Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 15, No. 2, pp. 317 – 329, 2019

A NEW TRUST REGION METHOD FOR NONLINEAR EQUATIONS INVOLVING FRACTIONAL MODEL*

Meilan Zeng† and Qin Ni

Abstract: In this paper, we propose a new trust region method for solving nonlinear equations, which bases on a fractional model of the nonlinear function. The global and quadratic convergence are obtained under suitable conditions. Our method possesses better approximation property than Newton method. The preliminary numerical results illustrate that our method is more efficient than the trust region Newton method for some nonlinear equations.

Key words: *nonlinear equations, fractional model, trust region, quadratic convergence*

Mathematics Subject Classification: *65H10*

1 Introduction

We consider the numerical solution of system of nonlinear equations

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

where $F : R^n \to R^n$ is a nonlinear mapping, $F(x) = (f_1(x), \ldots, f_n(x))^T$.

In this paper we introduce a new trust region method for solving the nonlinear equations (1.1). The feature of the method is that it bases on a fractional model.

The most famous iterative method for solving (1.1) is Newton method, which bases each iteration upon a linear model of the function $F(x)$ around the current iterate $x_k \in R^n$:

$$M_N(x_k + d) = F(x_k) + J(x_k)d, \tag{1.2}$$

where $d \in R^n$ is the step and $J(x)$ is Jacobian matrix of F at x,

$$J(x) = (\nabla f_1(x), \ldots, \nabla f_n(x))^T.$$

The distinguishing feature of this method is that if $J(x)$ is Lipschitz continuous in a neighborhood containing the root x^*, and $J(x^*)$ is nonsingular, then the sequence of iterates produced by (1.2) converges locally and quadratically to x^*.

Global methods for solving (1.1) are divided into two classes: line search and trust region [1, 2, 4–7, 9, 10, 15, 19]. Both line search and trust region strategies preserve the local

*This work was supported by National Natural Science Foundation of China (11771210, 11601138).
†Corresponding author.

convergence properties of the parent method. In this paper, we intend to extend the second type of methods.

Let $f(x) = \frac{1}{2}\|F(x)\|^2$. Suppose that $F(x)$ has a zero point, then the nonlinear equation problem (1.1) is equivalent to the global optimization problem

$$\min f(x), x \in R^n. \tag{1.3}$$

For the traditional trust region methods, at each iterative point x_k, the trial step d_k is obtained by solving the following trust region subproblem

$$\min q_k(d) \quad \text{such that} \quad \|d\| \leq \Delta_k, \tag{1.4}$$

where $q_k(d) = \frac{1}{2}\|F_k + J_k d\|^2, F_k = F(x_k), J_k = J(x_k)$. The trust region methods are globally and superlinearly convergent under the condition that $J(x^*)$ is nonsingular. However, if the curvatures of the functions change dramatically, the trust region Newton methods are not suitable.

The objective of this paper is to propose a new trust region method for solving (1.1), which is based on a fractional model. This method may possess better approximate property. It is globally and quadratically convergent. The preliminary numerical results have shown that our method is more efficient than the trust region Newton method for some nonlinear equations.

The paper is organized as follows. The new trust region method based on a fractional model and its property are presented in Section 2. In Section 3, we show the convergence theory of the method. Section 4 demonstrates preliminary numerical results on test problems.

2 New Trust Region Method Based on Fractional Model and its Property

In this section, we propose a new trust region method for solving nonlinear equations. The method bases on a fractional model of $F(x)$ that has the form

$$M_F(x_k + d) = F_k + \frac{J_k d}{1 - a_k^T d}, \tag{2.1}$$

where J_k is Jacobian matrix of F at x_k, the level vector a_k satisfies

$$\|a_k\| \leq \frac{1 - \epsilon_0}{\Delta_k}, \tag{2.2}$$

for $\epsilon_0 \in (0, 1), \Delta_k > 0$, which implies that

$$\epsilon_0 \leq 1 - \|a_k\|\|d\| \leq 1 - a_k^T d \leq 1 + \|a_k\|\|d\| \leq 2 - \epsilon_0 \tag{2.3}$$

always holds for $\|d\| \leq \Delta_k$.

Model (2.1) for solving (1.1) is inspired by the conic model for unconstrained optimization (see [3, 16]). If $a_k = 0$, then (2.1) reduces to the well-known Newton method.

Let $M_F(x_k + d) = 0$, then

$$[J_k - F_k a_k^T]d = -F_k. \tag{2.4}$$

By the definition of f, we have

$$\nabla f(x) = (J(x))^T F(x),\ \nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^{n} f_i(x) \nabla^2 f_i(x). \tag{2.5}$$

Based on the above process, we propose a new algorithm as follows.

Algorithm 2.1.

Step 0. Choose $0 < \eta_1 < \eta_2 < 1$, $0 < \gamma_1 < 1 < \gamma_2$, $\bar{\Delta}$ and $\varepsilon \geq 0$. Initialize $x_0, a_0, 0 < \Delta_0 < \bar{\Delta}$ (generally $a_0 = 0$). Set $k = 0$.

Step 1. Evaluate $F_k, J_k, f_k = f(x_k)$, if $\|F_k\| \leq \varepsilon$, terminate.

Step 2. Solve

$$\begin{aligned} \min \quad & q_k(d) = \frac{1}{2}\|M_F(x_k + d)\|_2^2 \\ \text{s.t.} \quad & \|d\| \leq \Delta_k, \end{aligned} \tag{2.6}$$

obtain d_k.

Step 3. Compute

$$\theta_k = \frac{Ared_k}{Pred_k}, \tag{2.7}$$

where $Ared_k = f_k - f(x_k + d_k)$, $Pred_k = \frac{1}{2}\left(\|F_k\|^2 - \|M_F(x_k + d_k)\|^2\right)$. If $\theta_k < \eta_1$, then $\Delta_k = \gamma_1 \Delta_k$, go to Step 2. Otherwise, go to Step 4.

Step 4. $x_{k+1} = x_k + d_k$;

$$\Delta_{k+1} = \begin{cases} \gamma_2 \Delta_k, & \text{if } \theta_k \geq \eta_2, \\ \Delta_k, & \text{otherwise.} \end{cases}$$

Step 5. Update a_{k+1} by

$$a_{k+1} = \frac{\eta_k - \xi_k}{\xi_k \|d_k\|^2} d_k, \tag{2.8}$$

where

$$\xi_k = d_k^T(F_{k+1} - F_k), \ \eta_k = d_k^T J_{k+1} d_k. \tag{2.9}$$

If (2.2) is not satisfied, then compute

$$a_{k+1} := \frac{1 - \epsilon_0}{\Delta_k \|a_{k+1}\|} a_{k+1}.$$

Set $k := k + 1$, and go to Step 1.

Remark. According to Lemma 3.1 in Section 3, we can solve a equivalent trust region subproblem in Step 2. Hence, a lot of methods (see [17]) for solving general trust region subproblem can be used to solve the equivalent problem of (2.6).

From (1.4) and (2.1) it follows that

$$\begin{aligned} M_N(x_k + 0) &= F_k, \ \nabla_d M_N(x_k + 0) = J_k, \\ M_F(x_k + 0) &= F_k, \ \nabla_d M_F(x_k + 0) = J_k. \end{aligned}$$

In addition, $M_F(x_k)$ possesses the following property which is a motivation of Algorithm 2.1.

Lemma 2.1. *If a_{k+1} defined in (2.8) satisfies (2.2), then $M_F(x_k)$ satisfies*

$$d_k^T(F(x_k) - M_F(x_k)) = 0. \tag{2.10}$$

Proof. From (2.1), (2.8) and (2.9), it follows that

$$\begin{aligned} M_F(x_k) &= M_F(x_{k+1}-d_k) = F_{k+1} - \frac{J_{k+1}d_k}{1+a_{k+1}^T d_k} \\ &= F_{k+1} - J_{k+1}d_k\frac{\xi_k}{\eta_k}. \end{aligned}$$

Hence, we have

$$\begin{aligned} & d_k^T(F(x_k) - M_F(x_k)) \\ = \;& d_k^T(F_k - F_{k+1}) + d_k^T J_k d_k \frac{\xi_k}{\eta_k} \\ = \;& -\xi_k + \eta_k\frac{\xi_k}{\eta_k} = 0. \end{aligned}$$

The proof of this lemma is complete. □

However, $M_N(x_k)$ in (1.4) does not satisfy (2.10). This means that our method may have better approximation of $F(x_k)$ along direction d_k than Newton method.

3 Convergence Analysis

In order to prove the convergence of Algorithm 2.1, we make the following assumptions.

Assumption 3.1 (1) $F(x)$ is twice continuously differentiable.

(2) The level set $\Omega = \{x \in R^n | f(x) \le f(x_0)\}$ is bounded.

By Assumption 3.1(1) there exists a positive constant κ_J such that

$$\|J(x)^T J(x)\| \le \kappa_J, \text{ for } x \in \Omega.$$

The solution of (2.6) is determined by the following lemma.

Lemma 3.1. *d_k is a solution of (2.6) if and only if there exists $\mu_k \ge 0$ such that $J_k^T J_k + \mu_k(I - \Delta_k^2 a_k a_k^T)$ is semipositive definite, and*

$$(J_k^T J_k - J_k^T F_k a_k^T + \mu_k I)d_k = -J_k^T F_k + \mu_k \Delta_k^2 a_k, \tag{3.1}$$

$$\mu_k(\|d_k\| - \Delta_k) = 0, \quad \|d_k\| \le \Delta_k. \tag{3.2}$$

Proof. This lemma follows from Theorem 4.1 in [12]. □

Lemma 3.2. $|Ared_k - Pred_k| = O(\|d_k\|^2) = O(\Delta_k^2)$.

Proof. From the definitions of $Ared_k$ and $Pred_k$, we get

$$\begin{aligned} |Ared_k - Pred_k| &= |q_k(d_k) - f(x_k+d_k)| \\ &= \frac{1}{2}\left|\|F(x_k) + J_k\frac{d_k}{1-a_k^T d_k}\|^2 - \|F(x_k) + J_k d_k + O(\|d_k\|^2)\|^2\right| \\ &= \left|\frac{a_k^T d_k}{1-a_k^T d_k}F(x_k)^T J_k d_k + O(\|d_k\|^2) + O(\|d_k\|^4)\right| \\ &= O(\|d_k\|^2) = O(\Delta_k^2). \end{aligned}$$

This completes the proof. □

Lemma 3.3. *Let Assumption 3.1 hold, and d_k is a solution of (2.6). If $J_k^T F_k \neq 0$, then*

$$Pred_k \geq \frac{1}{2(2-\epsilon_0)^2}\|J_k^T F_k\| \min\left\{\epsilon_0^2 \frac{\|J_k^T F_k\|}{\|J_k^T J_k\|}, (2-\epsilon_0)\Delta_k\right\}. \tag{3.3}$$

Proof. Define $d_k(\tau) = -\tau J_k^T F_k$, then $d_k(\tau)$ is feasible for (2.6), and

$$Pred_k = \frac{1}{2}\left(\|F_k\|^2 - \|M_F(x_k + d_k)\|^2\right) \geq \frac{1}{2}\left(\|F_k\|^2 - \|M_F(x_k + d_k(\tau))\|^2\right) \tag{3.4}$$

for all $\tau \in [0, \frac{\Delta_k}{\|J_k^T F_k\|}]$. Hence by (2.1) we have

$$\begin{aligned}
\frac{1}{2}\left(\|F_k\|^2 - \|M_F(x_k + d_k(\tau))\|^2\right) &= \tau \frac{\|J_k^T F_k\|^2}{1 - a_k^T d_k(\tau)} - \frac{\tau^2}{2}\frac{\|J_k J_k^T F_k\|^2}{(1 - a_k^T d_k(\tau))^2} \\
&\geq \tau \frac{\|J_k^T F_k\|^2}{2-\epsilon_0} - \frac{\tau^2}{2\epsilon_0^2}\|J_k^T J_k\|\|J_k^T F_k\|^2 \\
&= \|J_k^T F_k\|^2 \left[\frac{\tau}{2-\epsilon_0} - \frac{\tau^2}{2\epsilon_0^2}\|J_k^T J_k\|\right] \\
&\geq \|J_k^T F_k\|^2 \max_{0 \leq \tau \leq \frac{\Delta_k}{\|J_k^T F_k\|}} \left[\frac{\tau}{2-\epsilon_0} - \frac{\tau^2}{2\epsilon_0^2}\|J_k^T J_k\|\right] \\
&\geq \|J_k^T F_k\|^2 \cdot \frac{1}{2(2-\epsilon_0)^2} \min\left\{\frac{\epsilon_0^2}{\|J_k^T J_k\|}, \frac{(2-\epsilon_0)\Delta_k}{\|J_k^T F_k\|}\right\} \\
&= \frac{1}{2(2-\epsilon_0)^2}\|J_k^T F_k\| \min\left\{\epsilon_0^2 \frac{\|J_k^T F_k\|}{\|J_k^T J_k\|}, (2-\epsilon_0)\Delta_k\right\},
\end{aligned}$$

for all $\tau \in [0, \frac{\Delta_k}{\|J_k^T F_k\|}]$. □

Lemma 3.4. *Algorithm 2.1 does not circle between Steps 2 and 3 infinitely.*

Proof. If Algorithm 2.1 circles between Steps 2 and 3 infinitely, then for all $i = 1, 2, \ldots,$ we have $x_{k+i} = x_k$, and $\|F_k\| > \varepsilon$ which implies that $\theta_k < \eta_1, \Delta_k \to 0$. From Lemmas 3.2 and 3.3 it follows that

$$|\theta_k - 1| = \frac{|Ared_k(d_k) - Pred_k(d_k)|}{|Pred_k(d_k)|} \leq \frac{2(2-\epsilon_0)O(\Delta_k^2)}{\Delta_k\|J_k^T F_k\|}. \tag{3.5}$$

Hence, for k sufficiently large

$$\theta_k \geq \eta_1, \tag{3.6}$$

which contradicts the fact $\theta_k < \eta_1$. □

Theorem 3.5. *Let Assumption 3.1 hold and $\{x_k\}$ be generated by Algorithm 2.1. Then either there exists some finite k_0 such that $J_{k_0}^T F_{k_0} = 0$ or*

$$\lim_{k\to\infty} \inf \|J_k^T F_k\| = 0. \tag{3.7}$$

Proof. If the theorem is not true, then there exists a positive constant κ_g and an infinite subsequence $\{k_i\}$ such that

$$\|J_k^T F_k\| \geq \kappa_g,$$

which, together with Lemma 3.3, implies that

$$Pred_k \geq \frac{1}{2(2-\epsilon_0)^2}\kappa_g \min\{\kappa_1, (2-\epsilon_0)\Delta_k\}, \tag{3.8}$$

where $\kappa_1 = \frac{\epsilon_0^2 \kappa_g}{\kappa_J}$. Let $K = \{k | \|J_k^T F_k\| \geq \kappa_g\}$.

Let $S_0 = \{k | \theta_k \geq \eta_2\}$. Then from (3.8) and Assumption 2.1(2) we obtain

$$\sum_{k\in S_0}[f(x_k) - f(x_{k+1})] \geq \sum_{k\in S_0}\eta_2 \cdot Pred_k \geq \sum_{k\in S_0}\eta_2 \cdot \frac{1}{2(2-\epsilon_0)^2}\kappa_g \min\{\kappa_1, (2-\epsilon_0)\Delta_k\}. \tag{3.9}$$

Since $\{f(x_k)\}$ is convergent, we have

$$\sum_{k\in S_0}\Delta_k < \infty. \tag{3.10}$$

From Steps 3-4 of Algorithm 2.1 it follows that

$$\Delta_{k+1} \leq \Delta_k \tag{3.11}$$

for all $k \notin S_0$, thus (3.10) means

$$\sum_{k\in K}\Delta_k < \infty. \tag{3.12}$$

Therefore there exists x^* such that

$$\lim_{k\to\infty} x_k = x^*. \tag{3.13}$$

By (3.12), we have $\Delta_k \to 0$, which implies

$$Pred_k \geq \frac{\kappa_g}{2(2-\epsilon_0)}\Delta_k$$

for all sufficiently large k. The fact that $|Ared_k - Pred_k| = O(\Delta_k^2)$ indicates that

$$\lim_{k\to\infty}\theta_k = 1,$$

which yields that, for sufficiently large k and $k \in K$,

$$\Delta_{k+1} \geq \Delta_k.$$

The above inequality contradicts (3.11). The contradiction proves the theorem. □

Remark. Theorem 3.5 shows that the iterative sequence $\{x_k\}$ generated by Algorithm 2.1 satisfies $\|J_k^T F_k\| \to 0$. If x^* is a cluster point of $\{x_k\}$ and $J(x^*)$ is nonsingular, then we have $\|F_k\| \to 0$.

In order to discuss the quadratic convergence of Algorithm 2.1, we give two lemmas in [14] and [2].

Lemma 3.6 ([14]). *Suppose that $A, C \in R^{n\times n}$, A is nonsingular, $\|A^{-1}\| \leq \lambda$, $\|A-C\| \leq \mu$, and $\lambda\mu < 1$, then C is nonsingular, and*

$$\|C^{-1}\| \leq \frac{\lambda}{1-\mu\lambda}.$$

Lemma 3.7 ([2]). *Let $D \subset R^n$ be an open convex set, $F : R^n \to R^n$ be differentiable. If any $x, y \in D$, there exists $L > 0$, such that $\|J(y) - J(x)\| \le L\|y - x\|$, then*

$$\|F(y) - F(x) - J(x)(y - x)\| \le \frac{L}{2}\|y - x\|^2.$$

The following lemma means that the solution of the subproblem in Step 2 is close to Newton direction when the number of iterations is sufficiently large.

Lemma 3.8. *Let the conditions of Theorem 3.5 hold, d_k be generated by Algorithm 2.1. If $\{x_k\}$ converges to x^* and*

$$\|a_k\| \le \frac{\|d_k\|}{\Delta_k^2}$$

for all k, then there exists a positive integer K such that

$$d_k = -(J_k - F_k a_k^T)^{-1} F_k, \tag{3.14}$$

for all $k \ge K$.

Proof. Because $\{x_k\}$ converges to x^*, $J(x^*)^T J(x^*)$ is positive definite, from the continuity of $J(x)^T J(x)$ there exist two positive numbers $M \ge m > 0$ such that

$$m\|d\|^2 \le d^T J_k^T J_k d \le M\|d\|^2,$$

for all $d \ne 0$ and all k sufficiently large. From (3.1), it follows that

$$-d_k^T J_k^T F_k(1 - a_k^T d_k) = d_k^T J_k^T J_k d_k + \mu_k d_k^T d_k - \mu_k \Delta_k^2 a_k^T d_k.$$

If $\|a_k\| \le \frac{\|d_k\|}{\Delta_k^2}$, we have

$$\begin{aligned} \Delta q_k &= -\frac{F_k^T J_k d_k}{1 - a_k^T d_k} - \frac{1}{2}\frac{d_k^T J_k^T J_k d_k}{(1 - a_k^T d_k)^2} \\ &= \frac{1}{2}\frac{d_k^T J_k^T J_k d_k}{(1 - a_k^T d_k)^2} + \frac{\mu_k(\|d_k\|^2 - \Delta_k^2 a_k^T d_k)}{(1 - a_k^T d_k)^2} \\ &\ge \frac{m}{2}\|d_k\|^2, \end{aligned}$$

$$\begin{aligned} \Delta f_k - \Delta q_k &= -F_k^T J_k d_k - \frac{1}{2} d_k^T \nabla^2 f(x_k) d_k + o(\|d_k\|^2) + \frac{F_k^T J_k d_k}{1 - a_k^T d_k} + \frac{1}{2}\frac{d_k^T J_k^T J_k d_k}{(1 - a_k^T d_k)^2} \\ &= \frac{F_k^T J_k d_k a_k^T d_k}{1 - a_k^T d_k} + \frac{d_k^T J_k^T J_k d_k [2a_k^T d_k - (a_k^T d_k)^2]}{2(1 - a_k^T d_k)^2} \\ &\quad -\frac{1}{2}\sum_{i=1}^n f_i(x_k) d_k^T \nabla^2 f_i(x_k) d_k + o(\|d_k\|^2) \\ &= o(\|d_k\|^2). \end{aligned}$$

Hence, we obtain

$$\left|\frac{\Delta f_k}{\Delta q_k} - 1\right| \le \frac{o(\|d_k\|^2)}{\frac{m}{2}\|d_k\|^2},$$

which implies that $\frac{\Delta f_k}{\Delta q_k} \to 1$.

Because $\{x_k\}$ converges to x^*, and $F(x^*) = 0$, then there exists a K_1 such that $J_k^T J_k - J_k^T F_k a_k^T$ is nonsingular and

$$\|(J_k^T J_k - J_k^T F_k a_k^T)^{-1} J_k^T F_k\| \le \tilde{\Delta} \le \Delta_k$$

holds for all $k \ge K_1$. In this case, we have

$$d_k = -(J_k^T J_k - J_k^T F_k a_k^T)^{-1} J_k^T F_k.$$

By $J(x^*)$ is nonsingular and Lemma 3.6, there exists $K_2 \ge K_1$ such that J_k^T is nonsingular for $k \ge K_2$. This implies (3.14) and completes the proof. □

Theorem 3.9. *Let the conditions of Theorem 3.8 hold. Suppose further that $J(x)$ is Lipschitz continuous. Then $\{x_k\}$ is quadratically convergent.*

Proof. Let $A(x) = J(x) - F(x)a^T$, then $A(x^*) = J(x^*)$. $F(x)$ is continuously differentiable in a neighborhood of x^*, so $A(x)$ is continuous in a neighborhood of x^*. Let $\beta = \|J(x^*)^{-1}\|$. For any $0 < \epsilon < \frac{1}{6\beta}$, there exists $K_3 \ge K_2$ such that

$$\|A(x_k) - A(x^*)\| \le \epsilon \tag{3.15}$$

$$\|F(x_k) - F(x^*)\| \le \epsilon \tag{3.16}$$

$$\|J_k - J(x^*)\| \le \epsilon \tag{3.17}$$

$$\|F(x_k) - F(x^*) - J(x^*)(x_k - x^*)\| \le \epsilon \|x_k - x^*\| \tag{3.18}$$

if $k \ge K_3$.

According to (3.15) and Lemma 3.6, $A(x_k)$ is nonsingular and

$$\|A(x_k)^{-1}\| \le \frac{\beta}{1 - \epsilon\beta} \le \frac{6}{5}\beta, \tag{3.19}$$

for $k \ge K_3$. Therefore, we have

$$\begin{aligned}
\|x_k + d_k - x^*\| &= \|x_k - A(x_k)^{-1}F(x_k) - x^*\| \\
&= \| - A(x_k)^{-1}[F(x_k) - J_k(x_k - x^*) + F(x_k)a_k^T(x_k - x^*)]\| \\
&\le \frac{6}{5}\beta\|F(x_k) - J(x^*)(x_k - x^*) - (J_k - J(x^*))(x_k - x^*) \\
&\quad +F(x_k)a_k^T(x_k - x^*)\| \\
&\le \frac{6}{5}\beta[\|F(x_k) - F(x^*) - J(x^*)(x_k - x^*)\| + \|J_k - J(x^*)\|\|x_k - x^*\| \\
&\quad +\|F(x_k) - F(x^*)\|\|a_k^T\|\|x_k - x^*\|] \\
&\le \frac{6}{5}\beta[\epsilon\|x_k - x^*\| + \epsilon\|x_k - x^*\| + \epsilon\|x_k - x^*\|] \\
&\le \frac{3}{5}\|x_k - x^*\|,
\end{aligned}$$

for $k \ge K_3$. Then the sequence $\{x_k\}$ generated by Algorithm 2.1 is well defined and converges to x^*.

By (3.18), it follows

$$\|F(x_k) - F(x^*) - J(x^*)(x_k - x^*)\| \leq \frac{1}{6\beta}\|x_k - x^*\|,$$

if $k \geq K_3$. Let $\gamma = \|J(x^*)\| + \frac{1}{6\beta}$. Since

$$F(x_k) = J(x^*)(x_k - x^*) + [F(x_k) - F(x^*) - J(x^*)(x_k - x^*)],$$

taking norms,

$$\begin{aligned} \|F(x_k)\| &\leq \|J(x^*)(x_k - x^*)\| + \|F(x_k) - F(x^*) - J(x^*)(x_k - x^*)\| \\ &\leq [\|J(x^*)\| + \frac{1}{6\beta}]\|x_k - x^*\| \leq \gamma\|x_k - x^*\| \end{aligned} \tag{3.20}$$

if $k \geq K_3$.

Because $J(x)$ is Lipschitz continuous, then there exists constant $L > 0$, such that

$$\|J_k - J(x^*)\| \leq L\|x_k - x^*\|. \tag{3.21}$$

By Lemma 3.7, we have

$$\|F(x_k) - F(x^*) - J(x^*)(x_k - x^*)\| \leq \frac{L}{2}\|x_k - x^*\|^2. \tag{3.22}$$

By (3.19), (3.21), (3.22) and (3.20), we have

$$\begin{aligned} \|x_k + d_k - x^*\| &= \|x_k - A(x_k)^{-1}F(x_k) - x^*\| \\ &\leq \frac{6}{5}\beta[\|F(x_k) - F(x^*) - J(x^*)(x_k - x^*)\| + \|J_k - J(x^*)\|\|x_k - x^*\| \\ &\quad +\|F(x_k)\|\|a_k^T\|\|x_k - x^*\|] \\ &\leq \frac{6}{5}\beta[\frac{L}{2}\|x_k - x^*\|^2 + L\|x_k - x^*\|^2 + \gamma\|x_k - x^*\|^2] \\ &\leq \frac{3}{5}\beta(3L + 2\gamma)\|x_k - x^*\|^2, \end{aligned}$$

for $k \geq K_3$.

This proves the quadratic convergence of Algorithm 2.1. □

4 Numerical Tests

In this section, we discuss numerical test results for the Algorithm 2.1 and the traditional Newton algorithm. We choose 11 test functions as follows:

Function 1 ([13]) $f_1(x) = x_1,$

$$f_2(x) = \frac{10x_1}{x_1 + 0.1} + 2x_2.$$

Initial guess: $x_0 = (3, 1)^T$.

Function 2 ([2]) $f_1(x) = x_1^2 + x_2^2 - 2,$

$$f_2(x) = e^{x_1 - 1} + x_2^3 - 2.$$

Initial guess: $x_0 = (2, 0.5)^T$.
Function 3 ([13])

$$f_1(x) = (x_1 + 3)(x_2^3 - 7) + 28,$$

$$f_2(x) = \sin(x_2 e^{x_1} - 1).$$

Initial guess: $x_0 = (-0.5, 1.4)^T$.
Function 4 Rosebrock function ([11])

$$f_1(x) = 10(x_2 - x_1^2),$$

$$f_2(x) = 1 - x_1.$$

Initial guess: $x_0 = (-1.2, 1)^T$.
Function 5

$$f_1(x) = 3x_1^2 - 2x_2 - e^{x_3},$$

$$f_2(x) = x_1 x_2 - x_3,$$

$$f_3(x) = \frac{1}{x_1} + x_2 - x_3.$$

Initial guess: $x_0 = (1, 1, 0)^T$.
Function 6 ([2])

$$f_1(x) = x_1^2 + x_2^2 + x_3^2 - 1,$$

$$f_2(x) = 2x_1^2 + x_2^2 - 4x_3,$$

$$f_3(x) = 3x_1^2 - 4x_2^2 + x_3^2.$$

Initial guess: $x_0 = (0.5, 0.5, 0.5)^T$.
Function 7

$$f_1(x) = 3x_1 - \cos(x_2 x_3) - \tfrac{1}{2},$$

$$f_2(x) = x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06,$$

$$f_3(x) = e^{-x_1 x_2} + 20x_3 + \frac{1}{3}(10\pi - 3).$$

Initial guess: $x_0 = (0.5, 0.5, 0.5)^T$.
Function 8 Powell singular function ([11])

$$f_1(x) = x_1 + 10x_2,$$

$$f_2(x) = \sqrt{5}(x_3 - x_4),$$

$$f_3(x) = (x_2 - 2x_3)^2,$$

$$f_4(x) = \sqrt{10}(x_1 - x_4)^2.$$

Initial guess: $x_0 = (3, -1, 0, 1)^T$.
Function 9 Logarithmic function ([20])

$$f_i(x) = \ln(x_i + 1) - \frac{x_i}{n}, i = 1, 2, 3, \ldots, n.$$

Initial guess: $x_0 = (1, 1, \ldots, 1)^T$.
Function 10 Brown almost linear function ([11])

$$f_i(x) = x_i + \sum_{j=1}^{n} x_j - (n + 1), 1 \le i < n,$$

$$f_n(x) = \prod_{j=1}^{n} x_j - 1.$$

Initial guess: $x_0 = (1.5, 1.5, \ldots, 1.5)^T$.
Function 11 Penalty function ([20])

$$f_i(x) = \sqrt{10^{-5}}(x_i - 1), 1 \le i < n,$$

$$f_n(x) = (\frac{1}{4n}) \sum_{j=1}^{n} x_j^2 - \frac{1}{4}.$$

Initial guess: $x_0 = (\frac{1}{3}, \frac{1}{3}, \ldots, \frac{1}{3})^T$.

In the experiments, we choose the parameters

$$\Delta_0 = 1, \epsilon_0 = 0.2, \varepsilon = 10^{-5}, \eta_1 = 0.001, \eta_2 = 0.75, \gamma_1 = 0.5, \gamma_2 = 2.0,$$

where ε is the stop criteria, and stop the program if the iteration number is larger than 1000. d_k in (2.6) is determined by Dogleg method in [8, 18], the program is coded in MATLAB 7.8. The results are summarized in Table 4.1, where Dim is the dimension of the problem, Iters is the total number of iterations, Time is the average time of iterations and measured in seconds.

Function	Dim	Algorithm 2.1			Newton algorithm		
		Iters	$\|F(x_k)\|$	Time(s)	Iters	$\|F(x_k)\|$	Time(s)
1	n=2	10	2.3534e-006	0.0058	25	2.6327e-015	0.0070
2	n=2	4	5.4746e-006	0.0034	5	6.9000e-010	0.0023
3	n=2	6	3.8802e-011	0.0021	8	7.6723e-006	0.0464
4	n=2	12	2.0960e-007	0.0082	19	2.2204e-015	0.0116
5	n=3	5	2.4408e-006	0.0028	7	6.0515e-009	0.0027
6	n=3	3	3.9171e-006	0.0019	4	2.0143e-011	0.0072
7	n=3	6	7.5617e-010	0.0034	6	1.6409e-009	0.0023
8	n=4	10	9.1895e-006	0.0066	11	3.4467e-006	0.0437
9	n=30	4	3.0986e-008	0.0051	6	3.6565e-010	0.0837
10	n=30	15	3.8521e-008	0.0707	14	9.7796e-006	0.0732
11	n=30	5	5.3536e-007	0.0079	5	4.5000e-006	0.0121

Table 4.1: Numerical results

From Table 4.1, we find that Algorithm 2.1 and Newton algorithm are efficient for solving these problems. In some cases, our algorithm is better. We hope that Algorithm 2.1 could be further developed, then it can become more efficient.

Acknowledgement

The authors are very grateful to the editors and the two referees for their valuable suggestions and constructive comments, which have considerably improved the presentation of the paper.

References

[1] J.E. Dennis and J.J. Moré, A characterization of superlinear convergence and its application to quasi-Newton methods, *Math. Comp.* 28 (1974) 1171–1190.

[2] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* Science Press, Beijing, 2009.

[3] S. Di and W. Sun, Trust region method for conic model to solve unconstained optimization, *Optimization Methods and Software* 6 (1996) 237–263.

[4] J.Y. Fan, A modified Levenberg-Marquardt algorithm for singular system of nonlinear equations, *J. Comput. Math.* 21 (2003) 625–636.

[5] A. Griewank, The 'global' convergence of Broyden-like methods with a suitable line search, *J. Austral. Math. Soc. Ser. B.* 28 (1986) 75–92.

[6] G.Z. Gu, D.H. Li, L.Q. Qi and S.Z. Zhou, Descent directions of quasi-Newton methods for symmetric nonlinear equations, *SIAM J. Numer. Anal.* 40 (2002) 1763–1774.

[7] D.H. Li and M. Fukushima, A global and superlinear convergent Gauss-Newton-based BFGS method for symmetric nonlinear equations, *SIAM J. Numer. Anal.* 37 (1999) 152–172.

[8] X.P. Lu and Q. Ni, A dogleg method for solving new trust-region subproblems of conic model, *ACTA Math. Appl.* 30 (2007) 855–872.

[9] J.M. Martinez, Practical quasi-Newton methods for solving nonlinear systems, *J. Comput. Appl. Math.* 124 (2000) 97–121.

[10] J.J. Moré, Recent developments in algorithm and software for trust region methods, in *Mathematical Programming: the State of Art*, A. Bachem, M. Grotschel and B. Kortz (eds.), Springer, Berlin, 1983, pp. 258–287.

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

[12] Q. Ni, Optimality conditions for trust-region subproblems involving a conic model, *SIAM J. Optimization* 15 (2005) 826–837.

[13] J. Nocedal, S.J. Wright, *Numerical Optimization,* Science Press, Beijing, 2006.

[14] J.M. Ortega, W.S. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables,* Academic Press, New York, 1970.

[15] D.C. Sorensen, Newton's method with a model trust region modifications, *SIAM J. Numer. Anal.* 19 (1982) 409–426.

[16] W. Sun, Optimization methods for non-quadratic model, *Asia and Pacific Journal of Operational Research* 13 (1996) 43–63.

[17] W. Sun, Y. Yuan, *Optimization Theory and Method: Nonlinear Programming,* Springer, New York, 2006.

[18] Y.J. Wang, N.H. Xiu, *Theory and algorithm for nonlinear programming,* Shanxi Science and Technology Press, Xi'an, 2004.

[19] N. Yamashita, M. Fukushima, On the rate of convergence of the Levenberg-Marquardt method, *Computing* 15 (2001) 239–249.

[20] G.L. Yuan, Z.X. Wei and X.W. Lu, A BFGS trust-region method for nonlinear equations, *Computing* 92 (2011) 317–333.

Manuscript received 1 March 2013
revised 29 July 2013
accepted for publication 1 March 2014

Meilan Zeng
School of Mathematics and Statistics
Hubei Engineering University
Hubei 432000, P.R. China
E-mail address: zml571@126.com

Qin Ni
College of Sciences, Nanjing University
of Aeronautics and Astronautics
Nanjing 211106, P.R. China
E-mail address: niqfs@nuaa.edu.cn