SIAM J. Sci. Stat. Comput.
Vol. 4, No. 3, September 1983

0196-5204/83/0403-0009/$01.25/0

COMPUTING A TRUST REGION STEP*

JORGE J. MORÉ and D. C. SORENSEN †

Abstract. We propose an algorithm for the problem of minimizing a quadratic function subject to an ellipsoidal constraint and show that this algorithm is guaranteed to produce a nearly optimal solution in a finite number of iterations. We also consider the use of this algorithm in a trust region Newton's method. In particular, we prove that under reasonable assumptions the sequence generated by Newton's method has a limit point which satisfies the first and second order necessary conditions for a minimizer of the objective function. Numerical results for GQTPAR, which is a Fortran implementaton of our algorithm, show that GQTPAR is quite successful in a trust region method. In our tests a call to GQTPAR only required 1.6 iterations on the average.

Key words. Newton's method, trust region, ellipsoidal constraint, global convergence

1. Introduction.

In an important class of minimization algorithms called "trust region methods" (see, for example, Sorensen [1982]), the calculation of the step between iterates requires the solution of a problem of the form

$$\min\{\psi(w) : \|w\| \leqslant \Delta\} \tag{1.1}$$

where Δ is a positive parameter, $\|\cdot\|$ is the Euclidean norm in R^n, and

$$\psi(w) \equiv g^T w + \tfrac{1}{2} w^T B w, \tag{1.2}$$

with $g \in R^n$, and $B \in R^{n \times n}$ a symmetric matrix. The quadratic function ψ generally represents a local model to the objective function defined by interpolatory data at an iterate and thus it is important to be able to solve (1.1) for any symmetric matrix B; in particular, for a matrix B with negative eigenvalues.

In trust region methods it is sometimes helpful to include a scaling matrix for the variables. In this case, problem (1.1) is replaced by

$$\min\{\psi(v) : \|Dv\| \leqslant \Delta\} \tag{1.3}$$

where $D \in R^{n \times n}$ is a nonsingular matrix. The change of variables $Dv = w$ shows that problem (1.3) is equivalent to

$$\min\{\hat{\psi}(w) : \|w\| \leqslant \Delta\} \tag{1.4}$$

where $\hat{\psi}(w) \equiv \psi(D^{-1}w)$, and that the solutions of problems (1.3) and (1.4) are related by $Dv = w$. Because of this equivalence, we only consider problem (1.1). Also note that if, as is usually the case, D is a diagonal matrix, then it is easy to explicitly carry out the change of variables and solve problem (1.4).

The use of a trust region method in a nonlinear optimization problem requires the solution of many problems of type (1.1). These problems do not usually require accurate solutions, but in all cases we must be able to find an approximate solution with a reasonable amount of computational effort, and the approximate solution found must guarantee that the trust region method has the

* Received by the editors December 16, 1981 and in revised form July 15, 1982.

† Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois 60439. Work supported in part by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under Contract W-31-109-Eng-38.

right sort of convergence properties. In this paper we are concerned with these two issues; namely, robust and stable algorithms for the solution of (1.1) and the impact of these algorithms on the convergence properties of trust region methods.

Goldfeld, Quandt, and Trotter [1966], Hebden [1973], Fletcher [1980], Gay [1981], and Sorensen [1982], have discussed (1.1) in connection with trust region methods. Their algorithms are based on the fact that if (1.1) has a solution on the boundary of $\{w : \| w \| \leqslant \Delta\}$ then, in most cases, a solution of (1.1) can be found by determining $\lambda \geqslant 0$ such that $B + \lambda I$ is positive definite and

$$\| (B + \lambda I)^{-1} g \| = \Delta \tag{1.5}$$

In one case - the hard case - equation (1.5) has no solution with $B + \lambda I$ positive definite, and this leads to numerical difficulties. Hebden [1973] proposed an algorithm for the solution of (1.1) which is basically sound except for its treatment of the hard case. Gay [1981] improved Hebden's scheme and showed that under certain conditions the approximate solution determined by his algorithm is nearly optimal. His algorithm, however, may require a large number of iterations in the hard case.

We propose an algorithm for the solution of (1.1) which is guaranteed to produce a nearly optimal solution in a finite number of steps. Specifically, given parameters σ_1 and σ_2 in (0,1), the approximate solution s satisfies

$$\psi(s) - \psi^* \leqslant \sigma_1(2-\sigma_1)\max\{|\psi^*|,\sigma_2\} , \quad \| s \| \leqslant (1+\sigma_1)\Delta ,$$

where ψ^* is the optimal value of (1.1). We also consider the use of our algorithm in a trust region Newton's method. In particular, we prove that under reasonable assumptions the sequence $\{x_k\}$ generated by Newton's method has a limit point x^* which satisfies the first and second order necessary conditions for a minimizer of the objective function f. Numerical results for GQTPAR, which is a Fortran implementation of our algorithm, show that GQTPAR is quite successful in a trust region method. In our tests a call to GQTPAR only required 1.6 iterations on the average.

The outline of the paper is as follows. The theoretical basis of an algorithm for the solution of (1.1) is laid out in Section 2, while in Section 3 we present the algorithm and show that the solution generated by the algorithm is nearly optimal. In Section 4 we consider the use of this algorithm in a trust region Newton's method and prove that the combined algorithm has very strong convergence properties. Numerical results are presented in Section 5.

2. Structure of the problem.

Problem (1.1) has a tremendous amount of structure and it is important to understand this structure in order to construct a suitable algorithm. The following results expose this structure and provide a theoretical basis for the numerical algorithm. Note that these results provide necessary and sufficient conditions for a point $p \in R^n$ to be a solution to (1.1) and that there is no "gap" between the necessary and sufficient conditions.

LEMMA(2.1). *If p is a solution to* (1.1) *then p is a solution to an equation of the form*

$$(B + \lambda I)p = -g, \tag{2.2}$$

with $B + \lambda I$ positive semidefinite, $\lambda \geqslant 0$, and $\lambda(\Delta - \| p \|) = 0$.

LEMMA(2.3). *Let* $\lambda \in R$, $p \in R^n$ *satisfy* (2.2) *with* $B + \lambda I$ *positive semidefinite.*

(*i*) *If* $\lambda = 0$ *and* $\|p\| < \Delta$ *then* p *solves* (1.1).

(*ii*) p *solves* $\psi(p) = \min\{\psi(w) : \|w\| = \|p\|\}$.

(*iii*) *If* $\lambda \geqslant 0$ *and* $\|p\| = \Delta$ *then* p *solves* (1.1).

If $B + \lambda I$ *is positive definite then* p *is the only solution to* (1.1).

Simple proofs of these lemmas are given by Sorensen [1982]. Lemma (2.3) is important from a computational standpoint since it provides a perturbation result that is useful in setting termination rules for the iterative method used to solve (1.1).

The solution of (1.1) is straightforward if (1.1) has no solutions on the boundary of $\{w : \|w\| \leqslant \Delta\}$. In fact, (1.1) has no solution p with $\|p\| = \Delta$ if and only if B is positive definite and $\|B^{-1}g\| < \Delta$. To prove this claim, first note that if B is positive definite and $\|B^{-1}g\| < \Delta$ then Lemma (2.3) immediately shows that $p = -B^{-1}g$ is the solution to (1.1). On the other hand, if (1.1) has a solution p with $\|p\| < \Delta$ then Lemma (2.1) shows that $\lambda = 0$ and that B is positive semidefinite. If B were singular then $Bz = 0$ for some z with $\|p+z\| = \Delta$ and then Lemma (2.3) implies that $p+z$ is a solution to (1.1) on the boundary. This contradiction establishes our claim.

Now assume that (1.1) has a solution on the boundary of $\{w : \|w\| \leqslant \Delta\}$. If g is not perpendicular to the eigenspace

$$S_1 \equiv \{z : Bz = \lambda_1 z, \ \ z \neq 0\}$$

where λ_1 is the smallest eigenvalue of B, then the nonlinear equation $\|p_\alpha\| = \Delta$ where

$$p_\alpha = -(B + \alpha I)^{-1} g$$

has a solution $\lambda \geqslant 0$ in $(-\lambda_1,\infty)$. Moreover, Lemma (2.3) implies that p_λ is the solution of problem (1.1). Reinsch [1967,1971] and Hebden [1973] observed independently that to solve $\|p_\alpha\| = \Delta$ great advantage could be taken of the fact that the function $\|p_\alpha\|^2$ is a rational function in α with second order poles on a subset of the negatives of the eigenvalues of the symmetric matrix B . To see this consider the decomposition

$$B = Q \Lambda Q^T \text{ with } \Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ \cdots, \lambda_n) \text{ and } Q^T Q = I,$$

and observe that

$$\|p_\alpha\|^2 = \|Q(\Lambda + \alpha I)^{-1} Q^T g\|^2 = \sum_{j=1}^{n} \frac{\gamma_j^2}{(\lambda_j + \alpha)^2} \tag{2.4}$$

where γ_i is the ith component of $Q^T g$. In the next section we elaborate on the importance of this observation.

If g is perpendicular to S_1 then the equation $\|p_\alpha\| = \Delta$ may still have a solution $\lambda \geqslant 0$ in $(-\lambda_1,\infty)$, and in this case the solution to (1.1) can be obtained as above. If, however, $\|p_\alpha\| = \Delta$ has no solutions in $(-\lambda_1,\infty)$ then this leads to numerical difficulties. We call this situation the "hard case".

A characteristic difficulty of the hard case is that $\|p_\alpha\| < \Delta$ whenever

$B + \alpha I$ is positive definite. For example, if

$$B = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad g = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

then $\lambda_1 = -1$, and if $B + \alpha I$ is positive definite then $\|p_\alpha\|^2 \leqslant 1/2$. In the hard case, a solution to (1.1) can be obtained by solving

$$(B - \lambda_1 I)p = -g$$

for p with $\|p\| \leqslant \Delta$ and by determining an eigenvector $z \in S_1$. Then

$$(B - \lambda_1 I)(p + \tau z) = -g$$

and $\|p + \tau z\| = \Delta$ for some τ. Lemma (2.3) now shows that $p + \tau z$ solves (1.1).

3. The algorithm. Consider the solution of $\|p_\alpha\| = \Delta$. The rational structure of $\|p_\alpha\|^2$ may be exploited by applying Newton's method to the zero finding problem

$$\phi(\alpha) \equiv \frac{1}{\Delta} - \frac{1}{\|p_\alpha\|} = 0. \tag{3.1}$$

Newton's method is very efficient when applied to (3.1) since ϕ is almost linear on $(-\lambda_1, \infty)$. Moreover, the computation of the Cholesky factorization of $B + \alpha I$ makes it possible to compute the necessary derivative whenever $\alpha \in (-\lambda_1, \infty)$. There is no need to compute the eigensystem of B as suggested by Goldfeld, Quandt, and Trotter [1966]. The following algorithm updates λ by Newton's method applied to (3.1).

```
ALGORITHM(3.2). Let $\lambda \geqslant 0$ with $B + \lambda I$ positive definite and $\Delta > 0$ be given.
1) Factor $B + \lambda I = R^T R$ ;
2) Solve $R^T R p = -g$ ;
3) Solve $R^T q = p$ ;
4) Let $\lambda := \lambda + \left(\frac{\|p\|}{\|q\|}\right)^2 \left(\frac{\|p\| - \Delta}{\Delta}\right)$ ;
```

In this algorithm $R^T R$ is the Cholesky factorization of $B + \lambda I$ with $R \in R^{n \times n}$ upper triangular. Although not represented in this simplified version, it is necessary to safeguard λ in order to obtain a positive definite $B + \lambda I$ and guarantee convergence. These safeguards, and the convergence criteria for this algorithm are discussed later on in this section.

If properly safeguarded, the iteration produced by Algorithm (3.2) is sufficiently rapid to solve most problems of type (1.1) expected in practice. However, in the hard case this scheme may require a large number of iterations to converge; in particular, if $g = 0$ then Algorithm (3.2) breaks down. In the hard case it is necessary to supply a vector z that is an approximate eigenvector of B corresponding to λ_1. Indeed, as pointed out at the end of Section 2, in the hard case a solution to (1.1) is

$$p = -(B - \lambda_1 I)^{\dagger} g + \tau z, \tag{3.3}$$

where the superscript $\dagger$ denotes the Moore-Penrose generalized inverse, $z \in S_1$, and τ is chosen so that $\| p \| = \Delta$. Note that actual computation of the two orthogonal components of p indicated in (3.3) may require more computational effort than is reasonable in the context of an optimization algorithm. Moreover, recognizing that a solution of the form (3.3) is required may also be time consuming. Fortunately, there is a completely acceptable alternative. The following result provides a foundation for this technique.

LEMMA(3.4). *Let* $0 < \sigma < 1$ *be given and suppose that*

$$B + \lambda I = R^T R \ , \ (B + \lambda I)p = -g \ , \lambda \geqslant 0.$$

Let $z \in R^n$ *satisfy*

$$\| p + z \| = \Delta \ , \ \| Rz \|^2 \leqslant \sigma (\| Rp \|^2 + \lambda \Delta^2). \tag{3.5}$$

Then

$$-\psi(p+z) \geqslant \tfrac{1}{2}(1-\sigma)(\| Rp \|^2 + \lambda\Delta^2) \geqslant (1-\sigma)|\psi^*| \ ,$$

where ψ^* *is the optimal value of* (1.1).

Proof. First note that for any $z \in R^n$,

$$\psi(p+z) = -\tfrac{1}{2}(\| Rp \|^2 + \lambda \| p+z \|^2) + \tfrac{1}{2}\| Rz \|^2. \tag{3.6}$$

Then for any z which satisfies (3.5),

$$-\psi(p+z) \geqslant \tfrac{1}{2}(1-\sigma)(\| Rp \|^2 + \lambda\Delta^2).$$

Moreover, if $\psi^* = \psi(p+z^*)$ where $\| p + z^* \| \leqslant \Delta$, then (3.6) implies

$$-\psi(p+z^*) \leqslant \tfrac{1}{2}\left(\| Rp \|^2 + \lambda\Delta^2\right).$$

The last two inequalities yield Lemma (3.4). ■

A consequence of Lemma (3.4) is that $|\psi(p+z) - \psi^*| \leqslant \sigma|\psi^*|$. This shows that if (3.5) holds then $p + z$ is a nearly optimal solution to problem (1.1).

A consequence of the proof of Lemma (3.4) is equation (3.6). This expression is quite useful and will be used throughout this section.

Gay [1981] has a result similar to Lemma (3.4) but the assumptions are stronger than those in Lemma (3.4). Instead of (3.5), Gay's assumptions imply that $\max\{\| p \|, \| z \|\} < \Delta$ and that

$$\| p + z \| = \Delta \ , \ \| (B + \lambda I)z \| \leqslant \left(\frac{\sigma}{\Delta}\right)(\| Rp \|^2 + \lambda \| p \|^2). \tag{3.7}$$

Since $\| p \| < \Delta$ and $\| z \| < \Delta$, it is not difficult to show that (3.7) implies (3.5). A weakness of (3.7) is that in the hard case it may be a severe restriction on λ. This claim can be made precise by first noting that (3.7) implies that

$$|\lambda + \lambda_1| \| z \| \leqslant \left(\frac{\sigma}{\Delta}\right)(\| R \|^2 + \lambda) \| p \|^2 \ .$$

Now, when $\| p \| \leqslant \epsilon\Delta$ for some ϵ in (0,1) then $\| z \| \geqslant (1-\epsilon)\Delta$ and thus

$$|\lambda + \lambda_1| \leqslant \left(\frac{\sigma}{1-\epsilon}\right)(\| R \|^2 + \lambda)\epsilon^2 \ .$$

Since ϵ can be quite small (specially if g is small) in the hard case, this estimate shows that if ϵ is small then λ must be very close to $-\lambda_1$ before Gay's assumptions are satisfied. As a consequence, an algorithm based on (3.7) may require a large number of iterations in the hard case. Note that this weakness is not shared by (3.5).

The main use of Lemma (3.4) is in the hard case. In this situation we have $\lambda \geqslant 0$ with $B+\lambda I$ positive definite and the solution p of (2.2) satisfies $\|p\| < \Delta$. We can then attempt to satisfy (3.5) with $z=\tau\hat{z}$ by letting τ satisfy $\|p+\tau\hat{z}\| = \Delta$ and by choosing $\hat{z}$ with $\|\hat{z}\| = 1$ such that $\|R\hat{z}\|$ is as small as possible.

Given $\hat{z}$ with $\|\hat{z}\| = 1$ and p with $\|p\| < \Delta$ there are usually two choices of τ which satisfy $\|p+\tau\hat{z}\| = \Delta$, and equation (3.6) implies that the choice with the smaller magnitude minimizes $\psi(p+\tau\hat{z})$. This choice is

$$\tau = \frac{\Delta^2 - \|p\|^2}{p^T\hat{z} + \operatorname{sgn}(p^T\hat{z})[(p^T\hat{z})^2 + (\Delta^2 - \|p\|^2)]^{1/2}} \ .$$

A vector $\hat{z}$ with $\|\hat{z}\| = 1$ such that $\|R\hat{z}\|$ is as small as possible can be obtained with the LINPACK technique (Cline, Moler, Stewart, and Wilkinson [1979]) for estimating the smallest singular value of a triangular matrix R. In this technique a vector e with components ± 1 is selected so that the solution w to the system $R^Tw = e$ is large. Essentially the idea is to select the sign of the components of e to cause maximum local growth of w as the forward substitution proceeds. Then the system $Rv = w$ is solved for v. The vector v has the property that if

$$\hat{z} \equiv \frac{v}{\|v\|},$$

then $\|R\hat{z}\|$ is usually close to the smallest singular value of R. In the appendix to this paper we show that with the LINPACK technique, $\|R\hat{z}\|$ is close to zero if λ is near $-\lambda_1$. This property guarantees that (3.5) is satisfied by $z=\tau\hat{z}$ when λ is sufficiently close to $-\lambda_1$. The LINPACK technique is attractive because it is computationally inexpensive (roughly n^2 arithmetic operations) and quite reliable, but there are other possibilities, for example, the algorithms of Cline, Conn, and Van Loan [1982]. We emphasize that the only property of $\hat{z}$ required by our algorithm is that $\|R\hat{z}\|$ approach zero as λ approaches $-\lambda_1$.

An important ingredient of the iteration is the safeguarding required to ensure that a solution is found. The safeguarding depends on the fact that ϕ is convex and strictly decreasing on $(-\lambda_1,\infty)$. This fact was discovered by Reinsch [1971] and follows from (2.4). It implies that Newton's method started from $\lambda \in (-\lambda_1,\infty)$ with $\phi(\lambda) > 0$ produces a monotonically increasing sequence converging to the solution of $\phi(\alpha) = 0$. In addition, if $\lambda \in (-\lambda_1,\infty)$ and $\phi(\lambda) < 0$ then the next Newton iterate λ_+ is such that either $\lambda_+ \leqslant -\lambda_1$, or $\phi(\lambda_+) \geqslant 0$.

The safeguarding scheme uses parameters λ_L, λ_U, and λ_S such that $[\lambda_L, \lambda_U]$ is an interval of uncertainty which contains the desired λ, and λ_S is a lower bound on $-\lambda_1$.

```
Safeguard λ :
    1) $\lambda := \max(\lambda, \lambda_L)$;
    2) $\lambda := \min(\lambda, \lambda_U)$;
```

```
3) If $\lambda \leqslant \lambda_S$ then $\lambda := \max(0.001\lambda_U, (\lambda_L\lambda_U)^{1/2})$;
```

Safeguarding schemes of this type have been used by Moré [1978] for the case in which B is positive semidefinite and by Gay [1981] for the general case. The first two steps of the safeguarding scheme ensure that $\lambda \in [\lambda_L, \lambda_U]$ while the third step guarantees that the length of the interval of uncertainty is reduced. The third step is crucial; if the length of the interval of uncertainty remains bounded away from zero then the third step can only be executed a finite number of times. This last point will become clear once we set down the rules for updating the safeguarding parameters.

Given λ_S and a trial λ, the rules for revising λ_S are as follows. If $\lambda \in (-\lambda_1, \infty)$ and $\phi(\lambda) < 0$ then $\|p_\lambda\| < \Delta$ and we can compute τ and $\hat{z}$ as described above and set

$$\lambda_S := \max(\lambda_S, \lambda - \|R\hat{z}\|^2). \tag{3.8}$$

Since R is the Cholesky factor of $B + \lambda I$, it follows that for any $\hat{z}$ such that $\|\hat{z}\| = 1$ we have

$$\lambda - \|R\hat{z}\|^2 \leqslant -\lambda_1 .$$

Thus, if λ_S is a lower bound on $-\lambda_1$ then (3.8) guarantees that the updated λ_S is also a lower bound on $-\lambda_1$. If $\lambda \leqslant -\lambda_1$ then we can update λ_S by noting that during the Cholesky decomposition of $B + \lambda I$ it is possible to find $\delta \geqslant 0$ such that the leading submatrix of order $l \leqslant n$ of

$$B + \lambda I + \delta e_l e_l^T$$

is singular. In addition, it is possible to determine $u \in R^n$ such that

$$(B + \lambda I + \delta e_l e_l^T) u = 0$$

with $u_l = 1$ and $u_i = 0$ for $i > l$. It follows that

$$\lambda_S := \max\left(\lambda_S, \lambda + \frac{\delta}{\|u\|^2}\right) \tag{3.9}$$

is a lower bound on $-\lambda_1$.

Gay [1981] proposed updating λ_S via (3.9) but (3.8) is new. Since $\|R\hat{z}\|^2$ is usually close to $\lambda + \lambda_1$, updating λ_S via (3.8) tends to avoid trial λ for which $B + \lambda I$ is not positive definite and thus reduces the number of iterations required for convergence.

The rules for updating λ_L and λ_U are fairly simple; they are presented in the following summary of the updating rules for the safeguarding parameters.

```
Update $\lambda_L$, $\lambda_U$, $\lambda_S$ :
    1) If $\lambda \in (-\lambda_1, \infty)$ and $\phi(\lambda) < 0$ then
            $\lambda_U := \min(\lambda_U, \lambda)$ ;
       else
            $\lambda_L := \max(\lambda_L, \lambda)$;
    2) Update $\lambda_S$ using (3.8) and (3.9);
    3) Let $\lambda_L := \max(\lambda_L, \lambda_S)$;
```

Initial values for the safeguarding parameters are

$$\lambda_S = \max\{-\beta_{ii}\}$$

where β_{ii} is the ith diagonal element of B, and

$$\lambda_L = \max\left\{0, \lambda_S, \frac{\|g\|}{\Delta} - \|B\|_1\right\},$$

$$\lambda_U = \frac{\|g\|}{\Delta} + \|B\|_1 \ .$$

These choices of λ_L and λ_U are similar to those used by Gay [1981]. They are based upon the observation that (2.4) implies that

$$\frac{\|g\|}{|\lambda + \lambda_n|} \leqslant \|p_\lambda\| \leqslant \frac{\|g\|}{|\lambda + \lambda_1|}, \quad -\lambda_1 < \lambda ,$$

where λ_1 and λ_n are, respectively, the smallest and largest eigenvalues of B. Other choices of initial values are possible, but these are simple and are particularly effective for large values of $\|g\|/\Delta$.

The final ingredient of the iteration is the convergence criteria. The idea is to terminate the iteration with a nearly optimal solution of problem (1.1). Given σ_1 and σ_2 in (0,1), and a trial $\lambda \geqslant 0$ such that $B + \lambda I$ is positive definite, a vector p is computed as in Algorithm (3.2). If

$$(3.10) \qquad |\Delta - \|p\|| \leqslant \sigma_1\Delta, \text{ or } \|p\| \leqslant \Delta,\ \lambda = 0,$$

then the algorithm terminates with $s = p$ as an approximate solution. The hard case is taken into account by computing p and $\hat{z}$ whenever $\|p\| < \Delta$, and if

$$(3.11) \qquad \|R(\tau\hat{z})\|^2 \leqslant \sigma_1(2-\sigma_1)\max\{\sigma_2, \|Rp\|^2 + \lambda\Delta^2\}$$

then the algorithm terminates with $s = p + \tau\hat{z}$ as the approximate solution.

An additional subtlety of the convergence tests is that if both (3.10) and (3.11) are satisfied then we choose the approximate solution s for which $\psi(s)$ is least. This is easy to do because (3.6) shows that $\psi(p+\tau\hat{z}) \leqslant \psi(p)$ if and only if

$$\|R(\tau\hat{z})\|^2 \leqslant \lambda(\Delta^2 - \|p\|^2).$$

This subtlety is not theoretically necessary but is nice to have from a computational point of view. Also note that the factor of $\sigma_1(2-\sigma_1)$ in (3.11) is needed so that in each case $\psi(s)$ satisfies a bound of the same form. This is made clear in the discussion that follows.

We now show that (3.10) and (3.11) guarantee that if the algorithm terminates then the approximate solution s satisfies

$$(3.12) \qquad \psi(s) - \psi^* \leqslant \sigma_1(2-\sigma_1)\max\{|\psi^*|, \sigma_2\}, \quad \|s\| \leqslant (1+\sigma_1)\Delta ,$$

and thus s is a nearly optimal solution of (1.1). First consider (3.11). If $\|Rp\|^2 + \lambda\Delta^2 > \sigma_2$ then the assumptions of Lemma (3.4) are satisfied when σ is replaced by $\sigma_1(2-\sigma_1)$ and hence (3.12) holds for $s = p + \tau\hat{z}$. Now suppose that $\|Rp\|^2 + \lambda\Delta^2 \leqslant \sigma_2$. To establish that (3.12) holds in this case, first note that if $\psi^* = \psi(p + z^*)$ where $\|p + z^*\| \leqslant \Delta$ then (3.6) implies that

$$|\psi^*| = -\psi^* \leqslant \tfrac{1}{2}(\|Rp\|^2 + \lambda\Delta^2) \leqslant \tfrac{1}{2}\sigma_2 \ .$$

We now use this result and (3.6) to obtain that

$$\psi(p+\tau\hat{z}) = -½(\|Rp\|^2 + \lambda\Delta^2) + ½\|R(\tau\hat{z})\|^2 \leqslant \psi^* + ½\sigma_1(2-\sigma_1)\sigma_2.$$

Hence, (3.12) also holds in this case.

The next result shows that if the algorithm terminates when (3.10) is satisfied then (3.12) holds with $s = p$.

LEMMA (3.13). *Let* $0 < \sigma < 1$ *be given and suppose that*

$$B + \lambda I = R^T R \ , \ (B + \lambda I)p = -g \ , \lambda \geqslant 0.$$

If ψ^* *is the optimal value of* (1.1) *and if* $\|p\| \geqslant (1-\sigma)\Delta$ *then*

$$-\psi(p) \geqslant ½(1-\sigma)^2(\|Rp\|^2 + \lambda\Delta^2) \geqslant (1-\sigma)^2|\psi^*|.$$

Proof. Just as in the proof of Lemma (3.4), note that (3.6) holds for any $z \in R^n$ and hence,

$$-\psi^* \leqslant ½\left(\|Rp\|^2 + \lambda\Delta^2\right).$$

Moreover, (3.6) with $z = 0$ also implies that

$$-\psi(p) \geqslant ½(1-\sigma)^2(\|Rp\|^2 + \lambda\Delta^2).$$

The last two inequalities yield Lemma (3.13). ■

We have now discussed all the ingredients of the iterative scheme for solving problem (1.1). The following algorithm summarizes these ingredients and defines a typical iteration.

ALGORITHM (3.14).

```
1) Safeguard $\lambda$;
2) If $B + \lambda I$ is positive definite then factor $B + \lambda I = R^T R$ ;
   otherwise go to 5;
3) Solve $R^T R p = -g$;
4) If $\|p\| < \Delta$, compute $\tau$ and $\hat{z}$ ;
5) Update $\lambda_L, \lambda_U, \lambda_S$;
6) Check the convergence criteria;
7) If $B + \lambda I$ is positive definite and $g \neq 0$ then update $\lambda$ via steps
   3 and 4 of Algorithm (3.2); otherwise update $\lambda$ via $\lambda := \lambda_S$ ;
```

The last step of Algorithm (3.14) deserves some explanation. If $B + \lambda I$ is positive definite and $g \neq 0$ then the Newton iterate of Algorithm (3.2) tends to be a lower bound on $-\lambda_1$ for $\|g\|$ sufficiently small and thus updating λ via λ_S is a reasonable choice when $g = 0$. Also note that setting λ to λ_S forces a safeguarded choice of λ in the next iteration, and that this is a desirable strategy whenever the Newton iteration cannot be used.

We now claim that after a finite number of iterations Algorithm (3.14) produces a $\lambda \in (-\lambda_1,\infty)$ with $\phi(\lambda) \geqslant 0$ or an arbitrarily small interval of uncertainty. If we assume that the length of the interval of uncertainty remains bounded away from zero then the third step of the safeguarding scheme guarantees that $\lambda \leqslant \lambda_S$ only happens a finite number of times. Now, if $\lambda \leqslant -\lambda_1$ then $\lambda \leqslant \lambda_S$ holds on the next iteration. Finally, if $\lambda \in (-\lambda_1,\infty)$ and $\phi(\lambda) < 0$ then recall that the next

Newton iterate λ_+ is such that either $\lambda_+ \leqslant -\lambda_1$, or $\phi(\lambda_+) \geqslant 0$. The above argument shows that $\lambda_+ \leqslant -\lambda_1$ can only happen a finite number of times, so eventually $\lambda_+ \in (-\lambda_1,\infty)$ and $\phi(\lambda_+) \geqslant 0$. This establishes our claim.

The importance of the above claim should be evident; given $\lambda \in (-\lambda_1,\infty)$ with $\phi(\lambda) \geqslant 0$ then Algorithm (3.14) eventually satisfies (3.10), while if the interval of uncertainty is small then R is nearly singular and it is then possible to satisfy (3.11). Thus Algorithm (3.14) terminates in a finite number of iterations with an approximate solution s which satisfies (3.12).

A frequent application of Algorithm (3.14) is to the solution of a sequence of problems of the form (1.1) in which only Δ is changing. In particular, in trust region methods we need to solve a sequence of problems for decreasing values of Δ and then it is possible to improve the initial choice of λ_L. Assume that λ and λ_L are the final values of these parameters for a specific value of Δ. Given a new value $\Delta_+ < \Delta$ then λ_L is still a lower bound for the new problem. Moreover, the convexity and monotonicity of ϕ shows that an update of λ based on a Newton step for

$$\phi_+(\alpha) \equiv \frac{1}{\Delta_+} - \frac{1}{\| p_\alpha \|}$$

is also a lower bound for the new problem. This improvement on the initial choice of λ_L follows a suggestion of Ron Dembo.

One of the differences between Gay's [1981] algorithm and Algorithm (3.14) is that in Gay's algorithm $\lambda = 0$ is always tried first. It is not at all clear that this is a desirable strategy, and it seems preferable to try $\lambda = 0$ first only if the safeguarded λ is zero. Note that if B is positive definite and $\| B^{-1}g \| \leqslant \Delta$ then Algorithm (3.14) terminates in at most two iterations. In fact, if initially $\lambda > 0$ then the convexity and monotonicity of ϕ and the positive definiteness of B guarantee that the next trial λ is zero.

We have already mentioned that another difference is the updating of λ_S via (3.8). A final difference occurs when $g = 0$; Gay's algorithm is not defined in this situation, but Algorithm (3.14) handles this case appropriately.

4. Trust region methods in unconstrained minimization.

We now consider the use of Algorithm (3.14) in the context of trust region methods for unconstrained minimization and show how Algorithm (3.14) can be used to produce an efficient and reliable version of Newton's method.

Let $f : R^n \to R$ be a twice continuously differentiable function with gradient ∇f and Hessian $\nabla^2 f$. In Newton's method with a trust region strategy, each iterate x_k has a bound Δ_k such that

$$f(x_k + w) \approx f(x_k) + \psi_k(w), \quad \| w \| \leqslant \Delta_k \;,$$

where

$$\psi_k(w) = \nabla f(x_k)^T w + \tfrac{1}{2} w^T \nabla^2 f(x_k) w \;.$$

In other words, ψ_k is a model of the reduction in f within a neighborhood of the iterate x_k. This suggests that it may be desirable to compute a step s_k which approximately solves the problem

$$\min\{\psi_k(w) : \| w \| \leqslant \Delta_k\}. \tag{4.1}$$

If the step is satisfactory in the sense that $x_k + s_k$ produces a sufficient reduction in f, then Δ_k can be increased; if the step is unsatisfactory then Δ_k should be decreased.

ALGORITHM(4.2). *Let* $0 < \mu < \eta < 1$ and $0 < \gamma_1 < \gamma_2 < 1 < \gamma_3$ *be specified constants.*

```
1) Let $x_0 \in R^n$ and $\Delta_0 > 0$ be given.
2) For $k = 0,1,2, \cdots$ until"convergence"
    a) Compute $\nabla f(x_k)$ and $\nabla^2 f(x_k)$.
    b) Determine an approximate solution $s_k$ to problem (4.1).
    c) Compute $\rho_k = (f(x_k+s_k) - f(x_k))/ \psi_k(s_k)$ .
    d) If $\rho_k \leqslant \mu$ then $\Delta_k := \Delta \in [\gamma_1\Delta_k,\gamma_2\Delta_k]$ and go to b).
    e) $x_{k+1} = x_k + s_k$.
    f) If $\rho_k \leqslant \eta$ then $\Delta_{k+1} \in [\gamma_2\Delta_k,\Delta_k]$ else $\Delta_{k+1} \in [\Delta_k,\gamma_3\Delta_k]$.
```

This is a basic form of Newton's method which does not include a scaling matrix for the variables. To include a scaling matrix, subproblem (4.1) is replaced by

$$\min\{\psi_k(w) : \| D_k w \| \leqslant \Delta_k\}$$

where D_k is a nonsingular matrix. We shall not discuss this generalization here; however, it is important to note that our results hold if $\{D_k\}$ has uniformly bounded condition numbers.

In this section we are mainly interested in conditions on the approximate solution s_k of problem (4.1) which guarantee that the sequence $\{x_k\}$ generated by Algorithm (4.2) is convergent to a point x^* with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ positive semidefinite. A minimal requirement on s_k is that there is a $\beta \in (0,1)$ such that

$$-\psi(s_k) \geqslant \beta\max\{-\psi(w) : w = \alpha\nabla f(x_k) , \| w \| \leqslant \Delta_k\}, \quad \| s_k \| \leqslant \Delta_k \ .$$

Under this assumption, Powell [1975] proved that if $\mu = 0$ then some subsequence of $\{\nabla f(x_k)\}$ converges to zero, while Thomas [1975] showed that if $\mu > 0$ then the whole sequence $\{\nabla f(x_k)\}$ converges to zero. These results indicate that we can expect $\{x_k\}$ to converge to a point x^* with $\nabla f(x^*) = 0$. Sorensen [1982] proved that we can also expect to have $\nabla^2 f(x^*)$ positive semidefinite provided there is a constant $\sigma \in (0,1)$ such that

$$\psi_k(s_k) = \min\{\psi_k(w) : \| w \| \leqslant \delta_k\}$$

with

$$\| s_k \| \leqslant \delta_k \in [(1-\sigma)\Delta_k,(1+\sigma)\Delta_k] .$$

Unfortunately the termination criterion (3.11) is not necessarily consistent with these conditions and thus this result does not allow the choice of s_k provided by Algorithm (3.14). An appropriate generalization of Sorensen's results can be obtained by assuming that there are constants $\beta_1 > 0$ and $\beta_2 > 0$ such that

$$-\psi_k(s_k) \geqslant \beta_1 |\psi_k^*| \quad \text{with} \quad \| s_k \| \leqslant \beta_2 \Delta_k. \tag{4.3}$$

An immediate consequence of (3.12) is that if $\psi_k^* \neq 0$ then the approximate solution s_k provided by Algorithm (3.14) with $\sigma_2 = 0$ satisfies (4.3). Of course, if

$\psi_k^* = 0$ then $\nabla f(x_k) = 0$ and $\nabla^2 f(x_k)$ is positive semidefinite and thus Algorithm (4.2) terminates at x_k. Lemma (3.13) shows that Sorensen's assumptions imply that (4.3) holds.

Assumption (4.3) can be expressed in an alternate form which is more convenient for proofs of convergence. If $p_k \in R^n$ is a solution to problem (4.1) then Lemma (2.1) implies that there is a parameter λ_k such that

$$\nabla^2 f(x_k) + \lambda_k I = R_k^T R_k \ , \ (\nabla^2 f(x_k) + \lambda_k I) p_k = -\nabla f(x_k) \ , \ \lambda_k \geqslant 0$$

and with $\lambda_k(\Delta_k - \| p_k \|) = 0$. A calculation now shows that

$$|\psi_k^*| = \tfrac{1}{2}(\| R_k p_k \|^2 + \lambda_k \Delta_k^2). \tag{4.4}$$

This expression for ψ_k^* shows that if (4.3) holds then

$$-\psi_k(s_k) \geqslant \tfrac{1}{2}\beta_1(\| R_k p_k \|^2 + \lambda_k \Delta_k^2), \tag{4.5}$$

and thus the iterates $\{x_k\}$ generated by Algorithm (4.2) satisfy

$$f(x_k) - f(x_{k+1}) \geqslant \tfrac{1}{2}\mu\beta_1(\| R_k p_k \|^2 + \lambda_k \Delta_k^2). \tag{4.6}$$

These two inequalities are essential to the proof of our next result.

THEOREM(4.7). *Let $f : R^n \rightarrow R$ be twice continuously differentiable on the level set $\Omega = \{x : f(x) \leqslant f(x_0)\}$ and consider the sequence $\{x_k\}$ produced by Algorithm* (4.2) *where s_k satisfies* (4.3). *If Ω is a compact set then either the algorithm terminates at $x_l \in \Omega$ because $\nabla f(x_l) = 0$ and $\nabla^2 f(x_l)$ is positive semidefinite , or $\{x_k\}$ has a limit point $x^* \in \Omega$ with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ positive semidefinite.*

Proof. If $\nabla f(x_l) = 0$ and $\nabla^2 f(x_l)$ is positive semidefinite for some iterate $x_l \in \Omega$ then the algorithm terminates; otherwise (4.3) implies that $\psi_k(s_k) < 0$ for $k \geqslant 0$ and thus $\{x_k\}$ is well defined and lies in Ω.

Let us now prove the result under the assumption that $\{\lambda_k\}$ is not bounded away from zero. If some subsequence of $\{\lambda_k\}$ converges to zero then since Ω is compact we can assume, without loss of generality, that the same subsequence of $\{x_k\}$ converges to some x^* in the level set Ω. Since $\nabla^2 f(x_k) + \lambda_k I$ is positive semidefinite, $\nabla^2 f(x^*)$ is also positive semidefinite , and $\nabla f(x^*) = 0$ follows by noting that

$$\| R_k p_k \|^2 \geqslant \frac{\| \nabla f(x_k) \|^2}{\| \nabla^2 f(x_k) \| + \lambda_k}$$

and that (4.6) implies that $\{\| R_k p_k \|\}$ converges to zero.

We can show that $\{\lambda_k\}$ is not bounded away from zero by contradiction. If $\lambda_k \geqslant \epsilon > 0$ then (4.3) and (4.5) yield that

$$-\psi_k(s_k) \geqslant \tfrac{1}{2}\beta_1 \lambda_k \Delta_k^2 \geqslant \tfrac{1}{2}\left(\frac{\beta_1}{\beta_2^2}\right)\epsilon \| s_k \|^2.$$

Now, a standard estimate is that

$$\begin{aligned} &|f(x_k+s_k) - f(x_k) - \psi_k(s_k)| \\ &\qquad \leqslant \tfrac{1}{2}\| s_k \|^2 \max_{0 \leqslant \xi \leqslant 1} \| \nabla^2 f(x_k + \xi s_k) - \nabla^2 f(x_k) \|, \end{aligned} \tag{4.8}$$

and thus the last two inequalities show that

$$\left|\rho_k - 1\right| \leqslant \left(\frac{\beta_2^2}{\beta_1\epsilon}\right) \max_{0\leqslant\xi\leqslant 1} \| \nabla^2 f(x_k+\xi s_k) - \nabla^2 f(x_k) \| \ . \tag{4.9}$$

Inequality (4.6) implies that $\{\Delta_k\}$ converges to zero and hence $\{\| s_k \|\}$ also converges to zero. Thus the uniform continuity of $\nabla^2 f$ on Ω together with (4.9) implies that $\rho_k > \eta$ for all k sufficiently large and then the updating rules for Δ_k yield that $\{\Delta_k\}$ is bounded away from zero. This is in contradiction of the fact that $\{\Delta_k\}$ converges to zero. ■

The result we have just established is only a sample of the available results for Algorithm (4.7) under assumption (4.3) for s_k. All of the results of Sorensen [1982] hold, and in particular, it can be shown that if f has a finite number of critical points in the level set Ω then every limit point of the sequence $\{x_k\}$ satisfies the second order necessary conditions. We now prove a stronger version of this result.

LEMMA(4.10). *Let x^* be an isolated limit point of a sequence $\{x_k\}$ in R^n. If $\{x_k\}$ does not converge then there is a subsequence $\{x_{l_j}\}$ which converges to x^* and an $\epsilon > 0$ such that $\| x_{l_j+1} - x_{l_j} \| \geqslant \epsilon$.*

Proof. Choose $\epsilon > 0$ so that if $\| x - x^* \| \leqslant \epsilon$ and x is a limit point of $\{x_k\}$ then $x = x^*$. If $\| x_{k_j} - x^* \| \leqslant \epsilon$ then define l_j by

$$l_j = \max\{ l : \| x_i - x^* \| \leqslant \epsilon,\ i = k_j, \ldots, l\}.$$

In this manner, a subsequence $\{x_{l_j}\}$ is defined such that

$$\| x_{l_j} - x^* \| \leqslant \epsilon,\ \ \| x_{l_j+1} - x^* \| > \epsilon,$$

It follows that $\{x_{l_j}\}$ converges to x^* and thus $\| x_{l_j} - x^* \| \leqslant ½\epsilon$ for all l_j sufficiently large. Hence,

$$\| x_{l_j+1} - x_{l_j} \| \geqslant \| x_{l_j+1} - x^* \| - \| x_{l_j} - x^* \| \geqslant ½\epsilon$$

as desired. ■

THEOREM(4.11). *Let $f : R^n \rightarrow R$ be twice continuously differentiable on the level set $\Omega = \{x : f(x) \leqslant f(x_0)\}$ and consider the sequence $\{x_k\}$ produced by Algorithm (4.2) where s_k satisfies (4.3). If x^* is an isolated limit point of $\{x_k\}$ then $\nabla^2 f(x^*)$ is positive semidefinite.*

Proof. If $\{x_k\}$ converges to x^* then Theorem (4.7) shows (the compactness of Ω is only used to guarantee that $\{x_k\}$ is bounded) that $\nabla^2 f(x^*)$ is positive semidefinite. If $\{x_k\}$ does not converge then Lemma (4.10) applies. Thus, if $\{x_{l_j}\}$ is the subsequence guaranteed by Lemma (4.10) then $\Delta_{l_j} \geqslant \beta_2\epsilon$, and since (4.6) shows that $\{\lambda_k \Delta_k^2\}$ converges to zero, $\{\lambda_{l_j}\}$ must then converge to zero. We can now use the positive semidefiniteness of $\nabla^2 f(x_k) + \lambda_k I$ to conclude that $\nabla^2 f(x^*)$ is positive semidefinite. ■

We have already noted that Thomas [1975] proved that $\{\nabla f(x_k)\}$ converges to zero. Hence, if $\nabla^2 f(x^*)$ is nonsingular at a limit point x^* of $\{x_k\}$ then x^* is an isolated limit point, and Theorem (4.11) shows that $\nabla^2 f(x^*)$ is positive definite.

Since $\psi_k(s_k) \leqslant 0$ we have that

$$\tfrac{1}{2}\| s_k \| \leqslant \| \nabla^2 f(x_k)^{-1} \| \, \| \nabla f(x_k) \| \tag{4.12}$$

whenever $\nabla^2 f(x_k)$ is positive definite, and thus Lemma (4.10) shows that $\{x_k\}$ converges to x^*. This establishes the following result.

THEOREM(4.13). *Let $f : R^n \rightarrow R$ be twice continuously differentiable on the level set $\Omega = \{x : f(x) \leqslant f(x_0)\}$ and consider the sequence $\{x_k\}$ produced by Algorithm (4.2) where s_k satisfies (4.3). If x^* is a limit point of $\{x_k\}$ and $\nabla^2 f(x^*)$ is nonsingular then $\{x_k\}$ converges to x^* and $\nabla^2 f(x^*)$ is positive definite.*

An additional result which is helpful in establishing rate of convergence results is that under the assumptions of Theorem (4.13) the sequence $\{\Delta_k\}$ is bounded away from zero. To prove this first note that if $\epsilon_0 > 0$ is a lower bound on the eigenvalues of $\nabla^2 f(x_k)$ then (4.4) shows that

$$|\psi_k^*| \geqslant \tfrac{1}{2}\epsilon_0 \min\{\Delta_k^2, \| s_k^N \|^2\}$$

where

$$s_k^N \equiv -\nabla^2 f(x_k)^{-1} \nabla f(x_k).$$

Now, (4.12) implies that $\tfrac{1}{2}\| s_k \| \leqslant \kappa \| s_k^N \|$ where κ is an upper bound on the condition number of $\nabla^2 f(x_k)$, and hence assumption (4.3) shows that there is a constant $\epsilon_1 > 0$ with

$$-\psi_k(s_k) \geqslant \tfrac{1}{2}\epsilon_1 \| s_k \|^2.$$

This estimate and (4.8) then yield that

$$\left| \rho_k - 1 \right| \leqslant \left(\frac{1}{\epsilon_1} \right) \max_{0 \leqslant \xi \leqslant 1} \| \nabla^2 f(x_k + \xi s_k) - \nabla^2 f(x_k) \| ,$$

and thus $\rho_k > \eta$ for all k sufficiently large. It follows that $\{\Delta_k\}$ is bounded away from zero as desired.

Rate of convergence results can be obtained with the additional - but mild - assumption that there is a constant $\beta_3 > 0$ such that if $\| s_k \| \leqslant \beta_3 \Delta_k$ then $s_k = s_k^N$. For example, Algorithm (3.14) satisfies this assumption because if we have $\| s_k \| < (1-\sigma_1)\Delta_k$ then $\lambda_k = 0$. In particular, note that if $\| s_k \| < \Delta_k$ then (3.11) cannot be satisfied because then Algorithm (3.14) would set $\| s_k \| = \Delta_k$.

Under the above assumption, Theorem (4.13) can be extended to show that $\{x_k\}$ converges to x^* with a Q-superlinear rate of convergence and that if $\nabla^2 f$ is Lipschitz continuous then the rate of convergence is quadratic. The proof is not difficult; since $\tfrac{1}{2}\| s_k \| \leqslant \kappa \| s_k^N \|$ where κ is an upper bound on the condition number of the Hessian matrix at x_k, eventually $\| s_k \| \leqslant \beta_3 \Delta_k$, and then $\{x_k\}$ becomes an unmodified Newton's method. The rate of convergence results are then standard.

5. Computational results.

Algorithm (3.14) has been implemented in a Fortran subroutine GQTPAR, and in this section we present the results of GQTPAR on two sets of test problems. Since our main concern here is the performance of GQTPAR in a trust region method, we used $\sigma_1 = 0.1$ and $\sigma_2 = 0$ in the convergence criteria (3.10) and (3.11). The reason for setting $\sigma_2 = 0$ is that σ_2 is only required to deal with the case where $g = 0$ and B is positive semidefinite and singular, and in this situation, a trust region method terminates. The initial choice of λ depends on the test and is described below. All tests were performed in double precision on a VAX 11/780. This provides an accuracy of about 17 significant decimal digits.

The first set of tests is concerned with the performance of GQTPAR in the context of a trust region Newton's method. The test problems used are the 18 unconstrained minimization problems described in Moré, Garbow, and Hillstrom [1981a]. For each problem it is possible to specify a set of starting points, and in 8 of the problems it is also possible to specify the dimension. A particular set of test cases is defined by the data provided to the test drivers. We used the sample data provided in Moré, Garbow, and Hillstrom [1981b] which defines 52 test cases. The dimensions of the problems specified in this set of tests range from 2 to 12.

The trust region Newton's method used follows Algorithm (4.2) and proved to be quite successful on these problems. Details of the Newton method will appear elsewhere. For the purposes of this paper it suffices to remark that on the first call to GQTPAR the initial λ is zero, but on succeeding calls the initial λ is the same as the final λ from the previous call of GQTPAR. At the end of Section 3 we pointed out that it is possible to obtain a more educated guess for the initial λ, but this choice provides a stringent test of GQTPAR.

The performance of GQTPAR on these problems was very satisfactory. There were 2580 calls to GQTPAR and the average number of iterations per call was 1.63; the largest number of iterations was 10. In about 20% of the calls convergence criterion (3.11) was satisfied.

The second set of tests is designed to exercise the various features of GQTPAR as an individual algorithm on problems of type (1.1). For these problems we decided to use

$$\lambda = \frac{\| g \|}{\Delta} \tag{5.1}$$

as the initial λ. Unless other information is available, this is a reasonable automatic choice. In these problems we generated sequences of uniformly distributed random numbers with the RAND function of Schrage [1979]. Given an integer seed, RAND generates a random number in (0,1) and changes the seed. Thus a sequence of random numbers can be generated by repeated calls to RAND.

A convenient way to define a problem of type (1.1) is to set $B = QDQ^T$ for some orthogonal matrix Q and diagonal matrix D, and to then let $g = Q\hat{g}$ for some vector $\hat{g}$. This makes it possible to generate a (potentially) hard case by setting to zero the component of $\hat{g}$ corresponding to the smallest element of D. The structure of B is scrambled by choosing the orthogonal matrix Q of the form $Q_1 Q_2 Q_3$ where

$$Q_j = I - 2\frac{w_j w_j^T}{\| w_j \|^2} \ , \ j = 1,2,3,$$

and the components of w_j are random numbers uniformly distributed in $(-1,1)$. A problem of type (1.1) can be generated be specifying Δ, $\hat{g}$, and D; different choices lead to problems with various characteristics.

We consider four different ways of specifying $\hat{g}$ and D. In all four cases, the elements of $\hat{g}$ and D are initially chosen as uniformly distributed random numbers in $(-1,1)$. This choice leads to the general case; as mentioned above, a hard case can then be obtained by setting to zero the component of $\hat{g}$ corresponding to the smallest element of D. A positive definite case is obtained by replacing D by $|D|$, and in the saddle point case all the components of g are set to zero.

The choice of Δ is critical; if Δ is chosen from $(0,1)$ then the tests are easy because (5.1) is almost always an excellent choice. A harder test is obtained if Δ is chosen as uniformly distributed from $(0,100)$, and this choice is made in our tests. We have observed that a wider distribution in the choice of Δ does not affect the results significantly, and that the range $(0,100)$ appears to be the hardest choice for these problems.

We now present the results of tests in each of the above four cases and for dimensions 10, 20, 40, 60, 80, and 100. For each case and each dimension we generated 5 problems and recorded both the average and the maximum number of iterations required for convergence. The results are presented in the tables below.

TABLE 1
The general case.

	Number of iterations	
Dimension	Average	Maximum
10	2.0	4
20	2.6	5
40	3.2	4
60	3.0	4
80	3.2	4
100	4.0	5

TABLE 2
The hard case.

	Number of iterations	
Dimension	Average	Maximum
10	1.6	3
20	2.2	3
40	3.0	3
60	2.8	3
80	3.2	4
100	3.2	4

TABLE 3
The saddle point case.

	Number of iterations	
Dimension	Average	Maximum
10	1.6	3
20	2.0	2
40	2.6	3
60	3.0	4
80	3.6	4
100	3.2	4

TABLE 4
The positive definite case.

	Number of iterations	
Dimension	Average	Maximum
10	2.4	4
20	2.0	2
40	2.4	3
60	2.4	3
80	2.4	3
100	3.0	4

An interesting aspect of the results for the general case is that Algorithm (3.14) terminated on condition (3.11) in 26 out of the 30 cases. This shows that (3.11) is powerful enough to terminate the algorithm even on nonhard cases. For smaller values of σ_1, however, it is more difficult to satisfy (3.11) and this gives GQTPAR a chance to produce an iterate $\lambda > -\lambda_1$ with $\phi(\lambda) > 0$. Once this occurs, the Newton iteration converges quadratically and (3.10) is eventually satisfied. As noted above, the results improve for smaller choices of Δ, and for example, if Δ is chosen from (0,1) then the maximum number of iterations is 2.

The results of Table 2 show that the hard case, once recognized and treated properly, can be handled with the same computational effort as the general case. In contrast to the general case, the results for the hard case are sensitive to the choice of σ_1 since in this case it is necessary to determine λ_1 and Algorithm (3.14) determines λ_1 with a bisection-type process. Another interesting point is that for these problems Algorithm (3.14) does not always terminate on condition (3.11) since the hard case only occurs if $\Delta > \|p_\alpha\|$ for $\alpha \geqslant -\lambda_1$. This situation is avoided in the saddle point case by choosing $g = 0$.

The saddle point case is unusual because the algorithm and the results are independent of the choice of Δ, and termination always occurs on condition

(3.11). Although setting $g = 0$ is an extreme choice, the numerical results are insensitive to the choice of g provided the components of g are sufficiently small. For example, if the components of g are chosen from $(-10^{-8},10^{-8})$ then the number of iterations increases by 1 in two of the problems, but otherwise the results are unchanged.

In the positive definite case, the choice of Δ as a uniformly distributed random number from (0,100) resulted in exits with $\lambda = 0$ in about half the problems, and this explains why the average number of iterations is close to 2. On the other hand, if Δ is chosen from (0,1) then (5.1) leads to termination on the first iteration.

These results show that GQTPAR performs adequately in all cases. As expected, a smaller value of σ_1 requires more iterations, but the increase is surprisingly small in most cases. The choice $\sigma_1 = 0.1$ is very satisfactory in many cases since it does not require a large number of iterations and produces a nearly optimal approximate solution as predicted by the theory.

6. Concluding remarks.

We have presented an algorithm for the constrained quadratic minimization problem (1.1) and reported the computational results of the implementation GQTPAR. This implementation uses the Cholesky factorization to solve systems of the form

$$(B + \lambda I)u = v \ ,$$

but it is possible to use other factorizations. For example, the decomposition

$$B = QTQ^T \tag{6.1}$$

where Q is orthogonal and T is tridiagonal leads to systems of the form

$$(T + \lambda I)w = Q^T v \ , \quad u = Qw,$$

and since Algorithm (3.14) is invariant with respect to orthogonal transformations, it is possible to produce an implementation which only requires on the order of n arithmetic operations per iteration. We have not used this factorization because we expect GQTPAR to be used in a trust region method , and in this case our numerical results show that a call to GQTPAR requires less than two Cholesky factorizations on the average.

Another argument against the use of factorization (6.1) is that it usually ignores the structure of B. In particular, for sparse systems the Cholesky factorization offers many advantages. Good software based on the Cholesky factorization currently exists for the solution of positive definite linear systems, and this together with an estimator of the smallest singular value of a sparse upper triangular matrix is all that is required to provide a trust region Newton's method for optimization problems with a sparse Hessian matrix.

It would be of interest to develop a method for large scale problems of type (1.1) which does not require the solution of linear systems. Iterative approaches along the lines of conjugate directions or Lanczos type methods have been considered, but a complete solution is not known to us.

Appendix. The purpose of this appendix is to prove that if $\hat{z}$ is the LINPACK estimate of a right singular vector of R corresponding to the smallest singular value, then $\|R\hat{z}\|$ is near zero when R is nearly singular.

As mentioned in Section 3, the LINPACK technique constructs a vector e with components ± 1 such that the solution w to the system $R^T w = e$ is large. Then the system $Rv = w$ is solved for v, and

$$\hat{z} = \frac{v}{\|v\|}$$

is the LINPACK estimate of the singular vector. If R is exactly singular then the following description must be modified, but in this case $R\hat{z} = 0$ is obtained so there is nothing to prove. Therefore, we now assume that R is nonsingular and show that if ρ_{ij} is the (i,j) element of R and

$$\text{(A.1)} \qquad \sum_{j=1}^{n} \sum_{i=1}^{j} |\rho_{ij}| \leqslant \rho ,$$

then

$$\text{(A.2)} \qquad \|R\hat{z}\| \leqslant n^{1/2}(1+\rho)\min\{|\rho_{kk}| : 1 \leqslant k \leqslant n\} .$$

Thus, if $B + \lambda I = R^T R$ and λ is near $-\lambda_1$ then some diagonal element of R is near zero and (A.2) implies that $\|R\hat{z}\|$ is near zero.

To establish (A.2) we first note that for any vectors v and w such that $R^T w = e$ and $Rv = w$ we have

$$\text{(A.3)} \qquad \|w\|^2 \leqslant n^{1/2}\|v\| .$$

In fact, the Cauchy-Schwarz inequality shows that

$$\|w\|^2 = \|R^{-T}e\|^2 = e^T R^{-1} R^{-T} e \leqslant \|e\| \|v\| ,$$

and thus (A.3) follows because each of the components of e are ± 1 . Also note that (A.3) implies that

$$\text{(A.4)} \qquad \|R\hat{z}\| = \frac{\|w\|}{\|v\|} \leqslant \frac{n^{1/2}}{\|w\|} .$$

We now show that the LINPACK technique for selecting w does cause $\|w\|$ to grow if R is nearly singular. To prove this, recall (Cline, Moler, Stewart, and Wilkinson [1979]) that if $\omega_1, \ldots, \omega_{k-1}$ have been specified then we compute

$$p_i^{(k-1)} = \sum_{l=1}^{k-1} \rho_{li} w_l , \quad k \leqslant i \leqslant n ,$$

and consider two possible choices of w_k:

$$\omega_k^+ = \frac{1 - p_k^{(k-1)}}{\rho_{kk}} , \qquad \omega_k^- = \frac{-1 - p_k^{(k-1)}}{\rho_{kk}} .$$

If

$$|\omega_k^+| + \sum_{i=k+1}^{n} |p_i^{(k-1)} + \rho_{ki}\omega_k^+| \geqslant |\omega_k^-| + \sum_{i=k+1}^{n} |p_i^{(k-1)} + \rho_{ki}\omega_k^-|$$

then $\omega_k = \omega_k^+$ is chosen, and otherwise $\omega_k = \omega_k^-$ is chosen. Since

$$\max\{|\omega_k^+|, |\omega_k^-|\} \geqslant \frac{1}{|\rho_{kk}|},$$

it follows that

$$\frac{1}{|\rho_{kk}|} \leqslant |\omega_k| + \sum_{i=k+1}^{n} |p_i^{(k-1)} + \rho_{ki}\omega_k| .$$

Thus, in view of (A.1) we have that

$$\frac{1}{|\rho_{kk}|} \leqslant (1+\rho)\|w\| , \quad 1 \leqslant k \leqslant n.$$

This inequality together with (A.4) shows that (A.2) holds as desired.

Estimate (A.2) is quite crude, and it certainly is not being offered as an indication of the accuracy of the LINPACK estimate. The only purpose of (A.2) is to validate the use of the LINPACK estimate in Algorithm (3.14). It is of interest to note that the same proof techniques show that the look-behind algorithm (with unit weights) of Cline, Conn, and Van Loan [1982] also satisfies an estimate of the form (A.2).

Acknowledgment. We would like to thank David Gay for his comments on a preliminary version of this manuscript and for sharing his UNIX expertise with us.

REFERENCES

A. K. Cline, A. R. Conn, and C. Van Loan, [1982], *Generalizing the LINPACK condition estimator,* Numerical Analysis, J. P. Hennart, ed., Lecture Notes in Mathematics 909, Springer-Verlag, Berlin.

A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson, [1979], *An estimate for the condition number of a matrix,* SIAM J. Numer. Anal., 16, pp. 368–375.

R. Fletcher [1980], *Practical Methods of Optimization, Volume* 1: *Unconstrained Optimization,* John Wiley, New York.

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

S. Goldfeld, R. Quandt and H. Trotter, [1966], *Maximization by quadratic hill climbing,* Econometrica, 34, pp. 541–551.

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

J. J. Moré [1978], *The Levenberg-Marquardt algorithm: Implementation and theory,* Proceedings of the Dundee Conference on Numerical Analysis, G. A. Watson, ed., Springer-Verlag, Berlin.

J. J. Moré, B. S. Garbow and K. E. Hillstrom [1981a], *Testing unconstrained optimization software,* ACM Trans. Math. Software, 7, pp. 17–41.

J. J. Moré, B. S. Garbow, and K. E. Hillstrom, [1981b], *Fortran subroutines for testing unconstrained optimization software,* ACM Trans. Math. Software, 7, pp. 136–140.

M. J. D. Powell [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.

C. H. Reinsch [1967], *Smoothing by spline functions,* Numer. Math., 10, pp. 177–183.

——— [1971], *Smoothing by spline functions* II, Numer. Math., 16, pp. 451–454.

L. Schrage [1979], *A more portable Fortran random number generator,* ACM Trans. Math. Software, 5, pp. 132–138.

D. C. Sorensen, [1982], *Newton's method with a model trust-region modification,* SIAM J. Numer. Anal., 19, pp. 409–426.

——— [1982], *Trust region methods for unconstrained minimization,* in Nonlinear Optimization 1981, M. J. D. Powell, ed., Academic Press, New York.

S. W. Thomas [1975], *Sequential estimation techniques for quasi-Newton algorithms,* Ph.D. dissertation, Cornell University, Ithaca, NY.