SIAM J. NUMER. ANAL.
Vol. 22, No. 1, February 1985

A FAMILY OF TRUST-REGION-BASED ALGORITHMS FOR UNCONSTRAINED MINIMIZATION WITH STRONG GLOBAL CONVERGENCE PROPERTIES*

GERALD A. SHULTZ†, ROBERT B. SCHNABEL† AND RICHARD H. BYRD†

Abstract. This paper has two aims: to exhibit very general conditions under which members of a broad class of unconstrained minimization algorithms are globally convergent in a strong sense, and to propose several new algorithms that use second derivative information and achieve such convergence. In the first part of the paper we present a general trust-region-based algorithm schema that includes an undefined step selection strategy. We give general conditions on this step selection strategy under which limit points of the algorithm will satisfy first and second order necessary conditions for unconstrained minimization. Our algorithm schema is sufficiently broad to include line search algorithms as well. Next, we show that a wide range of step selection strategies satisfy the requirements of our convergence theory. This leads us to propose several new algorithms that use second derivative information and achieve strong global convergence, including an indefinite line search algorithm, several indefinite dogleg algorithms, and a modified "optimal-step" algorithm. Finally, we propose an implementation of one such indefinite dogleg algorithm.

1. Introduction. In this paper we discuss the convergence properties of a broad class of algorithms for the unconstrained minimization problem

$$\min_{x \in R^n} f(x) : R^n \to R \tag{1.1}$$

where it is assumed that f is twice continuously differentiable. The algorithms discussed are of the trust region type, but the algorithm schema used is sufficiently general that our convergence results apply to many algorithms of the line search type as well.

In the first part of the paper we give a general condition under which the limit points of a broad class of trust region algorithms satisfy the first order necessary conditions for problem (1.1). In this paper we shall call such an algorithm "first order stationary point convergent." At the same time, we give a general condition that shows how the limit points of these algorithms may satisfy the second order necessary conditions for (1.1) by incorporating second order information. We shall refer to such an algorithm as "second order stationary point convergent."

In the second part of the paper, we show that many algorithms satisfy these conditions for first and second order stationary point convergence, and we suggest several new algorithms that use second order information.

The convergence results presented here are a generalization of those given by Sorensen [1982]. Sorensen proves strong convergence properties for a specific trust region algorithm, which uses second order information. Related results are found in Fletcher [1980] and Moré and Sorensen [1981]. Others, including Fletcher and Freeman [1977], Goldfarb [1980], Kaniel and Dax [1979], McCormick [1977], Moré and Sorensen [1979], Mukai and Polak [1978], and Vial and Zang [1975], have discussed and proven the second order stationary point convergence of algorithms that use second order information but are not of the trust region type. Powell [1970], [1975] and Thomas [1975], on the other hand, discuss the first order stationary point convergence properties of a class of trust region algorithms.

In § 2 we define our general algorithm schema, state the conditions for the types of convergence mentioned above, and prove the convergence results. In § 3 we take

* Received by the editors April 21, 1982, and in final revised form January 11, 1984.
† Department of Computer Science, University of Colorado, Boulder, Colorado 80309.

the first step toward showing the applicability of the class of algorithms by commenting that practically all trust radius adjusting strategies in use fit into our algorithm schema. In §§ 4 and 5 we further show the meaning of the schema by discussing a variety of different types of step selection strategies that satisfy the conditions given in § 2. Finally in § 6 we propose an implementation of one of these, an "indefinite dogleg" algorithm.

In the remainder of the paper we use the following notation:

$\|\cdot\|$ is the Euclidean norm.

$g(x) \in R^n$ is the gradient of f evaluated at x.

$H(x) \in R^{n\times n}$ is the Hessian of f evaluated at x.

$\{x_k\}$ is a sequence of points generated by an algorithm, and $f_k = f(x_k)$, $g_k = g(x_k)$, and $H_k = H(x_k)$.

$\lambda_1(B)$ and $\lambda_n(B)$ are the smallest and largest eigenvalues, respectively, of the symmetric matrix B.

$[u_1, \cdots, u_m]$ is the subspace of R^n spanned by the vectors $u_1, \cdots, u_m$.

2. Global convergence of a general trust region algorithm. In this section we describe a class of trust region algorithms in a way that includes most trust region algorithms, as well as many other algorithms, and that isolates conditions sufficient for them to have various convergence properties.

The form of most existing trust region algorithms is basically as follows. The algorithm generates a sequence of points x_k. At the kth iteration, it forms a quadratic model of the objective function about x_k,

$$\psi_k(w) = f_k + g_k^T w + \tfrac{1}{2} w^T B_k w,$$

where $w \in R^n$ and $B_k \in R^{n\times n}$ is some symmetric matrix, and finds an initial value for the trust radius, Δ_k. Then a "minor iteration" is performed, possibly repeatedly. The minor iteration consists of using the current trust radius Δ_k and the information contained in the quadratic model to compute a step

$$p_k(\Delta_k) = p(g_k, B_k, \Delta_k),$$

and then comparing the actual reduction of the objective function

$$\text{ared}_k(\Delta_k) = f_k - f(x_k + p_k(\Delta_k))$$

to the reduction predicted by the quadratic model

$$\text{pred}_k(\Delta_k) = f_k - \psi_k(p_k(\Delta_k)).$$

If the reduction is satisfactory, then the step can be taken, or a larger trust region tried. Otherwise the trust region is reduced and the minor iteration is repeated.

Three aspects of this algorithm are unspecified, namely how to form the matrix B_k for the quadratic model, how the step computing function $p(g, B, \Delta)$ is performed on each minor iteration, and how the trust radius Δ_k is adjusted. In our abstract definition of a trust region algorithm below, the minor iterations and the strategy for adjusting the trust region are replaced by a condition that the step and trust radius must satisfy upon quitting the major iteration. This allows the description to cover a wide variety of trust region strategies. The methods of computing B_k and $p(g, B, \Delta)$ are left unspecified, since we later want to give conditions on these quantities that ensure the convergence properties. For our abstract definition of a trust region algorithm it is enough to know that they are computed in such a way that the algorithm is well defined.

We now define the general trust region algorithm:

ALGORITHM 2.1.

(0) Choose $\gamma_1, \eta_1, \eta_2 \in (0, 1)$, $x_1 \in R^n$, $\Delta_0 > 0$, and let $k = 1$.

(1) Compute $f_k = f(x_k)$, $g_k = g(x_k)$, symmetric $B_k \in R^{n \times n}$.

(2) Find Δ_k and compute $p_k = p_k(\Delta_k)$ satisfying:
$\| p_k \| \leqq \Delta_k$ and

(a) $\dfrac{\text{ared}_k(\Delta_k)}{\text{pred}_k(\Delta_k)} \geqq \eta_1$ and

(b) either $\Delta_k \geqq \Delta_{k-1}$, or
$\Delta_k \geqq \| B_{k-1}^{-1} g_{k-1} \|$ with B_{k-1} positive definite, or
for some $\Delta \leqq \dfrac{1}{\gamma_1}\Delta_k$, $\dfrac{\text{ared}_k(\Delta)}{\text{pred}_k(\Delta)} < \eta_2$ or $\dfrac{\text{ared}_{k-1}(\Delta)}{\text{pred}_{k-1}(\Delta)} < \eta_2$.

(3) Let $x_{k+1} = x_k + p_k$ and $k = k + 1$.

(4) Go to (1).

Again, note that the computations of B_k, $p_k(\Delta)$, and Δ_k are left unspecified. In Theorem 2.2 we give conditions on B_k and $p(g, B, \Delta)$ that yield various convergence properties. In § 3 we will discuss a number of trust radius adjusting strategies that satisfy the requirements in Algorithm 2.1, step (2).

Now we set forth conditions which the step computing function $p(g, B, \Delta)$ may satisfy and prove that if it meets these conditions then the convergence results follow. In §§ 4 and 5 we will discuss various step computing algorithms that fulfill the conditions below.

The first condition says that the step must give sufficient decrease of the quadratic model. The second condition requires that when $H(x)$ is indefinite the step give as good a decrease of the quadratic model as a direction of sufficient negative curvature. The third condition simply says that if the Hessian is positive definite and the Newton step lies within the trust region, then the Newton step is chosen.

Before stating the conditions we define some additional notation.

$$\text{pred}(g, B, \Delta) = -g^T p(g, B, \Delta) - \tfrac{1}{2} p(g, B, \Delta)^T B p(g, B, \Delta).$$

Our conditions that a step selection strategy may satisfy are:

Condition 1. There are $\bar{c}_1, \sigma_1 > 0$ such that for all $g \in R^n$, for all symmetric $B \in R^{n \times n}$, and for all $\Delta > 0$, $\text{pred}(g, B, \Delta) \geqq \bar{c}_1 \| g \| \min \{\Delta, \sigma_1 \| g \| / \| B \|\}$.

Condition 2. There is a $\bar{c}_2 > 0$ such that for all $g \in R^n$, for all symmetric $B \in R^{n \times n}$, and for all $\Delta > 0$, $\text{pred}(g, B, \Delta) \geqq \bar{c}_2(-\lambda_1(B))\Delta^2$.

Condition 3. If B is positive definite and $\| -B^{-1} g \| \leqq \Delta$, then $p(g, B, \Delta) = -B^{-1} g$.

We now state and prove the convergence theorem. The proofs are similar to those of Sorensen [1982]. Conditions 1, 2, and 3 constitute a major generalization of the properties of the optimal step

$$p(g, B, \Delta) = \text{argmin} \{ g^T w + \tfrac{1}{2} w^T B w : \| w \| \leqq \Delta \}$$

that is assumed by Fletcher [1980]. They easily cover Sorensen's assumption that

$$p(g, B, \Delta) = \text{argmin} \{ \tilde{g}^T w + \tfrac{1}{2} w^T B w : \| w \| \leqq \tilde{\Delta} \}$$

where $\| \tilde{g} - g \| \leqq \varepsilon_1 \| g \|$ and $| \tilde{\Delta} - \Delta | \leqq \varepsilon_2 \Delta$. Conditions 1 and 2 can be shown to be a strict relaxation of Moré and Sorensen's conditions that pred (g, B, Δ) be at least some

fixed fraction of the optimal decrease of the quadratic model. In § 3 we show that a slight modification to Algorithm 2.1 leads to a further strengthening, namely that under the assumptions of part III of Theorem 2.2, $H(x_*)$ is positive semi-definite at any limit point x_*.

Our first order results extend the results of Powell [1970] and Thomas [1975], except that they do not include the possibility of shrinking the trust region to a smaller step size when B_k is indefinite and the trust region acceptance test in (2a) of Algorithm 2.1 is satisfied. This does not occur in modern trust region algorithms. The first order results are a minor generalization of Powell [1975].

THEOREM 2.2. *Let $f: R^n \to R$ be twice continuously differentiable and bounded below, and let $H(x)$ satisfy $\|H(x)\| \leqq \beta_1$ for all $x \in R^n$. Suppose that an algorithm satisfying the conditions of Algorithm* 2.1 *is applied to $f(x)$, starting from some $x_1 \in R^n$, generating a sequence $\{x_k\}$, $x_k \in R^n$, $k = 1, 2, \cdots$. Then:*

I. *If $p(g, B, \Delta)$ satisfies Condition* 1 *and $\|B_k\| \leqq \beta_2$ for all k, then g_k converges to* 0 *(first order stationary point convergence).*

II. *If $p(g, B, \Delta)$ satisfies Conditions* 1 *and* 3, *$B_k = H(x_k)$ for all k, $H(x)$ is Lipschitz continuous with constant L, and x_* is a limit point of $\{x_k\}$ with $H(x_*)$ positive definite, then x_k converges q-quadratically to x_*.*

III. *If $p(g, B, \Delta)$ satisfies Conditions* 1 *and* 2, *$B_k = H(x_k)$ for all k, and x_k converges to x_*, then $H(x_*)$ is positive semi-definite (second order stationary point convergence, with I).*

Proof. Each of the three parts uses the following:

By Taylor's theorem, for any k and any $\Delta > 0$,

$$\begin{aligned}|\text{ared}_k(\Delta) - \text{pred}_k(\Delta)| &= |f_k - f(x_k + p_k(\Delta)) - (f_k - f_k - g_k^T p_k(\Delta) - \tfrac{1}{2} p_k(\Delta)^T B_k p_k(\Delta))| \\ &= \left| \tfrac{1}{2} p_k(\Delta)^T B_k p_k(\Delta) - \int_0^1 p_k(\Delta)^T H(x_k + \xi p_k(\Delta)) p_k(\Delta)(1-\xi)\, d\xi \right| \\ &\leqq \|p_k(\Delta)\|^2 \int_0^1 \|B_k - H(x_k + \xi p_k(\Delta))\|(1-\xi)\, d\xi.\end{aligned}$$

So,

$$\left| \frac{\text{ared}_k(\Delta)}{\text{pred}_k(\Delta)} - 1 \right| \leqq \frac{\|p_k(\Delta)\|^2 \int_0^1 \|B_k - H(x_k + \xi p_k(\Delta))\|(1-\xi)\, d\xi}{|\text{pred}_k(\Delta)|}. \tag{2.1}$$

All three parts proceed by using the relevant hypotheses and the above argument to bound $\text{pred}_k(\Delta)$ below by a term that is $O(\Delta^2)$.

Proof of I. Consider any m with $\|g_m\| \neq 0$. For any x, $\|g(x) - g_m\| \leqq \beta_1 \|x - x_m\|$, so if $\|x - x_m\| < \|g_m\|/2\beta_1$, then

$$\|g(x)\| \geqq \|g_m\| - \|g(x) - g_m\| \geqq \frac{\|g_m\|}{2}.$$

Let $R = \|g_m\|/2\beta_1$ and $B_R = \{x: \|x - x_m\| < R\}$.

Now, there are two possibilities. Either for all $k \geqq m$, $x_k \in B_R$, or eventually $\{x_k\}$ leaves the ball B_R. It turns out that the sequence cannot stay in the ball. If $x_k \in B_R$ for all $k \geqq m$, then for all $k \geqq m$, $\|g_k\| \geqq \|g_m\|/2$, which we shall denote by ε. Thus, by Condition 1,

$$\text{pred}_k(\Delta) \geqq \sigma \|g_k\| \min\left\{\Delta, \frac{\|g_k\|}{\|B_k\|}\right\} \geqq \sigma\varepsilon \min\left\{\Delta, \frac{\varepsilon}{\beta_2}\right\}$$

for all $k \geqq m$, where $\sigma = \bar{c}_1 \min\{1, \sigma_1\}$ is used to simplify the notation. So,

$$\left|\frac{\text{ared}_k(\Delta)}{\text{pred}_k(\Delta)} - 1\right| \leqq \frac{\Delta^2 \int_0^1 \|B_k - H(x_k + \xi p_k(\Delta))\|(1-\xi)\, d\xi}{\sigma\varepsilon \min\{\Delta, \varepsilon/\beta_2\}}$$
$$\leqq \frac{\Delta^2(\beta_1+\beta_2)}{\sigma\varepsilon \min\{\Delta, \varepsilon/\beta_2\}} \leqq \frac{\Delta(\beta_1+\beta_2)}{\sigma\varepsilon}$$

for all $k \geqq m$ and $\Delta \leqq \varepsilon/\beta_2$. Thus for Δ sufficiently small and $k \geqq m$

$$\frac{\text{ared}_k(\Delta)}{\text{pred}_k(\Delta)} > \eta_2.$$

In addition, $\|B_k^{-1} g_k\| \geqq \|g_k\|/\|B_k\| \geqq \|g_m\|/2\beta_2$, so that for Δ_k sufficiently small none of the conditions allowing decreasing of Δ_k in (2b) of Algorithm 2.1 can hold. Therefore Δ_k is bounded away from zero. But, since

$$f_k - f_{k+1} = \text{ared}_k(\Delta_k) \geqq \eta_1 \text{pred}_k(\Delta_k) \geqq \eta_1 \sigma\varepsilon \min\left\{\Delta_k, \frac{\varepsilon}{\beta_2}\right\},$$

and f is bounded below, Δ_k converges to 0, which is a contradiction. Hence, eventually $\{x_k\}$ must be outside B_R for some $k > m$.

Let $l+1$ be the first index after m with x_{l+1} not in B_R. Then

$$f(x_m) - f(x_{l+1}) = \sum_{k=m}^{l} f(x_k) - f(x_{k+1})$$
$$\geqq \sum_{k=m}^{l} \eta_1 \text{pred}_k(\Delta_k) \geqq \sum_{k=m}^{l} \eta_1 \sigma\varepsilon \min\left\{\Delta_k, \frac{\varepsilon}{\beta_2}\right\}.$$

Now, if $\Delta_k \leqq \varepsilon/\beta_2$ for $m \leqq k \leqq l$, we have that $f(x_m) - f(x_{l+1}) \geqq \eta_1 \sigma\varepsilon \sum_{k=m}^{l} \Delta_k \geqq \eta_1 \sigma\varepsilon R$. Otherwise, we have that $f(x_m) - f(x_{l+1}) \geqq \eta_1 \sigma(\varepsilon^2)/\beta_2$. In either case,

$$f(x_m) - f(x_{l+1}) \geqq \eta_1 \sigma\varepsilon \min\left\{R, \frac{\varepsilon}{\beta_2}\right\}$$
$$= \eta_1 \sigma \frac{\|g_m\|}{2} \min\left\{\frac{\|g_m\|}{2\beta_1}, \frac{\|g_m\|}{2\beta_2}\right\} \geqq \|g_m\|^2 \eta_1 \frac{\sigma}{4} \min\left(\frac{1}{\beta_1}, \frac{1}{\beta_2}\right).$$

Now, since f is bounded below and $\{f(x_k)\}$ is monotonically decreasing, $\{f(x_k)\}$ converges to some limit, say f_*. Then by the above, for any k

$$\|g_k\|^2 \leqq \left(\eta_1 \frac{\sigma}{4} \min\left\{\frac{1}{\beta_1}, \frac{1}{\beta_2}\right\}\right)^{-1} (f(x_k) - f_*).$$

Thus since $\{f(x_k)\} \to f_*$, $\|g_k\| \to 0$.

Proof of II. By assumption, x_* is a limit point, say x_{k_j} converges to x_*. We will show first that x_k converges to x_*. By I, $g(x_*) = 0$. Since $H(x_*)$ is positive definite and H is continuous, we can find $\delta_1 > 0$ such that if $\|x - x_*\| < \delta_1$, then $H(x)$ is positive definite, and if $x \neq x_*$ then $g(x) \neq 0$. Let $B_1 = \{x: \|x - x_*\| < \delta_1\}$.

Since $g(x_*) = 0$, we can find $\delta_2 > 0$ with $\delta_2 < \delta_1/4$ and $\|H(x)^{-1} g(x)\| < \delta_1/2$ for all $x \in B_2 = \{x: \|x - x_*\| < \delta_2\}$.

Find j_0 such that $f(x_{k_{j_0}}) < \inf\{f(x): x \in B_1 - B_2\}$ and $x_{k_{j_0}} \in B_2$. Consider any x_l with $l \geqq k_{j_0}$, $x_l \in B_2$. We claim that $x_{l+1} \in B_2$, which implies that the entire sequence beyond

$x_{k_{j_0}}$ is in B_2. If x_{l+1} is not in B_2, then since $f_{l+1} < f_{k_{j_0}}$, x_{l+1} is not in B_1, either, so

$$\Delta_l \geq \|x_{l+1} - x_l\| \geq \|x_{l+1} - x_*\| - \|x_l - x_*\| \geq \delta_1 - \frac{\delta_1}{4} = \frac{3}{4}\delta_1 > \frac{\delta_1}{2} \geq \|B(x_l)^{-1} g(x_l)\|.$$

But, since the Newton step from x_l is within the trust region, by Condition 3, $p_l(\Delta_l) = -H(x_l)^{-1} g(x_l)$. But then since $\|p_l(\Delta_l)\| < \delta_1$, $x_{l+1} \in B_1$, which is a contradiction.

Thus for all $k \geq k_{j_0}$, $x_k \in B_2$, and so since $f(x_k)$ is a strictly decreasing sequence and x_* is the unique minimizer of f in B_2, we have that x_k converges to x_*.

Now, to show that the convergence rate is quadratic, we will show that eventually $\|H(x_k)^{-1} g(x_k)\|$ will always be less than Δ_k, and hence by Condition 3, the Newton step will always be taken. Then since $H(x_*)$ is nonsingular and H is Lipschitz continuous the quadratic convergence rate will follow.

To show that eventually the Newton step is always shorter than the trust radius, we need the appropriate lower bound on $\text{pred}_k(\Delta)$. By the assumptions of (II), for all k large enough, $B_k = H(x_k)$ is positive definite, so by Condition 3, either the Newton step is longer than the trust radius, or $p_k(\Delta)$ is the Newton step. In either case, $\|p_k(\Delta)\| \leq \|B_k^{-1} g_k\| \leq \|B_k^{-1}\| \, \|g_k\|$, so $\|g_k\| \geq \|p_k(\Delta)\| / \|B_k^{-1}\|$. By Condition 1,

$$\text{pred}_k(\Delta) \geq \sigma \|g_k\| \min\left\{\Delta, \frac{\|g_k\|}{\|B_k\|}\right\}.$$

Thus,

$$\text{pred}_k(\Delta) \geq \sigma \frac{\|p_k(\Delta)\|}{\|B_k^{-1}\|} \min\left\{\|p_k(\Delta)\|, \frac{\|p_k(\Delta)\|}{\|B_k^{-1}\| \, \|B_k\|}\right\} \geq \sigma \frac{\|p_k(\Delta)\|^2}{\|B_k^{-1}\|}.$$

So, by the continuity of H, for all k large enough,

$$\text{pred}_k(\Delta) \geq \frac{\sigma}{2} \frac{\|p_k(\Delta)\|^2}{\|H(x_*)^{-1}\|}.$$

Finally, note that by the argument leading up to (2.1) and Lipschitz continuity,

$$|\text{ared}_k(\Delta) - \text{pred}_k(\Delta)| \leq \|p_k(\Delta)\|^3 \frac{L}{2}.$$

Thus for any $\Delta > 0$ and k large enough,

$$\left|\frac{\text{ared}_k(\Delta)}{\text{pred}_k(\Delta)} - 1\right| \leq \frac{\|p_k(\Delta)\|^3 L/2}{\sigma \|p_k(\Delta)\|^2} 2\|H(x_*)^{-1}\|$$

$$= \frac{L\|H(x_*)^{-1}\|}{\sigma} \|p_k(\Delta)\| \leq \frac{L\|H(x_*)^{-1}\|}{\sigma} \Delta.$$

Thus, by step (2b) of Algorithm 2.1, there is a $\tilde{\Delta}$ such that if $\Delta_{k-1} < \tilde{\Delta}$ then Δ_k will be less than Δ_{k-1} only if $\Delta_k \geq \|B_{k-1}^{-1} g_{k-1}\|$. It follows from the quadratic convergence of Newton's method that for x_{k-1} close enough to x_* and k large enough, $\|B_k^{-1} g_k\| < \|B_{k-1}^{-1} g_{k-1}\|$. Now, if Δ_k is bounded away from 0 for all large k, then we are done. Otherwise, if for an arbitrarily large k Δ_k is reduced, i.e. $\Delta_k < \Delta_{k-1}$, then we have $\Delta_k \geq \|B_{k-1}^{-1} g_{k-1}\| > \|B_k^{-1} g_k\|$, so $\Delta_k > \|B_k^{-1} g_k\|$, and the full Newton step is taken. Inductively this occurs for all subsequence iterates and quadratic convergence follows.

Proof of III. Suppose to the contrary that $\lambda_1(H(x_*)) < 0$. There exists M such that if $k \geq M$, $\lambda_1(B_k) < \lambda_1(H(x_*))/2 < 0$. By Condition 2, for all $k \geq M$ and for all $\Delta > 0$,

$$\text{pred}_k(\Delta) \geq \bar{c}_2(-\lambda_1(B_k))\Delta^2 \geq \bar{c}_2(-\lambda_1(H(x_*))/2)\Delta^2.$$

Now, by (2.1)

$$\left|\frac{\text{ared}_k(\Delta)}{\text{pred}_k(\Delta)}-1\right| \leqq \frac{\|p_k(\Delta)\|^2}{\text{pred}_k(\Delta)} \int_0^1 \|H(x_k+\xi p_k(\Delta))-H(x_k)\|(1-\xi)\, d\xi.$$

Thus,

$$\left|\frac{\text{ared}_k(\Delta)}{\text{pred}_k(\Delta)}-1\right| \leqq \frac{\|p_k(\Delta)\|^2}{\Delta^2} \frac{\int_0^1 \|H(x_k+\xi p_k(\Delta))-H(x_k)\|(1-\xi)\, d\xi}{\bar{c}_2(-\lambda_1(H(x_*))/2)}.$$

Since this last quantity goes to 0 as Δ goes to 0 and since a Newton step is never taken for $k > M$, it follows from step (2b) of Algorithm 2.1 that for $k \geqq M$ and Δ_k sufficiently small, Δ_k cannot be decreased further. Thus, Δ_k is bounded below.

But

$$\text{ared}_k(\Delta_k) \geqq \eta_1 \text{pred}_k(\Delta_k) \geqq \eta_1 \bar{c}_2(-\lambda_1(H(x_*))/2)\Delta_k^2,$$

and since f is bounded below, $\text{ared}_k(\Delta_k)$ converges to 0, so Δ_k converges to 0, which is a contradiction. Hence $\lambda_1(H(x_*)) \geqq 0$. This concludes the proof of Theorem 2.2. □

The results of this theorem also apply to different shapes of trust region. Specifically we may wish to use a trust region defined by $\|D_k p\| \leqq \Delta$ for some nonsingular square matrix D_k such that $\|D_k\|$ and $\|D_k^{-1}\|$ are uniformly bounded in k. This satisfies the conditions of Algorithm 2.1 and Theorem 2.2, since if we make a change of variables replacing Δ by Δ times the upper bound on $\|D_k^{-1}\|$, then $\|p_k\| \leqq \Delta$, and the conditions otherwise do not involve $\|p\|$. The conditions are also not restricted to the Euclidean norm, and Theorem 2.2 applies as well to rectangular trust regions.

3. Some permissible trust region updating strategies. The conditions on the trust region radius Δ_k that we gave in step (2) of Algorithm 2.1 were chosen to be near minimal conditions that allow us to prove the results of Theorem 2.2. Obviously in implementing an algorithm involving trust regions, there are many detailed considerations in choosing and adjusting the trust region radius that we have not considered so far in this paper. Our purpose in Algorithm 2.1 was to set forth conditions that apply to almost any reasonable strategy. Here we indicate more specifically what types of strategies are covered.

Most approaches for choosing and adjusting the radius Δ_k follow the following general pattern. Iteration k of the algorithm begins with an initial trust radius which defines a step p. If this step is unsatisfactory, a sequence of smaller radii are tried until a satisfactory one is found. If the step p is satisfactory, it may be used or a larger trial trust region radius tried. At the next iterate $x_{k+1} = x_k + p_k$ and a new initial trust radius is generated. We consider now what form this pattern must take to satisfy the conditions we have placed on Algorithm 2.1.

To choose the initial trial radius at the kth iteration Algorithm 2.1 allows the possibility of making the initial trial radius larger than Δ_{k-1} by any method chosen, if that seems advantageous. However the initial trial radius may be decreased only in two circumstances determined by the previous step. In the first case, if the previous step failed the strengthened decrease condition, that is, if

$$\frac{\text{ared}_{k-1}(\Delta_{k-1})}{\text{pred}_{k-1}(\Delta_{k-1})} < \eta_2,$$

then the initial trial radius may be taken less than Δ_{k-1} as long as the ratio between the new trial radius and the previous Δ_{k-1} is bounded below by some positive constant that is fixed for the entire algorithm. This possibility is covered by the condition (b)

in step (2) of Algorithm 2.1. Second, if at the previous step the matrix B_{k-1} was positive definite, and the step taken was $p_{k-1} = -B_{k-1}^{-1} g_{k-1}$, which fell inside the trust region, then the initial trial radius is allowed to be set to any value greater than or equal to the length of p_{k-1}. This strategy is used, for example, by Powell [1975]. It is possible to generalize the proof to cover other ways of reducing the trust region radius but we believe we have covered most of those with a practical justification.

Increasing the trust region radius if the model proves to be good at the previous step is allowed but not required for the theory. It is clear, however, that doing this in some way would be advantageous in most cases. In fact if one does require that the trust region radius be increased when the model proves to be good, then one can actually improve the convergence results of Theorem 2.2. Specifically, part III can be strengthened to show that second order necessary conditions hold at any limit point of the iteration.

THEOREM 3.1. *Suppose Algorithm* 2.1 *is modified so that if whenever* $\text{ared}_{k-1}(\Delta_{k-1})/\text{pred}_{k-1}(\Delta_{k-1}) \geqq \eta_3$, *then at step k the quantity* $\text{ared}_k(\Delta)/\text{pred}_k(\Delta)$ *is computed for some* $\Delta \in [\gamma_2 \Delta_{k-1}, \gamma_3 \Delta_{k-1}]$, *and if the ratio is greater than* η_1 *then* $\Delta_k = \Delta$. *Here* $\gamma_3 > \gamma_2 > 1$, *and* $\eta_3 \in [\eta_1, 1)$. *Then Theorem* 2.2 *still holds with part* III *strengthened to say that if* x_* *is any limit point of the iteration, then* $H(x_*)$ *is positive semi-definite.*

Proof. Theorem 2.2 still holds since the modifications are allowed by Algorithm 2.1. To prove the strengthening of part III, suppose to the contrary that $\lambda_1(H(x_*)) < 0$. Then there exists $r > 0$ such that $\lambda_1(H(x_k)) < \lambda_1(H(x_*))/2 < 0$ for all x_k such that $\|x_k - x_*\| < r$. By Condition 2, for all such x_k and for all $\Delta > 0$,

$$\text{pred}_k(\Delta) \geqq \bar{c}_2(-\lambda_1(H(x_*))/2)\Delta^2,$$

and thus by (2.1)

$$\left|\frac{\text{ared}_k(\Delta)}{\text{pred}_k(\Delta)} - 1\right| \leqq \frac{\|p_k(\Delta)\|^2 \int_0^1 \|H(x_k) - H(x_k + \xi p_k(\Delta))\|(1-\xi)\, d\xi}{\bar{c}_2(-\lambda_1(H(x_*))/2)\Delta^2}.$$

Since this quantity goes to zero as Δ goes to zero, there exists $\bar{\Delta} > 0$ such that for all x_k within r of x_* and all $\Delta < \gamma_3 \bar{\Delta}$, $\text{ared}_k(\Delta)/\text{pred}_k(\Delta) \geqq \eta_3$. Therefore, for any x_{k-1} and x_k in this region with $\Delta_{k-1} < \bar{\Delta}$, a larger radius will be tried at step k and accepted, so $\Delta_k \geqq \gamma_2 \Delta_{k-1}$.

Now let x_m be an iterate such that $\|x_m - x_*\| < r/2$. Either the sequence will stay within the ball of radius r about x_* or it will leave it. If it stays within the ball, then for each $k > m$ the trust region radius will never become smaller than $\min\{\Delta_m, \bar{\Delta}\}$ and

$$f_{k-1} - f_k \geqq \eta_2 \bar{c}_2(-\lambda_1(H(x_*))/2)\Delta^2.$$

This is impossible since the sequence $\{f(x_k)\}$ is bounded below.

If the sequence leaves the ball, let x_{j+1} be the first iterate outside the ball. In that case

$$\frac{r}{2} \leqq \|x_{j+1} - x_m\| \leqq \sum_{k=m}^{j} \Delta_k.$$

If $\Delta_j < \bar{\Delta}$ then this quantity is less than

$$\sum_{k=m}^{j} \frac{\Delta_j}{\gamma_2^{j-k}} \leqq \frac{\gamma_2}{\gamma_2 - 1} \Delta_j.$$

Hence $\Delta_j \geqq \min\{\bar{\Delta}, r(\gamma_2 - 1)/2\gamma_2\}$. Therefore each time the sequences gets close to x_* and moves away, the objective function decreases by at least $\eta_2 \bar{c}_2(-\lambda_1(H(x_*))/2) \times$

$[\min\{\bar{\Delta}, r(\gamma_2-1)/2\gamma_2\}]^2$. This implies that the sequence can move from inside the ball of radius $r/2$ to outside the ball of radius r only a finite number of times. Since we have shown that it cannot stay within the larger ball indefinitely, then $\|x_k - x_*\| > r/2$ for all k sufficiently large, contradicting the assumption that x_* is a limit point with a strictly indefinite Hessian. □

Given the initial trial radius at the kth iteration, a sequence of trial radii may be tried until a satisfactory one is found. We only require that the trial radius be reduced when the previous trial step fails to satisfy the condition (a) in step (2) of Algorithm 2.1. The trust region may be reduced only for such a failure, and the reduction must be bounded below by a constant that is fixed for the entire algorithm. This is ensured by the condition

$$\frac{\text{ared}_k(\Delta)}{\text{pred}_k(\Delta)} < \eta_2$$

for some

$$\Delta \leqq \frac{1}{\gamma_1}\Delta_k$$

in Algorithm 2.1.

The conditions of Algorithm 2.1 also allow successively larger trial trust regions to be tried within the kth iteration whenever this seems advantageous. There is no restriction on the method used to increase the trial radius, nor on the amount of the increase, as long as the radius actually used satisfies condition (a) of step (2) in Algorithm 2.1. If a reasonable strategy for increasing the radius within an iteration is imposed, it can be shown that the convergence results of Theorem 2.2 can be strengthened in the same way as shown in Theorem 3.1. Of course from the point of view of efficiency, an intelligent strategy for increasing the trust region, either between steps or in the inner iteration, is essential.

4. Some permissible step selection strategies. In this section we present three lemmas describing useful conditions under which the step $p_k(\Delta)$ in Algorithm 2.1 will satisfy Conditions 1 and 2. Using these lemmas, we will see that a number of different methods for computing steps yield first and second order stationary point convergent trust region type algorithms.

First let us mention two types of step selection strategies that have been used in trust region algorithms to which we will refer.

The "optimal" trust region step selection strategy is to take

$$p_k(\Delta_k) = \text{argmin}\{f_k + g_k^T w + \tfrac{1}{2} w^T B_k w : \|w\| \leqq \Delta_k\}. \tag{4.1.}$$

Discussions and modifications of this strategy have been presented by many authors, see e.g. Gay [1981], Hebden [1973], Moré [1978], Moré and Sorensen [1981], and Sorensen [1982]. If B_k is positive definite and $\|B_k^{-1} g_k\| \leqq \Delta_k$, then $p_k = -B_k^{-1} g_k$ is the solution to (4.1). Otherwise, p_k satisfies $(B_k + \alpha_k I)p_k = -g_k$ for some nonnegative α_k such that $(B_k + \alpha_k I)$ is at least positive semi-definite and $\|p_k\| = \Delta_k$. If B_k is positive definite, then so is $(B_k + \alpha_k I)$ and

$$p_k = -(B_k + \alpha_k I)^{-1} g_k, \tag{4.2}$$

where α_k is uniquely determined by $\|p_k\| = \Delta_k$. If B_k has a negative eigenvalue, then p_k is still of the form (4.2) unless g_k is orthogonal to the null space of $(B_k - \lambda_1 I)$ and $\|(B_k - \lambda_1 I)^+ g_k\| < \Delta_k$; here the superscript + denotes the pseudoinverse and λ_1 denotes

the most negative eigenvalue of B_k. In this case, which Moré and Sorensen [1981] refer to as the "hard case," $p_k = -(B_k - \lambda_1 I)^+ g_k + \xi_k v_k$, where v_k is any eigenvector of B_k corresponding to the eigenvalue λ_1, and ξ_k is chosen so that $\| p_k \| = \Delta_k$. The lemmas of this section will lead to algorithms that are similar to this "optimal" algorithm and have the same convergence properties but are considerably easier to implement.

The second type of trust region step selection strategy includes the dogleg type algorithms of Dennis and Mei [1979], Powell [1970], and Thomas [1975]. These algorithms always choose $p_k \in [g_k, B_k^{-1} g_k]$ and thus seem mainly intended for the positive definite case, although Powell's algorithm does use a nonsingular Hessian approximation that may be indefinite. When $\Delta_k \geqq \| B_k^{-1} g_k \|$, p_k is the Newton step $-B_k^{-1} g_k$; when $\Delta_k \leqq \| g_k \|^3 / g_k^T B_k g_k \leqq \| B_k^{-1} g_k \|$, p_k is the steepest descent step of length Δ_k; when $\Delta_k \in (\| g_k \|^3 / g_k^T B_k g_k, \| B_k^{-1} g_k \|)$, p_k is the step of length Δ_k on a specified piecewise linear curve connecting $-\| g_k \|^2 / g_k^T B_k g_k g_k$ and $-B_k^{-1} g_k$ (see Dennis and Schnabel [1983] for further explanation). The lemmas of this section will lead to efficient extensions of these algorithms to the indefinite case. These extensions will satisfy the conditions of Theorem 2.2 for second order stationary point convergence.

The first lemma, a modification of Thomas [1975, Lemma 4.1], gives a very general condition that ensures satisfaction of Condition 1 and hence first order stationary point convergence. By way of motivation we note that if an algorithm simply took the "best gradient step," i.e. the solution to

$$\min \{ g_k^T w + \tfrac{1}{2} w^T B_k w : \| w \| \leqq \Delta, \ w \in [g_k] \},$$

then it would satisfy Condition 1. Lemma 4.1 is a slight generalization of this fact.

Here we slightly change our earlier notation and let

$$\text{pred}\,(s) = -g^T s - \tfrac{1}{2} s^T B s.$$

LEMMA 4.1. *Suppose there is a constant $c_1 \in (0, 1]$ such that at each iteration k,*

$$\text{pred}\,(p_k(\Delta)) \geqq -\min \{ g_k^T w + \tfrac{1}{2} w^T B_k w : \| w \| \leqq \Delta, \ w \in [d_k] \},$$

for some d_k satisfying

$$d_k^T g_k \leqq -c_1 \| d_k \| \, \| g_k \|.$$

Then $p_k(\Delta)$ satisfies Condition 1, *and hence a trust region algorithm using it is first order stationary point convergent.*

Proof. We will drop the subscripts k throughout and will show that

$$\text{pred}\,(s_*) \geqq \frac{c_1}{2} \| g \| \min \left\{ \Delta, \frac{c_1 \| g \|}{\| B \|} \right\},$$

where s_* solves the above minimization problem. This will clearly imply satisfaction of Condition 1 by $p(\Delta)$, since $\text{pred}\,(p(\Delta)) \geqq \text{pred}\,(s_*)$, by assumption.

Define $h(a) = -\text{pred}\,(ad) = a g^T d + (a^2/2)\, d^T B d$. Then $h'(a) = a d^T B d + g^T d$, and $h''(a) = d^T B d$.

Let $s_* = a_* d$, where a_* minimizes $h(a)$ subject to the constraint $a \| d \| \leqq \Delta$. Now, if $d^T B d > 0$, then either $a_* = -g^T d / d^T B d$, if $-g^T d / d^T B d \leqq \Delta / \| d \|$, or else $a_* = \Delta / \| d \|$. In the first case we have

$$\text{pred}\,(s_*) = \text{pred}\,(a_* d) = \frac{g^T d}{d^T B d} g^T d - \frac{1}{2} \left(\frac{g^T d}{d^T B d} \right)^2 d^T B d$$

$$= \frac{1}{2} \frac{(g^T d)^2}{d^T B d} \geqq \frac{1}{2} c_1^2 \frac{\| g \|^2 \| d \|^2}{d^T B d} \geqq \frac{1}{2} c_1^2 \frac{\| g \|^2}{\| B \|}.$$

In the second case, we have

$$\text{pred}(s_*) = -\frac{\Delta}{\|d\|} g^T d - \frac{1}{2}\frac{\Delta^2}{\|d\|^2} d^T B d \geqq -\frac{1}{2}\frac{\Delta}{\|d\|} g^T d$$

(with the inequality above true since $\Delta/\|d\| < -g^T d / d^T B d$)

$$\geqq \frac{c_1}{2}\Delta\|g\|.$$

Finally, if $d^T B d \leqq 0$, $a_* = \Delta/\|d\|$, and so we have

$$\text{pred}(s_*) = -\frac{\Delta}{\|d\|} g^T d - \frac{1}{2}\left(\frac{\Delta}{\|d\|}\right)^2 d^T B d \geqq -\frac{\Delta}{\|d\|} g^T d \geqq c_1 \Delta \|g\|.$$

Thus, s_* and hence $p(\Delta)$ satisfy Condition 1, with constants $\bar{c}_1 = c_1/2$ and $\sigma_1 = c_1$. □

We may summarize the lemma by saying that as long as an algorithm takes steps which do as well on the quadratic model as directions with "sufficient" descent, then Condition 1 is satisfied, and hence the algorithm is first order stationary point convergent.

Using Lemma 4.1, we can immediately note first order stationary point convergence for a number of algorithms. The lemma can be used to prove the first order stationary point convergence of many line search algorithms which keep the angle between the steps and the gradient bounded away from 90 degrees. In particular, most line search algorithms that require each step to decrease a local linear or quadratic model of the objective function are covered by our theory, because the trust region adjusting criteria can be shown to allow for most safeguarded steplength selection strategies. While a step acceptance criterion based on a quadratic model clearly is covered by Algorithm 2.1, a criterion based on a linear model also is covered by the permissible choice $B_k = 0$ at each iteration. Lemma 4.1 also applies to any dogleg type algorithm, e.g. Dennis and Mei [1979], Powell [1975], and Thomas [1975], since these algorithms always do at least as well as the "best gradient step." Finally, we note that the lemma applies immediately to the "optimal" algorithm described above, for the same reason.

The next lemma says, roughly, that if each step taken by the algorithm gives as much descent as a direction of sufficient negative curvature, when there is one, then Condition 2 is satisfied.

LEMMA 4.2. *Suppose there is a constant* $c_2 \in (0, 1]$ *such that at each iteration* k *where* $\lambda_1(H(x_k)) < 0$, *we have* $B_k = H(x_k)$ *and*

$$\text{pred}(p_k(\Delta)) \geqq \text{pred}(t_k),$$

where

$$t_k = \text{argmin}\{g_k^T w + \tfrac{1}{2} w^T B_k w : \|w\| \leqq \Delta,\ w \in [q_k]\},$$

for some q_k *satisfying*

$$q_k^T B_k q_k \leqq c_2 \lambda_1(H(x_k))\|q_k\|^2.$$

Then $p_k(\Delta)$ *satisfies Condition* 2.

Proof. We just have to show that for some $\bar{c}_2 > 0$, $\text{pred}(t_k) \geqq \bar{c}_2(-\lambda_1(H(x_k))\Delta^2)$, for all iterations with $\lambda_1(H(x_k)) < 0$. Again we will drop the subscripts k.

Define $w = -\text{sgn}(g^T q)(\Delta/\|q\|)q$. Then

$$\text{pred}(w) = \frac{|g^T g|}{\|q\|}\Delta - \frac{1}{2}\frac{\Delta^2}{\|q\|^2} q^T B q \geqq -\frac{\Delta^2}{2} c_2 \lambda_1(H(x)),$$

since $q^T B q \leq c_2 \lambda_1(H(x))\|q\|^2$. So, since $\text{pred}(w) \leq \text{pred}(t_k) \leq \text{pred}(p_k(\Delta))$, $p_k(\Delta)$ satisfies Condition 2 with $\bar{c}_2 = c_2/2$. □

So, if the steps taken by an algorithm satisfy the hypotheses of both Lemmas 4.1 and 4.2, then the algorithm is second order stationary point convergent. For example, if an algorithm uses any steps giving as much descent as

$$s = \text{argmin}\{g_k^T w + \tfrac{1}{2} w^T B_k w : \|w\| \leq \Delta,\ w \in [d_k, q_k]\},$$

where d_k satisfies the requirement in Lemma 4.1, and q_k satisfies the requirement in Lemma 4.2 when $\lambda_1(H(x_k)) < 0$ and is 0 otherwise, then it satisfies both Conditions 1 and 2. One such algorithm is mentioned in § 5.

Finally, we note that Lemma 4.2 applies to the "optimal" algorithm, since this algorithm always achieves at least as much descent as is possible in the eigenvector direction corresponding to the most negative eigenvalue of $H(x_k)$. Taken together with Theorem 2.2, the two lemmas prove that the "optimal" algorithm is second order stationary point convergent.

Lemmas 4.1 and 4.2 can also be used to show convergence of algorithms using scaled trust regions of the form $\{t : \|D_k t\| \leq \Delta_k\}$, where D_k is a positive diagonal scaling matrix that may change at every iteration. If we are using such a scaled region to determine a step otherwise satisfying the conditions of Lemma 4.1, then we are requiring

$$s_k = \text{argmin}\{s^T g_k + \tfrac{1}{2} s^T B_k s : \|D_k s\| \leq \Delta,\ s \in [d_k]\}.$$

This satisfies the conditions of Lemma 4.1 as stated but with Δ replaced by $\Delta/\|D_k\|$. Then by the lemma, Condition 1 is satisfied with $\bar{c}_1$ replaced by $\bar{c}_1/\|D_k\|$ and similarly for σ_1. The same argument with Lemma 4.2 shows that Condition 2 remains satisfied with a modified trust region. Thus if we require that $\|D_k\|$ and $\|D_k^{-1}\|$ be bounded for all k, then the convergence results from Lemmas 4.1 and 4.2 also apply when using such a scaled trust region. They also apply to steps using trust regions based on other norms, such as l_1 or l_∞.

The final lemma contains a different set of sufficient conditions for a step computing method to satisfy both Conditions 1 and 2. These conditions are related to the step (4.2) of the "optimal" algorithm; however, Lemma 4.3 is broad enough to prove the second order stationary point convergence of a variety of algorithms, including several discussed in §§ 5 and 6.

LEMMA 4.3. *Suppose* $B_k = H(x_k)$ *and* $p_k(\Delta)$ *satisfies Condition* 1 *whenever* $\lambda_1(H(x_k)) \geq 0$. *Suppose further that there exist constants* $c_3 > 1$ *and* $c_4 \in (0, 1]$ *such that whenever* $\lambda_1(H(x_k)) < 0$, *for some* $\alpha_k \in [-\lambda_1(H(x_k)), c_3 \max\{|\lambda_1(H(x_k))|, \lambda_n(H(x_k))\}]$, $p_k(\Delta)$ *satisfies*:

(i) *if* $\Delta < \|-(B_k + \alpha_k I)^{-1} g_k\|$, *then* $p_k(\Delta)$ *is any step satisfying Conditions* 1 *and* 2;

(ii) *if* $\Delta = \|-(B_k + \alpha_k I)^{-1} g_k\|$, *then* $p_k(\Delta) = -(B_k + \alpha_k I)^{-1} g_k$;

(iii) *if* $\Delta > \|-(B_k + \alpha_k I)^{-1} g_k\|$, *then* $p_k(\Delta) = -(B_k + \alpha_k I)^{-1} g_k + \xi q_k$, *for some* q_k *satisfying* $q_k^T B_k q_k \leq c_4 \lambda_1(B_k)\|q_k\|^2$, *where* $\xi \in R$ *is chosen so that* $\|p_k(\Delta)\| = \Delta$ *and* $\text{sgn}(\xi) = -\text{sgn}(q_k^T (B_k + \alpha_k I)^{-1} g_k)$.

Then $p_k(\Delta)$ *also satisfies Conditions* 1 *and* 2 *whenever* $\lambda_1(H(x_k)) < 0$, *and thus an algorithm using* $p_k(\Delta)$ *is second order stationary point convergent. Furthermore, if the step in* (i) *satisfies*

$$\text{pred}(p_k(\Delta)) \geq \text{pred}\left(\frac{-\Delta(B_k + \alpha I)^{-1} g_k}{\|(B_k + \alpha I)^{-1} g_k\|}\right), \tag{4.3}$$

then it satisfies Condition 2.

Proof. We will drop the subscripts k and let $\lambda_1 = \lambda_1(H(x_k))$ and $\lambda_n = \lambda_n(H(x_k))$. We will first show that the step in (i) satisfies Condition 2 if (4.3) holds, and then that the step in (iii) satisfies Conditions 1 and 2. The proof for step (ii) is just a special case of the proof for step (iii) and is omitted.

To prove that step (i) satisfies Condition 2 if it satisfies (4.3), it is necessary and sufficient to prove that the step

$$p(\Delta) = \frac{-\Delta}{\|(B+\alpha I)^{-1}g\|}(B+\alpha I)^{-1}g = -\mu(B+\alpha I)^{-1}g$$

satisfies the condition, where $\mu < 1$ because $\Delta < \|(B+\alpha I)^{-1}g\|$. By straightforward algebraic manipulation,

$$\begin{aligned}\text{pred}\,(p(\Delta)) &= \mu g^T(B+\alpha I)^{-1}g - \tfrac{1}{2}\mu^2 g^T(B+\alpha I)^{-1}B(B+\alpha I)^{-1}g \\ &= \left(\mu - \frac{\mu^2}{2}\right) g^T(B+\alpha I)^{-1}g + \frac{\alpha}{2}\|\mu(B+\alpha I)^{-1}g\|^2 \\ &= \left(\mu - \frac{\mu^2}{2}\right) g^T(B+\alpha I)^{-1}g + \frac{\alpha}{2}\Delta^2.\end{aligned}$$

Since the first term is positive due to $\mu < 1$,

$$\text{pred}\,(p(\Delta)) > \frac{\alpha}{2}\Delta^2 > \frac{-\lambda_1}{2}\Delta^2,$$

so Condition 2 is satisfied.

Now consider step (iii) and let $p(\Delta) = -(B+\alpha I)^{-1}g + \xi q$. Then by simple algebraic manipulation,

$$\begin{aligned}\text{pred}\,(p(\Delta)) &= -g^T(\xi q - (B+\alpha I)^{-1}g) - \tfrac{1}{2}(\xi q - (B+\alpha I)^{-1}g)^T B(\xi q - (B+\alpha I)^{-1}g) \\ &= g^T(B+\alpha I)^{-1}g - \xi g^T q - \frac{\xi^2}{2}q^T B q + \xi q^T B(B+\alpha I)^{-1}g \\ &\quad - \tfrac{1}{2}g^T(B+\alpha I)^{-1}B(B+\alpha I)^{-1}g \\ &= \tfrac{1}{2}g^T(B+\alpha I)^{-1}g - \frac{\xi^2}{2}q^T B q - \xi\alpha q^T(B+\alpha I)^{-1}g + \frac{\alpha}{2}\|(B+\alpha I)^{-1}g\|^2 \\ &\geqq \tfrac{1}{2}g^T(B+\alpha I)^{-1}g - \xi^2\frac{c_4\lambda_1}{2}\|q\|^2 - \xi\alpha q^T(B+\alpha I)^{-1}g + \frac{\alpha}{2}\|(B+\alpha I)^{-1}g\|^2 \\ &= \tfrac{1}{2}g^T(B+\alpha I)^{-1}g - \frac{c_4\lambda_1}{2}\|\xi q - (B+\alpha I)^{-1}g\|^2 \\ &\quad + (-\xi c_4\lambda_1 - \xi\alpha)q^T(B+\alpha I)^{-1}g + \left(\frac{\alpha}{2} + \frac{c_4\lambda_1}{2}\right)\|(B+\alpha I)^{-1}g\|^2 \\ &\geqq \tfrac{1}{2}g^T(B+\alpha I)^{-1}g + \frac{c_4}{2}(-\lambda_1)\|p(\Delta)\|^2\end{aligned}$$

since the last two terms in the next to last expression above are nonnegative due to $\alpha > -\lambda_1 > -c_4\lambda_1$ and $\xi q^T(B+\alpha I)^{-1}g \leqq 0$.

So, we see that

$$\text{pred}\,(p(\Delta)) \geqq \tfrac{1}{2}g^T(B+\alpha I)^{-1}g + \frac{c_4(-\lambda_1)}{2}\Delta^2$$

and since the first quantity is positive, Condition 2 is clearly satisfied. Also,

$$\text{pred}(p(\Delta)) \geqq \tfrac{1}{2} g^T (B+\alpha I)^{-1} g \geqq \tfrac{1}{2} \frac{\|g\|^2}{\|B+\alpha I\|} \geqq \frac{1}{2(c_3+1)} \frac{\|g\|^2}{\|B\|},$$

with the last inequality due to

$$\|B+\alpha I\| = \lambda_n + \alpha \leqq \lambda_n + c_3 \max(|\lambda_1|, \lambda_n) \leqq (c_3+1)\|B\|.$$

So, Condition 1 is also satisfied. □

The value of Lemma 4.3 is that it suggests many algorithms that are second order stationary point convergent but are relatively efficient to implement. The reader may have recognized that conditions (ii) and (iii) of Lemma 4.3 just give an easy-to-implement way to identify the "hard case" in a second order algorithm, and to choose a step in this case. The inequality concerning q_k in (iii) says that q_k must be a direction of sufficient negative curvature. The inequality concerning α_k says that we can overestimate the magnitude of $\lambda_1(H(x_k))$ by an amount proportional to $\|H(x_k)\|$ and still achieve global convergence. When we are not in this "hard case," Lemma 4.3 gives us considerable leeway in choosing the step p_k. The algorithms of § 5 are mainly based on Lemma 4.3.

5. New algorithms that use negative curvature. In this section we present several idealized step selection strategies for problem (1.1) that use second order information. The step selection strategies are all based on the lemmas of § 4 and so any algorithm that uses one of them within the framework of Algorithm 2.1 achieves second order stationary point convergence. They are idealized only in the sense that they may use the most negative eigenvalue of the Hessian matrix and a direction of sufficient negative curvature q_k without specifying how these quantities are to be computed. In § 6 we will suggest a possible implementation of one of these algorithms, including the computation of the most negative eigenvalue and negative curvature direction when required, that should be quite efficient in terms of arithmetic operations required per step.

The first step selection strategy shows how a line search using second order information can be extended to the indefinite case in a natural way that satisfies the conditions of Lemma 4.3 and so assures second order stationary point convergence. The strategy is related to an algorithm by Gill and Murray [1972].

In all of the following, let $B_k = H(x_k)$.

ALGORITHM 5.1. Indefinite Line Search Step

Let $\kappa \gg 1$, $\kappa \leqq 1/\text{machine } \varepsilon$.

(a) When $\lambda_1(B_k) \geqq 0$ and $\kappa_2(B_k) \leqq \kappa$
(κ_2 is the l_2 condition number),
if $\|B_k^{-1} g_k\| \leqq \Delta$,
then $p_k(\Delta) = -B_k^{-1} g_k$,
otherwise $p_k(\Delta) = -\dfrac{\Delta}{\|B_k^{-1} g_k\|} B_k^{-1} g_k$.

(b) When $\lambda_1(B_k) < 0$ or $\kappa_2(B_k) > \kappa$, α_k is
chosen such that $B_k + \alpha_k I$ is positive definite and
$\kappa_2(B_k + \alpha_k I) = \kappa$, and $p_k(\Delta)$ is chosen by
(bi) if $\|(B_k + \alpha_k I)^{-1} g_k\| \geqq \Delta$,

$$\text{then } p_k(\Delta) = -\frac{\Delta}{\|(B_k+\alpha_k I)^{-1} g_k\|} (B_k + \alpha_k I)^{-1} g_k,$$

(bii) otherwise, if $\lambda_1(B_k) \geqq 0$
then $p_k(\Delta) = -(B_k + \alpha_k I)^{-1} g_k$,
(biii) otherwise,
$p_k(\Delta) = -(B_k + \alpha_k I)^{-1} g_k + \xi q_k$, where ξ and q_k are selected as in Lemma 4.3.

The second order stationary point convergence of any algorithm of the form of Algorithm 2.1 that chooses its steps by Algorithm 5.1 can easily be proven by using Lemma 4.3 combined with Lemma 4.1. To apply Lemma 4.1, it is necessary to note that for any positive definite matrix M, $g^T M^{-1} g \geqq \|g\| \, \|M^{-1} g\| / \kappa(M)$; to note that in case (bi) of Algorithm 5.1, the minimizer of the quadratic model in the direction $-(B_k + \alpha_k I)^{-1} g_k$ falls outside the trust region; and to extend the proof of Lemma 4.1 slightly to cover case (bii). Note that the constant κ that is used in Algorithm 5.1 could easily be replaced by some appropriate interval.

The next two step selection strategies are extensions of the dogleg strategy to the indefinite case. Algorithm 5.2 shows how to construct a dogleg version of the "optimal" algorithm. It is not implementable, due to its use of the pseudoinverse and the most negative eigenvalue and corresponding eigenvector of B_k. We include it in order to motivate Algorithm 5.3, which is similar but is implementable, as we shall see in § 6. Both steps are easily seen to satisfy the conditions of Lemma 4.3, with Lemmas 4.1 and 4.2 again applying to the portion of the algorithm not specified in Lemma 4.3.

ALGORITHM 5.2. Indefinite Dogleg Step A.
(a) When $\lambda_1(B_k) > 0$,
$p_k(\Delta) = \operatorname{argmin} \{ g_k^T w + \frac{1}{2} w^T B_k w : \|w\| \leqq \Delta, \, w \in [g_k, B_k^{-1} g_k] \}$.
(b) When $\lambda_1(B_k) \leqq 0$,
(bi) if g_k is not orthogonal to the null space of $B_k - \lambda_1 I$,
or $\|(B_k - \lambda_1 I)^+ g_k\| \geqq \Delta$,
then $p_k(\Delta) = \operatorname{argmin} \{ g_k^T w + \frac{1}{2} w^T B_k w : \|w\| = \Delta, \, w \in [g_k, v_k] \}$,
where $B_k v_k = \lambda_1 v_k$;
(bii) otherwise $p_k(\Delta) = -(B_k - \lambda_1 I)^+ g_k + \xi v_k$,
where ξ is selected so that $\|p_k(\Delta)\| = \Delta$.

Of course, the step in (a) could be replaced by a usual dogleg or double dogleg step. Also note that minimizing the quadratic model over a two-dimensional subspace involves performing the "optimal" algorithm when $n = 2$, or, equivalently, solving one fourth degree polynomial in one unknown, meaning that its computational cost is negligible.

The following is the Indefinite Dogleg Step that we propose in practice. Again, the step (a) for the positive definite case could be replaced by a normal dogleg or double dogleg step.

ALGORITHM 5.3. Indefinite Dogleg Step B.
(a) When $\lambda_1(B_k) > 0$, do the same as in Dogleg A.
(b) When $\lambda_1(B_k) \leqq 0$, let α_k be chosen as in Lemma 4.3,
$r_k = -(B_k + \alpha_k I)^{-1} g_k$, and $p_k(\Delta)$ chosen by
(bi) if $\|r_k\| \geqq \Delta$, then
$p_k(\Delta) = \operatorname{argmin} \{ g_k^T w + \frac{1}{2} w^T B_k w : \|w\| \leqq \Delta, \, w \in [g_k, r_k] \}$;
(bii) otherwise
$p_k(\Delta) = r_k + \xi q_k$, where ξ and q_k are selected as in Lemma 4.3.

The advantage of Algorithm 5.3 is that it is fairly easy and efficient to implement, as

we will show in § 6, while also being second order stationary point convergent, and that it approximates the "optimal" step selection strategy to some extent. Note that in case (bi) of Algorithm 5.3, $p_k(\Delta)$ always satisfies $\|p_k(\Delta)\| = \Delta$, unless $\lambda_1(B_k) = 0$ and $g_k = 0$.

Algorithm 5.4 shows how a simpler indefinite dogleg step can be constructed that satisfies the conditions of Lemmas 4.1 and 4.2 and so also achieves second order stationary point convergence.

ALGORITHM 5.4. Simple Indefinite Dogleg Step.

(a) When $\lambda_1(B_k) > 0$, do the same as Doglegs A and B.

(b) When $\lambda_1(B_k) \leqq 0$, let q_k satisfy
$q_k^T B_k q_k \leqq c_4 \lambda_1(B_k) \|q_k\|^2$, where c_4 is a uniform constant for all k, as in Lemma 4.3, and $g_k^T q_k \leqq 0$, and let $p_k(\Delta) = \operatorname{argmin}\{g_k^T w + \frac{1}{2} w^T B_k w: \|w\| = \Delta, w \in [g_k, q_k]\}$.

Algorithm 5.4 has the redeeming feature that it may be implemented so as to require no matrix factorizations for most indefinite iterations. However, Algorithm 5.4 might require more iterations than Algorithm 5.3 to solve the minimization problems. In § 6 we propose an implementation of an algorithm that subsumes Algorithms 5.3 and 5.4.

Finally, we mention a slight generalization of the "optimal" step that still leads to a second order stationary point convergent algorithm.

ALGORITHM 5.5. Variation of "Optimal" Step.

(a) When $\lambda_1(B_k) > 0$, let $p_k(\Delta)$ be the "optimal" step.

(b) When $\lambda_1(B_k) \leqq 0$, let α_k and q_k be chosen as in Lemma 4.3,
let $r_k = -(B_k + \alpha_k I)^{-1} g_k$, and
(bi) if $\|r_k\| \geqq \Delta$, then $p_k(\Delta) = \operatorname{argmin}\{g_k^T w + \frac{1}{2} w^T B_k w: \|w\| = \Delta\}$;
(bii) otherwise $p_k(\Delta) = r_k + \xi q_k$, where ξ is chosen so that $\|p_k\| = \Delta$ and $\xi r_k^T q_k \leqq 0$.

This step differs from the "optimal" step in that it uses α_k, not necessarily a close estimate of the most negative eigenvalue, in identifying the hard case, and that it just uses the direction of negative curvature q_k in this case, not necessarily an eigenvector corresponding to the most negative eigenvalue. This should make it considerably more efficient to implement in the hard case. The second order stationary point convergence follows from Lemma 4.3.

6. An implementation of the indefinite dogleg algorithm. Now we propose one possible implementation of the step selection strategy in Algorithm 5.3. We are currently testing an algorithm using this implementation, and our results will be reported in a forthcoming paper. We include this section as an example of the sort of algorithm the theory has been aimed at, and as an indication of a new way in which we believe such algorithms can be efficiently implemented. In this section we will always use $B_k = H(x_k)$.

Our implementation differs from that of Moré and Sorensen [1981] in that it uses explicit approximations to the most negative eigenvalue λ_1 and corresponding eigenvector v_1. We believe such an approach may be more efficient. The bulk of the computational work in most optimization algorithms, aside from function and derivative evaluations, is made up by matrix factorizations. In our implementation there would be the additional work involved in obtaining the approximations to the most negative eigenvalue and the corresponding eigenvector. Computational experience shows that a good algorithm for this, e.g. the Lanczos method, usually can obtain approximations

to outer eigenvalues and eigenvectors of a symmetric matrix with good accuracy, in fewer operations than one matrix factorization. According to Parlett [1980], the Lanczos algorithm usually requires $O(n^{2.5})$ or fewer arithmetic operations. Thus, calculating the desired eigen-information explicitly may not introduce a significant additional cost, especially since our algorithm only requires that the Lanczos algorithm obtain one significant digit of the most negative eigenvalue.

Figure 6.1 below contains a rough diagram of our proposed implementation of Algorithm 5.3 in the ideal case where exact arithmetic is used. In the following paragraphs we discuss the algorithm in more detail, including the stopping criteria in the Lanczos algorithm that are important for efficiency and for global convergence. The modifications required for finite precision arithmetic are discussed briefly at the end of the section. The estimation of the smallest eigenvalue and the corresponding eigenvector of B_k would only be done at the first minor iteration of each major (kth) iteration. If additional minor iterations were required at this major iteration, the necessary eigen-information would already be known and so one would immediately calculate the step in part (a) or (b) of Algorithm 5.3.

By the "attempted Cholesky factorization" we mean that if the matrix M being factored, either B_k or $B_k + \alpha I$, is positive definite, the algorithm calculates its LL^T factorization. Otherwise, it terminates and returns a direction z such that $z^T M z \leqq 0$ (see, e.g. Gill, Murray, and Wright [1981, p. 111]). The factorization requires about $n^3/6$ multiplications and $n^3/6$ additions in all cases.

The Lanczos algorithm produces a monotonically nonincreasing sequence of estimates $\{\sigma_i\}$ of $\lambda_1(B_k)$, with $\sigma_i \geqq \lambda_1(B_k)$ for all i, and a corresponding sequence of vectors $\{w_i\}$ such that $w_i^T B_k w_i = \sigma_i w_i^T w_i$. In addition, for all i the interval $[\sigma_i - \delta_i, \sigma_i + \delta_i]$ is guaranteed to contain some eigenvalue of B_k, where $\delta_i = \|B_k w_i - \sigma_i w_i\| / \|w_i\|$ (see e.g. Parlett [1980, p. 69]). We propose to terminate the Lanczos algorithm at the first step j for which

$$\left| \frac{\delta_j}{\sigma_j} \right| < \rho \tag{6.1}$$

where in practice we use $\rho = 0.1$. We then set $\lambda = \sigma_j$ and $v = w_j$. If $\lambda > 0$ B_k may be positive definite so the Cholesky factorization of B_k is attempted, but this can happen at most once at each major iteration because if the Cholesky factorization of B_k fails, the Lanczos algorithm is restarted with the direction z mentioned in the previous paragraph, and $\sigma_1 = z^T B_k z / z^T z \leqq 0$. If $\lambda < 0$ the factorization of $B_k + \alpha I = B_k - \lambda(1+\bar{\rho})I$ is attempted, where $\bar{\rho} \geqq \rho$; thus $-\alpha$ is less than the eigenvalue contained in the interval $[\sigma_j - \delta_j, \sigma_j + \delta_j]$ which is contained in the interval $[(1+\rho)\lambda, (1-\rho)\lambda]$. In practice we intend to experiment with values of $\bar{\rho} \in [0.1, 1]$. Usually $B_k + \alpha I$ will be positive definite and the major iteration will require only one matrix factorization. However, it is possible that $\lambda(1+\bar{\rho}) \geqq \lambda_1(B_k)$; in this case the Cholesky factorization of $B_k + \alpha I$ will fail. Then at all subsequent calls of the Lanczos algorithm at this major iteration we require

$$\lambda(1-\rho) < \text{previous } \lambda(1+\rho), \tag{6.2}$$

as well as (6.1), meaning that the intervals produced by successive calls of the Lanczos algorithm during the same major iteration must be monotonically decreasing and disjoint. If $\bar{\rho} \geqq 2\rho/(1-\rho)$ (6.2) is implied by (6.1) because $\lambda < (1-\bar{\rho})$ previous λ, so we still only need relative accuracy ρ from the Lanczos algorithm. Since each interval produced by the Lanczos algorithm is guaranteed to contain some eigenvalue of B_k,

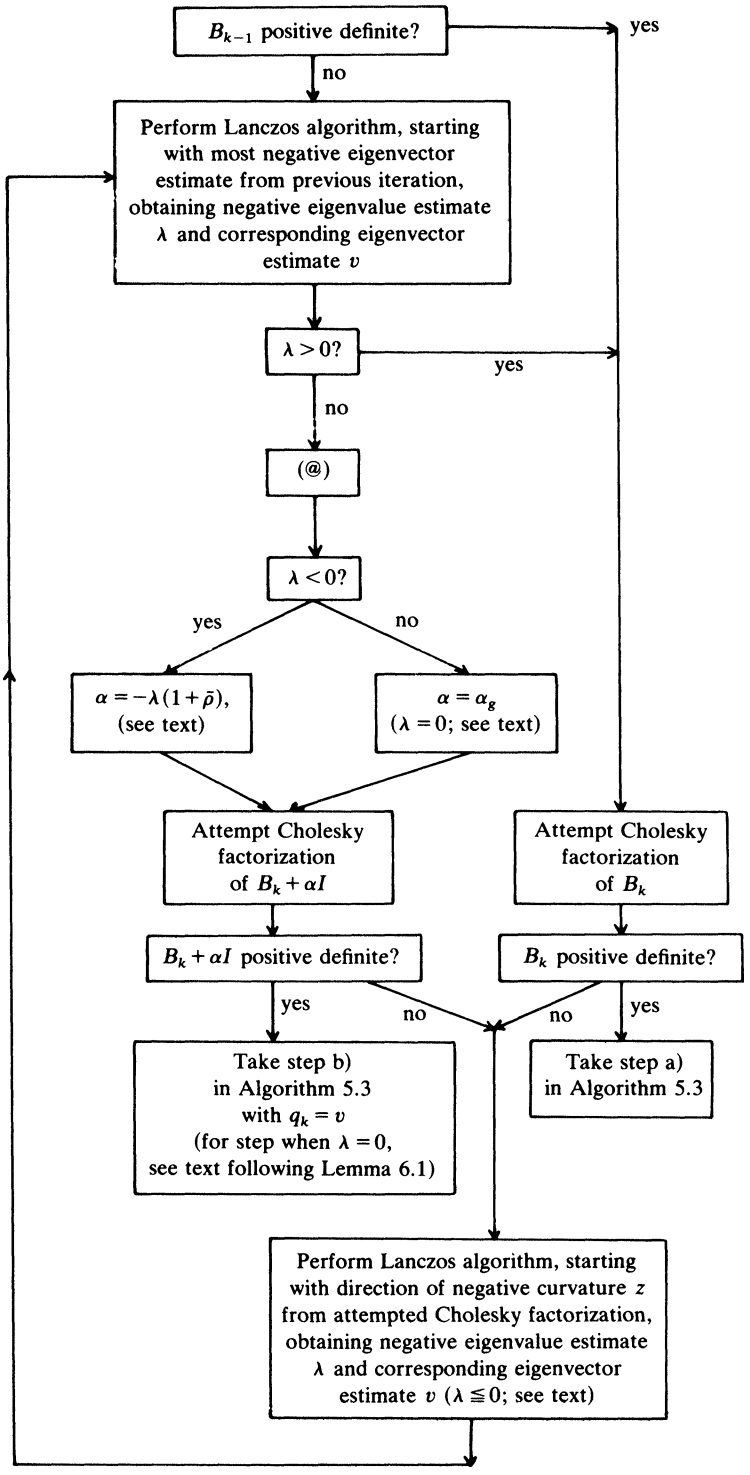

FIG. 6.1. *An implementation of the step selection strategy of Algorithm* 5.3.

since the starting vector produced by a failed Cholesky decomposition guarantees that the next Lanczos interval will contain a smaller eigenvalue than was contained in the previous interval, and since the above strategy guarantees that the intervals are disjoint, a maximum of n Lanczos calls is possible at any major iteration. In practice, we hope no more than 2 or 3 Lanczos calls ever are required at a major iteration. The number of Cholesky factorizations in the major iteration is equal to the number of Lanczos calls, or one more if B_{k-1} was positive definite. Once the iterates enter a region around the solution where $\nabla^2 f(x)$ is positive definite, no more Lanczos work is performed.

It is easy to show that the implementation in Fig. 6.1 satisfies the requirements of Theorem 2.2 and is thus globally convergent. Iterations where $\lambda_1(B_k) > 0$ are immediately covered by Lemma 4.1. Iterations where $\lambda_1(B_k) < 0$ are covered by Lemma 4.3, as follows. Since the successful value of α satisfies

$$0 > \lambda \geqq \lambda_1(B_k) > -\alpha$$

where

$$\alpha = -\lambda(1+\bar{\rho}),$$

we have that

$$\alpha = -\lambda(1+\bar{\rho}) < -\lambda_1(B_k)(1+\rho).$$

Since in addition v satisfies

$$v^T B_k v = \lambda v^T v,$$

it follows that

$$v^T B_k v = \lambda v^T v = \frac{-\alpha}{1+\bar{\rho}} v^T v < \frac{\lambda_1(B_k)}{1+\bar{\rho}} v^T v.$$

Thus the conditions of Lemma 4.3 on α and $q_k = v$ are satisfied. In fact, in our "hard case" where $\|(B_k+\alpha I)^{-1} g_k\| < \Delta$, since $\|\xi q_k\| \leqq \Delta$ by the assumptions in Lemma 4.3(iii), it follows from the above that

$$\xi^2 v^T (B_k + \alpha I) v = \left(-\frac{\alpha}{1+\bar{\rho}} + \alpha\right) \xi^2 v^T v \leqq \frac{\bar{\rho}}{1+\bar{\rho}} \alpha \Delta^2,$$

and thus Moré and Sorensen [1981, Lemma 3.4] implies that the reduction of the quadratic model by our step is at least $1/(1+\bar{\rho})$ of the optimal reduction. Iterations where $\lambda_1(B_k) = 0$ are covered by Lemma 6.1; the step we take in this case, (6.4), is motivated by the lemma.

LEMMA 6.1. *Suppose* $\text{predg} = -\min_{\tau \in R} \{\tau g_k^T g_k + \frac{1}{2}\tau^2 g_k^T B_k g_k : \|\tau g_k\| \leqq \Delta\}$, *i.e.* predg *is the maximum possible reduction of the quadratic model by steps of length* $\leqq \Delta$ *in the negative gradient direction. Let*

$$\alpha_g = \frac{\text{predg}}{\overline{c_2}\Delta^2}, \tag{6.3}$$

where $\overline{c_2}$ *is the constant in Condition* 2. *Then if* $B_k + \alpha_g I$ *is positive definite, any step* $p_k(\Delta)$ *for which* $\text{pred}(p_k(\Delta)) \geqq \text{predg}$ *satisfies Conditions* 1 *and* 2 *of* § 2. *In particular, the step*

$$p_k(\Delta) = \text{argmin}\{g_k^T w + \tfrac{1}{2} w^T B_k w : \|w\| \leqq \Delta,\ w \in [g_k, (B_k+\alpha I)^{-1} g_k]\} \tag{6.4}$$

satisfies these conditions.

Proof. From Lemma 4.1, Condition 1 is satisfied by any step that does as well as the best step in the negative gradient direction. If $B_k + \alpha_g I$ is positive definite, then $a_g > -\lambda_1(B_k)$ so by (6.3)

$$\text{pred}\,(p_k(\Delta)) \geqq \text{predg} > \overline{c_2}\Delta^2(-\lambda_1(B_k)),$$

which proves that Condition 2 is satisfied. □

If $\lambda_1(B_k) = 0$ and $g_k \neq 0$, then α_g given by (6.3) will be positive so $B_k + \alpha_g I$ will be positive definite. In this case we intend the algorithm of Fig. 6.1 to take the step (6.4), which satisfies the conditions for global convergence by Lemma 6.1 (independent of the value of $\overline{c_2}$ or α). If $\lambda_1(B_k) = 0$ and $g_k = 0$, x_k satisfies the second order necessary conditions for minimization. Thus the implementation of Fig. 6.1 is globally convergent in exact arithmetic. By Theorem 2.2 it is also locally q-quadratically convergent if $H(x_*)$ is positive definite.

In finite precision arithmetic, some minor modifications to the algorithm of Fig. 6.1 are required because of the problems that may be encountered when $\lambda_1(B_k)$ is near 0. For example, the Cholesky factorization may fail when B_k is barely positive definite, and $z^T B_k z$ may be positive. Alternatively, the Lanczos algorithm may produce $\lambda_1 < 0$ when B_k is positive definite or fail to produce a $\lambda_1 < 0$ when B_k is indefinite. And of course, the test $\lambda_1 = 0$ is unrealistic in finite precision arithmetic. We believe that the only real difficulty introduced by all these possibilities is remedied by setting $\alpha = \alpha_g$ when the Cholesky factorization of B_k has failed and the subsequent λ_1 is nonnegative. Our implementation will be discussed more thoroughly in a forthcoming paper.

Finally, we mention a modification to the implementation in Fig. 6.1 that may cause the algorithm to require no matrix factorizations at iterations where $\lambda_1(B_k) < 0$, while maintaining global convergence. The modification is to insert Fig. 6.2 at the spot marked by @ in Fig. 6.1. If

$$v^T B_k v < c_4 \lambda_{low} v^T v, \tag{6.5}$$

step (b) of Algorithm 5.4, with $q_k = v$, trivially satisfies the conditions of Lemma 4.2 for global convergence. The bound λ_{low} may be obtained at the start of a major iteration from the Gerschgorin circle theorem. If (6.5) is satisfied the first time through the loop, then no matrix factorizations are required at this major iteration. If (6.5) is not satisfied and the subsequent Cholesky factorization fails, another value for λ_{low} may be obtained from the factorization and, if it is larger than the previous value, may be used after the next call of Lanczos. Of course, it remains to be seen how this modification affects the total number of iterations required by the algorithm.

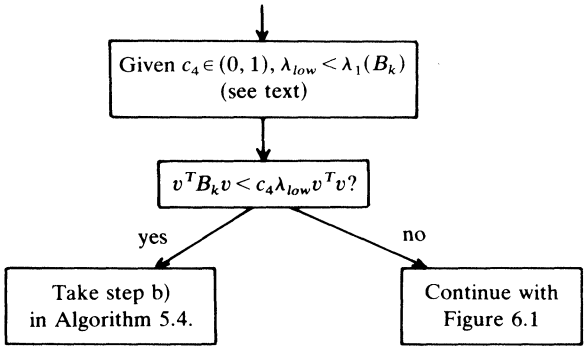

FIG. 6.2. *Augmentation of Fig.* 6.1 *with the step selection strategy of Algorithm* 5.4.

Acknowledgment. We thank the referees and associate editor for their helpful comments on an earlier version of this paper.

REFERENCES

J. E. DENNIS, JR. AND H. H. W. MEI (1979), *Two new unconstrained optimization algorithms which use function and gradient values*, J. Optim. Theory Appl., 28, pp. 453–482.

J. E. DENNIS, JR. AND R. B. SCHNABEL (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ.

R. FLETCHER (1980), *Practical Methods of Optimization, Vol.* 1, John Wiley, Chichester, New York, Brisbane, and Toronto.

R. FLETCHER AND T. L. FREEMAN (1977), *A modified Newton method for minimization*, J. Optim. Theory Appl., 23, pp. 357–372.

D. M. GAY (1981), *Computing optimal locally constrained steps*, SIAM J. Sci. Stat. Comput., 2, pp. 186–197.

P. E. GILL AND W. MURRAY (1972), *Quasi-Newton methods for unconstrained optimization*, J. Inst. Math. Appl., 9, pp. 91–108.

P. E. GILL, W. MURRAY AND M. H. WRIGHT (1981), *Practical Optimization*, Academic Press, New York.

D. GOLDFARB (1980), *Curvilinear path steplength algorithms for minimization which use directions of negative curvature*, Math. Programming, 18, pp. 31–40.

M. D. HEBDEN (1973). *An algorithm for minimization using exact second derivatives*, Atomic Energy Research Establishment report T.P. 515, Harwell, England.

S. KANIEL AND A. DAX (1979), *A modified Newton's method for unconstrained minimization*, this Journal, 16, pp. 324–331.

G. P. MCCORMICK (1977), *A modification of Armijo's step-size rule for negative curvature*, Math. Programming, 13, pp. 111–115.

J. J. MORÉ (1978), *The Levenberg-Marquardt algorithm: implementation and theory*, Lecture Notes in Mathematics 630, G. A. Watson, ed., Springer-Verlag, Berlin, Heidelberg, and New York, pp. 105–116.

J. J. MORÉ AND D. C. SORENSEN (1979), *On the use of directions of negative curvature in a modified Newton method*, Math. Programming, 16, pp. 1–20.

———, (1981), *Computing a trust region step*, Argonne National Laboratory report, Argonne, IL.; SIAM J. Sci. Stat. Comput., 4 (1983), pp. 553–572.

H. MUKAI AND E. POLAK (1978), *A second order method for unconstrained optimization*, J. Optim. Theory, Appl., 26, pp. 501–513.

B. N. PARLETT (1980), *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ.

M. S. D. POWELL (1970), *A new algorithm for unconstrained optimization*, Nonlinear Programming, J. B. Rosen, O. L. Mangasarian and K. Ritter, eds., Academic Press, New York, pp. 31–65.

———, (1975), *Convergence properties of a class of minimization algorithms*, Nonlinear Programming 2, O. L. Mangasarian, R. R. Meyer and S. M. Robinson, eds., Academic Press, New York, pp. 1–27.

D. C. SORENSEN (1982), *Newton's method with a model trust-region modification*, this Journal, 19, pp. 409–426.

S. W. THOMAS (1975), *Sequential estimation techniques for quasi-Newton algorithms*, Technical Report TR 75-227, Dept. Computer Science, Cornell Univ.

J. P. VIAL AND I. ZANG (1975), *Unconstrained optimization by approximation of the gradient path*, C.O.R.E. discussion paper.