Intelligent Control and Automation, **2010, 1, 28-35**
doi:10.4236/ica.201.11004 Published Online August 2010 (http://www.SciRP.org/journal/ica)

A Nonmonotone Line Search Method for Symmetric Nonlinear Equations*

Gonglin Yuan[1], Laisheng Yu[2]
[1]*College of Mathematics and Information Science, Guangxi University, Nanning, China*
[2]*Student Affairs Office, Guangxi University, Nanning, China*
E-mail: *glyuan@gxu.edu.cn*
Received February 7, 2010; *revised March* 21, 2010; *accepted July* 7, 2010

Abstract

In this paper, we propose a new method which based on the nonmonotone line search technique for solving symmetric nonlinear equations. The method can ensure that the search direction is descent for the norm function. Under suitable conditions, the global convergence of the method is proved. Numerical results show that the presented method is practicable for the test problems.

Keywords: Nonmonotone Line Search, Symmetric Equations, Global Convergence

1. Introduction

Consider the following nonlinear equations:

$$g\left(x\right)=0, x\in R^{n} \tag{1}$$

where $g: R^n \to R^n$ be continuously differentiable and its Jacobian $\nabla g(x)$ is symmetric for all $x \in R^n$. This problem can come from unconstrained optimization problems, a saddle point problem, and equality constrained problems [the detail see [1]]. Let $\phi(x)$ be the norm function defined by $\phi(x)=\frac{1}{2}\|g(x)\|^2$. Then the nonlinear equation problem (1) is equivalent to the following global optimization problem

$$\min \phi\left(x\right),\ x\in R^{n} \tag{2}$$

The following iterative formula is often used to solve (1) and (2):

$$x_{k+1}=x_k+\alpha_k d_k$$

where α_k is a steplength, and d_k is one search direction. To begin with, we briefly review some methods for (1) and (2) by line search technique. First, we give some techniques for α_k. Li and Fukashima [1] proposed an approximate monotone line search technique to obtain the step-size α_k satisfying:

$$\begin{aligned}&\varphi\left(x_k+\alpha_k d_k\right)-\varphi\left(x\right)\\&\le -\delta_1\left\|\alpha_k d_k\right\|^2-\delta_2\left\|\alpha_k g_k\right\|^2+\varepsilon_k\left\|g_k\right\|^2\end{aligned} \tag{3}$$

where $\delta_1 > 0$ and $\delta_2 > 0$ are positive constants, $\alpha_k = r^{i_k}$, $r \in (0,1), i_k$ is the smallest nonnegative integer i such that (3), and ε_k satisfies $\sum_{k=0}^{\infty}\varepsilon_k < \infty$. Combining the line search (3) with one special BFGS update formula, they got some better results (see [1]). Inspired by their idea, Wei [2] and Yuan [3] made a further study. Brown and Saad [4] proposed the following line search method to obtain the stepsize

$$\varphi\left(x_k+\alpha_k d_k\right)-\varphi\left(x_k\right)\le \sigma\alpha_k\nabla\varphi\left(x_k\right)^T d_k \tag{4}$$

where $\sigma \in (0,1)$. Based on this technique, Zhu [5] gave the nonmonotone line search technique:

$$\varphi\left(x_k+\alpha_k d_k\right)-\varphi\left(x_{l(k)}\right)\le \sigma\alpha_k\nabla\varphi\left(x_k\right)^T d_k \tag{5}$$

where $\varphi_{l(k)} = \max_{0\le j\le n(k)}\left\{\varphi_{k-j}\right\}$, $k=0,1,2,\cdots, n(k)=\min\{M,k\}$, and $M \ge 0$ is an integer constant. From these two techniques (4) and (5), it is easy to see that the Jacobian matric $\nabla g(x)$ must be computed at every iteration, which will increase the workload especially for large-scale problems or this matric is expensive. Considering these points, we [6] presented a new backtracking inexact technique to obtain the stepsize α_k

*This work is supported by China NSF grands 10761001, the Scientific Research Foundation of Guangxi University (Grant No. X081082), and Guangxi SF grands 0991028.

$$\left\|g\left(x_k+\alpha_k d_k\right)\right\|^2-\left\|g\left(x_k\right)\right\|^2 \le \delta\alpha_k^2 g\left(x_k\right)^T d_k \tag{6}$$

where $\delta \in (0,1)$ Second, we present some techniques for d_k. One of the most effective methods is Newton method. It normally requires a fewest number of function evaluations, and it is very good at handling ill-conditioning. However, its efficiency largely depends on the possibility to efficiently solve a linear system which arises when computing the search d_k at each iteration

$$\nabla g\left(x_k\right) d_k = -g\left(x_k\right) \tag{7}$$

Moreover, the exact solution of the system (7) could be too burdensome, or it is not necessary when d_k is far from a solution [7]. Inexact Newton methods [5,7] represent the basic approach underlying most of the Newton-type large-scale algorithms. At each iteration, the current estimate of the solution is updated by approximately solving the linear system (7) using an iterative algorithm. The inner iteration is typically "truncated" before the solution to the linear system is obtained. Griewank [8] firstly proposed the Broyden's rank one method for nonlinear equations and obtained the global convergence. At present, a lot of algorithms have been proposed for solving these two problems (1) and (2) (see [9-15]). The famous BFGS formula is one of the most effective quasi-Newton methods, where the d_k is the solution of the system of linear equations

$$B_k d_k + g_k = 0 \tag{8}$$

where B_k is generated by the following BFGS update formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k} \tag{9}$$

where $s_k = x_{k+1} - x_k$ and $y_k = g(x_{k+1}) - g(x_k)$. Recently, there are some results on nonlinear equations can be found at [6,17-20].

Zhang and Hager [21] present a nonmonotone line search technique for unconstrained optimization problem $\min_{x\in R^n} f(x)$, where the nonmonotone line search technique is defined by

$$f\left(x_k+\alpha_k d_k\right) \le C_k + \delta\alpha_k \nabla f\left(x_k\right)^T d_k,$$
$$\nabla f\left(x_k+\alpha_k d_k\right)^T d_k \ge \sigma \nabla f\left(x_k\right)^T d_k$$

where

$$0<\delta<\sigma<1,\ C_0 = f\left(x_0\right),\ Q_0 = 1,$$
$$Q_{k+1} = \eta_k Q_k + 1,\ C_{k+1} = \frac{\eta_k Q_k C_k + f\left(x_k+\alpha_k d_k\right)}{Q_{k+1}}$$

$\eta_k \in [\eta_{\min}, \eta_{\max}]$, and $0 \le \eta_{\min} \le \eta_{\max} \le 1$. They proved the global convergence for nonconvex, smooth functions, and R-linear convergence for strongly convex functions. Numerical results show that this method is more effective than other similar methods. Motivated by their technique, we propose a new nonmonotone line search technique which can ensure the descent search direction on the norm function for solving symmetric nonlinear Equations (1) and prove the global convergence of our method. The numerical results are reported too. Here and throughout this paper, ||.|| denote the Euclidian norm of vectors or its induced matrix norm.

This paper is organized as follows. In the next section, we will give our algorithm for (2). The global convergence and the numerical result are established in Section 3 and in Section 4, respectively.

2. The Algorithm

Precisely, our algorithm is stated as follows.

Algorithm 1.

Step 0: Choose an initial point $x_0 \in R^n$, an initial symmetric positive definite matrix $B_0 \in R^{n\times n}$, and constants $r \in (0,1), 0 \le \rho_1 \le \rho_2 \le 1, 0 < \delta_1 < 1, J_0 = \left\|g_0\right\|^2, E_0 = 1,$ and $k = 0;$

Step 1: If $g_k = 0;$ then stop; Otherwise, solving the following linear Equations (10) to obtain d_k and go to step 2;

$$B_k d_k + g_k = 0 \tag{10}$$

Step 2: Let i_k be the smallest nonnegative integer i such that

$$\left\|g\left(x_k+\lambda_k d_k\right)\right\|^2-\left\|g\left(x_k\right)\right\|^2 \le \delta\lambda_k^2 g\left(x_k\right)^T d_k \tag{11}$$

holds for $\lambda = r^i$. Let $\lambda_k = r^{i_k}$;

Step 3: Let $x_{k+1} = x_k + \alpha_k d_k$, $s_k = x_{k+1} - x_k$ and $y_k = g\left(x_{k+1}\right) - g\left(x_k\right)$. If $y_k^T s_k > 0$, update B_k to generate B_{k+1} by the BFGS formula (9). Otherwise, let $B_{k+1} = B_k$;

Step 4: Choose $\rho_k \in [0,1]$, and set

$$E_{k+1} = \rho_k E_k + 1,\ J_{k+1} = \frac{\rho_k E_k J_k + \left\|g\left(x_{k+1}\right)\right\|^2}{E_{k+1}} \tag{12}$$

Step 5: Let $k := k + 1$, go to Step 1.

Remark 1: 1) By the technique of the step 3 in the algorithm [see [1]], we deduce that B_{k+1} can inherits the positive and symmetric properties of B_k. Then, it is not difficult to get $d_k^T g_k < 0$.

2) It is easy to know that J_{k+1} is a convex combination of J_k and $\|g(x_{k+1})\|^2$. By $J_0 = \|g_0\|^2$, it follows that J_k is a convex combination of function values $\|g_0\|^2, \|g_1\|^2, \ldots, \|g_k\|^2$. The choice of ρ_k controls the degree of nonmonotonicity. If $\rho_k = 0$ for each k, then the line search is the usual monotone line search. If $\rho_k = 1$ for each k, then $V_k = \frac{1}{k+1}\sum_{i=0}^{k}\|g_i\|^2$ $J_k = V_k$, where is the average function value.

3) By (9), we have $B_{k+1}s_k = y_k = g_{k+1} - g_k \approx \nabla g_{k+1}s_k = \nabla g_{k+1}^T s_k$, this means that B_{k+1} approximate to ∇g_{k+1} along s_k.

3. The Global Convergence Analysis of Algorithm 1

In this section, we establish global convergence for Algorithm 1. The level set Ω is defined by

$$\Omega = \left\{x \in R^n \mid \|g(x) \le g(x_0)\|\right\}.$$

Assumption A. The Jaconbian of g is symmetric and there exists a constant $M > 0$ holds

$$\|g(x) - g(x_k)\| \le M\|x - x_k\| \tag{13}$$

for $x \in \Omega.$.

Since B_k approximates ∇g_k along direction s_k, we can give the following assumption.

Assumption B. B_k is a good approximation to ∇g_k, *i.e.*,

$$\|(\nabla g(x_k) - B_k)d_k\| \le \varepsilon\|g_k\| \tag{14}$$

where $\varepsilon \in (0,1)$ is a small quantity.

Assumption C. There exist positive constants b_1 and b_2 satisfy

$$g_k^T d_k \le -b_1\|g_k\|^2 \tag{15}$$

and

$$\|d_k\| \le b_2\|g_k\| \tag{16}$$

for all sufficiently large *k*..

By (10) and Assumption C, we have

$$b_1\|g_k\| \le \|d_k\| \le b_2\|g_k\| \tag{17}$$

Lemma 3.1. Let Assumption B hold and $\{\alpha_k, d_k, x_{k+1}, g_{k+1}\}$. be generated by Algorithm 1. Then d_k is descent direction for $\varphi(x)$ at x_k, *i.e.*,

$$\nabla\varphi(x_k)^T d_k < 0 \tag{18}$$

Proof. By (10), we have

$$\begin{aligned}\nabla\varphi(x_k)^T d_k &= g_k^T\nabla g_k d_k = g_k^T\left[(\nabla g_k d_k - B_k)d_k - g_k\right] \\ &= g_k^T(\nabla g_k d_k - B_k)d_k - g_k^T g_k\end{aligned} \tag{19}$$

Using (14) and taking the norm in the right-hand-side of (19), we get

$$\begin{aligned}\nabla\varphi(x_k)^T d_k &\le \left\|g_k^T(\nabla g_k d_k - B_k)d_k\right\| - \|g_k\|^2 \\ &\le -(1-\varepsilon)\|g_k\|^2\end{aligned} \tag{20}$$

Therefore, for $\varepsilon \in (0,1)$, we get the lemma.

By the above lemma, we know that the norm function $\varphi(x)$ is descent along d_k, then $\|g_{k+1}\| \le \|g_k\|$ holds.

Lemma 3.2. Let Assumption B hold and $\{\alpha_k, d_k, x_{k+1}, g_{k+1}\}$ be generated by Algorithm 1. Then $\{x_k\} \subset \Omega$. Moreover, $\{\|g_k\|\}$ converges.

Proof. By Lemma 3.1, we get $\|g_{k+1}\| \le \|g_k\|$. Then, we conclude that $\{\|g_k\|\}$ converges. Moreover, we have for all *k*

$$\|g_{k+1}\| \le \|g_k\| \le \cdots \le \|g_0\|.$$

Which means that $\{x_k\} \subset \Omega$.

The next lemma will show that for any choice of $\rho_k \in [0,1), J_k$ lies between $\|g_k\|^2$ and V_k.

Lemma 3.3. Let $\{\alpha_k, d_k, x_{k+1}, g_{k+1}\}$ be generated by Algorithm 1, we have $\|g_k\|^2 \le J_k \le V_k, J_{k+1} \le J_k$ for each *k*.

Proof. We will prove the lower bound for J_k by induction. For *k* = 0, by the initialization $J_0 = \|g_0\|^2$, this holds. Now we assume that $J_i \ge \|g_i\|^2$ holds for all $0 \le i \le k$. By (2.3) and $\|g_{i+1}\|^2 \le \|g_i\|^2$, we have

$$\begin{aligned}J_{i+1} &= \frac{\rho_i E_i J_i + \|g_{i+1}\|^2}{E_{i+1}} \ge \frac{\rho_i E_i\|g_i\|^2 + \|g_{i+1}\|^2}{E_{i+1}} \\ &\ge \frac{\rho_i E_i\|g_{i+1}\|^2 + \|g_{i+1}\|^2}{E_{i+1}} = \|g_{i+1}\|^2\end{aligned} \tag{21}$$

where $E_{i+1} = \rho_i E_i + 1$. Now we prove that $J_{k+1} \le J_k$ is true. By (12) again, and using $\|g_k\|^2 \ge \|g_{k+1}\|^2$, we obtain

$$J_{k+1} = \frac{\rho_k E_k J_k + \|g_{k+1}\|^2}{E_{k+1}} \le \frac{\rho_k E_k J_k + J_{k+1}}{E_{k+1}}.$$

Which means that $J_{k+1} \le J_k$ for all k is satisfied. Then we have

$$\|g_k\|^2 \le J_k \le J_{k-1} \tag{22}$$

Let $L_k : R \to R$ be defined by

$$L_k(t) = \frac{tJ_{k-1} + \|g_k\|^2}{t+1},$$

we can get

$$L_k'(t) = \frac{J_{k-1} + \|g_k\|^2}{(t+1)^2}.$$

By $J_{k-1} \le \|g_k\|^2$, we obtain $L_k'(t) \ge 0$ for all $t \ge 0$. Then, L_k is nondecreasing, and $\|g_k\|^2 = L_k(0) \le L_k(t)$ for all $t \ge 0$. Now we prove the upper bound $J_k \le V_k$ by induction. For $k = 0$, by the initialization $J_0 = \|g_0\|^2$, this holds. Now assume that $J_j \le V_j$ hold for all $0 \le j \le k$. By using $E_0 = 1$, (12), and $\rho_k \in [0,1]$, we obtain

$$E_{j+1} = 1 + \sum_{i=0}^{j} \prod_{p=0}^{i} \rho_{j-p} \le j+2 \tag{23}$$

Denote that L_k is monotone nondecreasing, (23) implies that

$$J_k = L_k(\rho_{k-1}E_{k-1}) = L(E_k - 1) \le L_k(k) \tag{24}$$

Using the induction step, we have

$$L_k(k) = \frac{kJ_{k-1} + \|g_k\|^2}{k+1} \le \frac{kV_{k-1} + \|g_k\|^2}{k+1} = V_k \tag{25}$$

Combining (24) and (25) implies the upper bound of J_k in this lemma. Therefore, we get the result of this lemma.

The following lemma implies that the line search technique is well-defined.

Lemma 3.4. Let Assumption A, B and C hold. Then Algorithm 1 will produce an iterate $x_{k+1} = x_k + \alpha_k d_k$ in a finite number of backtracking steps.

Proof. From Lemma 3.8 in [4] we have that in a finite number of backtracking steps, α_k must satisfy

$$\|g(x_k + \alpha_k d_k)\|^2 - \|g(x_k)\|^2 \le \sigma \alpha_k g(x_k)^T \nabla g_k d_k \tag{26}$$

where $\sigma \in (0,1)$. By (20) and (15), we get

$$\begin{aligned} &\alpha_k g(x_k)^T \nabla g_k d_k \le -\alpha_k (1-\varepsilon)\|g_k\|^2 \\ &= -\alpha_k (1-\varepsilon)\frac{g_k^T d_k}{g_k^T d_k}\|g_k\|^2 \le \alpha_k (1-\varepsilon)\frac{1}{b_1} g_k^T d_k \end{aligned} \tag{27}$$

Using $\alpha_k \in (0,1)$, we obtain

$$\begin{aligned} &\alpha_k g(x_k)^T \nabla g_k d_k \le \alpha_k (1-\varepsilon)\frac{1}{b_1} g_k^T d_k \\ &\le \alpha_k^2 (1-\varepsilon)\frac{1}{b_1} g_k^T d_k \end{aligned} \tag{28}$$

So let $\delta_1 \in \left(0, \min\left(1, \sigma(1-\varepsilon)\frac{1}{b_1}\right)\right)$. By Lemma 3.3, we know $\|g_k\|^2 \le J_k$. Therefore, we get the line search (11). The proof is complete.

Lemma 3.5. Let Assumption A, B and C hold. Then we have the following estimate for α_k, when *k* sufficiently large:

$$\alpha_k \ge b_0 > 0 \tag{29}$$

Proof. Assuming the step-size α_k such that (11). Then $\alpha_k' = \alpha_k / 1$ does not satisfy (11), *i.e.*,

$$\left\|g\left(x_k + \alpha_k' d_k\right)\right\|^2 - J_k > \delta_1 \alpha_k'^2 g(x_k)^T d_k$$

By $\|g_k\|^2 \le J_k$, we get

$$\begin{aligned} &\left\|g\left(x_k + \alpha_k' d_k\right)\right\|^2 - \|g_k\|^2 \\ &\ge \left\|g\left(x_k + \alpha_k' d_k\right)\right\|^2 - J_k > \delta_1 \alpha_k'^2 g(x_k)^T d_k \end{aligned}$$

Which implies that

$$-\delta_1 \alpha_k'^2 g(x_k)^T d_k > \|g(x_k + \alpha_k' d_k)\|^2 - \|g_k\|^2 \tag{30}$$

By Taylor formula, (19), (20), and (17), we get

$$\begin{aligned} &\left\|g\left(x_k + \alpha_k' d_k\right)\right\|^2 - \|g_k\|^2 = -2\alpha_k' g_k^T \nabla g_k d_k \\ &+ O\left(\alpha_k'^2 \|d_k\|^2\right) \ge \alpha_k' (1-\varepsilon)\|g_k\|^2 + O\left(\alpha_k'^2 \|d_k\|^2\right) \\ &\ge \alpha_k' (1-\varepsilon)\frac{1}{b_2^2}\|d_k\|^2 + O\left(\alpha_k'^2 \|d_k\|^2\right) \end{aligned} \tag{31}$$

Using (15), (17), (30), and (31) we obtain

$$\begin{aligned} &\alpha_k'^2 \left(2(1-\varepsilon)\frac{1}{b_2^2} + \delta_1 \frac{1}{b_1}\right)\|d_k\|^2 \\ &= 2\alpha_k'^2 (1-\varepsilon)\frac{1}{b_2^2}\|d_k\|^2 + \alpha_k'^2 \delta_1 \frac{1}{b_1}\|d_k\|^2 \\ &\ge 2\alpha_k'^2 (1-\varepsilon)\frac{1}{b_2^2}\|d_k\|^2 - \delta_1 \alpha_k'^2 g_k^T d_k \\ &> \left\|g\left(x_k + \alpha_k' d_k\right)\right\|^2 - \|g_k\|^2 \\ &\ge \alpha_k' (1-\varepsilon)\frac{1}{b_2^2}\|d_k\|^2 + O\left(\alpha_k'^2 \|d_k\|^2\right) \end{aligned} \tag{32}$$

which implies when k sufficiently large,

$$\alpha_k' \geq \frac{b_1(1-\varepsilon)}{2(1-\varepsilon)b_1 + b_2^2\delta_1}.$$

Let

$b_0 \in \left(0, \dfrac{b_1(1-\varepsilon)}{2(1-\varepsilon)b_1 + b_2^2\delta_1}\right)$. The proof is complete.

In the following, we give the global convergence theorem.

Theorem 3.1. Let $\{\alpha_k, d_k, x_{k+1}, g_{k+1}\}$ be generated by Algorithm 1, Assumption A, B, and C hold, and $\|g_k\|^2$ be bounded from below. Then

$$\liminf_{k\to\infty} \|g_k\| = 0 \tag{33}$$

Moreover, if $\rho_2 < 1$, then

$$\lim_{k\to\infty} \|g_k\| = 0 \tag{34}$$

Therefore, every convergent subsequence approaches to a point x^*, where $g(x^*) = 0$.

Proof. By (11), (15), (16), and (19), we have

$$\begin{aligned}\|g_{k+1}\|^2 &\leq J_k + \delta_1\alpha_k d_k^T g_k \leq J_k - \alpha_k\delta_1 b_1 \|g_k\|^2 \\ &\leq J_k - b_0\delta_1 b_1 \|g_k\|^2\end{aligned} \tag{35}$$

Let $\varsigma = \delta_1 b_0 b_1$. Combining (12) and the upper bound of (35), we get

$$\begin{aligned}J_{k+1} &= \frac{\rho_k E_k J_k + \|g_{k+1}\|^2}{E_{k+1}} \leq \frac{\rho_k E_k J_k + J_k - \varsigma\|g_k\|^2}{E_{k+1}} \\ &\leq J_k - \frac{\varsigma\|g_{k+1}\|^2}{E_{k+1}}\end{aligned} \tag{36}$$

Since $\|g_k\|^2$ is bounded from below and $\|g_k\|^2 \leq J_k$ for all *k*, we can conclude that J_k is bounded from below. Then, using (36), we obtain

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^2}{E_{k+1}} < \infty \tag{37}$$

By (23), we get

$$E_{k+1} < k + 2 \tag{38}$$

If $\|g_k\|^2$ were bounded away from 0, then (37) would violate (38). Hence, (33) holds. If $\rho_2 < 1$, by (23), we have

$$\begin{aligned}E_{k+1} &= 1 + \sum_{j=0}^{k}\prod_{i=0}^{j} \rho_{k-i} \leq 1 + \sum_{j=0}^{k} \rho_2^{j+1} \\ &\leq \sum_{j=0}^{k} \rho_2 = \frac{1}{1-\rho_2}\end{aligned} \tag{39}$$

Then, (37) implies (34). The proof is complete.

4. Numerical Results

In this section, we report the results of some numerical experiments with the proposed method.

Problem 1. The discretized two-point boundary value problem is the same to the problem in [22]

$$g(x) = Ax + \frac{1}{(n+1)^2} F(x),$$

where A is the $n \times n$ tridiagonal matrix given by

$$A = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 4 \end{bmatrix}$$

and $F(x) = (F_1(x), F_2(x), \cdots F_n(x))^T$, with $F_i(x) = \sin x_i - 1$, $i = 1, 2, \cdots n$.

Problem 2. Unconstrained optimization problem $\min f(x), x \in R^n$, with Engval function [23] defined by

$$f(x) = \sum_{i=2}^{n} \left[\left(x_{i-1}^2 + x_i^2\right)^2 - 4x_{i-1} + 3 \right]$$

The related symmetric nonlinear equation is

$$g(x) = \tfrac{1}{4}\nabla f(x)$$

where $g(x) = (g_1(x), g_2(x), \cdots, g_n(x))^T$ with

$$\begin{aligned} g_1(x) &= x_1\left(x_1^2 + x_2^2\right) - 1 \\ g_i(x) &= x_i\left(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2\right) - 1, i = 2, 3, \cdots, n-1 \\ g_n(x) &= x_n\left(x_{n-1}^2 + x_n^2\right) \end{aligned}$$

In the experiments, the parameters in Algorithm 1 were chosen as $r = 0.1, \delta_1 = 0.001, \rho_k = 0.8, B_0$ is unit matrix. The program was coded in MATLAB 6.1. We stopped the iteration when the condition $\|g(x)\|^2 \leq 10^{-6}$ was satisfied. **Tables 1** and **2** show the performance of the method need to solve the Problem 1. **Tables 3** and **4** show the performance of the method need to solve the Problem 2. The columns of the tables have the following meaning:

Dim: the dimension of the problem.

NI: the total number of iterations.

NG: the number of the function evaluations.

GG: the function evaluations.

From the above tabulars, we can see that the numerical

Test Result for the Problem 1

Table 1. Small-scale.

x_0	(4, ..., 4)	(20, ..., 20)	(100, ..., 100)	(-4, ..., -4)	(-20, ..., -20)	(-100, ..., -100)
Dim	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG
n = 10	16/22/3.714814e-7	16/20/2.650667e-7	16/20/4.014933e-7	16/22/3.723721e-7	16/20/2.643713e-7	16/20/4.013770e-7
n = 50	44/47/1.388672e-7	44/47/6.929395e	46/49/3.713174e-8	44/47/1.388793e-7	44/47/6.929516e-7	46/49/3.726373e-8
n = 100	68/71/5.905592e-7	70/73/8.759459e	72/75/3.125373e-7	68/71/5.905724e-7	70/73/8.759500e-7	72/75/3.125382e-7
x_0	(4, .0, ...,)	(20, 0, ..., 20)	(100, .0, ...,)	(-4, .0, ...,)	(-20, .0, ...,)	(-100, .0, ...,)
Dim	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG
n = 10	21/23/1.297275e-9	21/23/5.577021e-9	1/23/9.029402e-9	1/23/1.208563e-9	1/23/4.707707e-9	21/23/1.061736e-8
n = 50	63/65/8.204623e-7	67/69/9.997988e-7	9/71/4.511023e-7	3/65/8.204744e-7	7/69/9.997996e-7	69/71/4.511007e-7
n = 100	65/67/6.046233e-7	69/71/7.845951e-7	1/73/6.996085e-7	65/67/6.046254e-7	9/71/7.845962e-7	71/73/6.996092e-7

Table 2. Large-scale.

x_0	(4, ..., 4)	(20, ..., 20)	(30, ..., 30)	(-4, ..., -4)	(-20, ..., -20)	(-30, ..., -30)
Dim	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG
n = 300	70/73/7.844778e-7	76/79/7.741702e-7	8/81/6.759628e-7	0/73/7.844800e-7	6/79/7.741706e-7	78/81/6.759631e-7
n = 500	70/73/8.547195e-7	76/79/8.435874e-7	8/81/7.366072e-7	0/73/8.547204e-7	6/79/8.435876e-7	78/81/7.366073e-7
n = 800	68/70/6.505423e-7	74/76/6.414077e-7	4/76/9.621120e-7	8/70/6.505425e-7	4/76/6.414078e-7	74/76/9.621120e-7
x_0	(4, .0, ...,)	(20, 0, ..., 20)	(30, .0, ...,)	(-4, .0, ...,)	(-20, .0, ...,)	(-30, .0, ...,)
Dim	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG
n = 300	67/69/5.896038e-7	71/73/9.997625e-7	3/75/8.731533e-7	67/69/5.896038e-7	71/73/9.997625e-7	73/75/8.731533e-7
n = 500	67/69/7.145027e-7	73/75/7.057076e-7	5/77/6.163480e-7	7/69/7.145024e-7	73/75/7.057075e-7	75/77/6.163479e-7
n = 800	69/71/6.188110e-7	75/77/6.115054e-7	5/77/9.172559e-7	9/71/6.188106e-7	5/77/6.115053e-7	75/77/9.172558e-7

Test Result for the Problem 2

Table 3. Small-scale.

x_0	(1, ..., 1)	(3, ..., 3)	(4, ..., 4)	(1, 0, ...,)	(3, 0, , ...,)	(4, 0, ...,)
Dim	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG
n = 10	20/22/3.007469e-7	38/47/6.088293e-7	44/48/4.898591e-7	0/23/3.452856e-7	5/41/5.833715e-7	29/34/4.338894e-7
n = 50	36/38/6.966974e-7	76/88/6.845101e-7	99/114/8.556270e-7	6/39/7.812438e-7	9/77/4.466497e-7	67/75/4.681269e-7
n = 100	36/38/7.207203e-7	6/109/4.173166e-7	0/87/7.911692e-7	6/39/8.220367e-7	9/92/8.640158e-7	69/76/8.515673e-7

Table 4. Large-scale.

x_0	(1, ..., 1)	(3, ..., 3)	(4, ..., 4)	(1, 0, ...,)	(3, 0, ...,)	(4, 0, ...,)
Dim	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG
n = 300	40/42/4.452904e-7	4/106/6.146797e-7	63/68/4.232021e-7	0/43/9.348673e-7	6/72/7.638011e-7	92/106/9.095927e-
n = 500	44/46/9.611950e-7	9/171/4.445314e-7	18/133/7.054347-7e	2/45/8.280486e-7	3/79/7.159029e-7	85/98/4.229872e-7
n = 800	41/43/4.510999e-7	7/185/5.274922e-7	74/198/4.239839e-7	1/44/9.502624e-7	2/77/6.117626e-7	93/106/8.797380e-7

results are quite well for the test Problems with the proposed method. The initial points and the dimension don't influence the performance of the algorithm 1 very much. However, we find the started points will influence the result for the problem 2 a little in our experiment. In one word, the numerical are attractively. The method can be used to the system of nonlinear equations whose Jacobian is not symmetric.

5. Conclusion

In this paper, we propose a new nonmonotone line search method for symmetric nonlinear equations. The global convergence is proved and the numerical results show that this technique is interesting. The reason is that the new nonmonotone line search algorithm used fewer function and gradient evaluations, on average, than either the monotone or the traditional nonmonotone scheme. We hope the method will be a further topic for symmetric nonlinear equations.

6. Acknowledgements

We would like to thank these referees for giving us many valuable suggestions and comments that improve this paper greatly.

7. References

[1] D. Li and M. Fukushima, "A Global And Superlinear Convergent Gauss-Newton-based BFGS Method for Symmetric Nonlinear Equations," *SIAM Journal on Numerical Analysis*, Vol. 37, No. 1, 1999, pp. 152-172.

[2] Z. Wei, G. Yuan and Z. Lian, "An Approximate Gauss-Newton-Based BFGS Method for Solving Symmetric Nonlinear Equations," *Guangxi Sciences*, Vol. 11, No. 2, 2004, pp. 91-99.

[3] G. Yuan and X. Li, "An Approximate Gauss-Newton-based BFGS Method with Descent Directions for Solving Symmetric Nonlinear Equations," *OR Transactions*, Vol. 8, No. 4, 2004, pp. 10-26.

[4] P. N. Brown and Y. Saad, "Convergence Theory of Nonlinear Newton-Krylov Algorithms," *SIAM Journal on Optimization*, Vol. 4, 1994, pp. 297-330.

[5] D. Zhu, "Nonmonotone Backtracking Inexact Quasi-Newton Algorithms for Solving Smooth Nonlinear Equations," *Applied Mathematics and Computation*, Vol. 161, No. 3, 2005, pp. 875-895.

[6] G. Yuan and X. Lu, "A New Backtracking Inexact BFGS Method for Symmetric Nonlinear Equations," *Computers and Mathematics with Applications*, Vol. 55, No. 1, 2008, pp. 116-129.

[7] S. G. Nash, "A Survey of Truncated-Newton Methods," *Journal of Computational and Applied Mathematics*, Vol. 124, No. 1-2, 2000, pp. 45-59.

[8] A. Griewank, "The 'Global' Convergence of Broyden-Like Methods with a Suitable Line Search," *Journal of the Australian Mathematical Society Series B*, Vol. 28, No. 1, 1986, pp. 75-92.

[9] A. R. Conn, N. I. M. Gould and P. L. Toint, "Trust Region Method," Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[10] J. E. Dennis and R. B. Schnabel, "Numerical Methods for Unconstrained Optimization and Nonlinear Equations," Englewood Cliffs, Prentice-Hall, 1983.

[11] K. Levenberg, "A Method for The Solution of Certain Nonlinear Problem in Least Squares," *Quarterly of Applied Mathematics*, Vol. 2, 1944, pp. 164-166.

[12] D. W. Marquardt, An algorithm for least-squares estimation of nonlinear inequalities, *SIAM Journal on Applied Mathematics*, Vol. 11, 1963, pp. 431-441.

[13] J. Nocedal and S. J. Wright, "Numerical Optimization," Springer, New York, 1999.

[14] N. Yamashita and M. Fukushima, "On the Rate of Convergence of the Levenberg-Marquardt Method," *Computing*, Vol. 15, No. Suppl, 2001, pp. 239-249.

[15] Y. Yuan and W. Sun, "Optimization Theory and Algorithm," Scientific Publisher House, Beijing, 1997.

[16] G. Yuan and X. Li, "A Rank-One Fitting Method for Solving Symmetric Nonlinear Equations," *Journal of Applied Functional Analysis*, Vol. 5, No. 4, 2010, pp. 389-407.

[17] G. Yuan, X. Lu and Z. Wei, "BFGS Trust-Region Method for Symmetric Nonlinear Equations," *Journal of Computational and Applied Mathematics*, Vol. 230, No. 1, 2009, pp. 44-58.

[18] G. Yuan, S. Meng and Z. Wei, "A Trust-Region-Based BFGS Method with Line Search Technique for Symmetric Nonlinear Equations," *Advances in Operations Research*, Vol. 2009, 2009, pp. 1-20.

[19] G. Yuan, Z. Wang and Z. Wei, "A Rank-One Fitting Method with Descent Direction for Solving Symmetric Nonlinear Equations," *International Journal of Communications, Network and System Sciences*, Vol. 2, No. 6, 2009, pp. 555-561.

[20] G. Yuan, Z. Wei and X. Lu, "A Nonmonotone Trust Region Method for Solving Symmetric Nonlinear Equations," *Chinese Quarterly Journal of Mathematics*, Vol. 24, No. 4, 2009, pp. 574-584.

[21] H. Zhang and W. Hager, "A Nonmonotone Line Search Technique and Its Application to Unconstrained Optimization," *SIAM Journal on Optimization*, Vol. 14, No. 4, 2004, pp. 1043-1056.

[22] J. M. Ortega and W. C. Rheinboldt, "Iterative Solution of Nonlinear Equations in Several Variables," Academic Press, New York, 1970.

[23] E. Yamakawa and M. Fukushima, "Testing Parallel Bariable Transformation," *Computational Optimization and Applications*, Vol. 13, 1999, pp. 253-274.

