MATHEMATICS OF COMPUTATION, VOLUME 32, NUMBER 141
JANUARY 1978, PAGES 201–221

Revision of a Derivative-Free Quasi-Newton Method

By John Greenstadt

Abstract. A derivative-free Quasi-Newton (DFQN) method previously published [J. Greenstadt, *Math. Comp.*, v. 26, 1972, pp. 145–166] has been revised and simplified. The main modification has the effect of keeping all the successive approximants to the Hessian matrix positive-definite. This, coupled with some improvements in the line search, has enhanced the performance of the method considerably. The results of numerical trials on many of the "standard" test functions are displayed, in addition to comparisons with two other methods. These indicate that the present DFQN method is not too far behind that of Gill, Murray and Pitfield, the most efficient one presently known.

1. **Introduction.** The work to be described here is an extension of a previous attempt [1] to devise a derivative-free Quasi-Newton (DFQN) method, which does not make explicit use of difference approximations. Considerable improvements have been made, which have rendered the method much more robust and efficient than before.

As is usual, our problem is to minimize a function f of the argument x (which is a vector with N components). We assume that we have available only the value of f (for any x), but none of its derivatives. Part of our task is to estimate the gradient of f ($\equiv \{\partial f/\partial x_i\}$) and its Hessian ($\equiv \{\partial^2 f/\partial x_i \partial x_j\}$) using the available function values only. We shall denote the true values of the gradient and Hessian by $\bar{g}$ and $\bar{G}$, respectively, and the estimates by g and G. Naturally, our reason for making these estimates is so that we may calculate a good step δ, according to Newton's famous formula:

$$\delta = -\bar{G}^{-1}\bar{g}. \tag{1.1}$$

When $\bar{G}$ is positive-definite, formula (1.1) will always provide a descent direction, i.e., one in which $f(x)$ initially decreases. The principal difficulty in [1] was that the computed estimate, G, was often *not* positive-definite (even when the *true* Hessian $\bar{G}$ *was*). One of the main improvements of the present revision is a reliable way of preventing this mishap.

2. **Cycles of Steps.** The overall sequence of steps, by which the minimum of $f(x)$ is sought, is partitioned into subsequences, or *cycles*, of N steps each.[1] Each such cycle is handled independently of all the others, so that the notation we shall use will, for convenience, ignore the fact that there is really a sequence of cycles.

In fact, we shall refer the various points $\{x_i\}$, reached in a given cycle, to the starting point (x_0) of that cycle. The *relative* position vector τ_i within this cycle is

Received June 12, 1975; revised March 24, 1977.

AMS (MOS) subject classifications (1970). Primary 90C30, 90–04.

Key words and phrases. Unconstrained minimization, Quasi-Newton methods, minimization without derivatives.

[1] These were called "major steps" in [1].

then defined as follows:

$$(2.1)\qquad \tau_i \equiv x_i - x_0 \qquad (i = 1, \ldots, N).$$

Obviously, $\tau_0 = 0$. Further, we shall mostly regard f as a function of τ, rather than x.

We shall denote the step vectors within a typical cycle by $\{\sigma_i\}$, with $i = 1, \ldots, N$. Each step can also be defined in terms of a suitably normalized direction vector s_i, and a step length h_i. The sequence of successive relative positions $\{\tau_i\}$ within the cycle is, by definition, given by:

$$(2.2)\qquad \tau_i = \tau_{i-1} + \sigma_i.$$

In turn, σ_i is given by:

$$(2.3)\qquad \sigma_i = h_i s_i.$$

The step length h_i is to be found by a line search along s_i, starting from τ_{i-1}. For convenience, we parametrize the line through τ_{i-1}, and in the direction s_i, using the parameter α_i, so that any position $\tau(\alpha_i)$ along this line is given by:

$$(2.4)\qquad \tau(\alpha_i) \equiv \tau_{i-1} + \alpha_i s_i.$$

On this basis, the function $f(\tau(\alpha_i))$ can be denoted by $F_i(\alpha_i)$, so that

$$(2.5)\qquad F_i(\alpha_i) \equiv f(\tau_{i-1} + \alpha_i s_i).$$

During any line search, we evaluate $F_i(\alpha_i)$ for various values of α_i, and finally end up with a set of three such values $\{\alpha_i^{(1)}, \alpha_i^{(2)}, \alpha_i^{(3)}\}$ with the properties:

$$(2.6a)\qquad \alpha_i^{(1)} < \alpha_i^{(2)} < \alpha_i^{(3)},$$

$$(2.6b)\qquad F_i(\alpha_i^{(1)}) > F_i(\alpha_i^{(2)}) < F_i(\alpha_i^{(3)}).$$

(It is not necessary, however, that $F_i(\alpha_i^{(2)})$ be the minimum of $F_i(\alpha_i)$.)

We define the step length as follows:

$$(2.7)\qquad h_i \equiv \alpha_i^{(2)}.$$

The sequence of directions $\{s_i\}$ is chosen as follows:

(a) At τ_0 (the start of the cycle), we assume that we have estimates g_0 and G, to the true values $\bar{g}(\tau_0)$ and $\bar{G}$. The (unnormalized) direction δ_1 is calculated by the Newton formula:

$$(2.8)\qquad \delta_1 = -G^{-1} g_0.$$

(b) The normalized vector s_1 is calculated by:

$$(2.9)\qquad s_1 \equiv \frac{\delta_1}{\sqrt{\delta_1^T G \delta_1}}$$

(where the superscript T indicates the transpose) which results in:[2]

[2] This normalization is feasible because G can be kept positive definite. In [1], a different normalization was necessary. (Note, too, that all vectors are regarded as column matrices.)

$$s_1^T G s_i = 1. \tag{2.10}$$

(c) The subsequent s_i (for $i = 2, \ldots, N$) are selected recursively, in such a way that they all form a conjugate set with respect to G. Thus,

$$s_i^T G s_j = \delta_{ij} \qquad (i, j = 1, \ldots, N). \tag{2.11}$$

(In the program used for testing, successive coordinate directions were selected, and the Gram-Schmidt orthogonalization procedure was applied, with G as the weight matrix. The linear independence of each new direction vector was checked.)

With a set of $\{s_i\}$ that satisfy Eq. (2.11), the following considerations prove to be useful: Since the $\{s_i\}$ have been constructed so as to be linearly independent, we can form the nonsingular matrix S, whose columns consist of the vectors $\{s_i\}$ as follows:

$$S \equiv \{s_1, s_2, \ldots, s_N\}. \tag{2.12}$$

Also, we can form the matrix R, whose columns consist of the products $\{Gs_i\}$ as follows:

$$R \equiv \{Gs_1, Gs_2, \ldots, Gs_N\} = GS. \tag{2.13}$$

Forming the product $R^T S$, we have

$$R^T S = \{s_i^T G s_j\} = \{\delta_{ij}\} = I \tag{2.14}$$

as a consequence of (2.11). Hence, it is clear that

$$R^T = S^{-1} \tag{2.15}$$

and it follows that:

$$\sum_{i=1}^{N} s_i s_i^T G = SR^T = SS^{-1} = I. \tag{2.16}$$

3. The Quasi-Newton (QN) Conditions. All QN conditions may be regarded as identities on quadratic functions. Following this viewpoint, we approximate $f(\tau)$ locally[3] by a quadratic function $Q(\tau)$, defined by

$$Q(\tau) \equiv Q_0 + \tau^T g_0 + \tfrac{1}{2}\tau^T G \tau, \tag{3.1}$$

where g_0 and G are the approximations associated with the current cycle. After this cycle has been completed, the information gathered in regard to $f(\tau)$ is to be used to update g_0 and G. (These updates we shall denote by g_0^* and G^*.) This will be done in such a way that $Q(\tau)$ will match $f(\tau)$ on every step in the cycle. This updated Q (to be denoted by Q^*) is defined quite analogously to (3.1):

$$Q^*(\tau) \equiv Q_0^* + \tau^T g_0^* + \tfrac{1}{2}\tau^T G^* \tau. \tag{3.2}$$

[3] "Locally" means: On the set of points $\{\tau_i\}$, (with $i = 0, \ldots, N$) which make up a cycle.

Along the line defined by (2.4), $Q^*(\tau)$ depends only on α_i, so that for convenience, we shall define a function $R_i(\alpha_i)$ as follows:

$$R_i(\alpha_i) \equiv Q^*(\tau_{i-1} + \alpha_i s_i) \tag{3.3}$$

and, by expanding Q^*, we obtain:

$$\begin{aligned} R_i(\alpha_i) = {} & (Q_0 + \tau_{i-1}^T g_0^* + \tfrac{1}{2}\tau_{i-1}^T G^* \tau_{i-1}) \\ & + (s_i^T g_0^* + s_i^T G^* \tau_{i-1})\alpha_i + \tfrac{1}{2}(s_i^T G^* s_i)\alpha_i^2 . \end{aligned} \tag{3.4}$$

The three expressions in parentheses will be denoted by a_i, b_i and c_i, respectively, so that $R_i(\alpha_i)$ can be abbreviated to:

$$R_i(\alpha_i) = a_i + b_i\alpha_i + \tfrac{1}{2}c_i\alpha_i^2 . \tag{3.5}$$

We are now ready to match up the data developed in the line search, and summarized in (2.6), with the local approximation (3.5). We shall require that:

$$R_i(\alpha_i^{(1)}) = F_i(\alpha_i^{(1)}), \quad R_i(\alpha_i^{(2)}) = F_i(\alpha_i^{(2)}), \quad R_i(\alpha_i^{(3)}) = F_i(\alpha_i^{(3)}). \tag{3.6}$$

More explicitly, Eqs. (3.6) are:

$$\begin{aligned} a_i + b_i\alpha_i^{(1)} + \tfrac{1}{2}c_i(\alpha_i^{(1)})^2 &= F_i(\alpha_i^{(1)}), \\ a_i + b_i\alpha_i^{(2)} + \tfrac{1}{2}c_i(\alpha_i^{(2)})^2 &= F_i(\alpha_i^{(2)}), \\ a_i + b_i\alpha_i^{(3)} + \tfrac{1}{2}c_i(\alpha_i^{(3)})^2 &= F_i(\alpha_i^{(3)}), \end{aligned} \tag{3.7}$$

which can be solved for a_i, b_i and c_i in terms of the known quantities

$$\{\alpha_i^{(1)}, \alpha_i^{(2)}, \alpha_i^{(3)}\} \quad \text{and} \quad \{F_i(\alpha_i^{(1)}), F_i(\alpha_i^{(2)}), F_i(\alpha_i^{(3)})\} .$$

We may now regard the data gleaned in each line search as summarized implicitly in the calculated values of a_i, b_i and c_i.[4] Referring back to their definitions, we may write:[5]

$$s_i^T g_0^* + s_i^T G^* \tau_{i-1} = b_i, \tag{3.8a}$$

$$s_i^T G^* s_i = c_i; \tag{3.8b}$$

and we have thus generated conditions on g_0^* and G^* in terms of the known quantities $\{s_i, \tau_i, b_i, c_i\}$. These conditions hold for $i = 1, \ldots, N$, i.e., for every step in the cycle.

We now introduce additive corrections to g_0 and G, defined as follows:

$$g_0^* \equiv g_0 + \gamma, \tag{3.9a}$$

$$G^* \equiv G + \Gamma. \tag{3.9b}$$

Equations (3.8) can then be rewritten in terms of the new unknowns γ and Γ:

$$s_i^T\gamma + s_i^T\Gamma\tau_{i-1} = b_i - s_i^T g_0 - s_i^T G\tau_{i-1} \equiv \epsilon_i, \tag{3.10a}$$

[4] Because of (2.6) it may readily be proved that $c_i > 0$.

[5] It turns out that a_i need never be used.

$$s_i^T \Gamma s_i = c_i - s_i^T G s_i = c_i - 1. \tag{3.10b}$$

The last reduction follows from (2.11). We can also reduce (3.10a) by noting that, based on (2.2) and (2.3):

$$\tau_i = \tau_{i-1} + h_i s_i, \tag{3.11}$$

which implies that

$$\tau_i = \sum_{j=1}^{i} h_j s_j. \tag{3.12}$$

Since τ_{i-1} clearly does not include s_i, and since s_i is conjugate to all $\{s_j\}$ with $j < i$, we have:

$$s_i^T G \tau_{i-1} = 0 \tag{3.13}$$

from which it follows that ϵ_i can be reduced, so that (3.10a) becomes:

$$s_i^T \gamma + s_i^T \Gamma \tau_{i-1} = \epsilon_i = b_i - s_i^T g_0. \tag{3.14}$$

Equations (3.10b) and (3.14) are the QN conditions for this problem.

4. Variational Derivation of Γ. After having completed a cycle of N steps, we consider next how to use the information collected to estimate the corrections γ and Γ. In [1], a functional was constructed, involving both quantities; and a variational procedure was used to derive formulas for both. However, there were serious ambiguities in that approach,[6] so that we shall now depart from that scheme.

Our strategy will be to regard γ as merely a (vector) parameter, and to concentrate at first on Γ alone. If G (and hence Γ) be regarded as a covariant tensor of second rank (as it is when thought of as a "metric"), then the simplest *quadratic invariant* involving Γ would be (with a convenience factor of ½):

$$\Phi_0 \equiv \tfrac{1}{2} \operatorname{Tr}\{G^{-1} \Gamma G^{-1} \Gamma^T\} \tag{4.1}$$

(where the symbol Tr indicates the trace). We are not assuming Γ to be symmetric *a priori*, but will require it to come out that way.

To the bare functional Φ_0, we must adjoin the QN constraints, as well as the symmetry constraint on Γ. We use the Lagrange multipliers $\{\theta_i\}$, $\{\eta_i\}$ and Λ (a matrix). The complete functional is then:

$$\begin{aligned}\Phi = \tfrac{1}{2}\operatorname{Tr}\{G^{-1}\Gamma G^{-1}\Gamma^T\} - 2 \sum_{i=1}^{N} \theta_i \{s_i^T(\gamma + \Gamma\tau_{i-1}) - \epsilon_i\} \\ -\sum_{i=1}^{N} \eta_i \{s_i^T \Gamma s_i - c_i + 1\} - \operatorname{Tr}\{\Lambda(\Gamma - \Gamma^T)\}.\end{aligned} \tag{4.2}$$

We follow the method of solution described in [1], but shall not go into detail here; the formula for Γ turns out to be:

[6] As emphasized to me by M. J. D. Powell.

$$\Gamma = G \sum_{i=1}^{N} \{\theta_i(s_i\tau_{i-1}^T + \tau_{i-1}s_i^T) + \eta_i s_i s_i^T\}G. \tag{4.3}$$

(Note that, although θ_1 appears *formally*, it is not really included, because of the vanishing of τ_0.)

The η's may be immediately evaluated by applying QN condition (3.10b). We have:

$$\begin{aligned} s_k^T\Gamma s_k &= \sum_{i=1}^{N} \{\theta_i[s_k^T G s_i \tau_{i-1}^T G s_k + s_k^T G\tau_{i-1} s_i^T G s_k] + \eta_i s_k^T G s_i s_i^T G s_k\} \\ &= \sum_{i=1}^{N} \{\theta_i[\delta_{ik}\tau_{i-1}^T G s_k + \delta_{ik} s_k^T G\tau_{i-1}] + \eta_i \delta_{ki}\delta_{ik}\} \\ &= 2\theta_k \tau_{k-1}^T G s_k + \eta_k = \eta_k = c_k - 1. \end{aligned} \tag{4.4}$$

The various reductions follow from (2.11) and (3.13).

We next apply the remaining QN condition (3.14) to Γ and γ. Substituting for Γ from (4.3), we obtain:

$$\begin{aligned} s_i^T\Gamma\tau_{i-1} &= s_i G \sum_j \{\theta_j(s_j\tau_{j-1}^T + \tau_{j-1}s_j^T) + \eta_j s_j s_j^T\}G\tau_{i-1} \\ &= \sum_j \{\theta_j(\delta_{ij}\tau_{j-1}^T G\tau_{i-1} + s_i^T G\tau_{j-1} s_i^T G\tau_{i-1}) + \eta_j \delta_{ij} s_j^T G\tau_{i-1}\} \\ &= \theta_i \tau_{i-1}^T G\tau_{i-1} + \sum_i \theta_j (s_i^T G\tau_{j-1})(s_j^T G\tau_{i-1}) + \eta_i s_i^T G\tau_{i-1}. \end{aligned} \tag{4.5}$$

The last term above vanishes because of (3.13). The term preceding that vanishes too because regardless of the values of i and j, at least one of the factors is zero (again because of the conjugacy of the $\{s_i\}$). If we define:

$$\tau_{i-1}^2 \equiv \tau_{i-1}^T G\tau_{i-1}, \tag{4.6}$$

then we can write

$$s_i\Gamma\tau_{i-1} = \theta_i\tau_{i-1}^2. \tag{4.7}$$

We can greatly simplify (3.14), if we recall that the set of vectors $\{Gs_i\}$ is complete. This means that we can expand the vector γ as follows:

$$\gamma = \sum_{j=1}^{N} \mu_j G s_j \tag{4.8}$$

so that

$$s_i^T\gamma = \sum_{j=1}^{N} \mu_j \delta_{ij} = \mu_i, \tag{4.9}$$

and (3.14) reduces to:

$$\mu_i + \tau_{i-1}^2\theta_i = \epsilon_i. \tag{4.10}$$

Since the $\{c_i\}$ are known quantities, we need not concern ourselves further with the QN condition (4.4). On the other hand, the QN condition (4.10) involves *two* unknown quantities (viz., μ_i and θ_i) for each step. As we shall see, the constraints on $\{\theta_i\}$ which are necessary to insure the positive-definiteness of G^* will enable us to determine both quantities.

5. Maintenance of Positive-Definiteness. We shall first express G^* directly in terms of G by applying the correction (3.9b) explicitly. We obtain, with the help of (4.3), (4.4) and (2.16):

$$
\begin{aligned}
G^* &= G + \Gamma = \sum_{i=1}^{N} Gs_is_i^TG + \Gamma \\
&= \sum_{i=1}^{N} Gs_is_i^TG + \sum_{i=1}^{N} (c_i - 1)Gs_is_i^TG + \sum_{i=1}^{N} \theta_iG(s_i\tau_{i-1}^T + \tau_{i-1}s_i^T)G \qquad (5.1)\\
&= \sum_{i=1}^{N} \{c_iGs_is_i^TG + \tau_iG(s_i\tau_{i-1}^T + \tau_{i-1}s_i^T)G\}.
\end{aligned}
$$

Our subsequent analysis will be greatly simplified if we transform G^* as follows, to form B:

$$B \equiv S^TG^*S, \qquad (5.2)$$

where S is defined as in (2.12). The elements of B are given by:

$$
\begin{aligned}
B_{km} &= s_k^TG^*s_m \\
&= s_k^T \sum_{i=1}^{N} \{c_iGs_is_i^TG + \theta_iG(s_i\tau_{i-1}^T + \tau_{i-1}s_i^T)G\}s_m \\
&= \sum_{i=1}^{N} \{c_i(s_k^TGs_i)(s_i^TGs_m) + \theta_i[(s_k^TGs_i)(\tau_{i-1}^TGs_m) + (s_k^TG\tau_{i-1})(s_i^TGs_m)]\} \qquad (5.3)\\
&= \sum_{i=1}^{N} \{c_i\delta_{ki}\delta_{im} + \theta_i[\delta_{ki}(\tau_{i-1}^T Gs_m) + (s_k^TG\tau_{i-1})\delta_{im}]\} \\
&= c_k\delta_{km} + \theta_k(\tau_{k-1}^TGs_m) + \theta_m(s_k^TG\tau_{m-1}).
\end{aligned}
$$

The reductions are based on the conjugacy relation (2.11). Further simplification may be effected by generalizing (3.13), based on the expression (3.12) for τ_i. Since $(\tau_{k-1}^TGs_m)$ is the same as $(s_m^TG\tau_{k-1})$, we need consider only the latter. Clearly, if $k \leqslant m$, this expression vanishes, since τ_{k-1} does not then contain s_m. On the other hand, if $k > m$, then the surviving part of the inner product is $h_m(s_m^TGs_m)$ which, of course, is just equal to h_m. We can summarize as follows:

$$
\begin{aligned}
(s_m^TG\tau_{k-1}) &= 0 \quad \text{if } k \leqslant m, \qquad (5.4)\\
&= h_m \quad \text{if } k > m.
\end{aligned}
$$

On this basis, we can display B:

$$B = \begin{bmatrix} c_1 & h_1\theta_2 & h_1\theta_3 & h_1\theta_4 & \cdots & h_1\theta_N \\ h_1\theta_2 & c_2 & h_2\theta_3 & h_2\theta_4 & \cdots & h_2\theta_N \\ h_1\theta_3 & h_2\theta_3 & c_3 & & & \cdot \\ h_1\theta_4 & h_2\theta_4 & & \ddots & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & h_{N-1}\theta_N \\ \cdot & \cdot & & & & \\ h_1\theta_N & h_2\theta_N & & & h_{N-1}\theta_N & c_N \end{bmatrix}. \tag{5.5}$$

Equation (5.2) can be solved for G^* by multiplying by R and using (2.14). We have:

$$RBR^T = (RS^T)G^*(SR^T) = G^* \tag{5.6}$$

which shows, together with (5.2), that G^* will be positive-definite if and only if B is. We may, therefore, concentrate our efforts on B.

There are undoubtedly several ways of accomplishing our end; we shall consider two, but display numerical results for only one of them.

First, we shall concentrate on keeping all the eigenvalues of B positive. This may be done by the use of Gershgorin's Theorem [2]. If λ is any eigenvalue of B, then it satisfies:

$$|\lambda - B_{ii}| \leqslant \sum_{j \neq i} |B_{ij}| \tag{5.7}$$

which means that

$$\lambda \geqslant B_{ii} - \sum_{j \neq i} |B_{ij}|, \tag{5.8}$$

so that if, for some number $\phi_i > 0$, we insure that

$$B_{ii} - \sum_{j \neq i} |B_{ij}| \geqslant \phi_i, \tag{5.9}$$

we then have

$$\lambda \geqslant \min_i \phi_i > 0. \tag{5.10}$$

From (5.5), it is clear that:

$$B_{ii} = c_i, \tag{5.11a}$$

$$\sum_{j \neq i} |B_{ij}| = |\theta_i| \sum_{j=1}^{i-1} h_j + h_i \sum_{j=i+1}^{N} |\theta_j|. \tag{5.11b}$$

Since, as was indicated previously, all of the $\{c_i\}$ are positive, we may "scale" the ϕ_i, in a sense, by setting:

$$\phi_i = \beta_i c_i, \tag{5.12}$$

where $\beta_i > 0$. Substituting these relations in (5.9), we ask that:

$$c_i - |\theta_i| \sum_{j=1}^{i-1} h_j - h_i \sum_{j=i+1}^{N} |\theta_j| \geqslant \beta_i c_i, \tag{5.13}$$

in which case, we shall have:

$$\lambda \geqslant \min_i \beta_i c_i. \tag{5.14}$$

If we rewrite (5.13), we obtain:

$$\left(\sum_{j=1}^{i-1} h_j \right) |\theta_i| \leqslant (1 - \beta_i) c_i - h_i \sum_{j=i+1}^{N} |\theta_j|, \tag{5.15}$$

which for $i = 1, \ldots, N$ serves as a set of bounds on $\{ |\theta_i| \}$. (Clearly, β_i must be less than unity.) These bounds may be applied recursively, starting with θ_N. Thus, for example:

$$\left(\sum_{j=1}^{N-1} h_j \right) |\theta_N| \leqslant (1 - \beta_N) c_N, \tag{5.16a}$$

$$\left(\sum_{j=1}^{N-2} h_j \right) |\theta_{N-1}| \leqslant (1 - \beta_{N-1}) c_{N-1} - h_{N-1} |\theta_N|, \tag{5.16b}$$

etc.

We shall next consider another method[7] for bounding the θ's, related not to the eigenvalues of B, but to a sequence of principal minors of B.

If we define the matrix B_i as follows:

$$B_i \equiv \begin{bmatrix} c_1 & h_1\theta_2 & \cdots & h_1\theta_i \\ h_1\theta_2 & c_2 & \cdots & \vdots \\ \vdots & & & h_{i-1}\theta_i \\ h_1\theta_i & & h_{i-1}\theta_i & c_i \end{bmatrix} \tag{5.17}$$

and the vector q_i by:

$$q_i \equiv \{h_1, h_2, \ldots, h_i\} \tag{5.18}$$

then, clearly, we have the recursion:

$$B_i = \begin{bmatrix} B_{i-1} & q_{i-1}\theta_i \\ q_{i-1}^T\theta_i & c_i \end{bmatrix} \tag{5.19}$$

and we shall attempt to insure the positive-definiteness of B_i, given that of B_{i-1}. If this can be done for all i then, since $B_N = B$, we shall have our result.

To further facilitate the analysis, we transform B_i with the matrix Q_i, defined by:

[7] Which is based on a suggestion made by Dr. S. Schechter.

$$Q_i \equiv \begin{bmatrix} I_{i-1} & -\theta_i B_{i-1}^{-1} q_{i-1} \\ 0 & 1 \end{bmatrix} \tag{5.20}$$

to obtain the new matrix D_i:

$$D_i \equiv Q_i^T B_i Q_i = \begin{bmatrix} B_{i-1} & 0 \\ 0 & \phi_i \end{bmatrix}, \tag{5.21}$$

where

$$\phi_i \equiv c_i - (q_{i-1}^T B_{i-1}^{-1} q_{i-1})\theta_i^2 . \tag{5.22}$$

As before, if D_i can be kept positive-definite, then B_i will be also. Since B_{i-1} has been assumed to be positive-definite, then D_i is positive-definite if and only if $\phi_i > 0$. We therefore choose some positive number β_i, and require that:

$$\phi_i \equiv c_i - \omega_{i-1}\theta_i^2 \geqslant \beta_i c_i > 0, \tag{5.23}$$

where

$$\omega_i \equiv q_i^T B_i^{-1} q_i, \tag{5.24}$$

and this in turn establishes the constraint on θ_i:

$$\omega_{i-1}\theta_i^2 \leqslant (1 - \beta_i)c_i \tag{5.25}$$

(and again, β_i must be less than unity).

If this recursive process is continued until $i = N$, we then have a positive-definite D_N; hence a positive-definite B_N; hence a positive-definite G^*.

6. Selection of θ_i and μ_i. The remaining QN condition, Eq. (4.10) will now be used in conjunction with the constraints on $\{\theta_i\}$, to effect unique choices for θ_i and μ_i at each step. Clearly, for $i = 1$, we have the forced choice:

$$\mu_1 = \epsilon_1 \tag{6.1}$$

and, as remarked previously, θ_1 does not enter into the problem at all.

For $i > 1$, our strategy will be to choose the μ_i of smallest magnitude, consistent with the constraint on θ_i. This strategy is in the same spirit of "minimal correction" which prompted the formulation of the selection of Γ as a variational problem.

If there were *no* constraints on the θ's, the choice would obviously be

$$\left.\begin{aligned} &\mu_i = 0 &&(6.2a)\\ &\theta_i = \epsilon_i/\tau_{i-1}^2 &&(6.2b)\end{aligned}\right\}, \quad i > 1.$$

However, this strategy almost always leads to an indefinite G^*, with catastrophic results (as observed in practice). This is the reason for applying the constraints to keep G^* positive-definite.

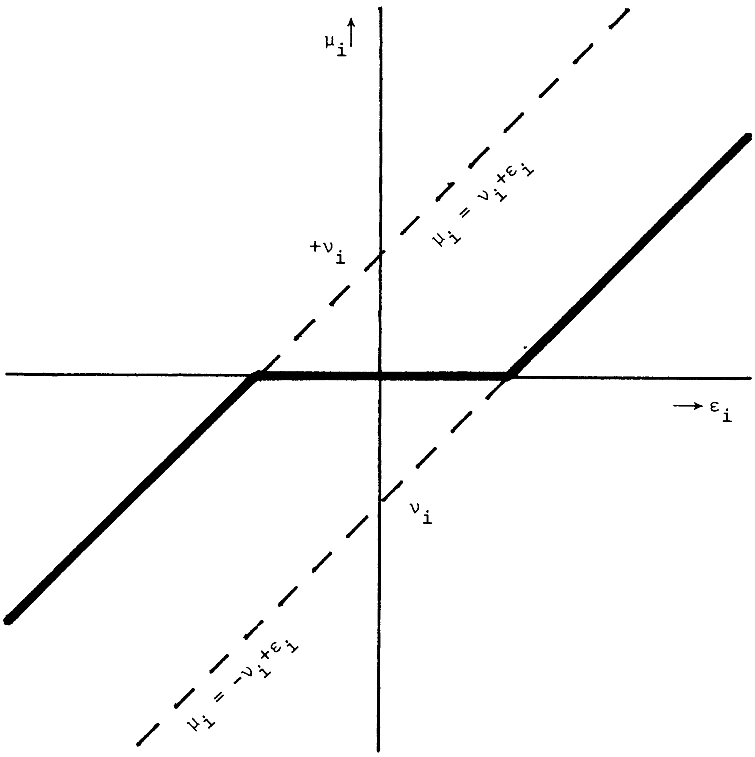

Figure 1

Since the constraints (5.15) and (5.25) may both be written in the same form:

(6.3) $$|\theta_i| \leqslant \lambda_i$$

with $\lambda_i > 0$, we shall treat them together. From (4.10) we have:

(6.4) $$\epsilon_i - \mu_i = \tau_{i-1}^2 \theta_i$$

so that

(6.5) $$|\epsilon_i - \mu_i| = \tau_{i-1}^2 |\theta_i| \leqslant \tau_{i-1}^2 \lambda_i \equiv \nu_i.$$

We now wish to choose μ_i as small as possible in magnitude consistent with (6.5). This is a (trivial) linear programming problem, which may be solved graphically. In Figure 1, the two oblique lines bound the region of the (ϵ, μ) plane wherein (6.5) is satisfied. The heavy line traces the minimum magnitude μ_i within this region. This solution may be written as:

(6.6) $$\mu_i = \text{sign}(\epsilon_i) \times \max(0, |\epsilon_i| - \nu_i)$$

with this choice of μ_i, θ_i may now be determined from (6.4). In this way, we have, so to speak, "apportioned" the increments to the corrections Γ and γ in a natural manner by using the constraints on G^*.

Thus, at the end of a cycle, we are in a position to update g_0 and G, according to (3.9), (4.3) and (4.8). In addition, because g_0 is assumed to vary linearly with x, we must perform a *translation* of it, to the new starting point. If we denote the translated value by g_0^{**}, we have

$$g_0^{**} = g_0^{*} + G^* \tau_N. \tag{6.7}$$

7. **Choice of $\{\beta_i\}$ in Second Method.** As a matter of experience the second method described in Section 5 for maintaining positive-definiteness turned out to be considerably the better. Hence, all of our results are for this method.

The choice of the β's remains arbitrary. By way of a guide, we shall examine the effect of the β-values on the determinant of G^*. We have:

$$\det G^* = (\det G^*)(\det G)^{-1}(\det G) = \det(G^*G^{-1})(\det G) \tag{7.1}$$

and, using (5.6):

$$\det(G^*G^{-1}) = \det(RBR^TG^{-1}) = \det(BR^TG^{-1}R) = \det(B)\ \det(R^TG^{-1}R). \tag{7.2}$$

But, from (2.15) and (2.14):

$$\det(R^TG^{-1}R) = \det(S^{-1}G^{-1}R) = \det[(GS)^{-1}R] ,= \det[R^{-1}R] = 1 \tag{7.3}$$

so that, finally:

$$\det G^* = (\det B)(\det G). \tag{7.4}$$

Next, from (5.21), we have:

$$\det D_i = \det(Q_i^TB_iQ_i) = (\det B_i) \times (\det Q_i)^2; \tag{7.5}$$

but it is clear from the form (5.20) of Q_i that $\det Q_i = 1$, so that

$$\det B_i = \det D_i. \tag{7.6}$$

On the other hand, it is also clear from (5.21) that

$$\det B_i = \det D_i = (\det B_{i-1}) \times \phi_i \tag{7.7}$$

which gives a recursion for $\det B_i$, and since $B_1 = \phi_1$, we conclude that:

$$\det B = \prod_{i=1}^{N} \phi_i. \tag{7.8}$$

Using all these results, together with the constraints (5.23), we can bound $\det G^*$ below as follows:

$$\begin{aligned} \det G^* &= (\det G) \times (\det B) = \det G \times \left(\prod_i \phi_i\right) \\ &\geqslant \det G \times \left(\prod_i c_i\right) \times \left(\prod_i \beta_i\right). \end{aligned} \tag{7.9}$$

Clearly, if some of the c's are small, $\det G^*$ will be much smaller than $\det G$ since $\beta_i < 1$. Whatever the case, it is obviously advantageous to try to keep the determinants as large as possible. This means that the β's should be fairly close to unity.

The most obvious way of "balancing" the β's is to set them all equal to a predetermined constant. We may simply choose a value, once and for all, or make use of (7.9) as a guide to fitting a value to each problem. If all of the β's are equal; (7.9) becomes:

$$\det G^* \geqslant \left[(\det G) \times \left(\prod_i c_i \right) \right] \times \beta^N. \tag{7.10}$$

Since we have no control over the factor within the brackets, we can ignore it, and concentrate our attention on β^N. If we demand that this factor should be no less than some fixed constant ρ, then we should set $\beta^N = \rho$, so that:

$$\beta = \rho^{1/N}, \tag{7.11}$$

which has the desirable property that β gets closer and closer to unity as N gets larger. The value of ρ must be established by numerical experiment.

8. **Numerical Results.** We have performed our tests on many of the "standard" functions in the literature using the "standard" starting points. We list the names of these functions here, with appropriate references, and add any comments that serve to clarify our results (N is the number of arguments):

(1) Helical Valley [3].

(2) Rosenbrock's Function [4].

(3) Wood's Function [5].

(4) Powell's Quartic Function [6].

(5) Watson's Function [7]. This has been tested for $N = 6$ and 9.

(6) Chebyquad [8]. This has been tested.for $N = 4$, 6, 8 and 20.

(7) Random Trigonometric Functions [3]. These are trigonometric polynomials whose coefficients are random variables (fixed, of course, for each case). The starting points are also random variables. Because of this, the behavior of each function so generated is unique and unpredictable, so that 3 runs were made for each case. Runs were done for $N = 3$, 5, 10, and 20, and the number of function evaluations averaged. Those runs wherein the method converged[8] to a minimum different from the predetermined one were ignored, since they do not support a fair comparison. All the runs shown to converge did so to the correct solutions.

(8) Biggs' Exponential Functions [9]. There are two functions, called EXP5 and EXP6 with 5 and 6 arguments, respectively.

In Table 1 are shown the numbers of function evaluations necessary for convergence for most of these functions, when the value β is fixed independently of N. Since $0 < \beta < 1$, the five β-values covering this range were tried. It is abundantly clear that, although fixing β may be satisfactory when N (indicated in parentheses) is small, it is totally unsatisfactory for large N, as evidenced by the failures of convergence (marked "F") for Chebyquad and the trigonometric functions when $N = 20$.

[8] For all functions but the random trigonometric functions, convergence was defined as requiring that $g^T G^{-1} g < 10^{-12}$. For the trigonometric functions, it was defined as requiring that $\max_i(|x_i - x_{0i}|) < 10^{-6}$, where x_0 was the known location of the correct minimum.

TABLE 1

Function evaluations vs β-values

β→	.1	.3	.5	.7	.9
Function					
Beale (2)	85	76	86	66	82
Hel (3)	285	287	315	232	239
Ros (2)	236	174	203	203	165
Wood (4)	508	316	314	266	288
Pow (4)	1060	668	568	467	711
Wat (6)	940	1064	662	456	540
Cheb (4)	222	176	326	122	138
(6)	343	403	314	346	253
(8)	3780	1524	684	836	603
(20)	F	F	F	5044	3499
Trig (3)	F	85	126	122	82
"	64	116	85	102	118
"	120	113	74	105	148
MEAN	F	105	95	110	116
Trig (5)	737	244	249	215	158
"	F	235	260	218	220
"	430	303	225	331	180
MEAN	F	261	245	255	186
Trig (10)	F	3336	743	874	978
"	F	3681	1422	512	728
"	22246	2548	1721	644	610
MEAN	F	3188	1295	677	772
Trig (20)	F	F	F	3611	1949
"	F	F	F	3988	2789
"	F	F	F	3471	2740
MEAN	F	F	F	3690	2493

The results with β determined from Eq. (7.11) are shown in Table 2 for nine representative values of ρ over its allowable range. Clearly, the performance is far better (since there are no failures) and the performance of the algorithm is relatively insensitive to the ρ-values. However, the value $\rho = .5$ seems slightly better than the others, so that this value was used for further runs.

For comparison with the results of Gill, Murray and Pitfield [10] (GMP) the convergence criterion was adjusted for each function, for termination when the difference between the function value at the end of a cycle and its known minimum value fell within the accuracy given by GMP. In Table 3 are shown the numbers of cycles (noted as ITER), the number of function evaluations (EVALS), and the final accuracy (ACCUR.). The DFQN method is comparable to GMP except for the Chebyquad cases, EXP5 and EXP6. The reason for this poor behavior is not known. (The L in the last line indicates that a local minimum was found.)

In Table 4, the DFQN method applied to the random trigonometric functions is compared with the results quoted by Powell [11] for his 1964 method requiring no derivatives. As can be seen, the DFQN method is slightly worse, but manages to keep up for large N. An additional set of three cases for $N = 50$ was run, with the

TABLE 2

Function evaluations vs ρ-values

ρ→	.1	.2	.3	.4	.5	.6	.7	.8	.9
Function									
Beale (2)	78	91	81	68	74	66	85	88	63
Hel (3)	251	269	250	202	194	204	255	285	282
Ros (2)	189	220	146	147	144	137	163	175	148
Wood (4)	287	277	331	301	277	261	257	243	228
Pow (4)	634	806	556	674	501	576	636	561	527
Wat (6)	476	582	385	371	369	354	346	386	478
Cheb (4)	228	122	122	129	121	121	130	130	134
(6)	273	341	359	426	268	328	329	360	366
(8)	563	503	567	744	503	578	593	587	635
(20)	3454	3155	3053	2963	3069	2820	3023	3099	3253
Trig (3)	78	63	96	90	78	66	104	75	92
"	73	77	60	76	68	121	63	74	108
"	77	71	59	74	71	132	93	107	115
MEAN	76	70	72	70	72	68	87	85	105
Trig (5)	219	204	179	212	183	282	267	259	155
"	294	181	176	178	183	202	199	242	224
"	187	207	216	517	207	178	214	224	189
MEAN	233	197	190	302	191	221	227	242	189
Trig (10)	822	538	577	617	664	754	933	579	708
"	778	606	715	674	626	928	863	727	612
"	598	566	753	636	768	696	617	841	751
MEAN	733	570	682	642	686	793	804	716	690
Trig (20)	2061	2314	1665	1838	1851	2158	2428	2231	2720
"	2428	2326	1958	2118	1861	2196	2106	2519	2513
"	2651	2260	2642	3223	1928	1998	2256	2170	2717
MEAN	2380	2300	2088	2393	1880	2117	2263	2307	2650

results and the mean shown. (The number of function evaluations for convergence of the DFQN method appears to be proportional to $N^{1.8}$.)

It is of interest to observe the detailed behavior of this algorithm for a few cases. In Tables 5 and 6 are shown the results for the Helical Valley and for Rosenbrock's Function. Not only is the convergence clearly superlinear near the solution, but the final estimate "GG" of the Hessian is quite close to that computed by central differences at the solution point.

The output for Powell's function with a quartic minimum is given in Table 7, and shows quite clearly that a method based on quadratic approximation hardly works at all near a higher-order minimum. The convergence is certainly not superlinear (barely linear!), and the final estimate for the Hessian is very far from the differenced estimate (which is very accurate). Oddly enough, the "Hadamard condition number", defined by:

$$C_H \equiv (\det G) \Bigg/ \left(\prod_{i=1}^{N} \left(\sum_{j=1}^{N} G_{ij}^2 \right)^{1/2} \right) \tag{8.1}$$

has almost the same value for both estimates. Since $\overline{G}$ is, in reality, singular, the conjugacy relations (2.11) become impossible to maintain with sufficient accuracy. Each time such a failure occurs, it is noted, and the total printed in the output, as shown.

TABLE 3

Comparison of DFQN and GMP methods

	DFQN			GMP		
Function	ITER	EVALS	ACCUR.	ITER	EVALS	ACCUR.
Hel	23	194	3.5×10^{-27}	27	165	2.5×10^{-26}
Ros	25	136	3.7×10^{-15}	26	133	2.8×10^{-14}
Wood	25	261	3.4×10^{-20}	55	395	4.4×10^{-19}
Pow	43	421	1.3×10^{-22}	41	398	1.6×10^{-22}
Wat 6	24	333	4.4×10^{-12}	33	351	1.0×10^{-11}
Wat 9	69	1388	1.4×10^{-10}	56	939	2.8×10^{-10}
Cheb 4	9	105	2.8×10^{-18}	8	67	2.9×10^{-15}
6	15	232	2.3×10^{-17}	13	135	2.5×10^{-15}
8	23	487	7.9×10^{-14}	20	251	1.6×10^{-13}
20	69	3069	2.8×10^{-13}	47	1189	2.5×10^{-13}
Exp 5	61	718	3.5×10^{-20}	44	401	4.9×10^{-18}
Exp 6	42	669	5.3×10^{-13}(L)	99	978	4.1×10^{-18}

TABLE 4

Comparison of DFQN and Powell's methods on random trigonometric functions

	DFQN	POWELL 1964
Trig 3	72	108
Trig 5	191	167
Trig 10	686	504
Trig 20	1880	2389
Trig 50	9989	
"	15078	
"	10943	
MEAN	12003	

TABLE 5

```
HELICAL VALLEY

CYCLE EVALS      F              X→
  0      4     2.5000E03       ¯1.0000E00    0.0000E00    0.0000E00
  1     32     2.0216E01       ¯9.2457E¯01   6.2522E¯01   4.1739E00
  2     40     1.7257E01       ¯8.6814E¯01   6.1197E¯01   4.0746E00
  3     49     1.1676E01       ¯4.9033E¯01   9.9748E¯01   3.2299E00
  4     57     8.7017E00       ¯2.7873E¯01   9.9702E¯01   2.8815E00
  5     65     5.5328E00        2.6616E¯01   1.0667E00    2.1264E00
  6     74     4.6213E00        4.3775E¯01   1.0103E00    1.8770E00
  7     83     1.8108E00        8.4808E¯01   6.6381E¯01   1.0132E00
  8     92     1.2519E00        7.9490E¯01   4.7963E¯01   8.5569E¯01
  9    101     8.1327E¯01       8.9252E¯01   2.9189E¯01   4.5458E¯01
 10    107     3.7603E¯01       9.1868E¯01   3.6711E¯01   5.9447E¯01
 11    113     2.4722E¯01       9.5125E¯01   2.8770E¯01   4.4645E¯01
 12    120     6.6381E¯02       9.7831E¯01   1.4257E¯01   2.2493E¯01
 13    129     3.0043E¯03       1.0035E00    5.8743E¯03   1.3294E¯02
 14    136     1.1976E¯03       1.0023E00    1.5532E¯02   2.4799E¯02
 15    143     5.3857E¯04       1.0010E00    1.1213E¯02   1.8699E¯02
 16    152     1.2269E¯04       1.0003E00    4.5152E¯03   6.3285E¯03
 17    158     8.5734E¯06       1.0000E00   ¯1.6533E¯03  ¯2.7338E¯03
 18    164     2.8928E¯07       1.0000E00   ¯3.3210E¯04  ¯5.2372E¯04
 19    170     6.0667E¯10       1.0000E00    6.1106E¯06   8.3140E¯06
 20    176     3.5394E¯12       1.0000E00    3.1216E¯07   6.5073E¯07
 21    182     2.2862E¯15       1.0000E00   ¯2.1773E¯08  ¯3.7078E¯08
 22    188     9.0942E¯21       1.0000E00    2.6327E¯11   3.6231E¯11
     CONVERGED
 23    194     3.4648E¯27       1.0000E00   ¯2.4880E¯14  ¯3.5224E¯14

GNORM,STEP    8.2842E¯14 1.349E¯10


    GC

  200.03        ¯0.025331     0.014072
   ¯0.025331    506.61       ¯318.31
    0.014072   ¯318.31        201.99


   GGDIF

  2.0000E2      1.4010E¯11  ¯7.9228E¯12
  1.4010E¯11    5.0661E2    ¯3.1831E2
 ¯7.9228E¯12   ¯3.1831E2     2.0200E2
```

9. Discussion. Although the performance of the DFQN algorithm is creditable enough in most cases, it is clearly inferior to the GMP method for Chebyquad, EXP5 and EXP6.

The possibility of improving this type of algorithm by generalizing it has been outlined by Powell [12]. He terms these methods "B-conjugate" methods.[9] The relations (2.11) are retained, but the QN conditions, instead of being restricted to (3.10b) and (3.14), are generalized by Powell to:

$$\sum_{ij} C_{ij\sigma} G^*_{ij} = r_\sigma, \qquad \sigma = 1, \ldots, m, \tag{9.1}$$

where the coefficients $\{C_{ij\sigma}\}$ and the quantities $\{r_\sigma\}$ are known in terms of values of x and of f. $\{G^*_{ij}\}$ is, of course, required to be symmetric. With these more general QN conditions, for example, it might not be necessary to achieve the conditions (2.6) in the line search, thus rendering it possible to reduce the number of evaluations of f.

[9] Or, with our notation for the Hessian, "G-conjugate".

TABLE 6

ROSENBROCK'S FUNCTION

```
CYCLE EVALS      F              X→
  0      3    2.4200E01     ¯1.2000E00    1.0000E00
  1     10    4.3754E00     ¯1.0098E00    1.0776E00
  2     16    3.4680E00     ¯7.7971E¯01   5.5312E¯01
  3     20    3.4240E00     ¯8.3303E¯01   7.1924E¯01
  4     24    3.1383E00     ¯7.5679E¯01   5.9554E¯01
  5     29    2.1103E00     ¯4.3391E¯01   2.1156E¯01
  6     34    1.8595E00     ¯3.6362E¯01   1.3299E¯01
  7     39    1.3154E00     ¯1.4380E¯01   2.9102E¯02
  8     45    1.0228E00      1.6316E¯02  ¯2.3224E¯02
  9     52    8.0424E¯01     1.1630E¯01  ¯1.7436E¯03
 10     57    4.6536E¯01     3.5187E¯01   1.4509E¯01
 11     62    3.1257E¯01     5.0231E¯01   2.2685E¯01
 12     68    2.5411E¯01     5.0955E¯01   2.4799E¯01
 13     74    2.3928E¯01     5.1840E¯01   2.6017E¯01
 14     80    1.4731E¯01     6.2616E¯01   3.8338E¯01
 15     86    4.3617E¯02     7.9123E¯01   6.2659E¯01
 16     92    3.6691E¯02     8.4213E¯01   6.9833E¯01
 17     99    2.9450E¯02     8.5294E¯01   7.1866E¯01
 18    105    1.2068E¯02     8.9858E¯01   8.0323E¯01
 19    110    1.8136E¯03     9.6289E¯01   9.2507E¯01
 20    115    7.0522E¯05     9.9808E¯01   9.9535E¯01
 21    120    5.2966E¯06     1.0021E00    1.0042E00
 22    124    5.1426E¯07     1.0000E00    1.0001E00
 23    128    8.1359E¯09     9.9993E¯01   9.9987E¯01
 24    132    3.8825E¯11     1.0000E00    1.0000E00
 25    136    3.6684E¯15     1.0000E00    1.0000E00
 26    140    7.1632E¯20     1.0000E00    1.0000E00
      CONVERGED
 27    144    4.6940E¯25     1.0000E00    1.0000E00

GNORM,STEP    9.6865E¯13 3.7896E¯10


     GG

  801.99 ¯399.99
 ¯399.99  200


     GGDIF

  802 ¯400
 ¯400  200
```

Powell reported mixed success with an algorithm he devised based on these ideas. His difficulties seemed to be a result of the lack of insurance, in conditions (9.1), that G^* would be positive-definite. Moreover, Powell made no provision for estimating g_0^*. If (9.1) is generalized further to:

$$\sum_{ij} C_{ij\sigma} G_{ij}^* + \sum_i d_{i\sigma} g_{0i}^* = q_\sigma, \tag{9.2}$$

thus introducing more variables $\{g_{0i}^*\}$, it would then be possible to constrain G^* so as to maintain positive-definiteness, while at the same time having the QN conditions (9.2) strictly satisfied. This might be done along the lines of Section 6 (also suitably generalized); i.e., some norm of γ would be minimized, subject to a set of inequality constraints on G^*. The exact QN conditions would then be used to complete the solution for the updates.

TABLE 7

```
  POWELL'S FUNCTION

CYCLE EVALS       F              X+
  0      5     2.1500E02       3.0000E00  ¯1.0000E00   0.0000E00   1.0000E00
  4     51     4.0171E¯04      6.4978E¯02 ¯7.3758E¯03  6.1345E¯02  6.4117E¯02
  8     90     6.9257E¯07      4.0372E¯03 ¯3.5742E¯04  1.2360E¯02  1.2275E¯02
 12    123     9.3623E¯08     ¯4.4864E¯03  4.4668E¯04  5.0931E¯03  5.0853E¯03
 16    168     6.7213E¯08     ¯2.9309E¯03  2.9034E¯04  5.5437E¯03  5.5250E¯03
 20    206     2.1248E¯10     ¯4.8166E¯04  4.8600E¯05  1.0142E¯03  1.0193E¯03
 24    242     4.4252E¯13     ¯4.4901E¯04  4.4846E¯05 ¯1.3276E¯04 ¯1.3284E¯04
 28    284     2.3800E¯16     ¯4.3286E¯05  4.3292E¯06  2.3392E¯05  2.3392E¯05
 32    319     2.9111E¯18     ¯2.9425E¯05  2.9425E¯06 ¯1.6880E¯05 ¯1.6881E¯05
 36    358     9.2835E¯19     ¯2.4931E¯05  2.4931E¯06 ¯1.2547E¯05 ¯1.2547E¯05
 40    393     5.9539E¯22     ¯7.7086E¯07  7.7085E¯08  1.0337E¯06  1.0337E¯06
 44    436     6.5258E¯23     ¯7.8105E¯09  7.8120E¯10  1.1271E¯06  1.1271E¯06
 48    490     1.6951E¯24     ¯4.2223E¯07  4.2223E¯08  1.6984E¯07  1.6983E¯07
    CONVERGED
**ORTHOGONALITY FAILURES    9
 49    501     1.2448E¯24     ¯4.4294E¯07  4.4294E¯08  1.4176E¯07  1.4176E¯07

GNORM,STEP   8.6802E¯13 7.63E¯13


    GG

  0.82669    8.267      1.8316    ¯1.8316
  8.267     82.671     18.317    ¯18.317
  1.8316    18.317     16.94     ¯16.94
 ¯1.8316   ¯18.317    ¯16.94      16.94


  GGDIF

  2.0000E0      2.0000E1      5.0487E¯17 ¯1.2102E¯10
  2.0000E1      2.0000E2     ¯4.1374E¯11  8.0779E¯16
  5.0487E¯17  ¯4.1374E¯11  1.0000E1    ¯1.0000E1
 ¯1.2102E¯10  8.0779E¯16  ¯1.0000E1     1.0000E1
```

10. Acknowledgments. I am indebted to M. J. D. Powell, S. Schechter, and G. Golub for provocative criticisms and suggestions (some of which have already been mentioned).

Appendix—Line Search. We shall sketch the line search here, touching on the principal precaution for avoiding catastrophes due to rounding error. (There are various other safeguards in the program, but these have little theoretical interest.)

The first phase of the search we term the "trap" phase. Starting with a normalized direction vector s, we are evaluating $F(\alpha)$ defined as $f(\tau + \alpha s)$ as described in Section 2. Our first value ($\alpha = 0$), we shall denote by α_2, and the corresponding value of $F(0)$ by F_2. We then increment α to the value α_3, and evaluate F_3. (If s is the first step direction—viz., the Newton direction, then α_3 is the value given by the Newton formula; however, in no case is α_3 permitted to exceed unity. For the other directions in the cycle, α_3 is estimated on the basis of the progress made in the first step—again, α_3 cannot exceed unity.)

If $F_3 < F_2$, the step α_3 is doubled, α_2 becomes α_1, α_3 becomes α_2, and a new α_3 is defined as $\alpha_2 + 2(\alpha_2 - \alpha_1)$. The function values are also relabeled. F_3 is next evaluated, and compared with F_2. If $F_3 < F_2$ another progressive step is made, etc. For some α_3, F_3 will be $\geqslant F_2$. In this case, we have "trapped" a smallest value of F.

Now, it can happen that, although G is positive-definite, even the "Newton direction" may not be a descending one, because we have only an *estimate* of the gradient, and not its true value. Hence, it can always happen that the initial F_3 is $\geqslant F_2$. In this case, we reverse the signs of s and α_3, denote α_3 and F_3 by α_1 and F_1, respectively, and make a new step α_3 in the opposite direction. The new F_3 may be $< F_2$, in which case, we proceed as in the preceding paragraph. Otherwise, we again have "trapped" the smallest value F_2.

Under certain circumstances, this would end the line search. However, there may be certain unsatisfactory conditions that necessitate a more refined "squeeze" of the middle point (α_2, F_2).[10] These are:

(a) It is the first step of the cycle and $\alpha_2 = 0$. (This might result in a null step for the entire cycle, thus unnecessarily terminating the algorithm.)

(b) The estimate of c gained from the α's and F's via the method of Section 3 exceeds 10. (Because of the normalization of the $\{s_i\}$, the value of c becomes very nearly unity near the solution. For this reason, large estimates of c are suspect, since a very bad value for c can render it very difficult or impossible to recover good estimates g and G during later cycles.)

The "squeeze" itself is based on first fitting a quadratic to the three points P_1, P_2, and P_3. The minimum of this quadratic will occur at α_4, with $\alpha_1 < \alpha_4 < \alpha_3$. When F_4 is now evaluated it may be $\geqslant F_2$. In this case, we perform a "cut", i.e., if, for example, $\alpha_4 > \alpha_2$, we compute[11]

$$\alpha_5 = \tfrac{1}{2}(\alpha_1 + \alpha_2) \tag{A1}$$

and "close" the interval, by discarding P_3. Then P_4 becomes P_3, and we evaluate F_5; P_5 then becomes the new P_4. If F_4 is again $> F_2$, we repeat the process. Note that the "cut" is always on the side away from P_4.

When $F_4 < F_2$, we fit a cubic to the four points P_1, P_2, P_3, P_4. Let this cubic be centered around α_4 as follows:

$$\kappa(\alpha) = c_0 + c_1(\alpha - \alpha_4) + c_2(\alpha - \alpha_4)^2 + c_3(\alpha - \alpha_4)^3 \tag{A2}$$

(with the c's having known values after the fitting). We can then solve for the minimum of $\kappa(\alpha)$, and we obtain the solution (for $c_2 \neq 0$):

$$\alpha_5 = \alpha_4 - \frac{c_1}{c_2}\left(\frac{1}{1 + \sqrt{1 - \rho}}\right), \tag{A3}$$

where

$$\rho \equiv 3c_1 c_3 / c_2^2, \tag{A4}$$

ρ is a dimensionless ratio, independent of the scaling of F or α, and, for a cubic, is bounded above by unity.

[10] We shall henceforth denote the pair (α_i, F_i) by P_i.

[11] This device was originally suggested to the author by Dr. Y. Bard.

The criterion for terminating the "squeeze" is based on the relative change in the estimated value of $F''(\alpha)$ from α_4 to α_5. In this case, the estimates are based on $\kappa(\alpha)$, and the values of κ'' at α_4 and α_5 turn out to be:

(A5a) $$\kappa''(\alpha_4) = 2c_2,$$

(A5b) $$\kappa''(\alpha_5) = 2c_2\sqrt{1 - \rho},$$

so that:

(A6) $$\kappa_5''/\kappa_4'' = \sqrt{1 - \rho}.$$

It can be shown that, when the values $\{\alpha_1, \alpha_3\}$ do not bracket the *maximum* of $\kappa(\alpha)$, then κ_5'' will be larger than κ_4''. Hence, we can expect that the "normal" state of affairs would be that the ratio in (A6) would be greater than unity, which means that ρ would be negative. Numerical tests have indicated that it is in fact reasonable to allow κ'' to increase by 20% but to restrict any decrease to 1%. This gives an allowable range for ρ as follows:

(A7) $$-.44 < \rho < .02$$

and when ρ is found to fall within this range, the squeeze is terminated.

The principal danger from rounding error occurs when the differences $(F_1 - F_2)$ and $(F_3 - F_2)$ are too small relative to $|F_2|$. Then, too many significant figures are lost, and the values of b and c become too inaccurate. This has the effect of spoiling the updates for g and G. Therefore, since the machine accuracy in this study is about 16 significant figures, the line search is terminated and no update is made when,

(A8) $$\min(F_1 - F_2, F_3 - F_2) < 10^{-12} \times F_2$$

so that we can expect at least a few correct figures in our update.

IBM Corporation
Palo Alto Scientific Center
1530 Page Mill Road
Palo Alto, California 94304

1. J. GREENSTADT, *Math. Comp.*, vol. 26, 1972, p. 145.

2. M. MARCUS & H. MINC, *A Survey of Matrix Theory and Matrix Inequalities*, Allyn and Bacon, Boston, Mass., 1964.

3. R. FLETCHER & M. J. D. POWELL, *Comput. J.*, vol. 6, 1963, p. 163.

4. H. H. ROSENBROCK, *Comput. J.*, vol. 3, 1960, p. 175.

5. A. R. COLVILLE, *A Comparative Study of Non-Linear Programming Codes*, IBM NY Scientific Center Report #320–2949, 1968.

6. M. J. D. POWELL, *Comput. J.*, vol. 5, 1962, p. 147.

7. R. P. BRENT, *Algorithms for Finding Zeros and Extrema of Functions Without Calculating Derivatives*, Stanford Univ. Comput. Sci. Report STAN-CS-71-198, 1971.

8. R. FLETCHER, *Comput. J.*, vol. 8, 1965, p.33.

9. M. C. BIGGS, *J. Inst. Math. Appl.*, vol. 8, 1972, p. 315.

10. P. E. GILL, W. MURRAY & R. A. PITFIELD, *The Implementation of Two Revised Quasi-Newton Algorithms for Unconstrained Optimization*, Nat. Phys. Lab. Report NAC 11, 1972.

11. M. J. D. POWELL, *Comput. J.*, vol. 7, 1964, p. 155.

12. M. J. D. POWELL, *ACM Trans. Math. Software*, vol. 1, 1975, p. 97.