Combining Trust Region and Line Search Techniques[*]

Jorge Nocedal[†] and Ya-xiang Yuan[‡]

March 14, 1998

Report OTC 98/04
Optimization Technology Center

Abstract

We propose an algorithm for nonlinear optimization that employs both trust region techniques and line searches. Unlike traditional trust region methods, our algorithm does not resolve the subproblem if the trial step results in an increase in the objective function, but instead performs a backtracking line search from the failed point. Backtracking can be done along a straight line or along a curved path. We show that the new algorithm preserves the strong convergence properties of trust region methods. Numerical results are also presented.

Key words: trust region methods, backtracking, unconstrained optimization, nonlinear optimization.

[*] To appear in *Advances in Nonlinear Programming*, (Kluwer, 1998), ed. Y. Yuan, pp. 153–175. This work was supported by National Science Foundation grants CCR-9101359 and CCR-9625613, and by Department of Energy grant DE-FG02-87ER25047.

[†] Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208-3118, USA.

[‡] Computing Center, Academia Sinica, Beijing 100080, China.

1. Introduction

In this paper we study a new type of trust region method for solving the unconstrained optimization problem,

$$\min_{x \in \Re^n} f(x). \tag{1.1}$$

Our goal is to design an algorithm that retains the excellent convergence properties of trust region methods, but is more economical to implement when the number of variables is large.

A trust region method calculates a trial step by solving the subproblem

$$\min_{d \in \Re^n} g_k^T d + \frac{1}{2} d^T B_k d \equiv \phi_k(d) \tag{1.2}$$

$$\text{subject to } ||d||_2 \leq \Delta_k, \tag{1.3}$$

where $g_k = \nabla f(x_k)$ is the gradient of the objective function at the current approximate solution, B_k is an $n \times n$ symmetric matrix which approximates the Hessian of f, and $\Delta_k > 0$ is a trust region radius. One of the advantages of trust region methods, as compared with line search methods, is that B_k is allowed to be indefinite.

After obtaining a *trial step* d_k, which is an exact or approximate solution of (1.2)-(1.3), trust region algorithms compute the ratio ρ_k between the actual reduction in the function, $f(x_k) - f(x_k + d_k)$, and the predicted reduction, $\phi_k(0) - \phi_k(d_k)$. The trust region radius Δ_k is updated according to the value of this ratio ρ_k. Now, if the step d_k is not successful, that is if $f(x_k + d_k) \geq f(x_k)$, one rejects the step, sets $x_{k+1} = x_k$, reduces the trust region radius, and resolves the problem (1.2)-(1.3). This strategy is quite adequate for small problems.

However, if the number of variables is large, resolving the trust region problem can be costly, since this requires solving one or more linear systems of the form

$$(B_k + \lambda I)d = -g_k \tag{1.4}$$

(see for example Dennis and Schnabel (1983)). In contrast, line search methods require very little computation to determine a new trial point. Therefore we ask how to incorporate backtracking line searches in a trust region method, so as to avoid resolving the subproblem when the step is not successful.

Introducing line searches, however, may weaken the convergence properties of the algorithm. Therefore we begin by discussing two problematic cases that may occur in practice, and how to cope with them.

When the search direction in a line search algorithm is nearly orthogonal to the steepest descent direction $-g_k$, a very small stepsize will normally be required to obtain an acceptable step. In some cases, rounding errors may cause the line search to fail. In similar circumstances, a trust region algorithm will reduce the trust region and the new trial step will tend to the steepest direction. This property makes the method more robust with respect to noise and rounding errors (Carter (1991)), and should be preserved.

The second difficult case is when the search direction in a line search algorithm, or the trial step in a trust region method, are excessively large, which may be caused by a very small matrix B_k. In this case, reducing the trust region will give trial steps that are nearly in the direction of the first failed trial step. The trust region method will, in this case, behave similarly to a backtracking line search method – except that its computational cost will be much higher. In this case it would be advantageous to perform a backtracking line search.

We conclude that backtracking should be performed provided the direction of search is sufficiently downhill. In this paper we show that this can safely be done either along a straight line, or along a curved path, since we find a way of solving (1.2)-(1.3) so that the trial step d_k is always a direction of *sufficient descent* for the objective function. By this we mean that the angle between d_k and $-g_k$ will be bounded away from $\pi/2$ if g_k is bounded away from zero, and if $\|d_k\|$ and $\|B_k\|$ are bounded above. This property will allow us to establish very satisfactory convergence results. Toint (1982) has also incorporated line searches in a trust region method, but in his algorithm line searches are carried out at every iteration. In our algorithm, a backtracking line search is performed only when the trial point $x_k + d_k$ fails to give a lower objective function value.

The theory and implementation of trust region methods has received much attention (see for example Fletcher (1987); Gay (1981); Moré (1983); Moré and Sorensen (1983); Powell (1975); Sorensen (1982a, 1982b); Powell (1984) and Eisenstat and Walker (1991)). The analysis of this paper is based on this work.

Notation. Throughout the paper $\|\cdot\|$ denotes the Euclidean vector norm or its induced matrix norm. The generalized inverse of a matrix A is denoted by A^+, and the angle between two vectors v_1 and v_2 is denoted by $< v_1,\ v_2 >$. The eigenvalues of a symmetric matrix A are denoted by $\sigma_1(A) \geq \ldots \geq \sigma_n(A)$. We indicate that a matrix is positive semi-definite by $A \geq 0$.

2. The Subproblem

In this section, we give some properties of the subproblem (1.2)-(1.3), and consider a few techniques for computing an approximate solution of it. We first recall the following well known result (see for example Moré and Sorensen (1983) and Gay (1981)).

Lemma 2.1 *A vector $d^* \in \Re^n$ is a solution of the problem*

$$\min_{d \in \Re^n} g^T d + \frac{1}{2} d^T B d \equiv \phi(d) \tag{2.1}$$

$$\textit{subject to } ||d|| \leq \Delta, \tag{2.2}$$

where $g \in \Re^n$, $B \in \Re^{n \times n}$ is a symmetric matrix, and $\Delta > 0$, if and only if $||d^*|| \leq \Delta$ and there exists $\lambda^* \geq 0$ such that

$$(B + \lambda^* I) d^* = -g \tag{2.3}$$

$$(B + \lambda^* I) \geq 0 \tag{2.4}$$

and

$$\lambda^* (\Delta - ||d^*||) = 0. \tag{2.5}$$

To express the solution of the trust region problem in closed form it is convenient to make use of some properties of generalized inverses. Suppose that A is a symmetric and semi-definite $n \times n$ matrix with spectral decomposition

$$A = Q \Lambda Q^T, \tag{2.6}$$

where $\Lambda =$ diag $(\sigma_1, \ldots, \sigma_r, 0, \ldots 0)$, with $\sigma_1 \geq \ldots \geq \sigma_r > 0$, and $Q = [q_1, \ldots, q_n]$ is orthogonal. We define the generalized inverse of A by

$$A^+ = Q \Lambda^+ Q^T \tag{2.7}$$

where $\Lambda^+ =$ diag $(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_r}, 0, \ldots, 0)$. By writing $A = \Sigma_{i=1}^r \sigma_i q_i q_i^T$ it is easy to show that if d solves

$$Ad = -g \tag{2.8}$$

for some vector $g \in \Re^n$, then

$$d = -A^+ g + v, \tag{2.9}$$

where v is in the null space of A. It is also clear that $g \in$ span $[q_1, \ldots, q_r]$. Using these facts we obtain

$$g^T A^+ g \geq \|g\|^2/\sigma_1. \tag{2.10}$$

Applying these results to the system (2.3) we see that the solution of the trust region problem is given by

$$d^* = -(B + \lambda^* I)^+ g \; + \; v, \tag{2.11}$$

for some v in the null space of $B + \lambda^* I$; we also have that $g^T v = 0$.

By considering the maximum reduction of the quadratic model ϕ along the steepest descent direction $-g$ one obtains a lower bound for the maximum reduction $\phi(0) - \phi(d)$ within the trust region $||d|| \leq \Delta$.

Lemma 2.2 *(Powell, 1975) If d^* is a solution of (2.1)-(2.2), then*

$$\phi(0) - \phi(d^*) \geq \frac{1}{2}||g|| \min\{\Delta, ||g||/||B||\}. \tag{2.12}$$

The global convergence theory of trust region algorithms only requires that the computed trial step d_k satisfies

$$\phi_k(0) - \phi_k(d_k) \geq \beta||g_k|| \min\{\Delta_k, ||g_k||/||B_k||\} \tag{2.13}$$

for all k, where β is a positive constant. Inequality (2.13) is clearly satisfied if d_k is the exact solution of (1.2)-(1.3). Some other choices of the trail step d_k, such as the dogleg step of Powell (1970), and the minimum of $\phi(d)$ over a two-dimensional subspace within the trust region (Dennis and Mei (1979); Shultz, Schnabel and Byrd (1985)), also satisfy (2.13). One of the main requirements on our algorithm will be that it satisfy (2.13).

Since our algorithm will perform a backtracking line search when the trial step d_k increases the objective function, we shall require that d_k be sufficiently downhill. Therefore we now study the descent properties of search directions generated by trust region methods. We first consider the case when the trust region constraint is active.

Lemma 2.3 *If d^* is a solution of (2.1)-(2.2), if $||d^*|| = \Delta$, and if $\lambda^* \geq 0$ satisfies (2.3)-(2.4), then*

$$0 \leq \lambda^* \leq ||g||/\Delta - \sigma_n(B) \tag{2.14}$$

and

$$d^{*T}g \le -\frac{||g||^2}{\sigma_1(B) - \sigma_n(B) + ||g||/\Delta}, \tag{2.15}$$

where $\sigma_1(B)$ and $\sigma_n(B)$ are the largest and smallest eigenvalues of B, respectively.

Proof. From (2.3) we have that

$$\sigma_n(B + \lambda^* I)||d^*|| \le ||g|| \tag{2.16}$$

which implies that

$$\sigma_n(B) + \lambda^* \le ||g||/\Delta \tag{2.17}$$

because $||d^*|| = \Delta$ and $\sigma_n(B + \lambda^* I) = \sigma_n(B) + \lambda^*$. This inequality and the fact that λ^* is non-negative give (2.14).

Equation (2.11) and inequalities (2.10) and (2.17) yield

$$\begin{aligned} d^{*T}g &= -g^T(B + \lambda I)^+ g \\ &\le -||g||^2/\sigma_1(B + \lambda^* I) \\ &= -||g||^2/(\sigma_1(B) + \lambda^*) \\ &\le -||g||^2/(\sigma_1(B) - \sigma_n(B) + ||g||/\Delta). \quad \textbf{QED} \end{aligned} \tag{2.18}$$

In the case when $||d^*|| < \Delta$, we have from (2.5) that $\lambda^* = 0$, and the analysis is simple. Equations (2.11) and (2.18) hold and from (2.10) we obtain

$$\begin{aligned} d^{*T}g &= -g^T B^+ g \\ &\le -||g||^2/\sigma_1(B). \end{aligned} \tag{2.19}$$

Combining (2.15) and (2.19), we obtain the following result.

Lemma 2.4 *If d^* is a solution of (2.1)-(2.2), then*

$$d^{*T}g \le -\frac{1}{2}||g|| \min\{\Delta,\ ||g||/(2||B||)\}. \tag{2.20}$$

Proof. If $||d^*|| < \Delta$, we can see that inequality (2.20) follows from relation (2.19) and the equality $\sigma_1(B) = ||B||$.

If $||d^*|| = \Delta$, by (2.15) and the fact that $\sigma_1(B) - \sigma_n(B) \leq 2||B||$ it follows that

$$\begin{array}{rcl} d^{*T}g & \leq & -||g||^2/(2||B|| + ||g||/\Delta) \\ & \leq & -||g||^2/2\max\{2||B||,\ ||g||/\Delta\} \\ & = & -\frac{1}{2}||g||\min\{\Delta,\ ||g||/(2||B||)\}. \quad \textbf{QED} \end{array}$$

This lemma shows that the optimal solution d^* satisfies

$$\begin{array}{rcl} \cos(< d^*,\ -g >) & = & -d^{*T}g/||d^*||\,||g|| \\ & \geq & \frac{1}{2}\min\{\Delta/||d^*||,\ ||g||/(2||B||\,||d^*||)\} \\ & \geq & \frac{1}{2}\min\{1,\ ||g||/(2||B||\,||d^*||)\}. \end{array} \tag{2.21}$$

We now show that the angle between d^* and $-g$ is a monotonic function of Δ. To establish this result we assume that in the *hard case* we always choose a solution at the boundary of the trust region. This will be spelled out in detail in the proof of the following result.

Lemma 2.5 *Assume that $g \neq 0$, let $\Delta_2 > \Delta_1 > 0$, and define d_{Δ_1} and d_{Δ_2} to be the solutions of (2.1)-(2.2) when $\Delta = \Delta_1$ and $\Delta = \Delta_2$, respectively. Then*

$$\cos(< d_{\Delta_1},\ -g >) \geq \cos(< d_{\Delta_2},\ -g >). \tag{2.22}$$

Proof. From Lemma 2.1, we know that there exist $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ such that

$$(B + \lambda_1 I)d_{\Delta_1} = -g, \qquad \lambda_1(\Delta_1 - ||d_{\Delta_1}||) = 0 \tag{2.23}$$

$$(B + \lambda_2 I)d_{\Delta_2} = -g, \qquad \lambda_2(\Delta_2 - ||d_{\Delta_2}||) = 0. \tag{2.24}$$

We first show that $\lambda_1 \geq \lambda_2$. By contradiction, assume that $\lambda_2 > \lambda_1 \geq 0$. A direct consequence of the condition $\lambda_2 > 0$ is $||d_{\Delta_2}|| = \Delta_2$. Using this and (2.23)-(2.24) we have

$$\begin{array}{rcl} 0 & \leq & (d_{\Delta_1} - d_{\Delta_2})^T(B + \lambda_1 I)(d_{\Delta_1} - d_{\Delta_2}) \\ & = & (d_{\Delta_1} - d_{\Delta_2})^T[-(B + \lambda_1 I)d_{\Delta_2} - g] \\ & = & (d_{\Delta_1} - d_{\Delta_2})^T(\lambda_2 I - \lambda_1 I)d_{\Delta_2} \\ & = & (\lambda_2 - \lambda_1)[d_{\Delta_1}^T d_{\Delta_2} - ||d_{\Delta_2}||^2 \\ & \leq & (\lambda_2 - \lambda_1)[\Delta_1\Delta_2 - \Delta_2^2] < 0, \end{array} \tag{2.25}$$

which is a contradiction. Therefore $\lambda_1 \geq \lambda_2$.

For the rest of the proof we consider three cases.

I) If $(B + \lambda_2 I)$ is positive definite, the lemma is true because one can show that the function

$$\theta(\lambda) = g^T(B + \lambda I)^{-1}g/||(B + \lambda I)^{-1}g|| \, ||g||, \tag{2.26}$$

which gives the cosine of the angle between $d = -(B + \lambda I)^{-1}g$ and $-g$, is monotonically increasing for all $\lambda \geq \max\{0, -\sigma_n(B)\}$.

II) If $B + \lambda_1 I$ is singular then $\lambda_1 = -\sigma_n(B)$, and we must have from the conditions $\lambda_1 \geq \lambda_2$ and $(B + \lambda_2 I) \geq 0$ that $\lambda_1 = \lambda_2 = -\sigma_n(B)$ and $g \in \mathit{Range}(B - \sigma_n(B)I)$. In this case, which is frequently called the *hard case*, there may be many solutions to both trust region problems, and as suggested by Moré and Sorensen (1983) we will choose a solution at the boundary of the trust region. Therefore we have that

$$\begin{aligned} \cos(< d_{\Delta_1},\ -g >) &= g^T(B - \sigma_n(B)I)^+ g/\Delta_1 ||g|| \\ &\geq g^T(B - \sigma_n(B)I)^+ g/\Delta_2 ||g|| \\ &= \cos(< d_{\Delta_2},\ -g >). \end{aligned} \tag{2.27}$$

III) To complete the proof, we assume that $B + \lambda_1 I$ is positive definite and $\lambda_2 = -\sigma_n(B)$. Again, we see that $g \in \mathit{Range}(B - \sigma_n(B)I)$, and by (2.11) we have that $||d_{\Delta_2}|| \geq ||(B - \sigma_n(B)I)^+ g||$. Using this, and the fact that $\theta(\lambda)$ is monotonically increasing we obtain

$$\begin{aligned} \cos(< d_{\Delta_1},\ -g >) &= g^T(B + \lambda_1 I)^{-1}g/||(B + \lambda_1 I)^{-1}g|| \, ||g|| \\ &\geq \lim_{\lambda \to -\sigma_n(B)} g^T(B + \lambda I)^{-1}g/||(B + \lambda I)^{-1}g|| \, ||g|| \\ &= g^T(B - \sigma_n(B)I)^+ g/||(B - \sigma_n(B)I)^+ g|| \, ||g|| \\ &\geq g^T(B - \sigma_n(B)I)^+ g/||d_{\Delta_2}|| \, ||g|| \ . \end{aligned} \tag{2.28}$$

This shows that the lemma is true. **QED**

All of these results concern the exact solution of the trust region problem. We now consider an approximate solution $d(\lambda)$ of (2.1)-(2.2), defined by

$$d(\lambda) = -(B + \lambda I)^{-1}g, \tag{2.29}$$

where $\lambda \geq \max\{0, -\sigma_n(B)\}$ is a parameter such that $B + \lambda I$ is positive definite. Let us assume for the moment that $d(\lambda)$ satisfies the inequality

$$\Delta/\gamma \leq ||d(\lambda)|| \leq \Delta, \tag{2.30}$$

for some constant $\gamma > 1$. Lemma 2.1 shows that $d(\lambda)$ is the solution of (2.1) subject to $||d|| \leq ||d(\lambda)||$. Consequently (2.12) and (2.20) give

$$\phi(0) - \phi(d(\lambda)) \geq \frac{1}{2}||g|| \min\{\Delta/\gamma,\ ||g||/||B||\} \tag{2.31}$$

and

$$d(\lambda)^T g \leq -\frac{1}{2}||g|| \min\{\Delta/\gamma,\ ||g||/2||B||\}. \tag{2.32}$$

These two properties – the reduction (2.31) in the model, and the sufficient descent condition (2.32) – will ensure that the algorithm possesses good convergence properties. However imposing the lower bound in (2.30) on the size of the step may not always be appropriate; for example a Newton step could be shorter than Δ/γ. Therefore we now derive an alternate condition that will ensure that (2.31) and (2.32) are satisfied, without imposing a lower bound on the size of the step.

Inequality (2.14) suggests the following upper bound for λ

$$\lambda \leq ||B|| + (1+\epsilon)||g||/\Delta, \tag{2.33}$$

where $\epsilon > 0$ is a small number that ensures that $B + \lambda I$ is positive definite. Consider any $\lambda \geq \max\{0, -\sigma_n(B)\}$ for which $B + \lambda I$ is positive definite, and for which (2.33) and

$$||d(\lambda)|| \leq \Delta \tag{2.34}$$

hold. For any such λ we have from (2.29) that

$$\begin{aligned} d(\lambda)^T g &= -g^T(B+\lambda I)^{-1} g \\ &\leq -||g||^2/(||B|| + \lambda) \\ &\leq -||g||^2/(2||B|| + (1+\epsilon)||g||/\Delta) \\ &\leq -\frac{1}{2}||g|| \min\{\Delta/2,\ ||g||/(2||B||)\}. \end{aligned} \tag{2.35}$$

Using (2.29) and (2.35) we obtain

$$\begin{aligned} \phi(0) - \phi(d(\lambda)) &= -g^T d(\lambda) - \frac{1}{2} d(\lambda)^T (B + \lambda I) d(\lambda) + \frac{1}{2}\lambda ||d(\lambda)||^2 \\ &= -\frac{1}{2} g^T d(\lambda) + \lambda ||d(\lambda)||^2 \\ &\geq \frac{1}{4}||g|| \min\{\Delta/2,\ ||g||/(2||B||)\}. \end{aligned} \tag{2.36}$$

We conclude from (2.31)-(2.32) and (2.35)-(2.36) that if either (2.30) or (2.33) hold, then a feasible trial step $d(\lambda)$ satisfies

$$\phi(0) - \phi(d(\lambda)) \geq \tau ||g|| \min\{\Delta,\ ||g||/||B||\} \tag{2.37}$$

and

$$d(\lambda)^T g \leq -\tau ||g|| \min\{\Delta,\ ||g||/||B||\}, \tag{2.38}$$

where $\tau = \min\{1/8, 1/2\gamma\}$. The following algorithm computes an approximate solution of the trust region problem (2.1)-(2.2) that satisfies either (2.30) or (2.33).

Algorithm 2.6

Step 1. Given constants $\gamma > 1$ *and* $\epsilon > 0$, *set* $\lambda := 0$.
If B *is positive definite go to Step 2; else find* $\lambda \in [0, ||B|| + (1+\epsilon)||g||/\Delta]$ *such that* $B + \lambda I$ *is positive definite.*

Step 2. Factorize $B + \lambda I = R^T R$, *where* R *is upper triangular, and solve* $R^T R d = -g$ *for* d.

Step 3. If $||d|| \leq \Delta$ *stop; else solve* $R^T q = d$ *for* q, *and compute*

$$\lambda := \lambda + \frac{||d||^2}{||q||^2} \frac{\gamma ||d|| - \Delta}{\Delta}; \tag{2.39}$$

go to Step 2.

Our update formula for λ is based on the Newton step for the nonlinear equation

$$\psi(\lambda) = \frac{1}{||d(\lambda)||} - \frac{\gamma}{\Delta} = 0 \tag{2.40}$$

instead of the equation

$$\bar{\psi}(\lambda) = \frac{1}{||d(\lambda)||} - \frac{1}{\Delta} = 0 \tag{2.41}$$

considered by Moré and Sorensen (1983), because we allow $d(\lambda)$ to lie in the interval $[\Delta/\gamma,\ \Delta]$. Due to the concavity of the function $\psi(\lambda)$ (cf. Dennis and Schnabel (1983)), it is easy to see that $d(\lambda)$ calculated by Algorithm 2.6

satisfies either (2.30) or (2.33). Thus the trial step $d(\lambda)$ computed by Algorithm 2.6 always satisfies inequalities (2.37) and (2.38). In the next section we describe a trust region method with backtracking line search based on these ideas.

3. The Algorithm and Convergence

If the trial step d_k is computed at each iteration by Algorithm 2.6, we know that there exists a positive constant τ such that

$$\phi_k(0) - \phi_k(d_k) \geq \tau ||g_k|| \min\{\Delta_k,\ ||g_k||/||B_k||\} \tag{3.1}$$

and

$$d_k^T g_k \leq -\tau ||g_k|| \min\{\Delta_k,\ ||g_k||/||B_k||\}. \tag{3.2}$$

Thus d_k is a direction of *sufficient descent* in the sense that the angle between d_k and $-g_k$ will be bounded away from $\pi/2$ if $||g_k||$ is bounded away from zero and $||B_k||$ is bounded above. Hence, if the trial step d_k is not acceptable, that is if $f(x_k + d_k) \geq f(x_k)$, we can safely perform a backtracking line search along d_k: we find the minimum positive integer i such that

$$f(x_k + \alpha^i d_k) < f(x_k), \tag{3.3}$$

where $\alpha \in (0,1)$ is a positive constant. Backtracking need not be along the line segment joining x_k and $x_k + d_k$. We can let the trial points be any sequence $d_k^{(i)}$ satisfying

$$||d_k^{(i+1)}|| \in [\alpha_1,\ \alpha_2]||d_k^{(i)}||, \qquad 0 < \alpha_1 < \alpha_2 < 1, \tag{3.4}$$

and

$$\cos(< d_k^{(i)},\ -g_k) \geq \cos(< d_k,\ -g_k >) \tag{3.5}$$

for $i = 0, 1, 2, ...$, with $d_k^{(0)} = d_k$. It is clear that the choice $d_k^{(i)} = \alpha^i d_k$ (backtracking along the direction d_k), as well as a variety of curved line searches, satisfy (3.4)-(3.5). It also follows from Lemma 2.5 that (3.4)-(3.5) are satisfied if $d_k^{(i)}$ is chosen to be an exact solution of the trust region problem (1.2)-(1.3) with Δ_k replaced by $\Delta_k^{(i)}$, where $\Delta_k^{(0)} = ||d_k||$ and $\Delta_k^{(i+1)} \in [\alpha_1,\ \alpha_2]\Delta_k^{(i)}$.

We can now describe the trust region algorithm with a backtracking line search.

Algorithm 3.1

Step 1. *Given* $x_1 \in \Re^n$ *and* $\Delta_1 > 0$*, choose constants* c_1, c_2, c_3 *and* c_4 *such that* $0 < c_3 < c_4 < 1 < c_1,\ 0 < c_2 < 1$*; set* $k := 1$.

Step 2. *Solve (1.2)-(1.3) inaccurately so that* $||d_k|| \le \Delta_k$*, and so that (3.1) and (3.2) are satisfied. (This can be done by means of Algorithm 2.6.)*

Step 3. *Compute* $f(x_k + d_k)$*. If* $f(x_k + d_k) < f(x_k)$ *go to Step 4; else find the minimum positive integer* i_k *such that*

$$f(x_k + d_k^{(i_k)}) < f(x_k) \tag{3.6}$$

where $d_k^{(i)}$ *is chosen so that (3.4)-(3.5) are satisfied; compute*

$$x_{k+1} = x_k + d_k^{(i_k)} \tag{3.7}$$

$$\Delta_{k+1} \in \{||x_{k+1} - x_k||, c_4\Delta_k\}; \tag{3.8}$$

go to Step 5.

Step 4. *Compute*

$$x_{k+1} = x_k + d_k, \tag{3.9}$$

and

$$\rho_k = \frac{f(x_k) - f(x_{k+1})}{\phi_k(0) - \phi_k(d_k)}. \tag{3.10}$$

If $\rho_k \ge c_2$ *and* $\|d_k\| < \Delta_k$*, set* $\Delta_{k+1} = \Delta_k$*, otherwise define*

$$\Delta_{k+1} \in \begin{cases} [c_3||d_k||,\ c_4\Delta_k] & \textit{if } \rho_k < c_2 \\ [\Delta_k,\ c_1\Delta_k] & \textit{if } \rho_k \ge c_2 \textit{ and } \|d_k\| = \Delta_k\ . \end{cases} \tag{3.11}$$

Step 5. *Compute* $g(x_{k+1})$ *and* B_{k+1}*; set* $k := k + 1$*; go to Step 2.*

We will not specify how the matrices B_k are computed; they could be defined to be exact Hessians or quasi-Newton approximations. We note for future reference that the algorithm decreases the trust region radius if backtracking takes place (see (3.8)) or if the ratio ρ_k of actual to predicted reduction in the function is not large enough (see (3.11)).

To analyze the new algorithm we make the following assumptions.

Assumptions 3.2
1) The sequence $\{x_k\}$ generated by Algorithm 3.1 is bounded, that is

$$x_k \in S \tag{3.12}$$

for all k, S being a closed convex set in $\Re^n$.
2) f is twice differentiable in S and there exists a constant M such that

$$||\nabla^2 f(x)|| \leq M \qquad \textit{for all } x \in S. \tag{3.13}$$

Under the above assumptions, we will prove that the sequence $\{x_k\}$ generated by Algorithm 3.1 is globally convergent in the sense that

$$\liminf_{k \to \infty} ||g_k|| = 0. \tag{3.14}$$

We proceed by contradiction. If (3.14) were not true, there would exist a constant $0 < \delta < 1$ such that

$$||g_k|| \geq \delta > 0 \tag{3.15}$$

for all k. This, together with (3.1) and (3.2), shows that there exists a positive constant μ such that

$$\phi_k(0) - \phi_k(d_k) \geq \mu \min\{\Delta_k,\ 1/||B_k||\} \tag{3.16}$$

and

$$d_k^T g_k \leq -\mu \min\{\Delta_k,\ 1/||B_k||\} \tag{3.17}$$

for all k.

We define I to be the set of integers k such that $\rho_k \geq c_2$ (see Step 4). Since Algorithm 3.1 ensures that $\{f(x_k)\}$ is monotonically decreasing we have

$$\begin{aligned} \sum_{k=1}^{\infty}[f(x_k) - f(x_{k+1})] &\geq \sum_{k \in I}[f(x_k) - f(x_{k+1})] \\ &\geq c_2 \sum_{k \in I}[\phi_k(0) - \phi_k(d_k)] \\ &\geq c_2 \mu \sum_{k \in I} \min\{\Delta_k, 1/||B_k||\}. \end{aligned} \tag{3.18}$$

Define the sequence

$$M_k = 1 + \max_{1 \leq i \leq k} ||B_i||. \tag{3.19}$$

Since Assumptions 3.2 imply that $\{f(x_k)\}$ is bounded below, we have from (3.18) and (3.19), that

$$\sum_{k\in I} \min\{\Delta_k,\ 1/M_k\} < \infty, \tag{3.20}$$

so that

$$\lim_{\substack{k\to\infty \\ k\in I}} \min\{\Delta_k, \frac{1}{M_k}\} = 0. \tag{3.21}$$

We now show that this limit holds also for $k \notin I$. If I is finite, we have from (3.8) and (3.11) that $\Delta_{k+1} \leq c_4\Delta_k$ for all large k, so that $\{\Delta_k\} \to 0$, which gives

$$\lim_{k\to\infty} \min\{\Delta_k, 1/M_k\} = 0. \tag{3.22}$$

If I is infinite we consider an index $j \notin I$, and let $\hat{j}$ be the greatest integer less than j such that $\hat{j} \in I$. Since $\{M_k\}$ is monotonically increasing, and since $\Delta_j \leq c_1\Delta_{\hat{j}}$ we see that

$$\min\{\Delta_j, \frac{1}{M_j}\} \leq c_1 \min\{\Delta_{\hat{j}}, \frac{1}{M_{\hat{j}}}\}, \tag{3.23}$$

which due to (3.21) implies (3.22).

Note that the limit (3.22) was established under the assumption that $||g_k||$ is bounded away from zero. It states that if the algorithm fails then either $\{\Delta_k\}$ converges to zero or $\|B_k\|$ is unbounded. We will use this limit to arrive to a contradiction. The next two lemmas consider the two cases under which the trust region is reduced by Algorithm 3.1, namely when $\rho_k < c_2$ (Step 4), and when x_{k+1} is defined as $x_k + d_k^{i_k}$ in the backtracking Step 3 of the algorithm.

Lemma 3.3 *If inequality (3.15) and Assumptions 3.2 are satisfied, then*

$$\Delta_k \geq ||d_k|| \geq \min\{\mu(1-c_2),\ 1\}/M_k \tag{3.24}$$

for all sufficiently large k for which x_{k+1} is defined by Step 4 of Algorithm 3.1, and for which $\rho_k < c_2$.

Proof. The algorithm ensures that $||d_k|| \leq \Delta_k$. If the second inequality is false, there exist infinitely many k_i $(i = 1, 2, ...)$ such that

$$\rho_{k_i} < c_2 \tag{3.25}$$

and

$$||d_{k_i}|| < \min\{\mu(1-c_2),\ 1\}/M_{k_i}\ . \tag{3.26}$$

The first inequality in (3.24), (3.22) and (3.26) show that

$$\lim_{i\to\infty} ||d_{k_i}|| = 0. \tag{3.27}$$

Using the mean value theorem, (3.13), (3.19) and (3.26) we obtain

$$\begin{aligned} \rho_{k_i} &= 1 + \frac{1}{2}\frac{(d_{k_i}^T B_{k_i} d_{k_i} - d_{k_i}^T \nabla^2 f(\xi_{k_i}) d_{k_i})}{\phi_{k_i}(0) - \phi_{k_i}(d_{k_i})} \\ &> 1 - \frac{1}{2}\frac{||d_{k_i}||\mu(1-c_2) + M||d_{k_i}||^2}{\phi_{k_i}(0) - \phi_{k_i}(d_{k_i})}. \end{aligned} \tag{3.28}$$

The limit (3.27) implies that for sufficiently large i

$$M||d_{k_i}||^2 \leq \frac{1}{2}||d_{k_i}||\,\mu(1-c_2). \tag{3.29}$$

Moreover, (3.26) implies that $||d_{k_i}|| < 1/M_{k_i}$, and since we also know that $||d_{k_i}|| \leq \Delta_k$, we see that

$$||d_{k_i}||/\min\{\Delta_k, 1/M_{k_i}\} \leq 1. \tag{3.30}$$

Using (3.16), (3.29) and (3.30) in (3.28) we have for sufficiently large i

$$\begin{aligned} \rho_{k_i} &> 1 - \frac{3}{4}\frac{||d_{k_i}||\,\mu(1-c_2)}{\mu\min\{\Delta_{k_i}, 1/M_{k_i}\}} \\ &\geq 1 - \frac{3}{4}(1-c_2) > c_2, \end{aligned} \tag{3.31}$$

since $c_2 < 1$. This contradicts (3.25). **QED**

We now show that a similar result holds if backtracking takes place.

Lemma 3.4 *If inequality (3.15) and Assumptions 3.2 are satisfied, there is a constant $0 < \bar{\mu} \leq 1$ such that*

$$\Delta_k > ||x_{k+1} - x_k|| \geq \bar{\mu}\min\{\mu(1-c_2),\ 1\}/M_k, \tag{3.32}$$

for all sufficiently large k for which x_{k+1} is given by $x_k + d_k^{i_k}$ in Step 3 of Algorithm 3.1.

Proof. The first inequality follows from (3.4). Assume that the second inequality does not hold. Then there exist infinitely many k such that $x_{k+1} = x_k + d_k^{(i_k)}$ and

$$||x_{k+1} - x_k|| = o(1/M_k). \tag{3.33}$$

Hence we have from (3.13), (3.4), (3.5) and (3.17) that

$$\begin{aligned} f(x_k) - f(x_k + d_k^{(i_k-1)}) &\geq ||d_k^{(i_k-1)}||\,||g_k|| \cos(< d_k^{(i_k-1)},\ -g_k >) \\ &\quad -M||d_k^{(i_k-1)}||^2/2 \\ &> ||x_{k+1} - x_k||\,||g_k|| \cos(< d_k,\ -g_k >) \\ &\quad -M||x_{k+1} - x_k||^2/(2\alpha_1^2) \\ &\geq ||x_{k+1} - x_k||\, \mu \min\{\Delta_k/||d_k||, 1/||B_k||\,||d_k||\} \\ &\quad -M||x_{k+1} - x_k||^2/(2\alpha_1^2) \\ &\geq \|x_{k+1} - x_k\| \mu \min\{1, 1/(M_k\Delta_k)\} \\ &\quad -M||x_{k+1} - x_k||^2\,/(2\alpha_1^2) \\ &> (1/M_k)||x_{k+1} - x_k||\,[\mu \min\{M_k, 1/(\Delta_k)\} \\ &\quad -M||x_{k+1} - x_k||M_k/(2\alpha_1^2)\Big]. \end{aligned} \tag{3.34}$$

The sequence $\{\Delta_k\}$ is bounded above since we have assumed that the iterates remain in a bounded set S, and since the algorithm only increases the trust region if the step is at the boundary of the trust region. Using this fact and (3.33) we see that the right hand side of (3.34) is positive for large k, which contradicts our definition of i_k. **QED**

Using these two lemmas we can establish a lower bound for Δ_k, for all iterates. We recall that c_3 is the constant used in Step 4 of Algorithm 3.1 to reduce the size of the trust region.

Lemma 3.5 *If inequality (3.15) and Assumptions 3.2 hold, then*

$$\Delta_k \geq \bar{\mu} c_3 \min\{\mu(1 - c_2),\ 1\}/M_k \tag{3.35}$$

for all sufficiently large k.

Proof. If there are only finitely many k such that $\Delta_{k+1} < \Delta_k$, there exists a positive constant $\bar{\delta}$ such that $\Delta_k > \bar{\delta}$ for all k. Due to (3.22), we have $\lim_{k\to\infty} 1/M_k = 0$, and hence (3.35) holds for all large k.

Now assume that there are infinitely many k such that $\Delta_{k+1} < \Delta_k$. From Lemma 3.3 and Lemma 3.4, there exists an integer $\bar{k}$ such that if $k \geq \bar{k}$ and $\Delta_{k+1} < \Delta_k$ then

$$\Delta_k \geq \bar{\mu} \min\{\mu(1-c_2),\ 1\}/M_k. \tag{3.36}$$

Let $\hat{k}$ be an integer such that $\hat{k} \geq \bar{k}$ and $\Delta_{\hat{k}+1} < \Delta_{\hat{k}}$. We now show that (3.35) holds for any $k \geq \hat{k}$. If $\Delta_{k+1} < \Delta_k$, then (3.36) holds, which implies (3.35) since $c_3 < 1$. If $\Delta_{k+1} \geq \Delta_k$, let $k\prime$ be the largest integer less than k such that $\Delta_{k\prime+1} < \Delta_{k\prime}$. We see from (3.8), (3.11), (3.24) and (3.32) that

$$\begin{aligned}\Delta_k &\geq \Delta_{k\prime+1} \geq c_3||x_{k\prime+1} - x_{k\prime}|| \\ &\geq c_3\bar{\mu}\min\{\mu(1-c_2),\ 1\}/M_{k\prime} \\ &\geq c_3\bar{\mu}\min\{\mu(1-c_2),\ 1\}/M_k, \qquad (3.37)\end{aligned}$$

which is (3.35). **QED**

We now state a lemma given by Powell (1984).

Lemma 3.6 *Let* $\{\Delta_k\}$ *and* $\{M_k\}$ *be two sequences such that* $\Delta_k \geq \nu/M_k \geq 0$ *for all* k*, where* ν *is a positive constant. Let* $\mathcal{I}$ *be a subset of* $\{1,2,3,...\}$*. Assume that*

$$\begin{aligned}\Delta_{k+1} &\leq c_1\Delta_k, && i \in \mathcal{I} && (3.38)\\ \Delta_{k+1} &\leq c_4\Delta_k, && i \notin \mathcal{I} && (3.39)\\ M_{k+1} &\geq M_k, && \textit{for all}\ \ k && (3.40)\\ \sum_{k\in I} 1/M_k &< \infty && && (3.41)\end{aligned}$$

where $c_1 > 1$, $c_4 < 1$ *are positive constants. Then*

$$\sum_{k=1}^{\infty} 1/M_k < \infty. \tag{3.42}$$

We can now establish a global convergence result. We only need to assume that $||B_k||$ does not grow too rapidly.

Theorem 3.7 *If Assumptions 3.2 are satisfied, and if* B_k *satisfies*

$$\sum_{k=1}^{\infty} 1/M_k = \infty, \tag{3.43}$$

where M_k is defined by (3.19), then the sequence $\{x_k\}$ generated by Algorithm 3.1 is not bounded away from stationary points of f, that is

$$\liminf_{k\to\infty} ||g_k|| = 0. \tag{3.44}$$

Proof. If (3.44) is not true, there exists a positive constant δ such that (3.15) holds for all k. In this case Lemma 3.5 shows that there exists a constant $\nu > 0$ such that $\Delta_k \geq \nu / M_k$ for all sufficiently large k. Let $\mathcal{I}$ in Lemma 3.6 be the set of iterations for which $\rho_k \geq c_2$ (see (3.11)). Then (3.38)-(3.40) hold from (3.8)-(3.11) and from the definition (3.19). We also know that (3.15) implies (3.35), which together with (3.20) gives (3.41). Thus the conditions of Lemma 3.6 are satisfied and (3.42) holds, contradicting (3.43).
QED

Based on this theorem, we can derive the following convergence results for Newton's method.

Corollary 3.8 *Suppose that whenever B_k is positive definite and $\|B_k^{-1} g_k\| \leq \Delta_k$, the algorithm chooses the step $d_k = -B_k^{-1} g_k$. Suppose also that Assumptions 3.2 are satisfied and that*

$$B_k = \nabla^2 f(x_k) \tag{3.45}$$

for all k. Then

1. *The sequence $\{x_k\}$ generated by Algorithm 3.1 satisfies (3.44).*
2. *If f is convex it follows that*

$$\lim_{k\to\infty} ||g_k|| = 0. \tag{3.46}$$

3. *If $\{x_k\}$ converges to a point x_* then $\nabla^2 f(x_*)$ is positive semi-definite.*
4. *If $\{x_k\}$ converges to a point x_* such that $\nabla^2 f(x_*)$ is positive definite, the rate of convergence is quadratic, i.e.*

$$||x_{k+1} - x_*|| = O(||x_k - x_*||_2^2) \quad . \tag{3.47}$$

Proof. Since B_k is given by (3.45) for all k, (3.43) is satisfied due to (3.13). Hence it follows from Theorem 3.7 that (3.44) is true. If $f(x)$ is convex, it can be easily shown that (3.44) implies (3.46) from the fact that $\{f(x_k)\}$

is monotonically decreasing and that all stationary points of f are global minimizers.

The rest of the proof is similar to those of Theorems 5.1.1 and 5.1.2 of Fletcher (1987). **QED**

We have thus been able to establish convergence results for the new trust region algorithm with line searches that are as strong as the results obtained previously for pure trust region methods. Note that we did not need to assume that the matrices B_k are bounded, but only that their norms increase at most linearly with k. In contrast, to establish the global convergence of line search methods, one normally needs to assume that the condition number of B_k does not increase too rapidly - or one has to study the trace and determinant of B_k (Powell (1976), Byrd, Nocedal and Yuan (1987)), which is technically difficult.

4. Numerical Results

We have implemented the new algorithm and compared it both with a line search algorithm and with a pure trust region algorithm. In the line search algorithm, inexact line searches are carried out so that

$$f(x_{k+1}) \leq f(x_k) + 0.01(x_{k+1} - x_k)^T g_k \tag{4.1}$$

and

$$|(x_{k+1} - x_k)^T g_{k+1}| \leq -0.9(x_{k+1} - x_k)^T g_k \tag{4.2}$$

for all k, and the BFGS formula is used to update B_k.

In the pure trust region algorithm, the trial step is computed by Algorithm 2.6 with $\gamma = 1.5$, and B_k is updated by the BFGS formula. However we do not update B_k if the curvature condition

$$s_k^T y_k > 0, \tag{4.3}$$

does not hold, where

$$s_k = x_{k+1} - x_k \tag{4.4}$$

$$y_k = g_{k+1} - g_k. \tag{4.5}$$

For the new trust region algorithm with backtracking we implemented two versions, one using the BFGS formula to update B_k, and skipping the update as in the pure trust region method, and the other using the SR1 formula to update B_k. To prevent overflow we apply the SR1 update only if

$$|s_k^T y_k| \geq 10^{-6} ||s_k||\,||y_k||. \tag{4.6}$$

Backtracking is performed along the direction of the failed trial step, and is based on truncated quadratic interpolation. If $f(x_k + d_k) \geq f(x_k)$ we compute

$$\alpha_k = \max\{0.1, 0.5/[1 + (f(x_k) - f(x_k + d_k))/d_k^T g_k], \} \tag{4.7}$$

set $d_k := \alpha_k d_k$, and repeat this process until a lower function value is obtained.

When using the SR1 update, B_k may not be positive definite. Hence we calculate the trial step using the subroutine GQTPAR which is designed to handle the indefinite case based on the ideas described by Moré and Sorensen (1983). We chose the initial trust region radius as $\Delta_1 = 1$, for all trust region algorithms, and terminated the iterations when $\|g_k\| \leq 10^{-8}$. In all tests, the initial matrix B_1 was chosen as the identity matrix. The algorithms were coded in FORTRAN, using double precision, and the tests were performed on a Sun Sparcstation 1.

We tested the algorithms on the 18 examples given by Moré, Garbow and Hillstrom (1981). These are small problems, with a number of variables ranging from 2 to 20. The results are given in Table 1. For the BFGS algorithm with line searches (column 1), we list the numbers of iterations, function and gradient evaluations. For the pure trust region algorithm using BFGS updating (column 2), we only give the number of iterations, since this equals the number of function and gradient evaluations (i.e. there is exactly one function and gradient evaluation per iteration). For the new trust region algorithm with backtracking (TR+BT), only the number of iterations and function evaluations are given, as the number of gradient evaluations is the same as the number of iterations.

TABLE I

Results on the 18 problems of Moré, Garbow and Hillstrom

		Line Search BFGS	Pure Trust Region BFGS	TR+BT BFGS	TR+BT SR1
	n	Iter (F-G)	Iter	Iter (F)	Iter (F)
1	3	28(40 - 30)	26	24(26)	28(34)
2	6	36(45 - 40)	43	35(36)	137(178)
3	3	3(5 - 4)	6	5(6)	4(7)
4	2	158(220 - 174)	204	175(212)	>300
5	3	19(32 - 27)	23	30(31)	44(55)
6	6	18(23 - 19)	17	17(17)	17(17)
7	9	68(76 - 69)	68	66(70)	45(61)
8	8	27(43 - 39)	52	70(82)	111(143)
9	3	8(12 - 10)	12	12(13)	23(41)
10	2	10(23 - 15)	37	36(37)	33(38)
11	4	25(43 - 29)	26	24(31)	23(27)
12	3	30(44 - 35)	36	30(34)	>300
13	20	46(51 - 49)	53	46(51)	51(66)
14	14	128(179 - 133)	128	112(138)	135(190)
15	16	102(123 - 105)	92	76(87)	99(132)
16	2	13(17 - 14)	16	16(16)	18(20)
17	4	81(114 - 87)	76	67(79)	114(156)
18	8	21(35 - 23)	71	23(33)	32(50)

We observe that the pure trust region algorithm performed better than the line search algorithm, in terms of function evaluations. It is also clear that BFGS updating outperformed SR1 updating in the new trust region method. When comparing the pure trust region algorithm and the new trust region method with backtracking, both using BFGS updating, (columns 2 and 3) we observe the following: (i) the number of function evaluations is similar – the pure trust region methods requires fewer function evaluations, but the difference is not significant; (ii) the number of iterations required by the new trust region method with backtracking is smaller than that of the pure trust region method in most of the problems.

These results suggest that the trust region algorithm with backtracking may be effective for solving large problems, but we will not discuss this here, since an efficient implementation for large problems requires careful consideration and is the subject of future research. Instead we look more carefully at the relative performance of BFGS and SR1 updating. The results in Table 1 are quite unfavorable towards SR1, and to investigate whether they are typical we tested a few additional medium-size problems. These problems are listed in Liu and Nocedal (1989) – we have used the same numbering system as in that paper. Surprisingly SR1 now clearly outperforms BFGS, and we are unable to conclude that one updating formula is preferable to the other one.

TABLE II

The new algorithm using two update formulae on medium-size problems

		TR + BT SR1	TR+BT BFGS
	n	Iterations (F)	Iterations (F)
1	50	32(37)	81(131)
2	50	51(73)	90(137)
3	50	50(74)	63(81)
7	100	85(124)	> 300
8	121	117(156)	128(193)
11	100	60(89)	95(186)
15	100	133(175)	136(239)
16	100	68(95)	72(140)

5. Final Remarks

We have described an algorithm that does not resolve the trust region subproblem when the trial step increases the objective function. Two other approaches achieve the same goal. The first is the dogleg method, but this is only applicable when the Hessian approximation B_k is positive definite, and can be slow to converge when B_k is very ill-conditioned. The second approach (Schultz et al (1985)) is a restricted subspace method where, after computing an unsuccessful trial step d_k, one reduces the trust region radius and solves a subproblem of the form

$$\min\{\phi(d) : \|d\| \leq \Delta,\ d \in \{g_k, d_k\}\}.$$

It follows from the results in Moré (1983) and Schultz et al (1985) that this restricted subspace method possesses the same theoretical properties as the algorithm proposed here; in particular, it is easy to show that (3.1)-(3.2) hold.

Nevertheless, the backtracking line search approach described in this paper may be more effective than adjusting the trust region radius in those cases when the trial step is very poor. We also believe that Algorithm 2.6 is novel in that it always solves positive definite systems and avoids the so called "hard case". We have shown that it is possible to compute a trust region step that it is sufficiently steep to allow for a safe backtracking line search; this is a topic that has not received much attention in the literature.

Acknowledgments. We would like to thank Jorge Moré for providing us with his subroutine GQTPAR, and Richard Byrd and Nick Gould for several useful conversations on the topic of this paper.

6. *

References

[1] R.H. Byrd, J. Nocedal and Y. Yuan (1987). "Global convergence of a class of quasi-Newton methods on convex problems", SIAM J. Numer. Analysis, 24, pp. 1171–1190.

[2] R.G. Carter (1991). "On the global convergence of trust region algorithms using inexact gradient information", SIAM J. Numerical Analysis, 28,1 pp. 251-265.

[3] J.E. Dennis, Jr. and H.H.W. Mei (1979). "Two new unconstrained optimization algorithms which use function and gradient values", *J. Optim. Theory Appl.* **28**, pp. 453-482.

[4] J.E. Dennis, Jr. and R.B. Schnabel (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* , Prentice-Hall, Inc., (Englewood Cliffs, NJ).

[5] S.C. Eisenstat and H.F. Walker (1991). "Globally convergent inexact Newton methods", Utah State University Mathematics and Statistics Department Research Report, February/91/51, University of Utah, Logan UT.

[6] R. Fletcher (1987). *Practical Methods of Optimization* (John Wiley and Sons, New York).

[7] D.M. Gay (1981). "Computing optimal locally constrained steps", *SIAM J. on Scientific and Statistical Computing* **2**, pp. 186–197.

[8] D.C. Liu and J. Nocedal (1989). "On the limited memory BFGS method for large scale optimization", *Mathematical Programming* **45**, pp. 503–528.

[9] J.J. Moré (1983). "Recent developments in software for trust region methods", in : A. Bachem, M. Grőtschel and B. Korte, eds., *Mathematical Programming, The State of the Art* (Springer-Verlag, Berlin), pp. 258–287.

[10] J.J. Moré, B.S. Garbow and K.E. Hillstrom (1981). "Testing unconstrained optimization software" *ACM Transactions on Mathematical Software* **7**, pp. 17–41.

[11] J.J. Moré and D.C. Sorensen (1983). "Computing a trust region step", *SIAM J. on Scientific and Statistical Computing* **4**, pp. 553–572.

[12] M.J.D. Powell (1970). "A new algorithm for unconstrained optimization", in: J.B. Rosen, O.L. Mangasarian and K. Ritter, eds., *Nonlinear Programming* (Academic Press, New York), pp. 31–66.

[13] M.J.D. Powell (1975). "Convergence properties of a class of minimization algorithms", in: O.L. Mangasarian, R.R. Meyer and S.M. Robinson, eds., *Nonlinear Programming* (Academic Press, New York) **2**, pp. 1–27.

[14] M.J.D. Powell (1976). "Some global convergence properties of a variable metric algorithm for minimization without exact line searches", in: R.W. Cottle and C.E. Lemke, eds., *Nonlinear Programming SIAM-AMS Proceedings* (SIAM publications) **9**, pp. 53–72.

[15] M.J.D. Powell (1984). "On the global convergence of trust region algorithm for unconstrained optimization", *Mathematical Programming* **29**, pp. 297–303.

[16] G.A. Schultz, R.B. Schnabel and R.H. Byrd (1985). "A family of trust-region-based algorithms for unconstrained minimization with strong

global convergence properties", *SIAM Journal on Numerical Analysis* **22** pp.47-67.

[17] D.C. Sorensen (1982a). "Trust region methods for unconstrained optimization", in: M.J.D. Powell, ed. *Nonlinear Optimization 1981* (Academic Press, London), pp. 29–38.

[18] D.C. Sorensen (1982b). "Newton's method with a model trust region modifications", *SIAM J. Numerical Analysis* **19**, pp. 409–426.

[19] Ph.L. Toint (1982). "Towards an efficient sparsity exploiting Newton method for minimization" in: I.S. Duff, ed., *Sparse Matrices and their Uses*, (Academic Press, New York), pp. 57–87.