should be necessary. In other cases, it is possible that the dimension of $\boldsymbol{G}$ might be unacceptably high. However, considering (4.1), it is clear that rather in addition to modifying the diagonal β_-, we are free to modify *as many nonzero elements of* $\boldsymbol{s}$ *as we like* without introducing extra fill in the factorization. Thus, if the diagonal of $\boldsymbol{S}$, in a row in which $\boldsymbol{s}$ has a nonzero element, is also small or negative, we should modify the corresponding element of $\boldsymbol{s}$ to increase the offending diagonal. All that we have said in this section about diagonal modifications equally applies for the more general perturbation, but the ith column of the matrix $\boldsymbol{V}$ now contains nonzeros in all positions for which the j_ith pivot column was modified.

Nonetheless, we have to recognize that there are some matrices for which this strategy is inappropriate, as the following example shows.

Example 4.1. Suppose $\boldsymbol{H} = -I_n$ and $\boldsymbol{A} = \boldsymbol{e}^T$, the vector of ones. Then a *ba* pivot is out of the question because the resulting Schur complement would be completely dense. But since each diagonal of $\boldsymbol{H}$ is negative, and there is no connectivity between the diagonals, n pseudo modifications will be required. Unfortunately, $\boldsymbol{G}$ will then be a dense n by n matrix.

Another possibility is to replace $\boldsymbol{G}$ by a simpler matrix as soon as $\boldsymbol{G}$ is found to be indefinite. If we replaced $\boldsymbol{G}$ by $\boldsymbol{G} + \Delta\boldsymbol{G}$, it is straightforward to show that this is equivalent to an actual modification of $\boldsymbol{B}$ by

$$\boldsymbol{V}^T \left(I_{m_-} - (I_{m_-} + \Delta\boldsymbol{G})^{-1}\right) \boldsymbol{V}. \tag{4.17}$$

Thus, provided that $\Delta\boldsymbol{G}$ is positive semidefinite, the actual modification will again be smaller than the pseudo modification. A simple scheme would be to replace $\boldsymbol{G}$ by $\tau\boldsymbol{I}_{m_-}$, where τ is chosen so large that $\tau\boldsymbol{I}_{m_-} - \boldsymbol{G}$ is positive definite whenever $\boldsymbol{G}$ is not positive definite. The advantage of this replacement is that the storage and factorization overheads associated with $\boldsymbol{G}$ may be considerably reduced. The disadvantages are that the size of the actual modification made may be higher than really necessary and that it is not obvious how to choose a satisfactory value for τ.

4.2. Explicit modifications. In the previous sections, we always chose to modify b_- pivots, with the knowledge that we could reverse the effect of the modification at a later stage. As we have seen, it may happen that a considerable number of pseudo modifications will be made and this may be undesirable because of the space and effort required to factorize $\boldsymbol{G}$. In this section, we take the opposite point of view and consider changing b_- pivots *only* when we know it is necessary to modify them. The intention with this alternative is thus to remove, or at least lessen, the need for pseudo modifications.

We now assume that a b_- pivot would be acceptable from the point of view of fill-in *and* stability. This is tantamount to assuming that the pivot is negative and of a reasonable size compared to the remaining entries in its row. We aim to investigate the consequences of using this pivot. Remember that our goal remains only to modify $\boldsymbol{B}$ if it fails to be second-order sufficient.

4.2.1. The condemned submatrix. Recall that we are supposing that we have eliminated the sparse m_e rows of $\boldsymbol{A}$ using Forsgren and Murray's pivoting rules and that $m_d \equiv m - m_e$ $\boldsymbol{A}$-rows remain within $\boldsymbol{S}$. Suppose that we have also eliminated n_e rows of $\boldsymbol{B}$ using Forsgren and Murray's rules.

We pick a nonsingular, square submatrix, the *condemned* submatrix, $\boldsymbol{C}$, of $\boldsymbol{S}$, which contains all the $\boldsymbol{A}$-rows and perhaps some of the $\boldsymbol{B}$-rows (but not the b_- pivot row) of $\boldsymbol{S}$ and has precisely m_d negative eigenvalues. The condemned submatrix will be eliminated last of all and thus any $\boldsymbol{B}$-rows included in $\boldsymbol{C}$ will not be generally available as pivots. The aim is that when only the rows that make up $\boldsymbol{C}$ remain to be eliminated, the uneliminated Schur complement will have precisely m_d negative eigenvalues and hence $\boldsymbol{K}$ will have exactly m negative eigenvalues. The Schur complement $\boldsymbol{S}$ has at least m_d negative eigenvalues. A suitable $\boldsymbol{C}$ may be obtained, for instance, by picking $m_{a-} \le \min(n_e - m_e, m_d)$ a_- followed by $m_d - m_{a-}$ ba pivots. We shall show how such a matrix may be obtained in section 4.2.4.

A factorization of the condemned submatrix should be obtained. As the $\boldsymbol{C}$-rows of $\boldsymbol{S}$ will ultimately invariably be dense, a full matrix factorization is appropriate and, because we may subsequently need to modify the factors, we recommend a QR or LQ factorization. This of course limits the scope of the current proposal because of the size of $\boldsymbol{C}$ which can be accommodated. We note that the dimension of $\boldsymbol{C}$ as constructed above is $2m_d - m_{a-}$ and hence lies between m_d and $2m_d$.

4.2.2. The consequences of pivoting. With this choice of $\boldsymbol{C}$, $\boldsymbol{S}$ is a permutation of the matrix

$$\left(\begin{array}{cc|c} \beta_- & \boldsymbol{s}_1^T & \boldsymbol{s}_2^T \\ \boldsymbol{s}_1 & \boldsymbol{C} & \boldsymbol{S}_{21}^T \\ \hline \boldsymbol{s}_2 & \boldsymbol{S}_{21} & \boldsymbol{S}_{22} \end{array}\right), \tag{4.18}$$

where $\beta_- < 0$ is the candidate b_- pivot. If we were now to pivot on $\boldsymbol{C}$ instead of β_-, we would have eliminated all m $\boldsymbol{A}$-rows of $\boldsymbol{K}$ and, because of the choice of $\boldsymbol{C}$, the factorized matrix (the submatrix of $\boldsymbol{K}$ corresponding to eliminated rows) would have exactly m negative eigenvalues. Thus $\boldsymbol{B}$ is second-order sufficient if and only if the matrix

$$\begin{pmatrix} \beta_- & \boldsymbol{s}_2^T \\ \boldsymbol{s}_2 & \boldsymbol{S}_{22} \end{pmatrix} - \begin{pmatrix} \boldsymbol{s}_1^T \\ \boldsymbol{S}_{21} \end{pmatrix} \boldsymbol{C}^{-1} \begin{pmatrix} \boldsymbol{s}_1 & \boldsymbol{S}_{21}^T \end{pmatrix} \tag{4.19}$$

is sufficiently positive definite. In particular, if $\beta_- - \boldsymbol{s}_1^T \boldsymbol{C}^{-1} \boldsymbol{s}_1$ is insufficiently positive, $\boldsymbol{B}$ is not second-order sufficient and should be modified.

With this in mind, if

$$\beta_- - \boldsymbol{s}_1^T \boldsymbol{C}^{-1} \boldsymbol{s}_1 < \sigma_1, \tag{4.20}$$

we modify $\boldsymbol{B}$ by replacing β_- by

$$\beta_+ \overset{\text{def}}{=} \max\left(\sigma_2 + \boldsymbol{s}_1^T \boldsymbol{C}^{-1} \boldsymbol{s}_1, ||\boldsymbol{s}||\right), \tag{4.21}$$

where $\boldsymbol{s}^T = \begin{pmatrix} \boldsymbol{s}_1^T & \boldsymbol{s}_2^T \end{pmatrix}$ and, as before, $0 < \sigma_1 \leq \sigma_2$. Conversely, if

$$\beta_- - \boldsymbol{s}_1^T \boldsymbol{C}^{-1} \boldsymbol{s}_1 \geq \sigma_1 > 0, \tag{4.22}$$

it is safe to pivot on β_-. Moreover, although this implies an increase (by one) in the number of negative eigenvalues that have been recorded, the increase is counteracted by a corresponding reduction in the number of available negative eigenvalues in the Schur complement $\boldsymbol{C} - \boldsymbol{s}_1 \boldsymbol{s}_1^T / \beta_-$. This follows directly from the inertial identity

$$\operatorname{In}\left(\boldsymbol{C} - \boldsymbol{s}_1 \boldsymbol{s}_1^T / \beta_-\right) = \operatorname{In}(\boldsymbol{C}) + \operatorname{In}\left(\beta_- - \boldsymbol{s}_1^T \boldsymbol{C}^{-1} \boldsymbol{s}_1\right) - \operatorname{In}(\beta_-) \tag{4.23}$$

for block decompositions of

$$\begin{pmatrix} \beta_- & \boldsymbol{s}_1^T \\ \boldsymbol{s}_1 & \boldsymbol{C} \end{pmatrix} \tag{4.24}$$

(see, e.g., [8]). We then pivot on the possibly modified value of β_- and replace $\boldsymbol{C}$ by $\boldsymbol{C} - \boldsymbol{s}_1 \boldsymbol{s}_1^T / \beta_-$—we update the matrix factorization to account for this (see [24]). We repeat this procedure until we have eliminated the remaining $\boldsymbol{S}_{22}$ rows, at which point the only noneliminated portion of $\boldsymbol{K}$ is the (updated) matrix $\boldsymbol{C}$.

Alternatively, once it has been determined that $\boldsymbol{B}$ is not second-order sufficient, we might modify all remaining $\boldsymbol{B}$ pivots. One possibility, in the same vein as [38], is to insist that all diagonals are larger than the sum of the absolute values of the (remaining) off-diagonal terms in $\boldsymbol{B}$-rows.

For the case of Example 4.1, the explicit modification scheme considered here would be preferable. The condemned submatrix might be made up from the last row of $\boldsymbol{H}$—indeed, any row of $\boldsymbol{H}$ will do—and the single $\boldsymbol{A}$-row. Examining (4.20) for each diagonal pivot in turn, it follows that $\boldsymbol{H}$ is not second-order sufficient, every pivot will be modified, but no fill-in takes place.

4.2.3. Other pivot types. If the only possible pivots in $\boldsymbol{B}$-rows are zero or small, we may again test them one at a time to see if they might be modified and then used as pivots. If the test reveals that the matrix is not second-order sufficient, we may modify the tested pivot and pivot on it. But if the test is inconclusive, we must either add a pseudo modification (see section 4.1.1) or reject the potential pivot and pass to the next.

It may be better to consider 2 by 2 pivots,

$$\begin{pmatrix} \beta_{11} & \beta_{21} \\ \beta_{21} & \beta_{22} \end{pmatrix}, \tag{4.25}$$

arising from the $\boldsymbol{B}$-rows of $\boldsymbol{S}$, especially when the only possible 1 by 1 pivots are small or zero. Then $\boldsymbol{S}$ is a permutation of the matrix

$$\left(\begin{array}{ccc|c} \beta_{11} & \beta_{21} & \boldsymbol{s}_{11}^T & \boldsymbol{s}_{21}^T \\ \beta_{21} & \beta_{22} & \boldsymbol{s}_{12}^T & \boldsymbol{s}_{22}^T \\ \boldsymbol{s}_{11} & \boldsymbol{s}_{12} & \boldsymbol{C} & \boldsymbol{S}_{21} \\ \hline \boldsymbol{s}_{21} & \boldsymbol{s}_{22} & \boldsymbol{S}_{21} & \boldsymbol{S}_{22} \end{array}\right) \tag{4.26}$$

and $\boldsymbol{B}$ is second-order sufficient only if the matrix

$$\left(\begin{array}{cc} \beta_{11} & \beta_{21} \\ \beta_{21} & \beta_{22} \end{array} \right) - \left(\begin{array}{c} \boldsymbol{s}_{11}^T \\ \boldsymbol{s}_{12}^T \end{array} \right) \boldsymbol{C}^{-1} (\boldsymbol{s}_{11} \quad \boldsymbol{s}_{12}) \tag{4.27}$$

is sufficiently positive definite. As before, if (4.27) is indefinite, the potential pivot (4.25) should be modified before use. The inertial result

$$\begin{aligned} &\operatorname{In} \left(\boldsymbol{C} - (\boldsymbol{s}_{11}^T \quad \boldsymbol{s}_{12}^T) \left(\begin{array}{cc} \beta_{11} & \beta_{21} \\ \beta_{21} & \beta_{22} \end{array} \right)^{-1} \left(\begin{array}{c} \boldsymbol{s}_{11} \\ \boldsymbol{s}_{12} \end{array} \right) \right) \\ &\quad = \operatorname{In}(\boldsymbol{C}) + \operatorname{In} \left(\left(\begin{array}{cc} \beta_{11} & \beta_{21} \\ \beta_{21} & \beta_{22} \end{array} \right) - \left(\begin{array}{c} \boldsymbol{s}_{11}^T \\ \boldsymbol{s}_{12}^T \end{array} \right) \boldsymbol{C}^{-1} (\boldsymbol{s}_{11} \quad \boldsymbol{s}_{12}) \right) - \operatorname{In} \left(\begin{array}{cc} \beta_{11} & \beta_{21} \\ \beta_{21} & \beta_{22} \end{array} \right) \end{aligned} \tag{4.28}$$

once again indicates that the updated $\boldsymbol{C}$ after the pivot inherits the correct number of negative eigenvalues.

4.2.4. Calculating the condemned submatrix. In this section, we consider one way in which the initial condemned submatrix, $\boldsymbol{C}$, may be found. We should stress that the definition of $\boldsymbol{C}$ is by no means unique.

Let the m_d uneliminated $\boldsymbol{A}$-rows in the Schur complement, $\boldsymbol{S}_{ba}$, following the m_e *ba* pivots, be $\boldsymbol{A}_{ba}$. Similarly, let the $n - n_e$ uneliminated $\boldsymbol{B}$-rows and columns in $\boldsymbol{S}_{ba}$ following these *ba* pivots be $\boldsymbol{B}_{ba}$. Furthermore, let

$$\mathcal{O} = \{ i_1, i_2, \cdots, i_{n-m_e} \} \tag{4.29}$$

be the ordered pivot sequence for the elimination of $\boldsymbol{B}_{ba}$. Now define the ordered sets

$$\mathcal{P}_1 = \{ i_1, i_2, \cdots, i_{n_e - m_e} \} \quad \text{and} \quad \mathcal{P}_2 = \{ i_{n-m_e}, i_{n-n_e-1}, \cdots, i_{n_e-m_e+1} \} \tag{4.30}$$

and the ordered set of *preferences*

$$\mathcal{P} = \mathcal{P}_1 \bigcup \mathcal{P}_2. \tag{4.31}$$

For example, suppose that $\boldsymbol{B}$-rows 1, 6, and 4 were involved in *ba* pivots; that the remaining *b* pivots were requested from rows 3, 7, 5, 8, and 2 in order (thus, $\mathcal{O} = \{3, 7, 5, 8, 2\}$); that the pivots from rows 3 and 7 were satisfactory; but that row 5 is a b_- pivot. Then $\mathcal{P}_1 = \{3, 7\}$, $\mathcal{P}_2 = \{2, 8, 5\}$ and $\mathcal{P} = \{3, 7, 2, 8, 5\}$.

Our intention is to find a well-conditioned, nonsingular subset, $\mathcal{C}$, of the columns of $\boldsymbol{A}_{ba}$ by pivoting. The row and column indices of the pivots would provide satisfactory *ba* pivots, if such pivots had not been disqualified on sparsity grounds, for $\boldsymbol{S}_{ba}$. Moreover, the submatrix, $\boldsymbol{C}_{ba}$, formed by taking the rows and columns of $\boldsymbol{S}_{ba}$ corresponding to these 2 by 2 pivots is nonsingular and has precisely m_d negative eigenvalues. If we now consider the subsets $\mathcal{C}_1 = \mathcal{C} \bigcap \mathcal{P}_1$ and $\mathcal{C}_2 = \mathcal{C} \bigcap \mathcal{P}_2$, and pivot on all $\boldsymbol{B}$-rows of $\boldsymbol{C}_{ba}$ whose indices occur in $\mathcal{C}_1$, the remaining Schur-complement still has precisely m_d negative eigenvalues and provides us with a suitable condemned submatrix, $\boldsymbol{C}$. This matrix has the correct inertia as the subblock of $\boldsymbol{C}_{ba}$ corresponding to the $\mathcal{C}_1$ pivots is contained within the subblock of $\boldsymbol{S}_{ba}$ corresponding to the $\mathcal{P}_1$ pivots, and the latter subblock is positive definite since the first $n_e - m_e$ *b* pivots on $\boldsymbol{S}_{ba}$ are positive.

It remains to describe how $\mathcal{C}$ is calculated. We consider how the first element is obtained, the remaining $m_d - 1$ elements following in exactly the same way. The set $\mathcal{C}$ is initially empty and the matrix $\boldsymbol{A}_c$ is initialized as $\boldsymbol{A}_{ba}$. The columns of

$\boldsymbol{A}_c$ are considered one at a time, in the order defined by $\mathcal{P}$. The nonzeros in the current column are examined one at a time. If the entry in row i and column j is that currently under examination and if the stability restriction (3.8) holds (where here $a_{i,j}$ are the entries of $\boldsymbol{A}_c$), column j is added to $\mathcal{C}$ and removed from $\mathcal{P}$, and $\boldsymbol{A}_c$ is reset to the Schur complement of $\boldsymbol{A}_c$ following a pivot on $a_{i,j}$. On the other hand, if (3.8) fails to hold, attention passes to the next nonzero in column j or, if there are no further unexamined entries in the column, to the next column in $\mathcal{P}$.

The order of the preferences $\mathcal{P}$ is chosen deliberately. It first encourages ba pivots whose b_+ component has already been used—the resulting a_- pivot is then available and reduces the possible dimension of $\boldsymbol{C}$. If $\mathcal{P}$ is not entirely made up from $\mathcal{P}_1$, the preference then encourages pivots from those $\boldsymbol{B}$-rows which are last in the elimination ordering—the intention here is that these are unlikely to be good pivots from a fill-in point of view and so it is better to include them in the dense matrix $\boldsymbol{C}$ from the outset.

A disadvantage of the preceding approach is that the order of the set $\mathcal{P}$ depends on at which stage a b_- pivot appears. This may be significant if more than one matrix factorization is required as changes in $\boldsymbol{B}$ may affect $\mathcal{P}$. It may, therefore, be preferable to redefine the preference as

$$\mathcal{P} = \{i_{n-m_e}, i_{n-m_e-1}, \cdots, i_1\}\,. \tag{4.32}$$

This will have the effect that the resulting $\boldsymbol{C}$ will generally be of dimension $2m_d$ but the advantage that the selection of $\mathcal{C}$ is made only once. As before, it will favor including disadvantageous $\boldsymbol{B}$-rows within the condemned submatrix.

5. Numerical experiments.

We are currently planning to implement a code to solve systems of the form (1.6) for the Harwell Subroutine Library. A key requirement is that $\boldsymbol{B}$ should be a second-order sufficient modification of $\boldsymbol{H}$. To test the efficacy of some of the ideas presented in this paper, we report on experiments conducted with a prototype, `KKTSOL`, of this code.

5.1. Implementation details.

We have written a prototype implementation of the algorithm outlined in section 4.1.3. This implicit modification algorithm divides naturally into an analysis, a factorization, and a solve phase.

The analysis phase need be performed only once for a sequence of systems so long as the matrix $\boldsymbol{A}$ and the sparsity structure of $\boldsymbol{H}$ are unchanged. Some numerical processing of the matrix $\boldsymbol{A}$ is performed in the analysis phase. There are several control parameters, in particular the pivot threshold tolerance υ (see (3.8)), the density δ_a of the Schur complement of $\boldsymbol{A}$ during ba pivoting at which the remaining rows of $\boldsymbol{A}$ may be treated as dense, and the density δ_b of the Schur complement of $\boldsymbol{B}$ during b pivoting at which the remaining rows of $\boldsymbol{B}$ may be treated as dense. We choose to switch to full-matrix code as soon as the density of the Schur complement of $\boldsymbol{B}$ during b pivoting exceeds δ_b. However, experience has shown that switching to b pivoting as soon as the density δ_a of the Schur complement of $\boldsymbol{A}$ during ba pivoting exceeds δ_a is sometimes inappropriate since further cheap ba pivots may be possible—remember that it helps to eliminate as many $\boldsymbol{A}$ rows as possible before the b pivoting stage. In particular, if the matrix $\boldsymbol{A}$ is highly structured, many essentially identical ba pivots occur and switching solely on the basis of δ_a may interrupt a promising sequence of pivots. Thus, we actually choose to switch as soon as the density has exceeded its tolerance and the Markowitz cost (3.10) next changes. This heuristic has worked well in our tests. Default values of $\upsilon = 0.0001$, $\delta_a = 0.1$, and $\delta_b = 0.25$ have proved quite reliable. We will indicate the effect of δ_a on the algorithm in section 5.3.

Table 5.1

Problem characteristics. Key: n = number of variables; m = number of equations; nnz A, H *= number of nonzeros in A and $H(x)$;* -eval *= number of negative eigenvalues of reduced Hessian;* nullity *= nullity of K;* convex? *= is the Hessian $H(x)$ positive definite?*

Problem	n	m	nnz A	nnz H	-eval	nullity	convex?
AUG2DCQP	3280	1600	6400	3280	0	0	yes
AUG2DQP	3280	1600	6400	3120	0	0	yes
AUG3DCQP	3873	1000	6546	3873	0	0	yes
AUG3DQP	3873	1000	6546	2673	0	0	yes
BLOCKQP1	2006	1001	9006	1005	1000	0	no
BLOCKQP2	2006	1001	9006	1005	1	0	no
BLOCKQP3	2006	1001	9006	1005	900	1	no
GOULDQP2	699	349	1047	697	0	0	yes
GOULDQP3	699	349	1047	1395	0	0	yes
HAGER2	2001	1000	3000	3001	0	0	yes
KSIP	1021	1001	21002	20	0	0	yes
MINC44	1113	1032	1203	0	0	0	yes
MOSARQP1	1500	600	3530	945	0	0	yes
NCVXQP1	1000	500	1498	3984	438	0	no
NCVXQP2	1000	500	1498	3984	320	0	no
NCVXQP3	1000	500	1498	3984	163	0	no
NCVXQP4	1000	250	749	3984	610	0	no
NCVXQP5	1000	250	749	3984	428	0	no
NCVXQP6	1000	250	749	3984	224	0	no
NCVXQP7	1000	750	2247	3984	250	0	no
NCVXQP8	1000	750	2247	3984	192	0	no
NCVXQP9	1000	750	2247	3984	127	0	no
QPCBOEI1	726	351	3827	384	0	0	yes
QPCBOEI2	305	166	1358	143	0	0	yes
QPCSTAIR	614	356	4003	467	0	0	yes
QPNBOEI1	726	351	3827	384	30	0	no
QPNBOEI2	305	166	1358	143	12	0	no
QPNSTAIR	614	356	4003	467	13	0	no
SOSQP1	2000	1001	4000	1000	0	0	no
UBH1	909	600	2400	303	0	0	yes

During the factorization phase, once a b pivot for which $\beta < \sigma_1$ has been detected, any b pivot which is smaller than the sum of absolute values of the off-diagonal terms in its column is pseudo modified. The pseudo modification is chosen to satisfy (4.2), where $\sigma_1 = 10^{-8}$ and $\sigma_2 = 1$. The rows of $\boldsymbol{A}$ and $\boldsymbol{H}$ which are left over following the sparse ba and b pivoting steps, along with the matrix $\boldsymbol{G}$, are treated as dense matrices. The highest appropriate levels of BLAS (see, for instance, [10]) are used to perform the dense operations wherever possible.

In addition, we have also implemented the explicit modification scheme suggested in section 4.2. This differs from the implicit modification scheme described above in two respects. First, the ordering of the b pivots may be altered to provide a nonsingular condemned matrix, if it is needed. We have implemented the method described in section 4.2.4 using the preference (4.32). Second, during the b phase of the factorization, b_+ pivots are used so long as no b_- pivot is detected. If a b_- pivot appears, the condemned matrix is formed and factorized, and the resulting QR decomposition is used to see if this pivot is acceptable or if it should be modified. Subsequent b_- pivots are treated in the same way, except that now the factors of the condemned matrix are obtained from its predecessor by updating. Slightly modified LAPACK routines (see [1]) are used to compute and update the QR factors.

All our tests were performed on an IBM RISC System/6000 3BT workstation

Table 5.2

*Dependence on the allowed density of A. Key: δ_a = density of updated A at which remaining rows are treated as dense (*no dense *means that no dense rows of A are allowed);* fill-in A, H, dense *= fill-in within A, H, and the final dense block;* dense rows A, H = *number of rows of A and H which are treated as dense;* dense rows G = *number of pseudo modifications made (dimension of G); #* mod = *number of diagonals of H actually modified;* anal., fact., solve = *times for analyze, factorize, and solve (cpu seconds).*

AUG3DQP:

	Fill-in			Dense rows			#	Time		
δ_a	A	H	dense	A	H	G	mod	anal.	fact.	solve
0.01	1043	16231	348195	777	57	0	0	.33	3.93	.05
0.05	3161	49458	144991	297	241	0	0	.60	2.30	.03
0.1	3860	67921	171991	198	388	0	0	.71	2.92	.04
0.2	4333	87143	180901	130	471	0	0	.80	3.07	.04
0.5	4773	109644	223446	73	595	0	0	.97	4.22	.04
1.0	5058	138324	245350	47	653	0	0	1.09	5.01	.04
no dense	5806	331483	245350	0	700	0	0	3.99	4.57	.05

QPNBOEI1:

	Fill-in			Dense rows			#	Time		
δ_a	A	H	dense	A	H	G	mod	anal.	fact.	solve
0.01	0	3816	73920	347	3	34	34	.02	.84	.02
0.05	4	3844	43365	250	13	31	33	.02	.40	.01
0.1	4	5229	21115	147	25	33	33	.04	.17	.01
0.2	4	5877	13203	98	38	26	33	.04	.11	.00
0.5	4	6577	9180	63	51	21	36	.04	.08	.00
1.0	16	6574	10296	46	78	19	38	.05	.08	.00
no dense	153	71225	70500	0	375	0	32	.92	.77	.01

with 64 megabytes of RAM; the codes are all double-precision Fortran-77, compiled under xlf with -O optimization, and IBM library BLAS are used.

5.2. Test examples. We considered all of the larger quadratic programming examples in the current CUTE test set (see [3]), except that we excluded those which are minor variants (namely BLOCKQP4, BLOCKQP5, HAGER4, MOSARQP2, and SOSQP2). The characteristics of this test set are described in Table 5.1. All general inequality constraints were converted to equations by the addition of slack variables. To simulate a typical early iteration of a barrier function method, a small value (one-tenth) was added to each diagonal entry of the given Hessian. For each test, the given matrix was factorized and modified if necessary. A right-hand side was then generated so that the required solution is a vector of ones.

5.3. Results. We first illustrate the effect of δ_a, the density of the Schur complement of $\boldsymbol{A}$ during *ba* pivoting at which the remaining rows of $\boldsymbol{A}$ may be treated as dense, on the performance of our algorithm. We consider two examples, AUG3DQP and QPNBOEI1, from our test set; the first is strictly convex while the reduced Hessian of the second has a few negative eigenvalues. The behavior on these examples is representative of the whole set.

In Table 5.2 we give our results on runs which used the explicit modification algorithm; similar results were observed for the implicit modification scheme. Examining the times taken during the analyze and factorize stages, we see that it is important not to let δ_a be too large, as the remaining Schur complement of $\boldsymbol{K}$ is then too dense. On the other hand, skipping pivoting on rows of $\boldsymbol{A}$ when δ_a is too small is also undesirable since the dimension of the resulting dense matrix is then large. Thus a compromise is necessary and we have found, empirically, that a density of around 10% is reasonable.

Table 5.3

Performance of `MA27` *and* `MA47` *(default settings). Key:* fill-in = *fill-in during factorization;* anal., fact., solve = *times for analyze, factorize, and solve (cpu seconds).*

	`MA27`				`MA47`			
Problem	fill-in	anal.	fact.	solve	fill-in	anal.	fact.	solve
AUG2DCQP	11038	.09	.09	.01	36179	.21	.13	.04
AUG2DQP	11198	.09	.09	.01	36339	.21	.13	.03
AUG3DCQP	10797	.11	.16	.01	50401	.25	.25	.04
AUG3DQP	11997	.10	.16	.01	51601	.25	.25	.04
BLOCKQP1	2015	1.26	.05	.00	16989	1.97	.09	.02
BLOCKQP2	2015	1.27	.06	.00	16989	1.97	.09	.02
BLOCKQP3	2015	1.27	.06	.01	16982	1.97	.08	.02
GOULDQP2	1749	.01	.01	.01	2927	.03	.02	.01
GOULDQP3	2787	.02	.02	.01	3357	.32	.03	.01
HAGER2	9	.04	.03	.01	6004	.07	.04	.02
KSIP	3029	.13	.13	.01	17860	1.94	.12	.02
MINC44	2241	.01	.01	.00	1548	.03	.02	.00
MOSARQP1	6466	.04	.07	.00	29104	.11	.10	.01
NCVXQP1	12539	.13	1.01	.02	273646	1.87	30.26	.06
NCVXQP2	12539	.12	1.01	.02	241150	1.83	29.00	.06
NCVXQP3	12539	.13	.98	.02	272372	1.83	31.53	.06
NCVXQP4	8461	.07	.45	.01	110796	.43	4.00	.02
NCVXQP5	8461	.07	.45	.01	104070	.42	3.53	.03
NCVXQP6	8461	.07	.44	.01	108867	.43	3.69	.02
NCVXQP7	15913	.20	2.60	.02	404465	2.84	61.43	.08
NCVXQP8	15913	.19	2.61	.03	419779	2.89	77.55	.09
NCVXQP9	15913	.20	2.57	.03	406633	2.84	68.19	.09
QPCBOEI1	3886	.08	.03	.00	13333	.16	.05	.00
QPCBOEI2	941	.01	.01	.00	4421	.03	.02	.00
QPCSTAIR	3318	.05	.05	.00	12444	.14	.08	.01
QPNBOEI1	3886	.08	.04	.00	13333	.15	.06	.01
QPNBOEI2	941	.01	.02	.00	4421	.03	.02	.00
QPNSTAIR	3318	.05	.05	.00	12444	.14	.08	.01
SOSQP1	5003	.16	.04	.00	3001	1.25	.06	.01
UBH1	2109	.02	.01	.00	3915	.05	.02	.01

As a yardstick, all of the test examples were factorized using the Harwell codes `MA27` and `MA47`, using default settings. Of course, these codes make no effort to modify $\boldsymbol{H}$ to produce a second-order sufficient $\boldsymbol{B}$; these results are included to indicate the sort of times we consider acceptable for a good factorization and thus the sort of times that we should be aiming for in our modified factorization. The results are given in Table 5.3. We note that although `MA47` was especially designed to cope with augmented systems of the form (1.6), it is often less efficient than the general purpose method `MA27`. In its defense, we sometimes observed that `MA47` obtained accurate solutions to (1.6) while its older sister failed to do so; the `NCVXQP` problems are cases in point.

In Table 5.4, we report on the performance of the implicit modification option from our prototype code, `KKTSOL`, on the test set. For these and subsequent runs, we restrict the total number of dense rows of $\boldsymbol{A}$ and $\boldsymbol{B}$ to be at most 350, although this means that the target densities δ_a or δ_b may be exceeded. We have found that although dense matrices are processed using high-performance BLAS, this restriction often has a beneficial effect on execution times. A value of roughly 350 has been observed empirically to give a good compromise between increased dense storage and the advantages of direct addressing of data.

We make two observations. First, `KKTSOL` performs well in many cases, at least in

Table 5.4

Performance of the implicit modification variant of KKTSOL. *Key:* fill-in A, H, dense = *fill-in within A, H, and the final dense block;* dense rows A, H = *number of rows of A and H which are treated as dense;* dense rows G = *number of pseudo modifications made (dimension of G);* # mod = *number of diagonals of H actually modified;* anal., fact., solve = *times for analyze, factorize, and solve (cpu seconds).*

	Fill-in			Dense rows			#	Time		
Problem	A	H	dense	A	H	G	mod	anal.	fact.	solve
AUG2DCQP	5369	42539	61425	168	182	0	0	.31	.58	.02
AUG2DQP	5369	42539	61425	168	182	0	0	.31	.58	.01
AUG3DCQP	3860	116416	61425	198	152	0	0	.55	1.36	.03
AUG3DQP	3860	116416	61425	198	152	0	0	.55	1.36	.03
BLOCKQP1	0	10998	507528	1	8	998	1003	1.16	16.29	.06
BLOCKQP2	0	10998	45	1	8	0	5	1.17	.04	.00
BLOCKQP3	0	10998	412686	1	8	899	904	1.17	12.46	.06
GOULDQP2	348	1862	253	0	22	0	0	.03	.01	.00
GOULDQP3	348	3479	703	0	37	0	0	.05	.01	.00
HAGER2	0	990	66	0	11	0	0	.06	.01	.00
KSIP	0	190	210	0	20	0	0	.32	.04	.00
MINC44	0	187	253	19	3	0	0	.01	.01	.00
MOSARQP1	0	11703	23005	0	214	0	0	.13	.06	.01
NCVXQP1	1378	13940	208981	73	277	296	515	.64	2.85	.03
NCVXQP2	1378	13940	203841	73	277	288	507	.64	2.78	.02
NCVXQP3	1378	13940	145530	73	277	189	400	.66	1.84	.02
NCVXQP4	288	8730	345696	49	301	481	771	.26	5.37	.03
NCVXQP5	288	8730	278631	49	301	396	682	.27	4.09	.03
NCVXQP6	288	8730	194376	49	301	273	556	.27	2.68	.03
NCVXQP7	2535	16182	80200	75	219	106	259	.90	.86	.02
NCVXQP8	2535	16182	76636	75	219	97	252	.89	.84	.01
NCVXQP9	2535	16182	76245	75	219	96	237	.89	.82	.02
QPCBOEI1	4	5229	14878	147	25	0	0	.03	.08	.00
QPCBOEI2	0	1349	6903	110	7	0	0	.01	.02	.00
QPCSTAIR	687	10021	23653	186	31	0	0	.07	.17	.00
QPNBOEI1	4	5229	21115	147	25	33	33	.04	.17	.01
QPNBOEI2	0	1349	8515	110	7	13	13	.01	.04	.00
QPNSTAIR	687	10021	28441	186	31	21	21	.06	.26	.01
SOSQP1	0	997	10	1	3	0	0	.12	.01	.00
UBH1	1288	5066	9316	97	39	0	0	.06	.03	.00

comparison with MA47. Clearly, restricting the pivot order has some detrimental effect on the fill-in. This is often compensated by our not requiring further pivoting during the factorization, to correct for an inappropriate pivot sequence from the analysis phase, which sometimes hampers MA47.

Second, for the nonconvex problems, a large number of pseudo modifications is required, but many of these later turn into actual modifications. This is especially noticeable for the BLOCK and NCVXQP problems. For many of these problems, significantly more actual modifications are needed than are strictly required to counter the negative eigenvalues in the reduced Hessian, but this is difficult to avoid without having good approximations to their related eigenvectors. BLOCKQP1 and BLOCKQP3 are generalizations of Example 4.1, and, as predicted, the implicit modification scheme is slow precisely because $\boldsymbol{G}$ is large.

In Table 5.5, we consider the performance of the explicit modification variant on the test set. We first note that the alteration of the b pivot order sometimes has a slightly detrimental effect on the analysis times, but this is not significant. However, the main differences are observed on the BLOCK and QPN examples. For the former, the explicit modification scheme clearly helps. Rather than requiring the factorization of

Table 5.5

Performance of the explicit modification variant of KKTSOL. *Key:* fill-in A, H, dense = *fill-in within A, H, and the final dense block;* dense rows A, H = *number of rows of A and H which are treated as dense;* # mod = *number of diagonals of H actually modified;* anal., fact., solve = *times for analyze, factorize, and solve (cpu seconds).*

				Dense					
	Fill-in			rows		#	Time		
Problem	A	H	dense	A	H	mod	anal.	fact.	solve
AUG2DCQP	5369	56098	61425	168	182	0	.75	.89	.02
AUG2DQP	5369	56098	61425	168	182	0	.76	.89	.02
AUG3DCQP	3860	145843	61425	198	152	0	1.93	2.19	.03
AUG3DQP	3860	145843	61425	198	152	0	1.94	2.14	.04
BLOCKQP1	0	10998	45	1	8	1003	1.17	.04	.00
BLOCKQP2	0	10998	45	1	8	5	1.17	.02	.01
BLOCKQP3	0	10998	45	1	8	904	1.17	.04	.01
GOULDQP2	348	1862	253	0	22	0	.03	.00	.01
GOULDQP3	348	3479	703	0	37	0	.04	.01	.00
HAGER2	0	990	66	0	11	0	.05	.01	.00
KSIP	0	190	210	0	20	0	.32	.03	.00
MINC44	0	162	741	19	19	0	.02	.00	.00
MOSARQP1	0	11703	23005	0	214	0	.12	.07	.00
NCVXQP1	1378	13940	61425	73	277	515	.66	3.93	.01
NCVXQP2	1378	13940	61425	73	277	507	.65	3.87	.01
NCVXQP3	1378	13940	61425	73	277	397	.66	3.91	.01
NCVXQP4	288	8730	61425	49	301	769	.28	3.16	.01
NCVXQP5	288	8730	61425	49	301	682	.27	3.14	.01
NCVXQP6	288	8730	61425	49	301	554	.28	3.13	.01
NCVXQP7	2535	16182	43365	75	219	259	.91	1.70	.01
NCVXQP8	2535	16182	43365	75	219	251	.91	1.70	.00
NCVXQP9	2535	16182	43365	75	219	237	.91	1.70	.01
QPCBOEI1	4	4040	43365	147	147	0	.17	.31	.01
QPCBOEI2	0	861	24310	110	110	0	.04	.11	.00
QPCSTAIR	687	1465	61425	186	164	0	.36	.33	.01
QPNBOEI1	4	4040	43365	147	147	41	.18	17.93	.01
QPNBOEI2	0	861	24310	110	110	24	.05	4.06	.00
QPNSTAIR	687	1465	61425	186	164	81	.36	22.02	.01
SOSQP1	0	997	10	1	3	0	.12	.01	.00
UBH1	1288	5041	18915	97	97	0	.10	.06	.00

a large matrix $\boldsymbol{G}$, the factorization and update of a trivial (2 by 2) condemned matrix is performed. For the QPNBOEI1 and QPNSTAIR examples, the roles are reversed. The condemned matrices are now large (of orders 292 and 372, respectively) and the updates quite inefficient. The only difference between the QPC and QPN examples is that the former are (strictly) convex. Thus, the differences in factorization times in Table 5.5 for these examples are purely because the QPN examples form and update their condemned submatrices, while the QPC examples do not need to.

Thus, we see that both the implicit and explicit modification schemes have their advantages and disadvantages. In many cases, these methods are able to compete with the nonmodification methods, and of course the proposals here have extra functionality. However, there are clearly some instances where there is a significant overhead caused by the restriction on the allowable pivots. Thus we must conclude that, so far, we have been only partially successful in fulfilling our stated aims.

6. Conclusions and further comments. In this paper we have shown that a number of modified factorization methods for linearly constrained optimization calculations may be derived, and we have indicated that these techniques hold some promise for large-scale computations. Our next task is to complete our code for the Harwell Subroutine Library. Because this code is of general interest, we intend to release a version, KKTSOL, into the public domain. Our ultimate goal is to provide implementations of barrier function-based methods for solving general linearly constrained nonlinear optimization problems with the Harwell Subroutine Library.

An outstanding theoretical question remains. It is relatively straightforward to obtain an upper bound on the (possible) perturbation $\boldsymbol{E}$ made to $\boldsymbol{H}$. Thus so long as $\boldsymbol{x}$ remains bounded and f has a continuous Hessian, $\boldsymbol{B}$ will remain bounded. However, to use the factorization with confidence within a general linearly constrained optimization algorithm, one also needs to ensure that $\boldsymbol{B}$ is *uniformly* second-order sufficient, i.e., the constant σ in (1.5) is independent of the iteration. We have been unable to provide such a bound for the methods suggested here, nor do we believe that it is likely to be easy. This is in contrast to the method of [21], where such a bound is obtained. As we have already mentioned, the difficulty with the Forsgren–Murray approach for sparse problems is that the required pivots may all prove unacceptable from a sparsity viewpoint. It remains an open question as to whether there is a satisfactory method for sparse problems from both the theoretical and practical perspectives.

We have purposely not attempted to derive directions of sufficient negative curvature for such problems (see, for example, [21], [20], and the references contained therein), although algorithms which use them offer stronger convergence guarantees—specifically, convergence to points for which second-order necessary optimality conditions hold. We intend to investigate this possibility for large-scale problems in future.

Acknowledgments. This paper is the successor to the technical report by Arioli et al. [2]. The author is extremely grateful to Mario Arioli, Tony Chan, Iain Duff, Jacek Gondzio, Phil Gill, John Reid, and Bobby Schnabel for stimulating discussions on much of the material contained here, and to four anonymous referees for a number of helpful suggestions. He would also like to thank CERFACS for the environment and facilities which made much of this research possible.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, 2nd ed., SIAM, Philadelphia, 1995.

[2] M. Arioli, T. F. Chan, I. S. Duff, N. I. M. Gould, and J. K. Reid, *Computing a Search Direction for Large-Scale Linearly Constrained Nonlinear Optimization Calculations*, Tech. Rep. RAL-93-066, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 1993.

[3] I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *CUTE: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[4] J. R. Bunch and L. C. Kaufman, *Some stable methods for calculating inertia and solving symmetric linear equations*, Math. Comp., 31 (1977), pp. 163–179.

[5] J. R. Bunch and B. N. Parlett, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal., 8 (1971), pp. 639–655.

[6] T. J. Carpenter, I. J. Lustig, J. M. Mulvey, and D. F. Shanno, *Higher-order predictor-corrector interior point methods with application to quadratic objectives*, SIAM J. Optim., 3 (1993), pp. 696–725.

[7] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer Ser. Comput. Math. 17, Springer-Verlag, Heidelberg, Berlin, New York, 1992.

[8] R. W. Cottle, *Manifestations of the Schur complement*, Linear Algebra Appl., 8 (1974), pp. 189–211.

[9] J. E. Dennis and R. B. Schnabel, *A view of unconstrained optimization*, in Handbook of Operations Research and Management Science, vol. 1. Optimization, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, eds., North–Holland, Amsterdam, 1989, pp. 1–72.

[10] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers*, SIAM, Philadelphia, PA, 1991.

[11] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct methods for sparse matrices*, Oxford University Press, Oxford, UK, 1986.

[12] I. S. Duff, N. I. M. Gould, J. K. Reid, J. A. Scott, and K. Turner, *The factorization of sparse symmetric indefinite matrices*, IMA J. Numer. Anal., 11 (1991), pp. 181–204.

[13] I. S. Duff and J. K. Reid, *MA*27 *: A Set of Fortran Subroutines for Solving Sparse Symmetric Sets of Linear Equations*, Report R-10533, AERE Harwell Laboratory, Harwell, UK, 1982.

[14] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.

[15] I. S. Duff and J. K. Reid, *Exploiting zeros on the diagonal in the direct solution of indefinite sparse symmetric systems,* ACM Trans. Math. Software, 22 (1996), pp. 227–257.

[16] I. S. Duff, J. K. Reid, N. Munksgaard, and H. B. Neilsen, *Direct solution of sets of linear equations whose matrix is sparse, symmetric and indefinite*, J. Inst. Math. Appl., 23 (1979), pp. 235–250.

[17] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley, Chicester, New York, 1968; reprinted as Classics Appl. Math. 4, SIAM, Philadelphia, PA, 1990.

[18] R. Fletcher, *Factorizing symmetric indefinite matrices*, Linear Algebra Appl., 14 (1976), pp. 257–272.

[19] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley, Chicester, New York, 1987.

[20] A. Forsgren, P. E. Gill, and W. Murray, *Computing modified Newton directions using a partial Cholesky factorization*, SIAM J. Sci. Comput., 16 (1995), pp. 139–150.

[21] A. L. Forsgren and W. Murray, *Newton methods for large-scale linear equality-constrained minimization*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 560–587.

[22] R. Fourer and S. Mehrotra, *Solving symmetrical indefinite systems in an interior-point method for linear programming*, Math. Programming, 62 (1993), pp. 15–39.

[23] A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1981.

[24] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, *Methods for modifying matrix factorizations*, Math. Comp., 28 (1974), pp. 505–535.

[25] P. E. Gill and W. Murray, *Newton-type methods for unconstrained and linearly constrained optimization*, Math. Programming, 7 (1974), pp. 311–350.

[26] P. E. Gill, W. Murray, D. B. Ponceléon, and M. A. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 292–311.

[27] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, New York, 1981.

[28] N. I. M. Gould, *On practical conditions for the existence and uniqueness of solutions to the general equality quadratic-programming problem*, Math. Programming, 32 (1985), pp. 90–99.

[29] N. I. M. Gould, *Constructing Appropriate Models for Large-Scale, Linearly-Constrained, Nonconvex, Nonlinear, Optimization Algorithms*, Tech. Rep. RAL-TR 95-037, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 1995.

[30] Harwell Subroutine Library, *A Catalogue of Subroutines (Release* 10*)*, Advanced Computing Department, Harwell Laboratory, Harwell, UK, 1990.

[31] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *Computational experience with a primal-dual interior point method for linear programming*, Linear Algebra Appl., 152 (1991), pp. 191–222.

[32] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *On implementing Mehrotra's predictor-corrector interior-point method for linear programming*, SIAM J. Optim., 2 (1992), pp. 435–449.

[33] H. M. Markowitz, *The elimination form of the inverse and its application to linear programming*, Management Sci., 3 (1957), pp. 255–269.

[34] S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM J. Optim., 2 (1992), pp. 575–601.

[35] J. J. Moré and D. C. Sorensen, *On the use of directions of negative curvature in a modified Newton method*, Math. Programming, 16 (1979), pp. 1–20.

[36] B. A. Murtagh and M. A. Saunders, *Large-scale linearly constrained optimization*, Math. Programming, 14 (1978), pp. 41–72.

[37] T. Schlick, *Modified Cholesky factorizations for sparse preconditioners*, SIAM J. Sci. Comput., 14 (1993), pp. 424–445.

[38] R. B. Schnabel and E. Eskow, *A new modified Cholesky factorization*, SIAM J. Sci. Statist. Comput., 11 (1991), pp. 1136–1158.

[39] G. W. Stewart, *Modifying pivot elements in gaussian elimination*, Math. Comp., 28 (1967), pp. 537–542.

[40] R. J. Vanderbei and T. J. Carpenter, *Symmetrical indefinite systems for interior point methods*, Math. Programming, 58 (1993), pp. 1–32.

SIAM J. OPTIM.
Vol. 9, No. 4, pp. 1064–1081

A TRUST REGION METHOD FOR PARABOLIC BOUNDARY CONTROL PROBLEMS*

C. T. KELLEY† AND E. W. SACHS‡

To our good friend John Dennis on his 60th birthday

Abstract. In this paper we develop a trust region algorithm for constrained parabolic boundary control problems. For the computation of a trust region step we propose an iterative scheme which is a projected form of the Steihaug trust region conjugate gradient method. To ensure the good local convergence properties in the terminal phase, a smoothing step at each iteration is added. This step and the projection require the modification of the standard trust region algorithm and its convergence proof. The algorithm has sup-norm convergence in the terminal phase and L^2 convergence in the global phase. The results are illustrated for a parabolic boundary control problem.

Key words. trust region methods, inexact Newton methods, optimal control

AMS subject classifications. 49K20, 65F10, 49M15, 65J15, 65K10

PII. S1052623496308965

1. Introduction.

In this paper we show how methods that combine conjugate gradient (CG) iteration and trust region globalization for optimization problems subject to simple bounds can be applied in an infinite dimensional setting to parabolic optimal control problems. This paper addresses the global convergence questions left open by our previous work [11], [13], [14] on fast multilevel algorithms for local convergence by showing how trust region CG algorithms can solve the coarse mesh problems needed to initialize the multilevel method in an efficient and mesh-independent way. The algorithm uses the postsmoothing step from [14] and [11] to improve the performance of the iteration.

For unconstrained problems, our approach differs from [19] and [20] only in that after a step and trust region radius have been accepted, a smoothing iteration like those in [11] and [14] is attempted. Unlike our previous work, however, an Armijo [1] line search is added to the smoothing step to ensure decrease in the objective function. This new form of the smoothing step is a scaled gradient projection [2] algorithm. The local theory from [11] and [14] implies that full smoothing steps are taken near the solution.

The effect of this in the infinite dimensional case is to allow one to make sup-norm error estimates in the terminal phase of the iteration [11], [14]. In the constrained case, we differ from the algorithm in [8] in more ways. We use an L^2 trust region and solve unconstrained trust region problems using the reduced Hessian at the current point to build the quadratic model. The reason for this is to make the trust region problem as easy to solve as possible and to eliminate the need to explicitly compute

*Received by the editors September 9, 1996; accepted for publication (in revised form) May 6, 1998; published electronically September 24, 1999.

http://www.siam.org/journals/siopt/9-4/30896.html

†Department of Mathematics and Center for Research in Scientific Computation, North Carolina State University, Box 8205, Raleigh, NC 27695-8205 (Tim_Kelley@ncsu.edu). The research of this author was supported by National Science Foundation grants DMS-9321938 and DMS-9700569 and North Atlantic Treaty Organization grant CRG 920067. Computing was partially supported by an allocation of time from the North Carolina Supercomputing Center.

‡FB IV—Mathematik and Graduiertenkolleg Mathematische Optimierung, Universität Trier, 54296 Trier, Germany (sachs@uni-trier.de). The research of this author was supported by North Atlantic Treaty Organization grant CRG 920067.

a generalized Cauchy point. We update the active set after the trust region step has been computed with a scaled projected gradient step (similar to [8]). The scaling serves the purpose of becoming an inexact implementation of the algorithm in [11] and [14] when full steps are taken. We then obtain fast local convergence in the L^∞-norm (not two-step, as in the more general case [15], because our constraints are simple bounds). Our local convergence theory does not depend on identification of the active set in finitely many iterations but instead applies the measure-theoretic ideas in [14]. Hence, our trust region algorithm becomes an inexact projected Newton method in the terminal phase of the iteration with local convergence properties covered by the theory developed in [3], [11], and [14].

We consider the problem of minimizing

$$f(u) = \frac{1}{2}\int_0^1 (y(u;T,x) - z(x))^2\,dx + \frac{\alpha}{2}\int_0^T u^2(t)\,dt, \tag{1.1}$$

where $\alpha > 0$ is given and $y(t,x) = y(u;t,x)$ is the solution to the nonlinear parabolic problem

$$\begin{aligned} &y_t(t,x) = y_{xx}(t,x), \quad 0 < x < 1, \quad 0 < t < T, \\ &y(0,x) = y_0(t), \quad 0 < x < 1, \\ &y_x(t,0) = 0, \quad y_x(t,1) = g(y(t,1)) + u(t), \quad 0 < t < T. \end{aligned} \tag{1.2}$$

In (1.1)–(1.2) u is constrained to be in the set

$$\mathcal{U} = \{u \in L^\infty([0,T]) \,|\, u_{min}(t) \le u(t) \le u_{max}(t), \text{for } a.e.\ t \in [0,T]\} \tag{1.3}$$

for fixed $u_{min}, u_{max} \in L^\infty([0,T])$ and the nonlinear function g is assumed to satisfy

$$g \in C^2(R), \quad g', g'' \in L^\infty(R). \tag{1.4}$$

See [23] for examples of applications.

The gradient of f in $L^2([0,T])$ is

$$(\nabla f(u))(t) = \alpha u(t) + d(t,1), \tag{1.5}$$

where $d(t,x)$ is the solution of the adjoint problem

$$\begin{aligned} &-d_t(t,x) = d_{xx}(t,x), \quad 0 < x < 1, \quad 0 < t < T, \\ &d(T,x) = y(T,x) - z(x), \quad 0 < x < 1, \\ &d_x(t,0) = 0, \quad d_x(t,1) = g'(y(t,1))d(t,1), \quad 0 < t < T. \end{aligned} \tag{1.6}$$

The map $u \to d(t,1)$ is completely continuous as a map on $C[0,T]$ and as a map from $L^q[0,T]$, $2 \le q \le \infty$, to $C[0,T]$ [18]. The fact that the mapping from the control to the gradient is the sum of a multiple of the identity and a completely continuous map is essential for the convergence analysis in this paper because the compactness of the map $u \to d(t,1)$ ensures that the performance of the CG iteration is independent of the discretization. This is consistent with results on linear equations and CG (see [9] and [22]). Another benefit of this compactness is that the reduced Hessian of f is a compact perturbation of a constant multiple of the identity and hence no preconditioning is needed for fast convergence. However, as the regularization parameter $\alpha > 0$ becomes small, the performance of CG will deteriorate, even near an optimal point.

The work in this paper is restricted to one space dimension because we use the regularity results from [18], which are valid only in one dimension. However, more general equations, say, with linear convection terms or nonlinear convection terms that are sufficiently smooth, are covered by the theory of this paper.

We base our methods on the work in [7], [8], and [19] (where $u_{max} = +\infty$ and $u_{min} = -\infty$ for unconstrained problems). These methods solve the trust region problem by searching along the piecewise linear path having the CG iterates as nodes, terminating either on the trust region boundary, with an inexact Newton step, or with a direction of negative curvature.

We close this section with some notation and definitions. Let $(\cdot, \cdot)$ denote the inner product in L^2 or the Euclidean inner product in any finite dimensional space. We denote the L^2-norm by $\| \cdot \|_2$ and the L^∞-norm by $\| \cdot \|_\infty$.

We let $\mathcal{P}$ be the L^2 projection onto $\mathcal{U}$ defined for any measurable u on $[0, T]$ and almost every $t \in [0, T]$ as

$$(\mathcal{P}u)(t) = \begin{cases} u_{min}(t) & \text{if } u(t) \leq u_{min}(t), \\ u(t) & \text{if } u_{min}(t) < u(t) < u_{max}(t), \\ u_{max}(t) & \text{if } u(t) \geq u_{max}(t). \end{cases} \tag{1.7}$$

We define

$$F(u)(t) = u(t) - \mathcal{P}(u(t) - \nabla f(u)(t)). \tag{1.8}$$

The nonsmooth nonlinear equation $F(u) = 0$ is a necessary condition for stationarity [2].

The map $\mathcal{K}_0$, given by

$$\mathcal{K}_0(u) = -\alpha^{-1} d(t, 1) = u - \alpha^{-1} \nabla f(u), \tag{1.9}$$

is a completely continuous map from $L^q[0, T] \to C[0, T]$ for any $1 \leq q \leq \infty$ [11], [14].

For $u \in \mathcal{U}$, we define the active set for u as

$$\begin{aligned} \mathcal{A}(u) \quad &= \quad \{t \,|\, u(t) = u_{max},\ \mathcal{K}_0(u)(t) \geq u_{max}(t)\} \\ &\cup \quad \{t \,|\, u(t) = u_{min},\ \mathcal{K}_0(u)(t) \leq u_{min}(t)\} \end{aligned} \tag{1.10}$$

and the inactive $\mathcal{I}(u)$ set as $[0, T] \setminus \mathcal{A}(u)$. It is clear that for any $\lambda > 0$

$$u(t) = \mathcal{P}(u(t) - \lambda \nabla f(u)(t)) \text{ for all } t \in \mathcal{A}(u). \tag{1.11}$$

2. Algorithms.

All the algorithms are based on the trust region CG method in [19] and the general convergence analysis in [21]. The trust region problem is solved approximately by using a piecewise linear path whose nodes are the CG iterates. This approximate solution of the trust region problem is used in a standard way [10], [19], [17] to test for sufficient decrease and adjust the trust region radius. We incorporate the trust region CG method into the inexact projected Newton approach of [11] to give a superlinearly convergent algorithm.

2.1. Inexact projected Newton algorithm.

To specify the algorithm we must define projections that correspond to the active and inactive set. For any measurable $S \subset [0, T]$ we define the multiplication operator $\mathcal{P}_S$ by

$$\mathcal{P}_S u(t) = \chi_S(t) u(t), \tag{2.1}$$

where χ_S is the characteristic function of S. In particular, if $u \in \mathcal{U}$ and $\mathcal{A}$ and $\mathcal{I}$ are approximations to $\mathcal{A}(u)$ and $\mathcal{I}(u)$, we will use

$$\mathcal{P}_{\mathcal{A}} w(t) = \chi_{\mathcal{A}}(t) w(t) \text{ and } \mathcal{P}_{\mathcal{I}} w(t) = \chi_{\mathcal{I}}(t) w(t). \tag{2.2}$$

We follow [14] and [11] and approximate the active set by

$$\begin{aligned} \mathcal{A} = \mathcal{A}_\epsilon(u) \quad &= \quad \{t \,|\, u(t) = u_{max},\, \mathcal{K}_0(u)(t) \geq u_{max}(t) + \epsilon\} \\ &\cup \quad \{t \,|\, u(t) = u_{min},\, \mathcal{K}_0(u)(t) \leq u_{min}(t) - \epsilon\} \end{aligned} \tag{2.3}$$

and let

$$\mathcal{I} = \mathcal{I}_\epsilon(u) = [0, T] \setminus \mathcal{A}_\epsilon(u).$$

The parameter $\epsilon > 0$ may be adjusted as the iteration progresses to give local superlinear convergence [11], [14].

Note that for all $\epsilon > 0$ we have

$$\mathcal{A}_\epsilon(u) \subset \mathcal{A}(u)$$

and hence

$$u(t) = \mathcal{P}(u(t) - \lambda \nabla f(u)(t)) \text{ for all } \lambda > 0 \text{ and } t \in \mathcal{A}_\epsilon(u). \tag{2.4}$$

In the constrained case, the necessary conditions for optimality can be expressed as a nondifferentiable compact fixed point problem

$$u = \mathcal{K}(u), \tag{2.5}$$

where

$$\mathcal{K}(u) = \mathcal{P}(\mathcal{K}_0(u)).$$

Recall from section 1 that the map $\mathcal{K}_0$ (and hence $\mathcal{K}$) is a compact map from $L^q[0, T]$ to $C[0, T]$ for any $1 \leq q \leq \infty$. In that sense $\mathcal{K}$ is a smoother. We will use that property to (1) improve the global convergence properties of our proposed algorithm and (2) provide a uniform norm local convergence theory as in [14] and [11].

We define the reduced Hessian $\mathcal{R}(u_c)$ at u_c by

$$\mathcal{R}(u_c) = \mathcal{P}_{\mathcal{A}} + \mathcal{P}_{\mathcal{I}} \nabla^2 f(u_c) \mathcal{P}_{\mathcal{I}} \tag{2.6}$$

with $\mathcal{I} = \mathcal{I}_\epsilon(u)$.

The inexact projected Newton algorithms from [11] and [14] have several stages. We describe the one from [11] here in terms of the transition from a current approximation u_c to a new approximation u_+. The understanding here is that the parameter ϵ in the approximation to the active set $\mathcal{A}_\epsilon(u_c)$ and the forcing term η in the inexact Newton process change as the iteration progresses.

Algorithm 2.1. `pnstep`$(u_c, u_+, f, \epsilon_c, \eta_c)$

1. **Identification:** *Given u_c and ϵ_c set $\mathcal{I} = \mathcal{I}_\epsilon(u_c)$.*
2. **Error Reduction:** *Find $s \in Range \mathcal{P}_{\mathcal{I}}$ which satisfies*

$$\|\mathcal{R}(u_c)s + \mathcal{P}_{\mathcal{I}} \nabla f(u_c)\|_2 < \eta_c \|\mathcal{P}_{\mathcal{I}} \nabla f(u_c)\|_2. \tag{2.7}$$

Set

$$u_{1/2} = \mathcal{P}(u_c + s)$$

to reduce the error in L^2.

3. **Postsmoothing:** *Set* $u_+ = \mathcal{K}(u_{1/2})$ *to recover convergence* $C[0,T]$.

The effectiveness of the postsmoothing step depends on the regularization parameter α. As α is reduced so is the radius of the ball about a solution u^* for which Algorithm `pnstep` produces an improved approximation to the solution. Our global strategy must take this into account.

In the context of this paper, in which global convergence is the issue, Algorithm `pnstep` presents two problems. First, the smoothing step is a scaled gradient projection step and may lead to dramatic increases in the objective function when u_c is far from the solution. We remedy this by adding an Armijo line search to this phase of the algorithm but do not demand $f(u_+) < f(u_{1/2})$, which may never be possible, but only that $f(u_+) < f(u_c)$ by a certain small amount. The results in [11] and [14] ensure that if u_c is sufficiently near the solution, then the full smoothing step will be accepted and hence the fast local convergence (the precise speed of convergence depends on the choice of forcing term η) will not be affected by the line search. Second, there is no guarantee that the reduced Hessian will be positive definite. We address this problem with an inexact trust region algorithm that will exploit any negative curvature direction that it finds.

As is standard we use the measure of nonstationarity

$$\sigma(u) = \|u - \mathcal{P}(u - \nabla f(u))\|_2 = \|F(u)\|_2. \tag{2.8}$$

This is used not only to decide on termination on the nonlinear iteration but also, as it was used in [14] and [11], to construct the tolerance for the linear inner iteration and to construct the approximate active and inactive sets.

For example, a locally convergent algorithm using Algorithm `pnstep` is the following.

```
ALGORITHM 2.2. pnlocal($u, f, \epsilon_0, \eta_0, \sigma_0$)
  1. If $\sigma(u) \le \sigma_0$, terminate the iteration.
  2. $\epsilon = \min(\epsilon_0, \sigma(u)^{1/2})$, $\eta = \min(\eta_0, \sigma(u)^{1/2})$.
  3. Take an inexact step pnstep($u, u_+, f, \epsilon, \eta$).
  4. $u = u_+$. Go to step 1.
```

The values of η and ϵ in step 2 will ensure superlinear convergence with q-order 5/4 [11], [14]. There is nothing special about the exponent 1/2 used to define ϵ and η. These concrete formulae are used only for an illustration.

Under standard assumptions [14], [11] Algorithm `pnlocal` will produce iterates that converge locally q-superlinearly (in the L^∞-norm) to a minimizer. q-linear convergence can be obtained if the formula for η in step 2 is replaced by $\eta = \eta_0$ and η_0 is sufficiently small. The purpose of this paper is to develop a trust region globalization for this algorithm that preserves the L^∞-norm local convergence in the terminal phase while converging globally in L^2.

2.2. Solution of the trust region problem.

We use a standard solver form [19] for our unconstrained trust region subproblems. The inputs to Algorithm `trcg`, which approximately solves the trust region problem, are the current point u, the objective f, a preconditioner M (which we will set to I), the forcing term η, the current trust region radius Δ, and a limit on the number of iterations $kmax$. The output is the approximate solution of the trust region problem d. We formulate the algorithm using the preconditioned CG framework from [12].

We will assume for the present that gradients are computed exactly and that

Hessian-vector product $\nabla^2 f(u)w$ is approximated by either a forward difference

$$D_h^2 f(u:w) = \begin{cases} 0, & w=0, \\ \dfrac{\nabla f(u+h\|u\|_2 w/\|w\|_2) - \nabla f(u)}{h\|u\|_2/\|w\|_2}, & w,u \neq 0, \\ \dfrac{\nabla f(hw/\|w\|_2) - \nabla f(0)}{h/\|w\|_2}, & u=0, w\neq 0, \end{cases} \tag{2.9}$$

or a centered difference

$$D_h^2 f(u:w) = \begin{cases} 0, & w=0, \\ \dfrac{\nabla f(u+h\|u\|_2 w/\|w\|_2) - \nabla f(u-h\|u\|_2 w/\|w\|_2)}{2h\|u\|_2/\|w\|_2}, & w,u \neq 0, \\ \dfrac{\nabla f(hw/\|w\|_2) - \nabla f(-hw/\|w\|_2)}{2h/\|w\|_2}, & u=0, w\neq 0. \end{cases} \tag{2.10}$$

We found that the additional accuracy in the centered difference gradient gave much better descent directions and was worth the expense. One reason for this may be that $D_h^2 f(u:w)$ is nonlinear in w. Using this as the approximation to the Hessian-vector product in the CG iteration is equivalent to applying that algorithm to a matrix that is a difference approximation of $\nabla^2 f$ based on the Krylov subspace basis [12]. This approximating matrix, while accurate up to the truncation error of the difference scheme, will not, in general, be symmetric. A more accurate difference will reduce the effects of that loss of symmetry.

We present the algorithm of [19] for approximate solution of the trust region problem

$$\min_{\|d\|_2\leq\Delta} (g,d) + .5(d,Bd).$$

```
ALGORITHM 2.3. trcg($d, u, g, B, M, \eta, \Delta, kmax$)
 1. $r = -g$, $\rho_0 = \|r\|_2^2$, $k = 1$, $d = 0$
 2. Do While $\sqrt{\rho_{k-1}} > \eta\|g\|_2$ and $k < kmax$
     (a) $z = Mr$
     (b) $\tau_{k-1} = (z, r)$
     (c) if $k = 1$ then $\beta = 0$ and $p = z$
         else
         $\beta = \tau_{k-1}/\tau_{k-2}$, $p = z + \beta p$
     (d) $w = Bp$
         If $(p, w) \leq 0$ then
         Find $\tau$ such that $\|d + \tau p\|_2 = \Delta$
         $d = d + \tau p$; return
     (e) $\alpha = \tau_{k-1}/(p, w)$
     (f) $r = r - \alpha w$
     (g) $\rho_k = (r, r)$
     (h) $\hat{d} = d + \alpha p$
     (i) If $\|\hat{d}\|_2 > \Delta$ then
         Find $\tau$ such that $\|d + \tau p\|_2 = \Delta$
         $d = d + \tau p$; return
     (j) $d = \hat{d}$; $k = k + 1$
```

Trust region algorithms for bound constrained problems have been analyzed in considerable generality in [7]. A concrete algorithm, proposed in [8], follows a piecewise linear path in a search for a generalized Cauchy point, freezes the active set at that point, and then solves the trust region problem approximately on the current active set. This process is important for the theory in [7] not only because it guarantees Cauchy decrease but also for the proof of superlinear convergence after the active set has been identified.

In the problems considered here, where there is a continuum of constraints, it is not clear how to use the method of [8] because the active set, being uncountable, will never be fully identified, and the construction of a path on which to search for a Cauchy point would lead to infinitely many knots to test. Instead we solve an unconstrained trust region problem for a reduced quadratic model and project the solution of that problem onto the active set.

Our approach to minimization of the reduced quadratic model also differs from that in [8]. In that paper, and in the convergence analysis in [7], the fact that all norms in finite dimension are equivalent was used to justify l^∞ trust region bounds. We use the standard L^2 trust region and therefore do not include the constraints explicitly in the trust region. We then use a smoothing step to deal with the nonequivalence of norms and recover fast uniform convergence in the terminal phase of the iteration.

Given $u_c \in \mathcal{U}$ and ϵ we consider the reduced quadratic model

$$m_c(u) = f(u_c) + (\mathcal{P}_{\mathcal{I}}\nabla f(u_c), u - u_c) + (u - u_c, \mathcal{P}_{\mathcal{I}}\mathcal{R}(u_c)\mathcal{P}_{\mathcal{I}}(u - u_c))/2. \tag{2.11}$$

In (2.11), the reduced Hessian $\mathcal{R}$ is given by (2.6). Note that the action of $\mathcal{R}(u)$ on a function can easily be computed by differences. This is a somewhat nonstandard model in that $\mathcal{P}_{\mathcal{I}}\nabla f(u)$ is used in the first order part of (2.11) rather than $\nabla f(u)$. If, however, u is the minimizer of the quadratic model in the trust region, then

$$\mathcal{P}_{\mathcal{I}}(u - u_c) = u - u_c$$

and hence u is also the minimizer of

$$f(u_c) + (\mathcal{P}_{\mathcal{I}}\nabla f(u_c), u - u_c) + (u - u_c, \mathcal{R}(u_c)(u - u_c))/2.$$

Hence, using the projection of $\mathcal{R}(u_c)$ onto the inactive set has only the effect of restricting the trust region step to the inactive set.

The reasons for this are that this model performed better in our numerical experiments and also makes a smoother transition to a fast local algorithm that can be analyzed with the ideas from [11] and [14]. The nonstandard quadratic term presents no problems; however, the linear term must be accounted for in the analysis. As we shall show next, this is easy to do because our algorithm is a dog leg.

As we shall see, the analysis of the global convergence is not affected by the linear term because, in view of (2.4), the trust region direction produced by our model is, as is the case with other models, simply the projected gradient direction if the trust region radius is small enough. To see this note that if the trust region radius is sufficiently small the solution of the trust region subproblem is

$$u = \mathcal{P}(u_c - \lambda \mathcal{P}_{\mathcal{I}}\nabla f(u_c))$$

for some $\lambda > 0$. However, since $\epsilon > 0$, we must have

$$\mathcal{P}(u_c - \lambda \mathcal{P}_{\mathcal{A}}\nabla f(u_c)) = 0 \tag{2.12}$$

and therefore

$$u = \mathcal{P}(u_c - \lambda \nabla f(u_c)),$$

a projected gradient step.

Algorithm `boxtr` returns a trial point u_t by using Algorithm `trcg` for the reduced quadratic model.

```
ALGORITHM 2.4. boxtr($d, u_c, u_t, g, M, \epsilon, \eta, \Delta, kmax$)
  1. Compute $\mathcal{I} = \mathcal{I}_\epsilon(u_c)$.
  2. Find the direction $d$ by calling
     trcg($d, u_c, \mathcal{P}_\mathcal{I} \nabla f(u_c), \mathcal{R}(u_c), M, \eta, \Delta, kmax$).
  3. $u_t = \mathcal{P}(u_c + d)$.
```

Having computed the trial point, one must decide whether to accept the new point or to change the trust region radius. Both decisions are based on a comparison of the actual reduction

$$ared = f(u_t) - f(u_c) \tag{2.13}$$

to a predicted reduction based on the reduced quadratic model. Here

$$pred = ((u_t - u_c), \mathcal{P}_\mathcal{I} \nabla f(u_c)) + ((u_t - u_c), \mathcal{R}(u_c)(u_t - u_c))/2. \tag{2.14}$$

In a typical trust region algorithm, the step is accepted if

$$\frac{ared}{pred} \geq \mu_1, \tag{2.15}$$

the trust region radius is reduced ($\Delta \rightarrow \omega_1 \Delta, \omega_1 < 1$) if

$$\frac{ared}{pred} < \mu_2, \tag{2.16}$$

and the trust region radius is increased ($\Delta \rightarrow \omega_2 \Delta, \omega_2 > 1$) if

$$\frac{ared}{pred} \geq \mu_3. \tag{2.17}$$

Here $\mu_1 \leq \mu_2 < \mu_3 < 1$. For example, in [8], $\mu_1 = \mu_2 = .25$ and $\mu_3 = .75$. We must add other conditions to our trust region management scheme to account for the possibility that $ared > 0$, i.e., the quadratic model is not reduced, which may happen because of our particular choice of model if the trust region radius is too large. For technical reasons, we test for sufficient decrease in the function before accepting the trust region–step combination. We require that for some $\mu_0 \in (0, \mu_1)$

$$f(u_t) - f(u_c) \leq -\mu_0 \sigma(u_c) \|u_c - \mathcal{P}(u_c - \hat{\lambda}_c \mathcal{P}_\mathcal{I} \nabla f(u_c))\|_2, \tag{2.18}$$

where

$$\hat{\lambda}_c = \min\left(\frac{\Delta}{\|\mathcal{P}_\mathcal{I} \nabla f(u_c)\|_2}, 1\right). \tag{2.19}$$

2.3. Termination.

No algorithm that depends on a measurement of decrease like *ared* is reliable if the decreases in the function are smaller than the accuracy with which the function is computed. Once we have resolved a local minimum to that point, our view is that the iteration has succeeded.

Hence we terminate the algorithm if either $\sigma(u) < \tau_g$ or

$$|ared| < \tau_f. \tag{2.20}$$

Here τ_g and τ_f are small tolerances. We test for (2.20) every time the trust region radius is changed, and if (2.20) holds at any point during an iteration, we terminate and accept the previous iteration. The stopping criterion (2.20) is an algorithmic detail and is used only in the implementation, not the analysis.

2.4. The complete algorithm.

The notation is that u_c is the current iteration. On exit from the trust region phase of the algorithm, we obtain $u_{1/2}$ and pass that intermediate iterate to the smoother to produce u_+. It is possible that $f(u_+) > f(u_{1/2})$; however, we do not permit $f(u_+) > f(u_c)$ and we use a line search to ensure that

$$f(u_+) - f(u_{1/2}) < -\mu_4 ared = \mu_4(f(u_c) - f(u_{1/2}))$$

for some $\mu_4 \in (0, 1)$. μ_4 is yet another trust region parameter. The line search reduces the smoothing step by a factor of α if the full step fails and then by a constant factor of $\beta \in (0, 1)$.

In the interest of clarity, we do not make the trust region algorithmic parameters, μ_0, μ_1, μ_2, μ_3, μ_4, ω_1, ω_2, *kmax*, the preconditioner M (which is the identity for us), and the initial trust region radius Δ formal arguments to the algorithm. The trust region radius is limited to a maximum value Δ_{max}.

The inputs are $u \in \mathcal{U}$, the bounds, and the function f.

```
ALGORITHM 2.5. trmin($u, f, u_{max}, u_{min}$)
1. Initialize $\Delta$, $k = 1$.
2. Test for termination
   if $\sigma(u) < \tau_g$ or
   if $k > 1$ and $|ared| < \tau_f$ terminate successfully.
3. Fix $\eta$ and $\epsilon$ for this iterate. Set $u_c = u$, $rflag = 0$.
4. Find a new trial point $u_t$ using Algorithm boxtr.
5. Set $\rho = ared/pred$.
   (a) if $\rho < \mu_1$ or (2.18) does not hold, then $\Delta = \omega_1\Delta$; $rflag = 1$, go to
       step 4.
   (b) if $\mu_1 \le \rho < \mu_2$, $\Delta = \omega_1\Delta$, $u_{1/2} = u_t$
   (c) if $\rho \ge \mu_2$ and $\Delta = \Delta_{max}$, $u_{1/2} = u_t$
   (d) if $\mu_2 \le \rho < \mu_3$, or $\rho \ge \mu_3$ and $rflag = 1$, $u_{1/2} = u_t$
   (e) if $\rho \ge \mu_3$ and $rflag = 0$, $\Delta = \min(\Delta_{max}, \omega_2\Delta)$, go to step 4
6. (a) Find the smallest integer $m \ge 0$ such that

           $f(\mathcal{P}(u_{1/2} - \xi_m\alpha^{-1}\nabla f(u_{1/2}))) - f(u_{1/2}) < -\mu_4 ared$,

       where $\xi_0 = 1$, $\xi_1 = \alpha$ and $\xi_m = \beta\xi_{m-1}$ for $m \ge 2$, then postsmooth, i.e.,
       set

           $u_+ = \mathcal{P}(u_{1/2} - \xi_m\alpha^{-1}\nabla f(u_{1/2}))$.

   (b) $u = u_+$; $k = k + 1$; go to step 2.
```

The flag $rflag$ is used to avoid an infinite loop of increasing and decreasing the trust region radius.

In the context of a globally convergent algorithm, attention must be paid to the postsmoothing step 6(a). The line search prevents divergence in the early phase of the iteration when the approximate solutions are not accurate.

3. Convergence results.

In this section we derive global convergence results for Algorithm `trmin`. Recall that our notation is that u_c (resp., u_k) is the current (resp., kth) iteration. On exit from the trust region phase of the algorithm, we obtain u_t (resp., $u_{k+1/2}$) and pass that intermediate iterate to the smoother to produce u_+ (resp., u_{k+1}).

3.1. Global convergence.

Given $u_c \in H$ and the quadratic model function m_c from (2.11) we must first show that the trust region radius can be bounded from below.

Our assumptions are as follows.

Assumption 3.1.

1. f is twice continuously differentiable in $\mathcal{U}$.
2. There is $r > 0$ such that for all $u \in \mathcal{U}$ and $z \in L^2[0,T]$

$$\|z\|_2^2/r \leq |(z, \mathcal{R}(u)z)| \leq r\|z\|_2^2$$

and $\|\nabla^2 f(u)\|_2 \leq r$.

The second assumption is a sort of second order sufficiency condition as it is common in the convergence analysis of higher order methods. The finite difference approximation (2.10) satisfies this assumption for h sufficiently small, if the assumption holds for the original Hessian $\mathcal{R}$.

In this section we will use some notation from [2]. For $u \in \mathcal{U}$ define

$$u(\lambda) = \mathcal{P}(u - \lambda \nabla f(u)). \tag{3.1}$$

We begin with the lower bound for the gradient projection step from [2].

Theorem 3.1. *Let Assumption* 3.1 *hold and let* $\mu \in (0,1)$. *Then there is* λ_{max} *such that for any* $0 < \lambda \leq \lambda_{max}$ *and* $u \in \mathcal{U}$,

$$f(u) - f(u(\lambda)) \geq \frac{\mu}{\lambda}\|u - u(\lambda)\|_2^2. \tag{3.2}$$

We will also use a lemma from convex analysis. We state this as a special case of a result from [21].

Lemma 3.2. *Let* $\lambda > 0$, $u \in \mathcal{U}$, *and* f *be differentiable. Then*

$$\begin{aligned} &\|u - u(\lambda)\|_2 \textit{ is an increasing function of } \lambda, \\ &\lambda^{-1}\|u - u(\lambda)\|_2 \textit{ is a decreasing function of } \lambda, \textit{ and} \\ &(\nabla f(u), u - u(\lambda)) \geq \|u - u(\lambda)\|_2^2/\lambda. \end{aligned} \tag{3.3}$$

From Theorem 3.1 we compute a lower bound on the trust region radius.

Theorem 3.3. *Let Assumption* 3.1 *hold. Let* $\mathcal{A}$ *be computed from* (1.10) *with* $\epsilon > 0$. *Then there is* $C > 0$ *such that on exit from step* 5 *of Algorithm* `trmin`

$$\Delta \geq \Delta_{min} = C\|\mathcal{P}_{\mathcal{I}} \nabla f(u_c)\|_2. \tag{3.4}$$

Proof. The idea of the proof is to show that Algorithm `trmin` will, in the worst case of a small trust region radius, take steps of the form

$$s = \mathcal{P}(u_c - \lambda \mathcal{P}_{\mathcal{I}} \nabla f(u_c)) - u_c = \mathcal{P}(u_c - \lambda \nabla f(u_c)) - u_c.$$

The second equality follows from $\epsilon > 0$ (see (2.12)). Hence, the algorithm will therefore behave like the gradient projection algorithm [2], since by (1.11) $\mathcal{P}_{\mathcal{A}} s = 0$. In view of this, we can use known properties of the gradient projection algorithm to bound the trust region radius from below.

Algorithm `boxtr` will return a trial point of the form

$$u_t = \mathcal{P}(u_c - \mathcal{P}_{\mathcal{I}} \lambda \nabla f(u_c))$$

if no CG iterations are needed (i.e., the minimizer of the m_c in the direction $\mathcal{P}_{\mathcal{I}} \nabla f(u_c)$ is outside the trust region or if $\nabla f(u_c)$ is a direction of negative curvature for $\mathcal{P}_I \mathcal{R}(u_c)$). In the former case,

$$\alpha_0 = \|\mathcal{P}_{\mathcal{I}} \nabla f(u_c)\|_2^2 / (\mathcal{P}_{\mathcal{I}} \nabla f(u_c), \mathcal{P}_{\mathcal{I}} \mathcal{R}(u_c) \mathcal{P}_{\mathcal{I}} \nabla f(u_c)) > 0$$

satisfies

$$\alpha_0 \|\mathcal{P}_{\mathcal{I}} \nabla f(u_c)\|_2 \geq \Delta. \tag{3.5}$$

In this case

$$u_t = u_c(\lambda_t) = \mathcal{P}(u_c - \lambda_t \mathcal{P}_{\mathcal{I}} \nabla f(u_c)) = \mathcal{P}(u_c - \lambda_t \nabla f(u_c)), \tag{3.6}$$

where $\lambda_t \leq \alpha_0$ is such that

$$\lambda_t = \Delta / \|\mathcal{P}_{\mathcal{I}} \nabla f(u_c)\|_2. \tag{3.7}$$

By Assumption 3.1, $|\alpha_0| \geq 1/r$. Hence, if $\Delta < \|\mathcal{P}_{\mathcal{I}} \nabla f(u_c)\|_2 / r$, then $\Delta < \|\mathcal{P}_{\mathcal{I}} \nabla f(u_c)\|_2 \alpha_0$. Since (3.5) holds, u_t will be given by (3.6), (3.7) holds, and $\lambda_t = \Delta / \|\mathcal{P}_{\mathcal{I}} \nabla f(u_c)\|_2 < 1/r \leq 1$.

Now if we set $\mu = \mu_1$ in Theorem 3.1 and

$$\Delta < \|\mathcal{P}_{\mathcal{I}} \nabla f(u_c)\|_2 \min(\lambda_{max}, 1/r), \tag{3.8}$$

then the above remarks, Theorem 3.1, and Lemma 3.2 imply that

$$ared \leq \frac{-\mu_1}{\lambda_t} \|u_c - u_c(\lambda_t)\|_2^2 \leq -\mu_1 \|u_c - u_c(\lambda_t)\|_2 \sigma(u_c). \tag{3.9}$$

Since, by (3.7) and (2.19),

$$\lambda_t = \Delta / \|\mathcal{P}_{\mathcal{I}} \nabla f(u_c)\|_2 \geq \hat{\lambda}_c = \min\left(\frac{\Delta}{\|\mathcal{P}_{\mathcal{I}} \nabla f(u_c)\|_2}, 1 \right),$$

we obtain from Lemma 3.2

$$\|u_c - u_c(\lambda_t)\|_2 \geq \|u_c - u_c(\hat{\lambda}_c)\|_2.$$

Therefore, since $0 < \mu_0 < \mu_1$, from (3.9) we obtain

$$ared \leq -\mu_0 \sigma(u_c) \|u_c - u_c(\hat{\lambda}_c)\|_2,$$

which is (2.18).

We summarize the analysis so far. We have shown that if (3.8) holds, then $u_t = u_c(\lambda_t)$ for some λ_t and (2.18) holds.

We now show how the upper bound (3.8) on Δ can be reduced to imply that $ared/pred \geq \mu_3$. Assume that (3.8) holds. By Lemma 3.2 and the fact that $u_t - u_c = \mathcal{P}_{\mathcal{I}}(u_t - u_c)$ we have

$$(\mathcal{P}_{\mathcal{I}}\nabla f(u_c), u_t - u_c) = (\nabla f(u_c), u_t - u_c) \leq -\|u_c - u_t\|_2^2/\lambda_t$$

and hence by the definition (2.14) of *pred*

$$pred = m_c(u_t) - f(u_c) = ((u_t - u_c), \mathcal{P}_{\mathcal{I}}\nabla f(u_c)) + ((u_t - u_c), \mathcal{P}_{\mathcal{I}}\mathcal{R}(u_c)\mathcal{P}_{\mathcal{I}}(u_t - u_c))/2,$$

we have

$$\begin{aligned} pred &\leq (\mathcal{P}_{\mathcal{I}}\nabla f(u_c), u_t - u_c) + r\|u_c - u_t\|_2^2/2 \\ &\leq (r/2 - 1/\lambda_t)\|u_c - u_t\|_2^2. \end{aligned} \tag{3.10}$$

Note that

$$|ared - pred| = |f(u_t) - m_c(u_t)| \leq r\|u_c - u_t\|_2^2$$

and

$$|pred| \geq |r/2 - 1/\lambda_t|\|u_c - u_t\|_2^2.$$

Since $\|u_t - u_c\|_2 \leq \Delta$ we have

$$\begin{aligned} \left|\frac{ared}{pred} - 1\right| &= \left|\frac{ared - pred}{pred}\right| \leq \frac{r\|u_c - u_t\|_2^2}{|r/2 - 1/\lambda_t|\|u_c - u_t\|_2^2} \\ &= \frac{r}{1/\lambda_t - r/2} \leq \frac{r}{\|\mathcal{P}_{\mathcal{I}}\nabla f(u_c)\|_2/\Delta - r/2}. \end{aligned} \tag{3.11}$$

Hence, if

$$\Delta \leq \|\mathcal{P}_{\mathcal{I}}\nabla f(u_c)\|_2/(r/2 + r/(1 - \mu_3)),$$

then

$$\frac{ared}{pred} \geq 1 - \left|\frac{ared}{pred} - 1\right| \geq 1 - \frac{r}{\|\mathcal{P}_{\mathcal{I}}\nabla f(u_c)\|_2/\Delta - r/2} \geq \mu_3.$$

Thus, if the trust region radius ever satisfies

$$\Delta \leq \min\left(\|\mathcal{P}_{\mathcal{I}}\nabla f(u_c)\|_2 \min(\lambda_{max}, 1/r), \|\mathcal{P}_{\mathcal{I}}\nabla f(u_c)\|_2(r/2 + r/(1 - \mu_3))^{-1}\right)/\omega_2,$$

where $\omega_2 > 1$ is the factor by which the trust region radius will be increased if $\frac{ared}{pred} \geq \mu_3$, then all acceptance tests will be passed and an increase will be attempted. Hence, on exit from step 5, (3.4) will hold with

$$C = \min(\lambda_{max}, 1/r, (r/2 + r/(1 - \mu_3))^{-1})/\omega_2,$$

which completes the proof. $\square$

Our main global convergence result is as follows.

Theorem 3.4. *Let Assumption* 3.1 *hold and let* u_k *be the sequence produced by Algorithm* `trmin`. *Then*

$$\lim_{k\to\infty} \sigma(u_k) = 0.$$

Proof. Step 6 of Algorithm `trmin`, (2.13), and (2.18) yield

$$\begin{aligned} f(u_{k+1}) - f(u_k) &= f(u_{k+1}) - f(u_{k+\frac{1}{2}}) + f(u_{k+\frac{1}{2}}) - f(u_k) \\ &\leq (1-\mu_4)(f(u_{k+\frac{1}{2}}) - f(u_k)) \\ &\leq -(1-\mu_4)\mu_0\sigma(u_k)\|u_k - u_k(\hat{\lambda}_k)\|_2 \end{aligned}$$

with $\hat{\lambda}_k$ defined by (2.19). The boundedness from below of f on $\mathcal{U}$ implies

$$\lim_{k\to\infty} \sigma(u_k)\|u_k - u_k(\hat{\lambda}_k)\|_2 = 0.$$

Since $\hat{\lambda}_k \leq 1$ by (2.19), Lemma 3.2 yields

$$\lim_{k\to\infty} \|u_k - u_k(\hat{\lambda}_k)\|_2 = 0.$$

Theorem 3.3 implies that either $\hat{\lambda}_k = 1$ or

$$\hat{\lambda}_k = \frac{\Delta_k}{\|\mathcal{P}_{\mathcal{I}}\nabla f(u_k)\|_2} \geq C.$$

Therefore, with $C_1 = \min(1, C)$, Lemma 3.2 implies that

$$\|u_k - u_k(\hat{\lambda}_k)\|_2^2 \geq \|u_k - u_k(C_1)\|_2^2 \geq C_1^2\sigma(u_k)^2,$$

completing the proof. □

3.2. Local convergence. In [14] we gave the conditions under which Algorithm `pnlocal` converges locally q-superlinearly to a solution u^*. This will imply fast local convergence if the iteration has a limit point that is a local minimum that satisfies the assumptions that we will outline in section 3.2.1.

To begin, note that if u_t (in the language of Algorithm `trmin`) is sufficiently near a fixed point of $\mathcal{K}$ in the L^2-norm, then a full smoothing step (i.e., $m = 0$) in step 6(a) of Algorithm `trmin` will be taken and then u_+ will be near u^* in the L^∞ sense. Hence once u_k is L^2 near to u^*, u_{k+1} will be close in the L^∞ sense and then the active and inactive sets will be accurately approximated. This will be important for the local convergence result as it was in [11] and [14].

In particular, the reduced Hessians will converge to the reduced Hessian at the solution. Therefore, using the methods in [19], if u_{k+1} is sufficiently near u^* in the L^∞-norm, then the approximate reduced Hessian is symmetric and positive definite and the trust region radius will increase if necessary so that both u^* and the minimizer of the local quadratic model are in the trust region. This fact allows us to apply the techniques in [11] and [14] to prove local convergence and estimate convergence rates.

3.2.1. Assumptions for superlinear convergence. We begin by reviewing the assumptions required for superlinear convergence of inexact projected Newton methods from [14]. We will express those assumptions in the less general language of the present paper. The assumptions were motivated by the functional analytic structure of the control problem and from geometric ideas in [14]. For example, Assumption 3.2 is motivated by the fact that the Hessian is a perturbation of the αI by an integral operator with a kernel that is continuously dependent on u in the L^∞-norm. Assumption 3.3 says that the active and inactive sets vary with ϵ in a controlled way. Geometrically, the meaning is that as $u(t)$ approaches u_{max} or u_{min} it does so in a nontangential way. This is essential for identification of the active set that is rapid enough to obtain superlinear convergence. See Figure 4.1 for an illustration of this nontangential behavior.

If u^* is a local minimizer that satisfies Assumptions 3.2 and 3.3 and the inexact projected Newton point is in the trust region, and the current iterate is sufficiently near u^* (in the L^2 sense), then the iteration will converge to u^* at a rate determined by the sequences $\{\epsilon_n\}$ and $\{\eta_n\}$.

The first assumption is the minimal regularity needed for any projected Newton method to converge rapidly.

Assumption 3.2. There is a solution u^* to (1.1)–(1.2) subject to $u \in \mathcal{U}$ such that f is twice Lipschitz continuously Fréchet differentiable at u^* in any $L^q[0,T]$ for $1 \le q \le \infty$ and in $C[0,T]$ and $\mathcal{K}_0$ is Lipschitz continuous as a map from $L^2[0,T]$ to $C[0,T]$. There are $\epsilon_{max}, \rho_{max}, M > 0$ such that for all $\epsilon \in (0, \epsilon_{max})$ and all u such that $\|u - u^*\|_\infty < \rho_{max}$ the approximate reduced Hessian computed using (2.6) is nonsingular. Moreover,

$$\|\mathcal{K}_0'(u)\| \le M, \tag{3.12}$$

where the norm in (3.12) is any of $\mathcal{L}(L^q, C[0,T])$, $2 \le q \le \infty$, or $\mathcal{L}(C[0,T])$ and

$$\|\mathcal{R}(u)\|_{L^2}, \|\mathcal{R}(u)^{-1}\|_{L^2} \le M.$$

The other assumption is related to a nondegeneracy assumption on the optimal control and a generalization of the fact that for finite dimensional problems the active set is identified in finitely many steps.

We define sets

$$\mathcal{I}_+ = \{t \,|\, F(u)(t) = \nabla f(u)(t)\} \cap \mathcal{I} \text{ and } \mathcal{I}_- = \{t \,|\, F(u)(t) \neq \nabla f(u)(t)\} \cap \mathcal{I}. \tag{3.13}$$

Clearly $[0,T] = \mathcal{I}_+ \cup \mathcal{I}_- \cup \mathcal{A}$.

Assumption 3.3. There is $\nu > 0$ such that $u_{min}(t) + \nu \le u_{max}(t)$ for all $t \in [0,T]$. Let $\mathcal{A}$ be given by (2.3) and let $p \in (0,1)$ be given. Then there are $\epsilon_{max}, \rho_{max}, M > 0$ such that for all $\epsilon \in (0, \epsilon_{max})$ and all $u = \mathcal{K}(v)$ such that $\|v - u^*\|_2 < \rho_{max}$

$$\|\mathcal{P}_\mathcal{A}(u - u^*)\|_2 \le M\epsilon \|u - u^*\|_\infty, \tag{3.14}$$

$$m(\mathcal{I}_-) \le M\epsilon, \tag{3.15}$$

and

$$\|u - \mathcal{P}(u - \nabla f(u))\|_\infty / M \le \|u - u^*\|_\infty \le M \|u - \mathcal{P}(u - \nabla f(u))\|_\infty. \tag{3.16}$$

In Assumption 3.3, m denotes Lebesgue measure.

On these assumptions we have the following local convergence result. The proof is, on the assumptions made above, a variation of those in [11] and [14], using the local theory in [19] to argue that the inexact Newton point is inside the trust region.

THEOREM 3.5. *Let Assumptions* 3.1, 3.2, *and* 3.3 *hold. Let* $M > 0$ *and* $p \in (0, 1)$ *be given. Then if* u_- *is sufficiently near* u^* *in* L^2, $u_c = \mathcal{K}(u_-)$,

$$\eta_c, \epsilon_c \leq M\sigma(u_c)^p, \tag{3.17}$$

and u_t *and* u_+ *are given by Algorithm* `trmin`, *then* $s_t = u_t - u_c$ *satisfies* (2.7) (*i.e., a full inexact Newton step is taken*),

$$\|u_t - u^*\|_2 = O(\|u_c - u^*\|_\infty^{1+p}),$$

and

$$\|u_+ - u^*\|_\infty = O(\|u_c - u^*\|_\infty^{1+p}).$$

4. Numerical example. All the results reported in this section were obtained on a Sun Ultra-1 running Sun Fortran f77 version 4.0 and Solaris 5.5.1.

Our numerical results are based on the problem posed by (1.1), (1.2), (1.3), (1.5), and (1.6). We set

$$g(y) = y^4/(100 + y^4/10), \quad T = 1, z(x) = 1, \quad \text{and } \alpha = .01.$$

We report results for both the constrained and the unconstrained problems. Our constraints are given by

$$u_{min}(t) = 0 \text{ and } u_{max}(t) = .1 + 4t.$$

Our initial iterate was $u_0 = 0$ for both cases.

In our numerical examples we discretized in space with piecewise linear finite elements as we did for the multilevel results reported in [14], and we integrated in time with the DAE solver DASPK (used in DASSL mode) [4], [5], [6], [16]. The accuracy parameters for DASPK were $rtol = atol = \delta_x^2/10$. With these parameters the numerical Hessian-vector products were accurate enough to observe superlinear convergence in the terminal phase of the iteration.

We summarize the algorithmic parameters in Table 4.1.

The radius of the trust region was initialized to $\Delta_{max} = 5$. Following [14] we limited the time step in DASPK to the spatial mesh width δ_x.

The solutions for the unconstrained (left) and constrained (right) problems, both with $\delta_x = 1/639$, are plotted in Figure 4.1. From the plot of the constrained minimizer on the right, one can see that both the upper and the lower bound constraints are attained on different parts of $[0, T]$.

TABLE 4.1
Parameters and tolerances.

$\{\mu_j\}_{0\le j\le 3}$	TR parameters , (2.18), (2.15)–(3.11) See steps 5(a)–5(e) in `trmin`	$10^{-4}, 10^{-4}, .25, .75$
$\Delta_{max}, \omega_1, \omega_2$	TR parameters See steps 5(a), 5(c), 5(e) in `trmin`	$5, .5, 2$
μ_4, β	Postsmoothing parameters See step 6(a) in `trmin`	.9999, .01
ϵ	Active set approximation, (2.3)	$\min(\sigma(u)^{.75}, .001)$
η	Forcing term, (2.7)	$.1\min(\sigma(u)^{.95}, .1)$
δ_x	Spatial mesh width	1/639
τ_g, τ_f	Termination tolerances, (2.20)	$\delta_x^2/10, 10^{-5}\delta_x^2$
h	Increment for FD Hessian, (2.10)	$\delta_x^{.67}$

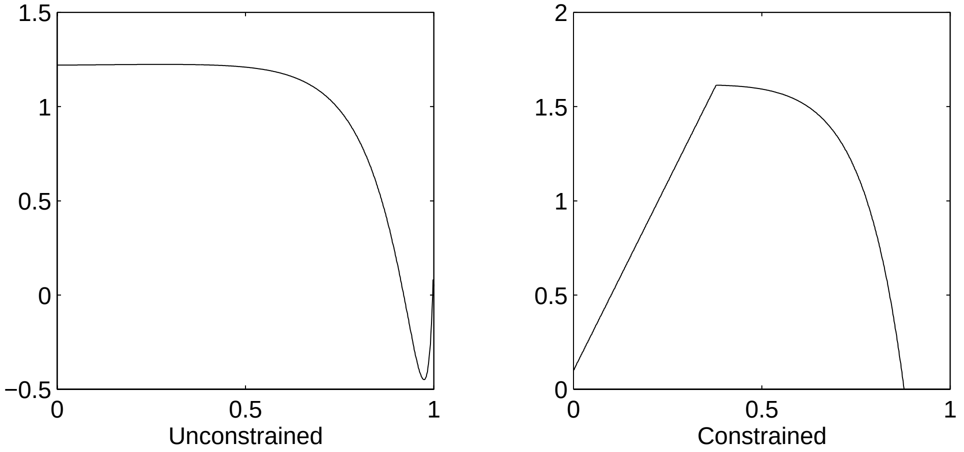


FIG. 4.1. *Minimizers,* $\delta_x = 1/639$.

For all computations we tabulate the iteration counter k; the value of the objective $f(u_k)$; the actual reduction (for $k \ge 1$); the norm of the projected gradient; $\sigma(u_k)$; the number of CG iterations required i_k (for $k \ge 1$); and the radius Δ of the trust region. For the constrained problem we also tabulate P_A, the fraction of points that are in the approximate active set. $i_k = 0$ means that the steepest descent or gradient projection step either went beyond the trust region boundary or satisfied the inexact Newton condition.

The iterations for both the unconstrained problem, reported in Table 4.2, and the constrained problem, summarized in Table 4.3, terminated when $\sigma(u) < \tau_g$. Full smoothing steps were taken for all but the first iteration in the unconstrained problem. For the constrained problem, which is harder, a full smoothing step was taken only for the final iterate, when local superlinear convergence sets in. Observations of full smoothing steps for smaller values of α would require a finer spatial mesh and smaller values of *rtol* and *atol* than the DAE solver would permit in our environment.

Table 4.2
Unconstrained problem, $\delta_x = 1/639$.

k	$f(u_k)$	$ared$	$\sigma(u_k)$	i_k	Δ
0	5.00e–01	0.00e+00	1.00e+00	0	5.00e+00
1	6.44e–03	–4.93e–01	5.33e–03	1	5.00e+00
2	6.20e–03	–3.71e–04	1.75e–02	12	5.00e+00
3	6.08e–03	–1.38e–04	5.17e–03	1	5.00e+00
4	6.07e–03	–1.19e–05	4.74e–04	1	5.00e+00
5	6.07e–03	–9.95e–08	5.37e–06	7	5.00e+00
6	6.07e–03	–3.31e–11	8.62e–09	17	5.00e+00

Table 4.3
Constrained problem, $\delta_x = 1/639$.

k	$f(u_k)$	$ared$	$\sigma(u_k)$	i_k	Δ	P_A
0	5.00e–01	0.00e+00	9.26e–01	0	5.00e+00	0.00
1	8.89e–03	–4.86e–01	2.52e–02	1	5.00e+00	0.30
2	7.36e–03	–1.29e–03	5.48e–03	7	5.00e+00	0.08
3	7.26e–03	–6.92e–05	2.76e–03	0	5.26e–02	0.41
4	7.17e–03	–8.24e–05	2.60e–04	3	1.05e–01	0.49
5	7.17e–03	–1.46e–06	3.80e–05	3	1.05e–01	0.50
6	7.17e–03	–1.35e–09	8.88e–06	4	1.05e–01	0.50
7	7.17e–03	–6.98e–11	1.69e–08	4	1.05e–01	0.50

REFERENCES

[1] L. Armijo, *Minimization of functions having Lipschitz-continuous first partial derivatives*, Pacific J. Math., 16 (1966), pp. 1–3.

[2] D. P. Bertsekas, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Trans. Automat. Control, 21 (1976), pp. 174–184.

[3] D. P. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.

[4] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Classics Appl. Math. 14, SIAM, Philadelphia, 1995.

[5] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold, *Using Krylov methods in the solution of large-scale differential-algebraic systems*, SIAM J. Sci. Comput., 15 (1994), pp. 1467–1488.

[6] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold, *Consistent initial condition calculation for differential-algebraic systems*, SIAM J. Sci. Comput., 19 (1998), pp. 1495–1512.

[7] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimization problems with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460.

[8] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comp., 50 (1988), pp. 399–430.

[9] J. W. Daniel, *The Approximate Minimization of Functionals*, Prentice–Hall, Englewood Cliffs, NJ, 1971.

[10] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Classics Appl. Math. 16, SIAM, Philadelphia, 1996.

[11] C. T. Kelley, *Identification of the support of nonsmoothness*, in Large Scale Optimization: State of the Art, W. W. Hager, D. W. Hearn, and P. Pardalos, eds., Kluwer Academic Publishers B.V., Boston, 1994, pp. 192–205.

[12] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers Appl. Math. 16, SIAM, Philadelphia, 1995.

[13] C. T. Kelley and E. W. Sachs, *Fast algorithms for compact fixed point problems with inexact function evaluations*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 725–742.

[14] C. T. Kelley and E. W. Sachs, *Multilevel algorithms for constrained compact fixed point problems*, SIAM J. Sci. Comput., 15 (1994), pp. 645–667.

[15] J. Nocedal and M. L. Overton, *Projected Hessian updating algorithms for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 22 (1985), pp. 821–850.

[16] L. R. Petzold, *A description of DASSL: A differential/algebraic system solver*, in Scientific Computing, R. S. Stepleman et al., eds., North–Holland, Amsterdam, The Netherlands, 1983, pp. 65–68.

[17] M. J. D. Powell, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1975, pp. 1–27.

[18] E. Sachs, *A parabolic control problem with a boundary condition of the Stefan-Boltzmann type*, Z. Angew. Math. Mech., 58 (1978), pp. 443–449.

[19] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[20] Ph. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., Academic Press, London, 1981, pp. 57–88.

[21] Ph. L. Toint, *Global convergence of a class of trust-region methods for nonconvex minimization in Hilbert space*, IMA J. Numer. Anal., 8 (1988), pp. 231–252.

[22] R. Winther, *A Numerical Galerkin Method for a Parabolic Problem*, Ph.D. thesis, Cornell University, Ithaca, New York, 1977.

[23] J. P. Yvon, *Controle optimal d'un four industriel*, Tech. report 22, INRIA, Le Chesnay, France, 1973.

SIAM J. OPTIM.
Vol. 9, No. 4, pp. 1082–1099

PATTERN SEARCH ALGORITHMS FOR BOUND CONSTRAINED MINIMIZATION[*]

ROBERT MICHAEL LEWIS[†] AND VIRGINIA TORCZON[‡]

To John Dennis, on his 60th birthday

Abstract. We present a convergence theory for pattern search methods for solving bound constrained nonlinear programs. The analysis relies on the abstract structure of pattern search methods and an understanding of how the pattern interacts with the bound constraints. This analysis makes it possible to develop pattern search methods for bound constrained problems while only slightly restricting the flexibility present in pattern search methods for unconstrained problems. We prove global convergence despite the fact that pattern search methods do not have explicit information concerning the gradient and its projection onto the feasible region and consequently are unable to enforce explicitly a notion of sufficient feasible decrease.

Key words. bound constrained optimization, convergence analysis, pattern search methods, direct search methods, globalization strategies, alternating variable search, axial relaxation, local variation, coordinate search, evolutionary operation, multidirectional search

AMS subject classifications. 49M30, 65K05

PII. S1052623496300507

1. Introduction.

This paper extends the class of pattern search methods for unconstrained minimization, considered in [16], to bound constrained problems:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & \ell \le x \le u, \end{array} \tag{1.1}$$

where $f : \mathbf{R}^n \to \mathbf{R}$, $\ell, x, u \in \mathbf{R}^n$, and $\ell < u$. We allow the possibility that some of the variables are unbounded either above or below by permitting $\ell_j, u_j = \pm\infty$, $j = 1, \ldots, n$.

Our convergence analysis is guided by that for pattern search methods for unconstrained problems [16]. We can guarantee that if the objective f is continuously differentiable, then a subsequence of the iterates produced by a pattern search method for problems with bound constraints converges to a stationary point of problem (1.1). By a stationary point of problem (1.1) we mean a feasible point x that satisfies the first-order necessary condition for optimality: for all feasible z, $(\nabla f(x)\ ,\ z - x) \ge 0$. Equivalently, x is a Karush–Kuhn–Tucker point for problem (1.1). As in the case of unconstrained minimization, pattern search methods for bound constrained problems accomplish this without an explicit representation of the gradient or the directional derivative. In particular, we prove global convergence in the bound constrained case

[*] Received by the editors March 13, 1996; accepted for publication (in revised form) January 12, 1998; published electronically September 24, 1999.

http://www.siam.org/journals/siopt/9-4/30050.html

[†] Institute for Computer Applications in Science and Engineering, Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23681-2199 (buckaroo@icase.edu). This research was supported by the National Aeronautics and Space Administration under NASA contract NAS1-19480.

[‡] Department of Computer Science, College of William & Mary, Williamsburg, VA 23187–8795 (va@cs.wm.edu). This research was supported by Air Force Office of Scientific Research grant F49620–95–1–0210, Sandia National Laboratories, Lawrence Livermore RFQ LC-2683, the National Aeronautics and Space Administration under NASA contract NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA, and by the Center for Research on Parallel Computation through NSF Cooperative Agreement 9120008.

even though pattern search methods do not have explicit information concerning the gradient and its projection onto the feasible region and consequently do not explicitly enforce a notion of sufficient feasible decrease.

In (1.1), near the boundary of the feasible region, the proximity of the boundary restricts the set of descent directions along which we can search *and* remain feasible for a sufficiently long distance. In projected gradient methods, one circumvents this inconvenience by combining knowledge about the local behavior of the objective f, namely, the gradient, with the global structure of the feasible region by conducting searches along the projected gradient path. In the case of pattern search methods, we do not have recourse to this strategy; nonetheless, we can specify the pattern so that it contains a sufficiently rich set of directions to ensure that we need not take too short a step to obtain a new iterate that produces decrease in f and is also feasible.

So far as we know, ours is the first convergence analysis for pattern search methods for bound constrained minimization. However, the observation that forms the basis of our analysis—the utility of having a sufficiently large subset of the pattern oriented along the coordinate directions in order to handle the bounds—is not new. For instance, in [10], Keefer notes that the pattern associated with the method of Hooke and Jeeves [9] is well suited for coping with bounds and proposes the Simpat algorithm, which combines the use of the Nelder–Mead simplex algorithm [12] in the interior of the feasible region with the use of the Hooke and Jeeves pattern search algorithm near the boundary.

The general specification of pattern search methods for bound constrained minimization gives us broad latitude in designing such algorithms. Moreover, as we shall discuss, classical pattern search methods for unconstrained minimization—such as coordinate search with fixed step sizes and the original pattern search of Hooke and Jeeves—can be generalized without modification to the bound constrained case. We also will show that not all pattern search methods for unconstrained minimization immediately generalize to bound constrained problems: in section 2 we present a counterexample that defeats G.E.P. Box's method of evolutionary operation using two-level factorial designs [1, 3, 14] and show how the convergence theory guides us to a remedy that uses composite designs [2], instead of the simpler factorial or fractional factorial designs. The multidirectional search algorithm of Dennis and Torczon [7, 15] also requires us to augment the pattern used for the algorithm; again we find a straightforward extension, but one that reveals much about the interesting behavior of the simplices which characterize that method.

2. Motivation.

Before giving the technical specification of pattern search methods for bound constrained minimization, we consider an example that illustrates what is needed for the generalization and how the bound constrained algorithms work. Consider the following simple linear problem:

$$\begin{array}{ll} \text{minimize} & f(x) = -(x^1 + 2x^2) \\ \text{subject to} & \quad 0 \le x^1 \le 1, \\ & \qquad x^2 \le 0. \end{array}$$

The solution of this problem is $x_* = (1, 0)^T$. Let us consider an iteration of the pattern search method of evolutionary operation applied to this problem starting at the initial iterate $x_0 = (0, 0)^T$.

The usual pattern is typically a factorial design comprising the points NW, NE, SW, and SE indicated by the open circles in Figure 2.1. We see that the values of f at the points NW and NE are lower than that at x_0. If there were no constraints, as

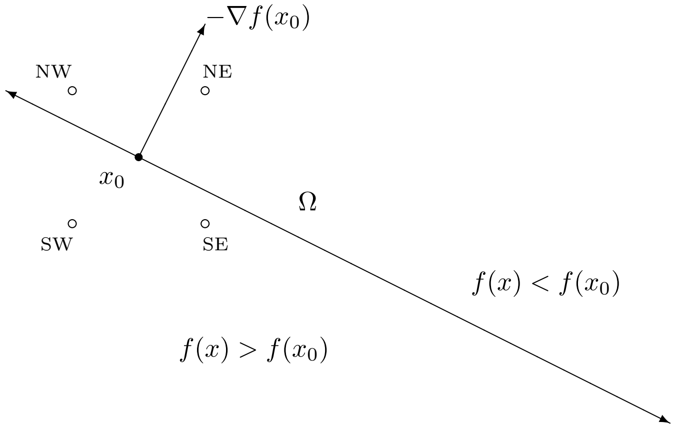


Fig. 2.1. *The pattern for factorial design in the unconstrained case.*

depicted in Figure 2.1, the algorithm could choose either of these points as the next iterate; most implementations would choose NE since it produces the greater decrease in f.

In the unconstrained case, pattern search methods work much like line-search quasi-Newton methods. Pattern search methods include sufficient search directions to guarantee that if the current iterate is not a stationary point, then at least one of the search directions is a descent direction. Moreover, one can prove that as the iterations progress, these "good" search directions cannot become increasingly orthogonal to the steepest descent direction. In the situation depicted above, for instance, regardless of the direction of $-\nabla f(x_0)$, one of the four directions from x_0 to the corners of the square the pattern defines must make an angle of 45° or less with $-\nabla f(x_0)$. Finally, the way the pattern is rescaled implements a form of backtracking that is the final piece needed to guarantee convergence.

Now consider what happens in our simple example when we take into account the constraints. We will consider only feasible points in the pattern, in order to ensure that the algorithm produces only feasible iterates. In Figure 2.2 we see that the only feasible point is SE. Unfortunately, this step will produce increase in f. We cannot remedy this by moving the pattern closer to x_0—backtracking along the directions from x_0 to the points in the pattern—since the only feasible points that will ensue lie along the line segment from x_0 to SE, and on this line segment f is larger than $f(x_0)$. Consequently, evolutionary operation will never move from x_0.

The problem is that while there are feasible directions of descent emanating from x_0, our pattern is not oriented in such a way as to capture any of this information from its feasible point SE. The pattern associated with evolutionary operation is not compatible with the geometry of the feasible region. A moment's reflection reveals that the problem is that the pattern does not allow us to move parallel to the bounds.

This problem goes away if, for instance, we augment the pattern using the idea of composite design [2] (as opposed to factorial design). An example of such a design is shown in Figure 2.3. We now have a feasible step along the active constraint $x^2 \leq 0$ that will produce descent.

This simple example captures the essential idea for the generalization of pattern search methods to bound constrained minimization. We restrict our attention to

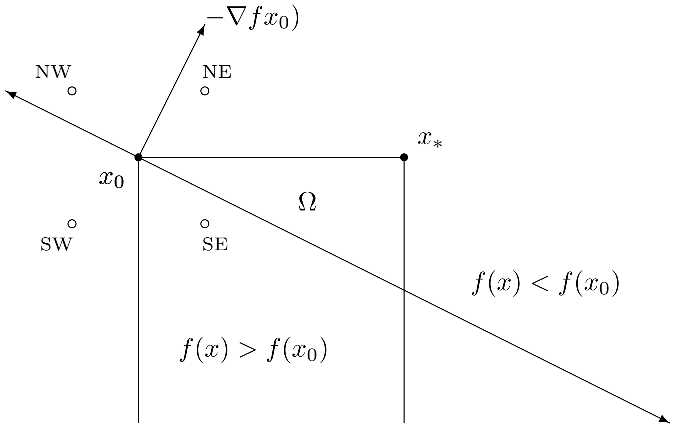


FIG. 2.2. *An illustration of what can go wrong with factorial design in the bound constrained case.*

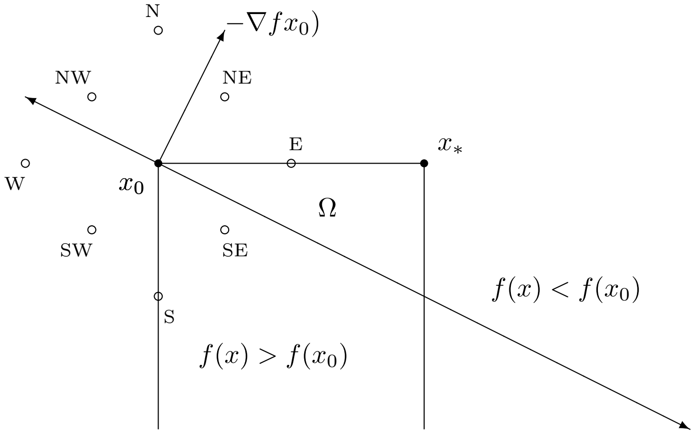


FIG. 2.3. *An illustration of how the problem can be circumvented using a composite design.*

patterns that reflect the geometry of the feasible region by including enough directions oriented along the coordinate axes so that we can move parallel and perpendicular to the boundary of the feasible region. We can then guarantee global convergence to a Karush–Kuhn–Tucker point.

Notation. We denote by $\mathbf{R}$, $\mathbf{Q}$, $\mathbf{Z}$, and $\mathbf{N}$ the sets of real, rational, integer, and natural numbers, respectively.

Unless otherwise noted, norms are assumed to be the Euclidean norm. The feasible region for problem (1.1) we denote by Ω:

$$\Omega = \{\, x \in \mathbf{R}^n \;\mid\; \ell \leq x \leq u \,\}.$$

The projection onto Ω we denote by P. If for scalar t we define

$$p_j(t) = \begin{cases} \ell_j & \text{if } t < \ell_j, \\ t & \text{if } \ell_j \leq t \leq u_j, \\ u_j & \text{if } t > u_j, \end{cases}$$

then the projection of $x = (x_1, \ldots, x_n)^T$ is given by

$$P(x) = \sum_{j=1}^{n} p_j(x_j) e_j,$$

where $\{e_j\}$, $j = 1, \ldots, n$, are the standard basis vectors. On those few occasions where we must denote components of subscripted vectors, we use the following notation: $q_{k,j}$ denotes the jth component of the vector q_k.

We will denote by $g(x)$ the gradient $\nabla f(x)$ of the objective. Finally, let

$$L_\Omega(y) = \{ \ x \in \Omega \ \mid \ f(x) \leq f(y) \ \}.$$

3. Pattern search methods.

We begin by defining the general pattern search method for the bound constrained problem (1.1); it differs from that for unconstrained problems [16] in only a few particulars, which we summarize in section 3.5.

3.1. The pattern.

As with pattern search methods for unconstrained problems, to define a pattern we need two components: a *basis matrix* and a *generating matrix*.

The basis matrix is a nonsingular matrix $B \in \mathbf{R}^{n \times n}$.

The generating matrix is a matrix $C_k \in \mathbf{Z}^{n \times p}$, where $p > 2n$. We partition the generating matrix into components

$$C_k \quad = \quad [\quad M_k \qquad -M_k \qquad L_k \quad] \quad = \quad [\quad \Gamma_k \qquad L_k \quad]. \tag{3.1}$$

We require that $M_k \in \mathbf{M} \subset \mathbf{Z}^{n \times n}$, where $\mathbf{M}$ is a finite set of nonsingular matrices, and that $L_k \in \mathbf{Z}^{n \times (p-2n)}$ and contains at least one column, a column of zeros.

A *pattern* P_k is then defined by the columns of the matrix $P_k = BC_k$. For convenience, we use the partition of the generating matrix C_k given in (3.1) to partition P_k as follows:

$$P_k \quad = \quad BC_k \quad = \quad [\quad BM_k \qquad -BM_k \qquad BL_k \quad] \quad = \quad [\quad B\Gamma_k \qquad BL_k \quad].$$

We also require the matrix BM_k to be diagonal:

$$BM_k = \operatorname{diag}(d_k^i), \quad i = 1, \ldots, n. \tag{3.2}$$

This condition, absent in the case of unconstrained minimization, is needed in order to ensure that we can find feasible points in the pattern that will also produce decrease in the objective. As we shall see, this condition is not especially restrictive and is satisfied by all of the commonly encountered pattern search algorithms or straightforward variants of them.

At iteration k, given $\Delta_k \in \mathbf{R}$ with $\Delta_k > 0$, we define a *trial step* to be a vector of the form $s_k^i = \Delta_k B c_k^i$ for some $i \in \{1, \ldots, p\}$, where c_k^i denotes the ith column of C_k (i.e., $C_k = [c_k^1 \cdots c_k^p]$). We call a trial step s_k^i *feasible* if $(x_k + s_k^i) \in \Omega$. At iteration k, a *trial point* is any point of the form $x_k^i = x_k + s_k^i$, where x_k is the current iterate.

3.2. The bound constrained exploratory moves.

Pattern search methods proceed by conducting a series of *exploratory moves* about the current iterate x_k to choose a new iterate $x_{k+1} = x_k + s_k$ for some feasible step s_k determined during the course of the exploratory moves. The hypotheses listed in Figure 3.1 on the result of the bound constrained exploratory moves allow a broad choice of exploratory moves while ensuring the properties required to prove convergence. By abuse of notation, if A is a matrix, $y \in A$ means that the vector y is a column of A.

1. $s_k \in \Delta_k P_k \equiv \Delta_k B C_k \equiv \Delta_k \, [B\Gamma_k \;\; BL_k]$.
2. $(x_k + s_k) \in \Omega$.
3. If $\min \{ \, f(x_k + y) \;\mid\; y \in \Delta_k B\Gamma_k, \;\; x_k + y \in \Omega \, \} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$.

FIG. 3.1. *Hypotheses on the result of the bound constrained exploratory moves.*

```
Let $x_0 \in \Omega$ and $\Delta_0 > 0$ be given.
For $k = 0, 1, \ldots,$
    a. compute $f(x_k)$.
    b. determine a step $s_k$ using a bound constrained exploratory moves algorithm.
    c. if $f(x_k + s_k) < f(x_k)$, then $x_{k+1} = x_k + s_k$. Otherwise $x_{k+1} = x_k$.
    d. update $C_k$ and $\Delta_k$.
```

FIG. 3.2. *The generalized pattern search method for bound constrained problems.*

```
Let $\tau \in \mathbf{Q}$, $\tau > 1$, and $\{w_0, w_1, \ldots, w_L\} \subset \mathbf{Z}$, $w_0 < 0$, and $w_i \geq 0$, $i = 1, \ldots, L$. Let
$\theta = \tau^{w_0}$ and $\lambda_k \in \Lambda = \{\tau^{w_1}, \ldots, \tau^{w_L}\}$.
    a. If $f(x_k + s_k) \geq f(x_k)$, then $\Delta_{k+1} = \theta \Delta_k$.
    b. If $f(x_k + s_k) < f(x_k)$, then $\Delta_{k+1} = \lambda_k \Delta_k$.
```

FIG. 3.3. *Rules for updating Δ_k.*

3.3. The generalized pattern search method. Figure 3.2 states the generalized pattern search method for minimization with bound constraints. To define a particular pattern search method, we must specify the basis matrix B, the generating matrix C_k, the bound constrained exploratory moves to be used to produce a feasible step s_k, and the algorithms for updating C_k and Δ_k.

3.4. The updates. Figure 3.3 specifies the rules for updating Δ_k. The aim of the update of Δ_k is to force a strict reduction in f. An iteration with $f(x_k + s_k) < f(x_k)$ is *successful;* otherwise, the iteration is *unsuccessful.* Note that to accept a step we require only *simple*, as opposed to *sufficient*, decrease.

The conditions on θ and Λ ensure that $0 < \theta < 1$ and $\lambda_i \geq 1$ for all $\lambda_i \in \Lambda$. Thus, if an iteration is successful it may be possible to increase the step length parameter Δ_k, but Δ_k is not allowed to decrease.

3.5. Differences between pattern search methods for unconstrained and bound constrained minimization. There are only two additional restrictions required of pattern search methods to ensure convergence for the bound constrained case.

First note that as we have defined them, pattern search methods for bound constrained minimization are *feasible point* methods; the search begins with a point that satisfies the bounds and maintains feasibility throughout the search. This can be seen in Figure 3.2, where we require $x_0 \in \Omega$. This requirement also appears in the hypotheses on the result of the bound constrained exploratory moves given in Figure 3.1: if simple decrease on the function value at the current iterate can be found among any of the feasible trial steps contained in the columns of $\Delta_k B\Gamma_k$, then the exploratory moves must produce a feasible step s_k that also gives simple decrease on the function value at the current iterate.

The second, and more interesting, restriction is that the *core pattern* BM_k must be defined by a diagonal matrix. Because the columns of the pattern matrix determine the directions of the steps that may be considered, we need to ensure that if we are not at a constrained stationary point, we have at least one feasible direction of descent. Moreover, we need a feasible direction of descent along which we will remain feasible for a sufficiently long distance to avoid taking too short a step. This is a crucial point since we do not enforce any notion of sufficient decrease. Practically, we must ensure that we have directions that allow us to move parallel to the constraints. Requiring BM_k to be a diagonal matrix is sufficient, and as we saw in section 2, such a requirement is unavoidable.

We note an equivalence between pattern search methods for bound constrained problems and an exact penalization approach to problem (1.1). Applying a pattern search method for problem (1.1) produces exactly the same iterates as applying such an algorithm to the unconstrained problem

$$\text{minimize} \quad F(x),$$

where

$$F(x) = \begin{cases} f(x) & \text{if } x \in \Omega, \\ \infty & \text{otherwise.} \end{cases}$$

In fact, this is one classical approach used with direct search methods to ensure that the iterates produced remain feasible (see, for instance, [10, 12, 13]). In the case of pattern search methods this formulation is not simply a conceptual approach; pattern search methods are directly applicable to this exact penalty function since they do not rely on derivatives. However, as we demonstrated in section 2, this exact penalization approach cannot be applied with an arbitrary pattern search method for unconstrained minimization; we require that BM_k be diagonal.

3.6. Results from the unconstrained theory. We recall the following results from [16], to which we refer the reader for the proofs. The first result indicates one sense in which Δ_k regulates step length.

LEMMA 3.1 (Lemma 3.1 from [16]). *There exists a constant $\zeta_* > 0$, independent of k, such that for any trial step $s_k^i \neq 0$ produced by a generalized pattern search method* (*Figure* 3.2), *we have $\| s_k^i \| \geq \zeta_* \Delta_k$.*

The next result is key to the convergence of pattern search methods. It states that the iterates produced by a pattern search method have a rigid algebraic structure.

THEOREM 3.2 (Theorem 3.2 from [16]). *Any iterate x_N produced by a generalized pattern search method* (*Figure* 3.2) *can be expressed in the following form:*

$$x_N = x_0 + \left(\beta^{r_{LB}} \alpha^{-r_{UB}}\right) \Delta_0 B \sum_{k=0}^{N-1} z_k, \tag{3.3}$$

where

- *x_0 is the initial guess;*
- *$\beta/\alpha \equiv \tau$, with $\alpha, \beta \in \mathbf{N}$ and relatively prime, and τ is as defined in the rules for updating Δ_k given in Figure* 3.3;
- *r_{LB} and r_{UB} are integers depending on N;*
- *Δ_0 is the initial choice for the step length control parameter;*

- *B is the basis matrix; and*
- $z_k \in \mathbf{Z}^n$, $k = 0, \ldots, N-1$.

The last result we recollect says, in conjunction with Lemma 3.1, that if we bound the size of the elements of the generating matrix (which is a reasonable thing to do), then Δ_k completely regulates the size of the steps a pattern search method takes. This result is a direct consequence of the fact that $s_k^i = \Delta_k B c_k^i$.

LEMMA 3.3 (Lemma 3.6 from [16]). *If there exists a constant $\mathcal{C} > 0$ such that for all k, $\mathcal{C} > \|c_k^i\|$, for all $i = 1, \ldots, p$, then there exists a constant $\psi_* > 0$, independent of k, such that for any trial step s_k^i produced by a generalized pattern search method (Figure 3.2) we have $\Delta_k \geq \psi_* \|s_k^i\|$.*

4. Convergence theory.

We now present the first-order constrained stationary point convergence theory for pattern search methods for bound constrained problems. We begin by defining, for feasible x, the quantity

$$q(x) = P(x - g(x)) - x.$$

In the bound constrained theory the quantity $q(x)$ plays the role of $g(x)$ in the unconstrained theory, giving us a continuous measure of how close we are to constrained stationarity, as in the theory for methods based explicitly on derivatives (e.g., [5, 6, 8]). The following proposition summarizes two results concerning q that we will shortly need, particularly the fact that x is a constrained stationary point for (1.1) if and only if $q(x) = 0$. While stated for the particular domain Ω, the proposition holds for any closed convex domain. The results are classical; see section 2 of [8], for instance.

PROPOSITION 4.1. *Let $x \in \Omega$. Then*

$$\| \, q(x) \, \| \leq \| \, g(x) \, \|,$$

and x is a stationary point for problem (1.1) if and only if $q(x) = 0$.

We can now state the first convergence result for the general pattern search method for bound constrained minimization. Henceforth we will assume that $L_\Omega(x_0)$ is compact and that f is continuously differentiable on an open neighborhood D of $L_\Omega(x_0)$.

THEOREM 4.2. *Let $L_\Omega(x_0)$ be compact and suppose f is continuously differentiable on an open neighborhood D of $L_\Omega(x_0)$. Let $\{x_k\}$ be the sequence of iterates produced by a generalized pattern search method for bound constrained minimization (Figure 3.2). Then*

$$\liminf_{k \to +\infty} \| \, q(x_k) \, \| = 0 \, .$$

The proof of this theorem is given in section 5.1, after we have established the necessary intermediate results.

We can strengthen the result given in Theorem 4.2 in the same way that we do in the unconstrained case [16]. First, we require the columns of the generating matrix C_k to remain bounded in norm, i.e., that there exists a constant $\mathcal{C} > 0$ such that for all k, $\mathcal{C} > \|c_k^i\|$, for all $i = 1, \ldots, p$. Second, we replace the original hypotheses on the result of the bound constrained exploratory moves with a stronger version, given in Figure 4.1. Third, we require that $\lim_{k \to +\infty} \Delta_k = 0$. All the algorithms described in section 6, except multidirectional search, satisfy this third condition because of the customary choice of $\Lambda = \{1\} \equiv \{\tau^0\}$. However, it is not necessary to force the steps to be nonincreasing.

1. $s_k \in \Delta_k P_k \equiv \Delta_k B C_k \equiv \Delta_k \left[B\Gamma_k \;\; BL_k \right]$.
2. $(x_k + s_k) \in \Omega$.
3. If $\min \{ \, f(x_k + y) \;\mid\; y \in \Delta_k B\Gamma_k, \; x_k + y \in \Omega \, \} < f(x_k)$, then $f(x_k + s_k) \leq \min \{ \, f(x_k + y) \;\mid\; y \in \Delta_k B\Gamma_k, \; x_k + y \in \Omega \, \}$.

FIG. 4.1. *Strong hypotheses on the result of the bound constrained exploratory moves.*

THEOREM 4.3. *Let $L_\Omega(x_0)$ be compact and suppose f is continuously differentiable on an open neighborhood D of $L_\Omega(x_0)$. In addition, assume that the columns of the generating matrices are uniformly bounded in norm, that $\lim_{k\to+\infty} \Delta_k = 0$, and that the generalized pattern search method for bound constrained minimization* (*Figure* 3.2) *enforces the strong hypotheses on the result of the bound constrained exploratory moves* (*Figure* 4.1). *Then, for the sequence of iterates $\{x_k\}$ produced by the generalized pattern search method for bound constrained minimization,*

$$\lim_{k\to+\infty} \| q(x_k) \| = 0 \, .$$

The proof will be found in section 5.2.

5. Proof of Theorems 4.2 and 4.3. Throughout this section, x_k will refer to an iterate produced by a pattern search algorithm for bound constrained minimization. By design, x_k is feasible, i.e., $x_k \in \Omega$. Given an iterate x_k, let $g_k = g(x_k)$ and $q_k = q(x_k)$. Let $B(x, \delta)$ be the ball with center x and radius δ, and let ω denote the following modulus of continuity of $g(x)$: given $x \in L_\Omega(x_0)$ and $\varepsilon > 0$,

$$\omega(x, \varepsilon) = \sup \{ \, \delta > 0 \;\mid\; B(x,\delta) \subset D \text{ and } \| \, g(y) - g(x) \, \| < \varepsilon \text{ for all } y \in B(x,\delta) \, \} \, .$$

We begin with an elementary proposition concerning descent directions.

PROPOSITION 5.1. *Let $s \in \mathbf{R}^n$ and $x \in L_\Omega(x_0)$. Assume, too, that $g(x) \neq 0$ and $g(x)^T s \leq -\varepsilon \| \, s \, \|$. Then, if $\| \, s \, \| < \omega(x, \frac{\varepsilon}{2})$,*

$$f(x + s) - f(x) \leq -\frac{\varepsilon}{2} \| \, s \, \| .$$

Proof. If $\| \, s \, \| < \omega(x, \frac{\varepsilon}{2})$, then the closed line segment $[x, x+s]$ from x to $x + s$ is contained in D, where f is continuously differentiable. We may thus apply the mean-value theorem; we have, for some y on the line segment between x and $x + s$,

$$\begin{aligned} f(x+s) - f(x) &= g(x)^T s + (g(y) - g(x))^T s \\ &\leq -\varepsilon \| \, s \, \| + \| \, g(y) - g(x) \, \| \| \, s \, \| . \end{aligned}$$

If $\| \, s \, \| < \omega(x, \frac{\varepsilon}{2})$, then $\| \, g(y) - g(x) \, \| \leq \frac{\varepsilon}{2}$ and the result follows. □

It is in the proof of the next result that the bound constrained and the unconstrained cases differ most. The proof of Proposition 5.2 implicitly relies on the fact that in the bound constrained case, the directions in the pattern defined by the columns of BM_k are coordinate directions and thus are oriented normal and tangent to the faces of the feasible region. That this is not merely convenient is clear from the example given in section 2.

PROPOSITION 5.2. *Suppose that $q_k \neq 0$. Then there exists a $\nu_k > 0$ such that if $\Delta_k < \nu_k$, then there is a trial step s_k^i defined by a column of $\Delta_k B\Gamma_k$ for which $(x_k + s_k^i) \in \Omega$ and*

$$g_k^T s_k^i \leq -n^{-\frac{1}{2}} \| \, q_k \, \| \| \, s_k^i \, \| .$$

Proof. We restrict our attention to the steps defined by the columns of $\Delta_k B\Gamma_k$; by hypothesis, $\Delta_k B\Gamma_k \equiv \Delta_k B[M_k \ -M_k] = \Delta_k[\mathrm{diag}(d_k^i) \ -\mathrm{diag}(d_k^i)]$ (see (3.2)). Choose an index m for which

$$|\, q_{k,m} \,| = \|\, q_k \,\|_\infty \geq n^{-\frac{1}{2}} \|\, q_k \,\|, \tag{5.1}$$

where $q_{k,m}$ is the mth component of q_k. Note that it is also the case that

$$|\, g_{k,m} \,| \geq |\, q_{k,m} \,| \tag{5.2}$$

and $\mathrm{sign}(g_{k,m}) = \mathrm{sign}(q_{k,m})$.

Let $s_k^i = -\,\mathrm{sign}(g_{k,m})\Delta_k \,|\, d_k^m \,|\, e_m$; this vector will be among the columns of $\Delta_k B\Gamma_k$. Since $x_k + q_k = P(x_k - g_k)$ is feasible, we have $\ell \leq x_k + q_k \leq u$ and thus

$$\ell_m \leq x_{k,m} + q_{k,m} \leq u_m.$$

It follows that if $\Delta_k \,|\, d_k^m \,| \leq |\, q_{k,m} \,|$, then the trial point $x_k^i = x_k + s_k^i$ will be feasible. Moreover, from (5.1) and (5.2),

$$g_k^T s_k^i = -\,\mathrm{sign}(g_{k,m})\Delta_k \,|\, d_k^m \,|\, g_{k,m} = -\|\, s_k^i \,\|\,|\, g_{k,m} \,| \leq -n^{-\frac{1}{2}} \|\, s_k^i \,\|\|\, q_k \,\|.$$

Defining $\nu_k = \|\, q_k \,\|_\infty / \,|\, d_k^m \,|$ then does the trick. □

PROPOSITION 5.3. *Given any $\eta > 0$, there exists $\delta > 0$, independent of k, such that if $\Delta_k < \delta$ and $\|\, q_k \,\| > \eta$, the pattern search method for bound constrained minimization* (*Figure* 3.2) *will find an acceptable step s_k, i.e., $f(x_k + s_k) < f(x_k)$ and $(x_k + s_k) \in \Omega$.*

If, in addition, the columns of the generating matrix remain bounded in norm and we enforce the strong hypotheses on the result of the bound constrained exploratory moves (*Figure* 4.1), *then, given any $\eta > 0$, there exist $\delta > 0$ and $\sigma > 0$, independent of k, such that if $\Delta_k < \delta$ and $\|\, q_k \,\| > \eta$, then*

$$f(x_{k+1}) \leq f(x_k) - \sigma \|\, q_k \,\|\|\, s_k \,\|.$$

Proof. Since $g(x)$ is uniformly continuous on $L_\Omega(x_0)$ and $L_\Omega(x_0)$ is a compact subset of the open set D, there exists $\omega_* > 0$ such that

$$\omega\left(x_k, n^{-\frac{1}{2}}\eta\right) \geq \omega_*$$

for all k for which $\|\, q_k \,\| > \eta$.

Next, choose $d^* > 0$ such that $d_k^i \leq d^*$ for all i and k. This we can do because the set $\{d_k^i\}$ is finite (see (3.2) and the conditions on M_k given in section 3.1). Let

$$\nu_* = \frac{n^{-\frac{1}{2}}\eta}{d^*};$$

then

$$\nu_* = \frac{n^{-\frac{1}{2}}\eta}{d^*} \leq \frac{n^{-\frac{1}{2}}\|\, q_k \,\|}{d^*} \leq \frac{\|\, q_k \,\|_\infty}{d^*} \leq \nu_k$$

for all k for which $\|\, q_k \,\| > \eta$, where ν_k is as in Proposition 5.2.

Finally, let

$$\delta = \min\left(\nu_*, \omega_*/d^*\right).$$

Now suppose $\| q_k \| > \eta$ and $\Delta_k < \delta$. Since $\Delta_k < \nu_k$, Proposition 5.2 assures us of the existence of a step s_k^i defined by a column of $\Delta_k B \Gamma_k$ such that $(x_k + s_k^i) \in \Omega$ and

$$g_k^T s_k^i \leq -n^{-\frac{1}{2}} \| q_k \| \| s_k^i \|.$$

At the same time, we also have

$$\| s_k^i \| \leq \Delta_k d^* \leq \omega_* \leq \omega\left(x_k, n^{-\frac{1}{2}} \| q_k \| \right).$$

So, by Proposition 5.1,

$$f(x_k + s_k^i) - f(x_k) \leq -\frac{1}{2} n^{-\frac{1}{2}} \| q_k \| \| s_k^i \|.$$

Thus, when $\Delta_k < \delta$, $f(x_k^i) \equiv f(x_k + s_k^i) < f(x_k)$ for at least one feasible $s_k^i \in \Delta_k B \Gamma_k$. The hypotheses on the result of the bound constrained exploratory moves guarantee that if

$$\min\{\, f(x_k + y) \ \mid \ y \in \Delta_k B \Gamma_k, \ x_k + y \in \Omega \,\} < f(x_k),$$

then $f(x_k + s_k) < f(x_k)$ and $(x_k + s_k) \in \Omega$. This proves the first part of the proposition.

If, in addition, we enforce the strong hypotheses on the result of the bound constrained exploratory moves, then we actually have

$$f(x_{k+1}) - f(x_k) \leq -\frac{1}{2} n^{-\frac{1}{2}} \| q_k \| \| s_k^i \|.$$

Lemma 3.1 then ensures that

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2} n^{-\frac{1}{2}} \zeta_* \Delta_k \| q_k \|.$$

Applying Lemma 3.3, we arrive at

$$f(x_{k+1}) \leq f(x_k) - \sigma \| q_k \| \| s_k \|,$$

where $\sigma = \frac{1}{2} n^{-\frac{1}{2}} \zeta_* \psi_*$. □

COROLLARY 5.4. *If* $\liminf_{k \to +\infty} \| q_k \| \neq 0$*, then there exists a constant* $\Delta_* > 0$ *such that for all* k*,* $\Delta_k > \Delta_*$*.*

Proof. By hypothesis, there exists K and $\eta > 0$ such that for all $k > K$, $\| q_k \| > \eta$. By Proposition 5.3, we can find δ such that if $k > K$ and $\Delta_k < \delta$, then we will find an acceptable step. In view of the rules for updating Δ_k given in Figure 3.3, we are assured that for all $k > K$, $\Delta_k > \theta\delta$. We may then take $\Delta_* = \min\{\Delta_0, \ldots, \Delta_K, \theta\delta\}$. □

The next theorem combines the strict algebraic structure of the iterates with the simple decrease condition of the generalized pattern search algorithm for bound constrained problems (Figure 3.2), along with the rules for updating Δ_k (Figure 3.3), to give us a useful fact about the limiting behavior of Δ_k.

THEOREM 5.5. *Assume that* $L_\Omega(x_0)$ *is compact. Then* $\liminf_{k \to +\infty} \Delta_k = 0$.

Proof. The proof is like that of Theorem 3.3 in [16]. Suppose $0 < \Delta_{LB} \leq \Delta_k$ for all k. Using the rules for updating Δ_k found in Figure 3.3, it is possible to write Δ_k as $\Delta_k = \tau^{r_k}\Delta_0$, where $r_k \in \mathbf{Z}$.

The hypothesis that $\Delta_{LB} \leq \Delta_k$ for all k means that the sequence $\{\tau^{r_k}\}$ is bounded away from zero. Meanwhile, we also know that the sequence $\{\Delta_k\}$ is bounded above because all the iterates x_k must lie inside the set $L_\Omega(x_0) = \{x \in \Omega : f(x) \leq f(x_0)\}$ and the latter set is compact; Lemma 3.1 then guarantees an upper bound Δ_{UB} for $\{\Delta_k\}$. This, in turn, means that the sequence $\{\tau^{r_k}\}$ is bounded above. Consequently, the sequence $\{\tau^{r_k}\}$ is a finite set. Equivalently, the sequence $\{r_k\}$ is bounded above and below.

Next we recall the exact identity of the quantities r_{LB} and r_{UB} in Theorem 3.2; the details are found in the proof of Theorem 3.3 in [16]. At iteration N we have

$$r_{LB} = \min_{0 \leq k < N} \{r_k\}, \qquad r_{UB} = \max_{0 \leq k < N} \{r_k\}.$$

If, in the matter at hand, we let

$$r_{LB} = \min_{0 \leq k < +\infty} \{r_k\}, \qquad r_{UB} = \max_{0 \leq k < +\infty} \{r_k\}, \tag{5.3}$$

then (3.3) holds for the bounds given in (5.3), and we see that for all k, x_k lies in the translated integer lattice G generated by x_0 and the columns of $\beta^{r_{LB}}\alpha^{-r_{UB}}\Delta_0 B$.

The intersection of the compact set $L_\Omega(x_0)$ with the lattice G is finite. Thus, there must exist at least one point x_* in the lattice for which $x_k = x_*$ for infinitely many k.

We now appeal to the simple decrease condition (c) in Figure 3.2, which guarantees that an iterate cannot be revisited infinitely many times since we accept a new step s_k if and only if $f(x_k) > f(x_k + s_k)$ and $(x_k + s_k) \in \Omega$. Thus there exists an N such that for all $k \geq N$, $x_k = x_*$, which implies that $f(x_k) = f(x_k + s_k)$.

We now appeal to the rules for updating Δ_k (Figure 3.3, part (a)) to see that $\Delta_k \to 0$, thus leading to a contradiction. □

5.1. Proof of Theorem 4.2. The proof is like that of Theorem 3.5 in [16]. Suppose that $\liminf_{k \to +\infty} \| q(x_k) \| \neq 0$. Then Corollary 5.4 tells us that there exists $\Delta_* > 0$ such that for all k, $\Delta_k \geq \Delta_*$. But this contradicts Theorem 5.5.

5.2. Proof of Theorem 4.3. The proof, also by contradiction, follows that of Theorem 3.7 in [16]. Suppose $\limsup_{k \to +\infty} \| q(x_k) \| \neq 0$. Let $\varepsilon > 0$ be such that there exists a subsequence $\| q(x_{m_i}) \| \geq \varepsilon$. Since

$$\liminf_{k \to +\infty} \| q(x_k) \| = 0,$$

given any $0 < \eta < \varepsilon$, there exists an associated subsequence l_i such that

$$\| q(x_k) \| \quad > \eta \qquad \text{for} \qquad m_i \leq k < l_i, \qquad \| q(x_{l_i}) \| \quad < \quad \eta.$$

Since $\Delta_k \to 0$, we can appeal to Proposition 5.3 to obtain for $m_i \leq k < l_i$, i sufficiently large,

$$f(x_k) - f(x_{k+1}) \quad \geq \quad \sigma \| q(x_k) \| \| s_k \| \quad \geq \quad \sigma\eta \| s_k \|,$$

where $\sigma > 0$. Then the telescoping sum,

$$\begin{aligned} &(f(x_{m_i}) - f(x_{m_i+1})) + (f(x_{m_i+1}) - f(x_{m_i+2})) + \cdots + (f(x_{l_i-1}) - f(x_{l_i})) \\ &\qquad \geq \sum_{k=m_i}^{l_i} \sigma\eta \| s_k \|, \end{aligned}$$

gives us

$$f(x_{m_i}) - f(x_{l_i}) \quad \geq \quad \textstyle\sum_{k=m_i}^{l_i} \sigma\eta \| \, s_k \, \| \quad \geq \quad c' \| \, x_{m_i} - x_{l_i} \, \|.$$

Since f is bounded below, $f(x_{m_i}) - f(x_{l_i}) \to 0$ as $i \to +\infty$, so $\| \, x_{m_i} - x_{l_i} \, \| \to 0$ as $i \to +\infty$. Then, because q is uniformly continuous, $\| \, q(x_{m_i}) - q(x_{l_i}) \, \| < \eta$ for i sufficiently large. However,

$$\| \, q(x_{m_i}) \, \| \quad \leq \quad \| \, q(x_{m_i}) - q(x_{l_i}) \, \| + \| \, q(x_{l_i}) \, \| \quad \leq \quad 2\eta. \tag{5.4}$$

Since (5.4) must hold for any η, $0 < \eta < \varepsilon$, we have a contradiction (e.g., try $\eta = \frac{\varepsilon}{4}$).

6. Examples of pattern search methods for bound constrained minimization. A section of [16] is devoted to showing that each of the following four algorithms are pattern search methods for unconstrained minimization:

- coordinate search with fixed step lengths,
- evolutionary operation using two-level factorial designs [1, 3, 14],
- the original pattern search method of Hooke and Jeeves [9], and
- the multidirectional search algorithm of Dennis and Torczon [7, 15].

In this section we will discuss how these algorithms may be extended to bound constrained problems. We shall see that coordinate search and the pattern search method of Hooke and Jeeves extend without modification to the bound constrained case. Conversely, in the case of multidirectional search, we must require the initial basis matrix to be a diagonal matrix (in the unconstrained case, we can allow any nonsingular basis matrix); in addition, we must augment the columns of the generating matrix to ensure a sufficient set of search directions. In the case of evolutionary operation, we also must augment the columns of the generating matrix, which we do using a classical variant of factorial designs [2].

The difference between pattern search methods for unconstrained problems and bound constrained problems lies in the two additional conditions discussed in section 3.5. First, pattern search methods for bound constrained problems must start with a feasible iterate and choose feasible trial steps. Second, the core pattern BM_k must be defined by a diagonal matrix.

We assume that we begin with a feasible iterate; by design, pattern search methods for bound constrained problems thereafter accept only feasible iterates. Thus, the only thing we need to check is that the core pattern BM_k is defined by a diagonal matrix.

It is this latter condition that causes us to restrict the admissible choice of the basis matrix in multidirectional search and then augment the columns of the generating matrix. Moreover, G.E.P. Box's method of evolutionary operation using two-level factorial designs does not satisfy this diagonality condition; in section 2 we presented a simple counterexample that showed how evolutionary operation can fail as a consequence in the bound constrained case.

6.1. Coordinate search and the pattern search method of Hooke and Jeeves. Coordinate search and the pattern search method of Hooke and Jeeves extend to bound constrained problems without change. In both cases the basis matrix B is typically chosen to be a diagonal matrix: either the identity or a matrix whose entries reflect the relative scaling of the variables. Furthermore, the first 3^n columns of C_k, which are fixed for all iterations k of both algorithms, are composed of all possible combinations of $\{-1, 0, 1\}$. In [16] these columns are organized so that the first $2n$ consist of the identity matrix I and its negative $-I$. In terms of our formalism, then, $M_k = I$ for all iterations k. It follows that BM_k is a diagonal matrix, as required.

6.2. Evolutionary operation using factorial design. In section 2 a simple example sufficed to show that evolutionary operation cannot be used for bound constrained minimization without alteration. In terms of our formalism, the problem is the following: For the evolutionary operation algorithm using factorial designs, the basis matrix B is usually selected to be the identity or a diagonal matrix chosen so that the entries along the diagonal represent the relative scaling among the variables. However, this convention is not sufficient to ensure that BM_k is a diagonal matrix. The problem lies with the generating matrix $C = [M \;\, -M \;\, L]$. (The generating matrix C is fixed across all iterations of evolutionary operation.) The generating matrix contains in its columns all possible combinations of $\{-1, 1\}$ to which is appended a column of zeros. Clearly, no subset of n columns of C can be chosen to form a diagonal matrix M.

As noted in section 2, one remedy would be to use a composite design [2]. An example of such a design that satisfies the requirements of the bound constrained global convergence theory would be to choose M to be the diagonal matrix with entries of 2 along the diagonal. These $2n$ columns augment the original pattern of factorial design. This was illustrated in Figure 2.3.

6.3. Multidirectional search. The reader should be forewarned that our description and discussion of multidirectional search take a point of view that is ostensibly at odds with the formalism of section 3.1. The generating matrix Γ is viewed as fixed; typically $\Gamma = [M \;\, -M] \equiv [I \;\, -I]$. The basis matrix, on the other hand, is viewed as varying from iteration to iteration so that B_k corresponds to the edges in the current simplex that are adjacent to the current iterate x_k. This is the reverse of the discussion in section 3.1, where B is fixed and Γ_k varies. However, the former view of multidirectional search is not incompatible with the formalism of pattern search methods, as noted in [16], and as we shall have reason to discuss here.

The extension of multidirectional search to problems with bound constraints requires us to restrict the choice of a starting simplex and to augment the columns of the generating matrix.

The first restriction is minor and is usually satisfied by the customary choices made in practice. In multidirectional search, the columns of B_0 are formed from the edges of an initial simplex adjacent to the initial iterate x_0, which is one of the $n+1$ vertices of the simplex. In the case of bound constraints, we restrict the starting simplex to be a right-angled simplex; i.e., the vertices of the simplex are x_0 and the points $x_0 + \alpha_i e_i$, where $\alpha_i \in \mathbf{R}$ and $i = 1, \ldots, n$. Because of this choice, $B_0 = \mathrm{diag}(\alpha_i)$. Since $M \equiv I$, the product $B_0 M$ is a diagonal matrix.

However, even if the initial simplex is restricted to be a right-angled simplex so that $B_0 M$ is diagonal, there is no guarantee that in subsequent iterations $B_k M$ will be diagonal. To understand why this is so, and how this may be corrected by augmenting the columns of the generating matrix, we need to discuss how multidirectional search fits within the formalism of pattern search methods. These details are absent from [16], so we present them here.

At iteration k, the basis matrix is

$$B_k = \left[b_k^1 \cdots b_k^n \right] = \left[(v_k^1 - v_k^0) \cdots (v_k^n - v_k^0) \right],$$

where v_k^i, $i = 0, \ldots, n$, are the vertices of the simplex associated with multidirectional search at this iteration. Define

$$T_i = \begin{cases} I, & i = 0, \\ -\left(I - e_i e_i^T - \sum_{m=1}^n e_i e_m^T\right), & i = 1, \ldots, n. \end{cases}$$

Now consider what happens in the next iteration. If the iteration is unsuccessful, then $v_{k+1}^0 = v_k^0$ and the new basis for the pattern, which is determined by the edges of the simplex emanating from v_{k+1}^0, is

$$B_{k+1} = B_k = B_k T_0.$$

If, on the other hand, the iteration is successful, then $v_{k+1}^0 = v_k^0 - (v_k^j - v_k^0)$ for some $j \in \{1, \ldots, n\}$, and the new basis will be the set of vectors

$$b_{k+1}^i = \begin{cases} b_k^j & \text{if } i = j, \\ -b_k^i + b_k^j & \text{otherwise.} \end{cases}$$

In this case,

$$B_{k+1} = B_k T_j.$$

Thus, in general,

$$B_{k+1} = B_k T_{j_{k+1}}, \tag{6.1}$$

and so

$$B_k = B_{k-1} T_{j_k} = B_{k-2} T_{j_{k-1}} T_{j_k} = \cdots = B_0 \prod_{i=1}^k T_{j_i}. \tag{6.2}$$

Our next goal is to simplify this relation further.

First note that

$$T_\ell e_i = \begin{cases} e_\ell & \text{if } i = \ell, \\ e_\ell - e_i & \text{if } i \neq \ell. \end{cases} \tag{6.3}$$

Let $E(i, \ell)$ denote the elementary permutation matrix that swaps the ith and ℓth columns when acting on matrices from the right; we have

$$E(i, \ell) = I - e_i e_i^T - e_\ell e_\ell^T + e_\ell e_i^T + e_i e_\ell^T.$$

Using (6.3), we find that if $i \neq \ell$, then

$$T_\ell E(i, \ell) = T_\ell + e_i e_i^T - e_i e_\ell^T \tag{6.4}$$

and

$$\left(T_i \left(-T_\ell\right)\right) e_i = e_\ell. \tag{6.5}$$

Meanwhile, a short calculation shows that for $i, \ell = 1, \ldots, n$,

$$T_i T_\ell = I - e_\ell e_\ell^T - \sum_{m=1}^n e_\ell e_m^T - e_i e_i^T + \delta_\ell^i e_i e_\ell^T + \delta_\ell^i \sum_{m=1}^n e_i e_m^T + e_i e_\ell^T,$$

where δ^i_ℓ is the Kronecker delta. If $i = \ell$, this reduces to

$$T_i T_i = I, \tag{6.6}$$

and if $i \neq \ell$, using (6.4) we obtain

$$\begin{aligned} T_i T_\ell &= I - e_\ell e_\ell^T - \sum_{m=1}^{n} e_\ell e_m^T - e_i e_i^T + e_i e_\ell^T \\ &= -T_\ell - e_i e_i^T + e_i e_\ell^T = -T_\ell E(i,\ell). \end{aligned} \tag{6.7}$$

From (6.6) and (6.7) we obtain the rule

$$T_i T_\ell = \begin{cases} I & \text{if } i = \ell, \\ -T_\ell E(i,\ell) & \text{otherwise.} \end{cases} \tag{6.8}$$

We can then use (6.8) to reduce (6.2) to

$$B_k = \pm B_0 T_{\ell_k} \Pi_k$$

for some T_{ℓ_k} and permutation matrix Π_k.

This relationship reveals several things. The first is that it reconciles the usual description of multidirectional search with the formal abstract definition of a pattern search method; the pattern matrix is given by

$$B_k C = \pm B_0 T_{j_k} \Pi_k [I \ -I \ 0] = B_0 [T_{j_k} \ \ -T_{j_k} \ 0] \Pi_k \equiv B C_k. \tag{6.9}$$

That is, we may interpret multidirectional search in terms of a fixed basis B and a changing generating matrix C_k.

We can also see that while $B\Gamma_0$ will be diagonal, this diagonality may be lost in subsequent iterations. However, the form of the generic pattern from the unconstrained algorithm suggests one way to circumvent this problem in the bound constrained case. This remedy will, moreover, preserve the geometric interpretation of the pattern in multidirectional search in terms of a simplex.

First, if we ignore the permutation in (6.9), which affects only column ordering, the pattern at iteration k in the unconstrained case is given by

$$B_k C \equiv B C_k = B_0 [T_{j_k} \ \ -T_{j_k} \ 0].$$

Suppose we augment the columns of C to include all the T_i:

$$C \ = \ [-T_0 \ -T_1 \ \cdots \ -T_n \ 0].$$

At any iteration k, up to a column permutation, the basis matrix is the matrix $B_k = \pm B T_{j_k}$, $j_k \in \{0, \ldots, n\}$. When we then form the pattern $P_k = \Delta_k B_k C$, we have

$$P_k \ = \ \Delta_k B_k C \ = \ \Delta_k B [\pm T_{j_k} T_0 \ \pm T_{j_k} T_1 \ \cdots \ \pm T_{j_k} T_n \ 0] \ \equiv \ \Delta_k B C_k.$$

Now note that (6.5) means that for $j_k \neq l$, the j_kth column of $-T_{j_k} T_\ell$ is the ℓth basis vector. Consequently, we are guaranteed that by a permutation of the columns of C_k,

$$C_k = [I \ -I \ L_k] \equiv [\Gamma \ L_k],$$

where L_k changes at each iteration, but Γ does not. Since we require the initial simplex to be a right-angled simplex, we may then be assured that $B\Gamma = [\mathrm{diag}(\alpha_i) \; -\mathrm{diag}(\alpha_i)]$, as required.

Moreover, this augmentation of C and the search through its columns can be implemented in a way that preserves the relationship of the pattern to the moving simplex that characterizes multidirectional search. This is possible because the matrices T_i, $i = 0, \ldots, n$, capture how the basis changes in association with a change of simplex. This is the gist of (6.1). The implications for any implementation of this modification to multidirectional search to handle bound constraints will appear elsewhere.

7. Conclusion.

We have presented a reasonable extension of pattern search methods for unconstrained minimization to bound constrained problems. The extension is supported by a global convergence theory as strong as that for the unconstrained case. The generalization imposes few additional requirements, and as we have seen in section 6, the classical pattern search methods for unconstrained minimization or straightforward variants thereof carry over to the bound constrained case.

One issue we have not discussed is that of identifying active constraints, as in [4, 5]. One would wish to show that if the sequence $\{x_k\}$ converges to a nondegenerate stationary point x_*, then in a finite number of iterations the iterates x_k land on the constraints active at x_* and remain thereafter on those constraints.

There are three difficulties in proving such a result for pattern search methods for bound constrained minimization. The first is minor. If the iterates x_k are to identify the active constraints for a stationary point on the boundary of the feasible region, we must ensure that the lattice manifest in Theorem 3.2 actually allows iterates to land on the boundary. This requires additional but straightforward conditions on such quantities as x_0, τ, Δ_0, and the pattern matrices P_k (see, for instance, [17]). A related but more subtle difficulty is that the relative sizes of the steps in the core pattern and the remaining points in the pattern must obey certain relations in order to ensure that the algorithm does not take a purely interior approach to a point on the boundary. This rules out, for instance, certain of the composite designs suggested by Box and Wilson [2].

The most serious obstacle is showing that ultimately the iterates will land on the active constraints and remain there. For algorithms such as those considered in [4, 5], this is not a problem because the explicit use of the gradient impels the iterates to do this in the neighborhood of a nondegenerate stationary point. However, pattern search methods do not have this information. On the other hand, the kinship of pattern search methods and gradient projection methods makes us hopeful that ultimately we will be able to prove that pattern search methods also identify the active constraints in a finite number of iterations.

One can also extend pattern search methods to linearly constrained minimization [11]. The specification of pattern search methods for handling general linear inequalities is more involved, and the analysis is lengthier and more complicated. For bound constrained problems the analysis is enormously simplified because of the straightforward geometry of the feasible region and the fact that we know the explicit form of the projected gradient.

Acknowledgments. We wish to thank Margaret Wright for her helpful comments, particularly her suggestion that we include an introductory example that provides some intuition about how these methods work. We also wish to thank Chen Xin for catching an error in an earlier version of the paper.

REFERENCES

[1] G. E. P. Box, *Evolutionary operation: A method for increasing industrial productivity*, Appl. Statist., 6 (1957), pp. 81–101.

[2] G. E. P. Box and K. B. Wilson, *On the experimental attainment of optimum conditions*, J. Roy. Statist. Soc. Ser. B, XIII (1951), pp. 1–45.

[3] M. J. Box, D. Davies, and W. H. Swann, *Non-Linear Optimization Techniques*, ICI Monograph 5, Oliver & Boyd, Edinburgh, Scotland, 1969.

[4] J. V. Burke and J. J. Moré, *On the identification of active constraints*, SIAM J. Numer. Anal., 25 (1988), pp. 1197–1211.

[5] P. H. Calamai and J. J. Moré, *Projected gradient methods for linearly constrained problems*, Math. Programming, 39 (1987), pp. 93–116.

[6] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460.

[7] J. E. Dennis, Jr. and V. Torczon, *Direct search methods on parallel machines*, SIAM J. Optim., 1 (1991), pp. 448–474.

[8] J. C. Dunn, *Global and asymptotic convergence rate estimates for a class of projected gradient processes*, SIAM J. Control Optim., 19 (1981), pp. 368–400.

[9] R. Hooke and T. A. Jeeves, *Direct search solution of numerical and statistical problems*, J. Assoc. Comput. Mach., 8 (1961), pp. 212–229.

[10] D. L. Keefer, *Simpat: Self-bounding direct search method for optimization*, Indust. Engrg. Chem. Process Design Develop., 12 (1973), pp. 92–99.

[11] R. M. Lewis and V. J. Torczon, *Pattern search methods for linearly constrained minimization*, SIAM J. Optim., to appear.

[12] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Comput. J., 7 (1965), pp. 308–313.

[13] W. Spendley, G. R. Hext, and F. R. Himsworth, *Sequential application of simplex designs in optimisation and evolutionary operation*, Technometrics, 4 (1962), pp. 441–461.

[14] W. H. Swann, *Direct search methods*, in Numerical Methods for Unconstrained Optimization, W. Murray, ed., Academic Press, London, New York, 1972, pp. 13–28.

[15] V. Torczon, *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1989; available as Tech. report 90-07, Department of Computational and Applied Mathematics, Rice University, Houston, TX.

[16] V. Torczon, *On the convergence of pattern search algorithms*, SIAM J. Optim., 7 (1997), pp. 1–25.

[17] M. W. Trosset and V. Torczon, *Numerical Optimization Using Computer Experiments*, ICASE Report No. 97-38, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, 1997.

SIAM J. OPTIM.
Vol. 9, No. 4, pp. 1100–1127

NEWTON'S METHOD FOR LARGE BOUND-CONSTRAINED OPTIMIZATION PROBLEMS*

CHIH-JEN LIN† AND JORGE J. MORɇ

To John Dennis on the occasion of his 60th birthday.

Abstract. We analyze a trust region version of Newton's method for bound-constrained problems. Our approach relies on the geometry of the feasible set, not on the particular representation in terms of constraints. The convergence theory holds for linearly constrained problems and yields global and superlinear convergence without assuming either strict complementarity or linear independence of the active constraints. We also show that the convergence theory leads to an efficient implementation for large bound-constrained problems.

Key words. bound-constrained optimization, preconditioned conjugate gradients, projected gradients, strict complementarity

AMS subject classifications. 65F10, 90C06, 90C30

PII. S1052623498345075

1. Introduction.

We analyze a trust region version of Newton's method for the optimization problem

$$\min \{ f(x) : x \in \Omega \} , \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable mapping on the bound-constrained set

$$\Omega = \{ x \in \mathbb{R}^n : l \le x \le u \}. \tag{1.2}$$

Our analysis relies on the geometry of Ω and applies, without change, to the case where Ω is the linearly constrained set

$$\Omega = \{ x \in \mathbb{R}^n : l_i \le \langle c_i, x \rangle \le u_i, \ i \in \mathcal{I} \}. \tag{1.3}$$

The convergence theory yields results that are independent of the representation of Ω in terms of constraints; in particular, we assume neither strict complementarity (nonzero multipliers) nor linear independence of the active constraints.

Our main interest is in algorithms for large optimization problems. Thus the convergence theory that we develop emphasizes algorithms that use iterative techniques to solve the trust region subproblem while retaining superlinear convergence of the trust region method. We show, in particular, how the convergence theory leads to an efficient implementation of Newton's method when the feasible set Ω is the bound-constrained set (1.2).

*Received by the editors September 23, 1998; accepted for publication (in revised form) May 21, 1999; published electronically September 24, 1999. This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under contract W-31-109-Eng-38.

http://www.siam.org/journals/siopt/9-4/34507.html

†Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan (cjlin@csie.ntu.edu.tw). The work of this author was supported in part by National Science Council of Taiwan grant NSC-88-2213-E-002-097 and National Science Foundation grant CCR-9321550.

‡Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 (more@mcs.anl.gov).

Our development of a convergence theory for Newton's method yields three main results. We first establish *global convergence* to a stationary point; that is, if $\{x_k\}$ is the sequence generated by the trust region method, then every limit point of the sequence is a stationary point for problem (1.1). We then establish the *identification properties* of the algorithm by showing that if $\{x_k\}$ converges to some x^*, then there is an integer k_0 such that x_k lands in the face *exposed* by $-\nabla f(x^*)$ for all $k \geq k_0$. Finally, we establish the *local convergence* properties of the algorithm. The main result shows that if a strong second-order sufficiency condition holds at a limit point x^* of the trust region iterates, then the whole sequence $\{x_k\}$ converges to x^* at a superlinear rate.

Global and superlinear convergence for linearly constrained problems has been established, in almost all cases, under the assumption of strict complementarity. Moreover, the algorithms that have been analyzed usually require the exact solution of systems of linear equations. See, for example, [2, 22, 33, 18] for algorithms that use ϵ-active constraints, [23, 20] for active set methods, [13, 25, 12, 21] for trust region methods, and [9, 16, 11, 10] for interior-point methods. In recent work Heinkenschloss, Ulbrich, and Ulbrich [24] analyzed an interior-point method without assuming strict complementarity, but they proved only local convergence.

Lescrenier [25] and Facchinei and Lucidi [19] were the first to analyze algorithms for bound-constrained problems that are superlinearly convergent without assuming strict complementarity. Lescrenier analyzes the trust region method of Conn, Gould, and Toint [13]. Facchinei and Lucidi analyze a line search algorithm based on a differentiable exact penalty function that, unlike the algorithms for bound-constrained problems that we have reviewed, generates iterates that need not be feasible.

We analyze a trust region method for the linearly constrained optimization problem (1.3) based on the convergence theory of Moré [27] and Burke, Moré, and Toraldo [7]. The analysis relies on the geometric approach of Burke and Moré [6] for general linearly constrained problems. We use projected searches [30] during the subspace minimization phase, and thus we are able to add many constraints during this phase. We show that global and superlinear convergence hold even if strict complementarity fails for the general linearly constrained optimization problem (1.3).

The convergence theory for trust region methods presented in section 2 depends on the definition of the Cauchy step s_k^C. The main result in this section shows that global convergence to a stationary point is guaranteed if the step s_k in the trust region method achieves a fraction of the reduction achieved by the Cauchy step.

The standard development of identification properties for an optimization algorithm shows that the active set settles down if the iterates converge to a stationary point x^*. This approach is not possible if strict complementarity does not hold at x^*. In section 3 we show that the sequence generated by the trust region method is trapped by the face exposed by $-\nabla f(x^*)$; section 3 provides a precise definition of the face of a convex set exposed by a vector. If strict complementarity holds at x^*, this result implies that the active set settles down.

In section 3 we also explore the concept of strict complementarity and its relationship to the concept of an exposed face. In this paper we use the term *nondegenerate stationary point* x^* if strict complementarity holds at x^* or, equivalently, if x^* is in the relative interior of the face exposed by $-\nabla f(x^*)$.

Section 4 defines the projected searches that are used to explore the current face of the feasible set. Projected searches are an important ingredient of the optimization algorithm because they allow wider latitude in the choice of the next iterate. In particular, the active constraints are allowed to change arbitrarily while requiring only the approximate solution of a linear system.

Section 5 contains the major convergence results for the trust region Newton's method. We show that if a strong second-order sufficiency condition holds at a limit point x^* of the trust region iterates, then the whole sequence $\{x_k\}$ converges to x^*. Previous results assumed strict complementarity and that the problem was bound-constrained. We also show that if the sequence $\{x_k\}$ converges to x^*, then the rate of convergence is at least superlinear.

Section 6 briefly outlines the implementation of TRON (version 1.0), a trust region Newton method for bound-constrained problems. Interesting features of this implementation include the use of projected searches and a preconditioned conjugate gradient method to determine the minor iterates and the use of a limited-memory preconditioner. We use the incomplete Cholesky factorization icfs of Lin and Moré [26] as a preconditioner since this factorization does not require the choice of a drop tolerance, and the amount of storage can be specified in advance.

Section 7 presents the results of a comparison between TRON and the LANCELOT [14] and L-BFGS-B [36] codes. These results show that on the problems described in this section, TRON is generally more efficient, in terms of computing time, than LANCELOT and L-BFGS-B. Caution must be exercised in drawing conclusions from these results since, as noted in section 7, there are many differences between TRON and LANCELOT.

2. Trust region methods. In this section we present a trust region method for the solution of optimization problems subject to linear constraints, but we emphasize the case where Ω is the bound-constrained set (1.2). The algorithm that we present was proposed by Moré [27] as a modification of the algorithm of Toint [35]. The development in this section follows Moré [27] and Burke, Moré, and Toraldo [7].

At each iteration of a trust region method there is an approximation $x_k \in \Omega$ to the solution, a bound Δ_k, and a model $\psi_k : \mathbb{R}^n \to \mathbb{R}$ of the possible reduction $f(x_k + w) - f(x_k)$ for $\|w\| \le \Delta_k$. We assume that the model ψ_k is the quadratic

$$\psi_k(w) = \langle \nabla f(x_k), w\rangle + \tfrac{1}{2}\langle w, B_k w\rangle$$

for some symmetric matrix B_k. The matrix B_k is arbitrary for many of the results, but the rate of convergence results usually requires that B_k be the Hessian matrix $\nabla^2 f(x_k)$. Of course, it is possible to choose $B_k = 0$, and then the model is linear.

The description of the algorithm in terms of the quadratic ψ_k is appropriate when we are interested in the step s_k. However, we also use the quadratic

$$q_k(x) = \psi_k(x - x_k) = \langle \nabla f(x_k), x - x_k\rangle + \tfrac{1}{2}\langle x - x_k, B_k(x - x_k)\rangle$$

to describe the algorithm in terms of the iterates x_k.

The iterate x_k and the bound Δ_k are updated according to rules that are standard in trust region methods for unconstrained minimization. Given a step s_k such that $x_k + s_k \in \Omega$ and $\psi_k(s_k) < 0$, these rules depend on the ratio

$$\rho_k = \frac{f(x_k + s_k) - f(x_k)}{\psi_k(s_k)} \tag{2.1}$$

of the actual reduction in the function to the predicted reduction in the model. Since the step s_k is chosen so that $\psi_k(s_k) < 0$, a step with $\rho_k > 0$ yields a reduction in the function. Given $\eta_0 > 0$, the iterate x_k is updated by setting

$$x_{k+1} = \begin{cases} x_k + s_k & \text{if} \quad \rho_k > \eta_0, \\ x_k & \text{if} \quad \rho_k \le \eta_0. \end{cases} \tag{2.2}$$

Any step s_k with $\rho_k > \eta_0$ is *successful*; otherwise the step in *unsuccessful*. Under suitable conditions, all steps (iterations) are eventually successful.

Updating rules for Δ_k depends on positive constants η_1 and η_2 with $\eta_1 < \eta_2 < 1$, while the rate at which Δ_k is updated depends on positive constants σ_1, σ_2, and σ_3 such that $\sigma_1 < \sigma_2 < 1 < \sigma_3$. The trust region bound Δ_k is updated by setting

$$\begin{array}{lll} \Delta_{k+1} \in [\sigma_1 \min\{\|s_k\|, \Delta_k\}, \sigma_2 \Delta_k] & \text{if} & \rho_k \leq \eta_1, \\ \Delta_{k+1} \in [\sigma_1 \Delta_k, \sigma_3 \Delta_k] & \text{if} & \rho_k \in (\eta_1, \eta_2), \\ \Delta_{k+1} \in [\Delta_k, \sigma_3 \Delta_k] & \text{if} & \rho_k \geq \eta_2. \end{array} \tag{2.3}$$

Similar rules are used in most modern trust region methods.

We choose a step s_k that gives as much reduction in the model ψ_k as the Cauchy step s_k^C generated by the gradient projection method applied to the subproblem

$$\min \{\psi_k(w) : x_k + w \in \Omega, \ \|w\| \leq \Delta_k\}.$$

The Cauchy step s_k^C is of the form $s_k(\alpha_k)$, where the function $s_k : \mathbb{R} \mapsto \mathbb{R}^n$ is defined by

$$s_k(\alpha) = P[x_k - \alpha \nabla f(x_k)] - x_k,$$

where $P : \mathbb{R}^n \mapsto \Omega$ is the projection into the feasible set Ω. If Ω is the bound-constrained set (1.2), then the projection can be computed with at most $2n$ comparisons by

$$P(x) = \operatorname{mid}(l, x, u),$$

where mid$(\cdot)$ is the componentwise median (middle) of the three vectors in the argument. The trust region method that we describe can be implemented efficiently if there is an efficient algorithm for computing the projection P.

The scalar α_k that determines the Cauchy step s_k^C is chosen so that $s_k(\alpha_k)$ produces a sufficient reduction. We require that

$$\psi_k(s_k(\alpha_k)) \leq \mu_0 \langle \nabla f(x_k), s_k(\alpha_k) \rangle, \qquad \|s_k(\alpha_k)\| \leq \mu_1 \Delta_k, \tag{2.4}$$

for positive constants μ_0 and μ_1 such that $\mu_0 < \frac{1}{2}$. We also require that there are positive constants γ_1, γ_2, and γ_3 such that

$$\alpha_k \in [\gamma_1, \gamma_3] \quad \text{or} \quad \alpha_k \in [\gamma_2 \widetilde{\alpha}_k, \gamma_3],$$

where $\widetilde{\alpha}_k > 0$ satisfies

$$\psi_k(s_k(\widetilde{\alpha}_k)) \geq (1 - \mu_0) \langle \nabla f(x_k), s_k(\widetilde{\alpha}_k) \rangle \quad \text{or} \quad \|s_k(\widetilde{\alpha}_k)\| \geq \mu_1 \Delta_k.$$

The requirements on the Cauchy step s_k^C can be satisfied [27, 7] with a finite number of evaluations of ψ_k. For additional details, see section 6.

We have described the requirements on the Cauchy step s_k^C in terms of the quadratic ψ_k, but we could also use q_k. In particular,

$$q_k(x_k + s_k^C) \leq q_k(x_k) + \mu_0 \left\langle \nabla q_k(x_k), s_k^C \right\rangle$$

is the sufficient reduction condition (2.4).

Given the Cauchy step s_k^C, we require that the step s_k satisfy

$$\psi_k(s_k) \le \mu_0 \psi_k(s_k^C), \qquad \|s_k\| \le \mu_1 \Delta_k, \qquad x_k + s_k \in \Omega. \tag{2.5}$$

This requirement is quite natural and can always be satisfied by choosing $s_k = s_k^C$. However, this choice is likely to lead to slow convergence, because the method would then reduce to a version of steepest descent. In the next section we explore other choices that lead to superlinear and quadratic convergence.

Algorithm 2.1 summarizes the computations required to implement the trust region method. We assume that $f : \mathbb{R}^n \mapsto \mathbb{R}$ is continuously differentiable on Ω and that $\Delta_0 > 0$ has been specified.

ALGORITHM 2.1 (Trust region method).

```
For $k = 0, \dots,$
    Compute the model $\psi_k$.
    Compute the Cauchy step $s_k^C$.
    Compute a step $s_k$ that satisfies (2.5).
    Compute the ratio $\rho_k$ and update $x_k$ by (2.2).
    Update $\Delta_k$ according to (2.3).
```

Burke, Moré, and Toraldo [7] analyzed the trust region method of Algorithm 2.1 in terms of the Cauchy point

$$x_k^C \equiv P\left[x_k + \alpha_k \nabla f(x_k)\right] = x_k + s_k^C.$$

Convergence results depend on a bound on the predicted decrease for the quadratic ψ_k. This bound is based on the inequality

$$-\left\langle \nabla f(x_k), s_k^C \right\rangle \ge \kappa_0 \left[\frac{\|x_k^C - x_k\|}{\alpha_k}\right] \min\left\{\Delta_k, \frac{1}{\|B_k\|}\left[\frac{\|x_k^C - x_k\|}{\alpha_k}\right]\right\}, \tag{2.6}$$

where κ_0 is a positive constant. This bound was obtained by Moré [27]. Other bounds obtained for problems with bound constraints and, more generally, convex constraints [13, 35, 12] do not yield the same information because they are not expressed in terms of the Cauchy point.

The choice of s_k^C is an important ingredient in the trust region method. Our choice of s_k^C is simple and can be implemented efficiently provided there is an efficient algorithm for computing the projection P. For other choices, see [13, 35, 12].

Many of the convergence results in Burke, Moré, and Toraldo [7] are expressed in terms of the *projected gradient*

$$\nabla_\Omega f(x) \equiv P_{T(x)}\left[-\nabla f(x)\right] = \operatorname{argmin}\{\|v + \nabla f(x)\| : v \in T(x)\},$$

where the *tangent cone* $T(x)$ is the closure of the cone of all feasible directions at $x \in \Omega$, and Ω is a general convex set. The term projected gradient is not entirely appropriate. Indeed, since

$$\min\left\{\langle \nabla f(x),\ v\rangle : v \in T(x), \|v\| \le 1\right\} = -\|\nabla_\Omega f(x)\|, \tag{2.7}$$

it might be more appropriate to call $\nabla_\Omega f(x)$ the *projected steepest descent direction.* The optimality property (2.7) follows from the properties of the projection on convex cones; Calamai and Moré [8] provide a direct proof of (2.7).

The projected gradient should not be confused with the *reduced gradient.* When Ω is the bound-constrained set (1.2), the reduced gradient is the vector with components $\partial_i f(x)$ if $l_i < x_i < u_i$, while for the projected gradient

$$-[\nabla_\Omega f(x)]_i = \begin{cases} \partial_i f(x) & \text{if } x_i \in (l_i, u_i), \\ \min\{\partial_i f(x), 0\} & \text{if } x_i = l_i, \\ \max\{\partial_i f(x), 0\} & \text{if } x_i = u_i \end{cases} \tag{2.8}$$

if $l_i < u_i$, with $[\nabla_\Omega f(x)]_i = 0$ in the exceptional case where $l_i = u_i$. The appearance of the minus sign in this expression for the projected gradient is only a minor nuisance because in our work we need only an expression for $\|\nabla_\Omega f(x)\|$.

The projected gradient $\nabla_\Omega f$ can be used to characterize stationary points because if Ω is a convex set, then $x \in \Omega$ is a stationary point of problem (1.1) if and only if $\nabla_\Omega f(x) = 0$. In general, $\nabla_\Omega f$ is discontinuous, but as proved by Calamai and Moré [8], if $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable on Ω, then the mapping $x \mapsto \|\nabla_\Omega f(x)\|$ is lower semicontinuous on Ω. This property implies that if $\{x_k\}$ is a sequence in Ω that converges to x^*, and if $\{\nabla_\Omega f(x_k)\}$ converges to zero, then x^* is a stationary point of problem (1.1). In section 3 we show that the continuity properties of the projected gradient are closely associated with the behavior of the optimization algorithm.

Theorem 2.1. *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be continuously differentiable on a closed, convex set Ω, and let $\{x_k\}$ be the sequence generated by the trust region method. Assume that $\{B_k\}$ is uniformly bounded. If x^* is a limit point of $\{x_k\}$, then there is a subsequence $\{x_{k_i}\}$ of successful steps that converges to x^* with*

$$\lim_{i\to\infty} \|\nabla_\Omega f(x_{k_i}^C)\| = 0. \tag{2.9}$$

Moreover, $\{x_{k_i}^C\}$ also converges to x^*, and thus x^* is a stationary point for problem (1.1).

This result is due to Burke, Moré, and Toraldo [7, Theorem 5.5]. Similar convergence results for bound-constrained and linearly constrained optimization algorithms assert that every limit point of the algorithm is stationary, but they do not yield any information on the projected gradient; in sections 3 and 5 we show that (2.9) in Theorem 2.1 plays an important role in the convergence analysis. For a sampling of recent convergence results, see [12, 18, 9, 16, 20, 33].

3. Exposing constraints.

Identification properties are an important component of the convergence analysis of an algorithm for linearly constrained problems. We show that if x^* is a stationary point and Ω is the polyhedral set (1.3), then the iterates $\{x_k\}$ generated by the trust region method tend to lie in the face exposed by the direction $-\nabla f(x^*)$.

The notion of an exposed face arises in convex analysis, where the face of a convex set Ω exposed by the vector $d \in \mathbb{R}^n$ is

$$E[d] \equiv \operatorname{argmax}\left\{x \in \Omega : \langle d, x\rangle\right\}.$$

A short computation shows that when $\Omega = [l, u]$ is the bound-constrained set (1.2) and $d = -\nabla f(x^*)$, then

$$E\left[-\nabla f(x^*)\right] = \{x \in [l, u] : x_i = l_i \text{ if } \partial_i f(x^*) > 0 \text{ and } x_i = u_i \text{ if } \partial_i f(x^*) < 0\}$$

is the face of (1.2) exposed by the direction $-\nabla f(x^*)$. A similar expression holds if Ω is the polyhedral set defined by (1.3). If x^* is a stationary point of the optimization problem (1.1), then there are Lagrange multipliers such that

$$\nabla f(x^*) = \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* c_i,$$

where λ_i^* is unrestricted in sign if $l_i = u_i$, but

$$\lambda_i^* \geq 0 \;\text{ if }\; \langle c_i, x^* \rangle = l_i, \qquad \lambda_i^* \leq 0 \;\text{ if }\; \langle c_i, x^* \rangle = u_i,$$

and $\mathcal{A}(x)$ is the set of active constraints at $x \in \Omega$ defined by

$$\mathcal{A}(x) = \{ i \in \mathcal{I} : \langle c_i, x \rangle \in \{ l_i, u_i \} \}.$$

Since this definition of the active set does not distinguish between lower and upper bounds, we avoid this problem by interpreting the inclusion $\mathcal{A}(x) \subset \mathcal{A}(y)$ to mean

$$\mathcal{A}_l(x) \subset \mathcal{A}_l(y), \qquad \mathcal{A}_u(x) \subset \mathcal{A}_u(y),$$

where

$$\mathcal{A}_l(x) = \{ i \in \mathcal{I} : \langle c_i, x \rangle = l_i \}, \qquad \mathcal{A}_u(x) = \{ i \in \mathcal{I} : \langle c_i, x \rangle = u_i \}.$$

With this interpretation, if $\langle c_i, x \rangle = l_i$ and $\mathcal{A}(x) \subset \mathcal{A}(y)$, then $\langle c_i, y \rangle = l_i$. For most results we need to know only that $\langle c_i, x \rangle \in \{ l_i, u_i \}$, and then the first definition of the active set is suitable.

The face exposed by $-\nabla f(x^*)$ is determined by the nonzero multipliers. Indeed, a computation based on the definition of a face shows that

$$E\,[-\nabla f(x^*)] = \{ x \in \Omega : \langle c_i, x \rangle = l_i \text{ if } \lambda_i^* > 0 \text{ and } \langle c_i, x \rangle = u_i \text{ if } \lambda_i^* < 0 \}. \tag{3.1}$$

Note that this expression for $E\,[-\nabla f(x^*)]$ is valid for any choice of Lagrange multipliers.

Burke and Moré [6] provide additional information on exposed faces. In particular, they note that for Ω convex, x^* is a stationary point for the optimization problem (1.1) if and only if $x^* \in E\,[-\nabla f(x^*)]$.

Dunn [17] defines x^* to be a nondegenerate stationary point if $-\nabla f(x^*)$ lies in the relative interior of the normal cone

$$N(x^*) = \{ u \in \mathbb{R}^n : \langle u, y - x^* \rangle \leq 0, \; y \in \Omega \}.$$

Burke and Moré [6] relate nondegeneracy to the geometry of $E\,[-\nabla f(x^*)]$ by proving that x^* is nondegenerate if and only if x^* lies in the relative interior of the face $E\,[-\nabla f(x^*)]$. These two definitions rely only on the geometry of Ω. If Ω is expressed in terms of constraints, then nondegeneracy can be shown [5] to be equivalent to the existence of a set of nonzero Lagrange multipliers. Thus, a stationary point x^* is nondegenerate as defined by Dunn [17] if and only if strict complementarity holds at x^*. We can also show [6, Theorem 5.3] that

$$x \in E\,[-\nabla f(x^*)] \qquad \Longleftrightarrow \qquad \mathcal{A}(x^*) \subset \mathcal{A}(x) \tag{3.2}$$

whenever x^* is nondegenerate. Thus, for nondegenerate problems, landing in the face $E\,[-\nabla f(x^*)]$ can be described in terms of active sets.

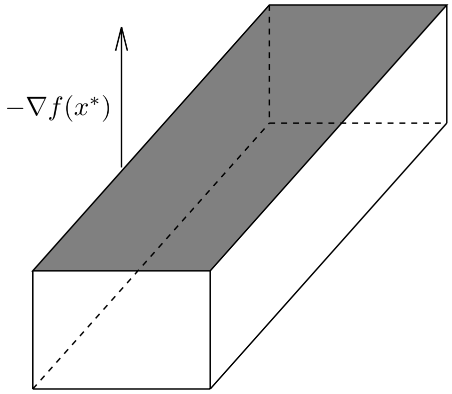


FIG. 3.1. *The exposed face $E\,[-\nabla f(x^*)]$ for a degenerate problem.*

Figure 3.1 illustrates some of the properties of exposed faces. In this case x^* is in the relative boundary of the face, so this problem is degenerate. Note that in this case (3.2) fails because $\mathcal{A}(x^*)$ may not be a subset of $\mathcal{A}(x)$ for $x \in E\,[-\nabla f(x^*)]$. Finally, note that $x - y$ is orthogonal to $\nabla f(x^*)$ whenever x and y are in $E\,[-\nabla f(x^*)]$. This last observation holds for any convex set Ω because the mapping $x \mapsto \langle \nabla f(x^*), x\rangle$ is constant on $E\,[-\nabla f(x^*)]$.

For nondegenerate problems we can show that eventually all iterates land in the relative interior of $E\,[-\nabla f(x^*)]$. For degenerate problems this is not possible, but we can show that eventually all iterates land in $E\,[-\nabla f(x^*)]$. We first prove a technical result that shows that if $\{x_k\}$ is any sequence that converges to a stationary point x^*, and x_k lands in $E\,[-\nabla f(x^*)]$, then x_k^C remains in $E\,[-\nabla f(x^*)]$. We need the following result of Burke and Moré [6, Theorem 4.2].

THEOREM 3.1. *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be continuously differentiable on the polyhedral set Ω, and let $\{x_k\}$ be any sequence in Ω that converges to a stationary point x^*. Then*

$$\lim_{k\to+\infty} \|\nabla_\Omega f(x_k)\| = 0$$

if and only if there is an index k_0 with $x_k \in E\,[-\nabla f(x^*)]$ for $k \ge k_0$.

Theorem 3.1 is of interest because it provides a means to show that iterates land in the exposed face $E\,[-\nabla f(x^*)]$. Note that in this result $\{x_k\}$ can be any sequence in Ω. We now show that if x_k lands in $E\,[-\nabla f(x^*)]$, then x_k^C remains in $E\,[-\nabla f(x^*)]$.

THEOREM 3.2. *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be continuously differentiable on the polyhedral set Ω, and let $\{x_k\}$ be any sequence that converges to a stationary point x^*. If x_k is in $E\,[-\nabla f(x^*)]$ for $k \ge k_0$, then*

$$P\,[x_k - \alpha_k \nabla f(x_k)] \in E\,[-\nabla f(x^*)]$$

for k sufficiently large.

Proof. The proof relies on Theorem 3.1 of Burke and Moré [6], which shows that for any sequence $\{d_k\}$ in $\mathbb{R}^n$ that converges to d^*

$$E[d_k] \subset E[d^*] \tag{3.3}$$

for all k sufficiently large. If $N(x)$ is the normal cone at $x \in \Omega$, the definition of the projection operator implies that

$$x_k - \alpha_k \nabla f(x_k) - P[x_k - \alpha_k \nabla f(x_k)] \in N(P[x_k - \alpha_k \nabla f(x_k)]).$$

The definition of the exposed face shows that $x \in E[d]$ if and only if $d \in N(x)$, and thus

$$P[x_k - \alpha_k \nabla f(x_k)] \in E[-\alpha_k \nabla f(x_k) + x_k - P[x_k - \alpha_k \nabla f(x_k)]] = E[d_k], \tag{3.4}$$

where we have defined the sequence $\{d_k\}$ by

$$d_k = -\nabla f(x_k) + \frac{x_k - P[x_k - \alpha_k \nabla f(x_k)]}{\alpha_k}.$$

We now claim that

$$\left\| \frac{P[x_k - \alpha_k \nabla f(x_k)] - x_k}{\alpha_k} \right\| \le \|\nabla_\Omega f(x_k)\|. \tag{3.5}$$

If we accept this claim, we can complete the proof by noting that, since $\{x_k\}$ converges to x^* and $x_k \in E\,[-\nabla f(x^*)]$, Theorem 3.1 and inequality (3.5) show that the sequence $\{d_k\}$ converges to $-\nabla f(x^*)$. Hence, (3.3) and (3.4) imply that $P[x_k - \alpha_k \nabla f(x_k)]$ belongs to $E\,[-\nabla f(x^*)]$ for all k sufficiently large.

The proof of (3.5) requires two inequalities. First note that the optimality property (2.7) of the projected gradient $\nabla_\Omega f$ implies that

$$-\langle \nabla f(x), v \rangle \le \|\nabla_\Omega f(x)\| \; \|v\|,$$

for any feasible direction v at x. In particular,

$$-\langle \nabla f(x), s(\alpha) \rangle \le \|\nabla_\Omega f(x)\| \; \|s(\alpha)\|,$$

where we have defined $s(\alpha) = P[x - \alpha \nabla f(x)] - x$. Next, note that the definition of the projection operator, $\langle P(x) - x, y - P(x) \rangle \ge 0$ for any $y \in \Omega$, implies that

$$-\langle \nabla f(x), s(\alpha) \rangle \ge \frac{\|s(\alpha)\|^2}{\alpha}.$$

The last two displayed inequalities imply (3.5) as desired. □

We want to show that all iterates eventually stay in the exposed face $E\,[-\nabla f(x^*)]$. Theorems 2.1 and 3.1 show that if the sequence $\{x_k\}$ converges to x^*, then x_k^C lands in $E\,[-\nabla f(x^*)]$ for some subsequence of successful iterates. We now restrict the step s_k so that the next iterate does not leave $E\,[-\nabla f(x^*)]$. The following result makes use of the observation that

$$x \in E\,[-\nabla f(x^*)], \quad \mathcal{A}(x) \subset \mathcal{A}(y) \qquad \Longrightarrow \qquad y \in E\,[-\nabla f(x^*)].$$

This observation follows directly from the expression (3.1) for $E\,[-\nabla f(x^*)]$.

Theorem 3.3. *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be continuously differentiable on the polyhedral set Ω, and let $\{x_k\}$ be the sequence generated by the trust region method. Assume that $\{B_k\}$ is uniformly bounded and that the step s_k satisfies*

$$\mathcal{A}(x_k^C) \subset \mathcal{A}(x_k + s_k), \qquad k \ge 0. \tag{3.6}$$

If $\{x_k\}$ converges to x^*, then there is an index k_0 such that

$$x_k \in E\,[-\nabla f(x^*)], \quad x_k + s_k \in E\,[-\nabla f(x^*)], \qquad k \ge k_0.$$

Proof. Theorem 2.1 shows that there is a sequence $\mathcal{K}$ of successful iterates such that if $k \in \mathcal{K}$, then $\{x_k^C\}$ converges to x^* and $\{\nabla_\Omega f(x_k^C)\}$ converges to zero. Hence, Theorem 3.1 shows that

$$x_k^C \in E\,[-\nabla f(x^*)], \qquad k \in \mathcal{K}.$$

Since every iterate in $\mathcal{K}$ is successful, assumption (3.6) implies that $x_{k+1} = x_k + s_k$ belongs to $E\,[-\nabla f(x^*)]$. In particular, there is an index k_0 such that x_{k_0} belongs to $E\,[-\nabla f(x^*)]$. We now show that x_k belongs to $E\,[-\nabla f(x^*)]$ for all $k \geq k_0$.

Assume that x_k belongs to $E\,[-\nabla f(x^*)]$ for some $k \geq k_0$. Theorem 3.2 shows that $x_k^C \in E\,[-\nabla f(x^*)]$. Hence, assumption (3.6) on the step yields that $x_k + s_k$ is in $E\,[-\nabla f(x^*)]$. If $x_{k+1} = x_k$, then x_{k+1} clearly belongs to $E\,[-\nabla f(x^*)]$, while if $x_{k+1} = x_k + s_k$, then we also have x_{k+1} in $E\,[-\nabla f(x^*)]$. Hence, in all cases x_{k+1} belongs to $E\,[-\nabla f(x^*)]$.

We have shown that $x_k \in E\,[-\nabla f(x^*)]$ for all $k \geq k_0$. Hence, Theorem 3.2 shows that $x_k^C \in E\,[-\nabla f(x^*)]$, and thus assumption (3.6) on the step yields that $x_k + s_k$ is in $E\,[-\nabla f(x^*)]$. □

4. Projected searches. The convergence theory of the trust region Newton method depends on generating the step s_k so that conditions (2.5) and (3.6) are satisfied. We determine s_k by computing $m+1$ minor iterates $x_{k,1}, \ldots, x_{k,m+1}$, where $x_{k,1} = x_k^C$. We require that

$$x_{k,j} \in \Omega, \qquad \mathcal{A}(x_k^C) \subset \mathcal{A}(x_{k,j}), \qquad \|x_{k,j} - x_k\| \leq \mu_1 \Delta_k, \tag{4.1}$$

and that the decrease condition

$$q_k(x_{k,j+1}) \leq q_k(x_{k,j}), \qquad 1 \leq j \leq m, \tag{4.2}$$

be satisfied. If the step is defined by $s_k = x_{k,m+1} - x_k$, then (2.5) and (3.6) are satisfied. Also note that there is no loss in generality in fixing m independent of the iteration; this imposes only an upper bound on the number of minor iterates because we can set $x_{k,j+1} = x_{k,j}$.

We can compute minor iterates that satisfy (4.1) and (4.2) by computing a descent direction for the subproblem

$$\min\left\{q_k(x_{k,j} + w) : \langle c_i, w\rangle = 0,\ i \in \mathcal{A}(x_{k,j})\right\}. \tag{4.3}$$

Given a descent direction $w_{k,j}$ with $\langle c_i, w_{k,j}\rangle = 0$ for $i \in \mathcal{A}(x_{k,j})$, we examine q_k in the ray $x_{k,j} + \beta w_{k,j}$, with $\beta \geq 0$, and use a line search to choose $\beta_{k,j}$ so that q_k is minimized. The minor iterate $x_{k,j+1} = x_{k,j} + \beta_{k,j} w_{k,j}$ may not be acceptable either because $x_{k,j+1}$ is not feasible or because $x_{k,j+1}$ does not satisfy the trust region constraint $\|x_{k,j+1} - x_k\| \leq \Delta_k$. Thus, if necessary, we modify $\beta_{k,j}$ so that both constraints are satisfied.

Instead of using a line search to determine $x_{k,j+1}$, we can use a projected search along the path defined by $P[x_{k,j} + \beta w_{k,j}]$. The advantage of this approach is that we would be able to add several constraints at once. For a line search we normally require a decrease of q_k on the line segment $[x_{k,j}, x_{k,j+1}]$, but for a projected search we need only require a decrease at $x_{k,j+1}$ with respect to the base point $x_{k,j}$. We require that

$$q_k(x_{k,j+1}) \leq q_k(x_{k,j}) + \mu_0 \min\left\{\langle \nabla q_k(x_{k,j}), x_{k,j+1} - x_{k,j}\rangle, 0\right\}. \tag{4.4}$$

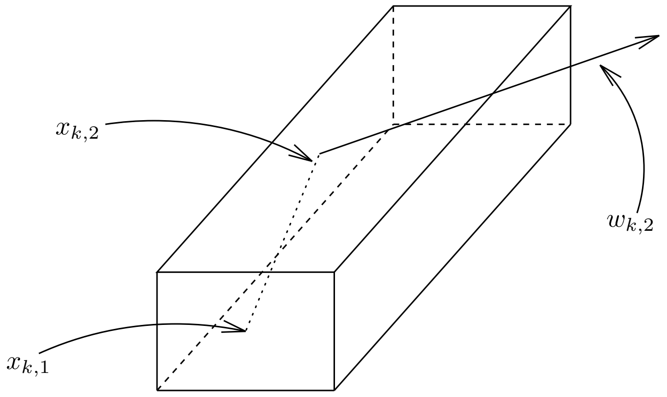


FIG. 4.1. *The minor iterates for a projected search.*

In most cases we require only (4.2), but for rate of convergence results we need (4.4). For additional details on projected searches, see Moré and Toraldo [30, section 4].

Figure 4.1 illustrates the projected search when Ω is the bound-constrained set (1.2). In this figure the iterate $x_{k,2}$ has been computed and a direction $w_{k,2}$ is determined that is orthogonal to the active constraint normals. If a line search is used, the search would be restricted to points in the ray $x_{k,2} + \beta w_{k,2}$ that lie in the feasible region. With a projected search, the search would continue along the piecewise linear path $P[x_{k,2} + \beta w_{k,2}]$. In either case, we require only that $x_{k,3}$ satisfy the decrease condition (4.4).

When Ω is the bound-constrained set (1.2), Lescrenier [25] determines the step s_k by computing minor iterates, but he requires that the line segment $\alpha x_{k,j+1}+(1-\alpha)x_{k,j}$ be feasible for $\alpha \in [0,1]$ and that

$$q_k(x_{k,j+1}) \le q_k(\alpha x_{k,j+1} + (1-\alpha)x_{k,j}), \qquad \alpha \in [0,1]. \tag{4.5}$$

This requirement can be satisfied if a line search is used to choose the minor iterates, but it rules out the projected searches that we have proposed. Also note that assumption (4.5) on the minor iterates is stronger than (4.2). This observation can be verified by proving that if $\phi : \mathbb{R} \mapsto \mathbb{R}$ is a quadratic on $[0,1]$ with $\phi'(0) < 0$, and $\phi(1) \le \phi(\alpha)$ for α in $[0,1]$, then

$$\phi(1) \le \phi(0) + \tfrac{1}{2}\phi'(0) \le \phi(0) + \mu\phi'(0)$$

for any $\mu \in [0, \frac{1}{2}]$.

5. Convergence results. We have been analyzing the trust region method under the assumption that $\{B_k\}$ is uniformly bounded. We now consider a trust region version of Newton's method so that B_k is the Hessian matrix $\nabla^2 f(x_k)$. The assumption that $\{B_k\}$ is uniformly bounded is then satisfied if Ω is bounded or, more generally, if $\nabla^2 f$ is bounded on the level set

$$\mathcal{L}(x_0) \equiv \{x \in \Omega : f(x) \le f(x_0)\}.$$

We also assume that Ω is the polyhedral set (1.3).

The local convergence analysis for the trust region version of Newton's method requires that we assume that some subsequence of the iterates $\{x_k\}$ generated by the trust region method converges to a stationary point x^* that satisfies a regularity condition. We assume that the Hessian matrix $\nabla^2 f(x^*)$ is positive definite on the subspace

$$S(x^*) = \operatorname{aff}\{E\,[-\nabla f(x^*)] - x^*\}, \tag{5.1}$$

where $\operatorname{aff}\{S\}$ denotes the affine hull of the set S. Thus, we require that the Hessian matrix be positive definite on the smallest subspace that contains $E\,[-\nabla f(x^*)] - x^*$. In the convergence analysis we use this regularity condition in the equivalent form

$$\langle v, \nabla^2 f(x^*) v \rangle \geq \kappa \|v\|^2, \qquad v \in S(x^*), \qquad \kappa > 0. \tag{5.2}$$

The *strong second-order sufficiency condition* (5.2) is equivalent to the standard second-order sufficiency condition if x^* is nondegenerate, but it is stronger than the standard second-order sufficiency condition for degenerate problems.

The strong second-order condition (5.2) is satisfied if $\nabla^2 f(x^*)$ is positive definite on the subspace

$$\left\{ v \in \mathbb{R}^n : \langle c_j, v \rangle = 0,\ j \in \mathcal{B}(x^*) \right\}, \tag{5.3}$$

where $\mathcal{B}(x^*)$ is the set of *strictly binding* constraints

$$\mathcal{B}(x^*) = \{ i \in \mathcal{I} : \lambda_i^* > 0 \text{ if } \langle c_i, x^* \rangle = l_i \text{ and } \lambda_i^* < 0 \text{ if } \langle c_i, x^* \rangle = u_i \}.$$

Gay [23], Lescrenier [25], and Robinson [32] use this condition in their work. A disadvantage of working with (5.3) is that $\mathcal{B}(x^*)$ depends on the representation of Ω and the choice of multipliers. On the other hand, (5.2) depends only on the geometry of Ω.

Burke and Moré [6] provide additional information on the regularity condition (5.2). In particular, they present an example where (5.2) holds but the Hessian matrix is not positive definite on (5.3).

The strong second-order sufficiency condition simplifies considerably when Ω is the bound-constrained set (1.2). In this case (5.2) requires that $\nabla^2 f(x^*)$ be positive definite on the subspace

$$S(x^*) = \{ w \in \mathbb{R}^n : w_i = 0,\ i \in \mathcal{B}(x^*) \}$$

of vectors orthogonal to the strictly binding constraints

$$\mathcal{B}(x^*) = \left\{ i \in \mathcal{A}(x^*) : \partial_i f(x^*) \neq 0 \right\}.$$

THEOREM 5.1. *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be twice continuously differentiable on the polyhedral set Ω, and let $\{x_k\}$ be the sequence generated by the trust region Newton method. Assume that $\nabla^2 f$ is bounded on the level set $\mathcal{L}(x_0)$ and that the step s_k satisfies* (3.6). *If $\{x_k\}$ has a limit point x^* that satisfies the strong second-order sufficiency condition* (5.2), *then $\{x_k\}$ converges to x^*.*

Proof. We first claim that (5.2) implies that x^* is an isolated limit point of $\{x_k\}$. This claim follows by noting that (5.2) implies that x^* is an isolated stationary point, and that every limit point of $\{x_k\}$ is stationary.

The proof is by contradiction. If we assume that $\{x_k\}$ does not converge to x^*, then Lemma 4.10 of Moré and Sorensen [29] shows that when x^* is an isolated limit point of $\{x_k\}$, there is a subsequence $\mathcal{K}$ such that $\{x_k\}$ converges to x^* for $k \in \mathcal{K}$, and an $\epsilon > 0$ with

$$\|x_{k+1} - x_k\| \geq \epsilon, \qquad k \in \mathcal{K}.$$

In particular, $\|s_k\| \geq \epsilon$ for $k \in \mathcal{K}$. We now prove that if the sequence $\{w_k\}$ is defined by

$$w_k = \frac{s_k}{\|s_k\|}, \qquad k \in \mathcal{K},$$

then any limit point w^* is a feasible direction at x^*. Note that $\|s_k\| \geq \epsilon$ implies that $x_k + \tau w_k$ belongs to Ω for τ in $[0, \epsilon]$, and hence $x^* + \tau w^*$ also belongs to Ω. This shows that w^* is a feasible direction at x^*.

We now show that $\langle \nabla f(x^*), w^* \rangle = 0$. Note that requirements (2.4), (2.5), and (2.6) on s_k show that if the iteration is successful, then

$$(5.4)\quad f(x_k) - f(x_{k+1}) \geq \eta_0 \mu_0 \kappa_0 \left[\frac{\|x_k^C - x_k\|}{\alpha_k} \right] \min \left\{ \Delta_k, \frac{1}{\|\nabla^2 f(x_k)\|} \left[\frac{\|x_k^C - x_k\|}{\alpha_k} \right] \right\}.$$

Our assumptions guarantee that the Hessian matrices $\nabla^2 f(x_k)$ are bounded, and since $\|s_k\| \leq \mu_1 \Delta_k$, and $\|s_k\| \geq \epsilon$ for $k \in \mathcal{K}$, the trust region bounds Δ_k are bounded away from zero. Hence, inequality (5.4) implies that

$$\lim_{k \in \mathcal{K}, k \to \infty} \frac{\|x_k^C - x_k\|}{\alpha_k} = 0.$$

Moreover, since $\{\alpha_k\}$ is bounded above, $\{\|x_k^C - x_k\|\}$ also converges to zero for $k \in \mathcal{K}$. Hence, Lemma 5.1 in Burke, Moré, and Toraldo [7] implies that

$$\lim_{k \in \mathcal{K}, k \to \infty} \left\| \nabla_\Omega f(x_k^C) \right\| = 0.$$

Theorem 3.1 now shows that x_k^C is in $E\left[-\nabla f(x^*)\right]$ for $k \in \mathcal{K}$, and thus assumption (3.6) on the step s_k implies that $x_k + s_k$ belongs to $E\left[-\nabla f(x^*)\right]$ for $k \in \mathcal{K}$. In particular,

$$\langle \nabla f(x^*), (x_k + s_k - x^*) \rangle = 0, \qquad k \in \mathcal{K}.$$

A computation using $\|s_k\| \geq \epsilon$ now shows that $\langle \nabla f(x^*), w^* \rangle = 0$.

We have shown that w^* is a feasible direction at x^* with $\langle \nabla f(x^*), w^* \rangle = 0$. Thus, w^* belongs to $S(x^*)$, and $\left\langle w^*, \nabla^2 f(x^*) w^* \right\rangle > 0$. On the other hand, $\psi_k(s_k) \leq 0$ implies that

$$\tfrac{1}{2} \|s_k\| \left\langle w_k, \nabla^2 f(x_k) w_k \right\rangle \leq - \langle \nabla f(x_k), w_k \rangle .$$

Since $\{x_k\}$ converges to x^*, $\{w_k\}$ converges to w^*, and $\|s_k\| \geq \epsilon$ for $k \in \mathcal{K}$, this inequality implies that

$$0 < \tfrac{1}{2} \epsilon \left\langle w^*, \nabla^2 f(x^*) w^* \right\rangle \leq - \langle \nabla f(x^*), w^* \rangle = 0.$$

This contradiction proves the result. □

Theorems 5.1 improves on previous convergence results for linearly constrained optimization algorithms because it does not assume strict complementarity. For recent convergence results, see [19, 12, 18, 9, 16, 20, 33].

Rate of convergence results depend on showing that eventually the trust region bound is not active. These results require additional assumptions on the step s_k. We assume that the minor iterates satisfy (4.1) and the decrease condition (4.4). We now estimate the decrease of the quadratic q_k if the minor iterates satisfy (4.4). The following result appears in Moré [28], but for completeness we provide the proof.

LEMMA 5.2. *Assume that* $\phi : \mathbb{R} \mapsto \mathbb{R}$ *is twice differentiable on* $[0, 1]$ *and that* $\phi''(\alpha) \geq \varepsilon$ *on* $[0, 1]$ *for some* $\varepsilon > 0$. *If*

$$\phi(1) \leq \phi(0) + \mu\phi'(0) \tag{5.5}$$

for some $\mu \in (0, 1)$, *then*

$$\phi(0) - \phi(1) \geq \frac{\mu}{2(1-\mu)}\varepsilon.$$

Proof. The mean value theorem shows that

$$\phi(1) = \phi(0) + \phi'(0) + \tfrac{1}{2}\phi''(\theta)$$

for some $\theta \in (0, 1)$, and thus (5.5) implies that $\frac{1}{2}\phi''(\theta) \leq (1-\mu)(-\phi'(0))$. Hence,

$$\phi(0) - \phi(1) \geq \mu(-\phi'(0)) \geq \frac{\mu}{2(1-\mu)}\phi''(\theta) \geq \frac{\mu}{2(1-\mu)}\varepsilon,$$

as desired. □

If we assume that the sequence $\{x_k\}$ converges to x^*, then Theorem 3.3 guarantees that all iterates belong to $E[-\nabla f(x^*)]$, and hence (4.1) shows that all the minor iterates also belong to $E[-\nabla f(x^*)]$. Now define

$$\phi(\alpha) = q_k\left(\alpha x_{k,j+1} + (1-\alpha)x_{k,j}\right)$$

and note that the decrease condition (4.4) guarantees that

$$q_k(x_{k,j+1}) \leq q_k(x_{k,j}) + \mu_0\langle \nabla q_k(x_{k,j}), x_{k,j+1} - x_{k,j}\rangle,$$

and thus (5.5) holds. Hence, if we assume that the strong second-order condition (5.2) holds, then Lemma 5.2 implies that there is a $\kappa_0 > 0$ such that

$$q_k(x_{k,j}) - q_k(x_{k,j+1}) \geq \kappa_0\|x_{k,j+1} - x_{k,j}\|^2. \tag{5.6}$$

We need this estimate for our next result.

THEOREM 5.3. *Let* $f : \mathbb{R}^n \mapsto \mathbb{R}$ *be twice continuously differentiable on the polyhedral set* Ω, *and let* $\{x_k\}$ *be the sequence generated by the trust region Newton method. Assume that* $\{x_k\}$ *converges to a solution* x^* *of* (1.1) *that satisfies the regularity condition* (5.2). *If the minor iterates satisfy* (4.1) *and* (4.4), *then there is an index* k_0 *such that all steps* s_k *with* $k \geq k_0$ *are successful and the trust region bound* Δ_k *is bounded away from zero.*

Proof. In the proof we bound $|\rho_k - 1|$, where ρ_k is defined by (2.1), and we show that the bounds converge to zero; the rules for updating Δ_k then show that all steps

s_k are ultimately successful and that Δ_k is bounded away from zero. We begin by noting that

$$\rho_k - 1 = \frac{f(x_k+s_k) - f(x_k) - \psi_k(s_k)}{\psi_k(s_k)}. \tag{5.7}$$

The denominator of (5.7) is estimated by noting that (5.6) implies that the decrease generated by s_k satisfies

$$\begin{aligned} -\psi_k(s_k) = q_k(x_k) - q_k(x_k+s_k) &\geq \kappa_0 \sum_{j=0}^{m} \|x_{k,j+1} - x_{k,j}\|^2 \\ &\geq \kappa_0 \max_{0 \leq j \leq m} \left\{ \|x_{k,j+1} - x_{k,j}\|^2 \right\}. \end{aligned}$$

On the other hand,

$$\|s_k\| \leq \sum_{j=0}^{m} \|x_{k,j+1} - x_{k,j}\| \leq (m+1) \max_{0 \leq j \leq m} \left\{ \|x_{k,j+1} - x_{k,j}\| \right\}.$$

Hence, $-\psi_k(s_k) \geq \kappa_1 \|s_k\|^2$ for $\kappa_1 = \kappa_0/(m+1)^2$. We estimate the numerator of (5.7) by noting that the mean value theorem implies that

$$|f(x_k+s_k) - f(x_k) - \psi_k(s_k)| \leq \sigma_k \|s_k\|^2,$$

where

$$\sigma_k = \sup_{0 \leq \theta \leq 1} \left\{ \|\nabla^2 f(x_k + \theta s_k) - \nabla^2 f(x_k)\| \right\}.$$

These estimates show that $|\rho_k - 1| \leq \sigma_k/\kappa_0$, so that our result will be established if we show that $\{\sigma_k\}$ converges to zero.

Since $\{x_k\}$ converges to x^*, the sequence $\{\sigma_k\}$ converges to zero if $\{s_k\}$ converges to zero. Theorem 3.3 shows that x_k and $x_k + s_k$ belong to $E\left[-\nabla f(x^*)\right]$, and thus the definition (5.1) implies that $s_k \in S(x^*)$. In particular, $s_k = P_{S(x^*)} s_k$, where $P_{S(x^*)}$ is the orthogonal projection onto $S(x^*)$. Since $\psi_k(s_k) \leq 0$,

$$\tfrac{1}{2} \left\langle s_k, \nabla^2 f(x_k) s_k \right\rangle \leq - \left\langle \nabla f(x_k), s_k \right\rangle,$$

and thus $s_k = P_{S(x^*)} s_k$ and the regularity condition (5.2) imply that there is a $\nu_0 > 0$ with

$$\|s_k\| \leq \nu_0 \|P_{S(x^*)} \nabla f(x_k)\|.$$

The gradient $\nabla f(x^*)$ is orthogonal to $S(x^*)$ because $\langle \nabla f(x^*), x \rangle = \langle \nabla f(x^*), x^* \rangle$ whenever x is in $E\left[-\nabla f(x^*)\right]$, and since $\{x_k\}$ converges to x^*, this implies that $\{P_{S(x^*)} \nabla f(x_k)\}$ converges to zero. Thus, the previous estimate shows that $\{s_k\}$ converges to zero, as desired. □

Lescrenier [25] proved an analogous result, but he assumed that the feasible set was bound constrained, that the quadratic was decreasing on the line segment $[x_{k,j}, x_{k,j+1}]$, and that the minor iterates satisfied (4.5). In particular, his result did not cover projected searches. Our assumptions in Theorem 5.3 are considerably weaker.

When the iterate x_k is far from the solution, the step s_k is usually determined because the trust region bound $\|x_{k,j} - x_k\| \le \mu_1 \Delta_k$ is encountered during the computation of $x_{k,j+1}$. However, as we converge, Theorem 5.3 shows that the trust region does not interfere with the computation of the step, so we are free to reduce q_k further by searching the feasible set.

We propose to compute the step s_k by computing minor iterates $x_{k,j}$ that satisfy (4.1) and the decrease condition (4.4). For each minor iterate $x_{k,j}$ let the columns of $Z_{k,j}$ form an orthonormal basis for the subspace

$$V_{k,j} = \{ w \in \mathbb{R}^n : \langle c_i, w \rangle = 0, \ i \in \mathcal{A}(x_{k,j}) \} .$$

Given $x_{k,j}$, we find an approximate minimizer of q_k on $x_{k,j} + V_{k,j}$. We require that if $x_{k,m+1}$ is the final iterate generated according to (4.1) and (4.4), then the step $s_k = x_{k,m+1} - x_k$ satisfies

$$\left\| Z_{k,m}^T [\nabla f(x_k) + \nabla^2 f(x_k) s_k] \right\| \le \xi_k \left\| Z_{k,m}^T \nabla f(x_k) \right\| , \qquad x_k + s_k \in \Omega. \tag{5.8}$$

We motivate these requirements by noting that if $\Psi_{k,m}(v) = q_k \left(x_{k,m} + Z_{k,m} v \right)$, then

$$\nabla \Psi_{k,m}(v) = Z_{k,m}^T [\nabla f(x_k) + \nabla^2 f(x_k)(x_{k,m} - x_k + Z_{k,m} v)],$$

where we have set $x_{k,0} = x_k$. Thus, the first condition in (5.8) is equivalent to finding $v_{k,m}$ such that

$$\| \nabla \Psi_{k,m}(v_{k,m}) \| \le \xi_k \left\| Z_{k,m}^T \nabla f(x_k) \right\| ,$$

and setting $s_k = x_{k,m} - x_k + Z_{k,m} v_{k,m}$. In particular, $x_{k,m+1} = x_{k,m} + Z_{k,m} v_{k,m}$ is a minimizer of q_k on $x_{k,m} + V_{k,m}$ if we choose $\xi_k = 0$.

At first sight it is not clear that we can always find a step that satisfies (5.8) since satisfying the first condition in (5.8) may violate the second condition. The simplest method of generating minor iterates $x_{k,j}$ that guarantees (5.8) is to set $x_{k,j+1}$ to the minimizer of q_k on $x_{k,j} + V_{k,j}$. With this choice $s_k = x_{k,j+1} - x_k$ satisfies the first condition in (5.8). If $x_k + s_k$ lies in Ω for this choice of $x_{k,j+1}$, then we are done. Otherwise, we can set $x_{k,j+1}$ to any point in Ω that satisfies (4.4) and such that $\mathcal{A}(x_{k,j+1})$ has at least one more active variable. This choice guarantees that, after computing at most n minor iterates, we reach a minor iterate with all variables active, and then (5.8) is trivially satisfied.

The procedure that we have outlined generates iterates $x_{k,j}$ that satisfy (4.1) and (4.4) with $\mathcal{A}(x_{k,j}) \subset \mathcal{A}(x_{k,j+1})$. The step $s_k = x_{k,m+1} - x_k$ satisfies (5.8), where $Z_{k,m}$ is defined by $x_{k,m}$. Geometrically this procedure searches for an approximate minimizer in the face defined by the active set $\mathcal{A}(x_{k,j})$, terminating if the approximate minimizer is on the relative interior of this face; otherwise, the search continues on the lower dimensional face defined by $\mathcal{A}(x_{k,j+1})$.

We have already noted that the step s_k is usually determined because the trust region bound $\|x_{k,j} - x_k\| \le \mu_1 \Delta_k$ is encountered during the computation of $x_{k,j+1}$. Thus, we need only assume that the step s_k satisfies (5.8) if $\|s_k\| \le \mu_* \Delta_k$ for some $\mu_* < \mu_1$.

Rate of convergence results when strict complementarity holds depend on the result that $\mathcal{A}(x_k) = \mathcal{A}(x^*)$ for all k sufficiently large. This result fails without strict complementarity. In this case the proof relies on showing that

$$V(x) \equiv \{ w \in \mathbb{R}^n : \langle c_i, w \rangle = 0, \ i \in \mathcal{A}(x) \} \subset S(x^*), \qquad x \in E\left[-\nabla f(x^*) \right]. \tag{5.9}$$

The subspace $V(x)$ is the largest subspace contained in the tangent cone $T(x)$.

For the rate of convergence results we assume that the sequence $\{x_k\}$ generated by the trust region Newton method converges to x^*. Theorems 3.2 and 3.3 show that x_k and x_k^C eventually land in $E\,[-\nabla f(x^*)]$ for all $k \geq k_0$. Since (4.1) guarantees that $\mathcal{A}(x_k^C)$ is a subset of $\mathcal{A}(x_{k,j})$ for any minor iterate $x_{k,j}$, we also have that $x_{k,j}$ is in $E\,[-\nabla f(x^*)]$. In particular, $x_{k,m} \in E\,[-\nabla f(x^*)]$. We shall need this result in the proof.

THEOREM 5.4. *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be twice continuously differentiable on the polyhedral set Ω, and let $\{x_k\}$ be the sequence generated by the trust region Newton method. Assume that $\{x_k\}$ converges to a solution x^* of* (1.1) *that satisfies the strong second-order sufficiency condition* (5.2). *If the step s_k is calculated by the algorithm outlined above, and* (5.8) *holds whenever $\|s_k\| \leq \mu_* \Delta_k$ for some $\mu_* < \mu_1$, then $\{x_k\}$ converges Q-linearly to x^* when ξ^* is sufficiently small, where*

$$\xi^* = \limsup_{k \to +\infty} \xi_k.$$

The rate of convergence is Q-superlinear when $\xi^* = 0$.

Proof. We first prove that (5.9) holds. The proof begins by noting that expression (3.1) for $E\,[-\nabla f(x^*)]$ shows that if λ_i^* are Lagrange multipliers, then

$$\{i : \lambda_i^* \neq 0\} \subset \mathcal{A}(x), \qquad x \in E\,[-\nabla f(x^*)].$$

Hence, if $w \in V(x)$, then $\langle \nabla f(x^*), w \rangle = 0$. Since any $w \in V(x)$ is a feasible direction, we also have that $x + \alpha w$ for all α sufficiently small. Hence, $\langle \nabla f(x^*), w \rangle = 0$ implies that $x + \alpha w$ belongs to $E\,[-\nabla f(x^*)]$. Moreover, since $x \in E\,[-\nabla f(x^*)]$ and $S(x^*)$ is a subspace,

$$\alpha w = ([x + \alpha w - x^*] - [x - x^*]) \in S(x^*).$$

Hence, $w \in S(x^*)$ as desired, and thus (5.9) holds.

We proved (5.9) for any $x \in E\,[-\nabla f(x^*)]$ because this result sheds light on the geometry behind the rate of convergence results, but for this proof we need only show that

$$V_{k,m} \subset S(x^*). \tag{5.10}$$

Since we have already noted that $x_{k,m} \in E\,[-\nabla f(x^*)]$, (5.9) implies that (5.10) holds.

We analyze the convergence rate in terms of the projection $P_k = Z_{k,m} Z_{k,m}^T$ onto the subspace $V_{k,m}$. Note, in particular, that since $V_{k,m}$ is a subspace of $S(x^*)$, an orthogonal basis for $V_{k,m}$ can be extended to a basis for $S(x^*)$, and thus

$$\|P_k w\| \leq \|P_{S(x^*)} w\|, \qquad w \in \mathbb{R}^n. \tag{5.11}$$

The main estimate needed for the rate of convergence result is obtained by noting that

$$\begin{aligned} \|P_k \nabla f(x_{k+1})\| \leq{}& \left\| P_k [\nabla f(x_{k+1}) - \nabla f(x_k) - \nabla^2 f(x_k) s_k] \right\| \\ &+ \left\| P_k [\nabla f(x_k) + \nabla^2 f(x_k) s_k] \right\|, \end{aligned}$$

assumption (5.8) on the step, and standard bounds yield that

$$\|P_k \nabla f(x_{k+1})\| \leq \varepsilon_k \|s_k\| + \xi_k \, \|P_k \nabla f(x_k)\| \tag{5.12}$$

for some sequence $\{\varepsilon_k\}$ converging to zero. Also note that the argument at the end of Theorem 5.3 shows that there is a constant ν_0 with

$$\|s_k\| \le \nu_0 \|P_{S(x^*)} \nabla f(x_k)\|. \tag{5.13}$$

If we make use of this estimate and (5.11) in (5.12) we obtain that

$$\limsup_{k\to+\infty} \frac{\|P_k \nabla f(x_{k+1})\|}{\|P_{S(x^*)} \nabla f(x_k)\|} \le \limsup_{k\to+\infty} \xi_k. \tag{5.14}$$

We complete the proof by estimating $\|P_k \nabla f(x_{k+1})\|$ and $\|P_{S(x^*)} \nabla f(x_k)\|$. We first show that

$$\|P_k \nabla f(x_{k+1})\| \ge (\nu_1 - \varepsilon_k)\|x_{k+1} - x^*\| \tag{5.15}$$

for some sequence $\{\varepsilon_k\}$ converging to zero.

The proof of (5.15) requires some preliminary results. We first show that $x_{k+1} - x^*$ is in $V_{k,m}$ for all k sufficiently large. This follows from the definition of $V_{k,m}$ because $\mathcal{A}(x_{k,m}) \subset \mathcal{A}(x_{k+1})$ and $\mathcal{A}(x_{k,m}) \subset \mathcal{A}(x^*)$. We also need to show that $P_k \nabla f(x^*) = 0$. This result follows because, as noted at the end of Theorem 5.3, $\nabla f(x^*)$ is orthogonal to $S(x^*)$, and since $V_{k,m}$ is a subspace of $S(x^*)$, we must also have $\nabla f(x^*)$ orthogonal to $V_{k,m}$. In particular, $P_k \nabla f(x^*) = 0$. The last result that we need for the proof of (5.15) is that

$$\|P_k \nabla^2 f(x^*) P_k v\| \ge \kappa \|v\|, \qquad v \in V_{k,m}. \tag{5.16}$$

To prove this result, note that if $v \in V_{k,m}$, then $P_k v = v$, and in view of (5.10), $P_k v$ is in $E\left[-\nabla f(x^*)\right]$. Hence, the regularity assumption (5.2) shows that (5.16) holds.

We now have all the ingredients to prove (5.15). Since $P_k \nabla f(x^*) = 0$,

$$P_k \nabla f(x_{k+1}) = P_k \nabla^2 f(x^*)(x_{k+1} - x^*) + P_k[\nabla f(x_{k+1}) - \nabla f(x^*) - \nabla^2 f(x^*)(x_{k+1} - x^*)],$$

and thus estimates of the last term show that

$$\left\|P_k \nabla^2 f(x^*)(x_{k+1} - x^*)\right\| \le \|P_k \nabla f(x_{k+1})\| + \varepsilon_k \|x_{k+1} - x^*\|,$$

where $\{\varepsilon_k\}$ converges to zero. Since $x_{k+1} - x^*$ is in $V_{k,m}$ for all k sufficiently large, (5.16) shows that

$$\|P_k \nabla^2 f(x^*) P_k (x_{k+1} - x^*)\| \ge \kappa \|x_{k+1} - x^*\|.$$

The last two inequalities show that (5.15) holds with $\nu_1 = \kappa$.

We estimate $\|P_{S(x^*)} \nabla f(x_k)\|$ by proving that

$$\|P_{S(x^*)} \nabla f(x_k)\| \le (\nu_2 + \varepsilon_k)\|x_k - x^*\| \tag{5.17}$$

for some sequence $\{\varepsilon_k\}$ converging to zero. Since $P_{S(x^*)} \nabla f(x^*) = 0$,

$$P_{S(x^*)} \nabla f(x_k) = P_{S(x^*)} \nabla^2 f(x^*)(x_k - x^*) + P_{S(x^*)}[\nabla f(x_k) - \nabla f(x^*) - \nabla^2 f(x^*)(x_k - x^*)],$$

and thus standard estimates of the last term show that

$$\|P_{S(x^*)} \nabla f(x_k)\| \le \left\|P_{S(x^*)} \nabla^2 f(x^*)(x_k - x^*)\right\| + \varepsilon_k \|x_k - x^*\|,$$

where $\{\varepsilon_k\}$ converges to zero. Since $P_{S(x^*)}(x_k - x^*) = x_k - x^*$, we obtain that

$$\|P_{S(x^*)}\nabla f(x_k)\| \le \nu_2 \|x_k - x^*\| + \varepsilon_k \|x_k - x^*\|, \qquad \nu_2 = \|P_{S(x^*)}\nabla^2 f(x^*) P_{S(x^*)}\|,$$

where $\{\varepsilon_k\}$ converges to zero. This proves (5.17).

Linear and superlinear convergence rates are obtained by noting that (5.14), together with estimates (5.15) and (5.17), show that

$$\limsup_{k\to+\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \le \left(\frac{\nu_2}{\nu_1}\right) \limsup_{k\to+\infty} \xi_k = \left(\frac{\nu_2}{\nu_1}\right) \xi^*.$$

Linear convergence takes place if $\nu_2 \xi^* < \nu_1$, and superlinear convergence holds if $\xi^* = 0$. □

A modification of the proof of Theorem 5.4 shows linear convergence for any $\xi^* < 1$ if the vectors $x_k - x^*$ lie in a fixed subspace V of $S(x^*)$ for all k sufficiently large. This result holds when x^* is nondegenerate (strict complementarity holds at x^*) since in this case $x_k - x^*$ belongs to $V(x_k) = S(x^*)$ for all k sufficiently large.

There are several interesting variations on Theorem 5.4. Note, in particular, that the minor iterate $x_{k,m}$ enters into the proof via the subspace $V_{k,m}$ and that the proof holds if P_k is a projection into any subspace of $S(x^*)$ that contains $x_{k+1} - x^*$. Thus we could have set P_k to the projection into $V(x_{k+1})$ and eliminated $x_{k,m}$ from the analysis. We did not make this simplification because with our choice of P_k the minor iterate $x_{k,m+1}$ is an approximate minimizer of q_k on $x_{k,m} + V_{k,m}$.

Lescrenier [25] and Facchinei and Lucidi [19] proved rate of convergence results without assuming strict complementarity, but the analysis was restricted to bound-constrained problems. Other convergence results for bound-constrained and linearly constrained optimization algorithms require strict complementarity. For recent convergence results, see [12, 18, 9, 16, 20, 33].

We can also show that quadratic convergence holds in Theorem 5.4 if we assume that $\nabla^2 f$ satisfies a Lipschitz condition at x^* and if

$$\xi_k \le \kappa_0 \|P_k \nabla f(x_k)\|, \qquad k \ge 0,$$

for a positive constant κ_0. With these assumptions we can follow the proof of Theorem 5.4. The main difference is that the inequality (5.12) can be replaced by

$$\|P_k \nabla f(x_{k+1})\| \le \kappa \|s_k\|^2 + \xi_k \|P_k \nabla f(x_k)\|,$$

where κ is the Lipschitz constant, and thus (5.11) and (5.13) yield that

$$\limsup_{k\to+\infty} \frac{\|P_k \nabla f(x_{k+1})\|}{\|P_{S(x^*)}\nabla f(x_k)\|^2} \le \kappa \nu_0^2 + \kappa_0.$$

The result now follows from estimates (5.15) and (5.17).

6. Implementation issues.

We now provide a brief outline of the implementation issues for a trust region Newton method for bound-constrained problems. We concentrate on discussing our choices for the trust region bound Δ_k, the Cauchy step, and the subspace step.

For the initial Δ_0 we used $\|\nabla f(x_0)\|$. This choice is appropriate in many cases, but more sophisticated choices are possible. We update the trust region bound Δ_k as outlined in section 2. We choose $\eta_0 = 10^{-3}$ in the algorithm (2.2) to update the current iterate; $\eta_1 = 0.25$, $\eta_2 = 0.75$ as the constants that determine when to increase or decrease the trust region Δ_k; and $\sigma_1 = 0.25$, $\sigma_2 = 0.5$, and $\sigma_3 = 4.0$ as the constants that govern the update of Δ_k in (2.3).

Given a step s_k, we attempt to choose Δ_{k+1} as $\alpha_k^* \|s_k\|$, where α_k^* is the minimum of a quadratic that interpolates the function $\alpha \mapsto f(x_k + \alpha s_k)$. In other words, we consider the quadratic ϕ such that

$$\phi(0) = f(x_k), \qquad \phi'(0) = \langle \nabla f(x_k), s_k \rangle, \qquad \phi(1) = f(x_{k+1})$$

and determine α_k^* as the minimum of this quadratic. If ϕ does not have a minimum, we set $\alpha_k^* = +\infty$. We choose Δ_{k+1} as $\alpha_k^* \|s_k\|$ if it falls in the desired interval; otherwise we set Δ_{k+1} to the closest endpoint.

The Cauchy step s_k^C is chosen by an iterative scheme that is guaranteed to terminate in a finite number of steps. Recall that the Cauchy step s_k^C is of the form $s_k(\alpha_k)$, where the function $s_k : \mathbb{R} \mapsto \mathbb{R}^n$ is defined by

$$s_k(\alpha) = P\left[x_k - \alpha \nabla f(x_k)\right] - x_k$$

and α_k satisfies the conditions specified in section 2. The simplest scheme is to set $\alpha_k^{(0)}$ to a constant and then generate a sequence $\{\alpha_k^{(l)}\}$ of trial values by decreasing the trial values by a constant factor until the sufficient decrease condition (2.4) is satisfied. We use a more sophisticated scheme. Given $\alpha_k^{(0)}$, we generate a sequence $\{\alpha_k^{(l)}\}$ of trial values. The sequence can be either increasing or decreasing, but in all cases we require that

$$\alpha_k^{(l+1)} \in \left[\beta_1 \alpha_k^{(l)}, \beta_2 \alpha_k^{(l)}\right],$$

where $\beta_1 \le \beta_2 < 1$ for a decreasing sequence and $1 < \beta_1 \le \beta_2$ for an increasing sequence. The decision to generate an increasing sequence or a decreasing sequence depends of the initial $\alpha_k^{(0)}$. If the initial $\alpha_k^{(0)}$ fails to satisfy the sufficient decrease condition (2.4), we decrease the trial values until (2.4) fails, and we set α_k to the last trial value that satisfies (2.4). If the initial $\alpha_k^{(0)}$ satisfies (2.4), we increase the trial values until (2.4) fails, and we set α_k to the last trial value that satisfies (2.4).

We use $\alpha_k^{(0)} = 1$ on the first iteration, but on all other iterations we use α_{k-1}. We use $\mu_0 = 10^{-2}$ and $\mu_1 = 1.0$ in the sufficient decrease condition (2.4).

The minor iterates generated in the trust region method are required to satisfy conditions (4.1) and (4.4). We generate the step between the minor iterates along the lines specified in section 4 but specialized to the case of bound constraints. Specifically, we compute the step from the trust region subproblem

$$\min\left\{q(x+w) : w_i = 0,\ i \in \mathcal{A}(x),\ \|Dw\| \le \Delta\right\},$$

where D is a scaling matrix. If $i_1, \ldots, i_m$ are the indices of the free variables, and the matrix Z is defined as the matrix in $\mathbb{R}^{n \times m}$ whose kth column is the i_kth column of the identity matrix in $\mathbb{R}^{n \times n}$, then this subproblem is equivalent to

$$\min\{q_{\scriptscriptstyle F}(v) : \|DZv\| \le \Delta\},$$

where $q_{\scriptscriptstyle F}$ is the quadratic in the free variables defined by

$$q_{\scriptscriptstyle F}(v) \equiv q(x + Zv) - q(x) = \tfrac{1}{2} v^T A v + r^T v.$$

The matrix A and the vector r are, respectively, the reduced Hessian matrix of q and reduced gradient of q_F at x with respect to the free variables.

Given a descent direction w for this subproblem, a projected line search guarantees that we can determine $\beta > 0$ such that the next iterate $x_+ = P[x + \beta w]$ satisfies conditions (4.1) and (4.4). The conditions in (4.1) are satisfied for any $\beta > 0$ provided D has a condition number that is bounded independent of the iterate. We use $\mu_0 = 10^{-2}$ in the sufficient decrease condition (4.4).

We generate the descent direction w with a preconditioned conjugate gradient method as suggested by Steihaug [34]. The conjugate gradient iterates are generated until the trust region is violated, a negative curvature direction is generated, or the convergence condition (5.8) is satisfied. As noted in section 5, this condition can be satisfied by choosing the minor iterates so that $\mathcal{A}(x_{k,j}) \subset \mathcal{A}(x_{k,j+1})$. For additional details, see the discussion in Lin and Moré [26].

In our algorithms we choose D from an incomplete Cholesky factorization. From a theoretical viewpoint, the choice of D is not important, but the numerical results are strongly dependent on the choice of D. We use the incomplete Cholesky factorization icfs of Lin and Moré [26]. The icfs incomplete Cholesky factorization does not require the choice of a drop tolerance. Moreover, the amount of storage used by the factorization can be specified in advance as $p \cdot n$, where p is set by the user and n is the number of variables. In our numerical results we use $p = 5$.

7. Computational experiments.

We now compare the performance of an implementation TRON (version 1.0) of the trust region method outlined in section 6 with the LANCELOT [14] and L-BFGS-B [36] codes. All computational experiments were done with the -O optimization compiler option on a Sun UltraSPARC2 workstation with 1,024 MB RAM.

LANCELOT implements Newton's method with a trust region strategy but differs from TRON in significant issues. In particular, LANCELOT does not use projected searches, and the default is a banded preconditioner. The L-BFGS-B code is a limited-memory variable metric method. An advantage of L-BFGS-B is that only the gradient is required, while Newton codes require an approximation to the Hessian matrix. On the other hand, for sparse problems the Hessian matrix can usually be obtained efficiently with differences of gradients if the sparsity pattern of the Hessian matrix is provided.

Our first set of computational results uses a set of bound-constrained problems from the CUTE collection [3]. We used the select tool to choose problems representative of problems that arise in applications and where the number of variables n could be changed. Since we are interested in large problems, we refined this selection by considering only problems where the number of variables was at least $5,000$. These requirements lead to a list of nine problems, with some of the problems having more than one version.

Table 7.1 presents the results obtained when LANCELOT and L-BFGS-B are used with the default options. For LANCELOT, exact second derivatives and a preconditioned conjugate gradient method with a banded preconditioner were used; all other default options are shown in Table 5 of [15]. In Table 7.1 we used the LANCELOT termination test

$$\|P[x - \nabla f(x)] - x\|_\infty \leq 10^{-5}, \tag{7.1}$$

where P is the projection into the feasible set (1.2).

The first column in Table 7.1 is the name of the test problem, and the second column is the number of variables n. For TRON and LANCELOT we record the number

Table 7.1
Performance on the CUTE *problems: default options.*

		TRON				LANCELOT				L-BFGS-B	
Problem	n	nh	nf	ncg	time	nh	nf	ncg	time	nfg	time
BDEXP	5000	11	11	10	1.43	10	11	12	1.19	15	0.60
CVXBQP1	10000	2	2	0	0.24	1	2	1	0.81	2	0.08
JNLBRNG1	15625	26	26	33	15.22	24	25	2029	165.42	999	198.75
JNLBRNG2	15625	16	16	27	9.21	14	15	898	74.16	577	105.18
JNLBRNGA	15625	23	23	29	12.46	21	22	1584	117.64	332	54.56
JNLBRNGB	15625	10	10	15	5.29	8	9	419	30.71	999	160.32
MCCORMCK	10000	6	7	6	1.46	4	5	4	1.10	15	1.76
NCVXBQP1	10000	2	2	0	0.24	4	5	0	3.01	2	0.08
NCVXBQP2	10000	10	10	10	1.44	6	7	84	3.35	178	6.85
NCVXBQP3	10000	10	10	10	1.39	6	7	163	2.96	388	14.87
NOBNDTOR	14884	38	38	71	22.03	36	37	1386	123.66	213	36.38
NONSCOMP	10000	9	9	8	1.44	8	9	8	1.45	51	4.24
OBSTCLAE	15625	27	27	51	14.48	5	6	7452	821.46	660	116.18
OBSTCLAL	15625	25	25	39	12.64	24	25	604	43.64	156	24.51
OBSTCLBL	15625	20	20	42	12.81	18	19	2088	199.04	272	49.28
OBSTCLBM	15625	8	8	15	5.41	5	6	1378	152.87	146	25.90
OBSTCLBU	15625	21	21	33	11.85	19	20	621	56.68	194	33.94
TORSION1	14884	39	39	64	19.85	37	38	1148	86.08	224	35.36
TORSION2	14884	19	19	43	11.10	14	15	2063	173.28	521	91.56
TORSION3	14884	20	20	26	9.06	19	20	332	21.13	76	10.66
TORSION4	14884	18	18	27	8.98	14	15	653	34.99	417	65.78
TORSION5	14884	11	11	12	4.67	9	10	93	5.74	40	5.06
TORSION6	14884	15	15	18	7.07	8	9	151	8.54	362	53.99
TORSIONA	14884	39	39	64	21.45	37	38	1147	98.23	205	37.38
TORSIONB	14884	24	24	50	14.54	15	16	1982	186.69	371	70.13
TORSIONC	14884	20	20	26	9.80	19	20	332	24.65	89	13.97
TORSIOND	14884	18	18	26	9.70	14	15	634	39.70	409	69.59
TORSIONE	14884	11	11	12	5.06	9	10	93	6.55	38	5.44
TORSIONF	14884	15	15	19	7.71	7	8	154	9.36	341	56.83

of Hessian evaluations nh, function evaluations nf, and conjugate gradient iterations ncg. For L-BFGS-B we record only the number of function and gradient evaluations nfg because L-BFGS-B always evaluates the function and gradient at the same time. The execution time (in seconds) is reported in the time column. In these results, all three codes obtained the same optimal function value at the final iterate.

A general observation on the results in Table 7.1 is that the number of function evaluations for TRON and LANCELOT is at most one more than the number of Hessian evaluations. Thus, for these problems all the iterations of the Newton codes are successful. We conclude that these problems do not fully test TRON or LANCELOT.

In analyzing computational results we do not discuss problems where L-BFGS-B requires less than 50 function and gradient evaluations. In general, we feel that if a limited-memory variable metric algorithm converges in less than 50 function and gradient evaluations on a problem with $10,000$ variables, then the starting point is exceptionally good.

An important observation on the results in Table 7.1 is that on these problems TRON requires less time than L-BFGS-B. These results support the conclusion that TRON is preferable to L-BFGS-B if the Hessian matrix can be obtained explicitly. We also expect TRON to outperform L-BFGS-B for sparse problems if the sparsity pattern of the Hessian matrix is provided because with this information the Hessian matrix can be obtained efficiently from differences of gradients.

The results in Table 7.1 also show that on these problems TRON requires less time than LANCELOT and requires significantly fewer conjugate gradient iterations than LANCELOT. Reducing the number of conjugate gradient iterations is important because this number is likely to increase as the number of variables increases. We note that since for these problems the cost of the conjugate gradient iterations is significant, fewer conjugate gradient iterations translates into smaller computing times.

Another observation made on the results of Table 7.1 is that LANCELOT usually requires fewer major iterations than TRON. Differences in the number of major iterations are due, in part, to the choice of Cauchy point and the use of projected searches. These algorithmic choices in TRON tend to add many constraints, and on some of these problems, they lead to a larger number of major iterations. We also note that a detailed examination of the output shows that even when both codes require the same number of iterations, the algorithms visit different faces of the feasible set.

As a minor point, note that TRON almost always requires the same number of function and Hessian evaluations. This is an algorithmic decision since we always evaluate the gradient and Hessian at successful iterates. On the other hand, if an iterate satisfies the termination criteria (7.1), LANCELOT returns without evaluating the Hessian matrix at the final iterate.

The number of conjugate gradient iterations in LANCELOT can usually be reduced by using other preconditioners instead of the default banded preconditioner. Other preconditioners, however, usually require more memory and more floating point operations per conjugate gradient iteration.

In Table 7.2 we present the results of using LANCELOT with Munksgaard's ma31 preconditioner [31], which is an incomplete Cholesky factorization with a drop tolerance. A disadvantage of using the ma31 preconditioner with LANCELOT is that the memory requirements are unpredictable. The user is asked to allocate a given amount of memory, and if this amount is not sufficient, then an error message is issued. On the other hand, the incomplete Cholesky factorization icfs used in TRON does not require the choice of a drop tolerance, and the amount of storage can be specified in advance. For the results presented in this section icfs uses $5n$ additional (double precision) words. For a comparison of ma31 with icfs, see Lin and Moré [26].

Comparison of the LANCELOT results in Table 7.1 with those in Table 7.2 show that in all cases the number of function evaluations and the number of Hessian evaluations for both preconditioners are identical and that the main difference is the number of conjugate gradient iterations. Also note that, with the exception of problems OBSTCLBL and OBSTCLBM, the number of conjugate gradient iterations and the time required to solve the problems with LANCELOT decreased when the ma31 preconditioner was used. Overall, these results show that for these problems the ma31 preconditioner is preferable in LANCELOT.

The results in Table 7.2 show that TRON requires fewer conjugate gradient iterations and, on most problems, less time than LANCELOT with the ma31 preconditioner. Also note that there were five problems (OBSTCLAE, OBSTCLBL, OBSTCLBM, TORSION2, and TORSIONB) where LANCELOT required more than $1,000$ conjugate gradient iterations, and note that on these problems the reductions in time over the default preconditioner were not substantial. For these problems the differences in conjugate gradient iterations are due not to the use of different preconditioners but to the methods used by TRON and LANCELOT to compute the minor iterates. LANCELOT uses a line search, and thus only one constraint is added at each

Table 7.2
Performance on the CUTE *problems:* LANCELOT *with* ma31.

		TRON				LANCELOT (ma31)			
Problem	n	nh	nf	ncg	time	nh	nf	ncg	time
BDEXP	5000	11	11	10	1.43	10	11	10	1.32
CVXBQP1	10000	2	2	0	0.24	1	2	1	0.80
JNLBRNG1	15625	26	26	33	15.22	24	25	179	28.69
JNLBRNG2	15625	16	16	27	9.21	14	15	70	13.09
JNLBRNGA	15625	23	23	29	12.46	21	22	166	24.29
JNLBRNGB	15625	10	10	15	5.29	8	9	46	7.56
MCCORMCK	10000	6	7	6	1.46	4	5	4	1.41
NCVXBQP1	10000	2	2	0	0.24	4	5	0	3.03
NCVXBQP2	10000	10	10	10	1.44	7	8	93	3.34
NCVXBQP3	10000	10	10	10	1.39	6	7	124	2.61
NOBNDTOR	14884	38	38	71	22.03	36	37	176	36.61
NONSCOMP	10000	9	9	8	1.44	8	9	8	1.66
OBSTCLAE	15625	27	27	51	14.48	2	3	7154	809.04
OBSTCLAL	15625	25	25	39	12.64	24	25	79	15.62
OBSTCLBL	15625	20	20	42	12.81	22	21	2346	307.67
OBSTCLBM	15625	8	8	15	5.41	5	6	1554	213.38
OBSTCLBU	15625	21	21	33	11.85	19	20	165	22.72
TORSION1	14884	39	39	64	19.85	37	38	159	27.97
TORSION2	14884	19	19	43	11.10	14	15	1592	143.66
TORSION3	14884	20	20	26	9.06	19	20	52	9.02
TORSION4	14884	18	18	27	8.98	14	15	438	25.91
TORSION5	14884	11	11	12	4.67	9	10	14	2.99
TORSION6	14884	15	15	18	7.07	8	9	116	7.46
TORSIONA	14884	39	39	64	21.45	37	38	175	31.80
TORSIONB	14884	24	24	50	14.54	15	16	1606	153.55
TORSIONC	14884	20	20	26	9.80	19	20	52	9.76
TORSIOND	14884	18	18	26	9.70	14	15	445	29.13
TORSIONE	14884	11	11	12	5.06	9	10	13	3.27
TORSIONF	14884	15	15	19	7.71	7	8	107	7.46

minor iteration. As a result many minor iterates can be generated, and determining a minor iterate almost certainly requires at least one conjugate gradient iteration. For these five problems LANCELOT generated, respectively, 7,155, 1,710, 1,184, 1,533, and 1,541 minor iterates. TRON, on the other hand, uses a projected search and thus is able to add many constraints at each minor iteration. For these problems TRON generated 27, 26, 10, 19, and 24 minor iterates.

These results support the conclusion that TRON tends to require significantly fewer minor iterations than LANCELOT. Moreover, the use of projected searches is the major reason for TRON requiring a small number of minor iterates.

General conclusions cannot be drawn from these results because, as already noted, this problem set does not fully test these algorithms. Our numerical results are also affected by nonalgorithmic differences between TRON and LANCELOT. We have already noted that these codes differ in the amount of memory required, but TRON and LANCELOT differ in other ways. For example, LANCELOT uses the partial separability structure, while TRON uses only the sparsity structure.

We also compared TRON with L-BFGS-B on a test set from the MINPACK-2 collection of large-scale problems [1]. The MINPACK-2 problems defined by Table 7.3 are finite-dimensional approximations of an infinite-dimensional variational problem defined over a grid with n_x and n_y grid points in each coordinate direction. The column

Table 7.3
Parameters for the MINPACK-2 *test problems.*

Problem	n	n_x	n_y	λ	l	u
EPT1	10000	200	50	1.0d0	default	default
EPT2	10000	200	50	5.0d0	default	default
EPT3	10000	200	50	10.0d0	default	default
PJB1	10000	100	100	0.1d0	default	1.0d2
PJB2	10000	100	100	0.5d0	default	1.0d2
PJB3	10000	100	100	0.9d0	default	1.0d2
MSA1	10000	200	50	0.0d0	-0.4d0	0.4d0
MSA2	10000	200	50	0.0d0	-0.2d0	0.2d0
MSA3	10000	200	50	0.0d0	-0.1d0	0.1d0
SSC1	10000	100	100	5.0d0	1.0d-1	1.0d0
SSC2	10000	100	100	5.0d0	1.0d-2	1.0d0
SSC3	10000	100	100	5.0d0	1.0d-3	1.0d0
SSC4	10000	100	100	5.0d0	1.0d-4	1.0d0
DGL2	10000	50	50	2.0d0	-1.0d20	1.0d20

labeled λ in Table 7.3 defines the value of a parameter associated with the problem, while the last two columns define the lower and upper bounds on the variables. For these results we used the termination test

$$\|\nabla_\Omega f(x)\|_2 \le 10^{-5}\|\nabla f(x_0)\|_2, \tag{7.2}$$

where $\nabla_\Omega f$ is the projected gradient (2.8). This termination test is generally preferable to (7.1) because (7.2) is invariant to changes in the scale of f.

The number of grid points n_x and n_y and the parameter λ can be modified easily in the MINPACK-2 problems, thereby providing a convenient means for generating difficult problems. In general, the problems become more difficult as the ratio n_y/n_x deviates from unity. We have restricted the testing to problems where this ratio lies in the interval $[0.25, 1]$, which leads to relatively easy problems. In some cases, the choice of λ and of lower and upper bounds also affects the performance of optimization algorithms.

In the first two problems in Table 7.4 we examine the behavior of TRON and L-BFGS-B as λ changes. For problem EPT (elastic-plastic torsion) the parameter λ is the force constant, and for this problem the number of active constraints increases as λ increases. The results in Table 7.4 show that EPT becomes easier to solve as λ increases. This finding is reasonable because the EPT problem tends to be increasingly linear as λ increases. The results for problem PJB (pressure in a journal bearing) show that this problem becomes increasingly harder to solve as λ approaches unity. For this problem λ is the eccentricity of the journal bearing, so this result is reasonable.

In problems MSA and SSC we examine the behavior of TRON and L-BFGS-B as the lower and upper bounds l and u change. The results of this testing were somewhat disappointing because for these problems there does not seem to be a strong correlation between the choice of bounds and the number of iterations. The most dramatic change in performance occurs for L-BFGS-B and the MSA problem. Note, on the other hand, that the performance of TRON is relatively insensitive to the choice of bounds.

Problem GL2 is unconstrained but is included in these results because it is a hard problem for algorithms that do not use second-order information. The reason seems to be that the GL2 problem has a saddle point that attracts L-BFGS-B.

Table 7.4
Performance on the MINPACK-2 *problems with* $n = 10,000$.

	TRON				L-BFGS-B	
Problem	nh	nf	ncg	time	nfg	time
EPT1	30	30	96	9.38	466	35.17
EPT2	31	31	61	7.69	445	27.81
EPT3	21	21	31	4.06	229	10.66
PJB1	22	22	42	5.92	717	49.25
PJB2	13	13	29	3.38	542	31.29
PJB3	7	7	17	1.76	2765	150.91
MSA1	27	48	94	19.06	776	65.35
MSA2	16	22	65	10.47	613	50.50
MSA3	19	19	48	9.89	487	39.79
SSC1	5	5	23	3.28	347	36.32
SSC2	6	6	25	4.11	345	36.83
SSC3	6	6	26	3.96	377	40.26
SSC4	6	6	26	3.99	293	30.91
GL2	8	8	364	34.73	3521	372.89

The most striking feature of the results in Table 7.4 is that TRON requires far fewer function and gradient evaluations than L-BFGS-B and that this translates into smaller computing times. This advantage is likely to increase as the number of variables increases because the number of iterations in a Newton method tends to grow slowly, while the number of iterations in limited-memory variable metric methods tends to grow rapidly as the number of variables increases. See, for example, the results of Bouaricha, Moré, and Wu [4].

Acknowledgments. The implementation of the Newton code benefited from the work of Ali Bouaricha and Zhijun Wu on the unconstrained version of the code. We thank Gail Pieper for her careful reading of the manuscript. We also thank the referees and the associate editor, Nick Gould, for their comments. One of the referees provided an extremely detailed and helpful report that led to improvements in the amount of detail and explanations in the paper.

REFERENCES

[1] B. M. Averick, R. G. Carter, J. J. Moré, and G.-L. Xue, *The MINPACK-2 test problem collection*, Preprint MCS-P153-0694, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1992.

[2] D. P. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.

[3] I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *CUTE: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[4] A. Bouaricha, J. J. Moré, and Z. Wu, *Preconditioning Newton's Method*, Preprint ANL/MCS-P715-0598, Argonne National Laboratory, Argonne, IL, 1998.

[5] J. V. Burke and J. J. Moré, *On the identification of active constraints*, SIAM J. Numer. Anal., 25 (1988), pp. 1197–1211.

[6] J. V. Burke and J. J. Moré, *Exposing constraints*, SIAM J. Optim., 4 (1994), pp. 573–595.

[7] J. V. Burke, J. J. Moré, and G. Toraldo, *Convergence properties of trust region methods for linear and convex constraints*, Math. Programming, 47 (1990), pp. 305–336.

[8] P. H. Calamai and J. J. Moré, *Projected gradient methods for linearly constrained problems*, Math. Programming, 39 (1987), pp. 93–116.

[9] T. F. Coleman and Y. Li, *An interior trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optim., 6 (1996), pp. 418–445.

[10] T. F. Coleman and Y. Li, *A Trust Region and Affine Scaling Interior Point Method for Nonconvex Minimization with Linear Inequality Constraints*, Technical Report TR97-1642, Cornell University, Ithaca, NY, 1997.

[11] T. F. Coleman and Y. Li, *Combining trust region and affine scaling for linearly constrained nonconvex minimization*, in Advances in Nonlinear Programming, Y. Yuan, ed., Kluwer Academic Publishers, Norwell, MA, 1998, pp. 219–250.

[12] A. R. Conn, N. I. M. Gould, A. Sartenaer, and P. L. Toint, *Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints*, SIAM J. Optim., 3 (1993), pp. 164–221.

[13] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460; correction in SIAM J. Numer. Anal., 26 (1989), pp. 764–767.

[14] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *LANCELOT*, Springer Ser. Comput. Math., Springer, New York, 1992.

[15] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Numerical experiments with the LANCELOT package (Release A) for large-scale nonlinear optimization*, Math. Programming, 73 (1996), pp. 73–110.

[16] J. E. Dennis and L. N. Vicente, *Trust-region interior-point algorithms for minimization problems with simple bounds*, in Applied Mathematics and Parallel Computing, Festschrift for Klaus Ritter, H. Fisher, B. Riedmüller, and S. Schäffler, eds., Physica, Heidelberg, 1996, pp. 97–107.

[17] J. C. Dunn, *On the convergence of projected gradient processes to singular critical points*, J. Optim. Theory Appl., 55 (1987), pp. 203–216.

[18] F. Facchinei, J. Júdice, and J. Soares, *An active set Newton's algorithm for large-scale nonlinear programs with box constraints*, SIAM J. Optim., 8 (1998), pp. 158–186.

[19] F. Facchinei and S. Lucidi, *A Class of Methods for Optimization Problems with Simple Bounds*, Technical Report R.336, IASI-CNR, Rome, Italy, 1992.

[20] A. Forsgren and W. Murray, *Newton methods for large-scale linear inequality-constrained minimization*, SIAM J. Optim., 7 (1997), pp. 162–176.

[21] A. Friedlander, J. M. Martínez, and S. A. Santos, *A new trust region algorithm for bound constrained minimization*, Appl. Math. Optim., 30 (1994), pp. 235–266.

[22] E. M. Gafni and D. P. Bertsekas, *Two-metric projection methods for constrained optimization*, SIAM J. Control Optim., 22 (1984), pp. 936–964.

[23] D. M. Gay, *A trust region approach to linearly constrained optimization*, in Numerical Analysis, D. F. Griffiths, ed., Lecture Notes in Math. 1066, Springer, New York, 1984, pp. 72–105.

[24] M. Heinkenschloss, M. Ulbrich, and S. Ulbrich, *Superlinear and Quadratic Convergence of Affine-Scaling Interior-Point Newton Methods for Problems with Simple Bounds without Strict Complementarity Assumption*, Technical Report TR97-30, Rice University, Houston, TX, 1997; Math. Programming, to appear.

[25] M. Lescrenier, *Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold*, SIAM J. Numer. Anal., 28 (1991), pp. 476–495.

[26] C.-J. Lin and J. J. Moré, *Incomplete Cholesky factorizations with limited memory*, SIAM J. Sci. Comput., 21 (1999), pp. 24–45.

[27] J. J. Moré, *Trust regions and projected gradients*, in Systems Modelling and Optimization, M. Iri and K. Yajima, eds., Lecture Notes in Control and Inform. Sci. 113, Springer, New York, 1988, pp. 1–13.

[28] J. J. Moré, *Global methods for nonlinear complementarity problems*, Math. Oper. Res., 21 (1996), pp. 589–614.

[29] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[30] J. J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.

[31] N. Munksgaard, *Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients*, ACM Trans. Math. Software, 6 (1980), pp. 206–219.

[32] S. M. Robinson, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.

[33] A. Schwartz and E. Polak, *Family of projected descent methods for optimization problems with simple bounds*, J. Optim. Theory Appl., 92 (1997), pp. 1–31.

[34] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[35] Ph. L. Toint, *Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space*, IMA J. Numer. Anal., 8 (1988), pp. 231–252.

[36] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, *L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization*, ACM Trans. Math. Softw., 23 (1997), pp. 550–560.

SIAM J. OPTIM.
Vol. 9, No. 4, pp. 1128–1134

POLYHEDRAL BOUNDARY PROJECTION*

O. L. MANGASARIAN[†]

With best wishes to a friend, colleague, and major contributor to mathematical programming: John Dennis, on the occasion of his 60th birthday.

Abstract. We consider the problem of projecting a point in a polyhedral set onto the boundary of the set using an arbitrary norm for the projection. Two types of polyhedral sets, one defined by a convex combination of k points in R^n and the second by the intersection of m closed half-spaces in R^n, lead to disparate optimization problems for finding such a projection. The first case leads to a mathematical program with a linear objective function and constraints that are linear inequalities except for a single nonconvex cylindrical constraint. Interestingly, for the 1-norm, this nonconvex problem can be solved by solving $2n$ linear programs. The second polyhedral set leads to a much simpler problem of determining the minimum of m easily evaluated numbers. These disparate mathematical complexities parallel known ones for the related problem of finding the largest ball, with radius measured by an arbitrary norm, that can be inscribed in the polyhedral set. For a polyhedral set of the first type this problem is NP-hard for the 2-norm and the ∞-norm [R. M. Freund and J. B. Orlin, *Math. Programming*, 33 (1985), pp. 139–145] and solvable by a single linear program for the 1-norm [P. Gritzmann and V. Klee, *Math. Programming*, 59 (1993), pp. 163–213], while for the second type this problem leads to a single linear program even for a general norm [P. Gritzmann and V. Klee, *Discrete Comput. Geom.*, 7 (1992), pp. 255–280].

Key words. polyhedral set, boundary projection, largest inscribed ball

AMS subject classifications. 15A39, 90C05, 90C30

PII. S1052623497329245

1. Introduction.

We consider a polytope in the n-dimensional real space R^n defined as a convex combination of k points and represented as follows:

$$S := \{y \mid y = Bz,\ z \geq 0,\ e^T z = 1\}, \tag{1.1}$$

where B is an $n \times k$ real matrix and e is a vector of ones. Given a point $s \in S$ we want to find a projection p onto the boundary $\text{bd}(S)$ of S using an arbitrary norm on R^n.[1] Similar problems arise in data envelopment analysis (DEA) [2], where the distance to the boundary is an efficiency measure of a decision making unit (DMU) represented by that point. Each DMU is represented by an n-dimensional column of B and each of the n dimensions measures components required or products generated by that DMU. The set of efficient points for the DMUs is that part of the boundary of the convex hull of S that cannot be improved on by any other DMU consuming less or equal amounts of components or generating more or equal amounts of products. Projecting a DMU onto the boundary of S gives an indication of how close to an efficient point that DMU is. We will show in section 2, by using an arbitrary-norm projection onto a hyperplane [9], that this is basically a nonconvex problem. However, the problem can be reduced (Theorem 2.2) to a mathematical program with a linear objective function

*Received by the editors October 27, 1997; accepted for publication (in revised form) July 3, 1998; published electronically September 24, 1999. This work was supported by National Science Foundation grant CCR-9322479 and Air Force Office of Scientific Research grant F49620-97-1-0326 as Mathematical Programming Technical Report 97-10, October 1997.

http://www.siam.org/journals/siopt/9-4/32924.html

[†]Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706 (olvi@cs.wisc.edu).

[1] The boundary projection problem for the set S using the 1-norm was suggested to the author by Holger Scheel of Dortmund University.

and linear constraints except for a single nonlinear equality constraint. The nonlinear constraint restricts a solution to the surface of a cylinder determined by the dual norm to that used in measuring the distance to the boundary of S. If the 1-norm is used to measure the distance, then because its dual is the ∞-norm, remaining on the ∞-norm cylinder and solving the problem can be achieved by solving $2n$ linear programs. A related problem to the boundary projection problem is that of determining the largest ball in R^n, with radius measured by an arbitrary norm, that is contained in S. This is a problem that has been studied by a number of authors [3, 4, 6, 7], and it is NP-hard except for the 1-norm which can be solved by a single linear program [7, Theorem 3.3]. By using our polyhedral boundary projection result we formulate this problem as a maxmin problem (Theorem 2.4); an upper bound obtained by interchanging the max and min, however, is twice as large as the upper bound of Gritzmann and Klee [6, (1.3)]. (See corollaries 2.5 and 2.6.)

In contrast to the nonconvex problems arising from the boundary projection and largest-inscribed-ball problems associated with a polyhedral set described by (1.1), the corresponding optimization problems are much simpler when the polyhedral set is described as the intersection of m closed half-spaces as follows:

$$T := \{x \mid Ax \geq b\}, \tag{1.2}$$

where A is an $m \times n$ real matrix and b is a vector in R^m. The largest-inscribed-ball problem in T was studied in [6, 7]. In section 3 we cite these results and state that for any norm on R^n the projection problem reduces to the rather trivial problem of finding the minimum of m easily calculated numbers, while the largest ball problem for any norm can be solved by a single linear program.

Section 4 gives a brief summary and conclusion.

We note that given a polytope set in the form of (1.1) it is not easy to derive the equivalent form (1.2) for that specific set; for example, the 1-norm unit ball has $2n$ vertices but 2^n faces. Going backward from (1.2) to (1.1) is also difficult—for example, the ∞-norm unit ball has $2n$ faces but 2^n vertices—and may not even be possible because T may be unbounded. In fact, by Motzkin's polyhedral decomposition theorem [5, Theorem 1] the polyhedral set T is equivalent to the algebraic sum of a convex combination of points in R^n (a set similar to S) plus a convex polyhedral cone, neither of which is easy to find.

1.1. Notation and background.

All vectors will be column vectors unless transposed to a row vector by a superscript T. The scalar product of two vectors x and y in the n-dimensional real space R^n will be denoted by x^Ty. For a mathematical program $\min_{x\in X}\ f(x)$, where $f : R^n \longrightarrow R$, the notation arg $\min_{x\in X}\ f(x)$ will denote the set of solutions of the mathematical program $\min_{x\in X}\ f(x)$. For $x \in R^n$ and $p \in [1, \infty)$, the norm $\|x\|_p$ will denote the p-norm $(\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$ and $\|x\|_\infty$ will denote $\max_{1\leq i\leq n} |x_i|$. For an $m \times n$ matrix A, A_i will denote row i of A and $A_{\cdot j}$ will denote column j of A. The identity matrix in a real space of arbitrary dimension will be denoted by I, while a column vector of ones of arbitrary dimension will be denoted by e, and a column vector of arbitrary dimension with zeros in every row except one in row i will be denoted by e^i. The symbol := will denote a definition.

A boundary point of a set $X \subseteq R^n$ is any point in R^n such that any open set containing the point contains points in X and points not in X. The closed set of all boundary points of X, denoted by bd(X), is contained in X if and only if X is closed. Hence the closed polyhedral sets S and T contain their boundaries. By a projection of a point $s \in R^n$ onto a closed set $X \subseteq R^n$ we mean an element of

$P := \arg\min_{x\in X} \|x-s\|$, where $\|\cdot\|$ is some specified norm on R^n. Because P may not be a singleton as a consequence of the nonconvexity of X or the norm being the 1-norm or ∞-norm, we shall mean by "projection of s onto X" any element of P, and similarly for "its projection." For a general norm $\|\cdot\|$ on R^n, the dual norm $\|\cdot\|'$ on R^n and the resulting Cauchy–Schwarz inequality are

$$\|y\|' := \max_{\|x\|=1} y^Tx, \ \ \pm y^Tx \le |y^Tx| \le \|y\|'\|x\|. \tag{1.3}$$

For $p, q \in [1,\infty]$, $\frac{1}{p}+\frac{1}{q}=1$, the p-norm and q-norm are dual norms.

We need, in Theorem 2.2 below, an explicit form for a projection of an arbitrary point onto a given hyperplane using a general norm. Gritzmann and Klee [6, Proof of (1.14)] give the distance between a point and its projection on the hyperplane but do not give a projection explicitly. For later use we state the following result.

Proposition 1.1 (see [9], arbitrary-norm projection onto a hyperplane). *Let $q \in R^n$ be any point in R^n not on the hyperplane:*

$$P := \{x \mid w^Tx = \gamma\},\ 0 \ne w \in R^n,\ \gamma \in R. \tag{1.4}$$

A projection $p(q) \in P$ using a general norm $\|\cdot\|$ on R^n is given by

$$p(q) = q - \frac{w^Tq-\gamma}{\|w\|'}r(w), \tag{1.5}$$

where $\|\cdot\|'$ is the dual norm to $\|\cdot\|$ and

$$r(w) \in \arg\max_{\|y\|=1} w^Ty. \tag{1.6}$$

Consequently, the distance between q and its projection $p(q)$ is given by

$$\|q-p(q)\| = \frac{|w^Tq-\gamma|}{\|w\|'}. \tag{1.7}$$

Explicit expressions for the 1-norm, 2-norm, and ∞-norm for (1.5)–(1.7) are given in [9, Corollaries 2.3–2.5].

2. Boundary projection and largest ball for polytope S. We begin with the problem of finding a projection of a point $s \in S$ onto the boundary $\mathrm{bd}(S)$ of S for an arbitrary norm on R^n. For that purpose we need to characterize the set $\bar{S}$ of points not in S by means of a separating hyperplane argument as follows.

Lemma 2.1 (characterization of the complement of S). *The set of points $\bar{S}$ in R^n not in S can be characterized as follows:*

$$\bar{S} = \{p \mid (p,y,\zeta) \in R^{2n+1},\ B^Ty + e\zeta \ge 0,\ p^Ty+\zeta < 0,\ \| y \|'= 1\}, \tag{2.1}$$

where $\|\cdot\|'$ is the dual of an arbitrary norm $\|\cdot\|$ on R^n and $p^Ty+\zeta = 0$ is a supporting hyperplane of S separating S from a point in $\bar{S}$.

Proof. By the strict separation theorem for convex sets [8, Theorem 3.2.6], $p \in \bar{S}$ if and only if there exists a hyperplane in R^n: $\{x \mid v^Tx+\xi = 0\}$ for some $v \ne 0$ and $\xi \in R$, which strictly separates p from S; that is,

$$v^Tp+\xi < 0,\ v^TBz+\xi \ge 0 \quad \forall z \ge 0,\ e^Tz = 1$$

or, equivalently, $p^T v + \xi < 0$ and $B^T v + e\xi \geq 0$. Hence

$$\bar{S} = \{p \mid (p, v, \xi) \in R^{2n+1},\ B^T v + e\xi \geq 0,\ p^T v + \xi < 0\}.$$

Since v cannot equal zero in the definition of the last set, it follows on normalization by dividing by $\| v \|'$ and defining $y := v / \| v \|'$ and $\zeta := \xi / \| v \|'$ that (2.1) holds. □

By using this lemma and Proposition 1.1 we are able now to state a mathematical program that characterizes the boundary projection problem.

THEOREM 2.2 (boundary projection $p(s)$ for $s \in S$). *The distance between $s \in S$ and its projection $p(s)$ onto the boundary* $\mathrm{bd}(S)$ *of S using a general norm $\| \cdot \|$ on R^n can be determined as follows:*

$$(2.2) \qquad \begin{aligned} \|s - p(s)\| &= \min_{y,\zeta}\{s^T y + \zeta \mid B^T y + e\zeta \geq 0,\ \|y\|' = 1\} \quad &\text{(a)} \\ &= \min_{\|y\|'=1} \left\{ s^T y - \min_{1 \leq j \leq k} y^T B_{\cdot j} \right\}. &\text{(b)} \end{aligned}$$

Furthermore, if $(\bar{y}, \bar{\zeta})$ is a solution of (2.2(a))*, then a projection $p(s)$ of s onto the boundary* $\mathrm{bd}(S)$ *of S is given by*

$$(2.3) \qquad p(s) = s - (s^T \bar{y} + \bar{\zeta}) r(\bar{y}), \ \textit{where}\ r(\bar{y}) \in \arg\max_{\|y\|=1} \bar{y}^T y.$$

Proof. Denote the feasible region of (2.2(a)) as

$$(2.4) \qquad Z = \{(y, \zeta) \mid (y, \zeta) \in R^{n+1},\ B^T y + e\zeta \geq 0,\ \|y\|' = 1\}.$$

The equality of (2.2(a)) and (2.2(b)) is obvious once we define

$$(2.5) \qquad \zeta := -\min_{1 \leq j \leq k} y^T B_{\cdot j}.$$

Since the objective function of (2.2(b)) is piecewise-linear convex and hence continuous, it attains a minimum on the compact unit sphere at some $\bar{y}$. The corresponding $\bar{\zeta}$ computed by (2.5) gives an optimal solution $(\bar{y}, \bar{\zeta})$ to (2.2(a)).

We now show that this minimum gives the distance between $s \in S$ and its projection onto the boundary of S. For each $p \in \bar{S}$ there exists, by Lemma 2.1, $(y, \zeta) \in R^{n+1}$ such that $(y, \zeta) \in Z$ and such that p lies in the open half-space $\{q \mid q^T y + \zeta < 0\}$ and such that S lies in the complementary closed half-space $\{q \mid q^T y + \zeta \geq 0\}$. Since $\|y\|' = 1$ for $(y, \zeta) \in Z$ it follows by Proposition 1.1 that the distance between s and its projection onto the hyperplane $\{q \mid q^T y + \zeta = 0\}$ separating points in S and $\bar{p} \in \bar{S}$ is $s^T y + \zeta$. Furthermore, the minimum $s^T \bar{y} + \bar{\zeta}$ of $s^T y + \zeta$ over all $(y, \zeta) \in Z$ is the desired distance between s and its projection onto the boundary $\mathrm{bd}(S)$ of S because of the following. Since $s^T \bar{y} + \bar{\zeta}$ is the distance from s to a projection of s onto a closest separating hyperplane that separates S from a point in its complement $\bar{S}$, any such projection is also a projection of s onto its boundary. This is because it lies on a separating hyperplane and hence cannot lie in the interior of S; if it were not in S, a point on the line segment joining the projection to s would intersect the boundary of S closer to s and the separating hyperplane at this boundary point would have a closer projection to s contradicting the minimality of $s^T \bar{y} + \bar{\zeta}$. Also, a projection $p(s)$ of S onto the boundary $\mathrm{bd}(S)$ of S is given by a projection of s onto the hyperplane $\{q \mid q^T \bar{y} + \bar{\zeta} = 0\}$, which again by Proposition 1.1 is given by (2.3). □

For the 2-norm and the ∞-norm the boundary projection problem (2.2) is NP-hard (see [7, Theorem 5.1], taking s to be the origin). However, for the 1-norm it can be solved in polynomial time by solving $2n$ linear programs as follows.

Corollary 2.3. *For the case of* $\|\cdot\| = \|\cdot\|_1$, *the mathematical program* (2.2) *can be solved by solving the following* $2n$ *linear programs:*

$$\text{(2.6)}\qquad \begin{aligned} \|s-p(s)\|_1 &= \min_{i=1,\ldots,n,\sigma=\pm1} P_{i\sigma}, \quad \textit{where} \\ P_{i\sigma} &:= \min_{y,\zeta}\{s^Ty+\zeta \mid B^Ty+e\zeta \ge 0,\ -e \le y \le e,\ y_i=\sigma\}. \end{aligned}$$

Any $(\bar y, \bar\zeta)$ *determined by solving* (2.6) *can be used in* (2.3) *to determine a projection* $p(s)$ *onto the boundary* $\mathrm{bd}(S)$ *of* S *using the* 1*-norm.*

We turn our attention to the problem of finding the radius, measured by an arbitrary norm, of the largest ball in R^n that is contained in S. By using Theorem 2.2 and maximizing over all s in S we can formulate this problem as follows.

Theorem 2.4 (largest ball inscribed in S). *The radius* ρ *of the largest ball, measured by an arbitrary norm* $\|\cdot\|$ *on* R^n, *that can be inscribed in* S *is given by*

$$\text{(2.7)}\qquad \rho = \|\bar s - p(\bar s)\| = \max_{s\in S}\ \min_{y,\zeta}\{s^Ty+\zeta \mid B^Ty+e\zeta \ge 0,\ \|y\|'=1\}.$$

Proof. The distance function $\|s-p(s)\|$ of (2.2) is an upper semicontinuous function of s on S [1, p. 115, Theorem 1]. Since S is compact it follows that the upper semicontinuous distance function $\|s-p(s)\|$ on S attains its maximum on S. □

Problem (2.7) is a difficult problem to solve for a general norm. Freund and Orlin [4] have shown that this is an NP-hard problem for the 2-norm and the ∞-norm, while Gritzmann and Klee [7, Theorem 3.3] have shown that for the 1-norm the problem can be formulated as a single linear program. For a general norm problem, (2.7) is a minmax problem over the product of two sets, one of which is nonconvex. The nonconvexity of the set $Z = \{(y,\zeta) \mid (y,\zeta) \in R^{n+1},\ B^Ty+e\zeta \ge 0,\ \|y\|'=1\}$ precludes the use of a minmax theorem to switch the maxmin to a minmax, which would simplify the problem. However, since the minmax is an upper bound to the maxmin, which is the case here because the various minima and maxima exist, we obtain the following corollary to the above theorem by using the minmax as an upper bound to the maxmin and then simplifying the resulting expression.

Corollary 2.5 (upper bound for largest inscribed ball). *An upper bound on the radius* ρ *of the largest ball, measured by an arbitrary norm* $\|\cdot\|$ *on* R^n, *that can be inscribed in* S *is given by*

$$\text{(2.8)}\qquad \rho = \|\bar s - p(\bar s)\| \le \min_{\|y\|'=1}\left(\max_{1\le j\le k} y^TB_{.j} - \min_{1\le j\le k} y^TB_{.j}\right).$$

Proof. The minmax upper bound to the maxmin of (2.7) is given by

$$\text{(2.9)}\ \rho = \|\bar s - p(\bar s)\| \le \min_{y,\zeta}\left\{\max_{s=Bz,\, e^Tz=1,\, z\ge 0}\{s^Ty+\zeta\} \mid B^Ty+e\zeta \ge 0,\ \|y\|'=1\right\}.$$

On noting that for a fixed y, $y^Ts = y^TBz$ is maximized over $s \in S$ or equivalently over $z \ge 0$, $e^Tz = 1$, by taking y^TBz equal to $\max_{1\le j\le k} y^TB_{.j}$, and noting that

$$\zeta = \max_{1\le j\le k} -y^TB_{.j} = -\min_{1\le j\le k} y^TB_{.j},$$

we find that the desired upper bound (2.8) follows from (2.9). □

Todd [10] gave a geometric interpretation of the upper bound (2.8) that can cut it by a factor of 2 as follows. He noted that for a fixed $y \in R^n$ such that $\|y\|' = 1$, the term in the parentheses of (2.8), e.g., by Proposition 1.1, gives the width of a slab parallel to the hyperplane $y^T x = 0$ and containing the set S. Hence the minimum, over all y such that $\|y\|' = 1$, of such slab widths bounds the diameter of the largest ball that can be inscribed in S, and hence the upper bound (2.8) can be cut by a factor of 2 as stated in the following corollary. This result is a special case of (1.3) in Gritzmann and Klee [6], which states that the inner n-radius is bounded above by the outer 1-radius. The bound is tight for symmetric convex bodies and is always within a factor of about $\sqrt{n}$ [6].

Corollary 2.6 (improved upper bound for largest inscribed ball).

$$\rho = \|\bar{s} - p(\bar{s})\| \leq \frac{1}{2} \min_{\|y\|'=1} \left(\max_{1\leq j\leq k} y^T B_{.j} - \min_{1\leq j\leq k} y^T B_{.j} \right). \tag{2.10}$$

In contrast to these rather nontrivial problems for the boundary projection and largest radius problems for a polyhedral set characterized by (1.1), we turn to the considerably simpler corresponding problems for a polyhedral set characterized by (1.2).

3. Boundary projection and largest ball for polyhedral set T.

We consider in this section the polyhedral set T defined by (1.2) and state results parallel to Theorems 2.2 and 2.4. The analysis below is essentially contained in [3] for the 2-norm and in [6], but the argument is very simple from our earlier considerations. For a given point $s \in T$, the distance between s and an arbitrary-norm projection of s onto any of the hyperplane $A_i x = b_i$, $i = 1, \dots, m$, defining T, is given by the proof of (1.14) of [6] or Proposition 1.1. Hence the closest point to s on the boundary $\mathrm{bd}(T)$ of T is given by the closest of these projections to s, which is the first boundary point that an expanding ball around s would touch. This leads then to the following straightforward result.

Theorem 3.1 (boundary projection $p(s)$ for $s \in T$). *The distance between $s \in T$ and its projection $p(s)$ onto the boundary $\mathrm{bd}(T)$ of T using a general norm $\|\cdot\|$ on R^n is given by*

$$\|s - p(s)\| = \min_{i=1,\dots,m} \frac{A_i s - b_i}{\|A_i\|'}. \tag{3.1}$$

A projection $p(s)$ of p onto boundary $\mathrm{bd}(T)$ is given by

$$p(s) = s - \frac{A_{\bar{i}} s - b_{\bar{i}}}{\|A_{\bar{i}}\|'} y(A_{\bar{i}}), \tag{3.2}$$

where $\|\cdot\|'$ is the dual norm to $\|\cdot\|$ and

$$y(A_{\bar{i}}) \in \arg \max_{\|y\|=1} A_{\bar{i}} y, \tag{3.3}$$

and $\bar{i}$ is any index that solves (3.1).

It is interesting to contrast the simplicity of finding a minimum of m numbers specified by (3.1) in order to determine the distance between $s \in T$ and its projection $p(s)$ onto the boundary $\mathrm{bd}(T)$ of T with the nonconvex program (2.2) required for the corresponding problem for the set S.

Using the above result we can recover the linear program of Gritzmann and Klee [6, (1.14)] for the problem of determining the largest ball in R^n, with radius measured by an arbitrary norm $\|\cdot\|$, that is contained in T. For the 2-norm a different linear programming formulation is given by Eaves and Freund [3, p. 143].

THEOREM 3.2 (largest ball inscribed in T). *The radius ρ of the largest ball, measured by an arbitrary norm $\|\cdot\|$ on R^n, that can be inscribed in T is given by*

$$\rho = \|\bar{s} - p(\bar{s})\| = \sup_{\rho,s} \{\rho \mid A_i s - \|A_i\|'\rho \geq b_i,\ i = 1, \ldots, m\}, \tag{3.4}$$

where $\|\cdot\|'$ denotes the dual norm to $\|\cdot\|$ on R^n.

4. Summary and conclusion. We have formulated the problem of projecting a point in a polytope, defined by a convex combination of points in R^n, onto its boundary, as a mathematical program that has linear constraints and objective function and one nonconvex cylindrical constraint. For the 1-norm this problem can be solved by solving $2n$ linear programs. When the set is given as the intersection of a number of closed half-spaces, the projection problem is a straightforward problem of finding the minimum of m numbers where m is the number of half-spaces defining the set. We have also related our boundary-projection problem to the largest-inscribed-ball problem considered by others [3, 4, 6, 7]. For the case of intersecting half-spaces, this problem can be solved by a single linear program. For the other case of a polyhedral set defined as a convex combination of given points, the problem is formulated as a maxmin problem of a bilinear function on the product of two sets, one of which contains a convex cylindrical constraint. Again, for the 1-norm, this problem can be solved by linear programming while for other norms it is NP-hard. It is interesting to note the disparate difficulty of the problems depending on the polyhedral set characterization.

Acknowledgments. I am indebted to M. J. Todd for many important suggestions for revising this paper and to two referees for constructive criticisms that made me aware of the vast world of computational geometry.

REFERENCES

[1] C. BERGE, *Topological Spaces*, McMillan, New York, 1963.

[2] A. CHARNES, W. COOPER, A. Y. LEWIN, AND L. M. SEIFORD, EDS., *Data Envelopment Analysis*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[3] B. C. EAVES AND R. M. FREUND, *Optimal scaling of balls and polyhedra*, Math. Programming, 23 (1982), pp. 138–147.

[4] R. M. FREUND AND J. B. ORLIN, *On the complexity of four polyhedral set containment problems*, Math. Programming, 33 (1985), pp. 139–145.

[5] A. J. GOLDMAN AND A. W. TUCKER, *Polyhedral convex cones*, in Linear Inequalities and Related Systems, H. W. Kuhn and A. W. Tucker, eds., Princeton University Press, Princeton, NJ, 1956, pp. 19–40.

[6] P. GRITZMANN AND V. KLEE, *Inner and outer j-radii of convex bodies in finite-dimensional normed spaces*, Discrete Comput. Geom., 7 (1992), pp. 255–280.

[7] P. GRITZMANN AND V. KLEE, *Computational complexity of inner and outer j-radii of polytopes in finite-dimensional normed spaces*, Math. Programming, 59 (1993), pp. 163–213.

[8] O. L. MANGASARIAN, *Nonlinear Programming*, McGraw–Hill, New York, 1969; reprint: Classics in Applied Mathematics 10, SIAM, Philadelphia, PA, 1994.

[9] O. L. MANGASARIAN, *Arbitrary-norm separating plane*, Oper. Res. Lett., 24 (1999), pp. 15–23.

[10] M. J. TODD, *Private communication,* July 1998.

SIAM J. Optim.
Vol. 9, No. 4, pp. 1135–1148

A REVISED MODIFIED CHOLESKY FACTORIZATION ALGORITHM[*]

ROBERT B. SCHNABEL[†] AND ELIZABETH ESKOW[†]

Dedicated to John Dennis on his 60th birthday, in appreciation for his wonderful contributions to the science and the people of nonlinear optimization

Abstract. A modified Cholesky factorization algorithm introduced originally by Gill and Murray and refined by Gill, Murray, and Wright is used extensively in optimization algorithms. Since its introduction in 1990, a different modified Cholesky factorization of Schnabel and Eskow has also gained widespread usage. Compared with the Gill–Murray–Wright algorithm, the Schnabel–Eskow algorithm has a smaller a priori bound on the perturbation, added to ensure positive definiteness, and some computational advantages, especially for large problems. Users of the Schnabel–Eskow algorithm, however, have reported cases from two different contexts where it makes a far larger modification to the original matrix than is necessary and than is made by the Gill–Murray–Wright method. This paper reports on a simple modification to the Schnabel–Eskow algorithm that appears to correct all the known computational difficulties with the method, without harming its theoretical properties or its computational behavior in any other cases. In new computational tests, the modifications to the original matrix made by the new algorithm appear virtually always to be smaller than those made by the Gill–Murray–Wright algorithm, sometimes by significant amounts. The perturbed matrix is allowed to be more ill-conditioned with the new algorithm, but this seems to be appropriate in the known contexts where the underlying problem is ill-conditioned.

Key words. Cholesky factorization, optimization, nonpositive definite

AMS subject classifications. 65F30, 65K10, 15A23

PII. S105262349833266X

1. Introduction.

Modified Cholesky factorizations are widely used in optimization. A numerically stable modified Cholesky factorization algorithm was introduced by Gill and Murray in 1974 [9]. Given a symmetric, not necessarily positive definite matrix $A \in R^{n\times n}$, a modified Cholesky factorization calculates a Cholesky (i.e., LL^T or LDL^T) factorization of a positive definite matrix $A+E$ in a way that attempts to satisfy four goals: (1) If A is safely positive definite, E is 0; (2) if A is indefinite, $\|E\|_\infty$ is not much greater than the magnitude of the most negative eigenvalue of A, $\lambda_1(A)$; (3) $A+E$ is reasonably well-conditioned; (4) the cost of the factorization is only a small multiple of n^2 operations more than the $O(n^3)$ cost of the standard Cholesky factorization.

The factorization of Gill and Murray was subsequently refined by Gill, Murray, and Wright [10] (hereinafter referred to as GMW81). This version has been widely used in optimization methods since its inception. More recently, Schnabel and Eskow [12] introduced a factorization (hereinafter referred to as SE90) that is based on different techniques. Both factorizations choose E to be diagonal. Both satisfy properties 1, 3, and 4 mentioned above; they differ in how closely they satisfy property 2. The SE90 factorization has a significantly smaller a priori bound on $\|E\|$, where in this paper $\|E\|$ is always the infinity norm, and in computational tests it appears

[*] Received by the editors January 19, 1998; accepted for publication (in revised form) February 2, 1999; published electronically September 24, 1999. This work was supported by Air Force Office of Scientific Research grant F49620-97-1-0164, Army Research Office contract DAAH04-94-G-0228, and NSF grant CDA-9502956.

http://www.siam.org/journals/siopt/9-4/33266.html

[†] Department of Computer Science, University of Colorado at Boulder, Campus Box 430, Boulder, CO 80309-0430 (bobby@cs.colorado.edu, eskow@cs.colorado.edu).

that $\|E\|$ is smaller for the SE90 factorization than the GMW81 factorization in most cases as well. In practice, both factorizations appear to be very satisfactory for use in optimization algorithms and both are now widely used.

While the overall computational experience with the SE90 factorization since its introduction appears to have been quite good, a few instances have arisen where its performance has been poor. The SE90 paper [12] contained one example where the amount that is added to A, while within the theoretical bounds, is far larger than the magnitude of $\lambda_1(A)$ and also far larger than the amount added by the GMW81 factorization. In the first years following the publication of the SE90 algorithm, Wolfgang Hartmann of SAS Institute made us aware of another problem with similar behavior. More recently, David Gay, Michael Overton, and Margaret Wright encountered a class of problems, arising in primal-dual interior methods for constrained optimization [8], where the SE90 factorization again sometimes added far too much while the GMW81 factorization performed well.

All the known examples where the SE90 factorization adds too much (i.e., the ratio of $\|E\|$ to the magnitude of the most negative eigenvalue of A is greater than, say, 5) turn out to be matrices A that are the sum of a large positive semidefinite matrix B and a much smaller (in norm) indefinite matrix C. In these cases, one wants $\|E\|$ to be of order $\|C\|$, but instead the SE90 algorithm sometimes produces $\|E\|$ of order $\|B\|$. In the experience of Gay, Overton, and Wright, this introduced difficulties in the constrained optimization algorithm using the SE90 factorization that were not experienced when using the GMW81 factorization.

This paper introduces a simple modification to the SE90 modified Cholesky factorization that remedies these difficulties without harming its computational performance in any other known cases. The modification is to tighten slightly the condition under which the algorithm switches from phase 1 (standard Cholesky factorization) to phase 2, thereby making it slightly more likely to stay in phase 1 at a given iteration of the factorization. The theoretical effect of this change is to increase the upper bound on $\|E\|$ by a factor of at most 1.1. The modification resolves all the problem cases for the SE90 factorization of which we are aware.

Section 2 contains brief background on the modified Cholesky factorization, including the methods of GMW81 and SE90. This section is not intended to be a comprehensive reference; for more background on the modified Cholesky factorization or its use in optimization, see [10, 12] or Dennis and Schnabel [7]. Section 3 motivates the change in the SE90 example that this paper introduces, using the problematic example from [12]. In section 4 we present the complete new algorithm; several other very minor changes related to the main change and to badly conditioned problems are included. Section 5 briefly presents the theoretical results for the new method. In section 6 we summarize the results of computational tests of the new algorithm, and the methods of GMW81 and SE90, on the problems of Gay, Overton, and Wright, on a problem of Hartmann, and on the random test problems that were used in [12] to assess the behavior of the factorizations. Fortran code for the revised factorization will be available from the authors.

2. Brief background on modified Cholesky factorizations.

The modified Cholesky procedures of GMW81 and SE90, like the standard Cholesky factorization, can be viewed as recursive procedures. At the beginning of stage j, an $(n-j+1) \times (n-j+1)$ submatrix A_j remains to be factored (with $A_1 = A$). We assume that A_j

has the form

$$A_j = \begin{bmatrix} \alpha_j & a_j^T \\ a_j & \hat{A}_j \end{bmatrix}, \tag{2.1}$$

where $\alpha_j \in R$ is the current jth diagonal element and is called the pivot, a_j is the current vector of elements in column j below the diagonal, and $\hat{A}_j \in R^{(n-j)\times(n-j)}$. The modified Cholesky factorization chooses a nonnegative amount δ_j to add to α_j and then calculates $L_{jj} = \sqrt{\alpha_j + \delta_j}$, $L_{ij} = (a_j)_i / L_{jj}$, $i = j+1, \ldots, n$, and

$$A_{j+1} = \hat{A}_j - \frac{a_j a_j^T}{\alpha_j + \delta_j}. \tag{2.2}$$

The challenge in the modified Cholesky factorization is choosing each δ_j. The algorithm must guarantee that each $\delta_j = 0$ if A turns out to be safely positive definite. It also must employ some form of lookahead so that if A is not positive definite, δ_j is chosen to be an appropriate positive quantity beginning at a sufficiently early iteration of the factorization. This is not trivial; for example, waiting to set $\delta_j > 0$ until α_j first becomes negative and then adding amounts $\delta_j > -\alpha_j$ is not satisfactory, as it usually will result in $\|E\|$ much greater than $|\lambda_1|(A)$.

The GMW81 algorithm chooses each δ_j to be the smallest nonnegative number for which

$$0 \leq \frac{\|a_j\|_\infty^2}{\alpha_j + \delta_j} \leq \beta^2 \tag{2.3}$$

(with a minimum of $\delta_j = -2\alpha_j$ if $\alpha_j < 0$), where β is an a priori bound selected to minimize a worst-case bound on $\|E\|$ and also to ensure that $E = 0$ if A is safely positive definite. The result, with ϵ denoting machine precision, is

$$\beta^2 = \max\{\gamma, \xi/\sqrt{n^2-1}, \epsilon\}, \text{ where } \gamma = \max_i |A_{ii}| \text{ and } \xi = \max_{j<i} |A_{ij}|. \tag{2.4}$$

The requirement that $\beta^2 \geq \gamma$ guarantees $E = 0$ if A is positive definite. The overall a priori bound on $\|E\|_{GMW}$ depends on the largest element in brackets in (2.4); the smallest upper bound is

$$n^2\gamma + 2(n-1)\xi, \tag{2.5}$$

which is achieved when $\beta^2 = \xi/\sqrt{n^2-1}$.

The SE90 method is divided into two phases. The first phase consists of a normal Cholesky factorization in which the factors are overwritten on A. Step j of phase 1 is allowed to proceed only if α_j is positive and the smallest diagonal of the remaining submatrix at the next step, i.e., at step $j+1$, is "safely" positive, using the following test. Let the vector ζ be defined as

$$\zeta_i = A_{ii} - A_{ij}^2/\alpha_j, \quad i > j. \tag{2.6}$$

SE90 completes step j of the standard Cholesky algorithm only if

$$\min_i \zeta_i \geq \tau\gamma, \quad \text{where } \tau = \epsilon^{\frac{1}{3}}, \tag{2.7}$$

and otherwise switches to phase 2. Note that the components of ζ would be the diagonal elements of A_{j+1} if step j of the unmodified Cholesky procedure were to be completed; see (2.2). Satisfaction of (2.7) thus guarantees that all diagonal elements of A_{j+1} are positive so that there is no test of positivity of α_j for $j > 1$.

Let K_1 denote the number of steps completed during phase 1 so that $\delta_i = 0$ for $i = 1, \dots, K_1$. If $K_1 = n$, A is positive definite and the algorithm terminates. If $K_1 < n$, let

$$\hat{\gamma} = \max_{K_1<i}|A_{ii}| \text{ and } \hat{\xi} = \max_{K_1<i,j<i}|A_{ij}|. \tag{2.8}$$

It is shown in SE90 that the test (2.7) for termination of phase 1 guarantees that

$$\hat{\gamma} \leq \gamma \text{ and } \hat{\xi} \leq \gamma + \xi. \tag{2.9}$$

If $K_1 < n-2$, then for $j = K_1+1, \dots, n-2$, the value of δ_j is

$$\delta_j = \max\{0, -\alpha_j \ + \max\{\|a_j\|_1, \tau\gamma\}\delta_{j-1}\}. \tag{2.10}$$

This choice of δ_j causes the Gerschgorin intervals of the principal submatrices A_j to contract at each iteration and leads to the following bound:

$$\|E\|_{SE} \leq G + \frac{2\tau}{1-\tau}(G+\gamma), \tag{2.11}$$

where

$$G \leq (n-(k+1))(\gamma+\xi) \text{ if } K_1 > 0 \ \text{ and } \ G \leq \gamma + (n-1)\xi \ \text{ if } K_1 = 0. \tag{2.12}$$

The elements δ_{n-1} and δ_n are chosen in a special way that depends on the eigenvalues of the final 2×2 submatrix and still causes (2.12) to be satisfied.

The fact that the bound on $\|E\|$ is linear in n for the SE90 factorization (2.11) and (2.12) and quadratic in n for GMW81 factorization (2.5) is a key distinction between the methods. In practice, however, both methods usually achieve $\|E\|$ far smaller than these bounds, and $\|E\|$ is often within a factor of 2 of $-\lambda_1(A)$ when $\lambda_1 < 0$. In comparative tests in [12], the value of $\|E\|$ for the SE90 factorization is almost always smaller than for GMW81, although the performance of both methods is quite good. The performance of both algorithms is greatly aided by diagonal pivoting strategies employed at each iteration, which do not affect the theoretical properties. The additional cost of both factorizations is at most a small integer multiple of n^2 operations, which is negligible in comparison to the cost of the Cholesky factorization.

Recently, Cheng and Higham [2] have proposed a third type of modified Cholesky factorization, based upon the bounded Bunch–Kaufman pivoting strategy [1]. This factorization differs fundamentally from GMW81 and SE90 in that it adds a nondiagonal matrix to A by computing the symmetric indefinite factorization LBL^T of a symmetric permutation of A, where L is unit lower triangular and B is block diagonal with 1×1 or 2×2 blocks, and then perturbing B. This approach can be shown to perform well when the condition number of L is not too large. However, as Cheng and Higham state, the bound on $\|E\|$ is weak if the condition number of LL^T is large, and the worst-case upper bound is exponential in n. It is too early to assess whether this version of the modified Cholesky factorization will have a significant impact in the optimization community.

3. Motivating example for change to SE90 algorithm. All of the known matrices for which $\|E\|_{SE}$ is inordinately large appear to be of the form $A = B + C$, where B is a large positive semidefinite matrix (i.e., $B = B_1 B_1^T$ for some $B_1 \in R^{n\times m}$, $m < n$) and C is an indefinite or negative definite matrix with $\|C\| \ll \|B\|$. (Any symmetric indefinite matrix whose largest positive eigenvalue is much larger in magnitude than its most negative eigenvalue can be written in this form.) The potential for a modified Cholesky factorization to have difficulty on matrices of this type is clear: if phase 2 begins at or before step m (the rank of B), then the size of δ_j is, according to (2.10), likely to be proportional to $\|B\|$ and therefore large. If, on the other hand, phase 2 begins after step m, δ_j is likely to be proportional to $\|C\|$, which is, by assumption, much smaller than $\|B\|$. Of course, the structure of A, including the value of m, is not known to the algorithm.

The example in [12] showing where that algorithm has difficulty,

$$A = \begin{bmatrix} 1,890.3 & -1,705.6 & -315.8 & 3,000.3 \\ -1,705.6 & 1,538.3 & 284.9 & -2,706.6 \\ -315.8 & 284.9 & 52.5 & -501.2 \\ 3,000.3 & -2,706.6 & -501.2 & 4,760.8 \end{bmatrix}, \tag{3.1}$$

is of this form with $m = 1$. Its eigenvalues are 8,242.9, -0.248, -0.343, and -0.378. After permuting the largest diagonal element to the (1,1) position, the values of ζ computed from (2.6) are -0.265, -0.451, and -0.517, so that condition (2.7) fails and the algorithm switches immediately to phase 2 with $K_1 = 0$. Using (2.10), $\delta_1 =$ 1,049.4 is added to the first diagonal, and this is the ultimate value of $\|E\|$. The large value of δ_1 occurs because the calculation of δ_1 is based upon the Gerschgorin bounds for the large (in magnitude) matrix A.

In contrast, with the GMW81 algorithm we have $\alpha_1 = \beta^2 = 4,760.8$ and $\|a_1\|_\infty = 3,000.3$. Thus, inequality (2.3) is satisfied with $\delta_1 = 0$, so that no modification is made to the (1,1) diagonal. At the second iteration the algorithm adds 1.033 to the diagonal, which turns out to be its maximum element of E for this problem. This small value of δ_2 results because it is based upon the elements of A_2, and $\|A_2\| \ll \|A\|$.

To avoid modifying too soon, the remedy for the SE90 algorithm is to relax condition (2.7), the test for continuing phase 1, to allow phase 1 to continue even if A_{j+1} will have some small negative diagonal elements. In particular, we show in section 5 that if phase 1 continues when there is a suitably positive pivot and

$$\min_i \zeta_i \geq -\mu\gamma, \text{ where } 0 < \mu \leq 1 \tag{3.2}$$

and ζ is defined by (2.6), then the bounds (2.11) on element growth in phase 1 are only slightly worse; see Theorem 5.1. The advantage of using (3.2) rather than (2.7) is that deferring modification may lead to a smaller $\|E\|$ because the principal submatrix of the later iteration may have smaller elements.

If the test (3.2) is used on example (3.1) with $\mu = 0.1$ (or with any $\mu > 1.1\times 10^{-4}$), the first step of the *unmodified* Cholesky is allowed to proceed, so that $\delta_1 = 0$ and

$$A_2 = \begin{bmatrix} -0.451 & -0.041 & 0.124 \\ -0.041 & -0.265 & 0.061 \\ 0.124 & 0.061 & -0.517 \end{bmatrix}. \tag{3.3}$$

Since all diagonal elements of A_2 are negative, $K_1 = 1$ and the procedure switches to phase 2, giving $E_{2,2} = 0.3666$, $E_{3,3} = E_{4,4} = 0.6649$. That is, the ratio $\|E\|/(-\lambda_1(A))$ is a very acceptable 1.76, as opposed to a poor 2,778 for the SE90 algorithm (and 2.73 for the GMW81 algorithm).

4. The complete revised factorization algorithm. A complete pseudocode description of our revised modified Cholesky factorization is given in Algorithm 4.1. The key change from the SE90 algorithm is the one discussed in section 3: the lookahead condition under which the algorithm switches from phase 1 to phase 2 is changed from $\min\{(A_{j+1})_{ii}\} \leq \tau\gamma$ for some small positive τ (2.7) to $\min\{(A_{j+1})_{ii}\} \leq -\mu\gamma$ for some $\mu \leq 1$ (3.2). Our implementation uses $\mu = 0.1$.

Several changes have been made to the algorithm in addition to checking (3.2) as part of continuing in phase 1:

1. Since we now allow small negative diagonal elements in A_j in phase 1, we must check that the pivot is positive. The test we insert to proceed with step j of phase 1 is that the pivot element α_j (the maximum diagonal element of A_j) must satisfy

$$\alpha_j \geq \bar{\tau}\gamma, \quad \text{where} \ \ \bar{\tau} = \epsilon^{\frac{2}{3}}. \tag{4.1}$$

This requirement ensures not only that the pivot is positive but also that the new algorithm retains a (mainly theoretically useful) bound on the condition number of L analogous to that for the SE90 algorithm.

2. At step j of phase 1, even if (4.1) is satisfied a branch is made to phase 2 if

$$\min\nolimits_{i>j} A_{ii} < -\mu\alpha_j, \tag{4.2}$$

where μ is the quantity from (3.2). Note that because (3.2) was satisfied at the previous step of phase 1, it must be true that $\min_{i>j} A_{ii} \geq -\mu\gamma$. When (4.2) holds, the remaining submatrix A_j tends to have at least one negative eigenvalue that is comparable in magnitude to the other eigenvalues of A_j. In this case, the test (4.2) leads to an earlier termination of phase 1. Practical experience suggests that this leads to a smaller $\|E\|$; this is illustrated in section 6.

3. A reduced lower bound, $\bar{\tau}\gamma$, is imposed on the modified diagonal $A_{jj} + \delta_j$, where $\bar{\tau}$ is defined in (4.1). (In the SE90 algorithm, this lower bound is the larger value $\tau\gamma$.) This change leads to two differences between the new algorithm and SE90 when applied to badly conditioned "barely indefinite" matrices for which $|\lambda_1| \ll \|A\|$: $\|E\|$ tends to be smaller with the new algorithm—only slightly larger than $-\lambda_1$; but the condition number of the modified matrix tends to be larger—roughly $1/\bar{\tau} = \epsilon^{-\frac{2}{3}}$ rather than $\epsilon^{-\frac{1}{3}}$ as in SE90. We expect that trading a larger condition number for a smaller modification often will be desirable—for example, when the Hessian at the solution is ill-conditioned and the reduced bound allows quadratic convergence to be retained.

4. Special logic is needed to treat the case when $K_1 = n - 1$. (With SE90, step $n-1$ proceeds only if step n can also be completed, so that this case does not occur.)

The only portions of our code for the modified Cholesky factorization that are not reflected in Algorithm 4.1 are brief special cases to deal with matrices of dimension one and zero matrices.

Algorithm 4.1. Revised modified Cholesky decomposition algorithm. Given $A \in \Re^{n\times n}$ symmetric (stored in lower triangle) and $\tau, \bar{\tau}, \mu$ (e.g., $\tau = (macheps)^{\frac{1}{3}}$, $\bar{\tau} = (macheps)^{\frac{2}{3}}$, $\mu = 0.1$), find factorization LL^T of $A + E$, $E \geq 0$

```
    $phaseone$ := true
$\gamma := \max_{1 \leq i \leq n}\{|A_{ii}|\}$
$j := 1$

(*Phase one, $A$ potentially positive definite*)
While $j \leq n$ and phaseone = true do
if $\max_{j \leq i \leq n}\{A_{ii}\} < \bar{\tau}\gamma$ or $\min_{j \leq i \leq n}\{A_{ii}\} < -\mu(\max_{j \leq i \leq n}\{A_{ii}\})$
    then phaseone := false (*go to phase two*)
else
    (*Pivot on maximum diagonal of remaining submatrix*)
        $i$ := index of $\max_{j \leq i \leq n}\{A_{ii}\}$
        if $i \neq j$, switch rows and columns of $i$ and $j$ of $A$
    if $\min_{j+1 \leq i \leq n}\{A_{ii} - A_{ij}^2/A_{jj}\} < -\mu\gamma$
        then phaseone := false (*go to phase two*)
        else (*perform $jth$ iteration of factorization*)
            $L_{jj} = \sqrt{A_{jj}}$ (*$L_{jj}$ overwrites $A_{jj}$*)
            For $i := j+1$ to $n$ do
                $L_{ij} := A_{ij}/L_{jj}$ (*$L_{ij}$ overwrites $A_{ij}$*)
                For $k := j+1$ to $i$ do
                    $A_{ik} = A_{ik} - L_{ij}L_{kj}$
            $j := j+1$
(*end phase one*)

(*Phase two, $A$ not positive definite*)
if $phaseone$ = false and $j = n$, then
    $\delta$ (* = $E_{nn}$*) := $-A_{nn} + \max\{\tau(-A_{nn})/(1-\tau), \bar{\tau}\gamma\}$
    $A_{nn} := A_{nn} + \delta$
    $L_{nn} = \sqrt{A_{nn}}$

if $phaseone$ = false and $j < n$, then
    $k := j - 1$ (*$k$ = number of iterations performed in phase one*)
    (* Calculate lower Gerschgorin bounds of $A_{k+1}$*)
        For $i := k+1$ to $n$ do
            $g_i := A_{ii} - \sum_{j=k+1}^{i-1}|A_{ij}| - \sum_{j=i+1}^{n}|A_{ji}|$


        (*Modified Cholesky Decomposition*)
    For $j := k+1$ to $n-2$ do
        (*Pivot on maximum lower Gerschgorin bound estimate*)
            $i$ := index of $\max_{j \leq i \leq n}\{g_i\}$
            if $i \neq j$, switch rows and columns of $i$ and $j$ of $A$
        (*Calculate $E_{jj}$ and add to diagonal*)
            $normj := \sum_{i=j+1}^{n}|A_{ij}|$
            $\delta(* = E_{nn}*) := \max\{0, -A_{jj} + \max\{normj, \bar{\tau}\gamma\}, \delta prev\}$
            if $\delta > 0$, then
                $A_{jj} := A_{jj} + \delta$
                $\delta prev := \delta$ (* $\delta prev$ will contain $\|E\|_\infty$*)
        (*Update Gerschgorin bound estimates*)
            if $A_{jj} \neq normj$, then
                $temp := 1 - normj/A_{jj}$
            for $i := j + 1$ to $n$ do
                $g_i := g_i + |A_{ij}| * temp$
        (*Perform $j$th iteration of factorization*)
            same code as in phase one

    (*Final $2 \times 2$ submatrix*)
        $\lambda_{lo}$, $\lambda_{hi}$ := eigenvalues of $\begin{bmatrix} A_{n-1,n-1} & A_{n,n-1} \\ A_{n,n-1} & A_{n,n} \end{bmatrix}$
        $\delta := \max\{0, -\lambda_{lo} + \max\{\tau(\lambda_{hi} - \lambda_{lo})/(1-\tau), \bar{\tau}\gamma\}, \delta prev\}$
        if $\delta > 0$, then
            $A_{n-1,n-1} := A_{n-1,n-1} + \delta$
            $A_{n,n} := A_{n,n} + \delta$
            $\delta prev := \delta$
        $L_{n-1,n-1} := \sqrt{A_{n-1,n-1}}$ (*overwrites $A_{n-1,n-1}$*)
        $L_{n,n-1} := A_{n,n-1}/L_{n-1,n-1}$ (*overwrites $A_{n,n-1}$*)
        $L_{n,n} := (A_{n,n} - L^2_{n,n-1})^{1/2}$ (*overwrites $A_{n,n}$*)
(*End phase two*)
```

5. Upper bound on $\|E\|$.

A key property of the SE90 factorization is the bound (2.7) on $\|E\|$. In this section we show that the relaxed lookahead strategy of the revised factorization causes only a small growth in this bound. In particular, the term $(\gamma+\xi)$ in (2.8) increases to $(1+\mu)\gamma+\xi$. (Recall that $\mu \le 1$; in our implementation $\mu = 0.1$.) Thus the bound grows by at most $(1+\mu)$ and is still linear in n.

There are two main components in the proof of the bound on $\|E\|$ in SE90. One is the proof [12, Lemma 5.1.1 and Theorem 5.1.2] that each δ_j in phase 2 is less than the magnitude of the most negative Gerschgorin bound of the matrix A_j when the algorithm enters phase 2. This result is unaffected by the changes in our revised algorithm. The second main component of the proof is the bound on the growth in the elements of A during phase 1 [12, Theorem 5.2.1]. This result and proof are modified in a minor way by the new lookahead strategy. For completeness, we include the new statement and proof of this result below. The only new portions are the various terms $\mu\gamma$ below, all of which are absent for the results in [12] about that algorithm. Note that Theorems 5.1 and 5.2 are true independent of whether pivoting is used at all or what pivoting strategy is used.

THEOREM 5.1. *Let $A \in R^{n\times n}$ and let $\gamma = \max\{|A_{ii}|, 1 \le i \le n\}$, $\xi = \max\{|A_{ij}|, 1 \le i < j \le n\}$. Suppose we perform the standard Cholesky decomposition as described in phase 1 of Algorithm 4.1 for $k \ge 1$ iterations, yielding the principal submatrix $A_{k+1} \in R^{(n-k)\times(n-k)}$ (whose elements are denoted $(A_{k+1})_{ij}, k+1 \le i, j \le n$), and let $\hat{\gamma} = \max\{|(A_{k+1})_{ii}|, k+1 \le i \le n\}$ and $\hat{\xi} = \max\{|(A_{k+1})_{ij}|, k+1 \le i < j \le n\}$. If $(A_{k+1})_{ii} \ge -\mu\gamma, k+1 \le i \le n$ for some $\mu \le 1$, then $\hat{\gamma} \le \gamma$ and $\hat{\xi} \le \xi + (1+\mu)\gamma$.*

Proof. Let $A = [\begin{smallmatrix} B & C^T \\ C & F \end{smallmatrix}]$, where $B \in R^{k\times k}$, $C \in R^{(n-k)\times k}$, $F \in R^{(n-k)\times(n-k)}$. After k iterations of the Cholesky factorization, the first k columns of the Cholesky factor L have been determined; denote them by $[\begin{smallmatrix} \bar{L} \\ M \end{smallmatrix}]$, where $\bar{L} \in R^{k\times k}$ is triangular and $M \in R^{(n-k)\times k}$. Then

$$B = \bar{L}\bar{L}^T, \quad C = M\bar{L}^T, \quad F = MM^T + A_{k+1}. \tag{5.1}$$

Let m_i^T denote the ith row of M. From (5.1), $F_{ii} = \|m_i^T\|_2^2 + (A_{k+1})_{ii}, k+1 \le i \le n,$

so that from $F_{ii} \leq \gamma$ and $(A_{k+1})_{ii} \geq -\mu\gamma$,

$$\|m_i^T\|_2^2 \leq (1+\mu)\gamma. \tag{5.2}$$

Thus for any off-diagonal element of A_{k+1}, (5.1), (5.2), and the definition of ξ imply

$$|(A_{k+1})_{ij}| \leq |F_{ij} - (m_i^T)(m_j^T)^T| \leq \xi + (1+\mu)\gamma, \tag{5.3}$$

which shows that $\hat{\xi} \leq \xi + (1+\mu)\gamma$. For all the diagonal elements of A_{k+1}, $(A_{k+1})_{ii} \geq -\mu\gamma$, $\mu \leq 1$, (5.1), and the definition of γ imply

$$-\mu\gamma \leq (A_{k+1})_{ii} \leq F_{ii} \leq \gamma, \tag{5.4}$$

which shows that $\hat{\gamma} \leq \gamma$ and completes the proof. □

The only other change in the revised algorithm that could affect the bound on $\|E\|$ is the use of $\bar{\tau}$ where SE90 uses τ. Since $\bar{\tau} < \tau$, this affects the statement of the main result but not the bound on $\|E\|$. The new growth bound is given below; it is a minor modification of Theorem 5.3.2 of [12].

THEOREM 5.2. *Let A, γ, and ξ be defined as in Theorem* 5.1 *and suppose the modified Cholesky factorization Algorithm* 4.1 *is applied to A, resulting in the factorization LL^T of $A + E$. If A is positive definite and at each iteration $L_{jj}^2 \geq \bar{\tau}\gamma$, then $E = 0$. Otherwise, E is a nonnegative diagonal matrix with*

$$\|E\| \leq \text{Gersch} + \frac{2\tau}{1-\tau}(\text{Gersch} + \gamma), \tag{5.5}$$

where Gersch *is the maximum of the negative of the lower Gerschgorin bounds $\{ g_i \}$ of A_{k+1} that are calculated at the start of phase* 2. *If $k = 0$, then* Gersch $\leq \gamma + (n-1)\xi$; *otherwise*

$$\text{Gersch} \leq (n-(k+1))((1+\mu)\gamma + \xi). \tag{5.6}$$

6. Computational results. We have tested our revised factorization method, and the GMW81 and SE90 methods, on the problems where the SE90 method had difficulties as well as on the broad test set from [12] and a modification of one of these problem sets designed to be especially difficult for our methods for reasons described below. This section summarizes and analyzes the computational results.

As mentioned in section 1, the modifications to the SE90 algorithm were motivated in a large part by the matrices sent to us by Gay, Overton, and Wright. These matrices are condensed primal-dual matrices used in barrier methods for constrained optimization. The 33 matrices sent to us were from problems in which the overall optimization method using the SE90 factorization performed worse than the same optimization method using other modified Cholesky factorizations, including GMW81. For each problem, Gay, Overton, and Wright attempted to locate the first optimization iteration where the algorithm using SE90 took a poorer step than the algorithm using other modified Cholesky factorizations, and they sent the Hessian matrix from this iteration. It turned out that for two-thirds of these matrices, the SE90 algorithm was adding more than GMW81, by as much as a factor of 10^2 to 10^7 in eight cases. The problems are quite small, with all but two having dimension between 6 and 15 and the remaining two having dimension 26 and 55.

Table 6.1 summarizes the performance of the GMW81 and SE90 algorithms, and our new Algorithm 4.1, on these 33 problems. The first column encodes the problem

Table 6.1
Performance of existing and new methods on indefinite Hessian matrices.

Problem	$\|E\|/(-\lambda_1(A))$			Log_{10} Cond'n number of $A+E$		
	GMW81	SE90	Revised SE90	GMW81	SE90	Revised SE90
A6_1	1.36	3.57e+02	1.08	5	1	9
A6_2	4.84	1.18	1.18	3	5	7
A6_3	4.84	1.19	1.20	4	5	6
A6_4	2.50	1.27	1.27	5	5	8
A6_5	2.34	6.50	1.44	5	3	9
A6_6	1.69	2.94	1.20	8	5	10
A6_7	1.95	4.61	1.33	12	5	10
A6_8	1.95	6.61	1.13	8	5	10
A6_9	1.95	47.22	1.12	8	5	10
A6_10	5.88	5.39e+06	1.07	8	1	11
A6_11	2.33	7.25e+06	1.64	8	1	7
A6_12	4.84	1.19	1.20	4	5	6
A6_13	2.18	1.32	1.32	2	5	6
A6_14	4.84	1.19	1.20	4	5	6
A6_15	5.18	1.09	1.09	2	5	5
A6_16	2.18	1.32	1.32	2	5	6
A6_17	1.52	1.24	1.24	2	5	6
A13_1	2.25	8.93e+03	1.18	10	5	10
A13_2	2.59	1.50e+04	1.31	8	5	10
A15_1	2.42	2.54e+07	1.89	9	5	11
A15_2	2.37	3.89e+05	1.44	9	3	10
A15_3	1.95	2.18	1.50	6	5	10
B6_1	4.90	52.41	1.77	3	1	8
B6_2	4.49	45.86	2.31	2	1	7
B7_1	1.66	3.45	1.06	2	2	2
B7_2	1.93	11.00	1.30	2	1	7
B7_3	1.96	6.99	1.22	2	1	6
B7_4	1.92	5.32	1.18	2	1	6
B8_1	4.16	871.2	1.27	12	5	10
B13_1	0	27.14 (abs)	0	9	5	9
B13_2	1.76	7.84	1.29	7	5	10
B26_1	9.83	2.23	2.36	1	3	7
B55_1	3.50	1.71	1.71	1	5	6

as follows: the set (A is the initial set sent to us, B a second, later set sent to us after we had made some but not all of the modifications reported in this paper), the dimension, and the sequence number within this set and dimension. The second, third, and fourth columns report the ratio of $\|E\|/(-\lambda_1(A))$ for each factorization. (For problem B13_1, which is positive definite, these columns contain $\|E\|$ instead.) The last three columns report the integer part of the base 10 log of the l_2 condition number of $A+E$.

The results show that the new algorithm produces a reasonable value of $\|E\|$ in all cases. The ratio $\|E\|/(-\lambda_1(A))$ is less than 2.4 for all 33 problems, less than 2 for 31 of the 33, and less than 1.4 for 24 of the 33 problems. The value of $\|E\|$ is smaller than that produced by the GMW81 algorithm on all 33 problems except the positive definite matrix, where both produce $E=0$. The values of $\|E\|/(-\lambda_1(A))$ produced by GMW81 are generally in the range of 2 to 5 for these problems. It is not clear, however, that this larger value makes the GMW81 algorithm any less effective in an optimization context. The value of $\|E\|/(-\lambda_1(A))$ for the new algorithm is essentially the same as for SE90 in 10 of the 33 cases and is lower in the other 23.

The results also show that the condition numbers of $A+E$ produced by the new algorithm are considerably higher than for the SE90 algorithm, with 13 of the 33 as high as 10^9 to 10^{11}. As discussed in section 4, this stems directly from the reduction in the minimum allowable value of $(A_{jj}+\delta_j)$ from $(macheps)^{1/3}\gamma$ to $(macheps)^{2/3}\gamma$. This reduction, however, allows the algorithm to produce values of $\|E\|$ hardly larger than $-\lambda_1(A)$ on indefinite problems where $-\lambda_1(A)$ is very small compared to $\|A\|$. The condition numbers produced by the GMW81 algorithm are almost always smaller than those produced by the new algorithm, although the two largest condition numbers produced by GMW81 on this test set, both roughly 10^{12}, exceed the largest condition numbers produced by the new algorithm. It should be noted that the original matrices in these problems are themselves extremely ill-conditioned, and it is important for the modified Cholesky to retain this property.

The change in performance of the new algorithm versus the SE90 algorithm on these problems is directly related to the new algorithm's ability to defer adding to the diagonal until a later iteration of the factorization. The new algorithm begins adding to the diagonal at the same iteration as SE90 in 10 cases (all where SE90 already performed satisfactorily) and later in the remaining 23 cases. In eight cases it begins adding only one iteration later, but even this can lead to $\|E\|$ being orders of magnitude smaller, as was shown by the example in section 3. In some cases the new algorithm begins adding 7 to 10 iterations later than SE90 on problems of dimension no greater than 15. GMW81 and the new algorithm are very similar when they begin adding to the diagonal: they begin at the same iteration in 21 of the 33 cases, with GMW81 beginning earlier in seven of the remaining 12 and later in the other five. The test (4.2) has an impact on five of these 33 problems (A6_3, A6_10, A6_12, A6_14, and B8_1), reducing $\|E\|/(-\lambda_1(A))$ from between 2 and 3.4 to 1.3 or less while also reducing the condition number of $A+E$ by about one order of magnitude in comparison to the new algorithm without (4.2).

We had received one other report of difficulties in using the SE90 algorithm, from Hartmann concerning problems arising in ridge regression. In the example sent to us by Hartmann, $n=6$ and the matrix

$$\begin{bmatrix} 14.8253 & -6.4243 & 7.8746 & -1.2498 & 10.2733 & 10.2733 \\ -6.4243 & 15.1024 & -1.1155 & -0.2761 & -8.2117 & -8.2117 \\ 7.8746 & -1.1155 & 51.8519 & -23.3482 & 12.5902 & 12.5902 \\ -1.2498 & -0.2761 & -23.3482 & 22.7967 & -9.8958 & -9.8958 \\ 10.2733 & -8.2117 & 12.5902 & -9.8958 & 21.0656 & 21.0656 \\ 10.2733 & -8.2117 & 12.5902 & -9.8958 & 21.0656 & 21.0656 \end{bmatrix} \tag{6.1}$$

is positive semidefinite with one zero eigenvalue and five positive eigenvalues ranging from 5 to 82. The positive semidefinite case can be considered a limiting case of the class of problems that motivated our revision. The SE90 algorithm adds 7.50 to the diagonal at iterations 3 through 6, which is undesirable. The GMW81 algorithm adds 1.67×10^{-14} at iteration 6 and produces a condition number of 4.9×10^{15} for $A+E$, whereas our new algorithm adds 1.90×10^{-9} at iteration 6 and produces a condition number of 8.7×10^{10}. Both seem reasonable; the higher value of δ_6 and lower condition number from the new algorithm, compared to GMW81, stem directly from our tolerance on the lowest allowable value of $(A_{jj}+\delta_j)$ discussed in section 4.

We also reran all the test problems reported in [12]. These consist of 120 randomly generated problems, 10 each of dimension 25, 50, and 75 for each of four eigenvalue ranges: -1 to 1, -10^{-4} to -1, -1 to 10^4 and one negative eigenvalue, and -1 to 10^4 and three negative eigenvalues.

The behavior of the new algorithm on the first two sets of problems (eigenvalue ranges [-1,1] and [-10^{-4}, -1]) is identical to the SE90 algorithm for all problems. As reported in [12], for both of these classes of matrices the SE90 algorithm (and the new algorithm) produce values of $\|E\|/(-\lambda_1(A))$ quite close to 1 and considerably lower than the GMW81 algorithm (by one to two orders of magnitude) while also producing much smaller condition numbers than the GMW81 algorithm (by about six orders of magnitude).

The behavior of the factorization algorithms on the sets with eigenvalue range -1 to 10^4 are very similar to the behavior on the Gay–Overton–Wright problems that helped motivate this paper, since their characteristics are very similar. Indeed, the example in section 3, given in [12], came from the [-1, 10^4] set with $n = 25$ and three negative eigenvalues and was the only bad case for the SE90 algorithm of the 120 test problems in that paper. In this paper, we include the results for the one negative eigenvalue set with $n = 75$ (Figures 6(a) and 6(b)), as they are typical but more marked than the $n = 25$ and 50 results. We also include a new set with $n = 75$, eigenvalue range -1 to 10^4, and nine negative eigenvalues (Figures 6(c)–(e)), as it is a more extreme example of the problems with the SE90 algorithm than the three negative eigenvalue sets.

The results of these two test sets again show that the values of $\|E\|/(-\lambda_1(A))$ produced by the new algorithm are very good, generally about 1.5 for the first set and 2 for the second. The values produced by the GMW81 algorithm are slightly higher but also very good. The values produced by the SE90 algorithm on the second set are very high (between 70 and 200) in four of the cases; for the first set they are satisfactory but the new algorithm is better. As with some of the Gay–Overton–Wright problems, the condition numbers produced by the new algorithm in these cases are around 10^{10}, while for the GMW81 algorithm they are around 10^5.

In summary, these results indicate that the modifications introduced in this paper have removed the known difficulties with the SE90 algorithm. The new algorithm produces values of $\|E\|/(-\lambda_1(A))$ in the range 1 to 2.5 for all test matrices considered, including all that are problematic for the SE90 algorithm. The values of $\|E\|/(-\lambda_1(A))$ are virtually always lower than those produced by the GMW81 algorithm, sometimes considerably so. The modifications result in condition numbers of $A + E$ of order $(macheps)^{-2/3}$ in cases when A is barely indefinite ($0 < -\lambda_1(A) \ll \|A\|$). The matrices produced by the GMW81 algorithm generally are better conditioned than those produced by the new algorithm in these cases, although the highest condition numbers produced by the GMW81 algorithm are higher than for the new algorithm. The new algorithm, like SE90, produces very well conditioned matrices in the other types of test cases.

In our opinion, these test results indicate good performance for both the GMW81 algorithm and the new algorithm. Which is used in an optimization context may depend upon the context or upon factors other than those considered in this paper. For example, SE90 has proved useful for large-scale codes, including multifrontal approaches, where one does not want to process the full matrix A at once [4]. Here the fact that GMW81 requires a preprocessing step that requires all of A (to compute ξ, which is not used in SE90 or the new algorithm) is the critical difference. (Pivoting is not used in these implementations; recall that this does not weaken the theoretical properties of our algorithm.) In a different context, the SE90 algorithm has led to very good performance when used as a preconditioner in conjugate gradient codes in the LANCELOT software package [3]; it has also been used in this manner by [11].

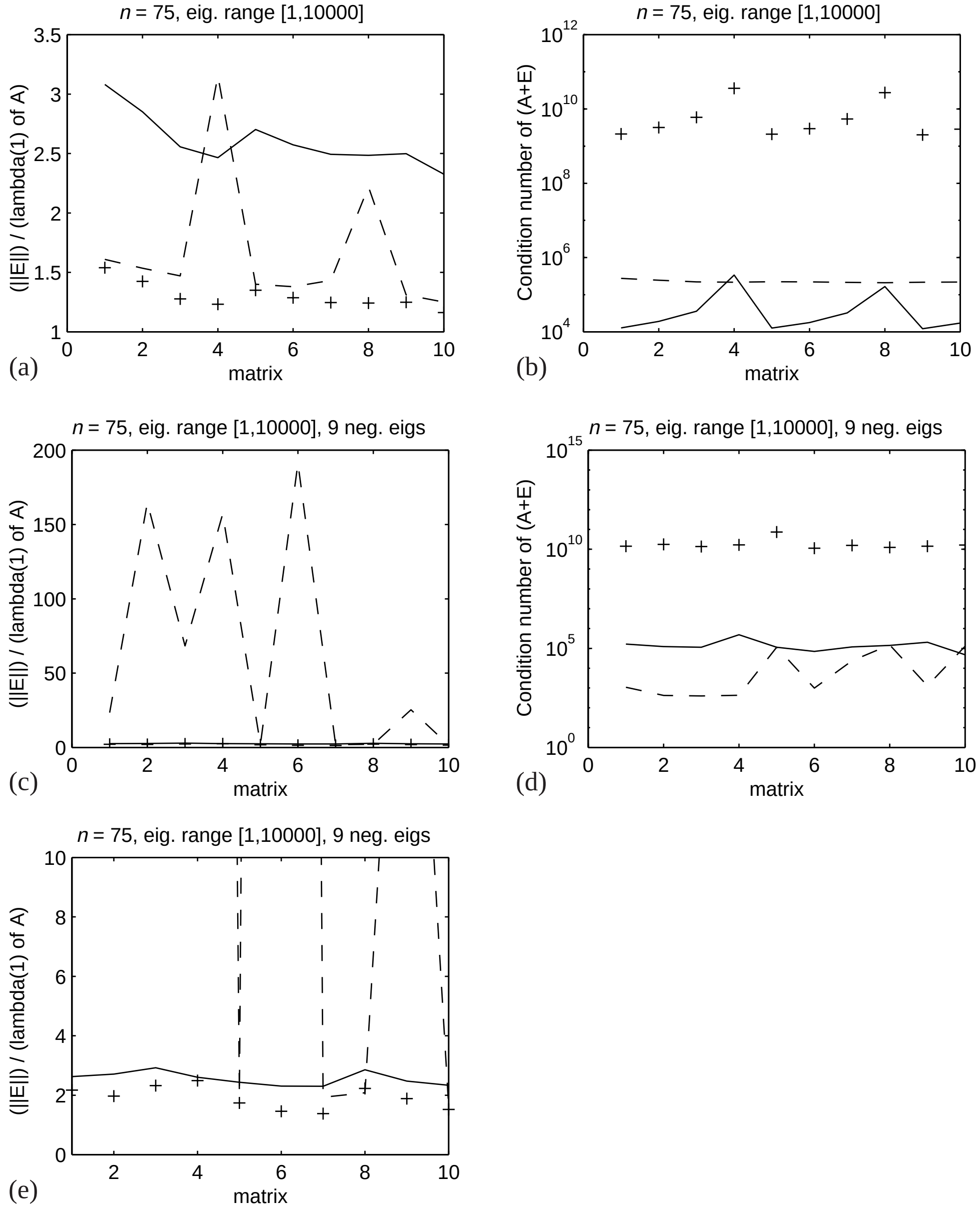


FIG. 6. (a), (b): *Performance of existing and new methods on* 10 *matrices, each containing one negative eigenvalue.* (c)–(e): *Performance of existing and new methods on* 10 *matrices, each containing nine negative eigenvalues. Methods: GMW*81 ———, *SE*90 − − −, *revised SE*90 + + +.

Additionally, it has proved to be useful in ensuring that the Winget factors within element-by-element preconditioners are definite [6], and it has been implemented in a block version of the factorization [5]. Finally, the results of this paper show that the new algorithm may be a useful way to obtain rough estimates of $-\lambda_1(A)$ in cases where this is useful, for example, in some trust region methods. For general optimization applications, both factorizations are likely to continue to be used; the lower a priori bound on $\|E\|$ for GMW81 and the new algorithm may not be a determining factor since the results of [12] and this section continue to show that both algorithms reliably produce values of $\|E\|$ that are far lower than these bounds in practice. If our test problems are a good indication, however, the apparently greater robustness of our new method in not producing poor values of $\|E\|$ or excessively high condition numbers of $A + E$ may be an asset.

Acknowledgments. We thank David Gay, Michael Overton, and Margaret Wright for alerting us to the difficulties of our original modified Cholesky algorithm on their problems from primal-dual methods and for supplying sample test problems. We also thank Margaret Wright for many helpful, detailed suggestions regarding the presentation of this paper.

REFERENCES

[1] C. Ashcraft, R. G. Grimes, and J. G. Lewis, *Accurate symmetric indefinite linear equation solvers*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 513–561.

[2] S. H. Cheng and N. J. Higham, *A modified Cholesky algorithm based on a symmetric indefinite factorization*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 1097–1110.

[3] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Numerical experiments with the LANCELOT package (Release A) for large-scale nonlinear optimization*, Math. Programming, 73 (1996), pp. 73–110.

[4] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *LANCELOT: A Fortran Package for Large-scale Nonlinear Optimization (Release A)*, Springer Ser. Comput. Math. 17, Springer-Verlag, Berlin, 1992.

[5] M. J. Daydé, *A Block Version of the Eskow-Schnabel Modified Cholesky Factorization*, Rapport Technique ENSEEIHT-IRIT RT/APO/95/8, 1995.

[6] M. J. Daydé, J.-Y. L'Excellent, and N. I. M. Gould, *Element-by-element preconditioners for large partially separable optimization problems*, SIAM J. Sci. Comput., 18 (1997), pp. 1767–1787.

[7] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983; reprinted as Classics Appl. Math. 16, SIAM, Philadelphia, PA, 1996.

[8] D. M. Gay, M. L. Overton, and M. H. Wright, *A primal-dual interior point method for nonconvex nonlinear programming*, in Advances in Nonlinear Prgramming, Y. Yuan, ed., Kluwer Academic Publishers, Dordrecht, the Netherlands, 1998, pp. 31–36.

[9] P. E. Gill and W. Murray, *Newton-type methods for unconstrained and linearly constrained optimization*, Math. Programming, 28 (1974), pp. 311–350.

[10] P. E. Gill, W. Murray, and M. H.Wright, *Practical Optimization*, Academic Press, London, 1981.

[11] T. Schlick *Modified Cholesky factorizations for sparse preconditioners*, SIAM J. Sci. Comput., 14, (1993), pp. 424–445.

[12] R. B. Schnabel and E. Eskow, *A new modified Cholesky factorization*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 1136–1158.

SIAM J. Optim.
Vol. 9, No. 4, pp. 1149–1158

PARALLEL ADAPTIVE GMRES IMPLEMENTATIONS FOR HOMOTOPY METHODS[*]

MARIA SOSONKINA[†], DONALD C. S. ALLISON[‡], AND LAYNE T. WATSON[§]

To our good friend John Dennis in celebration of his 60th birthday.

Abstract. The success of probability-one homotopy methods in solving large-scale optimization problems and nonlinear systems of equations on parallel architectures may be significantly enhanced by the accurate parallel solution of large sparse nonsymmetric linear systems. Iterative solution techniques, such as GMRES(k), favor parallel implementations. However, their straightforward parallelization usually leads to a poor parallel performance because of global communication incurred by processors. One variation of GMRES(k) considered here is to adapt the restart value k for any given problem and use Householder reflections in the orthogonalization phase, coupled with graph-based matrix partitioning, to achieve high accuracy and reduce the communication overhead. This particular GMRES implementation is tailored to the uniquely stringent requirements imposed on a linear system solver by probability-one homotopy algorithms: occasionally unusually high accuracy, ability to adapt to problems of widely varying difficulty, and parallelism.

Key words. globally convergent, GMRES method, Krylov subspace methods, nonsymmetric linear systems, probability-one homotopy, sparse matrix

AMS subject classifications. 65F10, 65F50, 65H10

PII. S1052623497329671

1. Introduction.

For numerous highly nonlinear realistic applications, probability-one homotopy methods are a primary solution choice, and are robust and accurate. Other nonlinear analysis methods, such as quasi-Newton methods, often fail whenever a good initial estimate of the solution to a given problem is hard to obtain. On the other hand, probability-one homotopy methods converge to a solution from an *arbitrary* starting point (outside of a set of measure zero). These methods have been successfully applied to solve Brouwer fixed point problems, zero finding problems, polynomial systems of equations, optimization problems, discretizations of nonlinear two-point boundary value problems based on shooting, finite differences, collocation, and finite elements [23], [24]. However, the global convergence together with robustness and accuracy of homotopy methods often results in a rather costly computational process. Sparse linear algebra for homotopy methods was addressed first in [25] and [26] and later in [1], [9], [13], [16], and elsewhere.

There are subtle, but fundamental, differences between continuation, homotopy methods, and probability-one homotopy methods. These differences have a major impact on the philosophy and construction of numerical algorithms. Suffice it to state that these differences are discussed in the literature (e.g., [12]) and that this paper

[*] Received by the editors November 6, 1997; accepted for publication (in revised form) January 28, 1999; published electronically September 24, 1999. The work of the first and third authors was supported in part by Department of Energy grant DE-FG05-88ER25068/A004, National Aeronautics and Space Administration grant NAG-1-1562, Air Force Office of Scientific Research grants F49620-96-1-0089 and F49620-96-1-0104, and National Science Foundation grant DMS-9625968.

http://www.siam.org/journals/siopt/9-4/32967.html

[†] Department of Computer Science, University of Minnesota, 320 Heller Hall, 10 University Drive, Duluth, MN 55812 (masha@d.umn.edu).

[‡] Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 (allison@cs.vt.edu).

[§] Departments of Computer Science and Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 (ltw@cayuga.cs.vt.edu).

concerns what are technically known as *globally convergent probability-one* homotopy algorithms.

Growing computer capability is a major reason for homotopy algorithms becoming an affordable alternative to other less robust methods. The benefits of parallel architectures for homotopy methods have been shown in [2]. Despite the serial nature of curve tracking, the burden of computation in homotopy methods is in the numerical linear algebra and user-defined function evaluations [8], which can be carried out in parallel. For large-scale problems, the cost of the serial work may be negligible in comparison with linear system solving (and user-written function evaluation). As concluded in [3], a high degree of parallelism in the linear system solutions contributes much to the overall parallel performance of homotopy methods.

Besides parallelism, the linear system solver in the inner loop of any curve tracking algorithm has to robustly handle linear systems with widely varying characteristics. Jacobian matrices can have significant qualitative differences along a homotopy zero curve, varying between straight sections of the curve and around sharp turning points, and also varying as the homotopy map moves through different physical regimes (e.g., as a fluid flow changes from inviscid to viscous or as the transistor properties change in a circuit). An iterative linear system solver in the inner loop cannot, even occasionally, fail to converge, as this may cause the entire curve tracking process to abort. This robustness requirement is addressed here by using an *adaptive* variant of GMRES(k) [20].

Another somewhat unique requirement from probability-one homotopy algorithms is the occasional need for very high accuracy (near machine precision for residuals) at some points along the homotopy zero curve γ, at least as compared to the accuracy required in the context of a PDE solution, a quasi-Newton iteration for a nonlinear system, or a nonlinear optimization step. This high accuracy requirement seems unavoidable, is well documented [16], [23], [24], and is absolutely crucial for certain classes of problems like analog DC circuit simulation [16] and postbuckling stability analysis of structures that exhibit snap-back and snap-through phenomena [20]. This accuracy requirement is addressed by using Householder reflections in the orthogonalization phase of GMRES(k). Note that there is no correlation between the step size in arc length along γ, the accuracy with which γ is computed, and the accuracy with which linear systems arising from the curve tracking algorithm are solved. For instance, γ might be computed very accurately with large steps and crude linear system solutions. Alternatively, very small steps and highly accurate linear system solutions might be required for even a crude approximation of γ.

Thus in the context of probability-one homotopy methods, an efficient parallel implementation of a linear system solver has to satisfy the requirements established for a sequential linear solver—robustness and high accuracy in the solution. To meet these requirements, a particular variation of the popular linear system solution tool GMRES [18] is proposed in this paper. This variation [20] uses an adaptive strategy to deal with varying difficulties of linear systems, which are to be solved by a homotopy method, and Householder reflections to achieve high accuracy. It turns out that the use of Householder reflections, coupled with graph-based matrix partitioning, also improves the parallel performance of GMRES(k).

Section 2 gives a description of the adaptive GMRES(k) algorithm. Its parallel implementations using Householder reflections and graph-based matrix partitioning are discussed in section 3, along with numerical results. Section 4 contains conclusions.

2. GMRES(k) algorithm.

The GMRES algorithm [18] is used to solve a linear system $Ax = b$ with an $n \times n$ nonsymmetric invertible coefficient matrix A. Similar to the classical conjugate gradient method, GMRES produces approximate solutions x_j which are characterized by a minimization property over the Krylov subspaces $K(j, A, r_0) \equiv \text{span}\{r_0,\ Ar_0,\ A^2r_0,\ \ldots,\ A^{(j-1)}r_0\}$, where $r_0 = b - Ax_0$ and j is the iteration number. However, unlike the conjugate gradient algorithm, the work and memory required by GMRES grow proportionately to the iteration number since GMRES needs all j vectors to construct an orthonormal basis for $K(j, A, r_0)$. This basis is often called an Arnoldi basis, since it is implemented by the Arnoldi procedure (see, e.g., [17]).

In practice, a restarted version GMRES(k) is used, where the algorithm is restarted every k iterations. GMRES(k) takes x_k as the initial guess for the next cycle of k iterations and continues until the residual norm is small enough. The disadvantage of the restarted version is that it may stagnate and never reach the solution. The essence of the adaptive GMRES strategy proposed in [20] is to adapt the parameter k to the problem, in the same way a variable order ODE algorithm tunes the order k. With modern programming languages, which provide pointers and dynamic memory management, dealing with the variable storage requirements implied by varying k is not difficult.

If k in GMRES(k) is not sufficiently large, GMRES(k) can stagnate. A test of stagnation developed in [7] detects an insufficient residual norm reduction in the restart number k of steps by estimating the GMRES behavior on a particular linear system. Specifically, GMRES(k) is declared to have stagnated and the iteration is aborted if, at the rate of progress over the last restart cycle of steps, the residual norm tolerance cannot be met in some large multiple (*bgv*) of the remaining number of steps allowed (*itmax* is a bound on the number of steps permitted). Slow progress of GMRES(k), which indicates that an increase in the restart value k may be beneficial [21], can be detected with a similar test. The near-stagnation test uses a different, smaller multiple (*smv*) of the remaining allowed number of steps. If near-stagnation occurs, the restart value k is incremented by some value m and the *same* restart cycle continues. Such incrementing is used, whenever needed, if the restart value k is less than some maximum value *kmax*. When the maximum value for k is reached, adaptive GMRES(k) proceeds as GMRES(*kmax*). The values of the parameters *smv*, *bgv*, and m are established experimentally and can remain unchanged for most problems.

The convergence of GMRES may also be seriously affected by roundoff error, which is especially noticeable when a high accuracy solution is required. The orthogonalization phase of GMRES is susceptible to numerical instability. Let Q be a matrix whose columns are obtained by orthogonalizing the columns of a matrix M, and define the error matrix $E = Q^TQ - I$. The error matrices E_{MGS}, E_{HR} using the modified Gram–Schmidt and Householder reflection methods, respectively, to construct Q from M satisfy

$$\|E_{MGS}\|_2 \sim \mathbf{u}\,\text{cond}(M), \qquad \|E_{HR}\|_2 \sim \mathbf{u},$$

where $\mathbf{u}$ is the machine unit roundoff [5]. Clearly the orthogonalization with Householder reflections is more robust. An implementation of GMRES(k) using Householder reflections and its block version are given in [22]. It has been shown theoretically [22] that the implementation of GMRES using Householder reflections is about twice as expensive as when modified Gram–Schmidt is used. However, the Householder reflection method produces a more accurate orthogonalization of the Krylov subspace basis when the basis vectors are nearly linearly dependent and the modified

Gram–Schmidt method fails to orthogonalize the basis vectors; this can result in fewer GMRES iterations compensating for the higher cost per iteration using Householder reflections. Let e_j be the jth standard basis vector and all norms the 2-norm. Pseudocode for an adaptive version of GMRES(k) with orthogonalization via Householder reflections implemented as in [7] and [22] is given in Figure 1. Call this algorithm AGMRES(k).

The rounding error of a sparse matrix-vector multiplication depends only on the nonzero entries in each row of the sparse matrix, so the error tolerance $xtol$ is proportional to the average number of nonzeros per row $avnz =$ (number of nonzeros in $A)/n$. Since GMRES convergence is normally measured by reduction in the initial residual norm, the convergence tolerance is $tol = \max\{\|r_0\|, \|b\|\}xtol$.

A possible symptom of AGMRES(k) going astray is an increase in the residual norm between restarts (the residual norm is computed by direct evaluation at each restart). If the residual norm on the previous restart is actually smaller than the current residual norm, then AGMRES(k) terminates. The solution is considered acceptable if $\|r\| < tol^{2/3}$, although this loss of accuracy may cause the client algorithm (the outer algorithm requiring solutions to linear systems) to work harder or fail. A robust client algorithm can deal gracefully with a loss of accuracy in the linear system solutions. If $\|r\| \geqq tol^{2/3}$, AGMRES(k) is deemed to have failed. In this latter case, the continuation of GMRES(k) would typically result in reaching a limit on the number of iterations allowed and a possible repetition of $\|r^{old}\| < \|r\|$ in later restarts. AGMRES(k) may exceed an iteration limit when it is affected by roundoff errors in the case of a (nearly) singular GMRES least-squares problem. The condition number of the GMRES least-squares problem is monitored by the incremental condition estimate [6] as in [7]. AGMRES(k) aborts when the estimated condition number is greater than $1/(50\mathbf{u})$.

3. Parallel implementations. In parallel environments, the choice of the orthogonalization process for the Krylov subspace basis vectors depends not only on the accuracy of the process but also on the amount and type of global communication it incurs. For some orthogonalization procedures, only one of the two requirements is satisfied. For example, in serial implementations of the GMRES method, the modified version of the Gram–Schmidt process is often used as being sufficiently accurate for a number of problems. However, in parallel GMRES implementations, other orthogonalization procedures are preferable since the modified Gram–Schmidt process exhibits a large communication overhead (see, e.g., [15]). Better efficiency is achieved with the classical Gram–Schmidt, but it is unstable. A compromise between the accuracy and communication overhead resulted in the development of parallel GMRES(k) variations with a non-Arnoldi basis that undergoes orthogonalization only at the end of a restart cycle (see, e.g., [4], [11]). At this point, an equivalent to the Arnoldi basis is recovered as the matrix Q in the QR factorization of the non-Arnoldi basis. The QR factorization is performed in parallel by a point or block version of Householder reflections [19] that have a high degree of parallelism and avoid all-to-all communications. Under the requirement of high accuracy, the errors associated with performing GMRES(k) iterations in a nonorthogonal basis are not acceptable. However, an efficient parallel implementation of Householder reflections can be employed successfully in a parallel version of the GMRES(k) algorithm, in which Householder reflections are used to compute the Arnoldi basis.

```
    choose $x, itmax, kmax, m$;
    $r := b - Ax$;    $itno := 0$;    $cnmax := 1/(50\mathbf{u})$;
    $xtol := \max\{100.0, 1.01 avnz\}\mathbf{u}$;    $tol := \max\{\|r\|, \|b\|\} xtol$;
    while $\|r\| > tol$ do
    begin
        $r^{old} := r$;
        determine $P_1 r = \pm\|r\| e_1$
          where the Householder transformation matrix $P_1$ is defined in [17];
        $k_1 = 1$;    $k_2 = k$;
L1:     for $j := k_1$ step 1 until $k_2$ do
        begin
            $itno := itno + 1$;
            $v := P_j \cdots P_1 A P_1 \cdots P_j e_j$;
            determine $P_{j+1}$ such that $P_{j+1} v$ has zero components
              after the $(j+1)$st;
            update $\|r\|$ as described in [18];
            estimate condition number $\mathrm{cond}(AV_j)$ of GMRES least squares
              problem via the incremental condition number $ICN$ as in [6];
            if $ICN > cnmax$ then abort;
            if $\|r\| \leqq tol$ then goto L2
        end
        $test := k_2 \times \log\left[tol/\|r\|\right] \Big/ \log\left[\|r\|/\left((1.0 + 10\mathbf{u})\,\|r^{old}\|\right)\right]$;
        if $k_2 \leqq kmax - m$ and $test \geqq smv \times (itmax - itno)$ then
            $k_1 := k_2 + 1$;    $k_2 := k_2 + m$;
            goto L1
        end if
L2:     $e_1 := (1, 0, \ldots, 0)^T$;    $k := k_2$;
        solve $\min_y \left\| \|r\| e_1 - \bar{H}_j y \right\|$ for $y_j$ where $\bar{H}_j$ is described in [18];
        $q := \binom{y_j}{0}$;    $x := x + P_1 \ldots P_j q$;    $r := b - Ax$;
        if $\|r\| \leqq tol$ then exit;
        if $\|r^{old}\| < \|r\|$ then
            if $\|r\| < tol^{2/3}$ then
                exit
            else
                abort
            end if
        end if
        $test := k \times \log\left[tol/\|r\|\right] \Big/ \log\left[\|r\|/\left((1.0 + 10\mathbf{u})\,\|r^{old}\|\right)\right]$;
        if $test \geqq bgv \times (itmax - itno)$ then
            abort
        end if
    end
```

FIG. 1. *Adaptive GMRES* (k).

```
if ($proc = 1$) then $s := j$ else $s := 0$ end if
determine $H_{s+1}$ such that $H_{s+1} v_{loc} \equiv w_{loc}$ has zeros
   after the $(s+1)$st component;
if ($proc = 1$) then
      send $w_{loc}(s+1)$ to $right$;
else
      receive $w$ from $left$;
      determine $G_1$ such that $w_{loc}(1) = 0$; update $w$;
      if ($proc \neq p$) send $w$ to $right$;
end if
```

Fig. 2. *Parallel Householder reflection generation.*

Here, an implementation of the Householder reflection orthogonalization in GMRES(k) proposed in [22] is adapted to work in parallel. The parallel version employs an algorithm developed in [19] for generating and applying Householder reflections. This algorithm avoids dot-products and all-to-all communications. The degree of parallelism equal to the number of processors in parallel Householder reflection generation is achieved by letting each processor create its local portion of the Householder reflection vector independently. After that, local Householder reflection vectors are assembled into a global Householder reflection vector using Givens rotations. The approach taken in [19] assumes a *fixed* ring of processors, in which the communication always starts from a fixed (first) processor.

Pseudocode for the algorithm generating Householder reflections (called HG) at the jth GMRES(k) iteration on the processor *proc* (in a ring of p processors) is given in Figure 2. Before applying a Householder reflection, a sequence of Givens rotations has to be applied. Thus, Householder reflection application can be viewed as the Householder reflection generation process reversed with respect to the order of using Givens rotations and Householder reflections.

In Figure 2, H and G denote the Householder transformation matrix and the Givens rotation matrix (as given in [17]), respectively; v_{loc} denotes a portion of the Krylov subspace vector $A^j r_0$ located on a processor; p, $left$, and $right$ are the processors with the highest rank, with the $proc - 1$ rank, and the $proc + 1$ rank, respectively. It is also assumed that the first processor has the jth row of the input matrix.

However, the design presented in Figure 2 admits only a special case of the matrix row distribution: assignment of a block of contiguous rows to each processor (call it block-striped partitioning), which is rarely advantageous for an arbitrary unstructured matrix. For large unstructured matrices graph-theoretical heuristics exist to produce partitioning (call it graph-based) that minimizes the communication to computation ratio of the distributed matrix-vector multiply. In the current implementation, a graph partitioning algorithm from the MeTiS package [14] is used and the parallel version of the matrix-vector multiply is performed as in [15]. The matrix-vector multiplication requires that the components of all vectors be distributed in accordance with the corresponding matrix rows and allows overlapping of computation and communication. To use the algorithms in Figure 2, the redistribution of a vector requires $\mathcal{O}(p^2)$ communications at each GMRES(k) iteration, which is highly impractical and reduces the efficiency gained by the distributed matrix-vector product. Thus, it is beneficial to develop an extension (call it MHG) of the algorithms in Figure 2, which accepts an arbitrary row distribution among processors (Figure 3). The key idea of this extension is to allow a *flexible* ring in performing Givens rotations. The flexible ring is the

```
if ($j = 1$) then
        $s := 1$;
else
        if (proc has ($j - 1$)st row) then $s := s + 1$;
end if
determine $H_s$ such that $H_s v_{loc} \equiv w_{loc}$ has zeros
    after the $s$th component;
if (proc has $j$th row) then
        ring_end := left;
        send $w_{loc}(s+1)$ and ring_end to right;
else
        receive $w$ and ring_end from left;
        determine $G_s$ such that $w_{loc}(s) = 0$; update $w$;
        if (proc $\neq$ ring_end) then send $w$ to right;
end if
```

FIG. 3. *Modified parallel Householder reflection generation.*

ring of processors in which the communication may start from an arbitrary processor that (in a given graph-based partitioning) holds the first component of the vector x. In practice, each processor in the flexible ring determines the current *ring_start* by consulting the global mapping array *mpart*, each entry of which is a processor number indexed by a matrix row number (where the row numbering refers to the original matrix before partitioning) located on that processor.

Usually, the subspace dimension is much smaller than the matrix dimension and the graph partitioning algorithm produces a balanced workload by assigning an almost equal number of rows to each processor. Thus, the case when the index s within v_{loc} becomes equal to the size of a local partition (size of v_{loc}) occurs rarely for large matrices, unless the number of processors is very large.

3.1. Experimental results. To compare the Householder reflections restricted to the block-striped partitioning with the proposed extension for the graph-based partitioning, GMRES was instrumented to collect the timing information relevant only to the parallel Householder reflection generation and application. The time spent in HG and MHG for generating and applying a global Householder reflection vector is independent of the type of partitioning and thus presents a good measure of the relative computational complexity of these two implementations.

The test problem is derived from a commercial circuit design at AT&T Bell Laboratories. Analog circuit simulation requires the calculation of DC operating points, which are solutions of large sparse nonlinear systems of equations. For realistic transistor models these nonlinear systems are fiercely nonlinear, causing the best damped Newton algorithms to frequently fail. Probability-one homotopy methods have been successful on such problems, but the homotopy zero curves are long and have extremely sharp turning points. The import of this is that near machine precision is required for linear system solutions, at least near the turning points. The results in Figure 4 are for a typical Jacobian matrix near a turning point on the homotopy zero curve, extracted from a production run at AT&T.

The specific circuit problem (bgatt) is described in detail in [16]. Here only the nonzero pattern of the corresponding matrix is presented in Figure 4. To get more meaningful CPU times for just one linear system solve, the actual Jacobian matrix

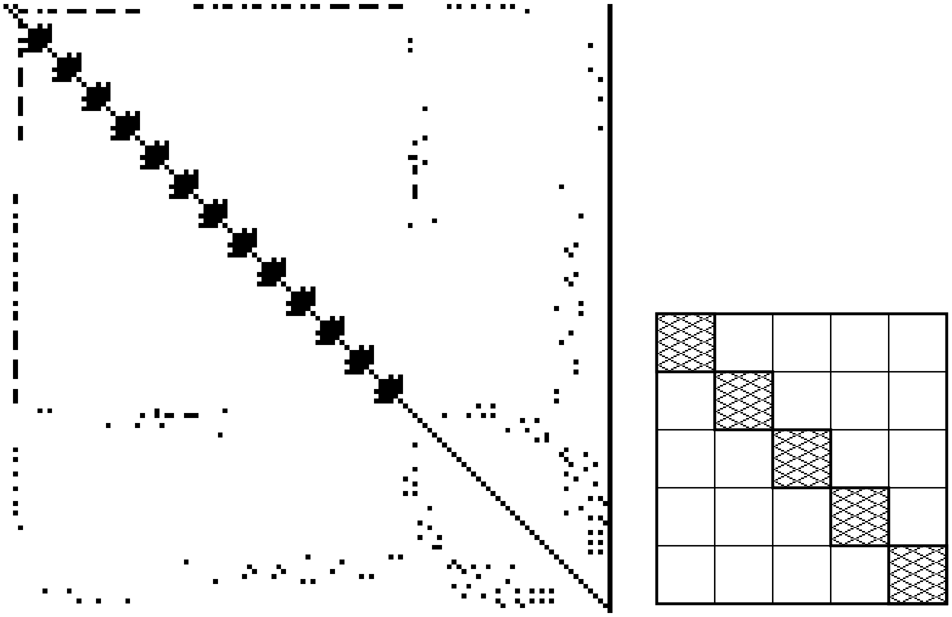


Fig. 4. *Nonzero pattern of the circuit matrix* (*left*), *and construction of a larger matrix* (*right*).

Table 1

Total and matrix-vector multiplication parallel times for GMRES(15) *with MHG and HG orthogonalizations.*

		4	5	15	16	25	32	66	95
MHG	T_p	1.76	1.67	1.54	1.44	1.86	2.09	3.93	5.47
	T^{MV}	0.37	0.39	0.23	0.11	0.20	0.07	0.29	0.35
	np	0	4	6	0	8	0	14	12
	Comm	0	16	46	0	64	0	210	183
HG	T_p	1.81	1.64	1.52	1.52	1.94	2.19	4.13	5.90
	T^{MV}	0.37	0.34	0.17	0.11	0.26	0.07	0.50	0.70
	np	0	2	2	0	2	0	2	2
	Comm	0	8	24	0	48	0	130	188

($n = 125$ with 782 nonzeros) was scaled 96 times; i.e., 96 replicas of the problem were assembled in an $N \times N$ matrix, where $N = 96 \times 125$ as shown in Figure 4 for five replicas.

MHG and HG generated the first Householder vector in 0.26s and 0.25s, respectively, whereas MHG was slightly faster (0.23s and 0.25s, respectively) than HG in applying a Householder reflection. (The times quoted here are the averaged parallel times over three experiments performed on eight processors of an Intel Paragon. The code was instrumented and compiled with the debug option, inflating the times, which should not be compared to the times in Table 1 from optimized code.) Thus, the proposed MHG implementation exhibits performance comparable to HG and outperforms HG in some instances.

For the given test problem, two restart cycles of GMRES(15) were executed to determine the dependence of the overall parallel time on the type of partitioning (and thus on HG and MHG) with an increase in the number of processors. Table 1 shows the averaged results over three experiments for 4, 5, 15, 16, 25, 32, 66, and 95 processors, where T_p denotes the overall parallel time for GMRES(15), T^{MV} denotes the maximum parallel time for matrix-vector multiplication, np is the maximum number of neighboring processors, and $Comm$ is the total number of communication channels among processors. (Note that MeTiS [14] is finding a minimum edge-cut, which minimizes neither the number of neighbors nor the number of communication channels. Apparently a minimum edge-cut does produce better (load balanced) partitions, resulting in better overall parallel times.) For the times observed in three experiments, the largest standard deviation was 0.02. All times are in seconds. The results show that (1) whenever graph-based partitioning produces partitions characterized by fast matrix-vector multiplication, the total parallel time is also small since MHG preserves the benefits of the graph-partitioning (cf. the columns for 25, 32, 66, and 95 processors); (2) whenever the T^{MV} values are the same (columns for 16, 32 processors) or similar the total parallel time with MHG is better due to more balanced partitions resulting from graph partitioning; (3) only when block-striped partitioning is obviously superior (columns for 5 and 15 processors) is the total parallel time with HG smaller than that with MHG; (4) as the number of processors increases, the difference in total times between MHG and HG is largely due to the difference in the matrix-vector multiply times T^{MV}. For 95 processors there is a factor-of-2 difference in T^{MV}. Since for very large problems the overall homotopy zero curve tracking time is dominated by the matrix-vector multiply time in the linear system solver, the superiority of the graph-based MHG method is likely to be even greater with increasing problem size.

4. Conclusions.

Probability-one homotopy algorithms for large (sparse) nonlinear systems of equations impose two atypical requirements on the iterative linear system solver used in the inner curve tracking loop: high accuracy (near machine precision for residuals) and the ability to deal gracefully with problems of widely varying difficulty. The adaptive GMRES(k) based on Householder reflections proposed here meets both these criteria. The proposed parallel implementation of adaptive GMRES(k) takes advantage of an efficient distributed matrix-vector multiplication, and the Householder reflection orthogonalization that avoids all-to-all communications has a high degree of parallelism. The modification proposed here extends the parallel Householder reflection orthogonalization from accepting a block-striped partitioning of rows to a general graph-based partitioning. Using the extended parallel implementation of Householder reflections preserves the advantage gained from applying graph-theoretical heuristics to partition a problem into subdomains. This advantage can be substantial for large problems and high degrees of parallelism and is likely to only improve as both of these factors increase.

REFERENCES

[1] E. L. Allgower, C.-S. Chien, and K. Georg, *Large sparse continuation problems*, J. Comput. Appl. Math., 26 (1989), pp. 3–21.

[2] D. C. S. Allison, A. Chakraborty, and L. T. Watson, *Granularity issues for solving polynomial systems via globally convergent algorithms on a hypercube*, J. Supercomputing, 3 (1989), pp. 5–20.

[3] D. C. S. Allison, K. M. Irani, C. J. Ribbens, and L. T. Watson, *High-dimensional homotopy curve tracking on a shared-memory multiprocessor*, J. Supercomputing, 5 (1992), pp. 347–366.

[4] Z. Bai, D. Hu, and L. Reichel, *A Newton basis GMRES implementation*, IMA J. Numer. Anal., 4 (1994), pp. 563–581.

[5] Å. Björck, *Solving linear least squares problems by Gram-Schmidt orthogonalization*, BIT, 7 (1967), pp. 1–21.

[6] C. H. Bischof and P. T. P. Tang, *Robust Incremental Condition Estimation*, Tech. Report CS-91-133, LAPACK Working Note 33, Computer Science Department, University of Tennessee, Knoxville, TN, 1991.

[7] P. N. Brown and H. F. Walker, *GMRES on* (*nearly*) *singular systems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 37–51.

[8] A. Chakraborty, D. C. S. Allison, C. J. Ribbens, and L. T. Watson, *The parallel complexity of embedding algorithms for the solution of nonlinear equations*, IEEE Trans. Parallel Distrib. Systems, 4 (1993), pp. 458–465.

[9] C.-S. Chien, Z.-L. Weng, and C.-L. Shen, *Lanczos-type methods for continuation problems*, Numer. Linear Algebra Appl., 4 (1997), pp. 23–41.

[10] I. Duff, R. G. Grimes, and J. G. Lewis, *Users' Guide for the Harwell-Boeing Sparse Matrix Collection (Release* I*)*, Tech. Report TR/PA/92/86, CERFACS, France, 1992.

[11] J. Erthel, *A parallel GMRES for general sparse matrices*, Electron. Trans. Numer. Anal., 3 (1995), pp. 160–176.

[12] Y. Ge, L. T. Watson, E. G. Collins, Jr., and D. S. Bernstein, *Probability-one homotopy algorithms for full and reduced order H^2/H^∞ controller synthesis*, Optimal Control Appl. Methods, 17 (1996), pp. 187–208.

[13] K. M. Irani, M. P. Kamat, C. J. Ribbens, H. F. Walker, and L. T. Watson, *Experiments with conjugate gradient algorithms for homotopy curve tracking*, SIAM J. Optim., 1 (1991), pp. 222–251.

[14] G. Karypis and V. Kumar, *MeTiS: Unstructured Graph Partitioning and Sparse Matrix Ordering System*, User's Guide—Version 2.0, Department of Computer Science, University of Minnesota, Minneapolis, MN, 1995.

[15] G.-C. Lo and Y. Saad, *Iterative Solution of General Sparse Linear Systems on Clusters of Workstations*, Tech. Report, UMSI 96/117, Supercomputer Institute, University of Minnesota, Minneapolis, MN, August 1996.

[16] W. D. McQuain, R. C. Melville, C. J. Ribbens, and L. T. Watson, *Preconditioned iterative methods for sparse linear algebra problems arising in circuit simulation*, Comput. Math. Appl., 27 (1994), pp. 25–45.

[17] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, 1996.

[18] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual method for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[19] R. B. Sidje, *Alternatives for parallel Krylov subspace basis computation*, Numer. Linear Algebra Appl., 4 (1997), pp. 305–331.

[20] M. Sosonkina, L. T. Watson, R. K. Kapania, and H. F. Walker, *A new adaptive GMRES algorithm for achieving high accuracy*, Numer. Linear Algebra Appl., 5 (1998), pp. 275–297.

[21] H. A. van der Vorst and C. Vuik, *The superlinear convergence behaviour of GMRES*, J. Comput. Appl. Math., 48 (1993), pp. 327–341.

[22] H. F. Walker, *Implementation of the GMRES method using Householder transformations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 152–163.

[23] L. T. Watson, *Globally convergent homotopy algorithms for nonlinear systems of equations*, Nonlinear Dynam., 1 (1990), pp. 143–191.

[24] L. T. Watson, *Globally convergent homotopy methods: A tutorial*, Appl. Math. Comput., 31 (1989), pp. 369–396.

[25] L. T. Watson, *Numerical linear algebra aspects of globally convergent homotopy methods*, SIAM Rev., 28 (1986), pp. 529–545.

[26] L. T. Watson, S. C. Billups, and A. P. Morgan, *Algorithm* 652: *HOMPACK: A suite of codes for globally convergent homotopy algorithms*, ACM Trans. Math. Software, 13 (1987), pp. 281–310.

SIAM J. Optim.
Vol. 9, No. 4, pp. 1159–1191

MODIFIED CHOLESKY FACTORIZATIONS IN INTERIOR-POINT ALGORITHMS FOR LINEAR PROGRAMMING*

STEPHEN J. WRIGHT†

To John Dennis, with appreciation, on the occasion of his 60th birthday.

Abstract. We investigate a modified Cholesky algorithm typical of those used in most interior-point codes for linear programming. Cholesky-based interior-point codes are popular for three reasons: their implementation requires only minimal changes to standard sparse Cholesky algorithms (allowing us to take full advantage of software written by specialists in that area); they tend to be more efficient than competing approaches that use alternative factorizations; and they perform robustly on most practical problems, yielding good interior-point steps even when the coefficient matrix of the main linear system to be solved for the step components is ill conditioned. We investigate this surprisingly robust performance by using analytical tools from matrix perturbation theory and error analysis, illustrating our results with computational experiments. Finally, we point out the potential limitations of this approach.

Key words. interior-point algorithms and software, Cholesky factorization, matrix perturbations, error analysis

AMS subject classifications. 65F05, 65G05, 90C05

PII. S1052623496304712

1. Introduction.

Most interior-point codes for linear programming share a common feature: their major computational operation at each iteration—solution of a large system of linear equations with a symmetric positive definite coefficient matrix—is performed by a direct sparse Cholesky algorithm. In this algorithm, row and column orderings are determined a priori by well-known heuristics (minimum degree, minimum local fill, nested dissection) that are based solely on the sparsity pattern and not on the numerical values of the nonzero elements. The ordering phase is followed by a symbolic factorization phase in which the nonzero structure of the Cholesky factor is determined and storage is allocated. Finally, a numerical factorization phase fills in the numerical values of the lower triangular Cholesky factor. In interior-point codes, the first two phases usually are performed just once, during either the first interior-point iteration or computation of a starting point.

In the interior-point context, the unadorned Cholesky algorithm can run into difficulties because of extreme ill conditioning. Some diagonal pivots encountered during the numerical factorization phase can be zero or negative, causing the standard Cholesky procedure to break down. Instead of crashing, most codes modify the Cholesky procedure so that it skips the unacceptable pivots or replaces them with workable values. For instance, the offending pivot element is sometimes replaced by a huge number, as in LIPSOL [20] and PCx [3]. In other codes, such as IPMOS [19], the pivot is replaced by a moderate number, but the corresponding right-hand-side

*Received by the editors June 4, 1996; accepted for publication (in revised form) November 14, 1998; published electronically September 24, 1999. This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under contract W-31-109-Eng-38. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/siopt/9-4/30471.html

†Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439 (wright@mcs.anl.gov).

element is set to zero, as are the off-diagonal elements in the corresponding column of the Cholesky factor. The net effects of these approaches, and the approaches used in other Cholesky-based codes, such as OB1 [9], HOPDM [6], and the APOS code of XPRESS-MP [1], are all quite similar to those of the algorithm **modchol** that we analyze in this paper: each small or negative pivot causes the Cholesky procedure to skip one stage, and the solution component corresponding to this pivot is set to zero (or to a very small number). To date, there has been little investigation of these pivot-skipping strategies from a numerical analysis viewpoint.

In the context of Cholesky factorization of general symmetric positive semidefinite matrices, Lawson and Hanson [8, p. 125] advocated the use of pivot skipping when negative pivots are encountered. They also suggested the alternative remedy of diagonal pivoting, in which a "large" diagonal element is selected from the unreduced portion of the matrix at each stage and moved to the pivot position by a symmetric row and column exchange. The procedure terminates when none of the remaining diagonal elements is large enough to qualify as a pivot, and an approximate solution is computed with the partial factors. Higham [7, Chapter 10] presented an error analysis of this approach, and M. H. Wright [15] considered its use in factoring the Hessian matrices that arise in the Newton/logarithmic-barrier method for nonlinear programming. This strategy is not practical in the context of interior-point linear programming codes because the matrices in question are too large to allow row and column exchanges to be performed efficiently. On the other hand, pivot-skipping strategies have the advantage that they can be implemented by changing just a few lines of a general sparse Cholesky code, so it is possible to take advantage of the long-term development effort that has gone into designing such codes and their underlying algorithms. (The recent codes LIPSOL [20] and PCx [3] make explicit use of the sparse Cholesky code of Ng and Peyton [10].) Moreover, the good practical performance of pivot-skipping strategies made the search for alternatives less urgent.

In this paper, we investigate the good performance of pivot-skipping strategies on the majority of practical problems. In section 3, we specify our representative pivot-skipping strategy, which we term **modchol** for convenience, and analyze the effects of the skipped pivots on the computed triangular factor and computed solution. In section 4, we incorporate the effects of finite-precision arithmetic into the analysis. Both sections are general in that they apply to general symmetric positive semidefinite matrices, not just the specific matrices that arise in the interior-point application. In section 5, however, we apply the results of sections 3 and 4 to the equations for calculating the interior-point step, showing how the errors in the computed steps affect the progress of the interior-point algorithm, suggesting a suitable termination criterion, and indicating possible shortcomings in the pivot-skipping approach. Our analysis in this section applies to primal- and dual-degenerate linear programs. We conclude with some computational results in section 6.

A number of other theoretical papers on linear algebra operations in barrier and interior-point methods have appeared in recent years. We mentioned above the paper of M. H. Wright [15], in which a Cholesky procedure with diagonal pivoting was used as the basis of an algorithm to construct steps that are accurate both in the subspace spanned by the active constraint Jacobian and its complement. Our focus in the current paper is on (possibly degenerate) linear programs rather than nondegenerate nonlinear programs. Moreover, we do not allow diagonal pivoting and, since our problem is a linear program, the issue of resolving the component of the step in the near-null space of the active constraint matrix is not as relevant.

In an earlier paper [18], S. J. Wright considered the stability of algorithms for the symmetric indefinite formulation of the step equations at each iteration of an interior-point method for linear programming. Ill conditioning of the coefficient matrix is the key issue in this formulation as well, but we showed that, in general, the calculated steps are good search directions for the interior-point method. Forsgren, Gill, and Shinnerl [5] perform a similar analysis in the context of logarithmic barrier methods for nonlinear problems, but they assume a certain ordering of the rows and columns of the coefficient matrix.

Notation. We summarize here the notation used in the remainder of the paper.

The ith singular value of a matrix B is denoted by $\sigma_i(B)$. We use σ_i alone to denote the ith singular value of the exact Cholesky factor L in section 3.

For any matrix M and index sets $\mathcal{I}$ and $\mathcal{K}$, $M_{\mathcal{I}\mathcal{K}}$ denotes the submatrix formed by the elements M_{ij} for $i \in \mathcal{I}$ and $j \in \mathcal{K}$. The jth column of M is denoted by $M_{\cdot j}$, the column submatrix consisting of columns $j \in \mathcal{K}$ is denoted by $M_{\cdot\mathcal{K}}$, and the submatrix of elements M_{ij} for $j \in \mathcal{K}$ is noted by $M_{i,\mathcal{K}}$. The submatrix consisting of rows and columns i through j is denoted by $M_{i:j,i:j}$.

Unit roundoff error, which we denote by $\mathbf{u}$, can be defined implicitly by the following statement (see, for example, Higham [7]). When α and ζ are any two floating-point numbers; op denotes $+$, $-$, $\times$, and $/$; and $\mathrm{fl}(\cdot)$ denotes the floating-point representation of a real number, we have

$$\mathrm{fl}(\alpha \,\mathrm{op}\, \zeta) = (\alpha \,\mathrm{op}\, \zeta)(1+\delta) \quad \text{for some } \delta \text{ satisfying } |\delta| \le \mathbf{u}.$$

We use $\mathrm{comp}(\cdot)$ to denote the calculated version of the quantity in question, taking into account the effects of roundoff error.

In estimating the sizes of various quantities that arise in the analysis, we use δ_1 to denote a constant whose magnitude depends at most cubically on the dimension m of the linear system. We often use $\delta_{\mathbf{u}}$ as a shorthand for $\delta_1\mathbf{u}$. Order notation $O(\cdot)$ and $\Theta(\cdot)$ is used as follows: if v (vector or scalar) and ϵ (nonnegative scalar) are two quantities that share a dependence on other variables, we write $v = O(\epsilon)$ if there is a moderate constant β_1 such that $\|v\| \le \beta_1\epsilon$ for all values of ϵ that are either sufficiently close to zero or sufficiently large, depending on the context. We write $v = \Theta(\epsilon)$ if there are constants β_1 and β_0 such that $\beta_0\epsilon \le \|v\| \le \beta_1\epsilon$ for ϵ in the ranges specified above.

The notation $\|\cdot\|$ denotes the Euclidean vector norm $\|\cdot\|_2$ and also its induced matrix norm, unless otherwise noted. For any matrix A, the matrix consisting of the absolute values of each element is denoted by $|A|$. We use $\mathbf{1}$ to denote the vector $(1,1,\ldots,1)^T$.

Finally, we mention the parameter ϵ that defines the pivot threshold in the modified Cholesky algorithm. A scaled quantity $\bar{\epsilon}$ defined by

$$\bar{\epsilon} \overset{\text{def}}{=} 2m^2\epsilon \tag{1.1}$$

appears frequently in the analysis, because the incorporation of the scaling term $2m^2$ saves some clutter.

2. Primal-dual algorithms for linear programming.

We consider the linear programming problem in standard form:

$$\min\, c^Tx \quad \text{subject to} \quad Ax = b, \qquad x \ge 0, \tag{2.1}$$

where $x \in \mathsf{R}^n$, $c \in \mathsf{R}^n$, $A \in \mathsf{R}^{m\times n}$, and $b \in \mathsf{R}^m$. The dual of (2.1) is

$$\max\, b^T\pi \quad \text{subject to} \quad A^T\pi + s = c, \qquad s \ge 0, \tag{2.2}$$

where $s \in \mathsf{R}^n$ and $\pi \in \mathsf{R}^m$. We assume throughout the paper that A has full row rank (which can be guaranteed by preprocessing the data), so that $m \le n$. The Karush–Kuhn–Tucker (KKT) conditions, which identify a vector triple (x, π, s) as a primal-dual solution for (2.1), (2.2), can be stated as follows:

$$A^T\pi + s = c, \tag{2.3a}$$
$$Ax = b, \tag{2.3b}$$
$$x_i s_i = 0, \qquad i = 1, 2, \ldots, n, \tag{2.3c}$$
$$(x, s) \ge 0. \tag{2.3d}$$

We assume throughout the paper that a primal-dual solution exists, but we make no assumptions about uniqueness or nondegeneracy. It is well known that the index set $\{1, 2, \ldots, n\}$ can be partitioned into two sets $\mathcal{B}$ and $\mathcal{N}$ such that for all primal-dual solutions (x^*, π^*, s^*) we have

$$x_i^* = 0 \quad \text{for all} \quad i \in \mathcal{N}, \qquad s_i^* = 0 \quad \text{for all} \quad i \in \mathcal{B}. \tag{2.4}$$

Primal-dual interior-point algorithms generate a sequence of iterates (x, π, s) that satisfy the strict inequality $(x, s) > 0$. They find search directions by applying a modification of Newton's method to the system of nonlinear equations that is equivalent to the first three KKT conditions (2.3a), (2.3b), (2.3c), namely,

$$Ax - b = 0, \qquad A^T\pi + s - c = 0, \qquad XS\mathbf{1} = 0, \tag{2.5}$$

where $X = \operatorname{diag}(x_1, x_2, \ldots, x_n)$ and $S = \operatorname{diag}(s_1, s_2, \ldots, s_n)$. In general, the search direction $(\Delta x, \Delta\pi, \Delta s)$ satisfies the following linear system:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\pi \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -r_{xs} \end{bmatrix}, \tag{2.6}$$

where the coefficient matrix is the Jacobian of (2.5) and the right-hand-side components r_b and r_c are defined by

$$r_b = Ax - b, \qquad r_c = A^T\pi + s - c. \tag{2.7}$$

In a pure Newton (affine-scaling) method, the remaining right-hand-side component r_{xs} is defined by

$$r_{xs} = XS\mathbf{1}, \tag{2.8}$$

and, in this case, we denote the solution of (2.6) by $(\Delta x^{\text{aff}}, \Delta\pi^{\text{aff}}, \Delta s^{\text{aff}})$. In a path-following method, we have

$$r_{xs} = XS\mathbf{1} - \zeta\mu\mathbf{1}, \tag{2.9}$$

where μ is the duality gap defined by

$$\mu = x^T s/n, \tag{2.10}$$

and $\zeta \in [0, 1]$ is a *centering parameter*. In the "Mehrotra predictor-corrector" algorithm, which is used as the basis of many practical codes, the search direction is calculated by setting

$$r_{xs} = XS\mathbf{1} + \Delta X^{\text{aff}}\Delta S^{\text{aff}}\mathbf{1} - \zeta\mu\mathbf{1}, \tag{2.11}$$

where ΔX^{aff} and ΔS^{aff} are the diagonal matrices formed from the affine-scaling step components Δx^{aff} and Δs^{aff}, and the value of ζ is determined by a heuristic based on the effectiveness of the affine-scaling direction. Mehrotra's method requires the solution of *two* linear systems at each iteration—the affine-scaling system (2.6), (2.7), (2.8), and the search direction system (2.6), (2.7), (2.11)—though the coefficient matrix is the same for both systems. Gondzio's [6] higher-order corrector method refines the step by solving additional linear systems, all with the same coefficient matrix as in (2.6).

Once a search direction has been determined, the primal-dual algorithm takes a step of the form

$$(x, \pi, s) + \alpha(\Delta x, \Delta\pi, \Delta s),$$

where α is chosen to maintain strict positivity of the x and s components; that is,

$$(x, s) + \alpha(\Delta x, \Delta s) > 0. \tag{2.12}$$

In most codes, α is chosen to be some fraction of the step-to-boundary $\alpha_{\max}$ defined as

$$\alpha_{\max} = \sup_{\alpha\in[0,1]} \{\alpha \,|\, (x, s) + \alpha(\Delta x, \Delta s) \geq 0\}. \tag{2.13}$$

A typical strategy is to set

$$\alpha = \eta\alpha_{\max},$$

where $\eta \in [.9, 1.0)$ approaches 1 as the iterates approach the solution set.

By applying block elimination to (2.6) and using the notation

$$D^2 = S^{-1}X, \tag{2.14}$$

we obtain the following equivalent system:

$$AD^2A^T\Delta\pi = -r_b - AD^2(r_c - X^{-1}r_{xs}), \tag{2.15a}$$

$$\Delta s = -r_c - A^T\Delta\pi, \tag{2.15b}$$

$$\Delta x = -S^{-1}(r_{xs} + X\Delta s). \tag{2.15c}$$

In many codes, the solution is obtained from just this formulation. A sparse Cholesky factorization, modified to handle small or negative pivots, is applied to the symmetric positive definite coefficient matrix AD^2A^T in (2.15a) and the solution $\Delta\pi$ is obtained by triangular substitution with the computed factor. The remaining direction components are recovered from (2.15b) and (2.15c). Computational experience shows that this technique yields steps that are useful search directions for the interior-point algorithm, even when AD^2A^T is ill conditioned and when the computed version of $\Delta\pi$ has few digits in common with the exact version. This observation is somewhat surprising, since a naive application of error analysis results would suggest that the combination of ill conditioning and roundoff would corrupt the direction hopelessly.

In section 5, we investigate this phenomenon by applying the error analysis developed in sections 3 and 4 to the solution of the system (2.15), assuming that our algorithm **modchol** is used to solve (2.15a) and that all computations are performed in finite-precision floating-point arithmetic. We examine the effects of the errors in the computed step on properties such as the value of $\alpha_{\max}$ (2.13) and on the updated values of the residuals r_b and r_c—properties that indicate whether the step is a useful one for the interior-point method.

We start by specifying **modchol** and analyzing its properties as they pertain to a general linear system $Mz = r$, where M is symmetric positive definite.

3. A modified Cholesky algorithm.

In this section, we describe and analyze **modchol**, a modified Cholesky algorithm designed to handle ill-conditioned matrices for which small or negative pivots may arise during the factorization.

Algorithm **modchol** accepts an $m \times m$ symmetric positive definite matrix M as input, together with a small positive user-defined parameter ϵ, which defines a threshold of acceptability for the pivot elements. If a candidate pivot element is smaller than this threshold, the algorithm simply skips a step of factorization. The output of **modchol** is an approximate lower triangular factor $\tilde{L}$ and an index set $\mathcal{J} \subset \{1, 2, \ldots, m\}$ containing the indices of the skipped pivots. In the following specification, we use $M^{(i)}$ to denote the unfactored part of M that remains after i steps of the algorithm.

ALGORITHM **modchol**.

```
Given $\epsilon$ with $0 < \epsilon \ll 1$;
Set  $M^{(0)} \leftarrow M$; $\tilde{L} \leftarrow 0$; $\mathcal{J} \leftarrow \emptyset$; $\beta = \max_{i=1,2,\ldots,m} M_{ii}$;
for  $i = 1, 2, \ldots, m$
     if   $M_{ii}^{(i-1)} \le \beta\epsilon$
          (* skip this elimination step *)
          Set $\mathcal{J} \leftarrow \mathcal{J} \cup \{i\}$ and
```

$$E^{(i)} = \left[\begin{array}{c|cccc} 0 & 0 & \cdots & \cdots & 0 \\ \hline 0 & M_{ii}^{(i-1)} & \cdots & \cdots & M_{im}^{(i-1)} \\ \vdots & \vdots & 0 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & M_{mi}^{(i-1)} & 0 & \cdots & 0 \end{array}\right], \qquad M^{(i)} = M^{(i-1)} - E^{(i)}; \tag{3.1}$$

```
     else
          (* perform the usual Cholesky elimination step *)
          $\tilde{L}_{ii} \leftarrow \sqrt{M_{ii}^{(i-1)}}$; $M^{(i)} \leftarrow 0$
          for  $j = i+1, i+2, \ldots, m,$
               $\tilde{L}_{ji} = M_{ij}^{(i-1)} / \tilde{L}_{ii}$ ;
          for  $j = i+1, i+2, \ldots, m$
               for  $k = i+1, i+2, \ldots, m,$
                    $M_{jk}^{(i)} \leftarrow M_{jk}^{(i-1)} - \tilde{L}_{ji}\tilde{L}_{ki}$.
```

The ith column of $\tilde{L}$ is zero for each $i \in \mathcal{J}$; that is, $\tilde{L}_{\cdot\mathcal{J}} = 0$. If we denote

$$E = \sum_{i \in \mathcal{J}} E^{(i)} \tag{3.2}$$

and denote the complement of $\mathcal{J}$ in $\{1, 2, \ldots, m\}$ by $\bar{\mathcal{J}}$, it follows from (3.1) that

$$E_{\bar{\mathcal{J}}\bar{\mathcal{J}}} = 0. \tag{3.3}$$

That is, the row or column index of each nonzero element in E must lie in $\mathcal{J}$. It follows from the algorithm that $\tilde{L}$ is the exact Cholesky factor of the perturbed matrix $M-E$, which we denote for convenience by $\tilde{M}$. That is, we have

$$\tilde{L}\tilde{L}^T = \tilde{M} = M - E. \tag{3.4}$$

By partitioning this equation into its $\mathcal{J}$ and $\bar{\mathcal{J}}$ components and using $\tilde{L}_{\cdot\mathcal{J}} = 0$ and (3.3), we obtain

$$M_{\bar{\mathcal{J}}\bar{\mathcal{J}}} = \tilde{L}_{\bar{\mathcal{J}}\cdot}\tilde{L}_{\bar{\mathcal{J}}\cdot}^T + E_{\bar{\mathcal{J}}\bar{\mathcal{J}}} = \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^T, \tag{3.5a}$$

$$M_{\bar{\mathcal{J}}\mathcal{J}} = \tilde{L}_{\bar{\mathcal{J}}\cdot}\tilde{L}_{\mathcal{J}\cdot}^T + E_{\bar{\mathcal{J}}\mathcal{J}} = \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{L}_{\mathcal{J}\bar{\mathcal{J}}}^T + E_{\bar{\mathcal{J}}\mathcal{J}}. \tag{3.5b}$$

The *exact* Cholesky factor L (whose existence is guaranteed by the assumed positive definiteness of M) satisfies

$$LL^T = M. \tag{3.6}$$

Given the linear system

$$Mz = r, \tag{3.7}$$

where M is the matrix factored by **modchol**, the exact solution obviously satisfies

$$z = M^{-1}r = L^{-T}L^{-1}r. \tag{3.8}$$

The approximate solution $\tilde{z}$ is chosen so that the partial vector $\tilde{z}_{\bar{\mathcal{J}}}$ solves the reduced system $M_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{z}_{\bar{\mathcal{J}}} = r_{\bar{\mathcal{J}}}$, while the complementary subvector $\tilde{z}_{\mathcal{J}}$ is set to zero. From (3.5a), we see that $\tilde{z}_{\bar{\mathcal{J}}}$ can be calculated by performing a pair of triangular substitutions; that is,

$$\tilde{z}_{\bar{\mathcal{J}}} = \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-T}\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-1}r_{\bar{\mathcal{J}}}, \qquad \tilde{z}_{\mathcal{J}} = 0. \tag{3.9}$$

Note that $z = \tilde{z}$ when $\mathcal{J} = \emptyset$. When $\mathcal{J} \neq 0$, on the other hand, the difference between $\tilde{z}$ and z can be large in a relative sense. We have

$$\|z - \tilde{z}\| = \left\| \begin{bmatrix} z_{\mathcal{J}} - 0 \\ z_{\bar{\mathcal{J}}} - \tilde{z}_{\bar{\mathcal{J}}} \end{bmatrix} \right\| \geq \|z_{\mathcal{J}}\|,$$

and there is no reason to expect $z_{\mathcal{J}}$ to be small with respect to the full vector z. However, in the main result of this section (Theorem 3.4), we show that the difference between $\tilde{L}^T z$ and $\tilde{L}^T \tilde{z}$ is small. As we see in section 5, this difference determines the usefulness of the computed solution of (2.15) as a search direction for the interior-point algorithm.

To simplify the analysis, we assume throughout the paper that

$$\beta = 1. \tag{3.10}$$

A trivial scaling, which affects neither the algorithm nor its analysis, can always be applied to the symmetric positive definite matrix M to yield (3.10).

We start with a simple result about the intermediate matrices $M^{(i)}$ that arise during **modchol**.

LEMMA 3.1. *If* (3.10) *holds, then the submatrix formed by the last $m-i$ rows and columns of $M^{(i)}$ is symmetric positive definite for all $i = 0, 1, \ldots, m-1$. Moreover, the diagonal elements of all these submatrices are bounded by* 1.

Proof. This observation follows by a simple inductive argument. By assumption, the starting matrix $M^{(0)} = M$ is positive definite. Suppose that the desired property holds for $M^{(i-1)}$. If $i \in \mathcal{J}$, then the lower right $(m-i) \times (m-i)$ submatrix of $M^{(i)}$ is identical to the same submatrix of $M^{(i-1)}$, which is positive definite by assumption. Otherwise, if $i \notin \mathcal{J}$, then $M^{(i)}$ is obtained by applying one step of Cholesky reduction to $M^{(i-1)}$, so its lower right $(m-i) \times (m-i)$ submatrix is positive definite in this case too.

The second claim follows immediately from the fact that $M_{ii} \le \beta = 1$, $i = 1, 2, \dots, m$, and the fact that the diagonal elements cannot increase during the execution of **modchol**. □

The next result bounds the remainder matrix E.

LEMMA 3.2. *Assume that* (3.10) *holds. We then have that*

$$\|E\|_2 \le \|E\|_F \le \bar{\epsilon}^{1/2}, \tag{3.11}$$

where $\bar{\epsilon} = 2m^2\epsilon$.

Proof. From Lemma 3.1, we have $(M_{i,l}^{(i-1)})^2 \le M_{i,i}^{(i-1)} M_{l,l}^{(i-1)}$ for each $l = i+1, \dots, m$. Suppose $i \in \mathcal{J}$, so that $M_{i,i}^{(i-1)} \le \epsilon$. Since the diagonals of each submatrix $M^{(i-1)}$ are bounded by 1, we have $M_{l,l}^{(i-1)} \le 1$ and therefore

$$\left| M_{i,l}^{(i-1)} \right| \le \left(M_{i,i}^{(i-1)} M_{l,l}^{(i-1)} \right)^{1/2} \le \epsilon^{1/2}, \qquad l = i+1, \dots, m.$$

Hence, we have

$$\|E^{(i)}\|_2^2 \le \|E^{(i)}\|_F^2 \le (M_{i,i}^{(i-1)})^2 + 2 \sum_{l=i+1}^{m} (M_{i,l}^{(i-1)})^2 \le \epsilon^2 + 2(m-i)\epsilon \le 2m\epsilon.$$

By using (3.2) and the fact that the nonzero elements of each $E^{(i)}$ occur in different locations, we have

$$\|E\|_F^2 = \sum_{i \in \mathcal{J}} \|E^{(i)}\|_F^2 \le |\mathcal{J}| 2m\epsilon \le 2m^2\epsilon,$$

thereby proving (3.11). □

The bound (3.11) suggests that the matrix $\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-1} E_{\bar{\mathcal{J}}\mathcal{J}}$, which proves to be critical in our analysis, can be estimated by

$$\|\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-1} E_{\bar{\mathcal{J}}\mathcal{J}}\| \le \|\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-1}\| \|E_{\bar{\mathcal{J}}\mathcal{J}}\| \le \bar{\epsilon}^{1/2} \|\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-1}\|.$$

The following theorem shows that in fact the factor $\|\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-1}\|$ can be omitted from the right-hand side. The resulting bound is much stronger, because the omitted factor is potentially quite large.

THEOREM 3.3. *Assume that* (3.10) *holds. We then have*

$$\|\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-1} E_{\bar{\mathcal{J}}\mathcal{J}}\| \le (m\epsilon)^{1/2}. \tag{3.12}$$

Proof. We start by choosing some arbitrary index $i \in \mathcal{J}$ and examining the structure of $E_{\cdot i}$. We note from (3.1) and (3.2) that

- $E_{ji} \neq 0$ for $j < i$ only if $j \in \mathcal{J}$;
- $E_{ii} = M_{ii}^{(i-1)}$;
- $E_{ji} = M_{ji}^{(i-1)} \neq 0$ in general for all $j > i$.

Therefore, we observe that the subvector

$$E_{\bar{\mathcal{J}},i} = [E_{ji}]_{j \in \bar{\mathcal{J}}}$$

has nonzeros only in locations indexed by j with $j > i$. If we define the index subsets $\bar{\mathcal{J}}_i$ and $\mathcal{J}_i$ by

$$\bar{\mathcal{J}}_i \overset{\text{def}}{=} \bar{\mathcal{J}} \cap \{i+1, i+2, \ldots, m\}, \qquad \mathcal{J}_i \overset{\text{def}}{=} \mathcal{J} \cap \{i+1, i+2, \ldots, m\}, \tag{3.13}$$

it follows that

$$E_{\bar{\mathcal{J}},i} = \begin{bmatrix} 0 \\ E_{\bar{\mathcal{J}}_i,i} \end{bmatrix}. \tag{3.14}$$

It follows from this property and the lower triangularity of $\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$ that

$$\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-1} E_{\bar{\mathcal{J}}i} = \begin{bmatrix} 0 \\ \tilde{L}_{\bar{\mathcal{J}}_i\bar{\mathcal{J}}_i}^{-1} E_{\bar{\mathcal{J}}_i i} \end{bmatrix}. \tag{3.15}$$

From Lemma 3.1, we have that $M_{i:m,i:m}^{(i-1)}$ is symmetric positive definite. We perform symmetric permutations on this matrix to group the components in $\mathcal{J}_i$ and $\bar{\mathcal{J}}_i$, and obtain

$$\begin{bmatrix} M_{ii}^{(i-1)} & M_{i,\bar{\mathcal{J}}_i}^{(i-1)} & M_{i,\mathcal{J}_i}^{(i-1)} \\ M_{\bar{\mathcal{J}}_i,i}^{(i-1)} & M_{\bar{\mathcal{J}}_i,\bar{\mathcal{J}}_i}^{(i-1)} & M_{\bar{\mathcal{J}}_i,\mathcal{J}_i}^{(i-1)} \\ M_{\mathcal{J}_i,i}^{(i-1)} & M_{\mathcal{J}_i,\bar{\mathcal{J}}_i}^{(i-1)} & M_{\mathcal{J}_i,\mathcal{J}_i}^{(i-1)} \end{bmatrix} = \begin{bmatrix} M_{ii}^{(i-1)} & E_{\bar{\mathcal{J}}_i,i}^T & E_{\mathcal{J}_i,i}^T \\ E_{\bar{\mathcal{J}}_i,i} & M_{\bar{\mathcal{J}}_i,\bar{\mathcal{J}}_i}^{(i-1)} & M_{\bar{\mathcal{J}}_i,\mathcal{J}_i}^{(i-1)} \\ E_{\mathcal{J}_i,i} & M_{\mathcal{J}_i,\bar{\mathcal{J}}_i}^{(i-1)} & M_{\mathcal{J}_i,\mathcal{J}_i}^{(i-1)} \end{bmatrix}, \tag{3.16}$$

which is still symmetric positive definite. The principal submatrix

$$\begin{bmatrix} M_{ii}^{(i-1)} & E_{\bar{\mathcal{J}}_i,i}^T \\ E_{\bar{\mathcal{J}}_i,i} & M_{\bar{\mathcal{J}}_i,\bar{\mathcal{J}}_i}^{(i-1)} \end{bmatrix} \tag{3.17}$$

is also symmetric positive definite. It is easy to see that steps $i+1, i+2, \ldots, m$ of **modchol** yield a modified Cholesky factorization of the form

$$M_{i+1:m,i+1:m}^{(i-1)} = \tilde{L}_{i+1:m,i+1:m} \tilde{L}_{i+1:m,i+1:m}^T + E_{i+1:m,i+1:m}.$$

As in (3.5a), we have that $E_{\bar{\mathcal{J}}_i,\bar{\mathcal{J}}_i} = 0$, so that by reordering and partitioning as in (3.16) and using $\tilde{L}_{\bar{\mathcal{J}}_i,\mathcal{J}_i} = 0$, we obtain

$$M_{\bar{\mathcal{J}}_i,\bar{\mathcal{J}}_i}^{(i-1)} = \tilde{L}_{\bar{\mathcal{J}}_i,\bar{\mathcal{J}}_i} \tilde{L}_{\bar{\mathcal{J}}_i,\bar{\mathcal{J}}_i}^T. \tag{3.18}$$

By the positive definite property of the matrix in (3.17), the Schur complement of $M_{ii}^{(i-1)}$ in this matrix must be positive, so we have from (3.18) that

$$0 < M_{ii}^{(i-1)} - E_{\bar{\mathcal{J}}_i,i}^T (M_{\bar{\mathcal{J}}_i,\bar{\mathcal{J}}_i}^{(i-1)})^{-1} E_{\bar{\mathcal{J}}_i,i} = M_{ii}^{(i-1)} - \|\tilde{L}_{\bar{\mathcal{J}}_i,\bar{\mathcal{J}}_i}^{-1} E_{\bar{\mathcal{J}}_i,i}\|^2.$$

Because $i \in \mathcal{J}$, we have $M_{ii}^{(i-1)} \le \epsilon$, and therefore, from (3.15), we have

$$\|\tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}} E_{\bar{\mathcal{J}},i}\| = \|\tilde{L}^{-1}_{\bar{\mathcal{J}}_i,\bar{\mathcal{J}}_i} E_{\bar{\mathcal{J}}_i,i}\| < \epsilon^{1/2}.$$

Since this bound holds for all $i \in \mathcal{J}$, we have

$$\|\tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}} E_{\bar{\mathcal{J}}\mathcal{J}}\| \le |\mathcal{J}|^{1/2} \epsilon^{1/2} \le (m\epsilon)^{1/2},$$

as required. □

We are now able to derive an estimate of the difference between $\tilde{L}^T z$ and $\tilde{L}^T \tilde{z}$.

THEOREM 3.4. *Suppose that* (3.10) *holds. For the exact solution z and approximate solution $\tilde{z}$ defined in* (3.8) *and* (3.9), *respectively, we have that*

$$\|\tilde{L}^T [z - \tilde{z}]\| = \|\tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}} E_{\bar{\mathcal{J}}\mathcal{J}} z_{\mathcal{J}}\| \le (m\epsilon)^{1/2} \|z_{\mathcal{J}}\|. \tag{3.19}$$

Proof. From (3.8) together with (3.5), we have

$$\begin{aligned} r_{\bar{\mathcal{J}}} &= M_{\bar{\mathcal{J}}\bar{\mathcal{J}}} z_{\bar{\mathcal{J}}} + M_{\bar{\mathcal{J}}\mathcal{J}} z_{\mathcal{J}} \\ &= \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}} \tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}} z_{\bar{\mathcal{J}}} + \left[\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}} \tilde{L}^T_{\mathcal{J}\bar{\mathcal{J}}} + E_{\bar{\mathcal{J}}\mathcal{J}} \right] z_{\mathcal{J}} \\ &= \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}} \tilde{L}^T_{\cdot\bar{\mathcal{J}}} z + E_{\bar{\mathcal{J}}\mathcal{J}} z_{\mathcal{J}}, \end{aligned}$$

while from (3.9), we have

$$r_{\bar{\mathcal{J}}} = \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}} \tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}} \tilde{z}_{\bar{\mathcal{J}}} = \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}} \left[\tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}} \tilde{z}_{\bar{\mathcal{J}}} + \tilde{L}^T_{\mathcal{J}\bar{\mathcal{J}}} \tilde{z}_{\mathcal{J}} \right] = \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}} \tilde{L}^T_{\cdot\bar{\mathcal{J}}} \tilde{z}.$$

By combining these two relations, we obtain

$$\tilde{L}^T_{\cdot\bar{\mathcal{J}}} [z - \tilde{z}] = -\tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}} E_{\bar{\mathcal{J}}\mathcal{J}} z_{\mathcal{J}}. \tag{3.20}$$

Since $\tilde{L}_{\cdot\mathcal{J}} = 0$, the result follows immediately. □

The remaining analysis of this section requires some additional assumptions on the distribution of the singular values of M and on the parameter ϵ. Accordingly, we introduce a little more notation. The eigenvalues of M are denoted by σ_i^2, $i = 1, 2, \dots, m$, where

$$\sigma_1^2 \ge \sigma_2^2 \ge \cdots \ge \sigma_m^2 > 0. \tag{3.21}$$

We define the diagonal matrix Σ by

$$\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \dots, \sigma_m). \tag{3.22}$$

It follows that there exists an orthogonal matrix Q such that

$$M = Q \Sigma^2 Q^T. \tag{3.23}$$

Because the largest diagonal in M is 1 by assumption (3.10), we have by elementary analysis that

$$1 \le \sigma_1^2 \le m. \tag{3.24}$$

In the subsequent analysis, we assume that there is an integer p with $1 \le p \le m$ such that

- ϵ is somewhat smaller than σ_p^2; and
- if $p < m$, there is a significant gap in the spectrum of M between σ_p^2 and σ_{p+1}^2.

(We will be more specific about these two assumptions presently. In particular, we show in Lemma 3.5 that they imply that $|\bar{\mathcal{J}}| \geq p$.) By partitioning the spectrum at the gap, we obtain

$$\Sigma_1 = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p), \qquad \Sigma_2 = \text{diag}(\sigma_{p+1}, \sigma_{p+2}, \ldots, \sigma_m). \tag{3.25}$$

From (3.23), Q can be partitioned accordingly to obtain

$$Q = [Q_1 \,|\, Q_2], \qquad M = Q_1 \Sigma_1^2 Q_1^T + Q_2 \Sigma_2^2 Q_2^T.$$

Since $M = LL^T$, it follows that σ_i, $i = 1, 2, \ldots, m$, are the singular values of L. In fact, we must have

$$L^T = U\Sigma Q^T = U_1 \Sigma_1 Q_1^T + U_2 \Sigma_2 Q_2^T \tag{3.26}$$

for some $m \times m$ orthogonal matrix $U = [U_1 \,|\, U_2]$, where Σ and Q are defined as above.

We use $\tilde{\sigma}_i^2$, $i = 1, 2, \ldots, m$, to denote the eigenvalues of the perturbed matrix $\tilde{M}$. It follows immediately from (3.4) that the singular values of $\tilde{L}$ are $\tilde{\sigma}_i$, $i = 1, 2, \ldots, m$. The rank of $\tilde{L}$ is $|\bar{\mathcal{J}}|$, because $\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$ is lower triangular with nonzero diagonals while $\tilde{L}_{\cdot\mathcal{J}} = 0$. Therefore, we have

$$\tilde{\sigma}_{|\bar{\mathcal{J}}|} > \tilde{\sigma}_{|\bar{\mathcal{J}}|+1} = \cdots = \tilde{\sigma}_m = 0. \tag{3.27}$$

As in (3.26), there are orthogonal $m \times m$ matrices $\tilde{U}$ and $\tilde{Q}$ such that

$$\tilde{M} = \tilde{Q}\tilde{\Sigma}^2\tilde{Q}^T = \tilde{Q}_1 \tilde{\Sigma}_1^2 \tilde{Q}_1^T + \tilde{Q}_2 \tilde{\Sigma}_2^2 \tilde{Q}_2^T, \tag{3.28a}$$

$$\tilde{L}^T = \tilde{U}\tilde{\Sigma}\tilde{Q}^T = \tilde{U}_1 \tilde{\Sigma}_1 \tilde{Q}_1^T + \tilde{U}_2 \tilde{\Sigma}_2 \tilde{Q}_2^T, \tag{3.28b}$$

where

$$\tilde{\Sigma}_1 = \text{diag}(\tilde{\sigma}_1, \tilde{\sigma}_2, \ldots, \tilde{\sigma}_p), \qquad \tilde{\Sigma}_2 = \text{diag}(\tilde{\sigma}_{p+1}, \ldots, \tilde{\sigma}_m), \tag{3.29}$$

with a corresponding partitioning for $\tilde{U} = [\tilde{U}_1 \,|\, \tilde{U}_2]$ and $\tilde{Q} = [\tilde{Q}_1 \,|\, \tilde{Q}_2]$. It is an immediate consequence of an eigenvalue perturbation result of Stewart and Sun [12, Corollary IV.4.13] and of our Lemma 3.2 that

$$\sum_{i=1}^{m} [\sigma_i^2 - \tilde{\sigma}_i^2]^2 \leq \|E\|_F^2 = \bar{\epsilon}. \tag{3.30}$$

The following result shows that if ϵ is sufficiently small relative to the pth eigenvalue of M, then at least p pivots are accepted during **modchol**.

LEMMA 3.5. *If $\bar{\epsilon}^{1/2} < \sigma_p^2$, we have $|\bar{\mathcal{J}}| \geq p$.*

Proof. If $|\bar{\mathcal{J}}| < p$, we have from (3.27) and (3.30) that

$$\sigma_p^2 \leq \sigma_{|\bar{\mathcal{J}}|+1}^2 = \left| \sigma_{|\bar{\mathcal{J}}|+1}^2 - \tilde{\sigma}_{|\bar{\mathcal{J}}|+1}^2 \right| \leq \bar{\epsilon}^{1/2},$$

contradicting our assumption that $\bar{\epsilon}^{1/2} < \sigma_p^2$. □

Our next result concerns the differences between the subspaces spanned by Q_1 and by $\tilde{Q}_1$, the spaces of "large" eigenvalues of M and $\tilde{M}$, respectively.

LEMMA 3.6. *Suppose that* $|\bar{\mathcal{J}}| < m$ *and that the values* σ_p *and* σ_{p+1} *from* (3.21) *and* ϵ *from* **modchol** *satisfy the conditions*

$$\sigma_p^2 - \sigma_{p+1}^2 > 5\bar{\epsilon}^{1/2}. \tag{3.31}$$

Then there are matrices

$\tilde{\Lambda}_1$	$p \times p$ *symmetric positive definite,*
$\tilde{\Lambda}_2$	$(m-p) \times (m-p)$ *symmetric positive semidefinite,*
$\bar{Q}_1$	$m \times p$ *orthonormal,*
$\bar{Q}_2$	$m \times (m-p)$ *orthonormal,*

such that

$$\tilde{M} = \bar{Q}\tilde{\Lambda}\bar{Q}^T = \bar{Q}_1\tilde{\Lambda}_1\bar{Q}_1^T + \bar{Q}_2\tilde{\Lambda}_2\bar{Q}_2^T, \tag{3.32}$$

$$\|\bar{Q}_1 - Q_1\| \le \frac{2\bar{\epsilon}^{1/2}}{\sigma_p^2 - \sigma_{p+1}^2 - 2\bar{\epsilon}^{1/2}}, \tag{3.33}$$

$$\|\tilde{\Lambda}_1 - \Sigma_1^2\| \le 2\bar{\epsilon}^{1/2}, \tag{3.34}$$

$$\|\tilde{\Lambda}_2 - \Sigma_2^2\| \le 2\bar{\epsilon}^{1/2}, \tag{3.35}$$

where

$$\bar{Q} = [\bar{Q}_1 \,|\, \bar{Q}_2], \qquad \tilde{\Lambda} = \begin{bmatrix} \tilde{\Lambda}_1 & 0 \\ 0 & \tilde{\Lambda}_2 \end{bmatrix}.$$

Moreover, there are matrices

$V_1 p \times p$ *orthogonal,*
$V_2 (m-p) \times (m-p)$ *orthogonal,*

such that

$$\tilde{\Sigma}_1^2 = V_1^T\tilde{\Lambda}_1V_1, \tilde{Q}_1 = \bar{Q}_1V_1, \tag{3.36a}$$

$$\tilde{\Sigma}_2^2 = V_2^T\tilde{\Lambda}_2V_2, \tilde{Q}_2 = \bar{Q}_2V_2, \tag{3.36b}$$

where $\tilde{\Sigma}$ *and* $\tilde{Q}$ *are defined as in* (3.28).

Proof. Note first that $p \le |\bar{\mathcal{J}}|$ by (3.31) and Lemma 3.5. The result is a straightforward consequence of Theorem V.2.8 of Stewart and Sun [12, p. 238]. Since $\tilde{M} = M - E$, we use (3.23) and partition as in (3.25) to obtain

$$Q^T\tilde{M}Q = Q^TMQ - Q^TEQ = \begin{bmatrix} \Sigma_1^2 & 0 \\ 0 & \Sigma_2^2 \end{bmatrix} - \begin{bmatrix} F_{11} & F_{12} \\ F_{12}^T & F_{22} \end{bmatrix}.$$

We now make the following identifications with the quantities in the cited result:

$$\tilde{\gamma} = \|F_{12}^T\| \le \|F\| = \|E\| \le \bar{\epsilon}^{1/2}, \qquad \tilde{\eta} = \|F_{12}\| \le \bar{\epsilon}^{1/2},$$
$$\tilde{\delta} = \operatorname{sep}(\Sigma_1^2, \Sigma_2^2) - \|F_{11}\| - \|F_{22}\| \ge \sigma_p^2 - \sigma_{p+1}^2 - 2\bar{\epsilon}^{1/2} > 2\bar{\epsilon}^{1/2},$$

where $\operatorname{sep}(\cdot,\cdot)$ denotes the minimum distance between the spectra of the two arguments. From the given result, there is a matrix P of dimension $(m-p) \times p$ such that the matrix $\bar{Q}_1$ defined by

$$\bar{Q}_1 = Q_1 + Q_2P \tag{3.37}$$

is an invariant subspace for $\tilde{M}$, where

$$\|P\| \leq \frac{\tilde{\gamma}}{\tilde{\delta}} \leq \frac{2\bar{\epsilon}^{1/2}}{\sigma_p^2 - \sigma_{p+1}^2 - 2\bar{\epsilon}^{1/2}} < 1. \tag{3.38}$$

Moreover, the representation of $\tilde{M}$ with respect to $\bar{Q}_1$ is

$$\bar{Q}_1^T \tilde{M} \bar{Q}_1 = \tilde{\Lambda}_1 = \Sigma_1^2 + F_{11} + F_{12}P. \tag{3.39}$$

The bound (3.33) follows from (3.37), (3.38), and $\|Q_2\| = 1$. It follows immediately from the first equality in (3.39) that $\tilde{\Lambda}_1$ is symmetric, and we have

$$\|\tilde{\Lambda}_1 - \Sigma_1^2\| \leq \|F_{11}\| + \|F_{12}\|\|P\| \leq 2\bar{\epsilon}^{1/2}, \tag{3.40}$$

verifying the inequality (3.34). This inequality implies that the smallest singular value of $\tilde{\Lambda}$ is no smaller than $\sigma_p^2 - 2\bar{\epsilon}^{1/2}$, which by (3.31) is positive, so $\tilde{\Lambda}_1$ is symmetric positive definite.

The cited result states further that the matrix $\bar{Q}_2 = Q_2 - Q_1 P^T$ is orthogonal to $\bar{Q}_1$ and also defines an invariant subspace for $\tilde{M}$, with

$$\bar{Q}_2^T \tilde{M} \bar{Q}_2 = \tilde{\Lambda}_2.$$

Symmetric positive semidefiniteness of $\tilde{\Lambda}_2$ follows immediately. By using the invariant subspace property, we obtain

$$[\bar{Q}_1 \,|\, \bar{Q}_2]^T \tilde{M} [\bar{Q}_1 \,|\, \bar{Q}_2] = \begin{bmatrix} \tilde{\Lambda}_1 & 0 \\ 0 & \tilde{\Lambda}_2 \end{bmatrix},$$

from which (3.32) follows immediately.

Similarly to (3.40), we have that

$$\|\tilde{\Lambda}_2 - \Sigma_2^2\| \leq 2\bar{\epsilon}^{1/2},$$

so the largest eigenvalue of $\tilde{\Lambda}_2$ is no larger than $\sigma_{p+1}^2 + 2\bar{\epsilon}^{1/2}$. Because of (3.31) and our earlier observation that the smallest eigenvalue of $\tilde{\Lambda}_1$ is no smaller than $\sigma_p^2 - 2\bar{\epsilon}^{1/2}$, we conclude that the eigenvalues of $\tilde{\Lambda}_1$ are the p largest eigenvalues $\tilde{\sigma}_1^2, \tilde{\sigma}_2^2, \ldots, \tilde{\sigma}_p^2$, while those of $\tilde{\Lambda}_2$ are the $(m-p)$ smallest eigenvalues. By our definition (3.29), we conclude that there are orthogonal matrices V_1 and V_2 such that

$$V_1 \tilde{\Sigma}_1^2 V_1^T = \tilde{\Lambda}_1 \quad \text{and} \quad V_2 \tilde{\Sigma}_2^2 V_2^T = \tilde{\Lambda}_2.$$

By substituting these expressions into (3.32) and setting $\tilde{Q}_1 = \bar{Q}_1 V_1$ and $\tilde{Q}_2 = \bar{Q}_2 V_2$, we recover (3.28a). □

Lemma 3.6 suggests a few other estimates and assumptions that will be useful in subsequent sections. When (3.31) holds, we have from (3.30) that

$$\tilde{\sigma}_1^2 \leq \sigma_1^2 + \bar{\epsilon}^{1/2} < \sigma_1^2 + .2\sigma_p^2 < 1.2\sigma_1^2 \leq 1.2m \tag{3.41}$$

(where the last inequality follows from (3.24)), and also that

$$\tilde{\sigma}_p^2 \geq \sigma_p^2 - \bar{\epsilon}^{1/2} \geq .8\sigma_p^2 \quad \Rightarrow \quad \tilde{\sigma}_p^{-1} \leq 1.2\sigma_p^{-1}. \tag{3.42}$$

When we make the additional assumption that

$$\frac{\sigma_{p+1}^2}{\sigma_p^2} \le \frac{1}{10} \tag{3.43}$$

(indicating that the gap in the spectrum actually separates the small and large eigenvalues), we derive that

$$\begin{aligned}\|\bar{Q}_1 - Q_1\| &\le \frac{2\bar{\epsilon}^{1/2}}{\sigma_p^2 - \sigma_{p+1}^2 - 2\bar{\epsilon}^{1/2}} \\ &= \frac{2\bar{\epsilon}^{1/2}}{\sigma_p^2}\left[1 - \frac{\sigma_{p+1}^2}{\sigma_p^2} - 2\frac{\bar{\epsilon}^{1/2}}{\sigma_p^2}\right]^{-1} \\ &\le \frac{2\bar{\epsilon}^{1/2}}{\sigma_p^2}[1 - .1 - .4]^{-1} \le \frac{4\bar{\epsilon}^{1/2}}{\sigma_p^2}. \end{aligned} \tag{3.44}$$

Another useful quantity that enters into our error bounds is the norm of $\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-1}$, which we denote by τ; that is,

$$\tau \overset{\text{def}}{=} \|\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-1}\| = \sigma_{|\bar{\mathcal{J}}|}(\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}})^{-1}, \tag{3.45}$$

where $\sigma_{|\bar{\mathcal{J}}|}(\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}})$ denotes the $|\bar{\mathcal{J}}|$th singular value of $\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$. Because of (3.5a) and the fact that all diagonals of $M_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$ are bounded by 1 (by our assumption (3.10)), we have that $\sigma_{|\bar{\mathcal{J}}|}(\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}) \le 1$ and therefore that

$$\tau \ge 1. \tag{3.46}$$

Using (3.5a) again, we have that

$$\|M_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-1}\| = \|\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^{-1}\|^2 = \tau^2. \tag{3.47}$$

Since $\|M_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\| \le \|M\| \le \sigma_1^2$, we have from (3.24) and (3.47) that

$$\kappa(M_{\bar{\mathcal{J}}\bar{\mathcal{J}}}) \le \sigma_1^2\tau^2 \le m\tau^2. \tag{3.48}$$

4. The effect of finite-precision computations.

In the analysis of the preceding section, we assumed for simplicity that all arithmetic was exact. In this section, we take account of the roundoff errors that are introduced when the approximate solution $\tilde{z}$ is calculated in a finite-precision environment.

Our analysis above focused on the approximate solution $\tilde{z}$ obtained from (3.9), where the subvector $\tilde{z}_{\bar{\mathcal{J}}}$ satisfies the system

$$M_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{z}_{\bar{\mathcal{J}}} = \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}^T\tilde{z}_{\bar{\mathcal{J}}} = r_{\bar{\mathcal{J}}}, \tag{4.1}$$

while the subvector $\tilde{z}_{\mathcal{J}}$ is fixed at zero. In this section, we use $\hat{z}$ to denote the finite-precision analog of $\tilde{z}$. We examine errors in $\hat{z}$ due to

- roundoff error in **modchol**,
- error arising during the triangular substitutions in (4.1), and
- error in the evaluation of the matrix M and the right-hand-side r.

Since **modchol** amounts to a standard Cholesky factorization/triangular-solve procedure on the matrix $M_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$, roundoff error in **modchol** and errors arising during the triangular substitutions can all be accounted for by adding a term $E^{\mathbf{u}}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$ to the coefficient matrix $M_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$ in (4.1), where

$$\|E^{\mathbf{u}}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\| \le \delta_{\mathbf{u}}\|M_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\| \le \delta_{\mathbf{u}}; \tag{4.2}$$

see, for example, Higham [7, Theorem 10.4]. (Recall from section 1 that $\delta_{\mathbf{u}}$ denotes a modest multiple of $\mathbf{u}$ and that $\|M_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\| \le \sqrt{n}$ because of (3.10).) We assume that the error in evaluating M can also be incorporated into $E^{\mathbf{u}}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$; this is certainly true in section 5, for instance. As we see in this section, the remaining source of error—the error that arises in evaluation of the right-hand side—plays a significant role in the interior-point application. Our results are strengthened if we account for some of this error by placing it explicitly in the range space of L; that is, we write it as $Lf + e$, for some vectors f and e. (We refer to e as the "unstructured error.") The computed solution $\hat{z}_{\bar{\mathcal{J}}}$ of the system (4.1) therefore satisfies

$$(M_{\bar{\mathcal{J}}\bar{\mathcal{J}}} + E^{\mathbf{u}}_{\bar{\mathcal{J}}\bar{\mathcal{J}}})\hat{z}_{\bar{\mathcal{J}}} = (r + Lf + e)_{\bar{\mathcal{J}}}. \tag{4.3}$$

The following result shows that we can repartition the right-hand-side error according to the approximate Cholesky factor $\tilde{L}$, a fact that is useful in the main error results of this section.

LEMMA 4.1. *Suppose that* (3.10), (3.31), *and* (3.43) *hold. Given vectors* $e, f \in \mathsf{R}^m$, *we have*

$$Lf + e = \tilde{L}\tilde{f} + \tilde{e}, \tag{4.4}$$

where

$$\|\tilde{f}\| \le \delta_1\sigma_p^{-1}\|f\|, \qquad \|\tilde{e}\| \le \delta_1\left(\bar{\epsilon}^{1/2}\sigma_p^{-3} + \sigma_{p+1}\right)\|f\| + \|e\|. \tag{4.5}$$

Proof. From (3.26), we have

$$Lf + e = Q_1\Sigma_1U_1^Tf + Q_2\Sigma_2U_2^Tf + e = Q_1\Sigma_1^2f_1 + e_1,$$

where the vectors f_1 and e_1 defined by

$$f_1 = \Sigma_1^{-1}U_1^Tf, \qquad e_1 = Q_2\Sigma_2U_2^Tf + e$$

satisfy the bounds

$$\|f_1\| \le \sigma_p^{-1}\|f\|, \qquad \|e_1\| \le \sigma_{p+1}\|f\| + \|e\|; \tag{4.6}$$

see (3.25). Using the notation of (3.28), (3.29), and (3.32), we define the vector $\tilde{e}$ by

$$\tilde{e} = (Q_1 - \bar{Q}_1)\tilde{\Lambda}_1f_1 + Q_1(\Sigma_1^2 - \tilde{\Lambda}_1)f_1 + e_1$$

and note that

$$Lf + e = Q_1\Sigma_1^2f_1 + e_1 = \bar{Q}_1\tilde{\Lambda}_1f_1 + \tilde{e}. \tag{4.7}$$

By using (3.34), (3.41), (3.44), and (4.6), we can bound the terms of $\tilde{e}$ to obtain

$$\begin{aligned}\|\tilde{e}\| &\le \|Q_1 - \bar{Q}_1\|\,\|\tilde{\Lambda}_1\|\,\|f_1\| + \|\Sigma_1^2 - \tilde{\Lambda}_1\|\|f_1\| + \|e_1\| \\ &\le 4\frac{\bar{\epsilon}^{1/2}}{\sigma_p^2}(1.2\sigma_1^2)\sigma_p^{-1}\|f\| + 2\bar{\epsilon}^{1/2}\sigma_p^{-1}\|f\| + \sigma_{p+1}\|f\| + \|e\|,\end{aligned}$$

from which the bound in (4.5) follows if we use the inequality (3.24). For the companion term on the right-hand side of (4.7), we have from (3.36) that

$$\bar{Q}_1\tilde{\Lambda}_1 f_1 = \bar{Q}_1 V_1(V_1^T\tilde{\Lambda}_1 V_1)(V_1^T f_1) = \tilde{Q}_1\tilde{\Sigma}_1(\tilde{\Sigma}_1 V_1^T f_1).$$

Using $\tilde{U}$ defined in (3.28b), we set

$$\tilde{f} = [\tilde{U}_1 \,|\, \tilde{U}_2]\left[\begin{array}{c} \tilde{\Sigma}_1 V_1^T f_1 \\ 0 \end{array}\right],$$

so from (3.28b) and (3.36a), we obtain that

$$\tilde{L}\tilde{f} = \tilde{Q}_1\tilde{\Sigma}_1\tilde{U}_1^T\tilde{f} + \tilde{Q}_2\tilde{\Sigma}_2\tilde{U}_2^T\tilde{f} = \tilde{Q}_1\tilde{\Sigma}_1(\tilde{\Sigma}_1 V_1^T f_1) = \bar{Q}_1\tilde{\Lambda}_1 f_1.$$

Hence, by substituting in (4.7), we obtain $Lf + e = \tilde{L}\tilde{f} + \tilde{e}$. To obtain the bound on $\|\tilde{f}\|$, we simply use its definition above together with (3.41), (4.6), and the orthonormality of $\tilde{U}_1$ and V_1. $\square$

Before stating our main result, we introduce two additional assumptions. The first is that finite precision does not affect cutoff decisions in **modchol**. That is, the presence of roundoff error in each submatrix $M^{(i-1)}$ does not affect whether the threshold criterion $M_{ii}^{(i-1)} \le \beta\epsilon$ passes or fails for each i. Provided that we have

$$\epsilon \ge 100\mathbf{u}, \tag{4.8}$$

say, the role of this assumption is to provide a convenient link between the results of sections 3 and 4. It is not really essential to the analysis, for reasons that we now explain. We can show by a standard error analysis argument that the matrix $\tilde{L}$ obtained in finite-precision arithmetic is the same as the one we would obtain by applying **modchol** in exact arithmetic to a perturbed matrix $M + \hat{E}^{\mathbf{u}}$, where $\|\hat{E}^{\mathbf{u}}\| \le \delta_{\mathbf{u}}\|M\| \le \delta_{\mathbf{u}}$. Hence, finite-precision arithmetic causes changes of size $\delta_{\mathbf{u}}$ in the diagonal elements that are tested against the threshold $\beta\epsilon$ in **modchol**. If $\mathbf{u}$ is significantly less than $\beta\epsilon$ (as in (4.8)), only a few skipping decisions would be affected by this perturbation. Moreover, we could generalize the analysis of section 3 so that it applies to the slightly perturbed matrix $M + \hat{E}^{\mathbf{u}}$ rather than to the exact matrix M, hence ensuring that the results of that section apply to the set $\mathcal{J}$ calculated in a finite-precision environment. We prefer to avoid the additional complication, however, and simply assume that the sets $\mathcal{J}$ that we discuss in sections 3 and 4 are one and the same. In any case, we note that when $\bar{\epsilon}$ falls in the gap between large and small eigenvalues, the makeup of $\mathcal{J}$ is not affected at all.

The second assumption is that

$$\tau\bar{\epsilon}^{1/2} = \delta_1. \tag{4.9}$$

We can expect this estimate to hold in all but pathological cases, since the elements of $\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$ are bounded by 1, and its diagonal elements lie in the range $[\bar{\epsilon}^{1/2}, 1]$.

In the following result, we bound the difference $L^T(\hat{z} - z)$ in terms of $\|\hat{z}\|$, $\|z\|$, and the norms $\|f\|$ and $\|e\|$ of the perturbation vectors. The explicit appearance of the computed solution $\|\hat{z}\|$ in the right-hand-side bound is not standard practice in error analysis, but we were motivated to include it by our numerical experience on practical linear programming problems. We can derive a rigorous bound on $\|\hat{z}\|$ in terms of $\|z\|$, $\|f\|$, and $\|e\|$, but numerical experience shows that this bound appears to be too pessimistic, so it turns out to be more illuminating to leave $\|\hat{z}\|$ in place and to work with a direct estimate of this quantity.

THEOREM 4.2. *Suppose that* $\hat{z}_{\bar{\mathcal{J}}}$ *solves* (4.3), *where* $E^{\mathbf{u}}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$ *is bounded as in* (4.2). *Suppose too that we set* $\hat{z}_{\mathcal{J}} = 0$ (*as in* (3.9)), *that* (3.10), (3.31), (4.9), *and* (3.43) *hold, and that roundoff error does not affect the composition of* $\mathcal{J}$. *We then have*

$$\|L^T(\hat{z}-z)\| \le \delta_1 \left[\sigma_p^{-2}(\tau\mathbf{u} + \bar{\epsilon}^{1/2}) + \sigma_{p+1}\right] \|\hat{z}\| + \delta_1 \left[\sigma_p^{-2}\bar{\epsilon}^{1/2} + \sigma_{p+1}\right] \|z\| \\ + \delta_1 \left(\sigma_p^{-4} + \tau\sigma_{p+1}\sigma_p^{-1}\right) \|f\| + \delta_1\tau\sigma_p^{-1}\|e\|, \tag{4.10}$$

where z *is the exact solution from* (3.7). *In the special case of* $J = \emptyset$, *we have*

$$\|L^T(\hat{z}-z)\| \le \tau\delta_{\mathbf{u}}\sigma_1\|\hat{z}\| + \|f\| + \tau\|e\|. \tag{4.11}$$

Proof. From (3.26), we have

$$\begin{aligned} \|L^T(\hat{z}-z)\| &= \left\| \begin{bmatrix} \Sigma_1 Q_1^T(\hat{z}-z) \\ \Sigma_2 Q_2^T(\hat{z}-z) \end{bmatrix} \right\| \\ &\le \|\Sigma_1\|\,\|Q_1^T(\hat{z}-z)\| + \|\Sigma_2\|\,\|\hat{z}-z\| \\ &\le \|\Sigma_1\|\,\|\bar{Q}_1^T(\hat{z}-z)\| + \|\Sigma_1\|\,\|Q_1 - \bar{Q}_1\|\|\hat{z}-z\| + \|\Sigma_2\|\,\|\hat{z}-z\|. \qquad (4.12) \end{aligned}$$

To bound the first term, we note from (3.28b) that

$$\|\tilde{L}^T(\hat{z}-z)\| = \left\| \begin{bmatrix} \tilde{\Sigma}_1 \tilde{Q}_1^T(\hat{z}-z) \\ \tilde{\Sigma}_2 \tilde{Q}_2^T(\hat{z}-z) \end{bmatrix} \right\|,$$

and therefore, from (3.36a) and (3.29), we have

$$\|\bar{Q}_1^T(\hat{z}-z)\| = \|\tilde{Q}_1^T(\hat{z}-z)\| \le \|\tilde{\Sigma}_1^{-1}\|\,\|\tilde{\Sigma}_1\tilde{Q}_1^T(\hat{z}-z)\| \le \tilde{\sigma}_p^{-1}\|\tilde{L}^T(\hat{z}-z)\|. \tag{4.13}$$

Since $\tilde{L}_{\cdot\mathcal{J}} = 0$ and $\hat{z}_{\mathcal{J}} = 0$, we have too that

$$\tilde{L}^T(\hat{z}-z) = \tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}}(\hat{z}_{\bar{\mathcal{J}}} - z_{\bar{\mathcal{J}}}) - \tilde{L}^T_{\mathcal{J}\bar{\mathcal{J}}} z_{\mathcal{J}}. \tag{4.14}$$

By substituting (3.42) and (4.14) into (4.13), we obtain

$$\|\bar{Q}_1^T(\hat{z}-z)\| \le 1.2\sigma_p^{-1}\|\tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}}(\hat{z}_{\bar{\mathcal{J}}} - z_{\bar{\mathcal{J}}}) - \tilde{L}^T_{\mathcal{J}\bar{\mathcal{J}}} z_{\mathcal{J}}\|. \tag{4.15}$$

From (4.3) and (4.4), and using (3.5a) and $\tilde{L}_{\cdot\mathcal{J}} = 0$, we have that

$$(\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}} + E^{\mathbf{u}}_{\bar{\mathcal{J}}\bar{\mathcal{J}}})\hat{z}_{\bar{\mathcal{J}}} = r_{\bar{\mathcal{J}}} + \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{f}_{\bar{\mathcal{J}}} + \tilde{e}_{\bar{\mathcal{J}}}.$$

Meanwhile, from (3.5) and (3.7), we have

$$\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}} z_{\bar{\mathcal{J}}} + \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{L}^T_{\mathcal{J}\bar{\mathcal{J}}} z_{\mathcal{J}} + E_{\bar{\mathcal{J}}\mathcal{J}} z_{\mathcal{J}} = r_{\bar{\mathcal{J}}}.$$

By combining these two equations and multiplying by $\tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$, we obtain

$$\tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}}(\hat{z}_{\bar{\mathcal{J}}} - z_{\bar{\mathcal{J}}}) - \tilde{L}^T_{\mathcal{J}\bar{\mathcal{J}}} z_{\mathcal{J}} = -\tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}} E^{\mathbf{u}}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\hat{z}_{\bar{\mathcal{J}}} + \tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}} E_{\bar{\mathcal{J}}\mathcal{J}} z_{\mathcal{J}} + \tilde{f}_{\bar{\mathcal{J}}} + \tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{e}_{\bar{\mathcal{J}}}.$$

By substituting into (4.15), and using the bounds (3.45), (3.12), and (4.2), we obtain

$$\|\bar{Q}_1^T(\hat{z}-z)\| \le \tau\delta_{\mathbf{u}}\sigma_p^{-1}\|\hat{z}_{\bar{\mathcal{J}}}\| + \delta_1\bar{\epsilon}^{1/2}\|z_{\mathcal{J}}\| + \|\tilde{f}_{\bar{\mathcal{J}}}\| + \tau\|\tilde{e}_{\bar{\mathcal{J}}}\|. \tag{4.16}$$

Turning now to the second and third terms in (4.12), we have from (3.25) that

$$\|\Sigma_1\| = \sigma_1 = \delta_1, \qquad \|\Sigma_2\| = \sigma_{p+1}. \tag{4.17}$$

By substituting (4.15), (4.16), (4.17), and (3.44) into (4.12), and using

$$\|\hat z - z\| \le \|\hat z\| + \|z\|, \qquad \|\hat z_{\bar{\mathcal{J}}}\| \le \|\hat z\|, \qquad \|z_{\mathcal{J}}\| \le \|z\|, \qquad 1 \le \delta_1 \sigma_p^{-1},$$

we obtain

$$\begin{aligned}
&\|L^T(\hat z - z)\| \\
&\le \delta_1\sigma_p^{-1}\left[\tau\mathbf{u}\|\hat z\| + \bar\epsilon^{1/2}\|z\| + \|\tilde f\| + \tau\|\tilde e\|\right] + \delta_1\left(\sigma_p{}^{-2}\bar\epsilon^{1/2} + \sigma_{p+1}\right)(\|\hat z\| + \|z\|) \\
&\le \delta_1\left[\sigma_p^{-2}(\tau\mathbf{u} + \bar\epsilon^{1/2}) + \sigma_{p+1}\right]\|\hat z\| + \delta_1\left[\sigma_p^{-2}\bar\epsilon^{1/2} + \sigma_{p+1}\right]\|z\| \\
&\quad + \delta_1\sigma_p^{-1}\|\tilde f\| + \delta_1\tau\sigma_p^{-1}\|\tilde e\|.
\end{aligned} \tag{4.18}$$

By substituting from (4.5) and using (4.9), we have

$$\delta_1\sigma_p^{-1}\|\tilde f\| + \delta_1\tau\sigma_p^{-1}\|\tilde e\| \le \delta_1\left(\sigma_p^{-4} + \tau\sigma_{p+1}\sigma_p^{-1}\right)\|f\| + \delta_1\tau\sigma_p{}^{-1}\|e\|.$$

By substituting into (4.18), we obtain (4.10).

For the case of $\mathcal{J} = \emptyset$, we have

$$\tilde L_{\bar{\mathcal{J}}\bar{\mathcal{J}}} = \tilde L = L, \qquad \hat z_{\bar{\mathcal{J}}} = \hat z, \qquad z_{\bar{\mathcal{J}}} = z, \qquad z_{\mathcal{J}} \;\text{ vacuous},$$

while from (4.4), we have $\tilde f = f$, $\tilde e = e$. By using these equivalences in (4.16), we obtain the result (4.11) directly. □

Note that in the case of $\mathcal{J} = \emptyset$, we have from (3.45) that

$$\tau = \|L^{-1}\| = \sigma_m^{-1},$$

so it follows from (4.11) that

$$\|\hat z - z\| \le \sigma_m^{-2}\delta_{\mathbf{u}}\|\hat z\| + \sigma_m^{-1}\|f\| + \sigma_m^{-2}\|e\|.$$

If we put all the right-hand-side perturbations into the vector e, and set $f = 0$, we can use the relation $\|M^{-1}\| = \sigma_m^{-2}$ to obtain

$$\|\hat z - z\| \le \|M^{-1}\|\left(\delta_{\mathbf{u}}\|\hat z\| + \|e\|\right),$$

which is a perturbation bound for (4.3) of the type that is usually found in the numerical analysis literature.

5. Application to the interior-point algorithm. We now return to the motivating application: primal-dual interior-point algorithms for linear programming and, in particular, the linear system (2.15) that is solved at each iteration. We apply the main result, Theorem 4.2, and examine the effect of the parameter ϵ and unit roundoff $\mathbf{u}$ on the quality of the computed search direction $(\widehat{\Delta x}, \widehat{\Delta \pi}, \widehat{\Delta s})$. Our focus is on the later iterations of the interior-point method, during which μ is small and the ill conditioning of AD^2A^T can become acute. Our results show where errors arise in $(\widehat{\Delta x}, \widehat{\Delta \pi}, \widehat{\Delta s})$, what effect these errors have on the steplength and the computed residual vectors r_b and r_c, and the accuracy that can be attained by the interior-point algorithm in finite precision. They also suggest a choice for the parameter ϵ and for the termination criterion.

Throughout this section, we use an informal style of analysis that combines the use of δ_1 and order notation defined in section 1. Specifically, we often replace the estimate $v = O(\epsilon)$ by $v = \delta_1 \epsilon$ instead. This convention allows us to analyze the dependence of certain quantities on the unit roundoff $\mathbf{u}$ and the duality measure μ jointly.

5.1. Size estimate for a general step. We start by estimating the sizes of the various constituents of the equations (2.15)—the residuals r_b and r_c of (2.7), the $\mathcal{B}$ and $\mathcal{N}$ components of x, s, and the diagonal matrix D. Each iterate (x, π, s) of a typical primal-dual interior-point iterate satisfies the following estimates (see, for example, S. J. Wright [17]):

$$\begin{aligned} \|r_b\| &= O(\mu), & \|r_c\| &= O(\mu), \\ x_i &= \Theta(1) \;\; (i \in \mathcal{B}), & x_i &= \Theta(\mu) \;\; (i \in \mathcal{N}), \\ s_i &= \Theta(\mu) \;\; (i \in \mathcal{B}), & s_i &= \Theta(1) \;\; (i \in \mathcal{N}). \end{aligned} \tag{5.1}$$

In theoretical algorithms, these estimates follow from a requirement that all iterates must belong to a certain neighborhood of the central trajectory. In practical algorithms, the conditions for membership of the neighborhood are rarely checked explicitly, but the estimates (5.1) are still observed to hold on the vast majority of practical problems in which the primal-dual solution set is nonempty and bounded. An immediate consequence of these estimates and the definition (2.14) is that

$$D_{ii}^2 = \Theta(\mu^{-1}) \;\; (i \in \mathcal{B}), \qquad D_{ii}^2 = \Theta(\mu) \;\; (i \in \mathcal{N}). \tag{5.2}$$

As mentioned in section 2, we assume that A has full rank.

We analyze a general step $(\Delta x, \Delta \pi, \Delta s)$ that satisfies the system (2.6), where r_b and r_c are given by (2.7) while r_{xs} has the form

$$r_{xs} = XS\mathbf{1} + w \quad \text{for some } w \text{ satisfying } w = O(\mu^2). \tag{5.3}$$

It is not difficult to show that the resulting step satisfies the estimate

$$(\Delta x, \Delta \pi, \Delta s) = O(\mu) \tag{5.4}$$

by using an argument based on splitting the step into an affine-scaling component $(\Delta x^{\text{aff}}, \Delta \pi^{\text{aff}}, \Delta s^{\text{aff}})$ of the step (obtained by setting $w = 0$; see (2.8)) and a "remainder" component $(\Delta x^w, \Delta \pi^w, \Delta s^w)$ that satisfies

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^w \\ \Delta \pi^w \\ \Delta s^w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -w \end{bmatrix}. \tag{5.5}$$

We have from [17, Theorem 7.5] that

$$\|(\Delta x^{\text{aff}}, \Delta s^{\text{aff}})\| = O(\mu), \tag{5.6}$$

while from (2.15b) and (5.1), we have

$$(AA^T)\Delta \pi^{\text{aff}} = A(-r_c - \Delta s^{\text{aff}}) = O(\mu),$$

and since A has full rank, we have $\Delta\pi^{\text{aff}} = O(\mu)$ as well. By performing block elimination on (5.5), we have that

$$AD^2A^T\Delta\pi^w = AD^2(X^{-1}w).$$

A well-known result (see Stewart [11], Todd [13], Dikin [4], and Vanderbei and Lagarias [14]) states that the norm $\|(AD^2A^T)^{-1}AD^2\|$ is bounded over the set of all positive definite diagonal matrices D. Therefore, we have that

$$\|\Delta\pi^w\| = O(\|X^{-1}w\|).$$

From (5.1), we have $\|X^{-1}\| = O(\mu^{-1})$, so from $w = O(\mu^2)$ it follows that $\Delta\pi^w = O(\mu)$. Similar arguments based on the Stewart–Todd result can be used to show that

$$\|\Delta x^w\| = O(\mu), \qquad \|\Delta s^w\| = O(\mu).$$

The general choice (5.3) of w encompasses the affine-scaling method (2.8), for which $w = 0$. It also includes as a special case the path-following choice (2.9) when $\zeta = O(\mu)$, which can be assumed to hold on the late iterations of a superlinearly convergent method. Finally, it usually includes the Mehrotra method (2.11), since by (5.6) we have that $\|\Delta X^{\text{aff}}\Delta S^{\text{aff}}\mathbf{1}\| = O(\mu^2)$, while the heuristic choice of the parameter ζ is usually chosen by a heuristic that ensures that $\zeta = O(\mu)$.

5.2. Steplength along the exact step. We have noted already in (5.4) that $(\Delta x, \Delta\pi, \Delta s) = O(\mu)$. We can be more specific about the sizes of the critical components Δx_i, $i \in \mathcal{N}$, and Δs_i, $i \in \mathcal{B}$. If we multiply the third block row in (2.6) by $(XS)^{-1}$, use the definition (5.3), and note from (5.1) that $(x_is_i)^{-1} = \Theta(\mu^{-1})$ for $i = 1, 2, \ldots, n$, we obtain

$$\frac{\Delta x_i}{x_i} + \frac{\Delta s_i}{s_i} = -1 + O(\mu), \qquad i = 1, 2, \ldots, n.$$

Therefore, from (5.1) and (5.4), we have for $i \in \mathcal{N}$ that

$$\frac{\Delta x_i}{x_i} = -1 + \frac{O(\mu)}{\Theta(1)} = -1 + O(\mu),$$

and therefore, using (5.1) again, we have

$$\Delta x_i = -x_i + O(\mu^2), \qquad i \in \mathcal{N}. \tag{5.7}$$

In a similar way, we obtain

$$\Delta s_i = -s_i + O(\mu^2), \qquad i \in \mathcal{B}. \tag{5.8}$$

From the estimates (5.4), (5.7), and (5.8), we can show that a near-unit step can be taken along the direction $(\Delta x, \Delta\pi, \Delta s)$ without violating positivity of the x and s components. By substituting in (2.13), we can show that

$$1 - \alpha_{\max} = O(\mu). \tag{5.9}$$

To verify this estimate, suppose that $s_i + \alpha\Delta s_i = 0$ for some index $i \in \mathcal{B}$. From (5.8), we have

$$s_i(1-\alpha) + O(\mu^2) = 0,$$

so it follows from (5.1) that

$$1-\alpha = O(\mu^2)/s_i = O(\mu).$$

For the corresponding component x_i, we have from (5.1) and (5.4) that $x_i = \Theta(1)$ and $\Delta x_i = O(\mu)$. Hence, for all μ sufficiently small and all $\alpha \in [0,1]$, we have $x_i + \alpha\Delta x_i > 0$. Similar logic can be applied to the remaining indices $i \in \mathcal{N}$, thereby proving (5.9).

5.3. Scaling the system (2.15a). We can use (5.2) to analyze the eigenstructure of the coefficient matrix AD^2A^T. We have

$$AD^2A^T = A_{\cdot\mathcal{B}}D_{\mathcal{B}}^2A_{\cdot\mathcal{B}}^T + A_{\cdot\mathcal{N}}D_{\mathcal{N}}^2A_{\cdot\mathcal{N}}^T,$$

where the first term on the right-hand side is a matrix whose rank is $\operatorname{rank} A_{\cdot\mathcal{B}}$ in which all the nonzero eigenvalues are of size $\Theta(\mu^{-1})$. By combining this observation with the full-rank assumption on A, we obtain that

$$\sigma_i(AD^2A^T) = \Theta(\mu^{-1}), \qquad i = 1,2,\dots,\operatorname{rank} A_{\cdot\mathcal{B}}, \tag{5.10a}$$

$$\sigma_i(AD^2A^T) = \Theta(\mu), \qquad i = \operatorname{rank} A_{\cdot\mathcal{B}}+1,\dots,m. \tag{5.10b}$$

To ensure (3.10), we work with a scaled version of the matrix AD^2A^T, in which the scaling factor ρ is chosen as

$$\rho = \left[\max_{i=1,2,\dots,m} (AD^2A^T)_{ii}\right]^{-1}. \tag{5.11}$$

Obviously, we have $\rho = \Theta(\mu)$, and by choosing p (see section 3) as

$$p = \operatorname{rank} A_{\cdot\mathcal{B}}, \tag{5.12}$$

we find that the eigenvalues $\sigma_1^2, \sigma_2^2, \dots, \sigma_m^2$ of ρAD^2A^T satisfy

$$\sigma_i^2 = \Theta(1), \qquad i = 1,2,\dots,p, \tag{5.13a}$$

$$\sigma_i^2 = \Theta(\mu^2), \qquad i = p+1,\dots,m. \tag{5.13b}$$

The exact Cholesky factor L satisfies

$$LL^T = \rho AD^2A^T. \tag{5.14}$$

Suppose now that **modchol** is used to compute the solution of the scaled version of the system (2.15a), namely,

$$\rho AD^2A^T\Delta\pi = -\rho r_b - \rho AD^2(r_c - X^{-1}r_{xs}), \tag{5.15}$$

where r_{xs} is defined as in (5.3). This process is carried out in finite-precision arithmetic, resulting in a computed solution $\widehat{\Delta\pi}$. The remaining step components $\widehat{\Delta s}$ and $\widehat{\Delta x}$ are obtained by substituting into (2.15b) and (2.15c), respectively, where once again we assume that finite-precision arithmetic is used.

5.4. Checking assumptions and estimates for Theorem 4.2. We now prepare to apply Theorem 4.2 by checking that its various assumptions are satisfied for μ sufficiently small. We assume that ϵ is set to the following value:

$$\epsilon = 100\mathbf{u}. \tag{5.16}$$

This choice is motivated by a desire to keep ϵ as small as possible, while trying to ensure that the set $\mathcal{J}$ of skipped pivot indices is not greatly affected by the use of finite-precision arithmetic (see the discussion surrounding (4.8)). The assumption (3.10) that the largest diagonal in ρAD^2A^T is 1 is satisfied by construction. From (5.13) and (5.12), the assumptions (3.31) and (3.43) hold trivially. As noted in the discussion following (4.9), this assumption too can be expected to hold in nonpathological cases. It follows immediately from (4.9) that

$$\tau = \delta_1 \bar{\epsilon}^{-1/2}, \tag{5.17}$$

giving us a "worst-case" bound for τ. When **modchol** correctly identifies the numerical rank of AD^2A^T—that is, when $|\bar{\mathcal{J}}| = p = \operatorname{rank} A_{\cdot\mathcal{B}}$, as often happens in the examples we present in the next section—we usually have that all the diagonals of $\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$ are of size δ_1, and hence that $\tau = \delta_1$. Surprisingly, however, our favorable results about the quality of the computed step $(\widehat{\Delta x}, \widehat{\Delta \pi}, \widehat{\Delta s})$ hold even when the algorithm admits some small diagonal elements into $\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$, yielding a computed factor $\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$ for which $|\bar{\mathcal{J}}| > p$.

Having verified that we can reasonably expect Theorem 4.2 to hold for the system (5.15), we now estimate the quantities on the right-hand side of the bound (4.10). From (5.13a), we have $\sigma_p^{-1} = \Theta(1)$, while from (5.13b), we have $\sigma_{p+1} = \Theta(\mu)$.

We need to account, too, for the errors incurred in finite-precision evaluation of the right-hand side of (5.15), and to apportion these errors between the error vectors f and e in (4.3). For the purpose of this discussion, and in the remainder of the paper, we assume that

$$\mu \geq \mathbf{u}. \tag{5.18}$$

(As we see later, the algorithm is usually terminated—and for good reason—when μ is significantly larger than $\mathbf{u}$, so this assumption is not restrictive.) We examine the contributions of the terms r_{xs}, r_b, and r_c to the right-hand side of (5.15) in turn.

In most codes, the contribution of r_{xs} to (5.15) is calculated by forming the vector r_{xs}, multiplying by $D^2X^{-1} = S^{-1}$, and then multiplying by A. Floating-point error in formation of r_{xs} from (5.3) can be bounded by a term of size $\delta_{\mathbf{u}}\mu$. This error is magnified to $\delta_{\mathbf{u}}$ when we multiply by S^{-1}, and further roundoff errors introduced in this operation result in an additional error of size $\delta_{\mathbf{u}}$. Multiplication by A yields additional errors of size $\delta_{\mathbf{u}}$. Therefore, the total contribution of this term to the error in the right-hand side of (5.15), after scaling by ρ, has magnitude $\delta_{\mathbf{u}}\mu$. We denote this error by e_{xs}; below, we include it in the unstructured error vector e in (4.3).

The vectors r_b and r_c both have size μ (see (5.1)), but they are calculated by summing and differencing real-number quantities of size δ_1, and hence incur cancellation error of size $\delta_{\mathbf{u}}$. We denote the calculated versions by $\hat{r}_b$ and $\hat{r}_c$, respectively, so that

$$\hat{r}_b - r_b = \delta_{\mathbf{u}}, \qquad \hat{r}_c - r_c = \delta_{\mathbf{u}}. \tag{5.19}$$

The contribution of the error in $\hat{r}_b$ to the right-hand side of (5.15) is small. After scaling by ρ, it contributes an error of size $\mu\delta_{\mathbf{u}}$, which we denote by e_b and incorporate into e.

The term involving r_c requires more careful consideration. Note from (5.1) and (5.19) that $\hat{r}_c = O(\mu) + \delta_{\mathbf{u}}$. When we multiply $\hat{r}_c$ by D^2, some of whose diagonal elements have size $\Theta(\mu^{-1})$, we incur additional error of $\delta_{\mathbf{u}}\mu^{-1}(\mu+\delta_{\mathbf{u}})$, which is equivalent to $\delta_{\mathbf{u}}$ because of (5.18). Therefore, we have

$$\mathrm{comp}(D^2\hat{r}_c) = D^2(r_c + \delta_{\mathbf{u}}) + \delta_{\mathbf{u}} = D^2 r_c + D^2(\hat{r}_c - r_c) + \delta_{\mathbf{u}},$$

which has size δ_1. Finally, on multiplying by A, we incur additional roundoff error of $\delta_{\mathbf{u}}$, so in summary we have

$$\mathrm{comp}(AD^2\hat{r}_c) = AD^2 r_c + AD^2(\hat{r}_c - r_c) + \delta_{\mathbf{u}}. \tag{5.20}$$

From (5.14), we have that

$$AD = \rho^{-1/2} L Q^T \tag{5.21}$$

for some orthogonal matrix Q, so by defining

$$f = \rho^{1/2} Q D(\hat{r}_c - r_c) = O(\mu^{1/2}) O(\mu^{-1/2})\delta_{\mathbf{u}} = \delta_{\mathbf{u}}, \tag{5.22}$$

we have that

$$\rho AD^2(\hat{r}_c - r_c) = \rho^{1/2} L Q^T D(\hat{r}_c - r_c) = L^T f.$$

Hence, from (5.20), we see that the computed version of the term $\rho AD^2 r_c$ on the right-hand side of (5.15) differs from the exact quantity by $Lf + e_c$, where f is defined as in (5.22) and $e_c = \mu\delta_{\mathbf{u}}$. By adding the unstructured error contributions from the three right-hand-side terms in (5.15), we find that

$$e = e_{xs} + e_b + e_c = \mu\delta_{\mathbf{u}}. \tag{5.23}$$

We have pointed out already (see (5.4)) that $\Delta\pi = O(\mu)$. The one remaining important quantity on the right-hand side of (4.10) is $\|\widehat{\Delta\pi}\|$. By making further assumptions on the relative sizes of τ, $\mathbf{u}$, and ϵ, we can bound this term strictly in terms of $\|\Delta\pi\|$, but the resulting estimate is observed to be too pessimistic. We found the following estimate to hold in all computational tests we performed:

$$\widehat{\Delta\pi} = O(\mu); \tag{5.24}$$

we use this estimate in the results below.

5.5. Errors in the computed step and their consequences. We now have all the estimates needed to apply Theorem 4.2 to (5.15). By substituting $z = \Delta\pi$ and $\hat{z} = \widehat{\Delta\pi}$, together with the estimates (5.13), (5.4), (5.24), (5.22), and (5.23), into (4.10), we obtain

$$\begin{aligned} \|L^T(\widehat{\Delta\pi} - \Delta\pi)\| &\le \delta_1 \left[(\tau\mathbf{u} + \bar{\epsilon}^{1/2}) + \mu\right]\mu + \delta_1(\bar{\epsilon}^{1/2} + \mu)\mu + (1 + \tau\mu)\delta_{\mathbf{u}} + \tau\mu\delta_{\mathbf{u}} \\ &= \delta_1\mu\left[\tau\mathbf{u} + \bar{\epsilon}^{1/2} + \mu + \mu^{-1}\mathbf{u}\right], \end{aligned} \tag{5.25}$$

and by substituting for τ from (5.17), we obtain

$$\|L^T(\widehat{\Delta\pi} - \Delta\pi)\| \le \delta_1\mu\left[\bar{\epsilon}^{-1/2}\mathbf{u} + \bar{\epsilon}^{1/2} + \mu + \mu^{-1}\mathbf{u}\right]. \tag{5.26}$$

From (5.21), and using orthogonality of Q, we can define

$$v = DA^T(\widehat{\Delta\pi} - \Delta\pi) \tag{5.27}$$

and note from (5.26) that

$$\|v\| = \rho^{-1/2}\|L^T(\widehat{\Delta\pi} - \Delta\pi)\| \le \delta_1\mu^{1/2}\left[\bar\epsilon^{-1/2}\mathbf{u} + \bar\epsilon^{1/2} + \mu + \mu^{-1}\mathbf{u}\right]. \tag{5.28}$$

From (1.1) and (5.16), we see that the right-hand side of this expression is minimized, with a value of $\delta_1\mathbf{u}^{1/2}$, when $\mu \approx \bar\epsilon^{1/2} = \delta_1\mathbf{u}^{1/2}$. This observation suggests that a termination criterion of

$$\mu \le \mathbf{u}^{1/2} \tag{5.29}$$

may be appropriate for the interior-point method. We justify this choice further below, after investigating the errors in the computed step and their effects on maximum steplength and on the updating of the residuals r_c and r_b.

Next, we examine the effect of the error in $\widehat{\Delta\pi}$ and the evaluation error in the right-hand side of (2.15b) on the calculated step $\widehat{\Delta s}$. From (5.4) and (5.24), we have that

$$\|\Delta\pi - \widehat{\Delta\pi}\| \le \|\Delta\pi\| + \|\widehat{\Delta\pi}\| = O(\mu). \tag{5.30}$$

The evaluation error of size $\delta_{\mathbf{u}}$ in the r_c term of (2.15b) (see (5.19)) is significant; the additional roundoff errors of size $\mu\delta_{\mathbf{u}}$ incurred in forming the matrix–vector product and in performing the vector addition to evaluate the right-hand side of (2.15b) are negligible. We conclude from (5.19) and (5.30) that the computed step $\widehat{\Delta s}$ and exact step Δs differ as follows:

$$\Delta s - \widehat{\Delta s} = -r_c + \hat r_c - A^T(\Delta\pi - \widehat{\Delta\pi}) + \mu\delta_{\mathbf{u}} = \delta_1(\mu + \mathbf{u}). \tag{5.31}$$

This estimate is potentially troubling: Since the exact step Δs has size $O(\mu)$, it indicates that the computed step $\widehat{\Delta s}$ may not be correct to any digits at all! This inaccuracy is not so important for the "large" components of s—namely, components in the subvector $s_{\mathcal{N}}$—since eventually μ is small in comparison to these components and errors in $\Delta s_{\mathcal{N}}$ have little effect on the steplength α or on the updated value of x^Ts. However, errors of the size indicated in (5.31) in the $\mathcal{B}$ components of Δs could be disastrous. The consequences could include that the maximum steplength $\alpha_{\max}$ to the boundary could be much smaller than 1 (an argument similar to the one following (5.9) indicates only that $1-\alpha_{\max} = \delta_1$) and in fact we cannot even be sure of decrease in x^Ts along this direction. Fortunately, a refined estimate of the error in $\widehat{\Delta s}_{\mathcal{B}}$ is possible. By using (5.19) in (5.31), we have that

$$\Delta s - \widehat{\Delta s} = -A^T(\Delta\pi - \widehat{\Delta\pi}) + \delta_{\mathbf{u}} = D^{-1}v + \delta_{\mathbf{u}}, \tag{5.32}$$

where v is defined as in (5.27). From (5.2), we have that $D_{ii}^{-1} = \Theta(\mu^{1/2})$ for $i \in \mathcal{B}$, and therefore, by using (5.28), we obtain

$$\Delta s_i - \widehat{\Delta s}_i = \delta_1\mu\left[\bar\epsilon^{-1/2}\mathbf{u} + \bar\epsilon^{1/2} + \mu + \mu^{-1}\mathbf{u}\right], \qquad i \in \mathcal{B}. \tag{5.33}$$

As in the discussion following (5.9), we find that $s_i + \alpha\widehat{\Delta s}_i = 0$ is possible only if

$$1 - \alpha = \delta_1\left[\bar\epsilon^{-1/2}\mathbf{u} + \bar\epsilon^{1/2} + \mu + \mu^{-1}\mathbf{u}\right]. \tag{5.34}$$

Finally, we estimate the errors in the computed step $\widehat{\Delta x}$ obtained from (2.15c) and estimate their effect on $\alpha_{\max}$ and on the updated value of r_b. Again, we consider the components $i \in \mathcal{B}$ and $i \in \mathcal{N}$ separately.

For $i \in \mathcal{B}$, the $\delta_{\mathbf{u}}\mu$ evaluation error in $(r_{xs})_i$ is magnified by the term $s_i^{-1} = \Theta(\mu^{-1})$. Floating-point error in forming the product $x_i\widehat{\Delta s}_i$ and in performing the addition yield additional errors of size at most $\delta_{\mathbf{u}}$, so we obtain

$$\Delta x_i - \widehat{\Delta x}_i = -s_i^{-1}x_i(\Delta s_i - \widehat{\Delta s}_i) + \delta_{\mathbf{u}}, \qquad i \in \mathcal{B}. \tag{5.35}$$

From (5.33) and (5.1), this formula implies that

$$\widehat{\Delta x}_i - \Delta x_i = \delta_1\left[\bar{\epsilon}^{-1/2}\mathbf{u} + \bar{\epsilon}^{1/2} + \mu + \mu^{-1}\mathbf{u}\right], \qquad i \in \mathcal{B}. \tag{5.36}$$

By the usual reasoning, we find that $x_i + \alpha\widehat{\Delta x}_i = 0$ is possible for $i \in \mathcal{B}$ only for α satisfying (5.34).

For $i \in \mathcal{N}$, the $\delta_{\mathbf{u}}\mu$ evaluation error in $(r_{xs})_i$ is not magnified appreciably by the term s_i^{-1} (which has size $\Theta(1)$), and we obtain

$$\Delta x_i - \widehat{\Delta x}_i = -s_i^{-1}x_i(\Delta s_i - \widehat{\Delta s}_i) + \mu\delta_{\mathbf{u}}, \qquad i \in \mathcal{N}. \tag{5.37}$$

By substituting from (5.31) and (5.1), we obtain

$$\widehat{\Delta x}_i - \Delta x_i = \delta_1[\mu^2 + \mu\mathbf{u}], \qquad i \in \mathcal{N}. \tag{5.38}$$

We deduce that $x_i + \alpha\widehat{\Delta x}_i = 0$ for $i \in \mathcal{N}$ only if

$$1 - \alpha = \delta_1[\mu + \mathbf{u}]. \tag{5.39}$$

From (5.34) and (5.39), we conclude that the value of $\alpha_{\max}$ defined by (2.13), with the calculated direction $(\widehat{\Delta x}, \widehat{\Delta \pi}, \widehat{\Delta s})$ replacing the exact search direction, satisfies the estimate

$$1 - \alpha_{\max} = \delta_1\left[\bar{\epsilon}^{-1/2}\mathbf{u} + \bar{\epsilon}^{1/2} + \mu + \mu^{-1}\mathbf{u}\right]. \tag{5.40}$$

Note from (5.30), (5.31), and (5.38) that, in an *absolute* sense, the errors in $\widehat{\Delta \pi}$, $\widehat{\Delta s}$, and $\widehat{\Delta x}_{\mathcal{N}}$ are small. By contrast, the $\mu^{-1}\mathbf{u}$ term in (5.36) implies that the errors in $\widehat{\Delta x}_{\mathcal{B}}$ increase as μ decreases below $\mathbf{u}^{1/2}$. These errors have consequences for the updated values of the residuals r_b and r_c at the new point

$$(x, \pi, s) + \alpha(\widehat{\Delta x}, \widehat{\Delta \pi}, \widehat{\Delta s}),$$

where $\alpha \in (0, \alpha_{\max})$ is the steplength chosen by the algorithm. From (2.7), we see that the computed value of r_c at this new point is given by

$$\text{comp}(\hat{r}_c^+) = A^T(\pi + \alpha\widehat{\Delta \pi}) + (s + \alpha\widehat{\Delta s}) - c + \delta_{\mathbf{u}},$$

where the final term accounts for both cancellation and roundoff errors. From (5.32), we see that this quantity differs from the exact value of r_c^+ by

$$\alpha A^T(\widehat{\Delta \pi} - \Delta\pi) + \alpha(\widehat{\Delta s} - \Delta s) + \delta_{\mathbf{u}} = \delta_{\mathbf{u}},$$

so we conclude that the effect of the errors in $(\widehat{\Delta x}, \widehat{\Delta\pi}, \widehat{\Delta s})$ on the r_c term is minimal (that is, it is of the same order as the cancellation error that arises in any case when this term is evaluated).

The computed version of r_b at the new point is

$$\text{comp}(\hat{r}_b^+) = A(x + \alpha\widehat{\Delta x}) - b + \delta_{\mathbf{u}},$$

which differs from the exact version r_b^+ as follows:

$$\text{comp}(\hat{r}_b^+) - r_b^+ = \alpha A(\widehat{\Delta x} - \Delta x) + \delta_{\mathbf{u}}.$$

By substituting from (5.35) and (5.37) and using (2.14), we obtain

$$\text{comp}(\hat{r}_b^+) - r_b^+ = \alpha A D^2(\Delta s - \widehat{\Delta s}) + \delta_{\mathbf{u}},$$

which, from (5.19) and (5.31) and the estimate $\|D^2\| = O(\mu^{-1})$, yields

$$\text{comp}(\hat{r}_b^+) - r_b^+ = \alpha A D^2 A^T(\widehat{\Delta\pi} - \Delta\pi) + \mu^{-1}\delta_{\mathbf{u}}. \tag{5.41}$$

From (5.21), (3.4), and (3.6), we have that

$$AD^2A^T = \rho^{-1}LL^T = \rho^{-1}(\tilde{L}\tilde{L}^T + E),$$

so by some elementary manipulation, we deduce that $\text{comp}(\hat{r}_b^+) - r_b^+$ equals the expression

$$\alpha\rho^{-1}\tilde{L}\tilde{L}^T(\widetilde{\Delta\pi} - \Delta\pi) + \alpha\rho^{-1}E(\widehat{\Delta\pi} - \Delta\pi) + \alpha\rho^{-1}\tilde{L}\tilde{L}^T(\widehat{\Delta\pi} - \widetilde{\Delta\pi}) + \mu^{-1}\delta_{\mathbf{u}}. \tag{5.42}$$

We bound this expression one term at a time, using results from earlier sections and identifying $\Delta\pi$ with z, $\widehat{\Delta\pi}$ with $\hat{z}$, and $\widetilde{\Delta\pi}$ with $\tilde{z}$. For the first term, we have from (3.10) that $\|\tilde{L}\| \le \delta_1$, while from Theorem 3.4, (5.16), and (5.4), we have

$$\|\tilde{L}^T(\Delta\pi - \widetilde{\Delta\pi})\| = \delta_{\mathbf{u}}^{1/2}\|\Delta\pi_{\mathcal{J}}\| = \mu\delta_{\mathbf{u}}^{1/2}. \tag{5.43}$$

For the second term in (5.42), we have from Lemma 3.2, (5.16), (5.30), and $\rho = \Theta(\mu)$ that

$$\rho^{-1}\|E(\widehat{\Delta\pi} - \Delta\pi)\| \le \delta_{\mathbf{u}}^{1/2}. \tag{5.44}$$

For the third term, recall that $\widetilde{\Delta\pi}_{\mathcal{J}} = \widehat{\Delta\pi}_{\mathcal{J}} = 0$ and $\tilde{L}_{\cdot\mathcal{J}} = 0$, so that

$$\|\tilde{L}\tilde{L}^T(\widetilde{\Delta\pi} - \widehat{\Delta\pi})\| \le \|\tilde{L}\|\|\tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}}(\widetilde{\Delta\pi}_{\bar{\mathcal{J}}} - \widehat{\Delta\pi}_{\bar{\mathcal{J}}})\| \le \delta_1\|\tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}}(\widetilde{\Delta\pi}_{\bar{\mathcal{J}}} - \widehat{\Delta\pi}_{\bar{\mathcal{J}}})\|. \tag{5.45}$$

From (3.9), we have

$$\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\widetilde{\Delta\pi}_{\bar{\mathcal{J}}} = r_{\bar{\mathcal{J}}},$$

and so from (3.5a), (4.3), and (4.4), we have

$$(\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}} + E^{\mathbf{u}}_{\bar{\mathcal{J}}\bar{\mathcal{J}}})\widehat{\Delta\pi}_{\bar{\mathcal{J}}} = \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\widetilde{\Delta\pi}_{\bar{\mathcal{J}}} + (\tilde{L}\tilde{f} + \tilde{e})_{\bar{\mathcal{J}}}.$$

By rearranging, we obtain

$$\tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}}(\widehat{\Delta\pi}_{\bar{\mathcal{J}}} - \widetilde{\Delta\pi}_{\bar{\mathcal{J}}}) = -\tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\left[E^{\mathbf{u}}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\widehat{\Delta\pi}_{\bar{\mathcal{J}}} - \tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\tilde{f}_{\bar{\mathcal{J}}} - \tilde{e}_{\bar{\mathcal{J}}}\right].$$

We now use the estimates

$$\begin{array}{rcll}
\|\tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\| & = & \delta_{\mathbf{u}}^{-1/2} & \text{from (3.45), (5.16), and (5.17),} \\
\|E^{\mathbf{u}}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\| & = & \delta_{\mathbf{u}} & \text{from (4.2),} \\
\|f\| & = & \delta_{\mathbf{u}} & \text{from (4.5), (5.13a), and (5.22),} \\
\|\tilde{e}\| & = & \delta_{\mathbf{u}}^{3/2} + \mu\delta_{\mathbf{u}} & \text{from (4.5), (5.13), (5.22), and (5.23),} \\
\|\widehat{\Delta\pi}_{\bar{\mathcal{J}}}\| & = & O(\mu) & \text{from (5.24)}
\end{array}$$

to yield the following bound:

$$\begin{aligned}
\|\tilde{L}^T_{\bar{\mathcal{J}}\bar{\mathcal{J}}}(\widehat{\Delta\pi}_{\bar{\mathcal{J}}} - \Delta\pi_{\bar{\mathcal{J}}})\| &\leq \|\tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\| \, \|E^{\mathbf{u}}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\| \, \|\widehat{\Delta\pi}_{\bar{\mathcal{J}}}\| + \|\tilde{f}_{\bar{\mathcal{J}}}\| + \|\tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\| \, \|\tilde{e}_{\bar{\mathcal{J}}}\| \\
&\leq \delta_{\mathbf{u}}^{-1/2}\delta_{\mathbf{u}}\mu + \delta_{\mathbf{u}} + \delta_{\mathbf{u}}^{-1/2}[\delta_{\mathbf{u}}^{3/2} + \mu\delta_{\mathbf{u}}] \\
&\leq \mu\delta_{\mathbf{u}}^{1/2} + \delta_{\mathbf{u}}.
\end{aligned}$$

Therefore, for the third term in (5.42), we have from (5.45) that

$$\|\tilde{L}\tilde{L}^T(\tilde{\Delta\pi} - \widehat{\Delta\pi})\| \leq \mu\delta_{\mathbf{u}}^{1/2} + \delta_{\mathbf{u}}. \tag{5.46}$$

By substituting (5.43), (5.44), (5.46), $\rho = \Theta(\mu)$, and $|\alpha| \leq 1$ into (5.42), we have

$$\text{comp}(\hat{r}_b^+) - r_b^+ = \delta_{\mathbf{u}}^{1/2} + \mu^{-1}\delta_{\mathbf{u}}. \tag{5.47}$$

This estimate suggests that the discrepancy between $\hat{r}_b^+$ and its approximation $\text{comp}(\hat{r}_b^+)$ is no greater than $\delta_{\mathbf{u}}^{1/2}$ until μ falls below approximately $\mathbf{u}^{1/2}$. This observation, together with (5.40), suggests strongly that the termination condition (5.29) is the appropriate one. These observations too are illustrated in section 6.

The convergence tolerances used by most interior-point codes—arrived at by practical experience rather than theoretical or analytical considerations—are generally consistent with (5.29). For instance, the code PCx declares optimality if the following three conditions are satisfied:

$$\frac{\|r_b\|}{1 + \|b\|} \leq \texttt{tol}, \qquad \frac{\|r_c\|}{1 + \|c\|} \leq \texttt{tol}, \qquad \frac{|c^T x - b^T \pi|}{1 + |c^T x|} \leq \texttt{tol},$$

where the default value of `tol` is 10^{-8}. Note that $10^{-8} \approx \mathbf{u}^{1/2}$ in double precision arithmetic on most machines.

5.6. Comments and observations. We conclude this section with a few comments about the results above.

Note first that our conclusions can always be defeated by poor scaling of the problem. Poor scaling may show up as imbalance in the size of the components of $x_{\mathcal{B}}$ or $s_{\mathcal{N}}$ (some may be much smaller than others) or as imbalance in the sizes of the nonzero components of the problem data A, b, and c. Difficulties such as these may cause the many factors δ_1 that appear in the analysis to actually be much larger than 1, thereby limiting the regime of applicability of our results and affecting our conclusions about appropriate choices of $\bar{\epsilon}$ and the termination criterion. Most interior-point codes try to avoid these potential difficulties by prescaling the matrix A by some heuristic procedures, for example, the one proposed by Curtis and Reid [2].

A second point concerns the matrix $A_{\cdot\mathcal{B}}$, the basic part of the constraint matrix A. Our analysis is quite general in that it allows $A_{\cdot\mathcal{B}}$ to be rank deficient. However, when the nonzero singular values of this matrix are widely separated, the assumed separation (5.13) between the $p = \operatorname{rank} A_{\cdot\mathcal{B}}$ largest and $m - p$ smallest eigenvalues of AD^2A^T will not appear until μ is very small. This may again limit the regime of applicability of our analysis. Prescaling of the matrix A may help but, in some sense, ill conditioning of this type is intrinsic to the problem. As in many other areas of numerical linear algebra, it is not possible to design algorithms that produce accurate results in finite-precision arithmetic regardless of the conditioning of the problem.

Third, we note that our analysis made no assumption to ensure that **modchol** eventually determines the numerical rank of AD^2A^T. That is, none of our results require that $|\bar{\mathcal{J}}| = p$ for all μ sufficiently small. Although we observed that $|\bar{\mathcal{J}}| = p$ in many numerical tests, the assumptions needed to guarantee this equality are not satisfying in certain respects. (Such assumptions did appear in an earlier version of this paper, but they were discarded.) The advantage of $|\bar{\mathcal{J}}| = p$ in the analysis is that the matrix $\tilde{L}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$ will have all its diagonal elements of size $\Theta(1)$, allowing us to use the estimate $\tau = \delta_1$ in place of the weaker estimate (5.17). This estimate in turn allows us to bound the norm $\|\hat{z}\|$ in (4.10) in terms of $\|z\|$, leading to a more rigorous bound on $\|L^T(\hat{z} - z)\|$.

A fourth, related point concerns our estimate (5.17) of the size of τ, which is based on the assumption that the norm of $\tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}$ can be estimated accurately by observing the sizes of its diagonal elements. While the resulting estimate appears to hold for the vast majority of practical problems of the type in question, there are cases in which it underestimates the value of $\|\tilde{L}^{-1}_{\bar{\mathcal{J}}\bar{\mathcal{J}}}\|$. See Lawson and Hanson [8, p. 31] for a classic example.

Finally, we note that when all the skipped pivots occur in the lower right corner of the matrix M (as happens on most of the smaller problems we tested), we can replace the bound $\|E\| \leq \bar{\epsilon}^{1/2}$ by the tighter bound $\|E\| \leq \bar{\epsilon}$. This tighter estimate allows some of our results to be strengthened, but since we observed some large linear programs in which the skipped pivots were not confined to the lower right corner, we omit a detailed analysis of this case.

6. Implementation and computational results.

The **modchol** approach can be implemented by making minimal changes to a standard sparse Cholesky code. We need to add a loop to calculate the largest diagonal element β, and a small pivot check immediately before the point at which the computation $L_{ii} = \sqrt{M_{ii}}$ is performed. The pivot skipping itself can be performed explicitly (by inserting a column of zeros in the Cholesky factor and maintaining a record of the set $\mathcal{J}$), or it can be "simulated," as in LIPSOL [20] and PCx [3], by inserting a huge element in the pivot position prior to the computation of the column of the Cholesky factor and updating of the remainder of the matrix. In PCx [3], we needed to change fewer than 20 lines of the sparse Cholesky code of Ng and Peyton [10].

To test that the analysis of this paper was reflected in computations, we coded a simple primal-dual interior-point algorithm and applied it to test problems with controlled degeneracy properties. At each iterate, we monitored various quantities, compared them against the estimates of section 5, and confirmed that convergence to a tolerance of approximately $\mathbf{u}^{1/2}$ could be attained even for difficult problems.

Our test problems have the form (2.1), with $m = 6$ and $n = 12$. The matrix A is fully dense, with elements $(\xi_1 - .5)10^{6(\xi_2 - .5)}$, where ξ_1 and ξ_2 are random variables drawn from a uniform distribution on the interval $[0, 1]$. (Of course, the values of ξ_1 and ξ_2 are different for each element of the matrix.) After fixing the number of indices to appear in $\mathcal{B}$, we set

$$|\mathcal{N}| = n - |\mathcal{B}|, \qquad \mathcal{N} = \{1, 2, \ldots, |\mathcal{N}|\}, \qquad \mathcal{B} = \{|\mathcal{N}| + 1, \ldots, n\}.$$

(Note that the problem is degenerate whenever $|\mathcal{B}| \neq 6$.) A primal solution x^* is constructed with

$$x_i^* = 0 \;\; (i = 1, 2, \ldots, |\mathcal{N}|), \qquad x_i^* = 10^{3\xi-1} \;\; (i = |\mathcal{N}| + 1, \ldots, n),$$

where ξ is again randomly drawn from the uniform distribution on $[0, 1]$. We choose the dual solution π^* to be the vector $(1, 1, \ldots, 1)^T$, and fix an optimal dual slack vector s^* to be

$$s_i^* = 10^{4\xi-2} \;\; (i = 1, 2, \ldots, |\mathcal{N}|), \qquad s_i^* = 0 \;\; (i = |\mathcal{N}| + 1, \ldots, n),$$

where ξ is random as above. Finally, we set $b = Ax^*$ and $c = A^T\pi^* + s^*$. Note that by our choice of $\mathcal{B}$, $A_{\cdot\mathcal{B}}$ consists of the last $|\mathcal{B}|$ columns of A. We modified A in some of the problems to introduce various types of rank deficiency.

The code was an implementation of the infeasible interior-point algorithm described by S. J. Wright [16]. The details of this algorithm are unimportant; we need note only that its iterates satisfy the estimates (5.1) in exact arithmetic and that the algorithm takes steps along the affine-scaling direction during its later iterations, provided that these steps make reasonable progress. At each iteration of the algorithm, we calculated the affine-scaling direction (whether or not it was actually used as a search direction) and kept a log of information about this step and about various other properties of the iterates and the **modchol** procedure. The parameter ϵ was set to 10^{-13}, which is about $500\mathbf{u}$ on the SPARCstation 5 that was used for the experiments. The results were not particularly sensitive to this parameter.

Results for various problems are shown in Tables 1, 2, 3, 4, and 5. For each iteration, we tabulate the norms $\|\widehat{\Delta x}^{\mathrm{aff}}\|_\infty$, $\|\widehat{\Delta \pi}^{\mathrm{aff}}\|_\infty$, and $\|\widehat{\Delta s}^{\mathrm{aff}}\|_\infty$ of the affine-scaling step calculated at that iterate, together with the duality measure μ and residual norm $\|(r_b, r_c)\|_\infty$ for that iterate. We also tabulate the number of small pivots encountered during the factorization, that is, the number of elements in $\mathcal{J}$. The step-to-boundary $\alpha_{\max}$ along the calculated affine-scaling direction is also tabulated. (The algorithm actually uses the affine-scaling direction if this parameter exceeds 0.8; otherwise, it uses a direction with a centering component.) A horizontal line in each table indicates the iterate at which termination would occur if we use the termination criterion of section 5.5.

In Table 1 we chose $|\mathcal{B}| = m = 6$, making the linear program nondegenerate and the primal-dual solution unique. Note that the pivot-skipping mechanism in **modchol** is not activated for this problem, since the matrix AD^2A^T is approaching a well-conditioned limit. It is clear from the table that $\widehat{\Delta \pi}^{\mathrm{aff}}$ and $\widehat{\Delta s}^{\mathrm{aff}}$ satisfy the estimates (5.24) and (5.31), respectively, even when the algorithm continues to iterate past the point of normal termination. The component $\widehat{\Delta x}^{\mathrm{aff}}$, on the other hand, clearly shows the influence of the $O(\mu^{-1}\mathbf{u})$ error term in (5.36) when μ falls below $\mathbf{u}$. As discussed in section 5.5, this error is transmitted to the computed residual r_b, destroying the quality of subsequent iterates. A similar deterioration is noted in the steplength $\alpha_{\max}$. These observations show that it is important for the interior-point algorithm to save the best iterate obtained so far, so that it can report this value if it happens to push beyond the appropriate point of termination.

Table 2 shows results for the case of a problem in which $|\mathcal{B}| = 4$ with $A_{\cdot\mathcal{B}}$ full rank, which causes the coefficient matrix in (2.15a) to have four eigenvalues of size $\Theta(\mu^{-1})$ and the remaining two of size $\Theta(\mu)$. The second column shows that **modchol** detects small pivots when μ becomes sufficiently small, and confirms that the quality

Table 1

Affine-scaling step properties for a problem with $m = 6$, $n = 12$, $|\mathcal{B}| = 6$, $\operatorname{rank} A_{\cdot\mathcal{B}} = 6$. $\|\cdot\| = \|\cdot\|_\infty$, and the horizontal line represents the normal point of termination.

	Small		log	log	log	log	
Iteration	pivots	$\log\mu$	$\|(r_b, r_c)\|$	$\|\widehat{\Delta x}^{\rm aff}\|$	$\|\widehat{\Delta \pi}^{\rm aff}\|$	$\|\widehat{\Delta s}^{\rm aff}\|$	$\alpha_{\max}$
⋮							
12	0	−0.6	−11.1	−0.1	−0.6	0.6	.26426
13	0	−1.4	−10.7	0.4	−1.1	0.1	.77520
14	0	−2.1	−10.7	1.2	−2.3	−1.1	.39373
15	0	−3.3	−10.4	−0.3	−1.3	−0.1	.62276
16	0	−4.8	−8.1	−1.1	−5.2	−3.9	.99697
17	0	−7.2	−10.5	−3.5	−8.3	−7.1	.99999
18	0	−12.0	−12.2	−8.2	−14.0	−12.5	>.99999
19	0	−21.0	−12.0	−3.6	−14.9	−13.9	.99975
20	0	−24.2	−4.6	−1.4	−15.0	−13.9	.93989
21	0	−26.2	−1.5	1.4	−15.3	−14.5	.06843
⋮							

Table 2

Affine-scaling step properties for a problem with $m = 6$, $n = 12$, $|\mathcal{B}| = 4$, $\operatorname{rank} A_{\cdot\mathcal{B}} = 4$. $\|\cdot\| = \|\cdot\|_\infty$, and the horizontal line represents the normal point of termination.

	Small		log	log	log	log	
Iteration	pivots	$\log\mu$	$\|(r_b, r_c)\|$	$\|\widehat{\Delta x}^{\rm aff}\|$	$\|\widehat{\Delta \pi}^{\rm aff}\|$	$\|\widehat{\Delta s}^{\rm aff}\|$	$\alpha_{\max}$
⋮							
12	0	−0.6	−12.0	0.1	−1.3	0.7	.95133
13	0	−1.9	−11.4	−1.5	−0.2	1.8	.51719
14	0	−2.4	−9.5	−1.8	−0.9	1.0	.90453
15	1	−3.4	−9.3	−2.7	−5.5	−3.5	.98770
16	2	−5.2	−9.1	−4.4	−7.2	−5.2	.99977
17	2	−8.5	−11.1	−7.7	−10.5	−8.5	>.99999
18	2	−14.4	−13.0	−12.5	−15.8	−14.2	>.99999
19	2	−25.1	−12.3	−1.5	−15.9	−13.7	>.99999
20	2	−29.7	1.2	6.7	−15.9	−13.3	.00016
⋮							

of interior-point steps remains high after this point, at least until an accuracy of $\mathbf{u}^{1/2}$ is achieved. The behavior of the algorithm for very small values of μ—beyond the point of normal termination—is the same as that of Table 1.

The locations of the small pivots detected by **modchol** for the problem reported in Table 2 were at the bottom left of the matrix. We noted earlier that when this is the case, we have that the estimate $\|E\| \le \bar{\epsilon}^{1/2}$ of Lemma 3.2 can be replaced by the stronger estimate $\|E\| \le \bar{\epsilon}$. To show that the algorithm's performance does not depend critically on this smaller value of the error, we modified A to obtain a number of examples in which the small pivots appeared in locations other than the lower right of the matrix. In the problem report in Table 3, we modified the matrix A by replacing all elements in rows 1 and 2 with zeros, except for the element in the last column. We chose $|\mathcal{B}| = 6$, so that the matrix $A_{\cdot\mathcal{B}}$ formed by the last 6 columns of A has rank 5. Moreover, the fact that rows 1 and 2 of A are multiples of each other ensures that the $(2, 2)$ pivot will be flagged as a small pivot in **modchol**. It also implies that the

Table 3

Affine-scaling step characteristics for a problem with $m = 6$, $n = 12$, $|\mathcal{B}| = 6$, $\operatorname{rank} A_{\cdot\mathcal{B}} = 5$ (rows 1 and 2 of A have a single nonzero each, in the same column location). $\|\cdot\| = \|\cdot\|_\infty$, and the horizontal line represents the normal point of termination.

	Small		log	log	log	log	
Iteration	pivots	$\log \mu$	$\|(r_b, r_c)\|$	$\|\widehat{\Delta x}^{\text{aff}}\|$	$\|\widehat{\Delta \pi}^{\text{aff}}\|$	$\|\widehat{\Delta s}^{\text{aff}}\|$	$\alpha_{\max}$
⋮							
11	1	−0.5	−12.6	0.3	1.6	0.8	.23771
12	1	−1.2	−10.3	0.6	1.0	0.2	.81949
13	1	−1.9	−10.3	0.9	0.1	−0.7	.67937
14	1	−2.4	−10.2	1.0	−0.9	−1.7	.50171
15	1	−3.4	−10.2	0.0	−2.3	−3.0	.95044
16	1	−4.7	−9.7	−1.0	−5.0	−5.0	.99199
17	1	−6.8	−11.3	−3.1	−7.1	−7.1	.99991
18	1	−10.9	−10.4	−0.3	−11.2	−11.1	.90487
19	1	−11.9	−10.3	0.3	−12.3	−12.2	.53423
⋮							

Table 4

Affine-scaling step characteristics for a problem with $m = 6$, $n = 12$, $|\mathcal{B}| = 4$, $\operatorname{rank} A_{\cdot\mathcal{B}} = 3$ ($A_{\cdot\mathcal{B}}$ has two dependent columns). $\|\cdot\| = \|\cdot\|_\infty$, and the horizontal line represents the normal point of termination.

	Small		log	log	log	log	
Iteration	pivots	$\log \mu$	$\|(r_b, r_c)\|$	$\|\widehat{\Delta x}^{\text{aff}}\|$	$\|\widehat{\Delta \pi}^{\text{aff}}\|$	$\|\widehat{\Delta s}^{\text{aff}}\|$	$\alpha_{\max}$
⋮							
11	0	−0.4	−12.5	0.2	−0.4	1.1	.86945
12	0	−1.3	−11.2	−0.9	0.6	2.5	.19214
13	0	−1.8	−9.3	−0.9	−3.4	−1.5	>.99999
14	0	−3.8	−11.9	−3.2	−2.3	−0.4	.99848
15	3	−6.7	−9.5	−5.0	−8.0	−6.1	.99999
16	3	−11.8	−12.5	−0.2	−13.1	−11.1	.98866
17	3	−13.8	−12.6	1.9	−13.8	−11.9	.85592
18	3	−14.7	−13.5	−5.3	−13.2	−11.3	.92960
19	3	−15.8	−6.5	−6.5	−13.7	−11.7	>.99999
⋮							

assumption that A has full rank is violated. Table 3 confirms that the quality of the interior-point steps remains high. The algorithm's behavior is qualitatively the same as in the earlier examples.

The results in Table 3 illustrate that, as predicted by the analysis, the use of **modchol** does not cause the interior-point algorithm to break down even when $A_{\cdot\mathcal{B}}$ is rank deficient. We confirm this observation in Tables 4 and 5 with two other experiments involving rank-deficient matrices. Table 4 reports a problem identical to that of Table 2 except that in the matrix A, the third-last column was replaced by a multiple of the second-last column. The matrices A and $A_{\cdot\mathcal{B}}$ are thereby rank deficient. When μ becomes sufficiently small, **modchol** detects a numerical rank of 3 in the matrix of (2.15a), and the interior-point algorithm behaves similarly to that in the earlier tables. In Table 5, the modifications of A used in Tables 3 and 4 were both performed, giving a matrix $A_{\cdot\mathcal{B}}$ of rank 3 such that the pivots are not all confined to the lower right corner of the matrix in (2.15a). (The $(2, 2)$ pivot is always small.) The behavior

Table 5

Affine-scaling step characteristics for a problem with $m = 6$, $n = 12$, $|\mathcal{B}| = 4$, $\operatorname{rank} A_{\cdot\mathcal{B}} = 3$ ($A_{\cdot\mathcal{B}}$ has two dependent columns, and the first two rows of A contain a single nonzero each, in the same column location). $\|\cdot\| = \|\cdot\|_\infty$, and the horizontal line represents the normal point of termination.

Iteration	Small pivots	$\log\mu$	$\log \|(r_b, r_c)\|$	$\log \|\widehat{\Delta x}^{\text{aff}}\|$	$\log \|\widehat{\Delta \pi}^{\text{aff}}\|$	$\log \|\widehat{\Delta s}^{\text{aff}}\|$	$\alpha_{\max}$
⋮							
11	1	−0.7	−10.0	0.3	2.9	2.4	.82144
12	1	−1.4	−9.3	−0.1	2.2	1.7	.85477
13	1	−2.2	−8.6	−0.5	−1.1	0.6	.50951
14	1	−2.5	−9.0	−0.8	−2.9	−1.3	.70461
15	2	−4.5	−10.5	−3.3	−2.0	−1.2	.99889
16	3	−7.5	−6.8	−5.4	−6.2	−4.2	>.99999
17	3	−12.9	−12.1	0.4	−11.9	−9.9	.95922
18	3	−14.3	−12.6	2.0	−13.3	−11.3	.20762
⋮							

is once again similar to that of the earlier tables. We note especially iteration 15, at which two pivots are classified as "small" while a third pivot is slightly greater than the threshold, giving rise to a large spread in the nonzero diagonal elements of $\tilde{L}$. The resulting iterate contains some inaccuracy that manifests itself in a slight increase in the residual r_b, but this is quickly corrected at iteration 16, at which the large and small pivots become clearly separated.

Finally, we note that we tried degenerate test problems in which $|\mathcal{B}| > m$. These are less interesting because **modchol** detects no small pivots in factoring the matrix of (2.15a). Their behavior is once again similar to that of the other test problems, so we omit the details.

Acknowledgments. I am most grateful to the editor and referees. Their constructive comments on the first version of the paper led to considerable improvements, and their extremely close and patient reading of both versions led to the elimination of many infelicities and typos.

REFERENCES

[1] E. D. Andersen and K. D. Andersen, *The MOSEK interior point optimizer for interior programming: An implementation of the homogeneous algorithm*, in High Performance Optimization Techniques, Kluwer Academic Publishers, Norwell, MA, to appear.

[2] A. R. Curtis and J. K. Reid, *On the automatic scaling of matrices for Gaussian elimination*, J. Inst. Math. Appl., 10 (1972), pp. 118–124.

[3] J. Czyzyk, S. Mehrotra, and S. J. Wright, *PCx User Guide*, Technical Report OTC 96/01, Optimization Technology Center, Argonne National Laboratory and Northwestern University, October 1996. Modified March 1997.

[4] I. I. Dikin, *On the speed of an iterative process*, Upravlyaemye Sistemy, (1974).

[5] A. Forsgren, P. Gill, and J. R. Shinnerl, *Stability of symmetric ill-conditioned systems arising in interior methods for constrained optimization*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 187–211.

[6] J. Gondzio, *HOPDM (version* 2.12*): A fast lp solver based on a primal-dual interior point method*, European J. Oper. Res., 85 (1995), pp. 221–225.

[7] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[8] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice–Hall, Englewood Cliffs, NJ, 1974; reprinted as Classics in Appl. Math. 15, SIAM, Philadelphia, 1995.

[9] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *Computational experience with a primal-*

dual interior point method for linear programming, Linear Algebra Appl., 152 (1991), pp. 191–222.

[10] E. Ng and B. W. Peyton, *Block sparse Cholesky algorithms on advanced uniprocessor computers*, SIAM J. Sci. Comput., 14 (1993), pp. 1034–1056.

[11] G. W. Stewart, *On scaled projections and pseudoinverses*, Linear Algebra Appl., 112 (1989), pp. 189–193.

[12] G. W. Stewart and J. Sun, *Matrix Perturbation Theory*, Comput. Sci. Sci. Comput., Academic Press, New York, 1990.

[13] M. J. Todd, *A Dantzig-Wolfe-like variant of Karmarkar's interior-point linear programming algorithm*, Oper. Res., 38 (1990), pp. 1006–1018.

[14] R. J. Vanderbei and J. C. Lagarias, *Dikin's convergence result for the affine-scaling algorithm*, Contemp. Math., (1990).

[15] M. H. Wright, *Some properties of the Hessian of the logarithmic barrier function*, Math. Programming, 67 (1994), pp. 265–295.

[16] S. J. Wright, *A path-following interior-point algorithm for linear and quadratic optimization problems*, Ann. Oper. Res., 62 (1996), pp. 103–130.

[17] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.

[18] S. J. Wright, *Stability of augmented system factorizations in interior-point methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 191–222.

[19] X. Xu, P. Hung, and Y. Ye, *A simplified homogeneous and self-dual linear programming algorithm and its implementation*, Ann. Oper. Res., 62 (1996), pp. 151–172.

[20] Y. Zhang, *Solving Large-Scale Linear Programs by Interior-Point Methods under the MATLAB Enviroment*, Technical Report TR96-01, Department of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, MD, 1996.

SIAM J. Optim.
Vol. 9, No. 4, pp. 1192–1204

THE QUASI-CAUCHY RELATION AND DIAGONAL UPDATING*

M. ZHU[†], J. L. NAZARETH[‡], and H. WOLKOWICZ[§]

Dedicated to Professor John E. Dennis on the occasion of his 60th birthday

Abstract. The quasi-Cauchy (QC) relation is the weak quasi-Newton relation of Dennis and Wolkowicz [*SIAM J. Numer. Anal.*, 30 (1993), pp. 1291–1314] with the added restriction that full matrices are replaced by diagonal matrices. This relation is justified and explored and, in particular, two basic variational techniques for updating diagonal matrices that satisfy it are formulated.

For purposes of illustration, a numerical experiment is described where a diagonal updated matrix with hereditary positive definiteness is used to precondition Cauchy's steepest-descent direction. The resulting QC algorithm is shown to be significantly accelerated.

In the concluding section, the following topics are briefly discussed: additional variational principles, use of diagonal updates within other optimization algorithms together with some further numerical experience (summarized in an appendix), and an interesting connection between QC-diagonal updating and trust-region techniques.

Key words. weak-quasi-Newton, quasi-Cauchy, diagonal updating, Cauchy algorithm, steepest-descent

AMS subject classifications. 49, 90, 65

PII. S1052623498331793

1. Introduction.

We consider the problem of finding a local minimum of a smooth, unconstrained nonlinear function, namely,

$$\text{minimize}_{x\in R^n} \quad f(x). \tag{1}$$

For purposes of discussion, it is useful to identify a hierarchy of relations that can be employed within Newton and Cauchy algorithms for solving (1) (see, for example, Dennis and Schnabel [4], Bertsekas [2], and Nazareth [14] for background):

- Quasi-Newton (QN). $M_+s = y$, where the n-dimensional vector $s = x_+ - x$ denotes the step corresponding to two different points x and x_+, and $y = g_+ - g$ denotes the gradient change corresponding to the gradients g and g_+ at the two points. Assume $s^Ty > 0$. M_+ is a full $n \times n$ matrix that approximates the Hessian of f. Both s and y are used explicitly and $O(n^2)$ storage is required for the matrix M_+.

  If M is chosen to be a positive definite diagonal matrix, say D, then one can recur only the diagonal elements of M_+ in a QN update formula, for example, the BFGS, yielding an updated diagonal matrix D_+. The matrix M_+ is positive definite, and hence D_+ is also positive definite, but obviously D_+ does not satisfy the QN relation. Only $O(n)$ storage is required to store D_+. This diagonal-updating approach is used in Gill and Murray [9] and Gilbert and Lemaréchal [8].

*Received by the editors December 15, 1998; accepted for publication (in revised form) March 5, 1999; published electronically September 24, 1999.
http://www.siam.org/journals/siopt/9-4/33179.html

[†]Microsoft Corporation, 1 Microsoft Way, Redmond, WA 98052 (minzhu@microsoft.com).

[‡]Department of Pure and Applied Mathematics, Washington State University, Pullman, WA (nazareth@amath.washington.edu).

[§]Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada (hwolkowi@orion.math.uwaterloo.ca).

- Weak-quasi-Newton. $s^T M_+ s = s^T y$. This relation was introduced and studied by Dennis and Wolkowicz [5]. For example, one of the updates proposed in [5] is as follows:

$$M_+ = M + \frac{(s^T y - s^T M s)}{(s^T M s)^2} M s s^T M, \tag{2}$$

where M is positive definite. The condition $s^T y > 0$ implies that M_+ is also positive definite. Again s and y are used explicitly and $O(n^2)$ storage is required.

As in the QN case, if M is taken to be a positive definite diagonal matrix D, the foregoing formula (2) can be restricted to updating only the diagonal elements of M_+, yielding a positive definite updated matrix, say D_+. In general, D_+ does not satisfy the weak-QN relation.

It is interesting to note that the quantity $s^T y$ in expression (2), which equals $g_+^T s - g^T s$, can be obtained directly from directional derivative differences along s that require only function values. Thus, knowledge of gradient vectors is not essential in this formula.

- Quasi-Cauchy (QC). $s^T D_+ s = s^T y$, where D_+ is required to be a diagonal matrix, i.e., the QC relation is the weak QN with matrices further restricted to be diagonal. The vectors s and y are assumed to be available. Only $O(n)$ storage is required to store the diagonal update. Additionally, we would like the matrix D_+ to be positive definite and thus able to define a metric. An obvious usage would be to precondition or rescale Cauchy's steepest-descent direction, which accounts for our choice of terminology.

Consider the well-known Oren–Luenberger scaling matrix, namely,

$$D_+ = (s^T y / s^T s) I,$$

where I is the identity matrix. It is interesting to note that this is precisely the unique matrix that would be obtained from the QC relation with the further restriction that the diagonal matrix is a scalar multiple of the identity matrix, i.e., the diagonal elements of the Hessian approximation D_+ are equal and the model function associated with it has contours that are hyperspheres. Thus, scaling matrices derived from the QC relation are a natural generalization of Oren–Luenberger scaling.

As in the foregoing discussion on the weak-QN relation, the quantity $s^T y$ in the right-hand side of the QC relation can be obtained by directional derivative differences along s. Thus, explicit use of gradient vectors can be circumvented, and the resulting diagonal update can find potential use in an algorithm that requires only approximations to gradients (quasi-gradients). The QC relation and variational-based diagonal updating were originally proposed in this setting in [15], [16].

The purpose of this article is to formulate two basic techniques for diagonal updating subject to the QC relation (section 2). These are based on variational principles that are analogous to ones employed in quasi-Newton updating. The first is the analogue of the principle from which the Powell symmetric Broyden (PSB) quasi-Newton update is derived—see, for example, Dennis and Schnabel [4]. Like PSB, the diagonal update does not have the hereditary positive definiteness property. The second is based on a principle analogous to that from which the BFGS update is commonly derived—again, see [4]. Like BFGS, the diagonal update has hereditary positive definiteness and can therefore be used to define a metric. So can its complementary form, which corresponds to DFP.

For purposes of illustration, the latter diagonal update is used to iteratively precondition (or rescale) Cauchy's steepest-descent algorithm, and the results of its numerical performance on a set of standard MINPACK-1 test problems are reported (section 3). The algorithm is shown to be significantly accelerated.

In the concluding section, we briefly discuss further variational principles; the use of diagonal updates within other optimization algorithms, in particular, the L-BFGS algorithm (some additional numerical results are summarized in an appendix); and an interesting connection with trust-region techniques.

More detail can be found in Zhu [19], [20], where a comprehensive theory of diagonal updating subject to the QC relation is developed and applied.

2. QC-diagonal updating.

Suppose $D > 0$ is a positive definite diagonal matrix and D_+, which is also diagonal, is the updated version of D. We require that the updated D_+ satisfy the QC relation and that the deviation between D and D_+ is minimized under some variational principle. (Here we will use only the Frobenius matrix norm to measure the deviation.) We would like the derived update to preserve positive definiteness in a natural way, i.e., we seek well-posed metric problems such that the solution D_+, through the diagonal updating procedure, incorporates available curvature information from the step and gradient changes, as well as that contained in D. As noted earlier, a diagonal matrix uses the same computer storage as a vector so only $O(n)$ storage is required. Thus, the resulting update will have potential use in algorithms where storage is at a premium.

We now focus on two basic forms of QC-diagonal updating.

2.1. Updating *D*.

Consider the variational problem

$$(P):\ \text{minimize}\ ||D_+ - D||_F$$

$$\text{subject to (s.t.)}\ s^T D_+ s = s^T y,$$

where $s \neq 0$, $s^T y > 0$, and $D > 0$. Let

$$D_+ = D + \Lambda, \quad a = s^T D s, \quad b = s^T y. \tag{3}$$

Then the variational problem can be stated alternatively as

$$(P):\ \text{minimize}\ \frac{1}{2}||\Lambda||_F^2$$

$$\text{s.t.}\ s^T \Lambda s = b - a.$$

In (P), the objective is strictly convex and the feasible set is convex. Therefore, there exists a unique solution to (P). Its Lagrangian function is

$$L(\Lambda,\ \mu) = \frac{1}{2}\mathrm{tr}(\Lambda^2) + \mu(s^T \Lambda s + a - b),$$

where μ is the Lagrange multiplier associated with the constraint and tr denotes the trace operator. Differentiating with respect to the diagonal elements, setting the result to zero, and invoking the constraint $s^T \Lambda s = b - a$, we have

$$\Lambda = \frac{b-a}{\mathrm{tr}(E^2)}E, \quad E = \mathrm{diag}\,[s_1^2,\ s_2^2,\ \ldots\ , s_n^2\,], \tag{4}$$

where s_i is the ith element of s. When $b < a$, note that the resulting $D_+ = D + \Lambda$ is not necessarily positive definite.

The foregoing update is the counterpart of the PSB update in the quasi-Newton setting and, like the latter, it does not preserve positive definiteness. Thus it is inappropriate for use within a metric-based algorithm.

2.2. Updating $\boldsymbol{D^{1/2}}$. An alternative approach to preserving positive definiteness through diagonal updating, which is the analogue of the principle used to derive the BFGS update in the quasi-Newton setting, is to update the square root or Cholesky factor $D^{1/2}$ to give the corresponding $D_+^{1/2}$ with

$$D_+^{1/2} = D^{1/2} + \Omega,$$

where Ω is chosen to

$$(5) \qquad (FP): \text{ minimize } \|\Omega\|_F$$

$$\text{s.t. } s^T(D^{1/2} + \Omega)^2 s = s^T y > 0.$$

The foregoing variational problem is well posed, being defined over the closed set of matrices for which the corresponding D_+ is positive semidefinite. Further, analogous to the full matrix case in standard QN updating, it always has a viable solution for which D_+ is positive definite, as we now show in the following theorem.

THEOREM 2.2.1. *Let $D > 0$, $s \neq 0$, and a, b, E be defined as in* (3) *and* (4). *There is a unique global solution Ω of (FP) which is given by*

$$(6) \qquad \Omega = \begin{cases} 0 & \text{if } b = a, \\ -\mu^* E(I + \mu^* E)^{-1} D^{1/2} & \text{if } b \neq a, \end{cases}$$

where μ^* is the largest solution of the nonlinear equation $F(\mu) = b$ and

$$(7) \qquad F(\mu) \overset{\text{def}}{=} s^T(D(I + \mu E)^{-2})s = \sum_{\{i: s_i \neq 0\}} \frac{d_i s_i^2}{(1 + \mu s_i^2)^2}.$$

Proof. In the process of the proof we will see that every expression above is well defined. Let $\Omega = \text{diag}(\omega_1, \ldots, \omega_n)$ and let ω denote the vector of diagonal elements $(\omega_1, \ldots, \omega_n)^T$. First, by a simple transformation, problem (FP) is equivalent to

$$(FP): \text{ minimize } \|\omega\|_2^2 = w^T w$$

$$\text{s.t. } \omega^T E \omega + 2w^T E r = b - a,$$

where

$$r = [d_1^{1/2},\ d_2^{1/2},\ \ldots,\ d_n^{1/2}\ ]^T.$$

When $b = a$, the global optimal solution is obviously $\omega = 0$, and hence $\Omega = 0$, which implies that $D_+ = D$ is positive definite. In the following discussion we assume that $b \neq a$. Problem (FP) has a strictly convex objective with the Hessian E of the constraint being positive semidefinite. By a theorem concerning a quadratic objective with also a quadratic constraint in [12], (FP) has a global solution. Differentiating its Lagrangian

$$L(\omega,\ \mu) = \omega^T \omega + \mu(\omega^T E \omega + 2w^T E r + a - b)$$

with respect to ω, where μ is the Lagrange multiplier, and setting the result to zero, we have

$$\omega_i = -\frac{\mu s_i^2 d_i^{1/2}}{(1+\mu s_i^2)}, \quad i = 1, \ldots, n.$$

Substituting these quantities into the constraint equation, we obtain

$$\begin{aligned} F(\mu) &\overset{\text{def}}{=} s^T(D(I+\mu E)^{-2})s \\ &= \sum_{i=1}^{n} \frac{d_i s_i^2}{(1+\mu s_i^2)^2} \\ &= \sum_{\{i: s_i \neq 0\}} \frac{d_i}{s_i^2(\mu + (1/s_i^2))^2} \\ &= b. \end{aligned}$$

Note that $F(\mu)$ has poles at $(-1/s_i^2)$, $i = 1, \ldots, n$. Let

$$j = \arg\max_{\{i, s_i \neq 0\}} \left(-\frac{1}{s_i^2} \right).$$

The derivative of $F(\mu)$ is

$$\frac{dF(\mu)}{\mu} = -2 \sum_{\{i: s_i \neq 0\}} \frac{r_i^2}{s_i^2(\mu + (1/s_i^2))^3},$$

which is less than zero on the interval

$$\left(-\frac{1}{s_j^2}, +\infty \right),$$

so $F(\mu)$ is strictly decreasing in the above interval from $+\infty$ to 0. Noting that $b > 0$, we see that there is a unique solution μ^* within this interval such that $F(\mu^*) = b$. Although the behavior of $F(\mu)$ is complicated in the entire domain, solutions for $F(\mu) = b$ except μ^* are of no interest. (Note that μ^* is the largest solution.) This is because a necessary condition [12] of the solution of (FP) requires the Hessian of the Lagrangian (with respect to ω), namely, $2(I + \mu E)$, to be positive semidefinite. This is equivalent to

$$1 + \mu s_i^2 \geq 0, \quad i = 1, \ldots, n,$$

and clearly μ^* is the unique solution of $F(\mu) = b$ satisfying the above inequalities. A key observation is that $I + \mu^* E$ is positive definite, and thus μ^* is the unique global minimizer for (FP). Returning to the relationship of ω and μ, we see that

$$\Omega^* = -\mu^* E(I + \mu^* E)^{-1} D^{-1/2}$$

is the unique solution of (FP). Note also that $\forall i = 1, \ldots, n$,

$$d_i^{1/2} - \frac{\mu^* s_i^2 d_i^{1/2}}{(1+\mu^* s_i^2)} = \frac{1}{1+\mu^* s_i^2} d_i^{1/2} \neq 0,$$

so D_+ is positive definite. This completes the proof. $\square$

The following is a direct result of the theorem.

COROLLARY 2.2.1. *The solution D_+ through the diagonal updating problem (FP) is positive definite and unique and is given by*

$$D_+ = \begin{cases} D & \text{if } b = a, \\ (I + \mu^* E)^{-2} D & \text{if } b \neq a. \end{cases} \tag{8}$$

Make the following definitions:

$$U = D^{-1}, \quad c = y^T U y, \quad G = [y_1^2, \dots, y_n^2].$$

One can obtain the update that is complementary to the update in the foregoing theorem by making the following transpositions:

$$\mu \leftrightarrow \nu, \quad s \leftrightarrow y, \quad a \leftrightarrow c, \quad D \leftrightarrow U, \quad D_+ \leftrightarrow U_+.$$

This is summarized in the following result, which is based on the analogue of the variational principle from which the DFP quasi-Newton update is derived.

COROLLARY 2.2.2. *The solution U_+ through the diagonal updating problem complementary to (FP) is positive definite and uniquely given by*

$$U_+ = \begin{cases} U & \text{if } b = c, \\ (I + \nu^* G)^{-2} U & \text{if } b \neq c, \end{cases} \tag{9}$$

where ν^* is the largest solution of $H(\nu) = b$ and

$$H(\nu) \overset{\text{def}}{=} y^T (U(I + \nu G)^{-2}) y = \sum_{\{i : y_i \neq 0\}} \frac{u_i y_i^2}{(1 + \nu y_i^2)^2}.$$

3. Numerical illustration. An immediate application for the diagonal update of the previous section, which we use for purposes of illustration, is to dynamically scale the steepest-descent direction at each iteration of Cauchy's algorithm.

The Cauchy direction is ideal when the contours of the objective f to be minimized are hyperspheres. For a general function that is not quadratic, a preconditioning can be used to make the transformed contours closer to hyperspheres such that the efficiency of the Cauchy direction in the transformed space is enhanced. The diagonal updating is a variable preconditioning which includes the updated curvature information, and its hereditary positive definiteness is naturally maintained when the Cholesky factor is updated as shown in the previous section. An expectation that the Cauchy method will be significantly accelerated using diagonal updating is supported by our numerical results.

Our source code is written in Fortran-90, with double precision algorithmic, running on an ULTRIX DEC workstation. Purely for convenience, we implemented the complementary updates which are defined in terms of the inverse matrix U_+. The numerical experiment is done within the MINPACK-1 testing environment. Test functions are the standard unconstrained problems collected in [11], which we identify by the numbering in Table 1.

We employ a line search routine of Moré and Thuente [13] along direction, say, d, which is based on cubic interpolation and satisfies the (strong) Wolfe conditions:

$$f(x_+) \leq f(x) + \alpha \lambda g^T d, \tag{10}$$

$$|g_+^T d| \leq \beta |g^T d|, \tag{11}$$

Table 1
MINPACK-1 test problems.

Number	Problem name
1	Helical valley function
2	Biggs exp6 function
3	Gaussian function
4	Powell badly scaled function
5	Box 3-dimensional function
6	Variably dimensioned function
7	Watson function
8	Penalty function I
9	Penalty function II
10	Brown badly scaled function
11	Brown and Dennis function
12	Gulf research and development function
13	Trigonometric function
14	Extended Rosenbrock function
15	Extended Powell function
16	Beale function
17	Wood function
18	Chebyquad function

where $x_+ = x + \lambda d$ and the line search parameters are chosen as in [6], namely, $\alpha = 10^{-4}$, $\beta = 0.9$. The stopping criterion is also as in [6]:

$$||g(x)|| \leq 10^{-5} max\{1.0, ||x||\}. \tag{12}$$

At any iterate, say, x_+, the corresponding search direction d_+ in the methods tested is as follows:

1. Standard Cauchy. The search direction is of the form $d_+ = -g_+$.
2. Cauchy with Oren–Luenberger scaling. This scales the search direction with Oren–Luenberger scaling [7] in its complementary form,
$$d_+ = -\frac{y^T s}{y^T y} g_+,$$
for all iterations except the first, where the initial steepest-descent search direction is employed.
3. DU-Cholesky. This implements the complementary diagonal update of Corollary 2.2.2 with $d_+ = -U_+ g_+$. In our numerical implementation, ν^* is obtained by a simple bisectional search within the interval from the largest pole of the function $H(\nu)$ to some large number on the axis such that the initial bisection condition of the endpoints is satisfied. Note that $H(0) = c$, and thus if $b > c$, then the solution $\nu^* < 0$; if $b < c$, then $\nu^* > 0$. Hence, the interval for the bisection is actually reduced with one endpoint being 0 in each case. (The cost of computing ν^* by bisection is a relatively minor portion of the algorithm. Note that more efficient reformulations and techniques, for example, Newton's method, for solving the subproblem for ν^* are possible, as discussed in the concluding section.)

The numerical comparative results are given in Table 2; it gives $nitr/nfg$, namely, the number of iterations and the number of calls for function and gradient evaluation. The symbol $*$ in the table indicates that the method takes too many iterations and is regarded as having failed to converge. The first and second columns in the table are

Table 2
Numerical results for diagonal updating.

Prob.	Dim.	Cauchy	Cauchy-OL	DU-Cholesky
1	3	2552/5229	431/756	370/688
2	6	24041/45488	2221/4353	1165/2120
3	3	2/4	2/6	2/6
4	2	*	*	238/1649
5	3	32535/65075	225/428	165/300
6	6	446/1001	574/877	157/274
6	8	981/2318	269/415	229/427
7	2	14/35	15/20	15/20
8	4	46282/46295	491/1386	491/1386
9	4	63/128	40/61	49/66
10	2	*	147/998	147/998
11	4	*	126/892	198/387
12	3	*	988/2506	*
13	4	76/93	35/46	67/85
13	8	134/169	109/156	80/120
14	2	1109/2248	242/558	289/701
15	4	70638/159377	2853/5081	428/827
16	2	188/377	315/471	104/167
17	4	2879/5795	1755/2347	525/1003
18	4	11/25	16/21	16/20
18	8	118/253	82/128	67/98

the numbers standing for the test problems and the problem dimensions, respectively. The remaining columns are the results for the three corresponding methods.

From the above results, we see that the Cauchy algorithms using diagonal updating are much faster than the standard Cauchy. The simple Oren–Luenberger scaling dramatically improves performance, and the DU-Cholesky diagonal update usually results in a very significant further acceleration.

One can expect similar and quite likely better performance from the diagonal update of Corollary 2.2.1 (whose quasi-Newton counterpart is the BFGS rather than the DFP update).

4. Conclusion.

As noted in section 1, any QN or weak-QN update formula can be converted immediately into a diagonal-updating formula. If the original update has hereditary positive definiteness, then the associated diagonal update will retain this property. The diagonal update does not satisfy any curvature condition a priori, and the approach is therefore heuristic—in particular because a QN update does not maintain a Hessian approximation in an element-to-element sense. Nevertheless, the usefulness of this approach within optimization algorithms, when storage is at a premium, has been nicely demonstrated in the works cited earlier, namely, [8] and [9]. Let us identify it by the name *QN-diagonal updating* (and, correspondingly, *weak-QN-diagonal updating* when derived from a weak-QN formula).

In this article we have developed an alternative, variational-based approach with more solid foundations. QC-diagonal updating is an attractive theory whose appeal arises from its simplicity, its elegant solutions, and the similarity of the variational techniques employed to those of QN methods.

We conclude by briefly itemizing some broader issues involving QC-diagonal updating:

- Additional variational principles. We have used only the Frobenius norm in the variational principles of section 2. Other updates can be derived using weighted Frobenius norms, again with variational counterparts in QN updating. Furthermore, a principle based on the deviation from violation of a previous QC relation can be formulated (analogous to the derivation of the LPD QN update; see Mifflin and Nazareth [10]). For more details, see Zhu [20].

  It is also possible to extend both weak-QN-diagonal updating and QC-diagonal updating along lines that parallel work in Yuan and Byrd [18] by substituting a higher-order estimate of curvature for the quantity b in the right-hand side of the weak-QN and QC relations.
- Other applications. When proposing a new algorithmic technique, it is essential to provide a basic (level 1) numerical illustration of viability. We have done this in section 3 for an obvious application—a diagonally preconditioned Cauchy algorithm applied to a standard set of (low-dimensional) MINPACK-1 problems. A much more detailed study of QC-diagonal updating within the limited-memory BFGS algorithm is given in Zhu [20] using more practical MINPACK-2 problems of high dimension. (Some numerical results from this study are briefly summarized in the appendix.) This study has reaffirmed the usefulness of QC-diagonal updating in this setting, thus paralleling the positive experience with QN-diagonal updating mentioned above. One can also envision using a QC-diagonal update within a conjugate gradient iteration (preliminary results along these lines are also reported in [20]) and within a truncated-Newton method.
- Connections to other techniques. Suppose n is not large and evaluating a function/gradient is relatively expensive (a common assumption in nonlinear optimization). Then the cost of solving the nonlinear equation $F(\mu) = b$ in Theorem 2.2.1, which we call the QC subproblem, is essentially trivial even when it is performed by a crude unidimensional algorithm, for example, using bisection. If greater efficiency is needed, it is useful to exploit a connection between problem (FP) of section 2.2 and a scaled trust-region subproblem as follows. This connection is particularly ironic because the QC method developed in this article is quintessentially *metric-based*, whereas trust-region techniques are the fundamental building blocks of *model-based* approaches (for terminology see Nazareth [14]).

  Write problem (FP) as

$$\text{minimize } \; ||D_+^{1/2} - D^{1/2}||_F$$

$$\text{s.t. } \; s^T D_+ s = b > 0.$$

  Then using the earlier definitions

$$E = \text{diag}\,[s_1^2,\; s_2^2,\; \dots\;, s_n^2\;],$$

$$r = [d_1^{1/2},\; d_2^{1/2},\; \dots,\; d_n^{1/2}\;]^T$$

  and defining the vector z to be the diagonal elements of $D_+^{1/2}$, we can reexpress the variational problem as follows:

TABLE 3
MINPACK-2 test problems.

Number	Problem name	Par.
1	Elastic-Plastic Torsion	0.5D+01
2	Pressure Distribution in a Journal Bearing	0.1D+00
3	(Enneper's) Minimal Surface	0.0D+00
4	Optimal Design with Composite Materials	0.8D-02
5	Steady-State Combustion	0.1D+01
6	Homog. Superconductors: 2-D Ginzburg–Landau	0.2D+01

$$\text{minimize} \quad -r^T z + \frac{1}{2} z^T z \qquad (13)$$
$$\text{s.t.} \;\; z^T E z = b,$$

where $b > 0$. When E is *nonsingular* and the equality in the constraint is replaced by a $\leq$ inequality, one obtains a standard trust-region subproblem in the metric defined by $E > 0$. It is likely that many of the techniques used to solve trust-region subproblems—see, in particular, Rendl and Wolkowicz [17]—can be suitably adapted to the task of solving the QC subproblem more efficiently if desired, based on the above interpretation of (*FP*) as a *nonstandard* trust-region problem (13).

- Convergence analysis. Interesting issues remain to be addressed, in particular, the convergence of algorithms that use diagonal updating, the convergence (or not) of diagonal updates to Hessian matrices of functions when these Hessians are themselves diagonal, and the impact of diagonal updating on finite termination of associated algorithms when applied to strongly convex quadratic functions.

Appendix. Some additional numerical experience with QC-diagonal updating within a limited-memory BFGS algorithm is described briefly in this appendix. We employ the MINPACK-2 testbed—a suite of test problems, each of which comes from a real application and is representative of other commonly encountered problems.

MINPACK-2 contains problems from such diverse fields as fluid dynamics, medicine, elasticity, combustion, molecular conformation, nondestructive testing, chemical kinetics, lubrication, and superconductivity; see Averick et al. [1]. In our experiment, we consider a subset of six MINPACK-2 problems (also employed in the study of Burke and Wiegmann [3]), which are suitable for testing the behavior of unconstrained nonlinear optimization algorithms. They are summarized in Table 3. (The first two are unconstrained versions of constrained problems, and the other four are unconstrained problems.) The last column of the table denotes the default parameters for the corresponding problems as used in our testing. For a complete description of these MINPACK-2 problems, see [1].

We give a numerical comparison of the following two limited-memory BFGS algorithms, which differ only in the choice of diagonal scaling matrix used to initiate the L-BFGS upate at each iteration:

- L-BFGS-OL. The diagonal matrix is obtained in the standard way by Oren–Luenberger scaling $y^T s / y^T y$ (for notation, see section 3).
- L-BFGS-DU(C). The diagonal matrix is obtained by QC-diagonal updating of Cholesky factors.

Table 4
MINPACK-2, $n = 400$.

Prob.	L-BFGS-OL	L-BFGS-DU(C)	Perf.
1	35/39	33/35	+
2	83/89	68/76	+
3	21/23	28/30	−−
4	58/61	49/56	+
5	45/49	45/50	=
6	204/215	175/193	+

Table 5
MINPACK-2, $n = 2{,}500$.

Prob.	L-BFGS-OL	L-BFGS-DU(C)	Perf.
1	89/95	81/84	+
2	185/191	125/162	++
3	77/78	70/71	+
4	230/236	174/201	+
5	120/126	106/108	+
6	480/495	381/423	+

The retention parameter in the two L-BFGS algorithms, i.e., the number m of preserved step/gradient-change pairs over which updating is performed at each iteration, is the standard choice $m = 5$; see Gilbert and Lemaréchal [8]. The line search routine employed is that of Moré and Thuente [13], which was also used in the experiments described in section 3, with its parameters in the strong Wolfe exit conditions (10)–(11) set as follows:

$$\alpha = 10^{-3} \text{ and } \beta = 0.9. \tag{14}$$

For other implementation details, see Zhu [20].

The algorithms used the starting points and stopping criterion of [1] for all tests. Details are again given in [20].

The two limited-memory BFGS algorithms were tested on the MINPACK-2 problems in Table 3 for problems of dimensions 400, 2,500, 10,000, and 40,000; see Tables 4, 5, 6, and 7.

The test results are given in these four tables—each analogous to Table 2—corresponding to the four different choices of problem dimension. Each table reports the results for the two limited-memory BFGS algorithms. The first column records the problem names. Each entry in the second and third columns contains a pair of numbers, namely, the number of iterations and the number of function/gradient calls—the number of times the evaluation routine that returns the function value and gradient vector at a specified point is called—for the corresponding algorithm. The entries in the last column assess relative performance as follows:

=	indicates that the function/gradient counts for the two algorithm are within 5 percent of each other;
+	indicates that the function/gradient count for L-BFGS-DU(C) is better by between 5 and 15 percent;
++	indicates that the foregoing count for L-BFGS-DU(C) is better by more than 15 percent;
−	indicates that L-BFGS-OL is better by between 5 and 15 percent;
−−	indicates that L-BFGS-OL is better by more than 15 percent.

TABLE 6
*MINPACK-*2*, $n = 10{,}000$.*

Prob.	L-BFGS-OL	L-BFGS-DU(C)	Perf.
1	177/188	113/143	++
2	368/387	237/245	++
3	176/182	94/105	++
4	377/385	244/256	++
5	223/230	143/155	++
6	773/802	769/793	=

TABLE 7
*MINPACK-*2*, $n = 40{,}000$.*

Prob.	L-BFGS-OL	L-BFGS-DU(C)	Perf.
1	312/321	319/321	=
2	758/784	710/767	=
3	403/414	449/453	−
4	866/874	1091/1165	−−
5	415/432	312/364	++
6	1444/1502	1360/1405	+

Acknowledgments. We gratefully acknowledge some very useful feedback from the reviewers of this article, which improved it significantly.

REFERENCES

[1] B.M. Averick, R.G. Carter, J.J. Moré, and G. Xue, *The MINPACK-*2 *Test Problem Collection*, Preprint MCS-P153-0692, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1992.

[2] D.P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[3] J.V. Burke and A. Wiegmann, *Notes on Limited Memory BFGS Updating in a Trust-Region Framework*, Preprint, Department of Mathematics, University of Washington, Seattle, WA, 1996.

[4] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[5] J.E. Dennis, Jr. and H. Wolkowicz, *Sizing and least-change secant methods*, SIAM J. Numer. Anal., 30 (1993), pp. 1291–1314.

[6] D.C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Math. Programming Ser. B, 45 (1989), pp. 503–528.

[7] D.G. Luenberger, *Linear and Nonlinear Programming*, 2nd. ed., Addison–Wesley, Reading, MA, 1994.

[8] J.C. Gilbert and C. Lemaréchal, *Some numerical experiments with variable-storage quasi-Newton algorithms*, Math. Programming Ser. B, 45 (1989), pp. 407–435.

[9] P.E. Gill and W. Murray, *Conjugate Gradient Methods for Large-Scale Nonlinear Optimization*, Technical Report SOL 79-15, Department of Operations Research, Stanford University, Stanford, CA, 1979.

[10] R.B. Mifflin and J.L. Nazareth, *The least prior deviation quasi-Newton update*, Math. Programming, 65 (1994), pp. 247–261.

[11] J.J. Moré, B.S. Garbow, and K.E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Software, 7 (1981), pp. 17–41.

[12] J.J. Moré, *Generalizations of the trust region problem*, Optim. Methods Softw., 2 (1993), pp. 189–209.

[13] J.J. Moré and D.J. Thuente, *Line search algorithms with guaranteed sufficient decrease*, ACM Trans. Math. Software, 20 (1994), pp. 286–307.

[14] J.L. Nazareth, *The Newton-Cauchy Framework: A Unified Approach to Unconstrained Nonlinear Minimization*, Lecture Notes in Comput. Sci. 769, Springer, New York, 1994.

[15] J.L. Nazareth, *If quasi-Newton then why not quasi-Cauchy?*, SIAG/OPT Views-and-News, 6 (1995), pp. 11–14.

[16] J.L. Nazareth, *The Quasi-Cauchy Method: A Stepping Stone to Derivative-Free Algorithms*, Technical Report 95-3, Department of Pure and Applied Mathematics, Washington State University, Pullman, WA, 1995.

[17] F. Rendl and H. Wolkowicz, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Math. Programming, 77 (1994), pp. 273–300.

[18] Y. Yuan and R.H. Byrd, *Non-quasi-Newton updates for unconstrained optimization*, J. Comput. Math., 13 (1995), pp. 95–107.

[19] M. Zhu, *Limited Memory BFGS Algorithms with Diagonal Updating*, M.Sc. Project Report, School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, 1997.

[20] M. Zhu, *Techniques for Nonlinear Optimization: Principles and Practice*, Ph.D. dissertation, Department of Pure and Applied Mathematics, Washington State University, Pullman, WA, 1997.