Math Meth Oper Res (2003) 58:283–298
DOI 10.1007/s001860300302

Mathematical Methods
of Operations Research

A new trust region method for nonlinear equations*

Ju-liang Zhang[1], **Yong Wang**[2]

[1] Department of Management Science and Engineering, School of Economics and Management, Tsinghua University, Beijing, 100084, PRC China (e-mail: jlzhang68@mail.tsinghua.edu.cn)
[2] Institute of Applied mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing, 100080, China (e-mail: ywang@amss.ac.cn)

Manuscript received: December 2002/Final version received: April 2003

Abstract. In this paper, a new trust region method for the system of nonlinear equations is presented in which the determining of the trust region radius incorporates the information of its natural residual. The global convergence is obtained under mild conditions. Unlike traditional trust region method, the superlinear convergence of the method is proven under the local error bound condition. This condition is weaker than the nondegeneracy assumption which is necessary for superlinear convergence of traditional trust region method. We also propose an approximate algorithm for the trust region subproblem. Preliminary numerical experiments are reported.

Key words: Nonlinear equation system, Trust region method, Local error bound, Superlinear convergence

1 Introduction

We consider the nonlinear equation system

$$F(x) = 0, \tag{1.1}$$

where $F : R^n \to R^m$ is a set of continuously differentiable functions. Throughout the paper, we assume that the solution set of (1.1) is nonempty and denoted by X^*. In all cases, $\|\cdot\|$ denotes 2-norm. Let $F'(x)$ denote the transpose of the Jacobian of $F(x)$, i.e., $F'(x) = (F_1'(x), \ldots, F_m'(x))^T$.

A problem which is closely related to (1.1) is the following minimization problem

* This work is supported in part by National Science Foundation of China (Grant No.10171055).

$$\min_{x\in R^n} \Psi(x) = \tfrac{1}{2}\|F(x)\|^2. \tag{1.2}$$

This problem is called the least square problem. Obviously, x^* solves (1.1) iff x^* solves (1.2) when X^* is nonempty.

(1.1) and (1.2) have many applications in engineering, such as nonlinear fitting, parameter estimating and function approximating. At present, a lot of algorithms have been proposed for solving these two problems, for examples, Gauss-Newton method, Levenberg-Marquardt method, trust region method, etc., see [1–3, 5, 6, 9, 11, 12]. These algorithms are superlinearly convergent if $F'(x^*)$ is nondegenerate. Here we are interested in trust region method since it has strong convergence and robustness. For the traditional trust region methods, at each iterative point x_k (nonstationary point), the trial step is obtained by solving the following trust region subproblem

$$\begin{array}{ll} \min & \Phi_k(d) = \tfrac{1}{2}\|F(x_k) + F'(x_k)d\|^2 \\ \text{s.t.} & \|d\| \le \Delta. \end{array} \tag{1.3}$$

It is well known that the trust region methods are globally convergent under suitable conditions and superlinearly convergent under the condition that $F'(x^*)$ (x^* is a solution of (1.1)) is nondegenerate. However, nondegeneracy of $F'(x^*)$ seems a too stringent requirement for the purpose of ensuring superlinear convergence. Recently, Yamashita and Fukushima [11] showed that Levenberg-Marquardt method with *Armijo* search is superlinearly convergent under the local error bound condition. From [11], we know that the local error bound condition is weaker than the nondegeneracy of $F'(x^*)$, i.e., $\|F(x)\|$ provides a local error bound on some neighborhood of x^* if $F'(x^*)$ is nondegenarate and the converse is not true (A example was provided in [11]). The local error bound condition for the system (1.1) is defined as follows.

Definition 1.1. *Let N be a subset of R^n such that $X^* \cap N \neq \emptyset$. We say that $\|F(x)\|$ provides a local error bound on N for the system (1.1) if there exists a positive constant c such that*

$$c \cdot dist(x, X^*) \le \|F(x)\|, \quad \forall x \in N. \tag{1.4}$$

where $dist(x, X^*) = \min_{y\in X^*}\{\|x - y\|\}$.

In this paper, we propose a new trust region method for (1.1) or (1.2) in which $\|F(x_k)\|$ is incorporated to determine the trust region radius. There are two advantages in using the $\|F(x_k)\|$ information: the first is that it is effective to determine trust region radius by using the information of $\|F(x_k)\|$. The second is in theoretic consideration since in this way we can prove that the new algorithm is superlinearly convergent under the local error bound condition. In this case, we obtain a new trust region method in which the conditions ensuring superlinear convergence are weaker than that in the traditional trust region methods.

For the trust region methods, the main computation is spent to solve the subproblem. It is well known that solving the trust region subproblem exactly is expensive. Hence developing approximate methods for the trust region subproblem has been a popular research topic since 1980's and numerous algorithms have been proposed. However, these existed algorithms are not suitable for our method since we do not assume the non-

degeneracy of the Jacobian. In Section 4, a new approximate method is proposed for the trust region subproblem. In this method, if an iterative point x_k is far from the solution, the accuracy needed for terminating the approximate method is low, then the computation cost for the trial step is less. As the iterative sequence approaches to the solution, the accuracy becomes high. Finally we prove that the new trust region method with the trial step generated by the approximate method is globally and superlinearly convergent under the local error bound condition.

The remainder of the paper is arranged as follows. In Section 2, the algorithm model is presented and the global convergence is studied. In Section 3, we show the local convergence of the algorithm. A new approximate algorithm for trust region method is presented in Section 4. In Section 5, numerical experiments on some classical test problems are reported to test the efficiency of the new algorithm. Some conclusions are given in Section 6.

2 Algorithm model and global convergence

In our algorithm, at each iterative point x_k, the trial step is obtained by solving the following subproblem

$$\begin{array}{ll} \min & \Phi_k(d) = \frac{1}{2}\|F(x_k) + F'(x_k)d\|^2 \\ \text{s.t.} & \|d\| \le c^p \|F(x_k)\|^\gamma \delta_k^p, \end{array} \tag{2.1}$$

where $0 < c < 1$, p is a nonnegative integer and $0 < \gamma < 1$. Let d_k^p be the solution of (2.1) corresponding to p.

Then we define the actual reduction as

$$Ared_k(d_k^p) = \Psi(x_k + d_k^p) - \Psi(x_k), \tag{2.2}$$

the predict reduction as

$$Pred_k(d_k^p) = \Phi_k(d_k^p) - \Psi(x_k), \tag{2.3}$$

and the ratio of actual reduction over predict reduction as

$$r_k^p = \frac{Ared_k(d_k^p)}{Pred_k(d_k^p)}.$$

Algorithm 2.1.

Step 0. Given initial point $x_0 \in R^n$, $0 < \eta < 1$, $0 < c < 1$, $\epsilon > 0$, $0.5 < \gamma < 1$, $p = 0$, set $k := 0$;

Step 1. If $\|F'(x_k)^T F(x_k)\| \le \epsilon$, stop. Otherwise, solve (2.1) to obtain the trial step d_k^p;

Step 2. Calculate $Pred_k(d_k^p)$, $Ared_k(d_k^p)$ and r_k^p. If $r_k^p \ge \eta$, then $x_{k+1} = x_k + d_k^p$, go to Step 3. Otherwise, set $p := p + 1$ go to Step 1.

Step 3. Set $k := k + 1$, set $p = 0$, choose $\gamma \in (0.5, 1)$, go to Step 1.

In order to analyze the global convergence of the algorithm, the following assumption are needed.

Assumption 2.1. (i). $F(x)$ is twice continuously differentiable;

(ii). $\{x_k\}$ is a bounded sequence.

By Assumption 2.1 there exists $M > 0$ such that

$$\|F'(x_k)^T F'(x_k)\| \leq M \quad \forall k. \tag{2.4}$$

First, we give several lemmas which are important for the analysis of the global convergence.

Lemma 2.1. $|Ared_k(d_k^p) - Pred_k(d_k^p)| = O(\|d_k^p\|^2)$.

Lemma 2.2. $Pred_k(d_k^p) \leq -\frac{1}{2}\min\{\|F'(x_k)^T F(x_k)\|/M, \delta_k^p\}\|F'(x_k)^T F(x_k)\|$.

Proof. By the definition of d_k^p, we know that for any $\alpha \in (0,1)$

$$\begin{aligned} Pred_k(d_k^p) &= \Phi_k(d_k^p) - \Psi(x_k) \\ &\leq \Phi_k\left(-\alpha\frac{\delta_k^p}{\|F'(x_k)^T F(x_k)\|}F'(x_k)^T F(x_k)\right) - \Psi(x_k) \\ &= -\alpha\delta_k^p\|F'(x_k)^T F(x_k)\| \\ &\quad + \frac{\alpha^2\delta_k^{p2}F(x_k)^T F'(x_k)F'(x_k)^T F'(x_k)F'(x_k)^T F(x_k)}{2\|F'(x_k)^T F(x_k)\|^2} \\ &\leq -\alpha\delta_k^p\|F'(x_k)^T F(x_k)\| + \frac{1}{2}\alpha^2\delta_k^{p2}M. \end{aligned}$$

Thus,

$$\begin{aligned} Pred_k(d_k^p) &\leq \min_{0\leq\alpha\leq 1}\left\{-\alpha\delta_k^p\|F'(x_k)^T F(x_k)\| + \frac{1}{2}\alpha^2\delta_k^{p2}M\right\} \\ &\leq -\frac{1}{2}\min\left\{\|F'(x_k)^T F(x_k)\|/M, \delta_k^p\right\}\|F'(x_k)^T F(x_k)\|. \end{aligned}$$

□

Lemma 2.3. *Algorithm* 2.1 *does not circle between Step* 1 *and Step* 2 *infinitely*.

Proof. If Algorithm 2.1 circles between Step 1 and Step 2 at x_k infinitely, then for all $i = 1, 2, \ldots$, we have $x_{k+i} = x_k$, $p = i$ and $\|F'(x_k)^T F(x_k)\| > \epsilon$. Hence

$$\delta_k^i \to 0 \text{ and } r_k^i < \eta. \tag{2.5}$$

Therefore by Lemma 2.1 and Lemma 2.2, as $i \to \infty$

$$|r_k^i - 1| = \frac{|Ared_k(d_k^i) - Pred_k(d_k^i)|}{|Pred_k(d_k^i)|} \leq \frac{O(\delta_k^{i\,2})}{0.5\delta_k^i\|F'(x_k)F(x_k)\|} \to 0.$$

Thus, for i sufficiently large

$$r_k^i \geq \eta, \tag{2.6}$$

Which contradicts (2.5). □

Theorem 2.1. *Suppose that Assumption* 2.1 *holds. Then either the algorithm terminates finitely at a stationary point of* $\Psi(x_k)$ *or generates an infinite sequence* $\{x_k\}$ *such that*

$$\lim_{k\to\infty} \|F'(x_k)^T F(x_k)\| = 0. \tag{2.7}$$

Proof. Since $\Psi(x) \geq 0$ for all $x \in R^n$, then the sequence $\{\Psi(x_k)\}$ is bounded from below. Suppose that the algorithm does not stop finitely. If (2.7) is not true, then there exists a positive constant $\bar{\epsilon}$ and an infinite subsequence $\{k_i\}$ such that $\|F'(x_{k_i})^T F(x_{k_i})\| \geq \bar{\epsilon}$. Let $T = \{k | \|F'(x_k)^T F(x_k)\| \geq \bar{\epsilon}\}$. Meanwhile, by Assumption 2.1 and $\|F'(x_k)^T F(x_k)\| \geq \bar{\epsilon} (k \in T)$, $\|F(x_k)\| (k \in T)$ is bounded away from 0. Without loss of generality, we can assume $\|F(x_k)\| \geq \bar{\epsilon}$, $\forall k \in T$.

By Step 2 and Lemma 2.2, we have

$$\sum_{k\in T} [\Psi(x_k) - \Psi(x_{k+1})] \geq -\sum_{k\in T} \eta \cdot Pred_k(d_k^{p_k}) \geq \sum_{k\in T} \eta \cdot \frac{1}{2} \min\{\bar{\epsilon}/M, c^{p_k}\bar{\epsilon}\} \cdot \bar{\epsilon}$$

where p_k is the largest p value obtained in Step 2 at the iterative point x_k.

Since $\{\Psi(x_k)\}$ is bounded from below, we have

$$\sum_{k\in T} \eta \cdot \frac{1}{2} \min\{\bar{\epsilon}/M, c^{p_k}\bar{\epsilon}\} \cdot \bar{\epsilon} < +\infty.$$

Then $p_k \to +\infty$ as $k \to +\infty$ and $k \in T$. Therefore, we can assume $p_k \geq 1$ for all $k \in T$.

From the determination of $p_k (k \in T)$ in Step 2, the solution $\tilde{d}_k$ corresponding to the following subproblem

$$\begin{array}{ll} \min & \Phi_k(d) = \frac{1}{2}\|F(x_k) + F'(x_k)d\|^2 \\ \text{s.t.} & \|d\| \leq c^{p_k - 1}\|F(x_k)\|^{\gamma} \end{array} \tag{2.8}$$

is unacceptable. Namely, if let $\tilde{x}_{k+1} = x_k + \tilde{d}_k$, we have

$$\frac{\Psi(x_k) - \Psi(\tilde{x}_{k+1})}{-Pred_k(\tilde{d}_k)} < \eta. \tag{2.9}$$

But from Lemma 2.2

$$-Pred_k(\tilde{d}_k) \geq \frac{1}{2} \min\{\bar{\epsilon}/M, c^{p_k - 1}\bar{\epsilon}\} \cdot \bar{\epsilon}.$$

Follows from Lemma 2.1 that

$$\Psi(\tilde{x}_{k+1}) - \Psi(x_k) - Pred_k(\tilde{d}_k) = O(\|\tilde{d}_k\|^2) = O(c^{2(p_k - 1)}).$$

Thus

$$\left| \frac{\Psi(\tilde{x}_{k+1}) - \Psi(x_k)}{Pred_k(\tilde{d}_k)} - 1 \right| \leq \frac{O(c^{2(p_k - 1)})}{\frac{1}{2}\min\{\bar{\epsilon}^{\gamma}/M, c^{p_k - 1}\bar{\epsilon}^{\gamma}\} \cdot \bar{\epsilon}^{\gamma}}.$$

Because $p_k \to +\infty$ as $k \to +\infty$ and $k \in T$, we have

$$\frac{\Psi(x_k) - \Psi(\tilde{x}_{k+1})}{-Pred_k(\tilde{d}_k)} \to 1 \quad \in T,$$

which contradicts (2.9). □

Remark. Theorem 2.1 says that the iterative sequence $\{x_k\}$ generated by our algorithm satisfies $\|F'(x_k)^T F(x_k)\| \to 0$. If x^* is a cluster point of $\{x_k\}$ and $F'(x^*)$ is nondegenerate, then we have $\|F(x_k)\| \to 0$. This is a standard convergence result for nonlinear equations. At present, there is no algorithm which has the property that the iterative sequence generated by the algorithm satisfies $\|F(x_k)\| \to 0$ without the assumption that $F'(x^*)$ is nodegenerate.

3 Superlinear convergence

In order to analyze the superlinear convergence, we need the following assumption.

Assumption 3.1. (i) $x_k \to x^*$, where x^* is a solution of (1.1);

(ii) There exist $b \in (0,1)$ and $c_1 \in (0,\infty)$ such that

$$\|F'(y)(x-y)-F(x)+F(y)\| \le c_1\|x-y\|^2 \quad \forall x,y \in N(x^*,b)=\{x|\|x-x^*\| \le b\};$$

(iii) $\|F(x)\|$ provides a local error bound on $N(x^*,b)$ for the system (1.1), i.e., there exists $c_2 \in (0,\infty)$ such that

$$c_2 \cdot dist(x,X^*) \le \|F(x)\| \quad \forall x \in N(x^*,b);$$

(iv) $0.5 < \gamma < 1$.

Assumption 3.1 (ii) holds when $F(x)$ is continuously differentiable and $F'(x)$ is Lipschitz continuous. Note that by Assumption 3.1 (ii) there exists $L > 0$ such that

$$\|F(x) - F(y)\| \le L\|x - y\| \quad \forall x, y \in N(x^*, b). \tag{3.1}$$

By Assumption 3.1(i) and the continuity of $\|F(x)\|$ we know that $\|F(x_k)\| \to 0$.

In what follows, let $\hat{x}_k \in X^*$ such that

$$dist(x_k, X^*) = \|\hat{x}_k - x_k\|. \tag{3.2}$$

From Assumption 3.1 (i), for k sufficiently large

$$\|x_k - x^*\| \le \frac{b}{2}.$$

Then

$$\|x_k - \hat{x}_k\| \le \|x_k - x^*\| \le \frac{b}{2}. \tag{3.3}$$

Thus,

$$dist(x_k, X^*) = \|\hat{x}_k - x^*\| \le \|x_k - \hat{x}_k\| + \|x_k - x^*\| \le 2\|x_k - x^*\| \le b. \tag{3.4}$$

Note that $\gamma < 1$ and $x_k \to \hat{x}_k$, we have that for k sufficiently large

$$\|\hat{x}_k - x_k\| \le c_2^{\gamma}\|\hat{x}_k - x_k\|^{\gamma} \le \|F(x_k)\|^{\gamma}. \tag{3.5}$$

Since p starts from 0 at each iterative point x_k, $\hat{x}_k - x_k$ is a feasible point of (2.1) corresponding to $p = 0$ for k sufficiently large. Hence it follows from Assumption 3.1 (ii) that

$$\begin{aligned}\Phi_k(d_k^0) &\le \Phi_k(\hat{x}_k - x_k)\\ &= \frac{1}{2}\|F(x_k) + F'(x_k)(\hat{x}_k - x_k)\|^2\\ &= \frac{1}{2}\|F(x_k) + F'(x_k)(\hat{x}_k - x_k) - F(\hat{x}_k)\|^2\\ &\le \frac{1}{2}c_1\|\hat{x}_k - x_k\|^4\\ &= \frac{1}{2}c_1(dist(x_k, X^*))^4. \end{aligned} \tag{3.6}$$

Therefore

$$\|F(x_k) + F'(x_k)d_k^0\| = O((dist(x_k, X^*))^2). \tag{3.7}$$

In addition, by (2.1)

$$\|d_k^0\| \le \|F(x_k)\|^\gamma = O(\|x_k - \hat{x}_k\|^\gamma) = O((dist(x_k, X^*))^\gamma). \tag{3.8}$$

Hence, for k sufficiently large we have $x_k + d_k^0 \in N(x^*, b)$. Then

$$\begin{aligned} c_2 \cdot dist(x_k + d_k^0, X^*) &\le \|F(x_k + d_k^0)\|\\ &\le \|F(x_k) + F'(x_k)d_k^0\| + O(\|d_k^0\|^2)\\ &= O(\|x_k - \hat{x}_k\|^2) + O(\|x_k - \hat{x}_k\|^{2\gamma})\\ &= O(\|x_k - \hat{x}_k\|^{2\gamma})\\ &= O((dist(x_k, X^*)^{2\gamma}). \end{aligned} \tag{3.9}$$

Now we prove that for k sufficiently large the iteration formula is as follows

$$x_{k+1} = x_k + d_k^0. \tag{3.10}$$

In fact, for k sufficiently large, Assumption 3.1 (ii) and (3.7) imply

$$\begin{aligned} \left|Ared_k(d_k^0) - Pred_k(d_k^0)\right| &= \left|\frac{1}{2}\|F(x_k + d_k^0)\|^2 - \Phi_k(d_k^0)\right|\\ &= \left|\frac{1}{2}\left\|F(x_k) + F'(x_k)d_k^0 + O(\|d_k^0\|^2)\right\|^2\right.\\ &\quad \left. - \frac{1}{2}\left\|F(x_k) + F'(x_k)d_k^0\right\|^2\right|\\ &= \left\|F(x_k) + F'(x_k)d_k^0\right\| \cdot O(\|d_k^0\|^2) + O(\|d_k^0\|^4)\\ &= O(\|x_k - \hat{x}_k\|^{2+2\gamma}) + O(\|x_k - \hat{x}_k\|^{4\gamma})\\ &= O\Big((dist(x_k, X^*))^{4\gamma}\Big). \end{aligned} \tag{3.11}$$

It follows from Assumption 3.1 (iii) that $\|F(x_k)\| \ge c_2\|x_k - \hat{x}_k\|$. Then by (3.6)

$$\begin{aligned} |Pred_k(d_k^0)| &= \left|\Phi_k(d_k^0) - \frac{1}{2}\|F(x_k)\|^2\right|\\ &\ge c_2\|x_k - \hat{x}_k\|^2 + O(\|x_k - \hat{x}_k\|^4)\\ &= O((dist(x_k, X^*))^2). \end{aligned} \tag{3.12}$$

Note that $\gamma > 0.5$, from (3.11) and (3.12) we know that

$$\lim_{k\to\infty} |r_k^0 - 1| = \lim_{k\to\infty} \frac{|Ared_k(d_k^0) - Pred_k(d_k^0)|}{|Pred_k(d_k^0)|} = 0.$$

Hence $r_k^0 > \eta$ for k sufficiently large. So the iteration formula is (3.10) for k sufficiently large. With the above discussion, The following theorem is obtained.

Theorem 3.1. *With Assumption 3.1, for k sufficiently large the iteration formula is as follows*

$$x_{k+1} = x_k + d_k^0,$$

and

$$dist(x_{k+1}, X^*) = O((dist(x_k, X^*))^{2\gamma}),$$

i.e., Algorithm 2.1 is superlinearly convergent.

4 Approximate algorithm for the subproblem

In the previous sections, we adopt the exact solution of (2.1) as the trial step. However, Solving (2.1) exactly is expensive, especially for large scale problems. Therefore, it is necessary to have an approximate algorithm for (2.1). In this section, we propose an algorithm which solves (2.1) approximately and adopt the obtained approximate solution as the trial step. Here we use the idea in § 7.3 in [2]. For simplicity, we omit the subscript and drop the term $\frac{1}{2}\|F(x_k)\|^2$ in (2.1). Then (2.1) is equivalent to the following problem

$$\begin{aligned} \min_{d\in R^n} \quad & \Theta(d) = \tfrac{1}{2} d^T F'^T F' d + F^T F' d \\ \text{s.t.} \quad & \|d\| \le \delta. \end{aligned} \tag{4.1}$$

It is well known that d^M solves (4.1) iff there exists $\lambda^M \ge 0$ such that

$$(F'^T F' + \lambda^M I) d^M = -F'^T F, \tag{4.2}$$

$$\lambda^M (\|d^M\| - \delta) = 0, \tag{4.3}$$

$$\|d^M\| \le \delta. \tag{4.4}$$

where I is $n \times n$ unit matrix.

Based on (4.2)–(4.4), an approximate algorithm is designed as follows (the algorithm is stated in detail in Algorithm 4.1): First, we solve $F'^T F' d = -F'^T F$ to obtain its least norm solution $d(0)$ and check whether $\|d(0)\| \le \delta$. If $\|d(0)\| \le \delta$, then we obtain a solution of (4.1). Otherwise, we further find an approximation $\hat{\lambda} > 0$ of λ^M such that the solution $d(\hat{\lambda})$ of $(F'^T F' + \hat{\lambda} I) d = -F'^T F$ satisfying $\|d(\hat{\lambda})\| \approx \delta$. Then $d(\hat{\lambda})$ is an approximate solution of (4.1).

We use Newton method to obtain such a $\hat{\lambda}$. To this end, for $\lambda > 0$, define

$$\phi(\lambda) = \frac{1}{\|d(\lambda)\|} - \frac{1}{\delta}, \tag{4.5}$$

where $d(\lambda)$ is the solution of the following system of linear equations

$$\left(F'^T F' + \lambda I\right) d = -F'^T F. \tag{4.6}$$

Then $\lambda^M > 0$ solves (4.2) and (4.3) if and only if $\phi(\lambda^M) = 0$. As shown in § 7.3 in [2], $\phi(\lambda)$ is almost a linear function for $\lambda > 0$, we can obtain an approximation $\hat{\lambda} > 0$ of λ^M very efficiently by using Newton method. Moreover, let

$$F'^T F' + \lambda I = LL^T,$$

$$LL^T d(\lambda) = -F'^T F,$$

$$Lw(\lambda) = d(\lambda).$$

Then the Newton step is

$$-\frac{\phi(\lambda)}{\phi'(\lambda)} = \left(\frac{\|d(\lambda)\| - \delta}{\delta}\right)\left(\frac{\|d(\lambda)\|^2}{\|w(\lambda)\|^2}\right).$$

Thus, the iteration formula is simple. Since we want to keep $\lambda > 0$ at each iteration, in Step 7 of Algorithm 4.1, set $\lambda^+ = \lambda - \frac{\phi(\lambda)}{\phi'(\lambda)} > 0$ if $\lambda - \frac{\phi(\lambda)}{\phi'(\lambda)} > 0$, otherwise, $\lambda^+ = \frac{1}{2}\lambda$ (λ^+ is the value of next iteration of λ).

Now we discuss the termination criteria. In order to ensure that Algorithm 2.1 is globally convergent, we need the approximate solution $\tilde{d}$ satisfying

$$\Theta(\tilde{d}) \leq -\frac{1}{4}\min\left\{\left\|F'^T F\right\| \Big/ \left\|F'^T F'\right\|, \delta\right\}\left\|F'^T F\right\|. \tag{4.7}$$

However, it is possible that $\|F'^T F'\|$ equals to 0. Set $\bar{M} = \max\{\|F'^T F'\|, 1\}$. Then, instead of (4.7), we require

$$\Theta(\tilde{d}) \leq -\frac{1}{4}\min\left\{\left\|F'^T F\right\| \Big/ \bar{M}, \delta\right\}\left\|F'^T F\right\|. \tag{4.8}$$

It is analogous to the proof of Lemma 2.2 that $d(\lambda)$ satisfies (4.8) if λ is sufficiently close to λ^M. In addition, we also need a condition similar to (3.7) in order to obtain the superlinear convergence of Algorithm 2.1. To this end we require that

$$|\delta - \|d(\lambda)\|| \leq \|F\|^{2\gamma}. \tag{4.9}$$

Our approximate algorithm is stated below.

Algorithm 4.1.

Step 1. Factorize $F'^T F' = U^T \bigwedge U$, where U is an orthogonal matrix, $\bigwedge$ is a diagonal matrix. Calculate $d = -U^T(\bigwedge)^+ UF'^T F$, where $(\bigwedge)^+$ is the Moore-Penrose generalized inverse. If $\|d\| \leq \delta$, stop.
Step 2. Choose $\lambda > 0$.
Step 3. Factorize $F'^T F' + \lambda I = LL^T$.
Step 4. Solve $LL^T d = -F'^T F$.
Step 5. If $\Theta(d) \leq -\frac{1}{4}\min\{\|F'^T F\|/\bar{M}, \delta\}\|F'^T F\|$ and $|\delta - \|d\|| \leq \|F\|^{2\gamma}$, stop. Otherwise go to Step 6.

Step 6. Solve $Lw = d$.
Step 7. Let

$$\lambda := \begin{cases} \frac{1}{2}\lambda, & \text{if } \lambda + \left(\frac{\|d\|-\delta}{\delta}\right)\left(\frac{\|d\|^2}{\|w\|^2}\right) \le 0; \\ \lambda + \left(\frac{\|d\|-\delta}{\delta}\right)\left(\frac{\|d\|^2}{\|w\|^2}\right), & \text{otherwise,} \end{cases}$$

go to Step 3.

Remark. For this algorithm, it is noted that when x is far from the solution of (1.1), $\|F'F\|$ and $\|F\|$ are large. Then Algorithm 4.1 can stop early at Step 5 and the computation of trial step is less. As an iterative sequence approaches to the solution, the difference between the trial step and the exact solution of (2.1) becomes small. In this way, we can obtain the superlinear convergence of our trust region method. Note that our algorithm is different from the traditional trust region methods since the algorithm is superlinearly convergent without assuming of the nondegeneracy of $F'(x^*)$. Thus it is possible that the Newton step does not exist in our case. Therefore the existing approximate algorithms for the trust region subproblem, which guarantee the global and local superlinear convergence of the traditional trust region methods, can not be applied to our new method.

For this algorithm, we have the following results.

Lemma 4.1. *Algorithm* 4.1 *terminates finitely*.

Proof. If $\phi(\lambda_J) < 0$ for some $J > 0$, then by Lemma 7.3.2 in [2], the sequence $\{\lambda_j\}$ generated by Algorithm 4.1 converges to λ^M. Thus $\{d(\lambda_j)\}$ converges to d^M. Similar to Lemma 2.2, we can prove that $\Theta(d^M) \le -\frac{1}{2}\min\{\|F'^T F\|/\bar{M}, \delta\}\|F'^T F\|$. Since $\phi(\lambda^M) = 0$, there exists an integer $J > 0$ such that λ_J satisfying the termination criteria in Step 5.

If $\phi(\lambda_0) > 0$, by Lemma 7.3.3 in [2] and Step 7, and note that the algorithm does not terminate at Step 1, then there exists an integer $J > 0$ such that $\phi(\lambda_J) \le 0$. Therefore the lemma is true. □

Lemma 4.2. *Let d be generated by Algorithm 4.1, then*

$$\Theta(d) \le -\frac{1}{4}\min\left\{\left\|F'^T F\right\| \Big/ \bar{M}, \delta\right\}\left\|F'^T F\right\|. \tag{4.10}$$

Proof. (4.10) is trivial if Algorithm 4.1 terminates at Step 5. If Algorithm 4.1 terminates at step 1, then d is an optimum of (4.1). It is similar to the proof of Lemma 2.2 that (4.10) holds. □

Now we consider Algorithm 2.1 in which the trial step is the approximate solution generated by Algorithm 4.1 at each iterative point x_k. Let $\tilde{d}_k^p$ be the trial step generated by Algorithm 4.1. From Lemma 4.2 we have

$$Pred_k(\tilde{d}_k^p) = \Theta(\tilde{d}_k^p) \le -\frac{1}{4}\min\left\{\left\|F'(x_k)^T F(x_k)\right\| \Big/ \bar{M}, \delta_k^p\right\}\left\|F'(x_k)^T F(x_k)\right\|.$$

Similar to Theorem 2.1, we have the following theorem.

Theorem 4.1. *Suppose that Assumption 2.1 holds. Let $\{x_k\}$ be generated by Algorithm 2.1 with the trial step generated by Algorithm 4.1, then*

$$\lim_{k\to\infty}\left\|F'(x_k)^T F(x_k)\right\| = 0.$$

In what follows, we analyze the superlinear convergence of Algorithm 2.1 with the trial step generated by Algorithm 4.1. To this end, assuming that Assumption 3.1 holds. First we give the following lemma, which says that the difference between the approximate solution generated by Algorithm 4.1 and the exact solution of (2.1) is sufficiently small for sufficiently large k.

Lemma 4.3. *There exists $L_1 > 0$ such that $\|d_k^0 - \tilde{d}_k^0\| \le L_1\|F(x_k)\|^{2\gamma}$, for all k sufficiently large, where d_k^0 is the exact solution of (2.1) corresponding to $p = 0$.*

Proof. If algorithm stops at Step 1, we have $\tilde{d}_k^0 = d_k^0$. The conclusion is trivial. If Algorithm 4.1 stops at Step 5, then $\|d_k^0\| = \delta_k^0$ and the corresponding Lagrange multiplier $\lambda^M > 0$. Hence $F'(x_k)^T F'(x_k) + \lambda_k^M I$ (where I is $n \times n$ unit matrix) is positive definite. From Corollary 5.1 and the statement following it in [4], there exist $\epsilon^0 > 0$ and $L_1 > 0$ such that for all Δ satisfying $\|\Delta - \delta_k^0\| < \epsilon^0$ the solution d_s of the following problem

$$\begin{aligned} \min_{d\in R^n} \quad & \tfrac{1}{2} d^T F'^T F' d + F^T F' d \\ \text{s.t.} \quad & \|d\| \le \Delta \end{aligned}$$

satisfies $\|d_s - d_k^0\| \le L_1\|\Delta - \delta_k^0|$.

Note that $\tilde{d}_k^0$ is the solution of the following problem

$$\begin{aligned} \min_{d\in R^n} \quad & \tfrac{1}{2} d^T F'^T F' d + F^T F' d \\ \text{s.t.} \quad & \|d\| \le \|\tilde{d}_k^0\|. \end{aligned}$$

By Assumption 3.1, $\|F(x_k)\|^{2\gamma} < \epsilon^0$ for sufficiently large k. Thus by Step 5 in Algorithm 4.1 $|\|\tilde{d}_k^0\| - \delta_k^0| \le \|F(x_k)\|^{2\gamma} < \epsilon^0$. Therefore,

$$\|\tilde{d}_k^0 - d_k^0\| \le L_1|\|\tilde{d}_k^0\| - \delta_k^0| \le L_1\|F(x_k)\|^{2\gamma}$$

for sufficiently large k. □

Lemma 4.4. *Suppose that Assumption 3.1 holds, then*

$$\|\tilde{d}_k^0\| = O(\|\hat{x}_k - x_k\|^{\gamma}), \tag{4.11}$$

$$\|F(x_k) + F'(x_k)\tilde{d}_k^0\| = O(\|\hat{x}_k - x_k\|^{2\gamma}), \tag{4.12}$$

$$dist(x + \tilde{d}_k^0, X^*) = O((dist(x_k, X^*))^{2\gamma}), \tag{4.13}$$

where $\hat{x}_k$ is defined as in Section 3.

Proof. By Assumption 3.1 and (3.2), we know that for sufficiently large k

$$\|F(x_k)\| = O(\|\hat{x}_k - x_k\|). \tag{4.14}$$

If Algorithm 4.1 stops at Step 1 at the k-th iterative point x_k, then $\tilde{d}_k^0 = d_k^0$. By (3.7) (3.8) and (3.9) the conclusions hold.

If Algorithm 4.1 stops at Step 5 at the k-th iterative point x_k, by Lemma 4.3, (4.14) and (3.8), we have for sufficiently large k

$$\begin{aligned}\|\tilde{d}_k^0\| &\le \|d_k^0\| + L_1\|F(x_k)\|^{2\gamma} \\ &= O(\|x_k - \hat{x}_k\|^{\gamma}) + O(\|x_k - \hat{x}_k\|^{2\gamma}) \\ &= O(\|x_k - \hat{x}_k\|^{\gamma}).\end{aligned}$$

Then (4.11) holds.

It follows from Lemma 4.3, (3.7) and (4.14) that

$$\begin{aligned}&\|F(x_k) + F'(x_k)\tilde{d}_k^0\|^2 \\ &= \|F(x_k) + F'(x_k)d_k^0\|^2 + (F(x_k) + F'(x_k)d_k^0)^T F'(x_k)(d_k^0 - \tilde{d}_k^0) \\ &\quad + (d_k^0 - \tilde{d}_k^0)^T F'(x_k)^T F'(x_k)(d_k^0 - \tilde{d}_k^0) \\ &= O(\|x_k - \hat{x}_k\|^4) + O(\|x_k - \hat{x}_k\|^{2+2\gamma}) + O(\|x_k - \hat{x}_k\|^{4\gamma}) \\ &= O(\|x_k - \hat{x}_k\|^{4\gamma}).\end{aligned}$$

Then (4.12) holds.

Since $x_k \to x^*$ and $\tilde{d}_k^0 \to 0$, then $x_k + \tilde{d}_k^0 \in N(x^*, b)$ for sufficiently large k. By Assumption 3.1 (iii), (4.11) and (4.12), we have

$$\begin{aligned}c_2 \cdot dist(x_k + \tilde{d}_k^0, X^*) &\le \|F(x_k + \tilde{d}_k^0)\| \\ &\le \|F(x_k) + F'(x_k)\tilde{d}_k^0\| + O(\|\tilde{d}_k^0\|^2) \\ &= O(\|x_k - \hat{x}_k\|^{2\gamma}) \\ &= O((dist(x_k, X^*))^{2\gamma}).\end{aligned}$$

Then (4.13) holds. □

Similar to (3.11) and (3.12), we can prove that

$$|Ared_k(\tilde{d}_k^0) - Pred_k(\tilde{d}_k^0)| = O(\|x_k - \hat{x}_k\|^{4\gamma}) \tag{4.15}$$

and

$$|Pred_k(\tilde{d}_k^0)| = O(\|x_k - \hat{x}_k\|^2). \tag{4.16}$$

(4.15), (4.16) and $\gamma > 0.5$ imply that

$$r_k(\tilde{d}_k^0) > \eta \tag{4.17}$$

for k sufficiently large. So for sufficiently large k the iteration formula is as follows

$$x_{k+1} = x_k + \tilde{d}_k^0. \tag{4.18}$$

The following theorem follows from (4.18) and Lemma 4.4.

Theorem 4.2. *Suppose that Assumption* 3.1 *holds. The sequence* $\{x_k\}$ *is generated by Algorithm* 2.1 *with the trial step* $\tilde{d}_k^p$ *generated by Algorithm* 4.1. *Then for sufficiently large* k *the iteration formula is as follows*

$$x_{k+1} = x_k + \tilde{d}_k^0$$

and

$$dist(x_{k+1}, X^*) = O((dist(x_k, X^*))^{2\gamma}),$$

i.e., Algorithm 2.1 is superlinearly convergent.

5 Numerical experiments

In order to see the efficiency of our method, the new trust region method is tested on some classical problems. The algorithm is implemented in Fortran90, and run in Compaq Visual Fortran 6.1 environment in PC. The test problems are created by modifying the problems given in [8] and have the same form as in [10]. The subroutine solving trust region subproblem was provided by Jorge J. Moré. The parameters are set as follows: $\epsilon = 10^{-8}$, $\eta = 10^{-6}$, $c = 0.5$.

First, we test our algorithm with different initial points. The three rows of every problem of the Table 5.1 indicates that the initial point are x_0, $10 \cdot x_0$ and $100 \cdot x_0$ respectively, where x_0 is suggested by Moré, Garbow and Hillstrom in [8]. And $\gamma = 0.6$. "iters" denotes the number of solving the trust region subproblem. $\frac{1}{2}\|F(x^*)\|^2$ is the half of the square of the 2-norm of the value of the function at the final iteration. Meanwhile, we compare the results obtained by our new algorithm with that obtained by the traditional trust region method. The parameters used in the traditional trust region methods are as follows: $\epsilon = 10^{-8}$, $\eta = 10^{-6}$, $\Delta_0 = 1$, $\beta_1 = 0.25$, $\beta_2 = 0.75$. And the trust region radius is updated as follows:

$$\Delta_{k+1} = \begin{cases} 0.5\Delta_k, & r_k \le \beta_1; \\ \Delta_k, & \beta_1 < r_k \le \beta_2; \\ 2\Delta_k, & r_k > \beta_2, \end{cases}$$

where ϵ is the stop criteria and η is the accept criteria. The results are summarized in Table 5.1.

From Table 5.1, we can see that our algorithm can solve these nonlinear equation systems from any initial point. However, the efficiency of the algorithm depends on different initial points. Generally speaking, the farther the initial point is from the solution, the worse the algorithm performs. But, for ROSENBROCK, POWELL SINGULAR, POWELL BADLY SCALED, HELICAL VALLEY, WATSON, DISCRETE INTEGRAL EQUATION, BROYDEN TRIDIAGONAL and BROYDEN BANDED, our algorithm is insensitive to the initial point. From Table 5.1, we also find that in most cases our new algorithm is more efficient than the traditional trust region method. However, for some problems (for example, CHEBYQUAD and VARIABLY DIMENSIONED), the traditional algorithm is robust. In summary, the results in Table 5.1 show that the performance of our new algorithm is notable.

Table 5.1. Numerical results with different initial points

Function	n	x_0	New algorithm		Traditional algorithm	
			iters	$\frac{1}{2}\|F(x^*)\|^2$	iters	$\frac{1}{2}\|F(x^*)\|^2$
Rosenbrock	2	1	15	0.44–16	23	0.22–17
		10	3	0.78–15	44	0.11–16
Powell singular	4	1	11	0.31–7	11	0.65–7
		10	14	0.47–7	54	0.24–7
		100	18	0.19–7	494	0.62–7
Powell badly scaled	2	1	731	0.46–9	104	0.71–8
		10	4	0.99–8	4	0.99–8
		100	3	0.10–5	9	0.10–5
Wood	4	1	31	0.74–9	45	0.47–12
		10	162	0.17–10	58	0.40–11
		100	480	0.13–10	568	0.12–13
Helical valley	3	1	10	0.65–14	9	0.44–14
		10	12	0.69–17	19	0.15–18
		100	12	0.31–10	108	0.13–11
Watson	30	1	557	0.23–6	2195	0.26–7
		10	557	0.23–6	2195	0.26–7
		100	557	0.23–6	2195	0.26–7
Chebyquad	30	1	289	0.82–3	316	0.82–3
		10	8934	0.11	1253	0.15
		100	–	–	2185	0.011
Brown almost linear	30	1	24	0.73–13	10	0.15–13
		10	497	1.13	47	0.20–12
		100	1260	0.99	293	0.52–13
Discrete boundary value	30	1	19	0.59–8	2	0.63–10
		10	83	0.16–11	11	0.41–9
		100	163	0.25–11	105	0.36–8
Discrete integral equation	30	1	3	0.98–16	3	0.98–16
		10	6	0.74–14	12	0.21–14
		100	11	0.12–11	101	0.83–10
Trigonometric	30	1	85	0.99–5	149	0.99–5
		10	111	0.76–4	196	0.76–4
		100	124	0.17–3	23	0.62–10
Variably dimensioned	30	1	519	0.17–10	20	0.26–9
		10	1173	0.25–10	36	0.57–11
		100	–	–	413	0.80–11
Broyden tridiagonal	30	1	4	0.11–10	5	0.68–16
		10	8	0.58–16	54	0.33–15
		100	11	0.61–13	547	0.16–15
Broyden banded	30	1	5	0.15–9	6	0.11–13
		10	11	0.57–13	55	0.80–11
		100	17	0.40–16	548	0.25–11

Then we test our algorithm with different choices of γ. We choose $\gamma = 0.5$, 0.8, 1, 1.5 respectively. And the initial point is set as the standard initial point suggested in [8]. We only record the number of solving the trust region subproblem. The results are summarized in Table 2. From Table 5.2, it is noted that the algorithm performs bad if $\gamma > 1$. For $\gamma \in [0.5, 1]$, the algorithm performs good for some problems but bad for the rest when the γ value is near to 1. Generally speaking, it is proper to choose $\gamma \in [0.6, 0.8]$.

Table 5.2. Numerical results with different choice of γ

	n	$\gamma = 0.5$	$\gamma = 0.8$	$\gamma = 1.0$	$\gamma = 1.5$
Rosenbrock	2	21	26	25	9459
Powell singular	4	11	43	268	37102
Powell badly scaled	2	260	5281	22926	126205
Wood	4	19	18	15	20393
Helical valley	3	11	11	12	359
Watson	30	351	442	947	11080
Chebyquad	30	333	331	238	218
Brown almost linear	30	17	1	2	9
Discrete boundary value	30	9	93	489	52995
Discrete integral equation	30	3	3	3	4050
Trigonometric	30	127	39	61	267
Variably dimensioned	30	464	174	25	105
Broyden tridiagonal	30	4	4	4	1533
Broyden banded	30	5	5	5	880

6 Conclusion

In this paper, we present a new trust region method which utilize $\|F(x_k)\|$ to determine the trust region radius. The algorithm is globally convergent under mild conditions. Furthermore, we have proved that the method is superlinearly convergent under the condition that $\|F(x)\|$ provides a local error bound for system (1.1) rather than the nondegeneracy of Jacobian of $F(x)$ at the solution. This condition is weaker than the nondegeneracy assumption used in the traditional trust region methods.

When realizing the trust region method, the main cost is to compute the trial step. In order to ensure the trust region method having good convergence properties and costing less computation, it is reasonable to recognize that the difference between the trial step and the exact solution of trust region subproblem should be large if the iterative point is far from the solution of (1.1) while the difference between them should become small as the iterative sequence approaches to the solution. The approximate method presented in Section 4 has this property. Moreover, we have proved that the new trust region method with the trial step generated by the approximate algorithm is globally and superlinearly convergent under the local error bound condition. This makes the algorithm more practicable.

Acknowledgements. The authors are indebted to our supervisor, Professor Y.-X. Yuan, for his excellent guidance and Jorge J. Moré for his subroutine. And we would like to thank the referees for their valuable suggestions and comments.

References

1. Bertsekas DP (1995) Nonlinear Programming, Belmont Athena Scientific
2. Conn AR, Gould NIM, Toint PL (2000) Trust Region Method, Philadelphia, PA: Society for Industrial and Applied Mathematics
3. Fletcher R (1987) Practical Methods of Optimizations. New York: John Wiley & Sons
4. Levy AB (2001) Lipschitzian multifunctions and a Lipschitzian inverse mapping theorem. Mathematics of Operations Research 26:105–118

5. Levenberg K (1944) A method for the solution of certain nonlinear problem in least squares. Quart. Appl. Math. 2:164–166
6. Marquardt DW (1963) An algorithm for least-squares estimation of nonlinear inequalities. SIAM J. Appl. Math. 11:431–441
7. Moré JJ (1983) Recent developments in algorithm and software for trust region methods. In: Bachem A, Grötschel M, Kort B (eds). Mathematical Programming: the state of art, Springer, Berlin, pp. 258–287
8. Moré JJ, Garbow BS, Hillstrom KH (1981) Testing unconstrained optimization software. ACM Transaction on Mathematical Software 7:17–41
9. Nocedal J, Wright SJ (1999) Numerical Optimization. New York: Springer
10. Schnabel RB, Frank PD (1984) Tensor method for nonlinear equation. SIAM J. Numerical Analysis, 21:815–843
11. Yamashita N, Fukushima M (2001) On the rate of convergence of the Levenberg-Marquardt method. Computing 15:239–249
12. Yuan Y, Sun W (1997) Optimization Theory and Algorithm. Scientific Publisher House, China