Accepted Manuscript

A modified two steps Levenberg-Marquardt method for nonlinear equations

Keyvan Amini, Faramarz Rostami

PII: S0377-0427(15)00266-6
DOI: http://dx.doi.org/10.1016/j.cam.2015.04.040
Reference: CAM 10154

To appear in: *Journal of Computational and Applied Mathematics*

Received date: 11 February 2015
Revised date: 24 April 2015

A MODIFIED TWO STEPS LEVENBERG-MARQUARDT METHOD FOR NONLINEAR EQUATIONS

Keyvan Amini*, Faramarz Rostami

Department of Mathematics, Faculty of Science, Razi University, Kermanshah, Iran

Abstract

The modified Levenberg-Marquardt (MLM) method to solve nonlinear equations was introduced by Fan in [4]. This method uses an addition of the Levenberg-Marquardt step and an approximate LM step as the trial step at every iteration. Using a trust region technique, the global and cubic convergence of the MLM method under the local error bound condition is proved [4]. Recently, Fan proposed an accelerated MLM algorithm by using a line search strategy to generate a modified LM step and showed that the convergence rate of the algorithm is $\min\{1+2\delta, 3\}$ which results the cubic convergence for $\delta \geq 1$ [5]. In this paper, by introducing an adaptive LM parameter for AMLM algorithm, we propose an efficient AMLM algorithm. The cubic convergence of the new algorithm is presented while numerical experiments show the new algorithm is promising.

Keywords: Nonlinear equations, Levenberg-Marquardt method, Local error bound condition, Line search.

1. Introduction

Consider the nonlinear system of equations

$$F(x) = 0, \tag{1}$$

where $F(x) : \mathbf{R}^n \to \mathbf{R}^n$ is a continuously differentiable function. We denote $F = (f_1, ..., f_n)^T$ and $J(x) = F'(x)$ for all $x \in \mathbf{R}^n$. This problem is one of the cornerstones in computation mathematics and often arise in the applied science such as physics, engineering, chemistry, technology and industry. Due to the nonlinearity of $F(x)$, (1) may have no solution. Throughout the paper, we assume that the solution set of (1) is nonempty and denote it by X^*. Many algorithms have been presented for solving the problem (1), for example, Gauss-Newton method, Newton's method, trust region methods, quasi-Newton methods and etc. ([2-7], [11], [21]). The most common method to solve (1) is Newton method which uses the trial step

$$d_k^N = -J_k^{-1} F_k,$$

in every iteration, where $F_k = F(x_k)$ and $J_k = F'(x_k)$ is the Jacobian of F at x_k. Although, it is known that if $J(x)$ is Lipschitz continuous and nonsingular at the solution then Newton method has quadratic convergence, but it has some disadvantages, especially when the Jacobian matrix is singular or near singular. The Levenberg-Marquardt (LM) family is one of the famous methods to overcome some of these disadvantages. This family computes the trial step by solving the linear system

$$(J_k^T J_k + \lambda_k I)d = -J_k^T F_k, \tag{2}$$

*Corresponding author
Email addresses: kamini@razi.ac.ir (Keyvan Amini*), fara.rostami58@gmail.com (Faramarz Rostami)

where λ_k is the LM parameter that is updated in every iteration. It is well-known that the LM method has quadratic convergence as the Newton method, if Jacobian matrix is nonsingular and Lipschitz continuous in the solution. But the nonsingularity is too strong condition. In [22], Yamashita and Fukushima used the local error bound condition, which is weaker than the nonsingularity of the Jacobian and proved that if the LM parameter is chosen as $\lambda_k = \|F(x_k)\|^2$, the algorithm has quadratic convergence. Fan and Yuan obtained a similar result when $\lambda_k = \|F(x_k)\|$ [7]. The numerical experiments are showed this choice is preferable. In [6], Fan and Pan under the local error bound condition, proved that if the LM parameter is chosen as $\lambda_k = \|F(x_k)\|^\delta$ for $\delta \in (0, 2]$ then the convergence order of the LM algorithm is $min\{1 + \delta, 2\}$. In [4], Fan introduced a modified Levenberg-Marquardt method (MLM) with cubic convergence. The MLM method, at each iteration, first obtains d_k by solving (2), then with setting $y_k = x_k + d_k$ solves the linear system

$$(J_k^T J_k + \lambda_k I)d = -J_k^T F(y_k), \tag{3}$$

to obtain $\hat{d}_k$. So, at every iteration the trial step is set as

$$s_k = d_k + \hat{d}_k.$$

It is noticeable that in the k-iteration, the method doesn't need to compute $J(y_k)$ and uses J_k in (3) that is useful when $F(x)$ is complication or n is large. Fan showed, with choosing

$$\lambda_k = \mu_k \|F_k\|^\delta,$$

with $\delta \in [1, 2]$ *and* $\mu_k > 0$, the MLM method has cubic convergence under the local error bound condition. Notice, if $\hat{d}_k$ is a descent direction of the merit function $\phi(x) = \|F(x)\|^2$ at y_k, then more reduction of ϕ at y_k can be expected. Hence Fan in [5] proposed a line search in y_k along $\hat{d}_k$ by solving the problem

$$\min_{\alpha>0} \|F(y_k + \alpha \hat{d}_k)\|^2,$$

which can be approximated by

$$\max_{\alpha>0} \Psi(\alpha) = \|F(y_k)\|^2 - \|F(y_k) + \alpha J_k \hat{d}_k\|^2. \tag{4}$$

It is clear that $\Psi(\alpha)$ is a quadratic function that its solution is attained at

$$\widetilde{\alpha}_k = 1 + \frac{\lambda_k \hat{d}_k^T \hat{d}_k}{\hat{d}_k^T J_k^T J_k \hat{d}_k} > 1.$$

Because $\widetilde{\alpha}_k$ may be very large when $J_k \hat{d}_k$ be close to 0, by using an upper bound for the step size α_k and setting

$$\alpha_k = min\{\widetilde{\alpha}_k \,,\, \hat{\alpha}\}. \tag{5}$$

where $\hat{\alpha} > 1$ is a positive constant, Fan modified the trial step as follows

$$s_k = d_k + \alpha_k \hat{d}_k,$$

and proved that the convergence order of this method is $c(\delta) = min\{1+2\delta, 3\}$. It is easily seen that $c(\delta) < 3$ when $\delta < 1$. In order to generate an algorithm with cubic convergence, one must focus on $\delta \geq 1$. It is noticeable when the sequence is far away from the solution set and $\delta \geq 1$ then $\lambda_k = \mu_k \|F_k\|^\delta$ may be very large which makes the LM step small and prevents the iterates from moving fast to the solution set. So it is seem, for $\delta \geq 1$, the AMLM method with $\lambda_k = \mu_k \|F_k\|^\delta$ takes backfire on the result of initial steps that are far away from of the solution set. In this paper, to overcome this disadvantage, we introduce a new adaptive choice for δ_k as follows

$$\delta_k = \begin{cases} \frac{1}{\|F_k\|} & \text{if } \|F_k\| \geq 1, \\ 1 + \frac{1}{k} & \text{otherwise,} \end{cases} \tag{6}$$

and set

$$\lambda_k = \mu_k \|F_k\|^{\delta_k}. \tag{7}$$

It is noticeable that even if $\|F_k\|$ is very large, $\lambda_k = \mu_k \|F_k\|^{1/\|F_k\|}$ isn't large and so the LM step isn't small too. This causes that the algorithm moves fast to the solution set. The numerical results on a famous family of test problems show the new choice is promising while we show the convergence rate of the new algorithm is cubic. The paper is organized as follows: In Section 2, we describe the method in more details. In Section 3, we show that the new algorithm preserves the same global convergence as the existing MLM algorithms under suitable conditions. In Section 4, we derive the convergence order of the new algorithm under the local error bound condition. Finally, in Section 5, we report some numerical results to compare the new algorithm along with the other algorithms.

2. The algorithm

We take

$$\phi(x) = \|F(x)\|^2, \tag{8}$$

as the merit function for (1). In AMLM method [5], the trial step is defined by

$$s_k = d_k + \alpha_k \hat{d}_k. \tag{9}$$

where d_k *and* $\hat{d}_k$ are computed by (2) and (3) respectively and α_k is computed by (5). It is clear that the actual reduction of $\phi(x)$ at the kth iteration is as follows

$$Ared_k = \|F_k\|^2 - \|F(x_k + d_k + \alpha_k \hat{d}_k)\|^2. \tag{10}$$

Note that the predicted reduction can not be defined as usual definition

$$Apred_k = \|F_k\|^2 - \|F_k + J_k(d_k + \alpha_k \hat{d}_k)\|^2, \tag{11}$$

because it can not be proven to be nonnegative that is required for the global convergence in the trust region methods. Hence similar to [5], we use the modified predicted reduction as follows

$$Pred_k = \|F_k\|^2 - \|F_k + J_k d_k\|^2 + \|F(y_k)\|^2 - \|F(y_k) + \alpha_k J_k \hat{d}_k\|^2, \tag{12}$$

this definition has some appropriate properties. An important property is described in the following lemma. The details of proof can be seen in [5].

Lemma 2.1. *Let the predicted reduction is defined by (12) then*

$$Pred_k \geq \|J_k^T F_k\| \min\{\|d_k\|, \frac{\|J_k^T F_k\|}{\|J_k^T J_k\|}\} + \|J_k^T F(y_k)\| \min\{\|\hat{d}_k\|, \frac{\|J_k^T F(y_k)\|}{\|J_k^T J_k\|}\}. \tag{13}$$

Remark 2.1. *Using (11), (12) and some simple calculations, it is concluded that*

$$\begin{aligned} Apred_k - Pred_k &= \|F_k + J_k d_k\|^2 - \|F(y_k)\|^2 + \|F(y_k) + \alpha_k J_k \hat{d}_k\|^2 - \|F_k + J_k(d_k + \alpha_k \hat{d}_k)\|^2 \\ &= 2\alpha_k \hat{d}_k^T J_k^T (F(y_k) - F_k - J_k d_k), \end{aligned}$$

which means

$$Apred_k = Pred_k + \|\hat{d}_k\| O(\|d_k\|^2) = O(Pred_k). \tag{14}$$

So, in practice we can use either $Apred_k$ or $Pred_k$ in our algorithm. In this paper, we use $Pred_k$ that is described in (12).

In [4-7], [25] and etc, the LM parameter $\lambda_k = \mu_k \|(F_k)\|^\delta$ is used where $\delta = 1$ is generally chosen in numerical results. They believe that numerical results with $\delta = 1$ are more stable and preferable. But there are some drawbacks, for example when the sequence is far away from the solution set, $\|F_k\|$ and so λ_k may be very large which makes the LM step small and hence prevents the iterates from moving fast to the solution set. Therefore we introduce δ_k as (6) that causes the LM parameter $\lambda_k = \mu_k \|F_k\|^{\delta_k}$ be neither very large nor very small, so the algorithm moves fast to the solution set, particularly in the initial iterations that the sequence is generally far away from the solution set.

Now, it is convenient to present the complete algorithm as follows.

Algorithm 2.1 (The new adaptive two-steps LM algorithm)

Input: $x_0 \in \mathbf{R}^n$, $\mu_1 > m > 0$, $0 < p_0 \leq p_1 \leq p_2 < 1$, $\hat{\alpha} > 1$ *and* $\epsilon > 0$.
Step 0. Set $k := 0$.
Step 1. Compute $F_k = F(x_k)$ and $J_k = J(x_k)$.
Step 2. If $\|J_k^T F_k\| < \epsilon$, stop. Otherwise set $\lambda_k = \mu_k \|F_k\|^{\delta_k}$ where

$$\delta_k = \begin{cases} \frac{1}{\|F_k\|} & \text{if } \|F_k\| \geq 1, \\ 1 + \frac{1}{k} & \text{otherwise.} \end{cases} \tag{15}$$

Step 3. Solve the linear system

$$(J_k^T J_k + \lambda_k I)d = -J_k^T F_k, \tag{16}$$

to compute d_k. Set $y_k = x_k + d_k$.
Step 4. Solve the linear system

$$(J_k^T J_k + \lambda_k I)d = -J_k^T F(y_k), \tag{17}$$

to obtain $\hat{d}_k$.
Step 5. Calculate α_k by (5) and set

$$s_k = d_k + \alpha_k \hat{d}_k. \tag{18}$$

Step 6. Compute $r_k = Ared_k / Pred_k$ where $Ared_k$ and $Pred_k$ are defined by (10) and (12), respectively.
Step 7. Set

$$x_{k+1} = \begin{cases} x_k + s_k & \text{if } r_k \geq p_0, \\ x_k & \text{otherwise.} \end{cases} \tag{19}$$

Step 8. Choose μ_{k+1} as

$$\mu_{k+1} = \begin{cases} 4\mu_k & \text{if } r_k < p_1, \\ \mu_k & \text{if } r_k \in [p_1, p_2], \\ \max\{\frac{\mu_k}{4}, m\} & \text{otherwise.} \end{cases} \tag{20}$$

Step 9. Set k=k+1 and go to step 1.

To prevent the steps from being too large when the sequence is near the solution, we require

$$\mu_k \geq m, \qquad \forall\, k \in \mathrm{N}, \tag{21}$$

where m is a positive constant.

Lemma 2.2. *Let the sequence $\{x_k\}$ is generated by Algorithm 2.1, then the sequence $\{\|F_k\|\}$ is decreasing.*

Proof. If $r_k < p_0$, (19) implies $x_{k+1} = x_k$ and so $\|F(x_{k+1})\| = \|F(x_k)\|$. So, we can let $r_k \geq p_0 > 0$. Because lemma 2.1 implies $Predk \geq 0$, it is concluded that $Aredk > 0$. Therefore by (10), we have $\|F_k\| > \|F(x_k + d_k + \alpha_k \hat{d}_k)\|$. So, in any case the sequence $\{\|F_k\|\}$ is a decreasing sequence. □

3. The global convergence

In this section, we study the global convergence of the new algorithm. In order to, we need the following assumption.

Assumption 3.1
$F(x)$ and $J(x)$ are both Lipschitz continuous, that is, there exists a positive constant L such that

$$\|J(y) - J(x)\| \leq L\|y - x\|, \quad \forall x, y \in \mathbf{R}^n, \tag{22}$$

and

$$\|F(y) - F(x)\| \leq L\|y - x\|, \quad \forall x, y \in \mathbf{R}^n. \tag{23}$$

By (22), it can be easily seen that

$$\|F(y) - F(x) - J(x)(y - x)\| \leq L\|y - x\|^2, \quad \forall x, y \in \mathbf{R}^n. \tag{24}$$

Theorem 3.1. *Let Assumption 3.1 hold. Then Algorithm 2.1 terminates in finite iterations or satisfies*

$$\lim_{k\to\infty} \|J_k^T F_k\| = 0. \tag{25}$$

Proof. It is clear that if $\|F_k\| \geq 1$ then $\delta_k = \frac{1}{\|F_k\|} \in (0, 1]$ and if $\|F_k\| < 1$ then $\delta_k = 1 + \frac{1}{k} \in (1, 2]$. So, for any k, $\delta_k \in (0, 2]$, thus the proof is similar to Theorem 2.3 in [4]. □

Lemma 2.2 along with Theorem 3.1 show that the sequence generated by Algorithm 2.1 converges to a stationary point of the merit function $\phi(x)$.

The following lemma shows, for sufficiently large k, $\delta_k = 1 + \frac{1}{k}$.

Lemma 3.2. *Let Assumption 3.1 hold and the sequence $\{x_k\}$ is generated by Algorithm 2.1, then the set $D = \{k \in N : \ \|F_k\| \geq 1\}$ is finite.*

Proof. By contradiction, suppose the set D is infinite. This along with lemma 2.2 imply that

$$\|F_k\| \geq 1, \quad \forall\, k \in \mathbf{N}. \tag{26}$$

By theorem 3.1, the sequence $\{x_k\}$ is converge to x_*. Because F is Lipschitz continuous, by (23), there exists a constant L so that

$$\|F(x_k)\| = \|F(x_k) - F(x_*)\| \leq L\|x_k - x_*\|, \tag{27}$$

this inequality along with (26) conclude that

$$\|x_k - x_*\| \geq \frac{1}{L}, \quad \forall\, k \in \mathbf{N},$$

which is a contradiction to the fact that $x_k \to x_*$. This shows the assumption is incorrect and the proof is completed. □

4. Local convergence rate of Algorithm 2.1

In this section, we study the local convergence properties of the new MLM algorithm. In a similar manner with [4], we show the sequence generated by algorithm converges cubically to the solution if $\|F_k\|$ provides a local error bound near some $x^* \in X^*$. we need some assumptions.

Definition 4.1. *Let N be a subset of $\mathbf{R}^n$ such that $N \cap X^* \neq \phi$, we say that $\|F(x)\|$ provides a local error bound on N for (1), if there exists a positive constant $c > 0$ such that*

$$\|F(x)\| \geq c\, dist(x, X^*), \quad \forall x \in N, \tag{28}$$

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

In the sequel, we denote $\bar{x}_k$, the vector in X^* satisfying

$$\|\bar{x}_k - x_k\| = dist(x_k, X^*). \tag{29}$$

Assumption 4.1
(a) There exists a solution $x^* \in X^*$ of (1).
(b) $F(x)$ and $J(x)$ are both Lipschitz continuous on $N(x^*, b)$, i.e., there exists a positive constant $L > 0$ such that

$$\|J(y) - J(x)\| \leq L\|y - x\|, \quad \forall x, y \in N(x^*, b), \tag{30}$$

and

$$\|F(y) - F(x)\| \leq L\|y - x\|, \quad \forall x, y \in N(x^*, b). \tag{31}$$

where

$$0 < b < 1\,, \quad N(x^*, b) = \{x \in \mathbf{R}^n \;\mid\; \|x - x^*\| \leq b\}.$$

(c) $F(x)$ is continuously differentiable and $\|F(x)\|$ provides a local error bound on $N(x^*, b)$ for (1).

Due to the Lipschitzness of the Jacobian, we have

$$\|F(y) - F(x) - J(x)(y - x)\| \leq L\|y - x\|^2, \quad \forall x, y \in N(x^*, b). \tag{32}$$

The following lemma describes an important property for the directions $d_k, \hat{d}_k$.

Lemma 4.1. *Under Assumption 4.1, if $x_k, y_k \in N(x^*, \frac{b}{2})$, for sufficiently large k, we have*

(a) $\|d_k\| \leq c_1\, dist(x_k, X^*)$,

(b) $\|\hat{d}_k\| \leq c_2\, dist(x_k, X^*)$,

(c) $\|s_k\| \leq c_3\, dist(x_k, X^*)$,

where c_1, c_2 and c_3 are positive constants.

Proof. The proof is similar to Lemma 3.2 in [5]. □

In the sequel, we show that $\{\mu_k\}$ is bounded above that plays an important role to estimate $\|F_k + J_k d_k\|$ and $\|F(y_k) + \alpha_k J_k \hat{d}_k\|$.

Lemma 4.2. *Under Assumption 4.1, if $x_k, y_k \in N(x^*, \frac{b}{2})$, then*

(a) *There exists a positive constant $M > m$ such that for all sufficiently large k,*

$$\mu_k \leq M, \tag{33}$$

holds.

(b) $\lambda_k = O(\|\bar{x}_k - x_k\|^{1+\frac{1}{k}})$, *for all sufficiently large k.*

Proof. If we note that $\delta_k = 1 + \frac{1}{k} \in (1,2)$ for sufficiently large k, The proof of (a) is similar to Lemma 3.3. in [5]. The relations (29), (31), (33) and Lemma 3.2 indicate that the LM parameter, for all sufficiently large k, satisfies

$$\lambda_k = \mu_k \|F_k\|^{1+\frac{1}{k}} \leq L^{1+\frac{1}{k}} M \|\bar{x}_k - x_k\|^{1+\frac{1}{k}}. \tag{34}$$

On the other hand, (29) along with $x_k \in N(x^*, \frac{b}{2})$, result that

$$\|\bar{x}_k - x^*\| \leq \|\bar{x}_k - x_k\| + \|x_k - x^*\| \leq 2\|x_k - x^*\| \leq b,$$

so, $\bar{x}_k \in N(x^*, b)$. Therefore by (21), (28) and (29), the LM parameter satisfies

$$\lambda_k \geq mc^{1+\frac{1}{k}} \|\bar{x}_k - x_k\|^{1+\frac{1}{k}}. \tag{35}$$

This inequality together with (34) complete the proof of (b). □

According to the result given by Behling and Iusem in ([1], Theorem 1) and without loss of generality, we assume $rank(J(\bar{x})) = r \;\; for\, all \;\bar{x} \in N(x^*, b) \bigcap X^*$. Suppose the singular value decomposition (SVD) of $J(\bar{x})$ is

$$J(\bar{x}) = [\bar{U}_1, \bar{U}_2] \begin{bmatrix} \bar{\Sigma}_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{V}_1^T \\ \bar{V}_2^T \end{bmatrix} = \bar{U}_1 \bar{\Sigma}_1 \bar{V}_1^T,$$

where $\bar{U} = [\bar{U}_1, \bar{U}_2]$ and $\bar{V} = [\bar{V}_1, \bar{V}_2]$ are two orthogonal matrixes and $\bar{\Sigma}_1$ is a positive diagonal matrix. Correspondingly, we consider the SVD of $J(x)$ by

$$J(x) = [U_1, U_2, U_3] \begin{bmatrix} \Sigma_1 & 0 & 0 \\ 0 & \Sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \\ V_3^T \end{bmatrix} = U_1 \Sigma_1 V_1^T + U_2 \Sigma_2 V_2^T, \tag{36}$$

where $U = [U_1, U_2, U_3]$ and $V = [V_1, V_2, V_3]$ are two orthogonal matrixes and $\Sigma_1 = diag(\sigma_1, \sigma_2, ..., \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0$, and $\Sigma_2 = diag(\sigma_{r+1}, \sigma_{r+2}, ..., \sigma_{r+q}) \; with \; \sigma_{r+1} \geq \sigma_{r+2} \geq ... \geq \sigma_{r+q} > 0$.
In the following, for clearness, we also neglect the subscription k in the decomposition of $J(x_k)$ and still write J_k as same as (36)

$$J_k = U_1 \Sigma_1 V_1^T + U_2 \Sigma_2 V_2^T. \tag{37}$$

By the theory of matrix perturbation [19] and the Lipschitzness of J_k, we have

$$\|diag(\Sigma_1 - \bar{\Sigma}_1, \Sigma_2, 0)\| \leq \|J_k - \bar{J}_k\| \leq L\|\bar{x}_k - x_k\|,$$

which yields

$$\|\Sigma_1 - \bar{\Sigma}_1\| \leq L\|\bar{x}_k - x_k\| \quad\; and \quad\; \|\Sigma_2\| \leq L\|\bar{x}_k - x_k\|. \tag{38}$$

Since $\{x_k\}$ converges to the solution set X^*, we assume that $L\|\bar{x}_k - x_k\| \leq \frac{\bar{\sigma}_r}{2}$ holds for all sufficiently large k. Then it follows from (38) that

$$\|\Sigma_1^{-1}\| \leq \frac{1}{\bar{\sigma}_r - L\|\bar{x}_k - x_k\|} \leq \frac{2}{\bar{\sigma}_r}, \tag{39}$$

moreover, for sufficiently large k, we have from (35), (38) and Lemma 3.2 that

$$\|\lambda_k^{-1} \Sigma_2\| = \frac{\|\Sigma_2\|}{\mu_k \|F_k\|^{1+\frac{1}{k}}} \leq \frac{L}{mc^{1+\frac{1}{k}}} \|\bar{x}_k - x_k\|^{-\frac{1}{k}}. \tag{40}$$

The next lemma plays an important role in the proof of cubic convergence of Algorithm 2.1.

Lemma 4.3. *Under Assumption 4.1, if $x_k \in N(x^*, \frac{b}{2})$, then we have*

(a) $\|U_1U_1^TF_k\| \le O(\|\bar{x}_k - x_k\|)$,

(b) $\|U_2U_2^TF_k\| \le O(\|\bar{x}_k - x_k\|^2)$,

(c) $\|U_3U_3^TF_k\| \le O(\|\bar{x}_k - x_k\|^2)$,

(d) $\|U_1U_1^TF(y_k)\| \le O(\|\bar{x}_k - x_k\|^2)$,

(e) $\|U_2U_2^TF(y_k)\| \le O(\|\bar{x}_k - x_k\|^3)$,

(f) $\|U_3U_3^TF(y_k)\| \le O(\|\bar{x}_k - x_k\|^3)$,

for all sufficiently large k.

Proof. We can find the proof of (a), (b) and (c) in [4]. Also, the proof of (d), (e) and (f) is similar to Lemma (3-5) in [5].

Lemma 4.4. *Let Assumption 4.1 is satisfied and the sequence $\{x_k\}$ is generated by Algorithm 2.1, then, for any sufficiently large k, we have*

(a) $\|\hat{d}_k\| \le O(\|\bar{x}_k - x_k\|^2)$,

(b) $\|F(y_k) + \alpha_k J_k \hat{d}_k\| \le O(\|\bar{x}_k - x_k\|^3)$.

Proof. Using the SVD of J_k, we get

$$\hat{d}_k = -V_1(\Sigma_1^2 + \lambda_k I)^{-1}\Sigma_1 U_1^T F(y_k) - V_2(\Sigma_2^2 + \lambda_k I)^{-1}\Sigma_2 U_2^T F(y_k), \tag{41}$$

and

$$\begin{aligned} &F(y_k) + J_k\hat{d}_k \\ &\quad = F(y_k) - U_1\Sigma_1(\Sigma_1^2 + \lambda_k I)^{-1}\Sigma_1 U_1^T F(y_k) - U_2\Sigma_2(\Sigma_2^2 + \lambda_k I)^{-1}\Sigma_2 U_2^T F(y_k) \\ &\quad = \lambda_k U_1(\Sigma_1^2 + \lambda_k I)^{-1} U_1^T F(y_k) + \lambda_k U_2(\Sigma_2^2 + \lambda_k I)^{-1} U_2^T F(y_k) + U_3U_3^T F(y_k). \end{aligned} \tag{42}$$

It follows from (35), (39-41) and Lemma (4.3) that

$$\begin{aligned} \|\hat{d}_k\| &= \| - V_1(\Sigma_1^2 + \lambda_k I)^{-1}\Sigma_1 U_1^T F(y_k) - V_2(\Sigma_2^2 + \lambda_k I)^{-1}\Sigma_2 U_2^T F(y_k)\| \\ &\le \|\Sigma_1\|^{-1}\|U_1^T F(y_k)\| + \|\lambda_k{}^{-1}\Sigma_2\|\|U_2^T F(y_k)\| \\ &\le O(\|\bar{x}_k - x_k\|^2) + O(\|\bar{x}_k - x_k\|^{3-\frac{1}{k}}) = O(\|\bar{x}_k - x_k\|^2). \end{aligned} \tag{43}$$

Computing α_k by (5) together with (34), (42) and Lemma (4.3) result

$$\begin{aligned} \|F(y_k) + \alpha_k J_k \hat{d}_k\| &\le \|F(y_k) + J_k\hat{d}_k\| \\ &\le \lambda_k\|\Sigma_1^{-2}\|\|U_1^T F(y_k)\| + \|U_2^T F(y_k)\| + \|U_3^T F(y_k)\| \\ &\le O(\|\bar{x}_k - x_k\|^{3+\frac{1}{k}}) + O(\|\bar{x}_k - x_k\|^3) + O(\|\bar{x}_k - x_k\|^3) \\ &= O(\|\bar{x}_k - x_k\|^3). \end{aligned} \tag{44}$$

The proof is completed. □

Now we can state the convergence result of Algorithm 2.1.

Theorem 4.5. *Let the sequence $\{x_k\}$ is generated by Algorithm 2.1, under the conditions of Assumption 4.1, the sequence $\{x_k\}$ converges cubically to a solution of (1).*

Proof. The relations (28), (30) and (32) conclude that

$$\begin{aligned} c\,\|\bar{x}_{k+1} - x_{k+1}\| &\le \|F(x_{k+1})\| \\ &= \|F(y_k + \alpha_k \hat{d}_k)\| \\ &\le \|F(y_k) + \alpha_k J(y_k)\hat{d}_k\| + L\alpha_k^2\|\hat{d}_k\|^2 \\ &\le \|F(y_k) + \alpha_k J_k\hat{d}_k\| + \alpha_k\|(J(y_k) - J_k)\hat{d}_k\| + L\alpha_k^2\|\hat{d}_k\|^2 \\ &\le \|F(y_k) + \alpha_k J_k\hat{d}_k\| + L\hat{\alpha}\|d_k\|\|\hat{d}_k\| + L\hat{\alpha}^2\|\hat{d}_k\|^2. \end{aligned}$$

Using Lemmas 4.1 and 4.4, we obtain

$$c\,\|\bar{x}_{k+1} - x_{k+1}\| \le O(\|\bar{x}_k - x_k\|^3) + O(\|\bar{x}_k - x_k\|^3) + O(\|\bar{x}_k - x_k\|^4) = O(\|\bar{x}_k - x_k\|^3), \tag{45}$$

which implies that x_k is cubically convergence to a solution of set X^*.
In other hand, it is clear that

$$\|\bar{x}_k - x_k\| = dist(x_k, X^*) \le \|\bar{x}_{k+1} - x_k\| \le \|\bar{x}_{k+1} - x_{k+1}\| + \|s_k\|.$$

It follows from (45) and lemma 4.1 that, for any k sufficiently large,

$$\|\bar{x}_k - x_k\| \le 2\|s_k\| \le 2c_3 dist(x_k, X^*) = 2c_3\|\bar{x}_k - x_k\|.$$

So, $\|s_k\| = O(\|\bar{x}_k - x_k\|)$ holds for all sufficiently large k. Hence we deduce from (45) that

$$\|s_{k+1}\| \le O(\|s_k\|^3), \tag{46}$$

which indicates the sequence $\{x_k\}$ converges cubically to a solution of (1). □

5. Numerical experiments

This section devote to report some numerical experiments to show the promising behavior of Algorithm 2.1 in comparison with the performance of modified Levenberg-Marquardt (MLM) method [4] and accelerating the modified Levenberg-Marquardt (AMLM) method [5] on some singular test problems. These test problems are constructed by modifying the standard test problems given in [16] by the following form

$$\hat{F}(x) = F(x) - J(x^*)A(A^TA)^{-1}A^T(x - x^*),$$

where $F(x)$ is the standard test function, $A \in \mathbf{R}^{n\times k}$ has full column rank with $1 \le k \le n$ and x^* is a solution of the equation $F(x) = 0$. Obviously

$$\hat{J}(x^*) = J(x^*)(I - A(A^TA)^{-1}A^T),$$

is the Jacobian matrix of $\hat{F}(x)$ with rank $n-k$ and $\hat{F}(x^*) = 0$. However, $\hat{F}(x)$ may have roots that are not roots of $F(x)$. We constructed two set of singular problems while $\hat{J}(x^*)$ have rank $n-1$ or $n-2$, by choosing

$$A = [1, 1, ..., 1]^T \in \mathbf{R}^{n\times 1},$$

and

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & ... & 1 \\ 1 & -1 & 1 & -1 & ... & \pm 1 \end{bmatrix}^T \in \mathbf{R}^{n\times 2},$$

respectively [18].

All codes are written in MATLAB R2009 programming environment on a personal PC with 2.5 GHz, 4 GB RAM, using Windows 7 operation system. For all algorithms, we set

$$p_0 = 0.0001,\ p_1 = 0.25,\ p_2 = 0.75,\ \mu_1 = 0.5,\ m = 10^{-6}\ and\ \tilde{\alpha} = 5.$$

The algorithms are terminated when the number of iterations exceeds 1000 or

$$\|J_k^T F_k\| \le 10^{-5}.$$

Tables 1 and 2 list the numerical results for the algorithms on the test problems with different starting points. The results for the first set problems with rank $n-1$ and the second with rank $n-2$ are listed in Tables 1, 2 respectively. We set $n = 1000$ in all problems, with the exception of Extended Helical valley function, that n is set 999. All test problems are run for four starting points $x_0,\ 10x_0,\ 100x_0\ and\ 1000x_0,\ where\, x_0$ is suggested in [16]. In Tables, "NF" and "NJ" represent the numbers of function evalutions, Jacobian evalutions and "NS?" returns Y(yes) or N(no) while "Y" shows the corresponding method is converged to x^* and "N" shows that it is converged to another solution. Besides, the sign "$-$" indicates that the number of iterations exceeds 1000 and "OF" indicates the algorithm had underflows or overflows. Note that, for general nonlinear equations, the evalutions of the Jacobian are usually n times of the function evalutions. So, we also use the values "$NT = NF + NJ * n$" for comparisons of the total evalutions.

It is seen from Table 1 that Algorithm 2.1 solves 87% of the problems in the least number of total evalutions while AMLM and MLM solve only 39% and 21% in the least number of total evalutions, respectively. In a similar view of Table 2, it is easy to see that Algorithm 2.1 wins over 89% of problems while AMLM and MLM algorithm wins only 50% and 32% of the test problems. Moreover, from these tables, we see that, the new algorithm performs better than two other algorithms; the number of "NT", "NJ" and "NF" are reduced. Furthermore, for Extended Helical valley problem, with $100x_0$ and $1000x_0$, our algorithm could successfully find x^* while other algorithms find another solution of $\hat{F}(x) = 0$ for this function. Hence, it seems that our new Levenberg-Marquardt algorithm is more efficient for nonlinear equations.

References

[1] R. Behling and A. Iusem, *The effect of calmness on the solution set of systems of nonlinear equations,* Math. Program. **137** (2013), 155-165.

[2] C. G. Broyden, *Quasi-Newton methods and their applications to function minimization.* Math. Comp. **21** (1967), 577-593.

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

[4] J. Fan, *The modified Levenberg-Marquardt method for nonlinear equations with cubic convergence.* Math. Comp. **81** (2012), 447-466.

[5] J. Fan, *Accelerating the modified Levenberg-Marquardt method for nonlinear equations.* Math. Comp. **83** (2014), 1173-1187.

[6] J. Fan and J. Pan, *Convergence properties of a self-adaptive Levenberg-Marquardt algorithm under local error bound condition.* Comput. Optim. Appl. **34** (2006), 47-62.

[7] J. Fan and Y. X. Yuan, *On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption.* Computing **74** (2005), 23-39.

[8] A. Fischer, *Local behavior of an iterative framework for generalized equations with nonisolated solutions.* Math. Program. **94** (2002), 91-124.

[9] A. Fischer and P. K. Shukla, *A Levenberg-Marquardt algorithm for unconstrained multicriteria optimization.* Math. Comp. **36** (2008), 643-646.

[10] A. Fischer, P. K. Shukla and M. Wang, *On the inexactness level of robust Levenberg-Marquardt methods.* Optimization **59** (2010), 273-287.

[11] C. T. Kelley, *Iterative Methods for Optimization.* SIAM, Philadelphia. **18** (1999).

[12] K. Levenberg, *A method for the solution of certain nonlinear problems in least squares.* Quart. Appl. Math. **2** (1944), 164-168.

[13] D. W. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters.* J. Soc. Indust. Appl. Math. **11** (1963), 431-441.

[14] J. J. Moré, *The Levenberg-Marquardt algorithm, implementation and theory.* Numerical analysis, Springer, Berlin Heidelberg, **630** (1978), 105-116.

[15] J. J. Moré, *Recent developments in algorithms and software for trust region methods.* Math. program. **85** (1983), 258-287.

Table 1: Numerical results for singular nonlinear equations with rank $(F'(x^*)) = n-1$

		MLM		AMLM		Algorithm 2.1	
Problem	x_0	$NF/NJ/NT$	$NS?$	$NF/NJ/NT$	$NS?$	$NF/NJ/NT$	$NS?$
Extended Rosenbrock	1	105/52/52105	Y	265/85/85265	Y	55/27/27055	Y
	10	131/53/53131	Y	269/88/88269	Y	29/15/15029	Y
	100	53/25/25053	Y	65/28/28065	Y	33/17/17033	Y
	1000	51/26/26051	Y	55/27/27055	Y	37/19/19037	Y
Extended Powell singular	1	17/9/9017	Y	17/9/9017	Y	15/8/8015	Y
	10	23/12/12023	Y	21/11/11021	Y	21/11/11021	Y
	100	27/14/14027	Y	43/17/17043	Y	25/13/13025	Y
	1000	33/17/17033	Y	33/17/17033	Y	29/15/15029	Y
Extended Powell badly	1	1337/402/403337	N	1919/528/529919	N	11/6/6011	N
	10	-		-		-	
	100	-		-		-	
	1000	-		-		-	
Extended Wood	1	27/14/14027	Y	27/14/14027	Y	25/13/13025	Y
	10	33/17/17033	Y	49/22/22049	Y	29/15/15029	Y
	100	135/51/51135	Y	181/77/77181	Y	33/17/17033	Y
	1000	975/309/309975	Y	1405/558/5581405	Y	39/20/20039	Y
Extended Helical valley	1	21/11/11021	N	35/14/14035	N	15/8/8015	N
	10	31/14/14031	N	25/12/12025	N	13/7/7013	N
	100	19/10/10009	N	19/10/10009	N	73/34/34073	Y
	1000	25/13/13015	N	23/12/12023	N	61/22/22061	Y
Brown almost-linear	1	15/8/8015	Y	13/7/7013	Y	13/7/7013	Y
	10	OF		OF		OF	
	100	OF		OF		OF	
	1000	OF		OF		OF	
Discrete boundary value	1	1/1/1001	Y	1/1/1001	Y	1/1/1001	Y
	10	15/8/8015	N	13/7/7013	N	13/7/7013	N
	100	27/14/14027	N	27/14/14027	N	25/13/13025	N
	1000	35/18/18035	N	31/16/16031	N	29/15/15029	N
Discrete integral equation	1	15/8/8015	N	13/7/7013	N	11/6/6011	N
	10	21/11/11021	N	19/10/10019	N	17/9/9017	N
	100	15/8/8015	N	15/8/8015	N	13/7/7013	N
	1000	23/12/12023	N	23/12/12023	N	23/12/12023	N
Trigonometric	1	23/8/8023	Y	25/7/7025	Y	33/7/7033	Y
	10	75/28/28075	Y	77/25/25077	Y	85/30/30085	Y
	100	59/19/19059	Y	61/20/20061	Y	51/15/15051	Y
	1000	73/24/24073	Y	77/25/25077	Y	69/25/25069	Y
Broyden tridiagonal	1	15/8/8015	Y	13/7/7013	Y	13/7/7013	Y
	10	21/11/11021	Y	21/11/11021	Y	19/10/10019	Y
	100	27/14/14027	Y	25/13/13025	Y	25/13/13025	Y
	1000	31/16/16031	Y	31/16/16031	Y	29/15/15029	Y
Broyden banded	1	17/9/9017	Y	17/9/9017	Y	17/9/9017	Y
	10	27/14/14027	Y	27/14/14027	Y	27/14/14027	Y
	100	35/18/18035	Y	35/18/18035	Y	35/18/18035	Y
	1000	43/22/22043	Y	43/22/22043	Y	43/22/22043	Y

[16] J. J. Moré, B. S. Garbow and K. E. Hillstrom, *Testing unconstrained optimization software.* ACM Trans. Math. Softw. **7** (1981), 17-41.
[17] M. J. D. Powell, *Convergence properties of a class of minimization algorithms: Nonlinear programming.* AERE, New York. (1974).
[18] R. B. Schnabel and P. D. Frank, *Tensor methods for nonlinear equations.* SIAM J. Numer. **21** (1984), 815-843.
[19] G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory, Computer Science and Scientific Computing.* Boston. (1990).
[20] W. Sun and Y. X. Yuan, *Optimization Theory and Methods: Nonlinear programming.* Springer, New York. (2006).
[21] Ph. L. Toint, *Nonlinear step size control, trust regions and regularization for unconstrained optimization.* Optim. Methods Softw. **28(1)** (2013), 82-95.
[22] N. Yamashita and M. Fukushima, *On the rate of convergence of the Levenberg-Marquardt method.* Comput. Suppl. **15** (2001), 237-249.
[23] Y. X. Yuan, *Trust region algorithms for nonlinear equations.* Hong Kong Baptist University, Department of Mathematics. (1998).
[24] Y. X. Yuan, *Recent advances in numerical methods for nonlinear equations and nonlinear least squares.* Numer. Algebra Control Optim. **1** (2011), 15-34.
[25] W. Zhou, *On the convergence of the modified Levenberg-Marquardt method with a nonmonotone second order Armijo type line search.* Comput. Appl. Math. **239** (2013), 152-161.

Table 2: Numerical results for singular nonlinear equations with rank $(F'(x^*)) = n-2$

		MLM		$AMLM$		$Algorithm 2.1$	
Problem	x_0	$NF/NJ/NT$	$NS?$	$NF/NJ/NT$	$NS?$	$NF/NJ/NT$	$NS?$
Extended Rosenbrock	1	17/9/9017	Y	17/9/9017	Y	17/9/9017	Y
	10	21/11/11021	N	21/11/11021	N	21/11/11021	N
	100	25/13/13025	N	25/13/13025	N	25/13/13025	N
	1000	31/16/16031	N	31/16/16031	N	29/15/15029	N
Extended Powell singular	1	17/9/9017	Y	17/9/9017	Y	15/8/8015	Y
	10	23/12/12023	Y	21/11/11021	Y	21/11/11021	Y
	100	27/14/14027	Y	43/17/17043	Y	25/13/13025	Y
	1000	33/17/17033	Y	33/17/17033	Y	29/15/15029	Y
Extended Powell badly	1	279/134/134279	N	483/169/169483	N	47/7/7047	N
	10	-		-		-	
	100	-		-		-	
	1000	-		-		-	
Extended Wood	1	23/12/12023	N	23/12/12023	N	23/12/12023	N
	10	27/14/14027	N	27/14/14027	N	27/14/14027	N
	100	31/16/16031	N	31/16/16031	N	31/16/16031	N
	1000	37/19/19037	N	37/19/19037	N	35/18/18035	N
Extended Helical valley	1	21/11/11021	N	35/14/14035	N	15/8/8015	N
	10	31/14/14031	N	25/12/12025	N	13/7/7013	N
	100	19/10/10009	N	19/10/10009	N	73/34/34073	Y
	1000	25/13/13015	N	23/12/12023	N	65/22/22065	Y
Brown almost-linear	1	15/8/8015	Y	13/7/7013	Y	13/7/7013	Y
	10	OF		OF		OF	
	100	OF		OF		OF	
	1000	OF		OF		OF	
Discrete boundary value	1	1/1/1001	Y	1/1/1001	Y	1/1/1001	Y
	10	15/8/8015	N	13/7/7013	N	13/7/7013	N
	100	27/14/14027	N	27/14/14027	N	25/13/13025	N
	1000	35/18/18035	N	31/16/16031	N	29/15/15029	N
Discrete integral equation	1	15/8/8015	N	13/7/7013	N	11/6/6011	N
	10	21/11/11021	N	19/10/10019	N	17/9/9017	N
	100	15/8/8015	N	15/8/8015	N	13/7/7013	N
	1000	33/13/13033	N	31/13/13031	N	31/12/12031	N
Trigonometric	1	23/8/8023	Y	25/7/7025	Y	33/7/7033	Y
	10	79/29/29079	Y	83/28/28083	Y	75/25/25075	Y
	100	59/19/19059	Y	61/20/20061	Y	51/15/15051	Y
	1000	99/35/35099	Y	91/30/30091	T	71/24/24071	Y
Broyden tridiagonal	1	15/8/8015	Y	13/7/7013	Y	13/7/7013	Y
	10	21/11/11021	Y	21/11/11021	Y	19/10/10019	Y
	100	27/14/14027	Y	25/13/13025	Y	25/13/13025	Y
	1000	31/16/16031	Y	31/16/16031	Y	29/15/15029	Y
Broyden banded	1	17/9/9017	Y	17/9/9017	Y	17/9/9017	Y
	10	27/14/14027	Y	27/14/14027	Y	27/14/14027	Y
	100	35/18/18035	Y	35/18/18035	Y	35/18/18035	Y
	1000	43/22/22043	Y	43/22/22043	Y	43/22/22043	Y