Available online at www.sciencedirect.com
ScienceDirect
Computers and Mathematics with Applications 55 (2008) 116–129
An International Journal computers & mathematics with applications
www.elsevier.com/locate/camwa

A new backtracking inexact BFGS method for symmetric nonlinear equations☆

Gonglin Yuan*, Xiwen Lu

School of Science, East China University of Science and Technology, Shanghai, 200237, PR China

Received 9 November 2005; received in revised form 18 September 2006; accepted 12 December 2006

Abstract

A BFGS method, in association with a new backtracking line search technique, is presented for solving symmetric nonlinear equations. The global and superlinear convergences of the given method are established under mild conditions. Preliminary numerical results show that the proposed method is better than the normal technique for the given problems.

Keywords: BFGS method; Line search; Symmetric nonlinear equations; Global convergence; Superlinear convergence

1. Introduction

It's well known that the quasi-Newton methods (see [1–7]) are very important methods for solving the unconstrained optimization problems $\min_{x\in\Re^n} f(x)$, and some modified BFGS methods with global and superlinear convergence have been proposed in [8–12] etc. For nonlinear equations, some techniques have been given [13–17].

In this paper, we consider the following system of nonlinear equations

$$g(x) = 0, \quad x \in \Re^n \tag{1.1}$$

where $g : \Re^n \to \Re^n$ is continuously differentiable, and the Jacobian $\nabla g(x)$ of g is symmetric for all $x \in \Re^n$. In fact, this problem can come from an unconstrained optimization problem, a saddle point problem, and equality constrained problems [14]. Let θ be the norm function defined by $\theta(x) = \frac{1}{2}\|g(x)\|^2$. Then the nonlinear equations problem (1.1) is equivalent to the following global optimization problem

$$\min \theta(x), \quad x \in \Re^n. \tag{1.2}$$

The BFGS method for solving (1.1) is to generate a sequence of iterates $\{x_k\}$ by letting $x_{k+1} = x_k + \alpha_k d_k$, where α_k is a steplength, and d_k is a solution of the system of linear equations

$$B_k d_k + g_k = 0, \tag{1.3}$$

☆ This work is supported by Guangxi University SF grands X061041.

* Corresponding author.
E-mail addresses: glyuan@tom.com (G. Yuan), xwlu@ecust.edu.cn (X. Lu).

0898-1221/$ - see front matter
doi:10.1016/j.camwa.2006.12.081

where $g_k = g(x_k)$, B_k is generated by the following BFGS update formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^{\mathrm{T}} B_k}{s_k^{\mathrm{T}} B_k s_k} + \frac{y_k {y_k}^{\mathrm{T}}}{{y_k}^{\mathrm{T}} s_k}, \tag{1.4}$$

where $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$. Generally, one method is applied to find a steplength α_k such that

$$\|g(x_k + \alpha_k d_k)\|^2 \le \|g(x_k)\|^2 + \sigma \alpha_k g_k^{\mathrm{T}} \nabla g_k^{\mathrm{T}} d_k, \tag{1.5}$$

where $\sigma \in (0, 1)$ is a given constant. The drawback of the technique (1.5) is the need to compute the Jacobian matrix $\nabla g(x)$ at every iteration, which will increase the computing difficulty, especially for the large-scale problems. In order to avoid computing the Jacobian matrix $\nabla g(x)$ when we find the steplength α_k, the following line search technique is used to get α_k

$$\|g(x_k + \alpha_k d_k)\|^2 \le \|g(x_k)\|^2 + \delta \alpha_k^2 g_k^{\mathrm{T}} d_k, \tag{1.6}$$

where $\delta \in (0, 1)$. In Section 3; we will show that (1.6) is reasonable.

The purpose of this paper is to propose a BFGS method with the above line search technique. The presented method has a norm descent property, whose global and superlinear convergence will be given under suitable conditions. Numerical results show that the method is very interesting.

This paper is organized as follows. In the next section, the backtracking inexact BFGS algorithm is stated. Under some reasonable conditions, we establish the global and superlinear convergence of the algorithms in Section 3 and in Section 4, respectively. Preliminary numerical results are proposed in Section 5.

2. Algorithms

This section will give the inexact BFGS method in association with the new backtracking line search technique (1.6) for (1.1). The 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 $r, \delta, \rho \in (0, 1)$, let $k := 0$.

Step 1: Stop if $\|g_k\| = 0$. Otherwise solve the following linear equation to get d_k

$$B_k d + g_k = 0. \tag{2.1}$$

Step 2: If

$$\|g(x_k + d_k)\| \le \rho \|g(x_k)\|, \tag{2.2}$$

Then take $\alpha_k = 1$ and go to Step 4. Otherwise go to Step 3.

Step 3: Let i_k be the smallest nonnegative integer i such that (1.6) holds for $\alpha = r^i$. Let $\alpha_k = r^{i_k}$.

Step 4: Let the next iteration be $x_{k+1} = x_k + \alpha_k d_k$.

Step 5: Put $s_k = x_{k+1} - x_k = \alpha_k d_k$, $y_k = g_{k+1} - g_k$. If $s_k^{\mathrm{T}} y_k > 0$, update B_k by (1.4), otherwise let $B_{k+1} = B_k$.

Step 6: Let $k := k + 1$. Go to step 1.

We also give an algorithm based on the line search technique (1.5) for (1.1).

Algorithm 2. δ and (1.6) in the Step 0 and Step 3 of Algorithm 1 are replaced by: $\sigma \in (0, 1)$ and (1.5), respectively.

From Algorithm 1, we have

Remark a. (i) By $y_k = g_{k+1} - g_k$, we have the approximate relations

$$y_k = g_{k+1} - g_k \approx \nabla g_{k+1} s_k.$$

Since B_{k+1} satisfies the secant equation $B_{k+1} s_k = y_k$ and ∇g_k is symmetric, we have approximately

$$B_{k+1} s_k \approx \nabla g_{k+1} s_k = \nabla g_{k+1}^{\mathrm{T}} s_k.$$

This means that B_{k+1} approximates ∇g_{k+1} along direction s_k.

(ii) The Step 5 of Algorithm 1 can ensure that B_k is always positive definite.
(iii) We call Step 3 *inner circle* in Algorithm 1.

Throughout this paper, we only discuss Algorithm 1. In the following section, we will concentrate on its global convergence.

3. Global convergence

Let Ω be the level set defined by

$$\Omega = \{x \mid \|g(x)\| \le \|g(x_0)\|\}. \tag{3.1}$$

In order to get the global convergence of Algorithm 1, we need the following assumptions.

Assumption A. (i) g is continuously differentiable on an open convex set Ω_1 containing Ω.
(ii) The Jaconbian of g is symmetric, bounded and positive definite on Ω_1, i.e., there exist positive constants $M \ge m > 0$ such that

$$\|\nabla g(x)\| \le M \quad \forall x \in \Omega_1 \tag{3.2}$$

and

$$m\|d\|^2 \le d^{\mathrm{T}} \nabla g(x) d \quad \forall x \in \Omega_1, d \in R^n. \tag{3.3}$$

Remark b. (1) Conditions (ii) in Assumption A imply that there exist constants $M \ge m > 0$ such that

$$m\|d\| \le \|\nabla g(x) d\| \le M\|d\| \quad \forall x \in \Omega_1, d \in R^n, \tag{3.4}$$

$$\frac{1}{M}\|d\| \le \|\nabla g(x)^{-1} d\| \le \frac{1}{m}\|d\| \quad \forall x \in \Omega_1, d \in R^n, \tag{3.5}$$

$$m\|x - y\| \le \|g(x) - g(y)\| \le M\|x - y\| \quad \forall x, y \in \Omega_1. \tag{3.6}$$

In particular, for all $x \in \Omega_1$, we have

$$m\|x - x^*\| \le \|g(x)\| = \|g(x) - g(x^*)\| \le M\|x - x^*\|, \tag{3.7}$$

where x^* stands for the unique solution of (1.1) in Ω_1.

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.,

$$\|(\nabla g_k - B_k) d_k\| \le \epsilon \|g_k\|, \tag{3.8}$$

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

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 a descent direction for $\theta(x)$ at x_k, i.e.,*

$$\nabla \theta(x_k)^{\mathrm{T}} d_k < 0. \tag{3.9}$$

Proof. By (2.1), we have

$$\begin{aligned} \nabla \theta(x_k)^{\mathrm{T}} d_k &= g(x_k)^{\mathrm{T}} \nabla g(x_k) d_k \\ &= g(x_k)^{\mathrm{T}} [(\nabla g(x_k) - B_k) d_k - g(x_k)] \\ &= g(x_k)^{\mathrm{T}} (\nabla g(x_k) - B_k) d_k - g(x_k)^{\mathrm{T}} g(x_k). \end{aligned} \tag{3.10}$$

Therefore, taking the norm on the right-hand-side of (3.10), we have from (3.8) that

$$\nabla \theta(x_k)^{\mathrm{T}} d_k \le \|g(x_k)\| \|(\nabla g(x_k) - B_k) d_k\| - \|g(x_k)\|^2 \le -(1 - \epsilon)\|g(x_k)\|^2. \tag{3.11}$$

Hence, for $\epsilon \in (0, 1)$, this lemma is true. □

By the above lemma, we can deduce that the norm function $\theta(x)$ is descent along d_k, which means that $\|g_{k+1}\| \le \|g_k\|$ is true.

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 have $\|g_{k+1}\| \le \|g_k\|$. Then we conclude from Lemma 3.3 in [19] that $\{\|g_k\|\}$ converges. Moreover, we have for all k

$$\|g_{k+1}\| \le \|g_k\| \le \|g_{k-1}\| \le \cdots \le \|g(x_0)\|.$$

This implies that $\{x_k\} \subset \Omega$. □

Lemma 3.3. *Let Assumption A be satisfied and $\{\alpha_k, d_k, x_{k+1}, g_{k+1}\}$ be generated by Algorithm 1. Then there exists a constant $m_1 > 0$ such that for all k*

$$y_k{}^{\mathrm{T}} s_k \ge m_1 \|s_k\|^2. \tag{3.12}$$

Proof. By the mean-value theorem, we have

$$y_k{}^{\mathrm{T}} s_k = s_k^{\mathrm{T}}(g_{k+1} - g_k) = s_k^{\mathrm{T}} \nabla g(\xi) s_k \ge m\|s_k\|^2, \tag{3.13}$$

where $\xi = x_k + \varsigma_1(x_{k+1} - x_k)$, $\varsigma_1 \in (0, 1)$; the last inequality follows from (3.3). Let $m_1 = m$, we get (3.12). The proof is complete. □

Using $y_k^{\mathrm{T}} s_k \ge m_1\|s_k\|^2 > 0$, B_{k+1} is always generated by the update formula (1.4), and we can deduce that B_{k+1} inherits symmetric and positive definiteness of B_k. Then, (2.1) has a unique solution for each k. By the above lemma and (3.6), we obtain

$$\frac{s_k^{\mathrm{T}} y_k}{\|s_k\|^2} \ge m, \qquad \frac{\|y_k\|^2}{s_k^{\mathrm{T}} y_k} \le \frac{M^2}{m}. \tag{3.14}$$

Lemma 3.4 (*Theorem 2.1 in [1]*). *Let B_k be updated by BFGS formula (1.4), and let B_0 be symmetric and positive definite. For any $k \ge 0$, s_k and y_k such that (3.14). Then there exist positive constants β_1, β_2 and β_3 such that, for any positive integer k'*

$$\beta_2\|d_k\|^2 \le d_k^{\mathrm{T}} B_k d_k \le \beta_3\|d_k\|^2, \qquad \|B_k d_k\| \le \beta_1\|d_k\| \tag{3.15}$$

hold for at least $\lceil k'/2 \rceil$ value of $k \in \{1, 2, \ldots, k'\}$.

According to Lemma 3.4, we can get

$$\beta_2\|d_k\| \le \|B_k d_k\| \le \beta_1\|d_k\| \tag{3.16}$$

and

$$g_k^{\mathrm{T}} d_k = -d_k^{\mathrm{T}} B_k d_k \le -\beta_2\|d_k\|^2, \qquad -\beta_3\|d_k\|^2 \le -d_k^{\mathrm{T}} B_k d_k = g_k^{\mathrm{T}} d_k. \tag{3.17}$$

Lemma 3.5. *Let Assumptions A and B 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 [18], we have that in a finite number of backtracking steps, α_k must satisfy

$$\|g(x_k + \alpha_k d_k)\|^2 - \|g(x_k)\|^2 \le \sigma\alpha_k g(x_k)^{\mathrm{T}} \nabla g(x_k) d_k. \tag{3.18}$$

By (3.11), (3.16), (3.17) and (2.1), we get

$$\begin{aligned}\alpha_k g(x_k)^{\mathrm{T}} \nabla g(x_k) d_k &\le -\alpha_k(1-\epsilon)\|g(x_k)\|^2 \\ &= -\alpha_k(1-\epsilon)\frac{g_k^{\mathrm{T}} d_k}{g_k^{\mathrm{T}} d_k}\|B_k d_k\|^2 \\ &\le \alpha_k(1-\epsilon)\frac{\beta_2^2}{\beta_3} g_k^{\mathrm{T}} d_k. \end{aligned} \tag{3.19}$$

By $\alpha_k \le 1$, we have

$$\alpha_k g(x_k)^{\mathrm{T}} \nabla g(x_k) d_k \le \alpha_k(1-\epsilon)\frac{\beta_2^2}{\beta_3} g_k^{\mathrm{T}} d_k \le \alpha_k^2(1-\epsilon)\frac{\beta_2^2}{\beta_3} g_k^{\mathrm{T}} d_k. \tag{3.20}$$

Let $\delta \in (0, \min\{1, \sigma(1-\epsilon)\frac{\beta_2^2}{\beta_3}\})$, then we get the line search (1.6). Thus we conclude the result of this lemma. The proof is complete. □

Lemma 3.5 shows that the line search technique (1.6) is well-defined. Now we establish the global convergence theorem for Algorithm 1.

Theorem 3.1. *Let Assumptions A and B hold, and $\{\alpha_k, d_k, x_{k+1}, g_{k+1}\}$ be generated by Algorithm 1. Then*

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{3.21}$$

Proof. By the acceptance rule (1.6) and (3.17), we have

$$\|g(x_{k+1})\|^2 - \|g(x_k)\|^2 \le \delta\alpha_k^2 g_k^{\mathrm{T}} d_k \le -\beta_2\delta\|\alpha_k d_k\|^2. \tag{3.22}$$

By Lemma 3.2, $\{\|g_k\|\}$ is convergent. We obtain from (3.22) that

$$\lim_{k\to\infty} \|\alpha_k d_k\|^2 = 0. \tag{3.23}$$

This means that either

$$\lim_{k\to\infty} \|d_k\| = 0 \tag{3.24}$$

or

$$\lim_{k\to\infty} \alpha_k = 0. \tag{3.25}$$

If Eq. (3.24) holds, we have that from (2.1) and (3.16),

$$\|g_k\| = \|B_k d_k\| \le \beta_1\|d_k\| \to 0. \tag{3.26}$$

Then we get (3.21). If (3.25) holds, then acceptance rule (1.6) means that, for large enough k,

$$\left\|g\left(x_k + \frac{\alpha_k}{r} d_k\right)\right\|^2 - \|g(x_k)\|^2 > \delta\frac{\alpha_k^2}{r^2} g_k^{\mathrm{T}} d_k. \tag{3.27}$$

Since

$$\left\|g\left(x_k + \frac{\alpha_k}{r} d_k\right)\right\|^2 - \|g(x_k)\|^2 = 2\frac{\alpha_k}{r} g_k^{\mathrm{T}} \nabla g(x_k) d_k + o\left(\frac{\alpha_k}{r}\|d_k\|\right). \tag{3.28}$$

Using this together with (3.27) and (3.19), we have

$$\left(2\frac{\delta}{\sigma} - \delta\frac{\alpha_k}{r}\right)\frac{\alpha_k}{r} g_k^{\mathrm{T}} d_k + o\left(\frac{\alpha_k}{r}\|d_k\|\right) \ge 0. \tag{3.29}$$

Dividing (3.29) by $\frac{\alpha_k}{r}\|d_k\|$ and noting that $2\frac{\delta}{\sigma}-\delta\frac{\alpha_k}{r}>0$ and $g_k^{\mathrm{T}}d_k\le 0$, we get

$$\lim_{k\to\infty}\frac{g_k^{\mathrm{T}}d_k}{\|d_k\|}=0. \tag{3.30}$$

Using (3.17), (3.30) implies (3.24) and therefore the conclusion of the theorem is true. □

By Lemma 3.2, $\{\|g_k\|\}$ converges. So, if (3.21) holds, then every accumulation point of $\{x_k\}$ is a solution of (1.1). Since $\nabla g(x)$ is positive definite on Ω_1, (1.1) has only one solution. Moreover, since Ω is bounded, $\{x_k\}\in\Omega$ has at least one accumulation point. Therefore $\{x_k\}$ itself converges to the unique solution x^* of (1.1).

4. Superlinear convergence

In order to obtain the superlinear convergence of Algorithm 1, we also need the following assumption.

Assumption C. ∇g is Hölder continuous at x^*, i.e., there are positive constants M_3 and γ such that for every x in a neighborhood of x^*

$$\|\nabla g(x)-\nabla g(x^*)\|\le M_3\|x-x^*\|^{\gamma}. \tag{4.1}$$

In the rest of the paper, we abbreviate $g(x^*)$ and $\nabla g(x^*)$ as g_* and ∇g_*, respectively.

Lemma 4.1. *Let Assumption A hold. If*

$$\lim_{k\to 0}\frac{\|(B_k-\nabla g_*)d_k\|}{\|d_k\|}=0, \tag{4.2}$$

then $\alpha_k\equiv 1$ for all k sufficiently large. Moreover, $\{x_k\}$ converges superlinearly.

Proof. Let

$$\eta_k=\frac{\|(B_k-\nabla g_*)d_k\|}{\|d_k\|}. \tag{4.3}$$

By (1.3) and (3.16), we have

$$\|d_k\|=\|B_k^{-1}g_k\|\le\frac{1}{\beta_2}\|g_k\|. \tag{4.4}$$

Using (1.3) again, we get

$$\begin{aligned}\nabla g_*(x_k+d_k-x^*) &= \nabla g_*(x_k-x^*)+\nabla g_*d_k\\ &= \nabla g_*(x_k-x^*)-g_k+(\nabla g_*-B_k)d_k\\ &= \nabla g_*(x_k-x^*)-(g_k-g_*)+(\nabla g_*-B_k)d_k\\ &= (\nabla g_*-G_k')(x_k-x^*)+(\nabla g_*-B_k)d_k\end{aligned}$$

where $G_k'=\int_0^1\nabla g(x^*+\tau(x_k-x^*))\mathrm{d}\tau$. It follows that

$$\begin{aligned}\|\nabla g_*(x_k+d_k-x^*)\| &\le \|(\nabla g_*-G_k')(x_k-x^*)\|+\|(\nabla g_*-B_k)d_k\|\\ &= \|(\nabla g_*-G_k')(x_k-x^*)\|+\eta_k\|d_k\|\\ &\le \|\nabla g_*-G_k'\|\|x_k-x^*\|+\eta_k\frac{1}{\beta_2}\|g_k\|\\ &\le \|\nabla g_*-G_k'\|\|x_k-x^*\|+\eta_k\frac{1}{\beta_2}M\|x_k-x_*\|\\ &= \eta_k\frac{1}{\beta_2}M\|x_k-x_*\|+o(\|x_k-x_*\|),\end{aligned} \tag{4.5}$$

where the second inequality follows (4.4) and the last inequality follows (3.7). Since $\eta_k \to 0$ and ∇g_* is positive, (4.5) implies

$$\frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} \to 0. \tag{4.6}$$

Moreover, we have

$$\begin{aligned} \|g(x_k + d_k)\| &= \|g(x_k + d_k) - g_*\| \\ &\le M\|x_k + d_k - x^*\| \\ &= \frac{M}{m}\frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} m\|x_k - x^*\| \\ &\le \frac{M}{m}\frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|}\|g_k\|, \end{aligned} \tag{4.7}$$

where the first and the last inequality follow (3.7). Combining (4.6) and (4.7), we obtain that (2.2) is satisfied for all k sufficiently large. This means the unit step-length is always accepted for all k sufficiently large. Moreover, (4.6) implies the superlinear convergence of $\{x_k\}$. □

The lemma shows that the Dennis–Moré condition (4.2) [19,20] ensures the superlinear convergence of Algorithm 1.

Lemma 4.2. *Let Assumptions A and B hold. If $\alpha_k \ne 1$, then we have the following estimate for α_k when k is sufficiently large:*

$$1 \ge \alpha_k \ge \epsilon_0, \quad \epsilon_0 \in (0, 1). \tag{4.8}$$

Proof. Since $\alpha_k \ne 1$, the step-size α_k was determined by Step 3 of Algorithm 1. Then $\alpha'_k = \frac{\alpha_k}{r}$ did not satisfy (1.6), i.e.,

$$\|g(x_k + \alpha'_k d_k)\|^2 - \|g(x_k)\|^2 > \delta\alpha'^2_k g_k^{\mathrm{T}} d_k,$$

This means that

$$-\delta\alpha'^2_k g_k^{\mathrm{T}} d_k > \|g(x_k)\|^2 - \|g(x_k + \alpha'_k d_k)\|^2. \tag{4.9}$$

By (3.28), (3.10), (3.11) and (3.16), we have

$$\begin{aligned} \|g(x_k)\|^2 - \|g(x_k + \alpha'_k d_k)\|^2 &= -2\alpha'_k g_k^{\mathrm{T}} \nabla g(x_k) d_k + o(\alpha'_k\|d_k\|) \\ &\ge 2\alpha'_k(1-\epsilon)\|g_k\|^2 + o(\alpha'_k\|d_k\|) \\ &\ge \alpha'_k\beta_2^2(1-\epsilon)\|d_k\|^2 + o(\alpha'_k\|d_k\|). \end{aligned} \tag{4.10}$$

Combining (4.9), (4.10) and (3.17), we obtain

$$\begin{aligned} \alpha'^2_k(2\beta_2^2(1-\epsilon) + \delta\beta_3)\|d_k\|^2 &= 2\alpha'^2_k\beta_2^2(1-\epsilon)\|d_k\|^2 + \delta\alpha'^2_k\beta_3\|d_k\|^2 \\ &\ge 2\alpha'^2_k\beta_2^2(1-\epsilon)\|d_k\|^2 - \delta\alpha'^2_k g_k^{\mathrm{T}} d_k \\ &> \|g(x_k)\|^2 - \|g(x_k + \alpha'_k d_k)\|^2 \\ &\ge \alpha'_k\beta_2^2(1-\epsilon)\|d_k\|^2 + o(\alpha'_k\|d_k\|), \end{aligned} \tag{4.11}$$

which means that for all k sufficiently large,

$$\alpha'_k \ge \frac{\beta_2^2(1-\epsilon)}{2\beta_2^2(1-\epsilon) + \delta\beta_3}.$$

Let $\epsilon_0 \in (0, \frac{\beta_2^2(1-\epsilon)r}{2\beta_2^2(1-\epsilon)+\delta\beta_3})$. Then we complete the proof of this lemma. □

Lemma 4.3. *Let Assumptions A and B hold. Then, for any fixed $\gamma > 0$, we have*

$$\sum_{k=0}^{\infty} \|x_k - x^*\|^{\gamma} < \infty. \tag{4.12}$$

Moreover, we have

$$\sum_{k=0}^{\infty} \chi_k(\gamma) < \infty, \tag{4.13}$$

where $\chi_k(\gamma) = \max\{\|x_k - x^*\|^{\gamma}, \|x_{k+1} - x^*\|^{\gamma}\}$.

Proof. First, we show that there exists an index i_0 and a constant $\rho_0 \in (0, 1)$ such that

$$\|g(x_{i+1})\|^2 \le \rho_0 \|g_i\|^2, \quad \forall i \ge i_0. \tag{4.14}$$

If the step-length α_i is determined by Step 2 of Algorithm 1, we have

$$\|g_{i+1}\|^2 \le \rho^2 \|g_i\|^2. \tag{4.15}$$

On the other hand, if α_i is determined by Step 3 of Algorithm 1, then (1.6) is satisfied with $k = i$. Using Lemma 4.2, (1.6), (3.16) and (3.17), we obtain

$$\begin{aligned} \|g_{i+1}\|^2 &\le \|g_i\|^2 + \delta \alpha_i^2 g_i^{\mathrm{T}} d_i \\ &\le \|g_i\|^2 - \delta \epsilon_0^2 \beta_2 \|d_i\|^2 \\ &= \|g_i\|^2 - \delta \epsilon_0^2 \beta_2 \frac{\beta_1^2}{\beta_1^2} \|d_i\|^2 \\ &\le \|g_i\|^2 - \delta \epsilon_0^2 \beta_2 \frac{1}{\beta_1^2} \|g_i\|^2. \end{aligned} \tag{4.16}$$

Then there exists a constant $\rho' \in (0, 1)$ such that $1 - \delta \epsilon_0^2 \beta_2 \frac{1}{\beta_1^2} \le \rho'$ holds for all $i \ge i_0$. Let $\rho_0 = \min\{\rho^2, \rho'\}$. Therefore, (4.14) follows (4.15) and (4.16).

Let J denote the set of indices i for which (4.14) holds. Also, let h_k denote the number of indices in J not exceeding k. Then we have $h_k \ge k - i_0$ for each k. Multiplying (4.14) for $i \in J$ and (4.16) for $i \notin J$ from $i = i_0$ to $i = k$ yields

$$\begin{aligned} \|g_{k+1}\|^2 &\le \prod_{i=i_0, i \notin J}^{k} \rho_0^{h_k} \|g(x_{i_0})\|^2 \\ &\le \prod_{i=0}^{k} \rho_0^{k-i_0} \|g(x_{i_0})\|^2 \\ &\le \rho_0^{k-(i_0+1)} \|g(x_{i_0})\|^2 \\ &= c_1 \rho_0^k, \end{aligned}$$

where $c_1 = \rho_0^{-(i_0+1)} \|g(x_{i_0})\|^2$. This, together with (3.7), shows that $\|x_{k+1} - x^*\|^2 \le m^{-2} c_1 \rho_0^k$ holds for all k large enough. Hence we have (4.12) for any γ.

Notice that $\chi_k(\gamma) \le \|x_k - x^*\|^{\gamma} + \|x_{k+1} - x^*\|^{\gamma}$, and from (4.12), we can get (4.13). □

Lemma 4.4. *Let Assumptions A–C hold. Then, for all k sufficiently large, there exists a positive constant M_4 such that*

$$\|y_k - \nabla g(x^*) s_k\| \le M_4 \chi_k \|s_k\|, \tag{4.17}$$

where $\chi_k = \max\{\|x_k - x^*\|^{\gamma}, \|x_{k+1} - x^*\|^{\gamma}\}$.

Proof. Since $x_k \to x^*$, (4.1) holds for all k large enough. For all k sufficiently large, using the mean value theorem we have

$$\begin{aligned}\|y_k - \nabla g(x^*)s_k\| &= \|\nabla g(x_k + t_0(x_{k+1} - x_k))s_k - \nabla g(x^*)s_k\| \\ &\le \|\nabla g(x_k + t_0(x_{k+1} - x_k)) - \nabla g(x^*)\|\|s_k\| \\ &\le M_3\|x_k + t_0(x_{k+1} - x_k) - x^*\|^{\gamma}\|s_k\| \\ &\le M_4\chi_k\|s_k\|, \end{aligned} \tag{4.18}$$

where $M_4 = M_3(2t_0 + 1)$, $t_0 \in (0, 1)$. Therefore, the inequality of (4.17) holds. □

Denote $Q = \nabla g_*{}^{-1/2}$. For an $n \times n$ matrix K, define a matrix norm $\|K\|_{Q,F} = \|Q^{\mathrm{T}} K Q\|_F$, where $\|.\|_F$ denotes the Frobenius norm of the matrix. We let H_k and H_{k+1} stand for the inverse matrices of B_k and B_{k+1}, respectively.

Lemma 4.5. *Let Assumptions A–C hold. Then, there are positive constants e_i, $i = 1, 2, 3, 4$, and $\eta \in (0, 1)$ such that for all large k,*

$$\|B_{k+1} - \nabla g(x^*)\|_{Q,F} \le (1 + e_1\chi_k)\|B_k - \nabla g(x^*)\|_{Q,F} + e_2\chi_k \tag{4.19}$$

and

$$\|H_{k+1} - \nabla g(x^*)^{-1}\|_{Q^{-1},F} \le (\sqrt{1 - \eta\varpi_k^2} + e_3\chi_k)\|H_k - \nabla g(x^*)^{-1}\|_{Q^{-1},F} + e_4\chi_k, \tag{4.20}$$

where ϖ_k is defined as follows:

$$\varpi_k = \frac{\|Q^{-1}(H_k - \nabla g(x^*)^{-1})y_k\|}{\|H_k - \nabla g(x^*)^{-1}\|_{Q^{-1},F}\|Qy_k\|}. \tag{4.21}$$

In particular, $\{\|B_k\|\}_F$ and $\{\|H_k\|\}_F$ are bounded.

Proof. From the BFGS update formula (1.4), we have

$$\begin{aligned}\|B_{k+1} - \nabla g(x^*)\|_{Q,F} &= \left\| B_k - \nabla g(x^*) + \frac{B_k s_k s_k^{\mathrm{T}} B_k}{s_k^{\mathrm{T}} B_k s_k} + \frac{y_k y_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k} \right\|_{Q,F} \\ &\le (1 + e_1\tau_k)\|B_k - \nabla g(x^*)\|_{Q,F} + e_2\chi_k,\end{aligned}$$

where the last inequality follows the inequality (49) of [6]. Hence, (4.19) holds.

By (4.17), in a way similar to that of [19], we can prove that (4.20) holds and that $\|B_k\|$ and $\|H_k\|$ are bounded. The proof is complete. □

Theorem 4.1. *Let $\{x_k\}$ be generated by Algorithm 1 and let the conditions in Assumptions A–C hold. Then*

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla g(x^*))s_k\|}{\|s_k\|} = 0. \tag{4.22}$$

Moreover, $\{x_k\}$ converges superlinearly and $\alpha_k \equiv 1$ for all k sufficiently large.

Proof. In a similar way to [19], it's not difficult to get

$$\lim_{k\to\infty} \frac{\|Q^{-1}(H_k - \nabla g(x^*)^{-1})y_k\|}{\|Qy_k\|} = 0. \tag{4.23}$$

On the other hand, we obtain

$$\begin{aligned}\|Q^{-1}(H_k - \nabla g(x^*)^{-1})y_k\| &= \|Q^{-1}H_k(\nabla g(x^*) - B_k)\nabla g(x^*)^{-1}y_k\| \\ &\ge \|Q^{-1}H_k(\nabla g(x^*) - B_k)s_k\| - \|Q^{-1}H_k(\nabla g(x^*) - B_k)(s_k - \nabla g(x^*)^{-1}y_k)\| \\ &\ge \|Q^{-1}H_k(\nabla g(x^*) - B_k)s_k\| \\ &\quad - \|Q^{-1}\|\|H_k\|(\|\nabla g(x^*)\| + \|B_k\|)\|\nabla g(x^*)^{-1}(y_k - \nabla g(x^*)s_k)\| \\ &\ge \|Q^{-1}H_k(\nabla g(x^*) - B_k)s_k\|\end{aligned}$$

$$- M_2\chi_k\|Q^{-1}\|\|H_k\|(\|\nabla g(x^*)\| + \|B_k\|)\|\nabla g(x^*)^{-1}\|\|s_k\|$$
$$= \|Q^{-1}H_k(\nabla g(x^*) - B_k)s_k\| - o(\|s_k\|),$$

where the last inequality follows from (4.17). We know $\{\|B_k\|\}$ and $\{\|H_k\|\}$ are bounded and $\{H_k\}$ is positive definite. By (3.6), we get

$$\|Qy_k\| \le M\|Q\|\|s_k\|. \tag{4.24}$$

Combining this with (4.23) and (4.24), we conclude that (4.22) holds. In view of Lemma 4.1, the proof of this theorem is complete. □

5. Numerical results

In this section, we report results of some preliminary numerical experiments with the two algorithms.

Problem 1. The discretized two-point boundary value problem such as the problem in [21]

$$g(x) \triangleq Ax + \frac{1}{(n+1)^2}F(x) = 0,$$

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

$$A = \begin{bmatrix} 8 & -1 & & & & \\ -1 & 8 & -1 & & & \\ & -1 & 8 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & -1 \\ & & & & -1 & 8 \end{bmatrix},$$

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

Problem 2. Unconstrained optimization problem

$$\min f(x), \quad x \in \Re^n,$$

with Engval function [22] $f : \Re^n \to \Re$ defined by

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

The related symmetric nonlinear equation is

$$g(x) \triangleq \frac{1}{4}\nabla f(x) = 0,$$

where $g(x) = (g_1(x), g_2(x), \ldots, g_n(x))^{\mathrm{T}}$ with

$$g_1(x) = x_1(x_1^2 + x_2^2) - 1,$$
$$g_i(x) = x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2) - 1, \quad i = 2, 3, \ldots, n-1,$$
$$g_n(x) = x_n(x_{n-1}^2 + x_n^2).$$

In the experiments, the parameters in Algorithms 1 and 2 were chosen as $r = 0.1$, $\rho = 0.5$, $\delta = 0.9$, $\sigma = 0.95$, B_0 is the unit matrix. For the Problem 2, we take one technique in finding the stepsize α_k, which is that the stepsize α_k will be accepted if the searching time is larger than fifteen in the inner circle. The program was coded in *MATLAB* 7.0.1. We stopped the iteration when the condition $\|F(x)\| \le 10^{-6}$ was satisfied. The columns of Tables 1–8 have the following meaning:

x_0: the starting point.
Dim: the dimension of the problem.

Table 1
Test results for small-scale Problem 1 (Test results for Algorithm 1)

$x0$	(10, ...,10)	(30, ...,30)	(−10, ...,−10)	(−30, ...,−30)	(−300, ...,−300)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 9$	14/18/7.879692e−008	14/18/1.993201e−007	14/18/7.889470e−008	14/18/1.994682e−007	16/20/5.176458e−007
$n = 45$	47/83/6.173797e−008	47/83/1.852692e−007	47/83/6.173597e−008	47/83/1.852415e−007	48/83/2.650969e−007
$n = 95$	87/168/3.614283e−007	88/170/6.212520e−007	87/168/3.614297e−007	88/170/6.212528e−007	89/170/9.001837e−007
$x0$	(10, 0, 10, 0, ...)	(30, 0, 30, 0, ...)	(−10, 0, −10, 0, ...)	(−30, 0, −30, 0, ...)	(−300, 0, −300, 0, ...)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 9$	14/17/3.905388e−008	14/17/1.304219e−007	14/17/3.972696e−008	14/17/1.325971e−007	16/21/9.219813e−007
$n = 45$	45/80/3.991382e−007	46/80/3.528174e−008	45/80/3.992002e−007	46/80/3.526862e−008	46/80/3.505431e−007
$n = 95$	82/155/8.651499e−007	84/157/5.596651e−007	82/155/8.651552e−007	84/157/5.596663e−007	86/159/8.210399e−007
$x0$	(10, −10, 10, −10, ...)	(30, −30, 30, −30, ...)	(10, −10, 10, −10, ...)	(30, −30, 30, −30, ...)	(300, −300, 300, −300, ...)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 9$	13/16/2.467412e−007	13/16/7.416507e−007	13/16/2.540474e−007	13/16/7.490320e−007	14/16/7.042643e−007
$n = 45$	44/77/2.360594e−007	44/77/7.062676e−007	44/77/2.343698e−007	44/77/7.045795e−007	45/77/2.902385e−007
$n = 95$	80/155/5.860856e−007	82/157/4.104328e−007	80/155/5.867282e−007	82/157/4.106177e−007	84/159/6.296641e−007

Table 2
Test results for small-scale Problem 1 (Test results for Algorithm 2)

$x0$	(10, ..., 10)	(30, ..., 30)	(−10, ..., −10)	(−30, ..., −30)	(−300, ..., −300)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 9$	14/19/7.280334e−008	14/19/2.127969e−007	14/19/7.292168e−008	14/19/2.129578e−007	17/23/6.576030e−007
$n = 45$	47/85/8.178643e−008	47/85/2.453743e−007	47/85/8.178272e−008	47/85/2.453736e−007	48/85/1.226699e−007
$n = 95$	90/175/1.656006e−007	90/175/4.967783e−007	90/175/1.656012e−007	90/175/4.967789e−007	92/177/1.319851e−007
$x0$	(10, 0, 10, 0, ...)	(30, 0, 30, 0, ...)	(−10, 0, −10, 0, ...)	(−30, 0, −30, 0, ...)	(−300, 0, −300, 0, ...)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 9$	14/17/3.905388e−008	14/17/1.304219e−007	14/17/3.972696e−008	14/17/1.325971e−007	16/21/9.219813e−007
$n = 45$	46/85/9.015149e−007	47/85/2.707176e−007	46/85/9.015441e−007	47/85/2.878560e−007	48/87/9.358423e−007
$n = 95$	86/169/9.643953e−007	87/169/6.204024e−007	86/169/9.643939e−007	87/169/6.204024e−007	89/171/5.471261e−007
$x0$	(10, −10, 10, −10, ...)	(30, −30, 30, −30, ...)	(10, −10, 10, −10, ...)	(30, −30, 30, −30, ...)	(300, −300, 300, −300, ...)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 9$	13/17/2.215423e−007	13/17/6.590841e−007	13/17/2.169937e−007	13/17/6.545964e−007	14/17/6.334122e−007
$n = 45$	44/81/2.328138e−007	44/81/6.982990e−007	44/81/2.327392e−007	44/81/6.982242e−007	45/81/2.972791e−007
$n = 95$	80/159/5.785503e−007	82/161/4.056331e−007	80/159/5.788913e−007	82/161/4.057626e−007	84/163/6.228919e−007

NI: the total number of iterations.

NG: the number of the function evaluations.

GF: the function norm evaluations.

The numerical results indicate that the proposed method performs better than Algorithm 2 for Problems 1 and 2 from the tables. Moreover, the starting points and the inverse initial points don't influence the performance of the two Algorithms for Problem 1. The number of the iterations and the function iterations on Algorithm 1 are less than those on Algorithm 2. However, we find that the numerical results of the two algorithms are not so good if the starting points are large for Problem 2 in the experiment.

Table 3
Test results for large-scale Problem 1 (Test results for Algorithm 1)

$x0$	(10, ..., 10)	(30, ..., 30)	(−10, ..., −10)	(−30, ..., −30)	(−300, ..., −300)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 300$	97/193/8.917331e−007	102/203/9.269283e−007	97/193/8.917333e−007	102/203/9.269284e−007	115/229/7.744250e−007
$n = 700$	96/189/8.112812e−007	101/199/9.434648e−007	96/189/8.112812e−007	101/199/9.434648e−007	113/223/9.157224e−007
$x0$	(10, 0, 10, 0, ...)	(30, 0, 30, 0, ...)	(−10, 0, −10, 0, ...)	(−30, 0, −30, 0, ...)	(−300, 0, −300, 0, ...)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 300$	88/175/8.188536e−007	94/186/9.230209e−007	88/175/8.188541e−007	94/186/9.230211e−007	106/210/9.029710e−007
$n = 700$	88/174/8.425363e−007	94/186/8.855595e−007	88/174/8.425364e−007	94/186/8.855596e−007	106/210/8.638554e−007
$x0$	(10, −10, 10, −10, ...)	(30, −30, 30, −30, ...)	(10, −10, 10, −10, ...)	(30, −30, 30, −30, ...)	(300, −300, 300, −300, ...)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 300$	79/158/9.585768e−007	85/170/8.990782e−007	79/158/9.585768e−007	85/170/8.990782e−007	97/193/9.395395e−007
$n = 700$	79/158/9.327258e−007	85/170/8.786763e−007	79/158/9.327258e−007	85/170/8.786763e−007	97/193/9.210128e−007

Table 4
Test results for large-scale Problem 1 (Test results for Algorithm 2)

$x0$	(10, ..., 10)	(30, ..., 30)	(−10, ..., −10)	(−30, ..., −30)	(−300, ..., −300)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 300$	97/193/8.917331e−007	102/203/9.269283e−007	97/193/8.917333e−007	102/203/9.269284e−007	115/229/7.744250e−007
$n = 700$	95/189/9.681355e−007	101/201/8.682123e−007	95/189/9.681356e−007	101/201/8.682123e−007	113/225/8.413398e−007
$x0$	(10, 0, 10, 0, ...)	(30, 0, 30, 0, ...)	(−10, 0, −10, 0, ...)	(−30, 0, −30, 0, ...)	(−300, 0, −300, 0, ...)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 300$	88/175/8.188536e−007	94/187/8.610052e−007	88/175/8.188541e−007	94/187/8.610054e−007	106/211/8.381249e−007
$n = 700$	87/173/9.805530e−007	93/185/9.548218e−007	87/173/9.805532e−007	93/185/9.548219e−007	105/209/9.557750e−007
$x0$	(10, −10, 10, −10, ...)	(30, −30, 30, −30, ...)	(10, −10, 10, −10, ...)	(30, −30, 30, −30, ...)	(300, −300, 300, −300, ...)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 300$	80/161/7.762464e−007	85/171/9.460917e−007	80/161/7.762464e−007	85/171/9.460917e−007	97/195/9.201093e−007
$n = 700$	79/159/9.726889e−007	85/171/9.110739e−007	79/159/9.726889e−007	85/171/9.110739e−007	97/195/8.882303e−007

Table 5
Test results for small-scale Problem 2 (Test results for Algorithm 1)

$x0$	(0.01, ..., 0.01)	(0.1, ..., 0.1)	(0.5, ..., 0.5)	(−0.01, ..., −0.01)	(−0.1, ..., −0.1)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 9$	21/148/4.506782e−007	21/148/9.113969e−007	18/117/7.546387e−008	21/148/3.984029e−007	20/119/3.168731e−007
$n = 45$	45/340/4.376572e−007	43/338/5.855291e−007	35/274/9.742033e−007	45/340/4.826550e−007	43/338/8.673252e−007
$n = 95$	43/324/6.907839e−007	43/324/5.262386e−007	37/290/7.376250e−007	43/324/8.591994e−007	45/340/4.261069e−007
$x0$	(0.01, 0, 0.01, 0, ...)	(0.1, 0, 0.1, 0, ...)	(0.5, 0, 0.5, 0, ...)	(−0.01, 0, −0.01, 0, ...)	(−0.1, 0, −0.1, 0, ...)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 9$	21/120/4.980854e−007	22/121/1.563643e−007	20/133/9.367519e−008	21/134/2.988128e−007	21/148/7.952513e−007
$n = 45$	45/340/4.319686e−007	41/322/5.175941e−007	39/306/7.642885e−007	45/340/5.040646e−007	45/354/7.728349e−007
$n = 95$	43/324/6.756781e−007	43/324/5.053860e−007	43/324/4.823266e−007	43/324/8.202966e−007	45/340/6.921257e−007

Table 6
Test results for small-scale Problem 2 (Test results for Algorithm 2)

$x0$	(0.01, ..., 0.01)	(0.1, ..., 0.1)	(0.5, ..., 0.5)	(−0.01, ..., −0.01)	(−0.1, ..., −0.1)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 9$	21/316/4.506782e−007	21/316/9.113969e−007	18/271/7.546387e−008	21/316/3.984029e−007	20/301/3.168731e−007
$n = 45$	45/676/4.376572e−007	43/646/5.855291e−007	35/526/9.742033e−007	45/676/4.826550e−007	43/646/8.673252e−007
$n = 95$	43/646/6.907839e−007	43/646/5.262386e−007	37/556/7.376250e−007	43/646/8.591994e−007	45/676/4.261069e−007
$x0$	(0.01, 0, 0.01, 0, ...)	(0.1, 0, 0.1, 0, ...)	(0.5, 0, 0.5, 0, ...)	(−0.01, 0, −0.01, 0, ...)	(−0.1, 0, −0.1, 0, ...)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 9$	21/316/4.980854e−007	22/331/1.563643e−007	20/301/9.367519e−008	21/316/2.988128e−007	21/316/7.952513e−007
$n = 45$	45/676/4.319686e−007	41/616/5.175941e−007	39/586/7.642885e−007	45/676/5.040646e−007	45/676/7.728349e−007
$n = 95$	43/646/6.756781e−007	43/646/5.053860e−007	43/646/4.823266e−007	43/646/8.202966e−007	45/676/6.921257e−007

Table 7
Test results for large-scale Problem 2 (Test results for Algorithm 1)

$x0$	(0.01, ..., 0.01)	(0.1, ..., 0.1)	(0.5, ..., 0.5)	(−0.01, ..., −0.01)	(−0.1, ..., −0.1)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 300$	45/340/4.258015e−007	43/324/5.101157e−007	41/322/4.955111e−007	45/340/4.751995e−007	45/340/5.346625e−007
$n = 700$	45/340/9.111422e−007	46/341/6.231190e−007	43/338/5.600939e−007	45/340/9.062300e−007	47/370/4.749276e−007
$x0$	(0.01, 0, 0.01, 0, ...)	(0.1, 0, 0.1, 0, ...)	(0.5, 0, 0.5, 0, ...)	(−0.01, 0, −0.01, 0, ...)	(−0.1, 0, −0.1, 0, ...)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 300$	45/340/4.331731e−007	45/340/5.138086e−007	43/338/8.468113e−007	45/340/4.637890e−007	45/340/5.301334e−007
$n = 700$	45/340/8.771359e−007	43/324/8.721209e−007	43/324/6.196297e−007	45/340/9.798305e−007	46/355/4.514799e−007

Table 8
Test results for large-scale Problem 2 (Test results for Algorithm 2)

$x0$	(0.01, ..., 0.01)	(0.1, ..., 0.1)	(0.5, ..., 0.5)	(−0.01, ..., −0.01)	(−0.1, ..., −0.1)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 300$	45/676/4.258015e−007	43/646/5.101157e−007	41/616/4.955111e−007	45/676/4.751995e−007	45/676/5.346625e−007
$n = 700$	45/676/9.111422e−007	46/691/6.231190e−007	43/646/5.600939e−007	45/676/9.062300e−007	47/706/4.749276e−007
$x0$	(0.01, 0, 0.01, 0, ...)	(0.1, 0, 0.1, 0, ...)	(0.5, 0, 0.5, 0, ...)	(−0.01, 0, −0.01, 0, ...)	(−0.1, 0, −0.1, 0, ...)
Dim	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF	NI/NG/GF
$n = 300$	45/676/4.331731e−007	45/676/5.138086e−007	43/646/8.468113e−007	45/676/4.637890e−007	45/676/5.301334e−007
$n = 700$	45/676/8.771359e−007	43/646/8.721209e−007	43/646/6.196297e−007	45/676/9.798305e−007	46/691/4.514799e−007

6. Conclusion

A new inexact backtracking line search technique is proposed for solving symmetric nonlinear equations in this paper, which can ensure that the search direction is descending for the norm function. The method possesses global and superlinear convergence, and the numerical results show that the method is successful for the test problems. We hope the method can be a further topic for the symmetric nonlinear equations.

Acknowledgement

The authors would like to thank the referees for their helpful suggestions and comments.

References

[1] R. Byrd, J. Nocedal, A tool for the analysis of quasi-Newton methods with application to unconstrained minimization, SIAM Journal on Numerical Analysis 26 (1989) 727–739.

[2] R. Byrd, J. Nocedal, Y. Yuan, Global convergence of a class of quasi-Newton methods on convex problems, SIAM Journal on Numerical Analysis 24 (1987) 1171–1189.

[3] Y. Dai, Convergence properties of the BFGS algorithm, SIAM Journal on Optimization 13 (2003) 693–701.

[4] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Pretice-Hall, Inc., Englewood Cliffs, NJ, 1983.

[5] R. Fletcher, Practical Methods of Optimization, 2nd ed., John Wiley & Sons, Chichester, 1987.

[6] A. Griewank, Ph.L. Toint, Local convergence analysis for partitioned quasi-Newton updates, Numerische Mathematik 39 (1982) 429–448.

[7] Y. Yuan, W. Sun, Theory and Methods of Optimization, Science Press of China, 1999.

[8] D. Li, M. Fukushima, A modified BFGS method and its global convergence in nonconvex minimization, Journal of Computational and Applied Mathematics 129 (2001) 15–35.

[9] M.J.D. Powell, A new algorithm for unconstrained optimation, in: J.B. Rosen, O.L. Mangasarian, K. Ritter (Eds.), Nonlinear Programming, Academic Press, New York, 1970.

[10] Z. Wei, L. Qi, X. Chen, An SQP-type method and its application in stochastic programming, Journal of Optimization Theory and Applications 116 (2003) 205–228.

[11] Z. Wei, G. Yu, G. Yuan, Z. Lian, The superlinear convergence of a modified BFGS-type method for unconstrained optimization, Computational Optimization and Applications 29 (2004) 315–332.

[12] Z. Wei, G. Li, L. Qi, New quasi-Newton methods for unconstrained optimization problems, Applied Mathematics and Computation 175 (2006) 1156–1188.

[13] A. Griewank, The 'global' convergence of Broyden-like methods with a suitable line search, Journal of the Australian Mathematical Society Series B 28 (1986) 75–92.

[14] D. Li, M. Fukushima, A global and superlinear convergent Gauss–Newton-based BFGS method for symmetric nonlinear equations, SIAM Journal on Numerical Analysis 37 (1999) 152–172.

[15] G. Yuan, X. Li, An approximate Gauss–Newton-based BFGS method with descent directions for solving symmetric nonlinear equations, OR Transactions 8 (4) (2004) 10–26.

[16] G. Yuan, X. Lu, A nonmonotone Gauss–Newton-based BFGS method for solving symmetric nonlinear equations, Journal of Lanzhou University (Natural Sciences) 41 (Suppl.) (2005) 851–855.

[17] D. Zhu, Nonmonotone backtracking inexact quasi-Newton algorithms for solving smmoth nonlinear equations, Applied Mathematics and Computation 161 (2005) 875–895.

[18] P.N. Brown, Y. Saad, Convergence theorey of nonlinear Newton–Kryloy algorithms, SIAM Journal on Optimization 4 (1994) 297–330.

[19] J.E. Dennis, J.J. Moré, A characteization of superlinear convergence and its application to quasi-Newton methods, Mathematics of Computation 28 (1974) 549–560.

[20] J.E. Dennis Jr., J.J. Moré, Quasi-Newton methods, motivation and theory, SIAM Review 19 (1977) 46–89.

[21] J.J. Moré, B.S. Garbow, K.E. Hillstrome, Testing unconstrained optimization software, Association for Computing Machinery. Transactions on Mathematical Software 7 (1981) 17–41.

[22] E. Yamakawa, M. Fukushima, Testing parallel bariable transformation, Computational Optimization and Applications 13 (1999) 253–274.