MATHEMATICS OF COMPUTATION, VOLUME 28, NUMBER 125, JANUARY, 1974

A New Algorithm for the Chebyshev Solution of Overdetermined Linear Systems

By Paul T. Boggs

Abstract. Let $x(p)$ be the point which minimizes the residual of a linear system in the l_p norm. It is known that under certain conditions $x(p) \to x^*$, the Chebyshev or l_∞ solution, as $p \to \infty$. A differential equation describing $x(p)$ is derived from which an iterative scheme is devised. A convergence analysis is given and numerical results are presented.

1. Introduction. The purpose of this paper is to develop an algorithm for finding the best solution to an overdetermined system of linear equations. Let

$$Ax = b \tag{1.1}$$

denote the linear system where A is an $(M \times N)$ matrix with $M > N$, $x \in R^N$ (real Euclidean N-space) and $b \in R^M$. Then if

$$r(x) = Ax - b, \tag{1.2}$$

the problem is to find a point $x^* \in R^N$ solving

$$\min_x \|r(x)\|_\infty. \tag{C}$$

The algorithm developed here is based on Pólya's algorithm (see, for example, Cheney [6, Chapter 2]) and the fact that the solution of (C) relies only on a particular set of $N + 1$ equations, called here the critical set. Pólya's idea is to find points $x(p)$ (assumed to be unique) solving the minimization problem

$$\min_x \|r(x)\|_p \tag{L}$$

for a set of values of p_i such that $p_i \to \infty$. It is known (Cheney [6, Chapter 2]) that, if x^* is unique, then $x(p_i) \to x^*$ as $i \to \infty$. If x^* is not unique, then $x(p_i)$ tends to one of the several solutions. In practice, one usually does not have to wait for $\{x(p_i)\}$ to converge since only the critical set of $N + 1$ equations need be determined after which the problem can be solved exactly. For "nice" problems, this critical set of equations is readily apparent for small values of p. (See Goldstein, Levine and Hereshoff [13] for computational aspects of Pólya's algorithm.)

The currently popular methods for obtaining the Chebyshev solution to an overdetermined linear system stem from the simplex method for linear programs. This approach was first advanced by Stiefel [21] and [22]. (See also Cheney [6, Chapter

Received March 27, 1972, revised March 12, 1973.

AMS (MOS) subject classifications (1970). Primary 65F20, 68A10; Secondary 68A10, 68H10.

Key words and phrases. Pólya algorithm, l_p solutions, differential equation for l_p solutions, stable integration techniques.

2].) The idea for Stiefel's exchange algorithm rests on solving the dual of a linear program derived from the original system. Many modifications to Stiefel's original exchange algorithm have been proposed including a stabilizing modification of Bartels and Golub [2] and a revised exchange rule to attempt to handle non-Haar systems by Duris [9] and Duris and Sreedharan [11].

Osborne and Watson [19] very nicely rederive the relationship between the exchange algorithm and the simplex method for solving the dual linear program mentioned above and show that the classical restrictions for computing the solution can be relaxed almost entirely. In particular, they show that the Haar condition is overly restrictive and that the difficulties arising from non-Haar systems can be resolved by using the standard degeneracy handling techniques associated with the simplex algorithm. (See for example Dantzig [8] or Hadley [14].) Barrodale and Young [1] provide ALGOL programs for solving the overdetermined linear systems using a modified simplex algorithm. The exchange algorithms, however, do not usually include such degeneracy handling procedures and thus may cycle when given non-Haar systems. Their advantage is that these extra procedures are not often needed (see Duris [9]) and thus the exchange algorithms are very efficient for this particular problem.

A sufficient condition for the above procedures to work is that the rank of A be N, although Osborne and Watson point out that the simplex algorithm can solve the problem even if the rank of A is less than N. A condition sufficient to guarantee convergence of the algorithms developed here (see Section 3) implies that A has rank N; thus our assumption is stronger. We may be able to relax this condition however, by using a generalized inverse on the right-hand side of (2.3); but this has not been pursued.

Although the linear programming approach appears to have completely solved the problem, Karlovitz [15] gives a procedure for efficiently obtaining solutions to (L) when p is even and, in [16], claims that this procedure is very efficient for obtaining the Chebyshev solution in the continuous case, i.e., for solving L_∞ problems.

The development here is mostly motivated by the similarity of Pólya's algorithm with other schemes involving the use of a parameter. For example, many authors consider the use of a one-parameter imbedding to solve general nonlinear systems. (See, for example, Meyer [17].) In this procedure, a sequence of related problems (one for each value of the parameter) is solved. Under certain continuity assumptions, the solutions of these related problems lie on a differentiable trajectory which is conveniently described by a differential equation and which tends to the solution of the original problem. In Boggs [3], this resulting differential equation is efficiently integrated by a class of weakly A-stable techniques. The problem here, therefore, is to derive the differential equation describing $x(p)$ and then to modify it so as to ensure certain desirable stability characteristics in the solution. In this approach, as in the technique of Karlovitz, the actual solution to problem (L) for the various values of p need not actually be found. The advantages of this approach are that the ideas seem to be directly generalizable to nonlinear systems and that the technique is not at all affected by non-Haar systems and thus no special techniques need be established.

In Section 2, we proceed informally to derive the differential equation describing $x(p)$, establish the basic algorithm and propose some computational modifications.

In Section 3, we cite conditions for the local convergence of these algorithms without regard to the conditions necessary for their derivation. Finally, in Section 4, we comment more specifically on the computational procedures and present some numerical results. These results indicate that our algorithms are potentially competitive with the exchange algorithms but that several problems still remain.

2. Algorithms. The basic differential equation describing $x(p)$ is derived by recalling that $x(p)$ minimizes (L). Thus, since $||r(x)||_p$ is differentiable with respect to (wrt) x, the gradient of $||r(x)||_p$ wrt x is zero at $x(p)$. To facilitate differentiation, we consider

$$\nabla(||r(x)||_p^p) = 0, \qquad p \geqq 1, \tag{2.1}$$

since the minimum of $||r(x)||_p$ must occur at the same point as the minimum of $||r(x)||_p^p$. Thus we obtain from (2.1) that

$$\nabla(||r(x)||_p^p) = p\tilde{A}^T\,|r(x)|^{p-1}$$

and, therefore, $x(p)$ is a solution of

$$\tilde{A}^T\,|r(x)|^{p-1} = 0, \tag{2.2}$$

where $\tilde{A} = (\tilde{a}_{ij})$ is such that

$$\tilde{a}_{ij} = \begin{cases} a_{ij} & \text{if } r_i(x) > 0, \\ -a_{ij} & \text{if } r_i(x) < 0, \end{cases}$$

and $|r(x)|^{p-1}$ is a vector the ith component of which is $|r_i(x)|^{p-1}$. Then, we substitute $x(p)$ into (2.2), differentiate both sides wrt p and solve for $x'(p)$ to obtain

$$x'(p) = -(p-1)^{-1}(A^TEA)^{-1}\,\tilde{A}^Tu \tag{2.3}$$

where $E = \operatorname{diag}(|r_i|^{p-2})$, i.e., a diagonal matrix with the ith diagonal entry being $|r_i|^{p-2}$, and u a vector whose ith component is $|r_i|^{p-1}\ln|r_i|$.

We may now equip (2.3) with an initial value, namely the easily obtained least squares solution

$$x(2) = (A^TA)^{-1}(A^Tb) \tag{2.4}$$

(assuming of course that the columns of A are independent) and consider the numerical integration of the initial value problem (IVP) (2.3), (2.4). Before proceeding further, however, we note that there is an apparent sign ambiguity if one of the residuals has value zero. This is easily resolved by noting that if $r_j = 0$ for some j then $|r_j|\ln r_j$ is also zero since $\lim_{x\to 0^+} x\ln x = 0$. Thus, in the multiplication of A^Tu, the ambiguous elements of A always multiply a zero element in u and hence either choice may be made.

If the columns of A are independent, then the solution $x(p)$ of (L) for each $p \in [2, \infty]$ is unique. This follows from the strict convexity of the l_p norm and, for example, the Unicity Theorem of Cheney [6, p. 23]. Furthermore, since $||Ax - b||_p$ is continuously differentiable, $x(p)$ solves (L) if and only if (2.2) holds. This implies that $x(p)$ is a solution to (2.3). Let $\phi(p)$ satisfy (2.3). Then it follows that $A^T\,|r(\phi(p))|^{p-1} = c$, where c is a constant, for all $p \geqq 2$. The initial condition $\phi(2) = x(2)$ implies that $c = 0$ and thus $\phi(p) = x(p)$ for $p \geqq 2$. Therefore, the IVP (2.3), (2.4) describes the curve $x(p)$. In the sequel however, we shall no longer consider problem (L), but we shall have occasion to consider other differential equations and we will denote their solutions by $x(p)$. (The particular differential equation will be clear from the context.)

To investigate the properties of (2.3), (2.4), we consider first the simplest case, i.e., when $M = N + 1$. In this case,

$$(2.5) \qquad |r_i(x^*)| = \Delta, \qquad i = 1, \cdots, M,$$

where Δ is a nonnegative constant and, if we define f by

$$f(x, p) = -(p-1)^{-1}(A^T E A)^{-1} \tilde{A}^T u,$$

we obtain that $f(x^*, p) = (p-1)^{-1}(\Delta \ln \Delta)(A^T A)^{-1} A^T e$ where $e = (1, 1, \cdots, 1)^T$. In general, $f(x^*, p)$ only goes to zero as $p \to \infty$. Numerical integration of (2.3), (2.4), however, produces a sequence which tends to make the residuals have value one, i.e., the numerical solution tends to an apparent zero of $u(x)$. However, $u(x)$ cannot have a zero unless $\Delta = 1$ and, therefore, the numerical scheme is being fooled by an approximate zero of $u(x)$. We thus consider rescaling the problem so that $\Delta = 1$, thereby moving the true solution to this attractive point. This procedure effectively creates an asymptotically stable solution of the differential equation and allows the use of weakly A-stable methods to efficiently integrate this equation. If $M = N + 1$, and $y \in R^N$ solves (L) for $p = 2$, then

$$\Delta = \sum_{i=1}^{N+1} |r_i(y)|^2 \Big/ \sum_{i=1}^{N+1} |r_i(y)|$$

and dividing A and b by Δ rescales the problem properly. Numerical integration by Euler's method with a step-size of 1 converges to the correct answer very quickly. (This choice of integration technique and step-size is justified later.)

In the case $M > N + 1$, Δ cannot be determined from the least squares solution y. However, we can approximate Δ by

$$(2.6) \qquad \delta(x) = \left(\sum_{i=1}^{N+1} |r_i(x)|^2\right) \Big/ \left(\sum_{i=1}^{N+1} |r_i(x)|\right),$$

rescale the problem as above and attempt the integration. It is possible (see Lemma 3.2) that, if $\delta \neq \Delta$, then convergence to x^* may not be obtained and, therefore, we consider some modifications to periodically rescale the problem.

First, we define

$$(2.7) \qquad f(x, p, \delta) = -\delta(A^T E_\delta A)^{-1} \tilde{A}^T u_\delta,$$

where

$$E_\delta = \operatorname{diag}(|r_i(x)/\delta|^{p-2}) \quad \text{and} \quad (u_\delta)_i = |r_i(x)/\delta|^{p-1} \ln |r_i(x)/\delta|.$$

This function can be derived by dividing A and b by δ and rederiving the differential equation. The factor $(p-1)^{-1}$ was dropped because, in the numerical testing, it tended to decrease the speed of convergence. The δ factors in the residual terms are left to decrease the possibility of overflow or underflow.

The algorithm is then:

1. Find $x_0 = (A^T A)^{-1} A^T b$ (the least-squares solution). Set $i = 0, p_0 = 2$.

2. Compute δ using (2.6) and x_i.
3. Fix $p_{i+1} > p_i$ and integrate

$$x' = f(x, p_{i+1}, \delta), \qquad x(0) = x_i,$$

sufficiently far. (See Theorem 3.5.) Call the result x_{i+1}.

4. Compute $r(x_{i+1})$ and test for correct selection of $N + 1$ equations. If correct, stop; if not, set $i = i + 1$ and go to 2.

In the computer tests of this algorithm (Section 4), we used $p_i = 2^i$. The rescaling procedure in these tests has been the most troublesome aspect and as yet has not been proven to be effective. However, in many cases, it performs well and modifications described in Section 4 have been successful in some problems. Heuristically, the rescaling does the right thing: If the r_i's are greater than one, the square terms dominate and $\delta > 1$; if not, then $\delta < 1$.

The integration is done by Euler's method, with a step-size of one (a choice justified in Section 3) and the "sufficiently far" in step 3 is satisfied when two successive iterates agree to some specified number of places or when a predetermined number of steps have been taken.

In considering the computational effort involved, it is quickly recognized that a system of linear equations must be solved at each step of each integration and that this could be avoided by some suitable approximation to $(A^T E_\delta A)^{-1}$. Several possibilities have been tried with some success.

We consider

(a) $$(A^T E_\delta A)^{-1} \approx (A^T A)^{-1},$$

(b) $$(A^T E_\delta A)^{-1} \approx (A^T E_\delta A)^{-1} \big|_{x=x_i}.$$

In case (b), we mean that, for each i, $(A^T E_\delta A)^{-1}$ is evaluated at the initial condition and not changed throughout the subsequent integration. All of these algorithms are analyzed in Section 3 with the assumption that some rescaling procedure is effective.

3. Convergence Analysis. In the convergence analysis, we consider the original method and the modifications considered in Section 2. The results we obtain are local, meaning that if we once get close enough to x^* and have a sufficiently good approximation to Δ, then the procedure converges to x^*. We are, however, assuming that the rescaling will be effective.

We also show that the $\ln|r_i(x)/\delta|$ can be approximated by two or more terms of its Taylor series around one without sacrificing convergence, thereby allowing a reduction of work at each step. The following definitions will be used throughout this section to simplify the notation:

$$f_1(x, p, \delta) = -\delta(A^T E A)^{-1} \tilde{A}^T u,$$

$$f_2(x, p, \delta) = -\delta(A^T A)^{-1} \tilde{A}^T u,$$

$$f_3(x, p, \delta, x_0) = -\delta(A^T E A)^{-1} \big|_{x=x_0} \tilde{A}^T u,$$

where $E = \operatorname{diag}(|r_i(x)/\delta|^{p-2})$ and $u_i = |r_i(x)/\delta|^{p-1} \ln|r_i/\delta|$. (The definition of E and u is made to eliminate the continual use of the δ subscript as used in Section 2.)

Let $W = \{i: |r_i(x^*)| = \Delta\}$ and let A^* be the matrix made up of the rows A^i for $i \in W$. We assume throughout this section that the columns of A^* are independent. Note that this implies that the columns of A are also independent since A^* contains at least $N + 1$ rows. Then we have

LEMMA 3.1. *Let f be any of f_i, $i = 1, \cdots, 3$. Then*

$$\lim_{p\to\infty} f(x^*, p, \Delta) = 0, \quad \textit{if } f = f_i,\ i = 1, 2,$$

and

$$\lim_{p\to\infty} f_3(x^*, p, \Delta, x^*) = 0.$$

Proof. We have that

$$|r_i(x^*)/\Delta| \begin{cases} = 1, & i \in W, \\ < 1, & i \notin W. \end{cases}$$

Thus, $u_i = 0$ for $i \in W$ and $u_i = |r_i(x^*)/\Delta|^{p-1} \ln|r_i(x^*)/\Delta|$, $i \notin W$. But $\lim_{p\to\infty}|r_i(x^*)/\Delta|^{p-1} \ln|r_i(x^*)/\Delta| = 0$ so that the vector $u \to 0$ as $p \to \infty$. Thus, for $f = f_2$, the result is established. For $f = f_1$ or $f = f_3$, we note that, from the definition of E, $\lim E = \operatorname{diag}(z_i)$, where $z_i = 1$, $i \in W$ and $z_i = 0$, $i \notin W$. Thus, for $x = x^*$ and $\delta = \Delta$, $\lim_{p\to\infty}(A^{*T}A^*)^{-1}$ and the result follows. Q.E.D.

Lemma 3.1 shows that, if the problem is properly scaled, then it is at least possible that the solution $x(p)$ of $x' = f(x, p, \Delta)$ to be such that $x(p) \to x^*$ as $p \to \infty$.

Our algorithm, however, calls for a sequence $\{\delta_j\}$ of rescaling factors such that $\delta_j \to \Delta$ as $i \to \infty$. If $\delta_j < \Delta$, then $|r_i(x^*)/\delta_j| > 1$, $i = 1, \cdots, M$, and the $\lim_{j\to\infty} f(x^*, p_j, \delta_j)$ may not be zero. However, if $|r_i(x^*)/\delta_j|^{p-1}$ remains bounded as $j \to \infty$ (which it will if $\delta_j \to \Delta$ fast enough) then $\lim_{j\to\infty}[|r_i(x^*)/\delta_j|\cdot\ln|r_i(x^*)/\delta_j|] = 0$. If $\delta_j > \Delta$, then $|r_i(x^*)/\delta| < 1, i = 1, \cdots, M$, and $|r_i(x^*)/\delta_j|^{p-2}, i = 1, \cdots, M$, may tend to zero if δ does not tend to Δ fast enough. In this case, (with $x = x^*$) $\lim_{p\to\infty} A^TEA = 0$, the null matrix, and numerical problems will certainly arise. If $|r_i(x^*)/\delta_j|^{p-2} \to 1$, $i \in W$, then we have shown

LEMMA 3.2. *Assume the conditions of Lemma 3.1. Let the sequence $\{\delta_j\}$ be chosen such that $\delta_j \to \Delta$ fast enough so that $\lim_{j\to\infty}|r_i(x^*)/\delta_j| = 1$, $i \in W$. Then*

$$\lim_{j\to\infty} f(x^*, p_j, \delta_j) = 0$$

with the obvious modification if $f = f_3$.

Uniqueness results for the Chebyshev solution of overdetermined linear systems are known (see for example, Cheney and Goldstein [7]). We remark here that, if the problem is scaled correctly wrt an isolated solution x^*, then, for $x \neq x^*$, $\lim_{j\to\infty} f(x, p_j, \Delta) \neq 0$ since, for at least some i, $|r_i(x)| > 1$. If, however, the problem is not scaled correctly and $\delta > \Delta$, then $\lim_{j\to\infty} f(x, p_j, \delta)$ may be zero. This remark provides the basis for choosing a lower bound for Δ, but Lemma 3.2, also points out the difficulty here.

One final lemma is needed before convergence can be shown.

LEMMA 3.3. *Assume the conditions of Lemma 3.2. Let $f = f_1$ or f_3 and let $x_0 = x^*$ for $f = f_3$. Then, if f_x is the matrix of first partial derivative wrt x, $\lim_{j\to\infty} f_x(x^*, p_j, \delta_j)$ is a matrix all of whose eigenvalues lie in the left half-plane.*

Proof. For $f = f_1$, we have

$$f_x = -\delta \frac{\partial}{\partial x} [(A^T E A)^{-1}] \tilde{A}^T u - \delta (A^T E A)^{-1} \tilde{A}^T \frac{\partial}{\partial x} u.$$

Now $\partial u/\partial x = (1/\delta) D(p) \tilde{A}$ where

$$D(p) = \text{diag}[|r_i/\delta|^{p-2} (1 + (p-1) \ln|r_i/\delta|)]$$

and

$$\lim_{j\to\infty\,;\,\delta=\Delta} D(p_j) = D = \text{diag}(z_i) \quad \text{where } z_i = \begin{cases} 1, & i \in W, \\ 0, & i \notin W. \end{cases}$$

Thus, since $\lim_{j\to\infty\,;\,\delta=\Delta} u = 0$ and $\lim_{j\to\infty\,;\,\delta=\Delta} E = D$, we have

$$\lim_{j\to\infty} f_x(x^*, p_j, \Delta) = -\Delta (A^T E A)^{-1} ((1/\Delta) \tilde{A}^T D \tilde{A}) = -I$$

which is clearly negative definite. For $f = f_3$, the result follows in the same way. Q.E.D.

For $f = f_2$,

$$f_x = -\delta (A^T A)^{-1} \tilde{A}^T D(p) \tilde{A}$$

and

$$\lim_{j\to\infty} f_x(x^*, p_j, \Delta) = -\Delta (A^T A)^{-1} ((1/\Delta) \tilde{A}^T D \tilde{A})$$
$$= -(A^T A)^{-1} (A^{*T} A^*).$$

Therefore, if the eigenvalues of $-(A^T A)^{-1}(A^{*T}A^*)$ lie in the left half-plane, then the conclusion of Lemma 3.3 holds for $f = f_2$. If not and the algorithm is not converging, then one may evaluate $(A^T E A)^{-1}$ at the current point and continue. For x and δ close enough to x^* and Δ, respectively, a continuity argument will show that the eigenvalues of $-(A^T E A)^{-1}(A^{*T}A^*)$ will lie in the left half-plane. More will be said on this subject in Section 4.

Remarks. 1. We note here that, for f as in Lemma 3.3, the matrices will, by continuity, have all their eigenvalues in the left half-plane for p sufficiently large and for δ sufficiently close to Δ.

2. If we let $u = (r_i(x)/\delta)^{p-1} T_n(r_i(x)/\delta)$ where $T_n(s)$ is the first n terms of the Taylor series expansion for $\ln(s)$ around one, we then can compute

$$\frac{\partial}{\partial x} u = (1/\delta) D(p) \tilde{A} \quad \text{where } D(p) = \text{diag}[|r_i/\delta|^{p-1} T_n' + T_n (p-1) |r_i/\delta|^{p-2}].$$

Now since $T_n(1) = 0$ and $T_n'(1) = 1$ for $n \geqq 2$, we see that this approximation can be used without sacrificing convergence.

For the main theorem, we need to compute a Lipschitz constant for f_x over an appropriate region in $R^N \times R \times R$ containing the point (x^*, ∞, Δ). Specifically, we need a constant K independent of p and δ such that for p sufficiently large and $|\delta - \Delta|$ sufficiently small

$$||f_x(x^*, p, \delta) - f_x(x, p, \delta)|| \leqq K \, ||x - x^*||.$$

This can be done by bounding the second partial derivative, but this is not worth pursuing since it is easy to convince oneself that such a bound can be computed and the bound so obtained is too crude to be of any computational value.

THEOREM 3.4. *Assume the conditions of Lemma* 3.3. *Let K be the Lipschitz constant for f_x. Then, for p sufficiently large, $|\delta - \Delta|$ sufficiently small, and $||x_0 - x^*||$ sufficiently small, any solution $x_p(t)$ of*

$$x_p'(t) = f(x, p, \delta), \qquad x(0) = x_0 \tag{3.1}$$

is such that

$$\limsup_{t\to\infty} ||x_p(t)|| \leqq c\, ||f(x^*, p, \delta)||,$$

where c is a constant independent of p.

Proof. We may write

$$\begin{aligned} f(x, p, \delta) &= f(x^*, p, \delta) + f_x(x^*, p, \delta)(x - x^*) \\ &\quad + [f(x, p, \delta) - f(x^*, p, \delta) - f_x(x^*, p, \delta)(x - x^*)]. \end{aligned} \tag{3.2}$$

By the remark following Lemma 3.3, p may be chosen large enough so that $f_x(x^*, p, \delta)$ has its eigenvalues in the left half-plane. Using Ortega and Rheinboldt [18, Theorem 3.2.12], the bracketed term in (3.2) may be bounded by $1/2K||x - x^*||^2$ and thus (3.1) satisfies the conditions of Boggs and Dennis [4, Theorem 1.1]. Therefore, the result holds for a constant c bounded by $2/|\sigma(p, \delta)|$ where $\sigma(p, \delta)$ is the eigenvalue of $f_x(x^*, p, \delta)$ which is smallest in absolute value. Let $\sigma(p_j, \delta_j)$ be the eigenvalue of smallest absolute value of $f(x^*, p_j, \delta_j)$. Then, clearly, $\sigma(p_j, \delta_j) \to \sigma^*$ as $j \to \infty$ where σ^* is the eigenvalue of smallest absolute value of $f_x(x^*, \infty, \Delta)$. Therefore, c may be bounded by $2/\min_{j>j_0}(|\sigma(p_j, \delta_j)|)$ where j_0 is sufficiently large. Q.E.D.

The convergence result is now

THEOREM 3.5. *Let x^* be an isolated solution of* (C). *Choose a sequence p_j, $j = 0, 1, \cdots$, such that $p_j \to \infty$ and p_0 is large enough to satisfy Theorem* 3.4. *Let x_j be any point on the solution curve of $x_p' = f(x, p_j, \delta_j)$ satisfying $||x_j - x^*|| < c||f(x^*, p_j, \delta_j)||$. Then $x_j \to x^*$ as $i \to \infty$.*

Proof. The result follows immediately from Lemma 3.2 and Theorem 3.4. Q.E.D.

To complete the analysis, we must now show that the numerical solutions of the IVP's have the same asymptotic characteristics as the actual solutions. This of course will depend on the method used, but if any weakly A-stable method is used (Boggs [3]), the numerical solutions will mirror the actual solution and numerical convergence will be obtained. We state the result as follows:

THEOREM 3.6. *If a method and a step-size h are chosen so that the method is weakly A-stable wrt the problem $x' = f(x^*, \infty, \Delta)$ and if y_i is the sequence generated by that method applied to* (3.1), *then the*

$$\limsup_{i\to\infty} ||y_i - x^*|| < c'\, ||f(x^*, p, \delta)||$$

where c' may be chosen independent of p and δ, but may depend on the method.

Before considering some numerical examples, we note that, for f_1, a choice of h making Euler's method weakly A-stable as convergence is neared is $h = 1$. Euler's method is chosen because of its low order and hence its rapid convergence near the solution (see Boggs [3]). These statements justify our choice of algorithms for the numerical examples.

4. Numerical Results. In this section, we describe the results of our algorithms applied to several examples and attempt some comparison with other methods. We remark at the outset, however, that examples can be chosen to make any method look better than other methods. For example, the algorithms of Section 2 usually determine the critical set very quickly when the residuals corresponding to the non-critical equations are much smaller in magnitude than Δ. Exchange algorithms work best when a fortunate choice of the initial reference set is made. Further, we note that exchange algorithms frequently begin by inverting an $N \times N$ matrix and that our algorithms begin by solving an $N \times N$ linear system. Thus, both approaches give rise to algorithms which are essentially $O(N^3)$. The examples presented therefore, were chosen to illustrate the behavior of our algorithms and no attempt was made to optimally choose the parameters involved. It appears that much more experience with the procedure is needed before such choices can be made.

The programs are written in G-level FORTRAN IV for the IBM 360 and run under the ALPHA time-sharing system. Since the procedures are iterative and since we need only determine the critical set of equations, we can afford to use only single precision arithmetic and Gaussian elimination (rather than the Businger-Golub [5] approach) to solve the linear systems involved. Therefore, we may save the L-U decomposition and hence save considerable effort for some of the algorithms. (Once the critical set is determined, one could certainly use more accurate methods to obtain the final values.)

As mentioned earlier, the choice of rescaling factors has been the most troublesome aspect of the development. We now derive upper and lower bounds for Δ and then comment on an experimentally determined modification which has worked well in practice.

THEOREM 4.1. *For problem* (C) *and* Δ *defined by* (2.5) *the bounds*

$$\sum_{i=1}^{M} r_i^2(y) \Big/ \sum_{i=1}^{M} |r_i(y)| \leqq \Delta \leqq \sum_{i=1}^{M} r_i^2(y) \Big/ \sum_{i=1}^{N+1} |r_i(y)|$$

are valid where y *is the least squares solution and the residuals* $r_i(y)$ *are ordered so that* $|r_1(y)| \geqq |r_2(y)| \geqq \cdots \geqq |r_M(y)|$.

Proof. The points $r(x)$ lie on the hyperplane

$$H = \left\{ v : \langle v, r(y) \rangle = \langle r(y), r(y) \rangle = \sum_{i=1}^{M} r_i^2(y) \right\} \quad \text{(see Cheney [6])}.$$

Thus, for x^*, we have

$$\langle r(y), r(y) \rangle = \langle r(x^*), r(y) \rangle \leqq \Delta \sum_{i=1}^{M} |r_i(y)|$$

and the lower bound is established. Also,

$$\langle r(y), r(y) \rangle = \langle r(x^*), r(y) \rangle \geqq \Delta \sum_{i=1}^{N+1} |r_i(y)|$$

and the upper bound is established. Q.E.D.

Although the number δ, computed by (2.6) using y (the least squares solution), is not in general an upper bound for Δ, it has usually been so in practice. It has also usually been a rather close approximation to Δ (usually within about 10 percent). We have, however, noted that an under-estimation of Δ generally gives faster convergence although at the risk of some instability. Therefore, for the first step, we used

$$\delta_0 = \delta(y)/W$$

where W is a constant greater than 1, usually 1.1, and $\delta(y)$ is computed by (2.6). Thereafter, we compute

$$\delta_i = \begin{cases} \delta(x_i) & \text{if } \delta(x_i) \leqq \delta(y), \\ \delta_0 & \text{if } \delta(x_i) > \delta(y). \end{cases}$$

This procedure has proved useful in practice.

Example 1 (*Cheney* [6, *p*. 44]). This example was chosen mostly for its simplicity and used primarily to debug the programs.

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 2 \\ 2 & 4 \\ 2 & 1 \\ 3 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} 3 \\ 1 \\ 7 \\ 11.1 \\ 6.9 \\ 7.2 \end{pmatrix}.$$

The Haar condition is clearly violated here as row 4 of A is twice row 3. The solution is at $x_1 = x_2 = 2$ with $\Delta = 1$, but our rescaling procedure is used anyway since this fact is not known prior to execution. The results are summarized in Table 1 using the following notation:

Algorithm:

1. $x' = -\delta(A^TEA)^{-1}\tilde{A}^Tu$,
2. $x' = -\delta(A^TA)^{-1}\tilde{A}^Tu$,
3. $x' = -\delta(A^TEA)^{-1}|_{x=x_0}\tilde{A}^Tu$,
4. Algorithm 1 with u replaced by v,
5. Algorithm 2 with u replaced by v,
6. Algorithm 3 with u replaced by v,

where $v_i = |r_i/\delta|^p\, T_3(|r_i/\delta|)$ where $T_3(x)$ is the first three terms of the Taylor series for $\ln x$.

In each case, the equation corresponding to a fixed p_i is integrated by Euler's method with a step-size of 1. The integration was halted when two successive iterates agreed to five significant figures, i.e., when their difference was less than 5×10^{-5} and the function values were less than 5×10^{-5} or when 10 iterations were completed. The p_i's chosen were $p_i = 2^i$, $i = 0, 1, \cdots$. Convergence of the process occurred when the correct answer was obtained to five decimal places. The number of iterations to correct selection is the total number of iterations needed for all the p_i's until the equations corresponding to largest $N + 1$ residuals comprise the critical set. The δ_i's are computed by the procedure described above with $W = 1.1$.

TABLE 1

Algorithm	Correct selection			Convergence		
	No. Iter.	Final i	No. Mults.	No. Iter.	Final i	No. Mults.
1	5	0	411	36	7	4776
2	5	0	411	>62	>10	>6337
3	5	0	411	39	7	4070
4	5	0	291	37	7	4011
5	5	0	291	>62	>10	>4851
6	5	0	291	40	7	3190

Note that, for $i = 0$, Algorithms 1, 2 and 3 reduce to the same algorithm and Algorithms 4, 5 and 6 also reduce to the same algorithm.

It is clear that the number of iterations do not always give an accurate measure of the work involved. Thus, it was decided to give the total number of multiplications needed in each case. The formulas used to count the multiplications are derived from the procedure of saving the LU decomposition of the matrix to be inverted. (See Forsythe and Moler [12].)

Algorithms 2 and 5 had not converged when we halted the program at $i = 10$, but four significant figures had been obtained. We note that the most efficient algorithm is Algorithm 6 and that the approximation of $\ln(r_i/\delta_i)$ by a quadratic resulted in about a 20% reduction of work over corresponding algorithms not using the approximation.

Example 2 (*Duris* [9]). This example satisfies the Haar condition and was easily solved by Duris' exchange rule in three iterations. It should be noted, however, that the initial reference set used by Duris included three of the five equations in the critical set. Clearly, Duris' rule required the fewest number of exchanges possible for this problem from the initial reference set. The approximate number of multiplications needed for Duris' algorithm is 660.

$$A = \begin{bmatrix} 1 & -3 & 9 & -27 \\ 1 & -2 & 4 & -8 \\ 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ -3 \\ -2 \\ 0 \\ 7 \\ -1 \\ 5 \\ 2 \end{bmatrix}.$$

This example is a good instance of the type of problem which is ill-conditioned with respect to our algorithms. None of the six algorithms was able to obtain the correct selection although at $i = 12$, they all had obtained two significant figures of accuracy in each component of the solution. The problem is that A is somewhat ill-conditioned containing, as it does, a subset of the Vandermonde matrix and, more importantly, the residuals at the solution corresponding to the noncritical set are very close in magnitude to the Chebyshev residual Δ. To demonstrate that this indeed is the case, we modified the b vector in such a way as to maintain the same solution but to cause Δ to be relatively larger than the remaining residuals. We chose $b = (4, -3, -3, 0, 8, -2, 5, 3)^T$.

The results are summarized in Table 2 which is organized exactly as Table 1.

TABLE 2

Algorithm	Correct selection			Convergence		
	No. Iter.	Final i	No. Mults.	No. Iter.	Final i	No. Mults.
1	7	0	1062	36	8	10604
2	7	0	1062	59	8	8958
3	7	0	1062	45	8	8198
4	8	0	930	40	9	10506
5	8	0	930	61	8	7302
6	8	0	930	47	9	7240

We first note that the best we could do is about 50% more work than the Duris algorithm. However, Stiefel's algorithm required six iterations or approximately 1200 multiplications so that Algorithm 6 is about 25% faster than Stiefel's algorithm.

Convergence is clearly not necessary before the correct selection at $i = 0$ is obtained. A check of the iterates reveals that the correct choice is made in three iterations or 470 multiplications for Algorithms 4, 5 and 6. Thus, by limiting the number of iterations for each value of i to three, we obtain a faster algorithm than Duris. However, such a drastic reduction in the number of iterations allowed for each i could cause trouble for problems for which correct selection is obtained only for larger values of i. This problem did occur in some examples run but not reported here.

Example 3 (*Duris* [9]).

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ -1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ -1 \\ 0 \\ -1 \\ 1 \\ 0 \\ 2 \\ 3 \\ -3 \\ -2 \end{pmatrix}$$

This system clearly violates the Haar condition. Duris' algorithm solved the problem in six iterations from the initial reference set (2, 3, 4, 5, 6, 8) which includes two of the six critical equations. The approximate number of multiplications required is 1895. The results of our algorithm are recorded in Table 3.

TABLE 3

Algorithm	Correct Selection			Convergence		
	No. Iter.	Final i	No. Mults.	No. Iter.	Final i	No. Mults.
1	20	1	7182	—	—	—
2	20	1	3862	—	—	—
3	20	1	4194	—	—	—
4	20	1	6382	—	—	—
5	20	1	3062	—	—	—
6	20	1	3394	—	—	—

Here we note that Algorithm 5 requires about 50% more work than Duris' algorithm. Again, by limiting the number of iterations allowed for each value of i to five, we obtain correct selection in 10 iterations ($i = 1$) and 1712 multiplications for Algorithm 5. Table 3 indicates that convergence is not obtained to our specified accuracy. This is caused by the inability of our scaling procedure to generate sufficiently good approximations to Δ. Convergence to two decimal places in each component of x is obtained at $i = 3$ after which the algorithms diverge.

Example 4 (*Goldstein, Hereshoff & Levine* [13]). The problem is to find the best approximation to x^{10} by a polynomial of degree 9 at 21 equally spaced points on the interval $[-1, 1]$. This results in a 21×10 system. Because the system is large, the computing time is also large and only Algorithm 6 was run. Correct selection was made in 10 iterations ($i = 0$) or 6994 multiplications. (In [13], it is reported that Pólya's algorithm did not give correct selection for the least 16th fit.) We were not able to obtain convergence to five places; Algorithm 6 yielded three significant figures in 36 iterations ($i = 3$) after which the procedure diverged. Several experiments with the value of W were run, but the best results yielded only four place accuracy before divergence.

5. Conclusions. We have shown that some integration techniques for a differential equation based on Pólya's algorithm are potentially competitive methods for obtaining the Chebyshev solution of an overdetermined linear system. Our tests have included only a very few of the possible algorithms so derivable and more tests are necessary in order to determine which algorithms are best in the sense of being most efficient or most stable. For example, the efficiency could be improved by using an algorithm which only computed the LU decomposition of A^TEA when necessary, say when convergence is not fast enough for some value of i. Also, as convergence is neared (p large), the numbers $|(r_i/\delta)|^{p-1}$, for i corresponding to noncritical equations, tend to cause underflows and thus can be set to zero. This means that at this point the ith equation can be eliminated and the subsequent work reduced.

An important addition to the program would be a scaling procedure which monitors the iterates and rescales automatically to maintain stability. Again, more testing would be necessary to install such a procedure. Also, more testing clearly needs to be done to determine the optimal criteria for terminating the integration for each value of i.

As a final remark, we note that the reference set corresponding to the final value of x at $i = 0$ often contains most of the critical indices and thus could be used as an excellent initial reference set for the exchange algorithms.

6. Acknowledgments. The author would like to thank Professor J. E. Dennis, Jr. of Cornell University for suggesting research along these lines and Professors H. McLaughlin, J. McLaughlin, and J. Ecker for useful discussions concerning this work. He would also like to thank the referee for his careful reading of the manuscript and helpful suggestions for the revision.

Department of Mathematics
Rensselaer Polytechnic Institute
Troy, New York 12181

1. I. Barrodale & A. Young, "Algorithms for best L_1 and L_∞ linear approximation on a discrete set," *Numer. Math.*, v. 8, 1966, pp. 295–306. MR **33** #5096.

2. R. Bartels & G. Golub, "Stable numerical methods for obtaining the Chebyshev solution to an overdetermined system of equations," *Comm. ACM*, v. 11, 1968, pp. 401–406. MR **39** #2302.

3. Paul T. Boggs, "The solution of nonlinear systems of equations by A-stable integration techniques," *SIAM J. Numer. Anal.* v. 8, 1971, pp. 767–785. MR **45** #6179.

4. Paul T. Boggs & J. E. Dennis, "Error bounds for the discretized steepest descent and the discretized Levenberg-Marquardt algorithms" (In preparation).

5. P. Businger & G. H. Golub, "Handbook series linear algebra. Linear least squares solutions by Householder transformations," *Numer. Math.*, v. 7, 1965, pp. 269–276. MR **31** #862.

6. E. W. Cheney, *Introduction to Approximation Theory*, McGraw-Hill, New York, 1966. MR **36** #5568.

7. E. W. Cheney & A. A. Goldstein, "Newton's method for convex programming and Tchebyscheff approximation," *Numer. Math.*, v. 1, 1959, pp. 253–268. MR **22** #316.

8. G. B. Dantzig, *Linear Programming and Extensions*, Princeton Univ. Press, Princeton, N.J., 1963. MR **34** #1073.

9. C. Duris, "An exchange method for solving Haar or non-Haar overdetermined linear equations in the sense of Chebyshev," in *Proceedings*-1968 *ACM National Conference*, 1968, pp. 61–66.

10. C. Duris, *An Algorithm for Solving Overdetermined Linear Equations in the l^p Sense*, Math. Report No. 70–10, Drexel University, 1970.

11. C. Duris & V. P. Sreeduaran, "Chebyshev and l^p-solutions of linear equations using least squares solutions," *SIAM J. Numer. Anal.*, v. 5, 1968, pp. 491–505. MR **38** #4011.

12. G. Forsythe & C. B. Moler, *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1967. MR **36** #2306.

13. A. A. Goldstein, N. Levine & J. B. Hereshoff, "On the 'best' and 'least qth' approximation of an overdetermined system of linear equations," *J. Assoc. Comput. Mach.*, v. 4, 1957, pp. 341–347. MR **20** #417.

14. G. Hadley, *Linear Programming*, Addison-Wesley Series in Industrial Management, Addison-Wesley, Reading, Mass., 1962. MR **24** #B1669.

15. L. Karlovitz, "Construction of nearest points in the L^p, p even, and L^∞ norms. I," *J. Approximation Theory*, v. 3, 1970, pp. 123–127. MR **42** #738.

16. L. Karlovitz, Private communication.

17. G. Meyer, "On solving nonlinear equations with a one-parameter operator imbedding," *SIAM J. Numer. Anal.*, v. 5, 1968, pp. 739–752. MR **39** #3697.

18. J. Ortega & W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970. MR **42** #8686.

19. M. Osborne & G. Watson, "On the best linear Chebyshev approximation," *Comput. J.*, v. 10, 1967, pp. 172–177. MR **36** #1892.

20. V. P. Sreedharan, "Least squares algorithms for finding solutions of overdetermined linear systems which minimize error in an abstract norm," *Numer. Math.*, v. 17, 1971, pp. 387–401. MR **45** #2899.

21. E. Stiefel, "Über diskrete und lineare Tschebyscheff-Approximationen," *Numer. Math.*, v. 1, 1959, pp. 1–28. MR **21** #6681.

22. E. Stiefel, "Note on Jordan elimination, linear programming, and Tschebyscheff approximation," *Numer. Math.*, v. 2, 1960, pp. 1–17. MR **22** #1988.