A New Trust Region Algorithm with Trust Region Radius Converging to Zero

Jin-yan Fan   Ya-xiang Yuan

State Key Laboratory of Scientific/Engineering Computing,
Institute of Computational Mathematics and Scientific/Engineering Computing,
The Academy of Mathematics and Systems Sciences,
Chinese Academy of Sciences, P.O.Box 2719, Beijing, 100080, P.R.China,
E-mail: fjy@lsec.cc.ac.cn,   yyx@lsec.cc.ac.cn

Abstract

In the traditional trust region algorithms for unconstrained optimization problem, as the sequence converges to the minimizer x^* of the problem, the trust region radius will be larger than a positive constant. Thus the trust region doesn't play the role at the end. In fact, it suffices for the convergence that the trust region radius be larger than $O(||x_k - x^*||)$ at the k-th iterate. In this paper, we propose a new trust region algorithm with the trust region radius converging to zero. We show that the new algorithm preserves the global convergence of the traditional trust region algorithms. And the superlinear convergence is also proved under certain conditions. Numerical results are presented to show that the algorithm is efficient for small size problems.

Keywords: trust region algorithm, unconstrained optimization, nonliner programming.

1 Introduction

Many trust region algorithms for the unconstrained optimization problem

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

where $f(x)$ is a continuous differential function in R^n, apply the following iterative method. At the beginning of the k-th iteration, one has an estimate x_k of the variables, an $n \times n$ symmetric matrix B_k which need not be positive definite, and a trust region radius Δ_k. The gradient

$$g_k = \nabla f(x_k)$$

is calculated, and , if it is nonzero, a trial step d_k is computed by solving the subproblem

$$\begin{array}{ll} \min\limits_{d \in R^n} & g_k^T d + \frac{1}{2} d^T B_k d := \phi_k(d) \\ s.t. & ||d|| \le \Delta_k, \end{array} \tag{1.2}$$

where $|| \cdot ||$ refers to the 2-norm. Let d_k be the solution of (1.2). The predicted reduction is defined by the reduction of the approximate model, that is,

$$Pred_k = \phi_k(0) - \phi_k(d_k),$$

and the actual reduction is defined by

$$Ared_k = f(x_k) - f(x_k + d_k).$$

The ratio between these two reductions is defined by

$$r_k = \frac{Ared_k}{Pred_k},$$

and it plays a key role to decide whether the trial step is acceptable and to adjust the new trust region radius. The next iterate x_{k+1} is chosen by the following formula:

$$x_{k+1} = \begin{cases} x_k + d_k & \text{if } r_k > c_0, \\ x_k & \text{otherwise,} \end{cases} \tag{1.3}$$

where $c_0 \in [0, 1)$ is a constant. The trust region radius for the next iteration is chosen as

$$\Delta_{k+1} \in \begin{cases} [c_3||d_k||, c_4\Delta_k] & \text{if } r_k \leq c_2, \\ [\Delta_k, c_1\Delta_k] & \text{otherwise,} \end{cases} \tag{1.4}$$

where $c_i (i = 1, 2, 3, 4)$ are positive constants that satisfy $c_1 > 1 > c_4 > c_3$ and $c_2 \in [c_0, 1)$. Typical values are $c_0 = 0$, or a small constant $c_0 = 0.0001$, $c_1 = 2, c_2 = 0.25, c_3 = 0.25, c_4 = 0.5$. The advantage of using zero c_0 is that a trial step is accepted whenever the objective function is reduced, hence, it would not throw away a "good point", which is a desirable property especially when the number of the varibles are large and when the function evaluations are very expensive. But we can only obtain the weakly global convergence for algorithms with $c_0 = 0$, that is, at least one of the accumulation points is a stationary point. While we can obtain the strong global convergence for those with $c_0 > 0$, that is, all accumulation points are stationary points, for more details, please see Yuan [14] and [16].

As it is known, when the sequence generated by the traditional trust region algorithm converges to the minimizer of (1.1), the ratio $\{r_k\}$ converges to one. By the updating rule (1.4), we know the trust region radius Δ_k will be larger than a positive constant for all large k. Since $\{||x_k - x^*||\}$ converges to zero, the trust region will not play the role at the end.

In the next section, we first show the relationship between the trust region method and the Levenberg-Marquardt method (see [3] [4]). Then based on the work of papers [13] and [1] concerned with the Levenberg-Marquardt method for nonlinear equations, we come out with the idea of a new trust region algorithm with the trust region converging to zero. In fact, when the sequence is sufficiently close to the minimizer, the trust region radius Δ_k need not be larger than a positive constant, it suffices for the convergence that Δ_k be larger than $O(||x_k - x^*||)$ at the k-th iteration, which means that $\{\Delta_k\}$ can converge to zero.

we consider the choice of the trust region radius as $\Delta_k = \mu_k||g_k||$ under some condition, where μ_k is updated according to the ratio r_{k-1}. Since the gradient $\{g_k\}$ converges to zero as the sequence converges to the minimizer, the radius $\{\Delta_k\}$ will converges to zero. We present the new algorithm in Section 3 and show that it preserves the global convergence properties of the traditional trust region algorithms. In Section 4, we prove the superlinear convergence of the algorithm under some suitable conditions. Finally in Section 5, we compare the new algorithm with the traditional trust region algorithm, and present the numerical results.

2 The Levenberg-Marquardt method and trust region

The Levenberg-Marquardt method is a method for solving nonlinear equations. It is often mentioned when the history of trust region method is discussed. The reason is that the techniques of trust region is, in some sense, equivalent to the Levenberg-Marquardt method.

Consider the system of nonlinear equations

$$F(x) = 0, \tag{2.1}$$

where $F(x) : R^n \rightarrow R^n$ is continuously differentiable. We try to compute a least squares solution, which means that we need to solve the nonlinear least squares problem

$$\min_{d \in R^n} \quad ||F(x)||_2^2. \tag{2.2}$$

The Gauss-Newton method for (2.2) computes the trial step at the k-th iterate by

$$d_k^{GN} = -J(x_k)^{+} F(x_k), \tag{2.3}$$

where $J(x_k) = F'(x_k)$ is the Jocabi, and $J(x_k)^{+}$ is the generalized inverse of $J(x_k)$. It is easy to see that the Gauss-Newton step (3.3) is the minimum norm solution of the subproblem

$$\min_{d \in R^n} \quad ||F(x_k) + J(x_k)d||_2^2, \tag{2.4}$$

which is an approximation model of problem (2.2) near the current iterate x_k. The difficulty of using the Gauss-Newton step is that the Jacobi $J(x_k)$ may be ill conditioned, which normally leads to a very large step d_k.

The Levenberg-Marquardt method for nonlinear equations (2.1) computes the trial step by

$$d_k = -(J(x_k)^T J(x_k) + \lambda_k I)^{-1} J(x_k)^T F(x_k), \tag{2.5}$$

where $\lambda_k \geq 0$ is a parameter being updated from iteration to iteration. The Levenberg-Marquardt step (2.5) is a modification of the Gauss-Newton step. The parameter λ_k is used to prevent d_k from being too large when $J(x_k)$ is nearly singular.

It is easily seen that d_k given by (2.5) is the solution of the problem

$$\min_{d \in R^n} ||F(x_k) + J(x_k)d||^2 + \lambda_k ||d||^2, \tag{2.6}$$

which is a modification of (2.4), the additional term $\lambda_k ||d||^2$ can be viewed as a penalty term which prevents the step d_k from being too large. Define

$$\Delta_k = ||(J(x_k)^T J(x_k) + \lambda_k I)^{-1} J(x_k)^T F(x_k)||, \tag{2.7}$$

then it is not difficult to show that the Levenberg-Marquardt step (2.5) ia also a solution to the following subproblem

$$\begin{aligned} &\min_{d \in R^n} \quad ||F(x_k) + J(x_k)d||^2 \\ &s.\ t. \qquad ||d|| \leq \Delta_k. \end{aligned} \tag{2.8}$$

In fact, if we let the trust region radius Δ_k be given by (2.7) in every iteration, then Levenberg-Marquardt algorithm is essentially a trust region algorithm, and in every iteration, this trust region algorithm has active constraint, that is, we have $\Delta_k = ||d_k||$. However, the general trust region algorithm updates the trust region directly, while the Levenberg-Marquardt algorithm modifies the parameter λ_k in every iteration, which in turn modifies the value Δ_k from (2.7) implicitly. Many other papers also consider the Levenberg-Marquardt method and the trust regionthe method, for more details, please see [5, 6, 15, 16], etc. .

There are various choices of Levenberg-Marquardt parameter λ_k [5]. Papers [13] and [1] consider $\lambda_k = ||F(x_k)||^2$ and $\lambda_k = ||F(x_k)||$ for the nonlinear equations (2.1), respectively. Under the condition that $F(x)$ provides a local error bound near the solution [13], which is weaker than the nonsingularity, and under some other normal conditions, they obtain the superlinear and quadratic convergence, which means that the step $\{d_k\} \to 0$. In other words, if the trust region radius is given by

$$\Delta_k = ||(J(x_k)^T J(x_k) + \lambda_k I)^{-1} J(x_k)^T F(x_k)|| \tag{2.9}$$

with $\lambda_k = ||F(x_k)||$ or $\lambda_k = ||F(x_k)||^2$, then it follows from $\Delta_k = ||d_k||$ that $\{\Delta_k\} \to 0$.

Upon the above observations, we see that we can construct a trust region algorithm with the trust region radius converging to zero. In fact, we only need to ensure that the trust region includes the solution x^* for all sufficiently large k, that is, the trust region radius should be larger than $O(||x_k - x^*||)$. In the following section, we present the new trust region algorithm with the trust region radius given by $\Delta_k = \mu_k ||g_k||$, where μ_k is updated according to the ratio r_{k-1} and the relation between $||d_k||$ and Δ_k.

3 The algorithm and global convergence

We first present our new trust region algorithm with trust region radius converging to zero, then show that it preserves the strong global convergence properties of the traditional trust region algorithms.

Algorithm 3.1. *(New trust region algorithm)*

Step 1 *Given* $x_1 \in R^n, B_1 \in R^{n\times n} symmetric, \varepsilon \geq 0, c_2 > c_0 \geq 0, c_5 < 1 \leq c_6, \mu_1 > 0, \Delta_1 = \mu_1 ||g_1||, k := 1.$

Step 2 *If* $||g_k|| \leq \varepsilon$*, then stop;*

Solve (1.2) giving d_k*.*

Step 3 *Compute* $r_k = Ared_k / Pred_k$*; set*

$$x_{k+1} = \begin{cases} x_k + d_k & \text{if } r_k > c_0, \\ x_k & \text{otherwise.} \end{cases} \tag{3.1}$$

Step 4 *Choose* μ_{k+1} *as*

$$\mu_{k+1} = \begin{cases} c_5 \mu_k & \text{if } r_k < c_2, \\ c_6 \mu_k & \text{if } r_k \geq c_2 \text{ and } ||d_k|| > \frac{1}{2}\Delta_k, \\ \mu_k & \text{otherwise;} \end{cases} \tag{3.2}$$

Compute $\Delta_{k+1} = \mu_{k+1} ||g_{k+1}||$*;*

Step 5 Update B_{k+1}; $k := k+1$; go to Step 2.

The subproblem (1.2) has been studied by many authors, and the following lemmas are well known (for example, see Gay [2], Moré and Sorensen [8]).

Lemma 3.1. *A vector $d^* \in R^n$ is a solution of the problem*

$$\begin{array}{ll} \min\limits_{d \in R^n} & g^T d + \frac{1}{2} d^T B d := \phi(d) \\ s.t. & ||d|| \leq \Delta, \end{array} \tag{3.3}$$

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

$$(B + \lambda^* I) d^* = -g, \tag{3.4}$$

$$\lambda^* (\Delta - ||d^*||) = 0, \tag{3.5}$$

and $B + \lambda^* I$ is positive semi-definite.

Lemma 3.2. *(Powell 1975) If d^* is a solution of (3.3), then*

$$\phi(0) - \phi(d^*) \geq \frac{1}{2} ||g|| \min\{\Delta, ||g||/||B||\}. \tag{3.6}$$

Lemma 3.2 shows that the reduction in the trust region model will not be very small unless $||g||\Delta$ or $||g||^2/||B||$ is very small. This property is very important for proving the global convergence of the trust region algorithms. The global convergence depends on the fact that the predicted reduction satisfies (3.6). Hence, in stead of solving (1.2) exactly, we can compute a trial step d_k that satisfies

$$\phi_k(0) - \phi_k(d_k) \geq \delta ||g_k|| \min\{\Delta_k, ||g_k||/||B_k||\}, \tag{3.7}$$

where δ is some positive constant. To compute a vector d_k that satisfies (3.7) is usually much easier than solving (1.2) exactly. The vector d_k can be calculated by dog-leg type techniques, (see Powell [10]) or by applying the Newton's method to the following nonlinear equation (see Gay [2], Moré and Sorensen[8]),

$$\varphi(\lambda) = \frac{1}{||(B_k + \lambda I)^{-1} g_k||} - \frac{1}{\Delta_k} = 0. \tag{3.8}$$

The subproblem can also be solved approximately by the novel algorithm proposed by Nocedal and Yuan [9] or by a preconditioned conjugate gradient method (Steihaug [12]).

Theorem 3.1. *Assume that $f(x)$ is differentiable and bounded below, $g(x)$ is uniformly continuously. Let $\{x_k\}$ be generated with d_k satisfying (3.7). If $\varepsilon = 0$ is chosen in Algorithm 3.1, and if there exists a positive constant M such that*

$$||B_k|| \leq M \tag{3.9}$$

holds for all k, then it follows that

$$\liminf_{k \to \infty} ||g_k|| = 0. \tag{3.10}$$

Proof. If the theorem is not true, there exists a positive constant τ such that

$$||g_k|| \geq \tau \tag{3.11}$$

holds for all k. Define the set

$$\mathbb{I} = \{k \mid r_k \geq c_2\}. \tag{3.12}$$

Since $f(x)$ is bounded below, it follows from (3.7),(3.9) and (3.11) that

$$\begin{aligned} +\infty &> \sum_{i=1}^{\infty}(f_k - f_{k+1}) \\ &\geq \sum_{k\in\mathbb{I}} c_2(\phi_k(0) - \phi_k(d_k)) \\ &\geq \sum_{k\in\mathbb{I}} \delta\tau c_2 \min\{\Delta_k, \frac{\tau}{M}\}. \end{aligned} \tag{3.13}$$

The above relation and (3.11) indicate that

$$\sum_{k\in\mathbb{I}} \min\{\mu_k, \frac{\tau}{M}\} < +\infty, \tag{3.14}$$

If $\mathbb{I}$ is finite, we have from (3.2) that $\mu_{k+1} = c_5\mu_k$ for all large k, so that $\{\mu_k\} \to 0$. If $\mathbb{I}$ is infinite, then we also have from (3.14) that

$$\lim_{k\in\mathbb{I}} \ \mu_k = 0. \tag{3.15}$$

On the other hand, we have from (3.7) that

$$\begin{aligned} r_k &= \frac{Ared_k}{Pred_k} \\ &= 1 + \frac{o(||d_k||) + O(||d_k||^2||B_k||)}{Pred_k} \\ &\leq 1 + \frac{o(||d_k||) + O(||d_k||^2||B_k||)}{||g_k|| \min\{\Delta_k, ||g_k||/||B_k||\}} \\ &\leq 1 + \frac{o(||d_k||)}{\Delta_k} \\ &\to 1. \end{aligned} \tag{3.16}$$

The inequality above implies that there exists a positive constant μ^* such that

$$\mu_k > \mu^* \tag{3.17}$$

holds for all sufficiently large k, which gives a contradiction to (3.15). Therefore we see that assumption (3.11) can not be true. The proof is completed. □

Theorem 3.2. *Under the conditions of Theorem 3.1, if $c_0 > 0$ and if $\{||B_k||\}$ satisfy (3.9), then the sequence generated by Algorithm 3.1 satisfies*

$$\lim_{k\to\infty} ||g_k|| = 0. \tag{3.18}$$

4 Local convergence

The first convergence result for the traditional trust region algorithm is given by Powell(1975). It is showed that under some certain conditions, the algorithm converges $Q-$ superlinearly. The trust region radius will be larger than some positive constant, and the trial step will take quasi-Newton step. In our new trust region algorithm with trust region radius converging to zero, we show that the sequence $\{\mu_k\}$ is bounded, which means that the constraint is inactive for the subproblem (1.2), so the algorithm also takes the Quasi-Newton step at the end. In the following, we give the superlinear convergence of our new algorithm.

Theorem 4.1. *Assume that the trial step d_k in Step 2 of Algorithm 3.1 is a solution of subproblem (1.2). If $\varepsilon = 0$ and the sequence $\{x_k\}$ generated by Algorithm 3.1 converges to x^*, if $\nabla^2 f(x)$ is continuous in a neighbourhood of x^* and $\nabla^2 f(x^*)$ is positive definite, and if the condition*

$$\lim_{k\to\infty} ||(\nabla^2 f(x^*) - B_k)d_k||/||d_k|| = 0 \tag{4.1}$$

is satisfied, then the sequence $\{x_k\}$ converges to x^* Q-superlinearly.

Proof. From (4.1) and the positive definiteness of $\nabla^2 f(x^*)$, there exists a constant $\bar{\delta} > 0$ such that

$$d_k^T B_k d_k \geq \bar{\delta}||d_k||^2 \tag{4.2}$$

for all sufficiently large k. Because d_k is a solution of the subproblem (1.2), there exists $\lambda_k \geq 0$ such that

$$g_k + (B_k + \lambda_k I)d_k = 0, \tag{4.3}$$

which implies that

$$Pred_k = \frac{1}{2}d_k^T B_k d_k + \lambda_k||d_k||^2. \tag{4.4}$$

The continuity of $\nabla^2 f(x)$ and (4.1) show that

$$\begin{aligned} Ared_k &= -g_k^T d_k - \frac{1}{2}d_k^T\nabla^2 f(x^*)d_k + o(||d_k||^2) \\ &= Pred_k + \frac{1}{2}d_k^T(B_k - \nabla^2 f(x^*))d_k + o(||d_k||^2) \\ &= Pred_k + o(||d_k||^2). \end{aligned} \tag{4.5}$$

Then, it follows from (4.2), (4.4) and (4.5) that

$$\lim_{k\to\infty} r_k = \frac{Ared_k}{Pred_k} = 1. \tag{4.6}$$

Now, we prove that $\{\mu_k\}$ is bounded. Otherwise, $\mu_k \to +\infty$, and inequality

$$||d_k|| > \frac{1}{2}\Delta_k = \frac{1}{2}\mu_k||g_k|| \tag{4.7}$$

holds for infinitely many iterations. The positive definiteness of $\nabla^2 f(x^*)$ indicates that there exist $\hat{M} > \hat{\delta} > 0$ such that

$$\hat{M}||x_k - x^*|| \geq ||g_k|| \geq \hat{\delta}||x_k - x^*|| \tag{4.8}$$

$$\hat{M}||x_k - x^*||^2 \geq f(x_k) - f(x^*) \geq \hat{\delta}||x_k - x^*||^2 \tag{4.9}$$

for all large k. Thus, (4.7)–(4.9) show that

$$\begin{aligned} \hat{M}||x_k - x^*||^2 &\geq f(x_k) - f(x^*) \\ &\geq f(x_k + d_k) - f(x^*) \\ &\geq \hat{\delta}||x_k + d_k - x^*||^2 \\ &\geq \hat{\delta}(||d_k|| - ||x_k - x^*||)^2 \\ &\geq \hat{\delta}(\frac{1}{2}\mu_k\hat{\delta} - 1)^2||x_k - x^*||^2, \end{aligned} \tag{4.10}$$

which is impossible if $\mu_k \to +\infty$. Therefore we see that $\{\mu_k\}$ is bounded. This implies

$$||d_k|| \leq \frac{1}{2}\Delta_k \tag{4.11}$$

for all large k. Thus, the trust region is inactive for all large k. Consequently, the superlinear convergence result follows from the standard results of Dennis and Moré. □

5 Numerical results

In this section, We implemented the new Algorithm 3.1 (NTR) in two versions and compared it with the traditional trust region algorithm (TTR).

In both the tests of TTR and NTR, the trial step d_k is computed approximatelyby the algorithm proposed by Nocedal and Yuan in [9] (Algorithm 2.6) for solving the subproblem (1.2). We present it as follows.

Algorithm 5.1. *(Algorithm for approximate solution of (1.2))*

Step 1 *Given constants* $\gamma > 1$ *and* $\varepsilon_0 > 0$*, set* $\lambda := 0$.

If B_k *is positive definite, go to Step 2;*

else find $\lambda \in [0, ||B_k|| + (1 + \varepsilon_0)||g_k||/\Delta_k]$ *such that* $B_k + \lambda I$ *is positive definite.*

Step 2 *Factorize* $B_k + \lambda I = R_k^T R_k$*, where* R_k *is upper triangular,*

and solve $R_k^T R_k d = -g_k$ *for* d_k.

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

$$\lambda := la + \frac{||d_k||^2}{||q||^2}\frac{\gamma||d_k|| - \Delta}{\Delta}; \tag{5.1}$$

go to Step 2.

The initial trust region radius for both the algorithms is $\Delta_1 = ||g_1||$, that is, $\mu_1 = 1$ for the new trust region algorithm NTR. B_1 is chosen as the identy matrix, and B_k is updated by the BFGS formula. However, we do not update B_k if

$$s_k^T y_k > 0 \tag{5.2}$$

fails, where

$$\begin{cases} s_k = x_{k+1} - x_k, \\ y_k = g_{k+1} - g_k. \end{cases} \tag{5.3}$$

We use the parameter $c_0 = 0.0001$, compute

$$\Delta_{k+1} = \begin{cases} \min\{\frac{\Delta_k}{4}, \frac{||d_k||}{2}\} & \text{if } r_k < 0.25, \\ \Delta_k & \text{if } r_k \in [0.25, 0.75], \\ \max\{4||d_k||, 2\Delta_k\} & \text{otherwise} \end{cases} \tag{5.4}$$

in the test of TTR, and compute

$$\mu_{k+1} = \begin{cases} c_5\mu_k & \text{if } r_k < 0.25, \\ c_6\mu_k & \text{if } r_k \geq 0.25 \text{ and } ||d_k|| > \frac{1}{2}\Delta_k, \\ \mu_k & \text{otherwise} \end{cases} \tag{5.5}$$

and

$$\Delta_{k+1} = \mu_{k+1}||g_{k+1}|| \tag{5.6}$$

in the test of NTR, where $c_5 = 1/6, c_6 = 6$ in Version 1 and $c_5 = 1/6, c_6 = 8$ in Version 2.

The test problems were those given by Moré, Garbow and Hillstrom [7], and we used the same numbering system as in [7]. The algorithm is terminated when the norm of the gradient at the k-th iterate $||g_k||$ is less than $\varepsilon = 10^{-8}$, or when the number of the iterations exceeds $100(n+1)$. The results are listed in Table 1. "NF" and "NG" represent the numbers of function calculations and gradient calculations, respectively; if the method failed to find the stational point in $100(n+1)$ iterations, we denoted it by the sign "–".

Table 1
Results on some of the problems of Moré, Garbow and Hillstrom

		TTR BFGS	NTR Version 1	NTR Version 2
Problem	n	NF/NG	NF/NG	NF/NG
1	3	34/28	38/29	34/24
2	6	46/42	44/39	51/43
3	3	7/6	7/6	7/6
5	3	44/38	28/25	29/26
6	3	15/10	12/8	12/8
7	9	78/71	80/71	76/69
8	8	83/64	100/80	63/49
9	2	15/12	16/11	14/12
10	2	46/27	—-	—-
12	3	47/37	45/37	44/37
13	6	25/24	26/26	28/26
14	6	94/74	97/66	104/63
15	8	79/66	98/88	82/67
16	2	20/17	17/15	20/18
17	4	117/81	107/71	77/50
18	9	50/34	51/34	49/33

From the table, we see that our new trust region algorithm NTR performs little better than the traditional trust region algorithm TTR in Version 1. While in Version 2, the NTR usually performs better than the TTR. For ten problems, the NTR outperforms the TTR, while only for five problems, the TTR outperforms the NTR. So it seems that Algorithm (3.1) is efficient in solving small size optimization problems.

References

[1] J. Y. Fan and Y. X. Yuan, On the convergence of a new Levenberg-Marquardt method, Report 2001-005, AMSS, Chinese Academy of sciences.

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

[3] K. Levenberg, A method for the solution of certain nonlinear problems in least squares, *Quart. Appl. Math.,* 2(1944), 164-166.

[4] D. W. Marquardt, An algorithm for least-squares estimation of nonlinear inequalities, *SIAM J. Appl. Math.,* 11(1963), 431-441.

[5] J. J. Moré, The Levenberg-Marquardt algorithm: implementation and theory, in: G. A. Watson, ed., *Lecture Notes in Mathematics 630: Numerical Analysis,* Springer-Verlag, Berlin, 1978, pp. 105-116.

[6] J. J. Moré, Recent developments in algorithms and software for trust region methods, In: A. Bachem, M. Grötschel and B. Korte, eds., *Mathematical Programming: The State of Art* (Springer, Berlin, 1983) pp. 258-287.

[7] J. J. Moré, B. S. Garbow and K. H. Hillstrom, Testing unconstrained optimization software, *ACM Trans. Math. Software,* 7(1981), 17-41.

[8] J. J. Moré and D. C. Sorensen, Computing a trust region step *SIAM J. Sci. Stat. Comp.,* 4(1983), 553-572.

[9] J. Nocedal and Y. X. Yuan, Combining trust region and line search techniques, in: Y. Yuan, ed., *Advances in Nonlinear Programming* Kluwer, 1998, pp. 153–175.

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

[11] M. J. D. Powell, 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, 1975, pp. 1-27.

[12] T. Steihaug, The conjugate gradient method and trust regions inlarge scale optimization, *SIAM J. Numer. Anal.,* 20(1983), 626-637.

[13] N. Yamashita and M. Fukushima, On the rate of convergence of the Levenberg-Marquardt method, Report, presented in the first Sino-Japan optimization conference, October, 2000.

[14] Y. X. Yuan, Nonlinear Programming: trust region algorithms, in: S.T. Xiao and F. Wu, eds., *Proceedings of Chinese SIAM annual meeting*, Tsinghua University, Beijing, 1994, pp. 83 –97.

[15] Y. X. Yuan, Trust region algorithms for nonlinear equations, *Information,* 1(1998), 7-20.

[16] Y. X. Yuan, A review of trust region algorithms for optimization, In: J.M. Ball and J.C.R. Hunt, eds., *ICM99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics*, Oxford University Press, 2000, pp. 271–282.

[17] Y. X. Yuan and W. Y. Sun, *Optimization Theories and Methods (in Chinese)*, Science Press, Beijing, 1997.