A Family of Trust Region Based Algorithms For Unconstrained Minimization With Strong Global Convergence Properties

By

Gerald A. Shultz, Robert B. Schnabel*,
And
Richard H Byrd

CU-CS-216-82

University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

* Supported by NSF NYI #CCR-9357740, ONR #N00014-96-1-0720, and a Packard Fellowship in Science and Engineering from the David and Lucile Packard Foundation.

b) When $\lambda_1(B_k) \le 0$, let α_k be chosen as in Lemma 4.5,

$r_k = -(B_k + \alpha_k I)^{-1} g_k$, and $p_k(\Delta)$ chosen by

bi) if $||r_k|| \ge \Delta$, then

$$p_k(\Delta) = argmin\{g_k^T w + \tfrac{1}{2} w^T B_k w : ||w|| = \Delta,\ w \varepsilon [-g_k, r_k]\};$$

bii) otherwise

$p_k(\Delta) = r_k + \xi q_k$, where ξ and q_k are selected as in Lemma 4.5.

The advantage of Algorithm 5.3 is that it is fairly easy and efficient to implement, as we will show in Section 6, while also being a continuous step selection strategy that is second order stationary point convergent , and that it approximates the "optimal" step selection strategy to some extent.

Algorithm 5.4 shows how a simpler indefinite dogleg step can be constructed that satisfies the conditions of Lemmas 4.3 and 4.4 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 $\lambda_1(B_k) \le 0$, let q_k satisfy

$$q_k^T B_k q_k \le -c_4 \lambda_1(B_k)\, ||q_k||^2,$$

where c_4 is a uniform constant for all k, as in Lemma 4.5, and $g_k^T q_k \le 0$, and let

$$p_k(\Delta) = argmin\{g_k^T w + \tfrac{1}{2} w^T B_k w : ||w|| = \Delta,\ w \varepsilon [-g_k, q_k]\}.$$

Algorithm 5.4 is not continuous as discussed above when $\lambda_1(B_k) = 0$ but if q_k is reasonably chosen this will not be a problem, and the algorithm 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 Section 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 (Sorensen [1980]) 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 $\lambda_1(B_k)\leq 0$, let α_k and q_k be chosen as in Lemma 4.5,

let $r_k=-(B_k+\alpha_k I)^{-1}g_k$, and

bi) if $||r_k||\geq\Delta$, then $p_k(\Delta)=argmin\{g_k^T w+\frac{1}{2}w^T B_k w : ||w||=\Delta\}$;

bii) otherwise $p_k(\Delta)=r_k+\xi q_k$, where ξ is chosen so that $||p_k||=\Delta$.

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 makes it considerably more efficient to implement in the hard case. The second order stationary point convergence follows obviously from Lemma 4.5.

6. An Implementation of the Indefinite Dogleg Algorithm

In this section we will always use $B_k = H(x_k)$.

Now we present one possible implementation of the step selection strategy in Algorithm 5.3 , both as an example of the sort of algorithm the theory has been aimed at, and as partial justification that such algorithms can be efficiently implemented.

Our implementation differs from More and Sorensen's [1981] in that it uses explicit approximations to the most negative eigenvalue λ_1 and corresponding eigenvector v_1. We claim that this approach may well 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 is the additional work involved in obtaining the approximations to the largest and smallest eigenvalues and the most negative eigenvector. Computational experience shows that a good algorithm for this, e.g. the Lanczos method, can obtain approximations to outer eigenvalues and eigenvectors of a symmetric matrix with guaranteed accuracy, with 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.

Figure 6.1 below contains a diagram of our proposed implementation of Algorithm 5.3. This implementation includes estimation of the extreme eigenvalues and the corresponding eigenvectors of B_k. This would only be done at the first minor iteration of each major (k-th) 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.

In two places in Figure 6.1 there are "attempted Cholesky decompositions", of B_k and $B_k+\alpha I$. These algorithms are given in Gill, Murray, and Wright [1981] or Dennis and Schnabel [1983]. If the matrix is numerically positive definite, the factorization algorithm calculates the LL^T factorization of the matrix. If it is not numerically positive definite, the factorization algorithm returns a lower bound λ_l on the most negative eigenvalue of the matrix and a direction of negative curvature v for the matrix (i.e. for B_k or $B_k+\alpha I$, respectively). The factorization algorithm requires about $\frac{n^3}{6}$ multiplications and additions in all cases. Since the Lanczos algorithm is restarted using this direction v, the λ_1 that results from the next use of the Lanczos algorithm at the same iteration must be smaller than the curvature of v. Thus in particular, the λ_1 resulting from the Lanczos algorithm can be positive only if B_{k-1} was not positive definite and one is going through the left-hand loop of Figure 6.1 for the first time in the k-th iteration.

A possible choice of α in Figure 6.1 is

$$\alpha := \frac{max(0,\lambda_n)}{\varepsilon} - \lambda_1$$

where $\varepsilon \geq \sqrt{machine\,\varepsilon}$. If $B_k+\alpha I$ is positive definite and step bii) is required, v almost certainly will satisfy the conditions on g_k in Lemma 4.5; this may be tested using $-\alpha$ which is a lower bound on $\lambda_1(B_k)$. It is theoretically possible that additional iterations of the Lanczos procedure would be required to find a satisfactory v in this case.

Figure 6.2 shows how our implementation of Algorithm 5.3 given in Figure 6.1 can be modified to sometimes substitute the simpler step b) of Algorithm 5.4 for step b) of Algorithm 5.3, when B_k is not positive definite. A lower bound λ_l on $\lambda_1(B_k)$ is always available, initially from the Gerschgorin theorem, and subsequently from the failed Cholesky decomposition. If the negative curvature direction v from the Lanczos algorithm satisfies the condition of Lemma 4.5 for g_k, using this lower bound λ_l in place of $\lambda_1(B_k)$, then step b) of Algorithm 5.4 may be taken. If the constant c_4 in Lemma 4.5 is chosen small, the first v probably will satisfy the condition of Lemma 4.5. If step b) of Algorithm 5.4 is taken as soon as it is possible, the step selection strategy of Figures 6.1 and 6.2 may require no matrix factorizations when B_k is not positive definite. Another alternative is to take this step only if some fixed number of Cholesky decompositions have failed, say two.

The implementations in Figures 6.1 and 6.2 strive to minimize the number of matrix factorizations. When B_k is positive definite, only one factorization will be needed, in addition the Lanczos work will be required only if B_{k-1} was not positive definite. When B_k is not positive definite, the algorithm will perform between zero and n factorizations, usually between 0 and 2 or 3. When the step in Figure 6.2 is taken on the first iteration, no factorizations are needed. Generally the Lanczos algorithm will yield a good enough approximation to $\lambda_1(B_k)$ that the first α will yield a positive definite $B_k+\alpha I$, and thus only one factorization will be required in the indefinite case. In certain rather pathological cases, the Lanczos algorithm can tend to converge not to the smallest eigenvalue but

Figure 6.2

Optional augmentation with the step selection strategy of Algorithm 5.4.

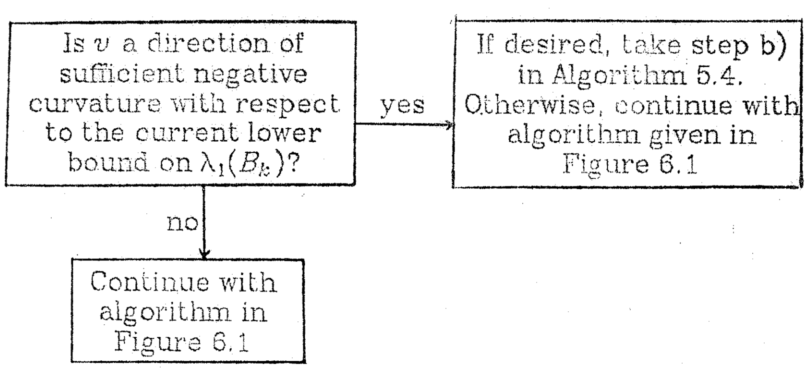
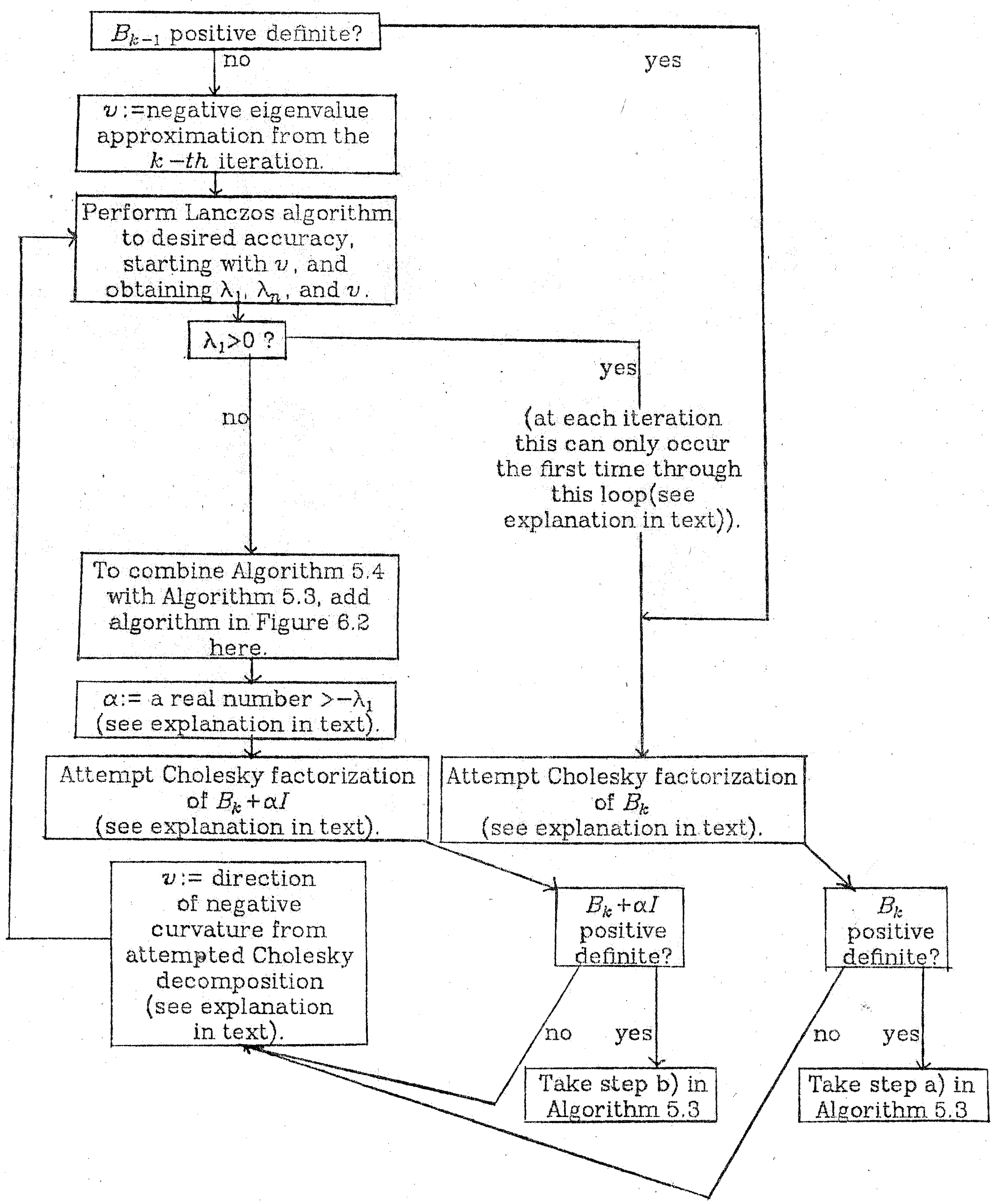


Figure 6.1
An implementation of the step selection strategy of Algorithm 5.3.



Powell, M. S. D. [1975]. **Convergence properties of a class of minimization algorithms**, in **Nonlinear Programming 2**, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, pp. 1-27.

Sorensen, D. C. [1980]. **Newton's method with a model trust-region modification**, Argonne National Laboratory , Report ANL-80-106, Argonne, Illinois. SIAM J. Num. Anal., to appear.

Vial, J. P. and Zang, I. [1975]. **Unconstrained optimization by approximation of the gradient path**, C.O.R.E. discussion paper.

to a larger one, in which case the Cholesky factorization will fail. Then the algorithm will use the direction of negative curvature from the Cholesky failure as a starting vector for the Lanczos process, which guarantees that the Lanczos algorithm will converge to a smaller eigenvalue than the last one. Thus, although we expect only one factorization to be required in the indefinite case, it is possible that several may be needed, but never more than n.

In summary, this implementation will require one factorization on all positive definite Hessian matrices, and most indefinite ones. In addition, when B_k is not positive definite it will require the work involved in the Lanczos process, which is likely to be considerably less than the work of one factorization when n is large. The implementation satisfies the requirements of Lemmas 4.3 and 4.5, and hence a computer code using this step in the framework of Algorithm 2.1 is second order stationary point convergent. Of course, by Theorem 2.2 it is also locally q-quadratically convergent. The techniques in Figure 6.1 could also be employed in the implementation of other step selection strategies, in particular the indefinite line search step given in Algorithm 5.1 or the modified "optimal" step given in Algorithm 5.3, leading again to implementations that are second order stationary point convergent.

7. References

Dennis, J. E. Jr. and Mei, H. H. W. [1979]. **Two new unconstrained optimization algorithms which use function and gradient values**, J.O.T.A., pp. 453-482.

Dennis, J. E. Jr. and Schnabel, R. B. [1983]. **Numerical methods for unconstrained optimization and nonlinear equations**, Prentice-Hall, Englewood Cliffs, New Jersey.

Fletcher, R. and Freeman, T. L. [1977]. **A modified Newton method for minimization**, J.O.T.A. 23 (3), pp. 357-372.

Gay, D. M. [1981]. **Computing optimal locally constrained steps**, SIAM J. Sci. Stat. Comput. 2, pp. 186-197.

Goldfarb, D. [1980]. **Curvilinear path steplength algorithms for minimization which use directions of negative curvature**, Math. Prog., 18, pp. 31-40.

Gill, P. E. and Murray, W. [1972]. **Quasi-Newton methods for unconstrained optimization**, The Journal of the Institute of Mathematics and its Applications, 9, pp. 91-108.

Gill, P. E., Murray, W., and Wright, M. [1981]. **Practical optimization**, Academic Press, New York.

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

Kaniel, S. and Dax, A. [1979]. **A modified Newton's method for unconstrained minimization**, SIAM J. Num. Anal., pp. 324-331.

McCormick, G. P. [1977]. **A modification of Armijo's step-size rule for negative curvature**, Math. Prog. 13, pp. 111-115.

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

More, J. J. and Sorensen, D. C. [1979]. **On the use of directions of negative curvature in a modified Newton method**, Math. Prog. 16, pp. 1-20.

More, J. J. and Sorensen, D. C. [1981]. **Computing a trust region step**, Argonne National Laboratory report, Argonne, Illinois.

Mukai, H. and Polak, E. [1978]. **A second order method for unconstrained optimization**, J.O.T.A. 26, pp. 501-513.

Parlett, B. N. [1980]. **The symmetric eigenvalue problem**, Prentice-Hall, Englewood Cliffs, New Jersey.

Powell, M. S. D. [1970]. **A hybrid method for nonlinear equations**, pp. 87-114 of **Numerical Methods for Nonlinear Algebraic Equations**, P. Rabinowitz, ed., Gordon and Breach, London.

Powell, M. S. D. [1975]. **Convergence properties of a class of minimization algorithms**, in **Nonlinear Programming 2**, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, pp. 1-27.

Sorensen, D. C. [1980]. **Newton's method with a model trust-region modification**, Argonne National Laboratory , Report ANL-80-106, Argonne, Illinois. SIAM J. Num. Anal., to appear.

Vial, J. P. and Zang, I. [1975]. **Unconstrained optimization by approximation of the gradient path**, C.O.R.E. discussion paper.

CONTENTS

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

$$\min_{x \varepsilon R^n} f(x): R^n \to R \qquad (1.1)$$

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 [1980]. Sorensen proves strong convergence properties for a specific trust region algorithm, which uses second order information. Others, including Fletcher and Freeman [1977], Goldfarb [1980], Kaniel and Dax [1979], McCormick [1977], More 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 [1975], on the other hand, discusses the first order stationary point convergence properties of a class of trust region algorithms.

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

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

$H(x) \varepsilon R^{n \times n}$ 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.

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

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 the conditions they may meet in order 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 k-th 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 $w \varepsilon R^n$ and $B_k \varepsilon R^{n \times n}$ 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

$$ared_k(\Delta_k)=f_k-f(x_k+p_k(\Delta_k))$$

to the reduction predicted by the quadratic model

$$pred_k(\Delta_k)=f_k-\psi_k(p_k(\Delta_k)).$$

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

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

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

   $||p_k|| \leq \Delta_k$ and

   a) $\dfrac{ared_k(\Delta_k)}{pred_k(\Delta_k)} \geq \eta_1$ and

   b) either $\Delta_k \geq \Delta_{k-1}$ or

   for some $\Delta \leq \dfrac{1}{\gamma_1}\Delta_k$,

   $\dfrac{ared_k(\Delta)}{pred_k(\Delta)} < \eta_2$ or $\dfrac{ared_{k-1}(\Delta)}{pred_{k-1}(\Delta)} < \eta_2$.

3) $x_{k+1} = x_k + p_k$, $k = k+1$.

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 Section 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 does meet these conditions then the convergence results follow. In Sections 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.

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

Our conditions that a step selection strategy may satisfy are:

Condition #1

There are $\bar{\sigma}_1$, $\sigma_1>0$ such that for all $g\,\varepsilon R^n$, for all symmetric $B\,\varepsilon R^{n\times n}$, and for all $\Delta>0$, $pred(g,B,\Delta)\geq\bar{\sigma}_1 ||g||\, min(\Delta,\sigma_1 \frac{||g||}{||B||})$.

Condition #2

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

Condition #3

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

We now state and prove the convergence theorem. The proofs are similar to those of Sorensen [1980]. Conditions #1, #2, and #3 constitute a major generalization of his assumption that

$$p(g,B,\Delta)=argmin\{\, g^T w+w^T Bw \;:\; ||w||\leq\Delta\,\}$$

Theorem 2.2

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

I. If $p(g,B,\Delta)$ satisfies Condition #1 and $||B_k|| \le \beta_2$ 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, $H(x)$ is uniformly continuous, and x_k converges to x_*, then $H(x_*)$ is positive semi-definite (second order stationary point convergence, with I.).

Proof:

Each of the proofs of I, II, and III use the following fact:

Lemma If there is a positive integer M and a function $w(\Delta)$ such that

1) $\lim_{\Delta \to 0^+} w(\Delta) = 0$,

2) for all $\Delta > 0$, for all $k \ge M$,

$$\left| \frac{ared_k(\Delta)}{pred_k(\Delta)} - 1 \right| \le w(\Delta), \text{ and}$$

3) each Δ_k satisfies the trust radius requirement in step 2b) of Algorithm 2.1,

then $\{\Delta_k\}$ is bounded away from 0.

Proof of the lemma: By 1) and 2), there is a $\bar{\Delta} > 0$ such that if $0 < \Delta < \bar{\Delta}$ and $k \ge M$, then $\frac{ared_k(\Delta)}{pred_k(\Delta)} \ge \eta_2$. Thus, for $k \ge M+1$, if $\Delta_k < \Delta_{k-1}$, then by 3) there must be some $\Delta \le \frac{1}{\gamma_1} \Delta_k$ which either has $\frac{ared_k(\Delta)}{pred_k(\Delta)} < \eta_2$ or $\frac{ared_{k-1}(\Delta)}{pred_{k-1}(\Delta)} < \eta_2$. But that means that $\Delta \ge \bar{\Delta}$, so $\Delta_k \ge \gamma_1 \Delta \ge \gamma_1 \bar{\Delta}$. Hence, for $k \ge M+1$, $\Delta_k \ge min(\Delta_{k-1}, \gamma_1 \bar{\Delta})$, so clearly $\{\Delta_k\}$ is bounded away from 0.

Each of the three parts also uses the following:

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

$$|ared_k(\Delta)-pred_k(\Delta)|$$
$$=|f_k-f(x_k+p_k(\Delta))-(f_k-f_k-g_k^Tp_k(\Delta)-\tfrac{1}{2}p_k(\Delta)^TB_kp_k(\Delta))|$$
$$=|\tfrac{1}{2}p_k(\Delta)^TB_kp_k(\Delta)-\int_0^1 p_k(\Delta)^TH(x_k+\xi p_k(\Delta))p_k(\Delta)(1-\xi)d\xi|$$
$$\leq ||p_k(\Delta)||^2\int_0^1 ||B_k-H(x_k+\xi p_k(\Delta))||(1-\xi)d\xi.$$

So,

$$|\frac{ared_k(\Delta)}{pred_k(\Delta)}-1|$$
$$\leq\frac{||p_k(\Delta)||^2\int_0^1 ||B_k-H(x_k+\xi p_k(\Delta))||(1-\xi)d\xi}{|pred_k(\Delta)|}.$$

All three parts proceed by using the relevant hypotheses and the above argument to bound $pred_k(\Delta)$ below by a term that is $O(\Delta^2)$, and then using the lemma above.

Proof of I: Consider any m with $||g_m||\neq 0$.

For any x, $||g(x)-g_m||\leq\beta_1||x-x_m||$, so if $||x-x_m||<\frac{||g_m||}{2\beta_1}$, then

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

Call $R=\frac{||g_m||}{2\beta_1}$, and $B_R=\{x : ||x-x_m||<R\}$.

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

$$pred_k(\Delta) \geq \sigma \, ||g_k|| \min(\Delta, \frac{||g_k||}{||B_k||})$$

$$\geq \sigma \varepsilon \min(\Delta, \frac{\varepsilon}{\beta_2})$$

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

$$\left| \frac{ared_k(\Delta)}{pred_k(\Delta)} - 1 \right|$$

$$\leq \frac{\Delta^2 \int_0^1 ||B_k - H(x_k + \xi p_k(\Delta))|| (1-\xi) d\xi}{\sigma \varepsilon \min(\Delta, \frac{\varepsilon}{\beta_2})}$$

$$\leq \frac{\Delta^2(\beta_1 + \beta_2)}{\sigma \varepsilon \min(\Delta, \frac{\varepsilon}{\beta_2})}$$

$$\leq \frac{\Delta(\beta_1 + \beta_2)}{\sigma \varepsilon}$$

for all $k \geq m$ and $\Delta \leq \frac{\varepsilon}{\beta_2}$. Applying the lemma with $w(\Delta) = \frac{\Delta(\beta_1 + \beta_2)}{\sigma \varepsilon}$, and $M = m$, we see that $\{\Delta_k\}$ is bounded away from 0. But, since

$$f_k - f_{k+1} = ared_k(\Delta_k) \geq \eta_1 pred_k(\Delta_k)$$

$$\geq \eta_1 \sigma \varepsilon \min(\Delta_k, \frac{\varepsilon}{\beta_2}),$$

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

$$f(x_{l+1}) - f(x_m) = \sum_{k=m}^{l} f(x_{k+1}) - f(x_k)$$

$$\geq \sum_{k=m}^{l} \eta_1 pred_k(\Delta_k) \geq \sum_{k=m}^{l} \eta_1 \sigma \, min(\Delta_k, \frac{\varepsilon}{\beta_2})$$

$$\geq \eta_1 \sigma \varepsilon min(\sum_{k=m}^{l} \Delta_k, (l-m)\frac{\varepsilon}{\beta_2})$$

$$\geq \eta_1 \sigma \varepsilon \min(\sum_{k=m}^{l} ||p_k(\Delta_k)||, (l-m)\frac{\varepsilon}{\beta_2})$$

$$\geq \eta_1 \sigma \varepsilon \min(R, (l-m)\frac{\varepsilon}{\beta_2})$$

$$= \eta_1 \sigma \frac{||g_m||}{2} \min(\frac{||g_m||}{2\beta_1}, (l-m)\frac{||g_m||}{2\beta_2})$$

$$= ||g_m||^2 \eta_1 \frac{\sigma}{4} \min(\frac{1}{\beta_1}, \frac{1}{\beta_2}).$$

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

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

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 in fact, if $H(x_*)$ is positive definite, then 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 $g(x) \neq 0$. Call $B_1=\{x : ||x-x_*||<\delta_1\}$.

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

Find j_0 such that $f(x_{k_{j_0}})<inf\{f(x) : x \varepsilon B_1-B_2\}$, and $x_{k_{j_0}} \varepsilon B_2$. Consider any x_l, with $l \geq k_{j_0}$, $x_l \varepsilon B_2$. We claim that $x_{l+1} \varepsilon B_2$ 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

$$\Delta_l = ||x_{l+1}-x_l|| \geq ||x_{l+1}-x_*|| - ||x_l - x_*|| \geq \delta_1 - \frac{\delta_1}{4} = \frac{3}{4}\delta_1$$

$$> \frac{\delta_1}{2} \geq ||B(x_l)^{-1}g(x_l)||.$$

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 $||p_l(\Delta_l)||<\delta_1$, $x_{l+1} \varepsilon B_1$, which is a contradiction.

Thus for all $k \geq k_{j_0}$, $x_k \varepsilon B_2$, 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 show that $\{\Delta_k\}$ is bounded away from 0, which gives the result, since $||H(x_k)^{-1}g(x_k)||$ converges to 0, so eventually, by Condition #3, the Newton step will always be taken. Then by a usual theorem the Lipschitz continuity of H implies the quadratic convergence rate.

To show that $\{\Delta_k\}$ is bounded away from 0, we will again use the lemma. In order to do so, we need the appropriate lower bound on $pred_k(\Delta)$.

By Condition #1,

$$pred_k(\Delta) \geq \sigma ||g_k|| \min(\Delta, \frac{||g_k||}{||B_k||}) \geq \sigma ||g_k|| \min(||p_k(\Delta)||, \frac{||g_k||}{||B_k||}),$$

and for all k large enough, $B_k = H(x_k)$ is positive definite, so 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 $||g_k|| \geq \frac{||p_k(\Delta)||}{||B_k^{-1}||}$. Thus,

$$pred_k(\Delta) \geq \sigma ||p_k(\Delta)|| \min(||p_k(\Delta)||, \frac{||p_k(\Delta)||}{||B_k^{-1}|| \, ||B_k||})$$

$$= \sigma ||p_k(\Delta)||^2 \min(1, \frac{1}{||B_k^{-1}|| \, ||B_k||}).$$

Now call $c_* = \frac{1}{2} min(1, \frac{1}{||H(x_*)^{-1}|| \, ||H(x_*)||}$, and note that by continuity there is an M such that for $k \geq M$, B_k is positive definite and $min(1, \frac{1}{||B_k^{-1}|| \, ||B_k||}) \geq c_*$.

Finally, note that by the argument given earlier and Lipschitz continuity,

$$|ared_k(\Delta)-pred_k(\Delta)|\le||p_k(\Delta)||^3\frac{L}{2},$$

thus for any $\Delta>0$ and $k\ge M$,

$$\left|\frac{ared_k(\Delta)}{pred_k(\Delta)}-1\right|\le\frac{||p_k(\Delta)||^3\frac{L}{2}}{\sigma c_*||p_k(\Delta)||^2}$$

$$=\frac{L||p_k(\Delta)||}{2\sigma c_*}\le\frac{L\Delta}{2\sigma c_*},$$

so by applying the lemma with $w(\Delta)=\frac{L\Delta}{2\sigma c_*}$, we have that $\{\Delta_k\}$ is bounded away from 0 and we are done.

Proof of III: Suppose to the contrary that $\lambda_1(H(x_*))<0$. By the uniform continuity of H, for any $\Delta>0$, and any k,

$$\left|\frac{ared_k(\Delta)}{pred_k(\Delta)}-1\right|\le\frac{||p_k(\Delta)||^2\overline{w}(\Delta)}{pred_k(\Delta)},$$

where

$$\overline{w}(\Delta)=\int_0^1||H(x_k+\xi p_k(\Delta))-H(x_k)||(1-\xi)d\xi,$$

and thus $\lim_{\Delta\to0^+}w(\Delta)=0$.

Find M such that if $k\ge M$, $\lambda_1(B_k)<\frac{\lambda_1(H(x_*))}{2}<0$. By Condition #2, for all $k\ge M$, and for all $\Delta>0$,

$$pred_k(\Delta)\ge\overline{c}_2(-\lambda_1(B_k))\Delta^2\ge\overline{c}_2(-\lambda_1(H(x_*))/2)\Delta^2,$$

so since $||p_k(\delta)||<\delta$, the lemma applies with

$$w(\Delta)=\frac{\overline{w}(\Delta)}{\overline{c}_2(-\lambda_1(H(x_*))/2)}.$$

Thus, $\{\Delta_k\}$ is bounded away from 0.

But,

$$ared_k(\Delta_k)\ge\eta_1 pred_k(\Delta_k)\ge\overline{c}_2(-\lambda_1(H(x_*))/2)\Delta_k^2,$$

and since f is bounded below $ared_k(\Delta_k)$ converges to 0, so Δ_k converges to 0, which is a contradiction. Hence, $\lambda_1(H(x_*))\ge0$. 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 $||D_k p|| \leq \Delta$ for some non-singular 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 $||p_k|| \leq \Delta$, and the conditions otherwise do not involve $||p||$. The conditions are also not restricted to 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.

To choose the initial trial radius at the k-th iteration, Algorithm 2.1 only requires that two conditions be met. First, the initial trial radius can be smaller than the final radius used for the previous step only if the previous step failed the sufficient decrease condition, i.e.

$$\frac{ared_{k-1}(\Delta_{k-1})}{pred_{k-1}(\Delta_{k-1})} < \eta_2 .$$

Second, in this case the ratio between the previous Δ_{k-1} and the new trial radius must be bounded by some constant that is fixed for the entire algorithm. These possibilities are covered by the condition b) in step 2) of Algorithm 2.1. 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. Clearly some methods for doing this could be very inefficient, but from the point of view of global convergence any increase is allowable.

One method for choosing the initial trial trust region at the k-th iteration which Algorithm 2.1 does not cover is basing the radius on the length of the previous step p_{k-1} even when p_{k-1} falls in the interior of the trust region Δ_{k-1}. We see little justification for this strategy, and including it in our theory, if possible, would make the analysis more cumbersome.

Given the initial trial radius at the k-th iteration, a sequence of trial radii may be tried until a satisfactory one is found. Algorithm 2.1 only requires that the trial radius be reduced when the previous trial step fails to satisfy the condition a) in step 2) of Algorithm 2.1 and only in this case, and that the reduction be bounded below by a constant that is fixed for the entire algorithm. This case is covered by the condition

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

and

$$\frac{ared_k(\Delta)}{pred_k(\Delta)} < \eta_2$$

in Algorithm 2.1. Of course, the trust region ultimately used must satisfy this condition.

The conditions of Algorithm 2.1 also allow successively larger trial trust regions to be tried within the k-th 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 final one used satisfies condition a) of step 2) in Algorithm 2.1. Notice that it is not necessary to increase the trust region at any point. Never increasing the trust region may cause great inefficiency, but convergence is still assured.

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

$$p_k(\Delta_k)=argmin\{f_k+g_k^Tw+\tfrac{1}{2}w^TB_kw \,:\, \|w\|\le\Delta_k\}. \qquad (4.1)$$

This strategy has been discussed and used by many authors, see e.g. Hebden [1973], More [1978], Sorensen [1980], and Gay [1981]. B_k is positive definite and $\|-B_k^{-1}g_k\|\le\Delta_k$, then $p_k=-B_k^{-1}g_k$ is the solution to (4.1). Otherwise, p_k satisfies $(B_k+\alpha_kI)p_k=-g_k$, for some non-negative α_k such that $(B_k+\alpha_kI)$ is at least positive semi-definite and $\|p_k\|=\Delta_k$. If B_k is positive definite, then so is $(B_k+\alpha_kI)$ and

$$p_k=-(B_k+\alpha_kI)^{-1}g_k\ , \qquad (4.2)$$

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 $(B_k-\lambda_1I)$ and $\|(B_k-\lambda_1I)^+g_k\|<\Delta_k$; here the superscript + denotes the generalized inverse and λ_1 denotes the most negative eigenvalue of B_k. In this case, which More and Sorensen [1981] refer to as the "hard case", $p_k=-(B_k-\lambda_1I)^+g_k+\xi_kv_k$, 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 Powell [1970] and Dennis and Mei [1979]. These algorithms are defined in the case when B_k is positive definite and always choose $p_k \varepsilon [-g_k, -B_k^{-1} g_k]$. When $\Delta_k \geq || -B_k^{-1} g_k ||$, p_k is the Newton step $-B_k^{-1} g_k$; when $\Delta_k \leq \frac{||g_k||^3}{g_k^T B_k g_k} \leq || -B_k^{-1} g_k ||$, p_k is the steepest descent step of length Δ_k; when $\Delta_k \varepsilon (\frac{||g_k||^3}{g_k^T B_k g_k}, || -B_k^{-1} g_k ||)$, p_k is the step of length Δ_k on a specified piecewise linear curve connecting $\frac{-||g_k||^2}{g_k^T B_k g_k} g_k$ and $-B_k^{-1} g_k$ (see Dennis and Schnabel [1983] for further explanation). The lemmas of this section will lead to natural and efficient extensions of these algorithms to the indefinite case which satisfy the conditions of Theorem 2.2 for second order stationary point convergence.

The first lemma gives a very general condition on the step at each iteration 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

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

then it would satisfy Condition #1. Lemma 4.3 is a slight generalization of this fact.

Here we slightly change our earlier notation and let

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

Lemma 4.3

Suppose there is a constant $c_1 \varepsilon (0,1]$ such that at each iteration k,

$$pred(p_k(\Delta)) \geq -\min\{ g_k^T w + \tfrac{1}{2} w^T B_k w \ : \ ||w|| \leq \Delta, w \varepsilon [d_k] \},$$

for some d_k satisfying

$$d_k^T g_k \leq -c_1 ||d_k|| \ ||g_k||.$$

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 $pred(s_*) \geq \frac{c_1}{2} ||g|| \min(\Delta, \frac{c_1 ||g||}{||B||})$, where s_* solves the above minimization problem. This will clearly imply satisfaction of Condition #1 by $p(\Delta)$, since $pred(p(\Delta)) \geq pred(s_*)$, by assumption.

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

Let $s_* = a_* d$, i.e. a_* is the multiple of d which minimizes the quadratic $g^T w + w^T Bw$ along that direction, subject to the constraint $||w|| \leq \Delta$. Now, if $d^T Bd > 0$, then either $a_* = \frac{-g^T d}{d^T Bd}$, if $\frac{-g^T d}{d^T Bd} \leq \Delta$, or else $a_* = \frac{\Delta}{||d||}$. In the first case we have

$$pred(s_*)$$

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

$$= \frac{1}{2} \frac{(g^T d)^2}{d^T Bd}$$

$$\geq \frac{1}{2} c_1^2 \frac{||g||^2 ||d||^2}{d^T Bd}$$

$$\geq \frac{1}{2} c_1^2 \frac{||g||^2}{||B||}.$$

In the second case, we have

$$pred(s_*)$$

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

$$\geq -\frac{1}{2} \frac{\Delta}{||d||} g^T d$$

(with the inequality above true since $\frac{\Delta}{||d||} < -\frac{g^T d}{d^T Bd}$)

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

Finally, if $d^TBd \leq 0$, $\alpha_* = \frac{\Delta}{\|d\|}$, and so we have

$$pred(s_*)$$

$$= -\frac{\Delta}{\|d\|} g^Td - \tfrac{1}{2}\left(\frac{\Delta}{\|d\|}\right)^2 d^TBd$$

$$\geq -\frac{\Delta}{\|d\|} g^Td \geq c_1\Delta \|g\|.$$

Thus, s_* and hence $p(\Delta)$ satisfy Condition #1, with constants $\bar{c}_1 = \frac{c_1}{2}$ 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 4.3, 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 most line search algorithms which keep the angle between the steps and the gradient bounded away from 90 degrees, because the step length adjusting strategy and step acceptance strategy in the line search can be shown to correspond to a trust radius adjusting strategy and step acceptance strategy allowed by Algorithm 2.1. In addition, it applies to any dogleg type algorithm, e.g. Powell [1970] and Dennis-Mei [1979], 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 4.4

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

$$pred(p_k(\Delta)) \geq pred(t_k),$$

where

$$t_k = argmin\{g_k^T w + \tfrac{1}{2} w^T B_k w \;:\; \|w\| \leq \Delta, w \varepsilon [q_k]\},$$

for some q_k satisfying

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

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

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

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

$$pred(w) = \frac{|g^T q|}{\|q\|}\Delta - \tfrac{1}{2} \frac{\Delta^2}{2\|q\|^2} q^T B q$$

$$\geq -\frac{\Delta^2}{2} c_2 \lambda_1(H(x)),$$

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

So, if the steps taken by an algorithm satisfy the hypotheses of both Lemmas 4.3 and 4.4, then the algorithm is second order stationary point convergent. For example, if an algorithm uses any steps giving as much descent as

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

where d_k satisfies the requirement in Lemma 4.3, and q_k satisfies the requirement in Lemma 4.4 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 Section 5.

Finally, we note that Lemma 4.4 applies to the "optimal" algorithm (Sorensen [1980]), 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 4.3 and 4.4 can also be used to show convergence of algorithms using scaled trust regions of the form $\{ t : ||D_k t|| \le \Delta_k \}$, 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 4.3, then we are requiring

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

This satisfies the conditions of Lemma 4.3 as stated but with Δ replaced by $\frac{\Delta}{||D_k||}$. Then by the Lemma, Condition #1 is satisfied with $\bar{\sigma}_1$ replaced by $\frac{\bar{\sigma}_1}{||D_k||}$ and similarly for σ_1. The same argument with Lemma 4.4 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 4.3 and 4.4 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 4.5 is broad enough to prove the second order stationary point convergence of a variety of algorithms, including several discussed in Sections 5 and 6.

Lemma 4.5

Suppose $B_k = H(x_k)$ and $p_k(\Delta)$ satisfies Condition #1 whenever $\lambda_1(H(x_k)) \ge 0$. Suppose further that there exist constants $c_3 > 1$ and $c_4 \varepsilon (0,1]$ such that whenever $\lambda_1(H(x_k)) < 0$, for some $\alpha_k \varepsilon (-\lambda_1(H(x_k)), c_3 \max(|\lambda_1|, \lambda_n]$, $p_k(\Delta)$ 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 $q_k^T B_k q_k \le c_4\lambda_1(B_k)||q_k||^2$, where $\xi \varepsilon R$ is chosen so that $||p_k(\Delta)||=\Delta$ and $sgn(\xi)=-sgn(q_k^T(B_k+\alpha_k I)^{-1}g_k)$.

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.

Proof: We will drop the subscripts k, and call $\lambda_1=\lambda_1(H(x_k))$. We will first show that the step in iii) satisfies Conditions #1 and #2, and then see from the same calculation that the step in ii) satisfies these conditions.

If $p(\Delta)=-(B+\alpha I)^{-1}g+\xi q$, then by simple algebraic manipulation we have that

$$pred(p(\Delta))=$$

$$=-g^T(\xi q-(B+\alpha I)^{-1}g)-\tfrac{1}{2}(\xi q-(B+\alpha I)^{-1}g)^T B(\xi q-(B+\alpha I)^{-1}g)$$

$$=g^T(B+\alpha I)^{-1}g-\xi g^Tq-\frac{\xi^2}{2}q^TBq+\xi q^TB(B+\alpha I)^{-1}g-\tfrac{1}{2}g^T(B+\alpha I)^{-1}B(B+\alpha I)^{-1}g$$

$$=\tfrac{1}{2}g^T(B+\alpha I)^{-1}g-\frac{\xi^2}{2}q^TBq-\xi\alpha q^T(B+\alpha I)^{-1}g+\frac{\alpha}{2}||(B+\alpha I)^{-1}g||^2$$

$$\ge\tfrac{1}{2}g^T(B+\alpha I)^{-1}g-\xi^2\frac{c_4\lambda_1}{2}||q||^2-\xi\alpha q^T(B+\alpha I)^{-1}g+\frac{\alpha}{2}||(B+\alpha I)^{-1}g||^2$$

$$=\tfrac{1}{2}g^T(B+\alpha I)^{-1}g-\frac{c_4\lambda_1}{2}||\xi q-(B+\alpha I)^{-1}g||^2$$

$$+(-\xi c_4\lambda_1-\xi\alpha)q^T(B+\alpha I)^{-1}g+(\frac{\alpha}{2}+\frac{c_4\lambda_1}{2})||(B+\alpha I)^{-1}g||^2$$

$$\ge\tfrac{1}{2}g^T(B+\alpha I)^{-1}g+\frac{c_4}{2}(-\lambda_1)||p(\Delta)||^2$$

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

So, we see that

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

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

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

$$\geq \frac{1}{2(c_3+1)} \frac{\|g\|^2}{\|B\|},$$

with the last inequality due to

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

So, Condition #2 is also satisfied.

Finally, note that in case ii), we can take $\xi=0$, and the same calculations yield satisfaction of Conditions #1 and #2 by the step in ii).

The value of Lemma 4.5 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 4.5 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 4.5 says that we have great leeway in choosing the step p_k. The algorithms of Section 5 are mainly based on Lemma 4.5.

5. New Algorithms That Use Negative Curvature

In this section we present several idealized step selection strategies for Problem 1.1 which use second order information. The step selection strategies are all based on the lemmas of Section 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 largest and smallest eigenvalues of the Hessian matrix and a direction of sufficient negative curvature q_k without specifying how these quantities are to be computed. In Section 6 we will suggest a possible implementation of one of these algorithms, including the computation of the extreme eigenvalues and negative curvature direction when required.

Before describing the step selection strategies we turn briefly to the question of judging these strategies. So far we have been concerned with convergence properties. We now consider two other factors, the computational work involved in calculating the step and the continuity of the step selection strategy. We define a continuous step selection strategy to be one where the function $p(g, B, \Delta)$ is a continuous function of g, B, and Δ. We note that the "optimal" strategy in Sorensen [1980] has this property except in the highly unusual case that the algorithm is at a point x with $\lambda_1(H(x))=0$, g orthogonal to the null space of $H(x)$, and $||H(x)^+ g|| < \Delta$. All of the strategies to follow will have the same property, except as otherwise noted. As for the computational work, the algorithm we present in Section 6 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 4.5 and so assures second order stationary point convergence. The strategy is related to an algorithm by Gill and Murray [1972].