Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 13, No. 4, pp. 645–663, October 2017

ADJOINT BROYDEN METHODS FOR SYMMETRIC NONLINEAR EQUATIONS*

Hui-Ping Cao and Dong-Hui Li†

Abstract: In this paper, we adopt a technique similar to the derivation of Powell's Symmetric Broyden (PSB) method to obtain a new class of quasi-Newton methods which we call symmetric adjoint Broyden methods. The symmetric adjoint method shares some nice properties as its non-symmetric version. By the use of a nonmonotone line search, we show that the symmetric adjoint Broyden method with adjoint Broyden tangent update is globally and superlinearly convergent when applied to solve symmetric nonlinear equations. We also do some preliminary numerical experiments to test the performance of the proposed method. The numerical results indicate that the proposed method is effective and competitive.

Key words: *symmetric adjoint Broyden update, nonlinear equations, global convergence, superlinear convergence, automatic differentiation*

Mathematics Subject Classification: *65H10, 65K05, 90C26*

1 Introduction

The general nonlinear equations can be stated as

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

where function $F(x) = (f_1(x), \ldots, f_n(x))^T$ with elements $f_i(x) : R^n \to R$, $i = 1, \ldots, n$ continuously differentiable. We will pay attention to the symmetric nonlinear equations in the sense that $F'(x) = F'(x)^T$, $\forall x \in R^n$, where $F'(x)$ denotes the Jacobian of F at x. So, without specification, throughout the paper, we always suppose that the Jacobian matrix $F'(x)$ is symmetric for any $x \in R^n$.

There has been some progress in the study of the numerical methods for solving symmetric nonlinear equations. Li and Fukushima [5] proposed a globally and superlinearly convergent Gauss-Newton-based BFGS method for solving symmetric nonlinear equations. Gu et al. [3] developed a norm descent BFGS method on the basis of the Gauss-Newton-based BFGS method. The authors in [6, 16] also studied quasi-Newton methods for solving symmetric nonlinear equations. Conjugate gradient type methods have also been applied to solve symmetric nonlinear equations. For examples, Li and Wang [8] extended the modified Fletcher-Reeves (FR) nonlinear conjugate gradient method proposed by Zhang et al. [17] to

*This research was supported by the National Natural Science Foundation of China (Grant number: 11371154, 11701577.

†Corresponding author.

solve symmetric equations. Recently, Zhou and Shen [18] proposed a derivative-free Polak-Ribiére-Polyak (PRP) method for solving symmetric nonlinear equations without computing exact gradient and Jacobian. The method is a generalization of the classical PRP method for unconstrained optimization problems [18].

In this paper, we will develop a new class of quasi-Newton methods for solving symmetric nonlinear equations. They are based on the recently developed adjoint Broyden methods for solving nonlinear equations [12–14]. Like general quasi-Newton method, the adjoint Broyden method generates a sequence of iterates $\{x_k\}$ recurrently by

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \ldots,$$

where α_k is the step length determined by some line search and the quasi-Newton direction $d_k \in R^n$ is the solution of the linear equations

$$B_k d + F(x_k) = 0.$$

The quasi-Newton matrix B_k is updated by the so-called adjoint Broyden formula [12]

$$B_{k+1} = B_k + \frac{\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k}(F'(x_{k+1}) - B_k), \tag{1.2}$$

where $\sigma_k \in R^n$. Typical choices of σ_k include [12]

(A) $\sigma_k = (F'(x_{k+1}) - B_k)s_k$ (adjoint Broyden tangent update),

(B) $\sigma_k = (F(x_{k+1}) - F(x_k))/\alpha_k - B_k s_k$ (adjoint Broyden secant update) and

(C) $\sigma_k = F(x_{k+1})$ (adjoint Broyden residual update).

Case (B) is an approximation to (A) of order $o(\|\alpha_k s_k\|)$ and case (C) is identical to (B) in the full step quasi-Newton method, namely $\alpha_k \equiv 1$. Our major concern in the paper is the adjoint Broyden tangent update, namely the case (A).

Unlike existing quasi-Newton methods where the quasi-Newton matrix satisfies the secant condition $B_{k+1}s_k = y_k$ with $s_k = x_{k+1} - x_k$ and $y_k = F(x_{k+1}) - F(x_k)$, the quasi-Newton matrix in the adjoint Broyden method satisfies the adjoint tangent condition

$$\sigma_k^T B_{k+1} = \sigma_k^T F'(x_{k+1}). \tag{1.3}$$

The adjoint Broyden update 1.2 maintains some nice properties of Broyden's update. In particular, it possesses the least change property in the sense that the inequality

$$\|B_{k+1} - B_k\|_F \le \|\frac{\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k}\|_F \cdot \|B - B_k\|_F = \|B - B_k\|_F$$

is satisfied for all symmetric matrices B satisfying $\sigma_k^T B = \sigma_k^T F'(x_{k+1})$. In addition, it enjoys some other nice properties as follows [12].

(i) Adjoint Broyden updates (A), (B) and (C) are invariant with respect to regular linear transformation of the state space of x, while Broyden's update is only invariant with respect to regular linear transformation of the range of $F(x)$.

(ii) The adjoint tangent Broyden update can be regarded as a particular two sided rank one (TR1) update [2] which can maintain the validity of previous tangent conditions. However, the heredity property [12] is not shared by Broyden's update.

(iii) The quasi-Newton direction generated by the adjoint Broyden update (C) is a descent direction of the residual function $f(x) = \frac{1}{2}F(x)^T F(x)$. Specifically, we have for $F(x_{k+1}) \neq 0$

$$\begin{aligned}\nabla f(x_{k+1})^T d_{k+1} &= F(x_{k+1})^T F'(x_{k+1}) d_{k+1} \\ &= F(x_{k+1})^T B_{k+1} d_{k+1} = -F(x_{k+1})^T F(x_{k+1}) < 0,\end{aligned}$$

where the second equality follows from the adjoint Broyden residual update.

Both theoretical analysis and numerical experiments have verified that the adjoint Broyden method is a kind of promising quasi-Newton method, though the computation of the adjoint Broyden method is slightly more expensive than that of Broyden's method [12]. It has been proved to be locally linearly and q-superlinearly convergent [12]. If some line search is used, the method can be globally and superlinearly convergent [14] under the same requirements on $F(x)$ as those for Newton's method. A range of test results in [12] have shown that the adjoint Broyden method usually outperforms Newton's and Broyden's methods in terms of CPU time as well as the number of iterations.

The nice properties and excellent numerical performance of the adjoint Broyden method encourage us to study its symmetric version for solving symmetric nonlinear equations. In this paper, in a way similar to the derivation of the PSB update, we will derive a symmetric adjoint Broyden update that takes the form

$$\begin{aligned}B_{k+1} \quad = \quad & B_k + \frac{\sigma_k \sigma_k^T (F'(x_{k+1}) - B_k) + (F'(x_{k+1}) - B_k)\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k} \\ & - \frac{\sigma_k^T (F'(x_{k+1}) - B_k)\sigma_k}{\sigma_k^T \sigma_k} \cdot \frac{\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k}.\end{aligned}$$

Details will be given in the next section. We will pay particular attention to the adjoint Broyden tangent update. By the use of a nonmonotone line search in [7], we will establish the global and superlinear convergence of the method. We will also present some numerical results to verify the efficiency of the method.

The remainder of the paper is organized as follows. In Section 2, we derive a symmetric adjoint Broyden update and show that it can preserve some nice properties of its non-symmetric version. In Section 3, we will prove that the symmetric adjoint Broyden tangent method with a nonmonotone line search is globally and superlinearly convergent when applied to solve a system of symmetric nonlinear equations. In Section 4, we list some preliminary numerical results to illustrate the efficiency of the method. Finally, we give some remarks.

2 Symmetric Adjoint Broyden Update

In this section, similar to the derivation of the PSB update [10], we will derive a symmetric adjoint Broyden update. For a given symmetric matrix $B \in R^{n\times n}$, we let $\bar{B}$ be determined by the adjoint Broyden update formula

$$\bar{B} = B + \frac{\sigma\sigma^T}{\sigma^T\sigma}(F'(\bar{x}) - B),$$

where $\sigma \in R^n$ with $\sigma \neq 0$.

We are going to generate a sequence of matrices $\{C_k\}$, whose limit $\tilde{B}$ is symmetric and satisfies the adjoint tangent condition

$$\sigma^T F'(\bar{x}) = \sigma^T \tilde{B}.$$

The idea of proof comes from [1, 15].

We let $C_0 = B$ and $C_1 = \bar{B}$ determined by the adjoint Broyden update. Generally, C_1 is not symmetric. So, we can construct a symmetric matrix

$$C_2 = \frac{C_1 + C_1^T}{2}.$$

We repeat the above procedure and generate a sequence of matrices $\{C_k\}$ by

$$\begin{cases} C_{2k+1} = C_{2k} + \dfrac{\sigma\sigma^T}{\sigma^T\sigma}(F'(\bar{x}) - C_{2k}), \\ C_{2k+2} = \dfrac{C_{2k+1} + C_{2k+1}^T}{2}, \end{cases} \quad k = 0, 1, \ldots. \tag{2.1}$$

By the construction of C_k, it is not difficult to see that for each k, matrix C_{2k+1} is the closest matrix in the sense of $\|\cdot\|_F$ to C_{2k} in the set where all the matrices satisfying the adjoint tangent condition (1.3)

$$Q(\sigma, F'(\bar{x})) \stackrel{\triangle}{=} \{A \in R^{n\times n} | \sigma^T A = \sigma^T F'(\bar{x})\},$$

and matrix C_{2k+2} is the closest matrix to C_{2k+1} in the set

$$\mathcal{S} \stackrel{\triangle}{=} \{A \in R^{n\times n} | A^T = A\}.$$

The proposition below will show that the limit of sequence $\{C_k\}$ is

$$\tilde{B} = B + \frac{\sigma\sigma^T(F'(\bar{x}) - B) + (F'(\bar{x}) - B)\sigma\sigma^T}{\sigma^T\sigma} - \frac{\sigma^T(F'(\bar{x}) - B)\sigma}{\sigma^T\sigma} \cdot \frac{\sigma\sigma^T}{\sigma^T\sigma}. \tag{2.2}$$

Proposition 2.1. *For any given $B \in \mathcal{S}$ and $\sigma \in R^n$ with $\sigma \neq 0$, the sequence of matrices $\{C_k\}$ generated by (2.1) with $C_0 = B$ converges to $\tilde{B}$ defined by (2.2). Moreover, we have*

$$\tilde{B} \in S \cap Q(\sigma, F'(\bar{x})). \tag{2.3}$$

Proof. First, the relation (2.3) can be verified by the definition of $\tilde{B}$ directly. We are going to show that $\tilde{B}$ is the limit of $\{C_k\}$. By the construction of $\{C_k\}$, we can see that if $\{C_{2k}\}$ converges to $\tilde{B}$, then the direct calculation shows that $\{C_{2k+1}\}$ also converges to $\tilde{B}$. So it suffices to prove that $\lim_{k\to\infty} C_{2k} = \tilde{B}$. Let $G_k = C_{2k}$. We can get from (2.1)

$$\begin{aligned} G_{k+1} &= G_k + \frac{1}{2}\frac{\sigma\sigma^T(F'(\bar{x}) - G_k) + (F'(\bar{x}) - G_k)\sigma\sigma^T}{\sigma^T\sigma} \\ &= G_k + \frac{1}{2}\frac{\sigma\omega_k + \omega_k^T\sigma^T}{\sigma^T\sigma}, \end{aligned} \tag{2.4}$$

where $\omega_k = \sigma^T(F'(\bar{x}) - G_k)$. By the definition of ω_k, one has

$$\omega_{k+1} = \sigma^T(F'(\bar{x}) - G_{k+1}) = \sigma^T\Big(F'(\bar{x}) - G_k - \frac{1}{2}\frac{\sigma\omega_k + \omega_k^T\sigma^T}{\sigma^T\sigma}\Big)$$

$$= \omega_k - \frac{1}{2}\frac{\sigma^T\sigma\omega_k + \sigma^T\omega_k^T\sigma^T}{\sigma^T\sigma} = \frac{1}{2}\omega_k\Big(I - \frac{\sigma\sigma^T}{\sigma^T\sigma}\Big) = \omega_k P,$$

where

$$P = \frac{1}{2}\Big(I - \frac{\sigma\sigma^T}{\sigma^T\sigma}\Big).$$

Notice that the matrix P has one zero eigenvalue and $n-1$ eigenvalues equal to $1/2$. By using the Sherman-Morrison theorem, we can obtain

$$\begin{aligned}\sum_{k=0}^{\infty}\omega_k &= \sigma^T(F'(\bar{x}) - B)\sum_{k=0}^{\infty}P^k \\ &= \sigma^T(F'(\bar{x}) - B)(I-P)^{-1} \\ &= 2\sigma^T(F'(\bar{x}) - B)\Big(I - \frac{1}{2}\frac{\sigma\sigma^T}{\sigma^T\sigma}\Big) \\ &= 2\sigma^T\Big(F'(\bar{x}) - B) - \sigma^T(F'(\bar{x}) - B\Big)\frac{\sigma\sigma^T}{\sigma^T\sigma}.\end{aligned}$$

This together with (2.4) implies

$$\begin{aligned}&\sum_{k=0}^{\infty}(G_{k+1} - G_k) = \frac{1}{2}\sum_{k=0}^{\infty}\frac{\sigma\omega_k + \omega_k^T\sigma^T}{\sigma^T\sigma} \\ &= \frac{1}{\sigma^T\sigma}\Big(\sigma\sigma^T(F'(\bar{x}) - B) - \frac{1}{2}\frac{\sigma\sigma^T(F'(\bar{x}) - B)\sigma\sigma^T}{\sigma^T\sigma} \\ &\quad + (F'(\bar{x}) - B)\sigma\sigma^T - \frac{1}{2}\frac{\sigma\sigma^T(F'(\bar{x}) - B)\sigma\sigma^T}{\sigma^T\sigma}\Big) \\ &= \frac{\sigma\sigma^T(F'(\bar{x}) - B) + (F'(\bar{x}) - B)\sigma\sigma^T}{\sigma^T\sigma} - \frac{\sigma^T(F'(\bar{x}) - B)\sigma}{\sigma^T\sigma}\cdot\frac{\sigma\sigma^T}{\sigma^T\sigma}.\end{aligned}$$

Consequently, we get

$$\begin{aligned}&\lim_{k\to\infty} G_k = B + \sum_{k=0}^{\infty}(G_{k+1} - G_k) \\ &= B + \frac{\sigma\sigma^T(F'(\bar{x}) - B) + (F'(\bar{x}) - B)\sigma\sigma^T}{\sigma^T\sigma} - \frac{\sigma^T(F'(\bar{x}) - B)\sigma}{\sigma^T\sigma}\cdot\frac{\sigma\sigma^T}{\sigma^T\sigma} \\ &= \tilde{B}.\end{aligned}$$

The proof is complete. □

We call the update (2.2) symmetric adjoint Broyden update. The recurrent form of the symmetric adjoint Broyden update formula is given by

$$\begin{aligned}B_{k+1} &= B_k + \frac{\sigma_k\sigma_k^T(F'(x_{k+1}) - B_k) + (F'(x_{k+1}) - B_k)\sigma_k\sigma_k^T}{\sigma_k^T\sigma_k} \\ &\quad - \frac{\sigma_k^T(F'(x_{k+1}) - B_k)\sigma_k}{\sigma_k^T\sigma_k}\cdot\frac{\sigma_k\sigma_k^T}{\sigma_k^T\sigma_k}.\end{aligned} \tag{2.5}$$

In what follows, we will show that the symmetric adjoint Broyden update can maintain some nice properties as the adjoint Broyden update. First, the following lemma gives the least change property of the symmetric adjoint Broyden update.

Lemma 2.2 (Least tangent change). *Given $B_k \in \mathcal{S}$ and $\sigma_k \in R^n$ with $\sigma_k \neq 0$. Let $M_k \in R^{n\times n}$ be a symmetric and nonsingular matrix satisfying $M_k\sigma_k = M_k^{-1}\sigma_k$. Then the matrix B_{k+1} defined by (2.5) is the unique solution of the following minimization problem*

$$\min\{\|\hat{B} - B_k\|_{M_k,F} : \hat{B} \in \mathcal{S} \cap Q(\sigma_k, F'(x_{k+1}))\}, \tag{2.6}$$

where $\|B\|_{M_k,F} = \|M_k B M_k\|_F$ and $Q(\sigma_k, F'(x_{k+1})) = \{A \in R^{n\times n} | \sigma_k^T A = \sigma_k^T F'(x_{k+1})\}$.

Proof. For any $\hat{B} \in \mathcal{S} \cap Q(\sigma_k, F'(x_{k+1}))\}$, we obviously have

$$(F'(x_{k+1}) - B_k)\sigma_k = (\hat{B} - B_k)\sigma_k \quad \text{and} \quad \sigma_k^T(F'(x_{k+1}) - B_k) = \sigma_k^T(\hat{B} - B_k).$$

Denote $z_k = M_k\sigma_k = M_k^{-1}\sigma_k$, $E_k = M_k(\hat{B}-B_k)M_k$ and $\bar{E}_k = M_k(B_{k+1}-B_k)M_k$. Multiply (2.5) from right and left hand sizes by M_k respectively, we obtain

$$\bar{E}_k = \frac{z_k z_k^T E_k + E_k z_k z_k^T}{z_k^T z_k} - \frac{z_k^T E_k z_k}{(z_k^T z_k)^2} z_k z_k^T.$$

It is clear that $\|\bar{E}_k z_k\|_2 = \|E_k z_k\|_2$. And if v is orthogonal to z_k, then $\|\bar{E}_k v\|_2 \leq \|E_k v\|_2$. So one has $\|\bar{E}_k\|_F \leq \|E_k\|_F$. On the other hand, function $\|\hat{B} - B_k\|_{M_k,F}$ is strictly convex on the convex set $\{\hat{B} \in R^{n\times n} | \hat{B} \in \mathcal{S} \cap Q(\sigma_k, F'(x_{k+1}))\}$. Consequently, the matrix B_{k+1} defined by (2.5) is the unique solution of (2.6). □

Lemma 2.3 (Heredity). *In the case that $F(x)$ is affine, the symmetric adjoint tangent update (2.5) maintains the validity of tangent conditions along the adjoint direction $\sigma_j = (F'(x^*) - B_j)s_j$, i.e.*

$$\sigma_j^T(B_k - F'(x^*)) = 0, \quad \forall j < k. \tag{2.7}$$

Proof. We verify (2.7) by induction.

First, it is easy to get that

$$\sigma_0^T B_1 = \sigma_0^T F'(x_1) = \sigma_0^T F'(x^*).$$

This shows (2.7) with $k = 1$.

Consider the case $k = 2$. We also get from the adjoint tangent condition that

$$\sigma_1^T B_2 = \sigma_1^T F'(x_2) = \sigma_1^T F'(x^*).$$

According to $\sigma_0^T B_1 = \sigma_0^T F'(x^*)$, we obtain

$$\sigma_0^T(F'(x^*) - B_1)s_1 = \sigma_0^T\sigma_1 = 0$$

and

$$\begin{aligned}
\sigma_0^T B_2 \;&=\; \sigma_0^T\Big(B_1 + \frac{\sigma_1\sigma_1^T(F'(x_2) - B_1) + (F'(x_2) - B_1)\sigma_1\sigma_1^T}{\sigma_1^T\sigma_1} \\
&\qquad - \frac{\sigma_1^T(F'(x_2) - B_1)\sigma_1}{\sigma_1^T\sigma_1} \cdot \frac{\sigma_1\sigma_1^T}{\sigma_1^T\sigma_1}\Big) \\
&=\; \sigma_0^T B_1 + \frac{\sigma_0^T\sigma_1\sigma_1^T(F'(x_2) - B_1) + \sigma_0^T(F'(x^*) - B_1)\sigma_1\sigma_1^T}{\sigma_1^T\sigma_1} \\
&\qquad - \frac{\sigma_1^T(F'(x_2) - B_1)\sigma_1}{\sigma_1^T\sigma_1} \cdot \frac{\sigma_0^T\sigma_1\sigma_1^T}{\sigma_1^T\sigma_1}
\end{aligned}$$

$$= \quad \sigma_0^T B_1 = \sigma_0^T F'(x^*).$$

This shows that the equality (2.7) is satisfied for $k = 2$.

Suppose that (2.7) holds for some $k > 1$, i.e.,

$$\sigma_j^T B_k = \sigma_j^T F'(x^*), \ \forall j < k.$$

We are going to show that (2.7) holds for $k+1$. It is clear that (2.7) is satisfied with $j = k$. For any $j = 0, 1, \dots, k-1$, we have

$$\begin{aligned}
\sigma_j^T B_{k+1} &= \sigma_j^T \Big(B_k + \frac{\sigma_k \sigma_k^T (F'(x_{k+1}) - B_k) + (F'(x_{k+1}) - B_k)\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k} \\
&\quad - \frac{\sigma_k^T (F'(x_{k+1}) - B_k)\sigma_k}{\sigma_k^T \sigma_k} \cdot \frac{\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k} \Big) \\
&= \sigma_j^T B_k + \frac{\sigma_j^T \sigma_k \sigma_k^T (F'(x_{k+1}) - B_k) + \sigma_j^T (F'(x^*) - B_k)\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k} \\
&\quad - \frac{\sigma_k^T (F'(x_{k+1}) - B_k)\sigma_k}{\sigma_k^T \sigma_k} \cdot \frac{\sigma_j^T \sigma_k \sigma_k^T}{\sigma_k^T \sigma_k} \\
&= \sigma_j^T B_k = \sigma_j^T F'(x^*).
\end{aligned}$$

The proof is complete. □

3 A Symmetric Adjoint Broyden Algorithm and Its Convergence

In this section, we propose an approximately norm descent quasi-Newton method for solving symmetric nonlinear equation (1.1) in which symmetric adjoint Broyden update is used. Under appropriate conditions, we establish its global and superlinear convergence.

First, we notice that the matrix B_{k+1} determined by (2.5) may be singular even if B_k is nonsingular. To overcome this drawback, we can use a nonsingular version of the symmetric adjoint Broyden update similar to Moré and Trangenstein's nonsingular Broyden's update [9]. The nonsingular update formula is given by

$$\begin{aligned}
B_{k+1} &= B_k + \theta_k \frac{\sigma_k \sigma_k^T (F'(x_{k+1}) - B_k) + (F'(x_{k+1}) - B_k)\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k} \\
&\quad - \theta_k^2 \frac{\sigma_k^T (F'(x_{k+1}) - B_k)\sigma_k}{\sigma_k^T \sigma_k} \frac{\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k},
\end{aligned} \tag{3.1}$$

where the parameter θ_k can be chosen to satisfy $| \theta_k - 1 | \le \hat{\theta} < 1$ with some constant $\hat{\theta} \in (0,1)$ so that when B_k is nonsingular, B_{k+1} is nonsingular too. It is easy to see that the symmetric adjoint Broyden update formula (2.5) corresponds to the case $\theta_k = 1$.

To globalize the method, we adopt a nonmonotone line search introduced by Li and Fukushima [7]. The nonmonotone line search can be stated as follows. Given constants $\rho, \lambda \in (0,1)$ and $\mu_1 > 0$. Find the smallest nonnegative integer i such that

$$\|F(x_k + \rho^i d_k)\| \le \|F(x_k)\| - \mu_1 \|\rho^i d_k\|^2 + \eta_k \|F(x_k)\|, \tag{3.2}$$

where $\{\eta_k\}$ is a positive sequence satisfying

$$\sum_{k=0}^{\infty} \eta_k \le \eta < \infty \tag{3.3}$$

with some constant $\eta > 0$. Let the step length $\alpha_k = \rho^{i_k}$, where i_k is the smallest nonnegative integer satisfies (3.2).

We propose a symmetric adjoint Broyden algorithm for solving (1.1) as follows.

Algorithm 3.1. Symmetric Adjoint Broyden Method

Step 0. Given constants $\rho, \lambda, \bar{\theta} \in (0,1)$, $\mu_1, \mu_2 > 0$, initial point $x_0 \in R^n$ and initial symmetric and nonsingular matrix $B_0 \in R^{n\times n}$. Let $k := 0$.

Step 1. Stop if $F(x_k) = 0$. Otherwise, solve the system of linear equations

$$B_k d + F(x_k) = 0 \tag{3.4}$$

to get d_k.

Step 2. If

$$\|F(x_k + d_k)\| \le \lambda\|F(x_k)\| - \mu_2\|d_k\|^2, \tag{3.5}$$

then we let $\alpha_k := 1$ and go to Step 4. Else, go to Step 3.

Step 3. Let α_k be determined by the nonmonotone line search (3.2) and $x_{k+1} := x_k + \alpha_k d_k$.

Step 4. Update B_k by the nonsingular symmetric adjoint Broyden update formula (3.1), where $s_k = x_{k+1} - x_k$ and $\sigma_k = (F'(x_{k+1}) - B_k)s_k$. The parameter θ_k is chosen so that $|\theta_k - 1| \le \bar{\theta}$ and B_{k+1} is nonsingular.

Step 5. Let $k := k + 1$. Go to Step 1.

The remainder of this section is devoted to the global and superlinear convergence of Algorithm 3.1. Without specification, we always assume that σ_k in Algorithm 3.1 is specified by

$$\sigma_k = (F'(x_{k+1}) - B_k)s_k.$$

The following two lemmas are obvious according to [7].

Lemma 3.2. *The sequence $\{x_k\}$ generated by Algorithm 3.1 is contained in the level set*

$$\Omega = \{x \in R^n \mid \|F(x)\| \le e^{\eta}\|F(x_0)\|\},$$

where η is defined by (3.3). Moreover, it holds that

$$\sum_{k=0}^{\infty} \|s_k\|^2 < \infty.$$

Lemma 3.3. *Let $\{x_k\}$ be generated by Algorithm 3.1. Then the sequence of function evaluations $\{\|F(x_k)\|\}$ is convergent.*

In order to establish the global convergence of Algorithm 3.1, we make the following assumption.

Assumption 3.1. (i) The level set

$$\Omega = \{x \mid \|F(x)\| \le e^{\eta}\|F(x_0)\|\}$$

is bounded, where η is defined as in (3.3).

(ii) Function F is continuously differentiable and F' is Lipschitz continuous in Ω, i.e. there exists a constant $L > 0$, such that

$$\|F'(x) - F'(y)\| \leq L\|x - y\|, \quad \forall x, y \in \Omega. \tag{3.6}$$

(iii) The Jacobian $F'(x)$ is symmetric and nonsingular for any $x \in \Omega$.

For the sake of convenience, we introduce some notations. We denote

$$\xi_k = \frac{\|\sigma_k^T(F'(x_{k+1}) - B_k)\|_2}{\|\sigma_k\|_2},$$

and

$$\delta_k = \frac{\|(B_k - F'(x_{k+1}))s_k\|_2}{\|s_k\|_2}.$$

By the definition of $\sigma_k = (F'(x_{k+1}) - B_k)s_k$, we immediately have

$$\delta_k = \frac{\|\sigma_k\|^2}{\|s_k\| \cdot \|\sigma_k\|} = \frac{\sigma_k^T(F'(x_{k+1}) - B_k)s_k}{\|s_k\| \cdot \|\sigma_k\|} \leq \xi_k. \tag{3.7}$$

The following lemma is slightly an extension of Lemma 2.6 in [7].

Lemma 3.4. *Let the sequence $\{x_k\}$ be generated by Algorithm 3.1. Suppose that the conditions in Assumption 3.1 hold. If*

$$\sum_{k=0}^{\infty} \|s_k\|^2 < \infty.$$

then we have

$$\lim_{l \to \infty} \frac{1}{l} \sum_{k=0}^{l-1} \xi_k^2 = 0.$$

In particular, there is a subsequence of $\{\xi_k\}$ tending to zero. If we further assume

$$\sum_{k=0}^{\infty} \|s_k\| < \infty,$$

then we have

$$\sum_{k=0}^{\infty} \xi_k^2 < \infty.$$

In particular, the whole sequence $\{\xi_k\}$ converges to zero.

Proof. By the adjoint tangent condition (1.3), we have $\sigma_k^T F'(x_{k+1}) = \sigma_k^T B_{k+1}$. We also have from the Lipschitz continuality of $F'(x)$

$$\|F'(x_{k+1}) - F'(x_k)\|_F \leq L\|s_k\|.$$

Let

$$a_k = \|B_k - F'(x_k)\|_F \text{ and } b_k = \|F'(x_{k+1}) - F'(x_k)\|_F.$$

Then by (3.1), we can deduce

$$B_{k+1} - F'(x_{k+1}) \quad = \quad B_k - F'(x_{k+1}) + \theta_k \frac{\sigma_k\sigma_k^T(F'(x_{k+1}) - B_k) + (\sigma_k^T(F'(x_{k+1}) - B_k)^T\sigma_k^T}{\sigma_k^T\sigma_k}$$

$$-\theta_k^2 \frac{\sigma_k^T (F'(x_{k+1}) - B_k)\sigma_k}{\sigma_k^T \sigma_k} \cdot \frac{\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k}$$
$$= (I - \theta_k \frac{\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k})(B_k - F'(x_{k+1}))(I - \theta_k \frac{\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k}).$$

This implies

$$\begin{aligned} a_{k+1}^2 &= \|(I - \theta_k \frac{\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k})(B_k - F'(x_{k+1}))(I - \theta_k \frac{\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k})\|_F^2 \\ &\le \|(I - \theta_k \frac{\sigma_k \sigma_k^T}{\sigma_k^T \sigma_k})(B_k - F'(x_{k+1}))\|_F^2 \\ &= \|B_k - F'(x_{k+1})\|_F^2 - \theta_k(2-\theta_k)\frac{\|\sigma_k \sigma_k^T (B_k - F'(x_{k+1}))\|_F^2}{\|\sigma_k\|^2}, \end{aligned}$$

Taking into account θ_k satisfying $\theta_k(2-\theta_k) \ge (1-\hat{\theta}^2)$, we get

$$(1-\hat{\theta}^2)\xi_k^2 \le \|B_k - F'(x_{k+1})\|_F^2 - a_{k+1}^2 \le (a_k + b_k)^2 - a_{k+1}^2 = a_k^2 - a_{k+1}^2 + 2a_k b_k + b_k^2,$$

which implies

$$a_{k+1}^2 \le (a_k + b_k)^2 - (1-\hat{\theta}^2)\xi_k^2.$$

The desired results then follow from Lemma 2.5 of [7]. □

As a corollary of Lemma 3.4 and inequality (3.7), we immediately have the following lemma.

Lemma 3.5. *Let the conditions in Assumption 3.1 hold and $\{x_k\}$ be generated by Algorithm 3.1. Then we have*

$$\lim_{l\to\infty} \frac{1}{l} \sum_{k=0}^{l-1} \delta_k^2 = 0.$$

In particular, there is an infinite index set K such that the subsequence $\{\delta_k\}_K$ converges to zero.

The following theorem establishes the global convergence of Algorithm 3.1.

Theorem 3.6. *Let the conditions in Assumption 3.1 hold. Then the sequence $\{x_k\}$ generated by Algorithm 3.1 converges to the unique solution of (1.1).*

Proof. By Lemma 3.3, the sequence $\{\|F(x_k)\|\}$ converges. It then suffices to verify

$$\liminf_{k\to\infty} \|F(x_k)\| = 0. \tag{3.8}$$

If there are infinitely many k for which α_k is determined by (3.5), then the inequality $\|F(x_{k+1})\| \le \lambda \|F(x_k)\|$ holds for infinitely many k. Let I be the index set for which (3.5) holds and i_k be the number of index j satisfying $j \le k$ and $j \in I$. It is clear that $i_k \to \infty$, as $k \to \infty$. For any $j \notin I$, we have $\|F(x_{j+1})\| \le (1+\eta_j)\|F(x_j)\|$. Therefore, we have for all k sufficiently large

$$\|F(x_{k+1})\| \le \lambda^{i_{k+1}} \prod_{i=0}^{k} (1+\eta_i)\|F(x_0)\| \le \lambda^{i_{k+1}} e^{\eta} \|F(x_0)\|,$$

where $\lambda \in (0,1)$. The last inequality yields (3.8).

Consider the case that there are only finitely many indices k for which α_k is determined by (3.5). According to Lemma 3.5, there is an infinite set K such that $\{\delta_k\}_K \to 0$. It is then not difficult to show by (3.4) that the sequence $\{\|d_k\|\}_K$ is bounded. Without loss of generality, we assume that $\{d_k\}_K \to \bar{d}$, which implies $B_k d_k \to F'(\bar{x})\bar{d}$ as $k \to \infty$ with $k \in K$. Taking limit in (3.4) as $k \to \infty$ with $k \in \bar{K}$, we obtain

$$F'(\bar{x})\bar{d} + F(\bar{x}) = 0. \tag{3.9}$$

Denote $\bar{\alpha} = \limsup_{k\in K, k\to\infty} \alpha_k$. It is clear that $\bar{\alpha} \geq 0$. If $\bar{\alpha} > 0$, then $\bar{d} = 0$ and hence $F(\bar{x}) = 0$. Suppose $\bar{\alpha} = 0$, or equivalently $\lim_{k\in K, k\to\infty} \alpha_k = 0$. By the line search rule, when $k \in K$ is sufficiently large,

$$\|F(x_k + \rho^{-1}\alpha_k d_k)\| - \|F(x_k)\| > -\mu_1\|\rho^{-1}\alpha_k d_k\|^2.$$

Multiplying both sides of the last inequality by

$$(\|F(x_k + \rho^{-1}\alpha_k d_k)\| + \|F(x_k)\|)/\rho^{-1}\alpha_k$$

and then taking limits as $k \to \infty$ with $k \in K$ yields

$$F(\bar{x})^T F'(\bar{x})\bar{d} \geq 0.$$

This together with (3.9) implies $F(\bar{x}) = 0$. The proof is complete. □

To get the superlinear convergence of Algorithm 3.1, we first show the following lemma.

Theorem 3.7. *Let the conditions in in Assumption 3.1 hold and the sequence $\{x_k\}$ be generated by Algorithm 3.1. Then there exist a constant $\delta > 0$ and an index $\bar{k}$ such that the inequality*

$$\|F(x_k + d_k)\| \leq \lambda\|F(x_k)\| - \mu_2\|d_k\|^2 < \lambda\|F(x_k)\|$$

holds for all $k \geq \bar{k}$ and $\delta_k \leq \delta$. In particular, $\alpha_k = 1$ whenever $\delta_k \leq \delta$ and $k \geq \bar{k}$.

Proof. Since by Theorem 3.6, $\{x_k\}$ converge to the unique solution of (1.1), say x^*, there exists a constant $C > 0$ such that $\|F'(x_k)^{-1}\| \leq C$ for all k sufficiently large. Similar to the proof of Theorem 3.6, it is not difficult to show that there are constants $\bar{\delta}$ and $M > 0$ such that the following inequality holds for all k satisfying $\delta_k \leq \bar{\delta}$

$$\| d_k \| \leq M\|F(x_k)\|.$$

We also have from (3.4)

$$\begin{aligned} F'(x_k)(x_k + d_k - x^*) &= F'(x_k)(x_k - x^*) + (F'(x_k) - B_k)d_k - (F(x_k) - F(x^*)) \\ &= (F'(x_k) - F'(x^*))(x_k - x^*) + (F'(x_k) - B_k)d_k \\ &\quad -F(x_k) + F(x^*) + F'(x^*)(x_k - x^*). \end{aligned}$$

This implies

$$\begin{aligned} \|x_k + d_k - x^*\| &\leq \|F'(x_k)^{-1}\|(\|F'(x_k) - F'(x^*)\|\|x_k - x^*\| + \|(F'(x_k) - B_k)d_k\| \\ &\quad + \|F(x_k) - F(x^*) - F'(x^*)(x_k - x^*)\|) \\ &\leq C(o(\|x_k - x^*\|) + \delta_k\|d_k\|) \\ &\leq C(o(\|x_k - x^*\|) + \delta_k M\|F(x_k) - F(x^*)\|) \end{aligned}$$

$$\leq \quad C(o(\|x_k - x^*\|) + \delta_k M\bar{M}\|x_k - x^*\|), \tag{3.10}$$

where $\bar{M}$ is an upper bound of $F'(x)$ in Ω. Also we have

$$\begin{aligned} \|F(x_k + d_k)\| &= \|F(x_k + d_k) - F(x^*)\| \\ &\leq \bar{M}\|x_k + d_k - x^*\| \\ &\leq C\bar{M}(o(\|x_k - x^*\|) + \delta_k M\bar{M}\|x_k - x^*\|). \end{aligned}$$

On the other hand, by the nonsingularity of $F'(x^*)$ and the fact $\{x_k\} \to x^*$, there is a constant $\bar{m} > 0$ such that the inequality

$$\|F(x_k)\| = \|F(x_k) - F(x^*)\| \geq \bar{m}\|x_k - x^*\| = \bar{m}\|x_k - x^*\| \tag{3.11}$$

holds for all k sufficiently large. It is not difficult to see from (3.10) and (3.11) that there is a constant $\delta' \leq \bar{\delta}$ such that when k is sufficiently large and $\delta_k \leq \delta'$, the inequality

$$\begin{aligned} & \|F(x_k + d_k)\| - \lambda\|F(x_k)\| + \mu_2\|d_k\|^2 \\ \leq \quad & C\bar{M}(o(\|x_k - x^*\|) + \delta_k M\bar{M}\|x_k - x^*\|) \\ & -\lambda\bar{m}\|x_k - x^*\| + \mu_2 M^2\|F(x_k)\|^2 \\ = \quad & -(\lambda\bar{m} - CM\bar{M}\delta_k)\|x_k - x^*\| + o(\|x_k - x^*\|) \end{aligned}$$

is satisfied for all k sufficiently large and $\delta_k \leq \delta'$. The proof is complete. □

The following theorem establishes the superlinear convergence of Algorithm 3.1.

Theorem 3.8. *Let the conditions in Assumption 3.1 hold. Then the sequence $\{x_k\}$ generated by Algorithm 3.1 converges to the unique solution x^* of (1.1) superlinearly.*

Proof. It follows from Theorem 3.7 and (3.10) that we only need to show $\{\delta_k\} \to 0$ as $k \to \infty$.

Let δ and $\bar{k}$ be as specified by Theorem 3.7. It follows from Lemma 3.5 that there is an index $\tilde{k}$ such that the following inequality hold for all $k \geq \tilde{k}$

$$\frac{1}{k}\sum_{j=0}^{k-1} \delta_j^2 \leq \frac{1}{2}\delta^2.$$

This shows that for any $k \geq \tilde{k}$, there are at least $\lceil \frac{k}{2} \rceil$ many indices $j \leq k$ such that $\delta_j \leq \delta$. Let $k^{'} = \max\{\bar{k}, \tilde{k}\}$. By Theorem 3.7, for any $k \geq 2k^{'}$, there are at least $\lceil \frac{k}{2} \rceil - k^{'}$ many indices $j \leq k$ such that $\alpha_j = 1$ and

$$\|F(x_{j+1})\| = \|F(x_j + d_k)\| \leq \lambda\|F(x_j)\|. \tag{3.12}$$

Let J_k be the set of indices for which (3.12) holds and j_k be the number of elements in J_k. Then $j_k \geq \frac{k}{2} - k^{'} - 1$. On the other hand, for each $j \notin J_k$, we have

$$\|F(x_{j+1})\| \leq (1 + \eta_k)\|F(x_j)\|. \tag{3.13}$$

Multiplying inequalities (3.12) with $j \in J_k$ and (3.13) with $j \notin J_k$, we can get for any $k > 2k'$

$$\|F(x_{k+1})\| \leq \lambda^{j_k}\|F(x_{k'})\|[\Pi_{j=k'}^{k}(1 + \eta_j)] \leq \|F(x_{k'})\|\lambda^{\frac{1}{2}k - k^{'} - 1}e^{\eta},$$

or equivalently

$$\|F(x_{k+1})\| \le \|F(x_{k'})\|\lambda^{\frac{k}{2}-k'-1}e^{\frac{\eta}{2}}.$$

So, we obtain

$$\sum_{k=0}^{\infty}\|F(x_k)\| < \infty.$$

This together with (3.11) implies

$$\sum_{k=0}^{\infty}\|s_k\| < \infty.$$

It then follows from Lemma 3.4 and (3.7) that $\{\delta_k\} \to 0$ as $k \to \infty$. Then the superlinear convergence follows from (3.10). □

4 Numerical Experiments

In this section, we will test the performance of the Symmetric Adjoint Broyden method (SAB) and compare it with PSB, BFGS methods and the Broyden-like method in [7] with the same line search. For all methods, we use the following condition as the termination criterion:

$$\|F(x_k)\|_2 \le 10^{-5}.$$

We also stop the iterative process if the total number of iterations has reached to 1500 while the last inequality is not satisfied. In that case, the method fails to find a solution of (1.1).

The parameters in the line search are specified as follows

$$\lambda = 0.9,\ \mu_1 = \mu_2 = 0.001,\ \beta = 0.45,\ \eta_k = \frac{1}{(k+1)^2}.$$

The initial matrix B_0 was set to be the identity matrix.

The computation of the adjoint Broyden update formula is based on the terms $F'(x)s$ and $\sigma^T F'(x)$, which can be obtanied by the AD tool TOMLAB/MAD v7.3. The numerical experiments were done by using MATLAB v7.10 on Core (TM) 2 PC with WinXP. The details of the problems are given as follows, where x_0 denotes the initial point.

Problem 1.

$$\begin{aligned} f_1(x) &= 9x_1 - x_2 + h^2\cos x_1, \\ f_i(x) &= 9x_i - x_{i-1} - x_{i+1} + h^2\cos x_i, i = 2,3,\ldots,n-1, \\ f_n(x) &= 9x_n - x_{n-1} - h^2\cos x_n, \\ h &= \frac{1}{n+1}. \end{aligned}$$

We will test the methods on this problem with different initial points.

Problem 2.

$$\begin{aligned} f_1(x) &= x_1(x_1^2 + x_2^2) - 1, \\ f_i(x) &= x_i(x_{i-1}^2 + x_i^2 + x_{i+1}^2) - 1,\ i = 2,3,\ldots,n-1, \\ f_n(x) &= x_n(x_{n-1}^2 + x_n^2). \\ x_0 &= (1,1,\ldots,1)^T. \end{aligned}$$

Problem 3. Trigonometric function [4]

$$\begin{aligned} f_i(x) &= 2\Big(n + i(1-\cos x_i) - \sin x_i - \sum_{j=1}^{n} \cos x_j\Big)(2\sin x_i - \cos x_i),\ i = 1,2,\ldots,n. \\ x_0 &= \Big(\frac{101}{100n}, \frac{101}{100n}, \ldots, \frac{101}{100n}\Big)^T. \end{aligned}$$

Problem 4. Logarithmic function [4]

$$\begin{aligned} f_i(x) &= \ln(x_i+1) - \frac{x_i}{n},\ i = 1,2,\ldots,n. \\ x_0 &= (1,1,\ldots,1)^T. \end{aligned}$$

Problem 5. Trigexp function [4]

$$\begin{aligned} f_1(x) &= 3x_1^3 + 2x_2 - 5 + \sin(x_1 - x_2)\sin(x_1 + x_2), \\ f_i(x) &= -x_{i-1}\exp(x_{i-1} - x_i) + x_i(4 + 3x_i^2) + 2x_{i+1} \\ &\quad + \sin(x_i - x_{i+1})\sin(x_i + x_{i+1}) - 8,\ i = 2,3,\ldots,n-1, \\ f_n(x) &= -x_{n-1}\exp(x_{n-1} - x_n) + 4x_n - 3. \\ x_0 &= (0,0,\ldots,0)^T. \end{aligned}$$

Problem 6. Extended Freudenstein and Roth function (n is even) [4]

$$\begin{aligned} f_{2i-1} &= x_{2i-1} + ((5 - x_{2i})x_{2i} - 2)x_{2i} - 13, \\ f_{2i} &= x_{2i-1} + ((1 + x_{2i})x_{2i} - 14)x_{2i} - 29,\ i = 1,2,\ldots,n/2. \\ x_0 &= (6,3,6,3,\ldots,6,3)^T. \end{aligned}$$

Problem 7. Discrete boundary value problem [4]

$$\begin{aligned} f_1(x) &= 2x_1 + 0.5h^2(x_1 + h)^3 - x_2, \\ f_i(x) &= 2x_i + 0.5h^2(x_i + hi)^3 - x_{i-1} + x_{i+1},\ i = 2,3,\ldots,n-1, \\ f_n(x) &= 2x_n + 0.5h^2(x_n + hn)^3 - x_{n-1}, \\ h &= \frac{1}{n+1}. \\ x_0 &= (h(h-1), h(2h-1), \ldots, h(nh-1))^T. \end{aligned}$$

Problem 8. Troesch problem [11]

$$\begin{aligned} f_1(x) &= 2x_1 + \rho h^2 \sinh \rho x_1 - x_2, \\ f_i(x) &= 2x_i + \rho h^2 \sinh \rho x_i - x_{i-1} - x_{i+1}, i = 2,3,\ldots,n-1, \\ f_n(x) &= 2x_n + \rho h^2 \sinh \rho x_n - x_{n-1} - 1, \\ \rho &= 10,\ h = \frac{1}{n+1}. \\ x_0 &= (1,1,\ldots,1)^T. \end{aligned}$$

Problem 9. The discretized Chandrasehar's H-equation [4]

$$f_i(x) = x_i - \Big(1 - \frac{0.9}{2n}\sum_{j=1}^{n} \frac{\mu_i x_j}{\mu_i + \mu_j}\Big)^{-1},\ i = 1,2,\ldots,n,$$

$$\begin{aligned} \mu_i &= (i-0.5)/n, 1 \le i \le n. \\ x_0 &= (1,1,\ldots,1)^T. \end{aligned}$$

Problem 10. [18]

$$\begin{aligned} f_1(x) &= 2x_1 - x_2 + \sin x_1 - 1, \\ f_i(x) &= 2x_i - x_{i+1} + \sin x_i - 1, i = 2,3,\ldots,n-1, \\ f_n(x) &= 2x_n + \sin x_n - 1. \\ x_0 &= (0.1, 0.1, \ldots, 0.1)^T. \end{aligned}$$

Problem 11. [7]

$$\begin{aligned} f_1(x) &= 2x_1 - x_2 + h^2(\arctan x_1 - 1), \\ f_i(x) &= 2x_i - x_{i-1} - x_{i+1} + h^2(\arctan x_i - 1), i = 2,3,\ldots,n-1, \\ f_n(x) &= 9x_n - x_{n-1} - h^2(\arctan x_n - 1), \\ h &= \frac{1}{n+1}. \end{aligned}$$

We will test the methods on the problem with different initial points.

The following three tables report the numerical results, where each column of the tables has the following meaning:

Init:	initial point;
n:	the dimension of the problem;
Iter	the total numbers of iterations;
$\|F(x_k)\|$:	the norm of the residual at the stopping point;
P:	the problem;
" − ":	method failed to find the solution of the problem.

We first compared the proposed method SAB with the PSB and BFGS methods on Problem 1 with different initial points. Those initial points are set to $x_0 = (0,0,\ldots,0)^T$, $x_1 = (1,1,\ldots,1)^T$, $x_2 = (10,10,\ldots,10)^T$, $x_3 = (100,100,\ldots,100)^T$, $x_4 = (1000,1000,\ldots,1000)^T$, $x_5 = (1,2,\ldots,n)^T$, $x_6 = (n,n-1,\ldots,1)^T$, $x_7 = -x_1$, $x_8 = -x_2$, $x_9 = -x_3$, $x_{10} = -x_4$, $x_{11} = -x_5$, $x_{12} = -x_6$, respectively. Table 1 lists numerical results. It indicates that the proposed method performs well for Problem 1. Moreover, the initial points seems not to affect the numbers of iteration very much.

We then compared the SAB, PSB, BFGS methods on Problems 2-10 with different dimensions. Table 2 lists the numerical results. The results show that SAB performed much better than PSB in the number of iterations. Compared with BFGS method, SAB also has better performance for some problems.

Following an anonymous referee's suggestion, we compared the performance of the proposed method SAB with that of the Broyden-like method in [7]. We tested both methods on Problem 11 with the same initial points given in Table 1. The results were listed in Table 3. The results in the table show that the proposed method performed better than the Broyden-like method did in the number of iterations, which verified that the SAB has better performance for symmetric nonlinear equations.

5 Final Remarks

We have derived a class of symmetric adjoint Broyden methods. They share some nice properties with its non-symmetric version and the well-known PSB method. By the use of

Table 1: Results for Problem 1

Init	n	SAB method		PSB method		BFGS method	
		Iter	$\|F(x_k)\|$	Iter	$\|F(x_k)\|$	Iter	$\|F(x_k)\|$
x_0	50	8	1.0031e-006	9	1.8136e-006	6	1.7050e-006
	100	5	9.8282e-006	5	2.4834e-006	5	1.1173e-006
	200	1	3.6917e-006	1	3.6917e-006	1	3.6917e-006
	500	1	8.9086e-006	1	8.9086e-006	1	8.9086e-006
x_1	50	20	2.1078e-006	31	1.2766e-006	21	1.7145e-006
	100	20	2.1067e-006	35	2.1423e-006	21	1.9059e-006
	200	20	2.1065e-006	36	3.3080e-006	22	2.0895e-006
	500	21	2.0378e-006	33	2.9456e-006	23	2.3547e-006
x_2	50	23	2.0158e-006	33	1.7489e-006	25	1.2106e-006
	100	23	2.0139e-006	42	2.1751e-006	24	1.3407e-006
	200	23	2.0134e-006	42	3.3109e-006	24	1.4515e-006
	500	23	1.5288e-006	40	2.6379e-006	25	1.5854e-006
x_3	50	26	1.8505e-006	36	1.7085e-006	27	1.5942e-006
	100	26	1.8498e-006	49	2.1925e-006	27	1.7725e-006
	200	26	1.8496e-006	50	2.3374e-006	27	1.9448e-006
	500	27	2.2348e-006	46	3.7093e-006	27	2.1409e-006
x_4	50	29	2.8375e-006	36	1.8177e-006	30	1.3.67e-006
	100	29	2.8368e-006	55	2.9192e-006	30	2.3297e-006
	200	26	1.2756e-006	56	3.4038e-006	30	2.5585e-006
	500	26	1.2756e-006	54	2.3772e-006	31	1.3902e-006
x_5	50	26	6.6734e-006	43	1.4629e-006	26	1.9809e-006
	100	26	1.3211e-006	47	1.7986e-006	27	1.1891e-006
	200	27	1.8280e-006	51	1.9210e-006	28	1.3128e-006
	500	26	1.3033e-006	55	2.3954e-006	30	9.6988e-006
x_6	50	26	6.6734e-006	43	1.4629e-006	26	1.9809e-006
	100	26	1.3211e-006	47	1.7986e-006	27	1.1891e-006
	200	27	1.8280e-006	51	1.9210e-006	28	1.3128e-006
	500	26	1.3033e-006	55	2.3954e-006	30	9.6988e-006
x_7	50	20	2.1078e-006	31	1.2766e-006	21	1.7145e-006
	100	20	2.1068e-006	35	2.1426e-006	21	1.9060e-006
	200	20	2.1065e-006	36	3.3081e-006	22	2.0895e-006
	500	21	2.0378e-006	33	2.9456e-006	23	2.3547e-006
x_8	50	23	2.0158e-006	33	1.7488e-006	25	1.2106e-006
	100	23	2.0139e-006	42	2.1752e-006	24	1.3407e-006
	200	23	2.0134e-006	42	3.3109e-006	24	1.4515e-006
	500	23	1.5288e-006	40	2.6379e-006	25	1.5854e-006
x_9	50	26	1.8505e-006	36	1.7085e-006	27	1.5942e-006
	100	26	1.8498e-006	49	2.1925e-006	27	1.7725e-006
	200	26	1.8496e-006	50	2.3374e-006	27	1.9448e-006
	500	27	2.2348e-006	46	3.7093e-006	27	2.1409e-006
x_{10}	50	29	2.8375e-006	36	1.8177e-006	30	1.3.67e-006
	100	29	2.8368e-006	55	2.9192e-006	30	2.3297e-006
	200	26	1.2756e-006	56	3.4038e-006	30	2.5585e-006
	500	26	1.2756e-006	54	2.3772e-006	31	1.3902e-006
x_{11}	50	26	6.6734e-006	43	1.4629e-006	26	1.9809e-006
	100	26	1.3211e-006	47	1.7986e-006	27	1.1891e-006
	200	27	1.8280e-006	51	1.9210e-006	28	1.3128e-006
	500	26	1.3033e-006	55	2.3954e-006	30	9.6988e-006
x_{12}	50	26	6.6734e-006	43	1.4629e-006	26	1.9809e-006
	100	26	1.3211e-006	47	1.7986e-006	27	1.1891e-006
	200	27	1.8280e-006	51	1.9210e-006	28	1.3128e-006
	500	26	1.3033e-006	55	2.3954e-006	30	9.6988e-006

Table 2: Results for Problems 2-10

P	n	SAB method		PSB method		BFGS method	
		Iter	$\|F(x_k)\|$	Iter	$\|F(x_k)\|$	Iter	$\|F(x_k)\|$
2	10	20	7.2821e-006	14	3.4576e-006	13	1.0168e-006
	50	24	2.1956e-006	18	1.6677e-006	14	1.3093e-006
	100	24	5.2156e-006	19	1.3138e-006	14	2.0005e-006
	200	22	2.5359e-006	19	1.6524e-006	15	1.2079e-006
	500	20	1.4271e-006	21	1.3632e-006	16	9.9629e-006
3	10	15	1.9674e-006	26	8.4612e-06	15	5.4518e-006
	50	9	8.6762e-006	-	-	11	8.5619e-006
	100	10	4.1627e-006	-	-	10	2.9304e-006
	200	10	3.0298e-006	-	-	10	6.6604e-006
	500	10	8.4306e-006	-	-	10	5.6241e-006
4	10	8	1.7963e-006	6	1.1667e-006	6	1.1667e-006
	50	11	9.8867e-006	5	6.8193e-006	5	6.8193e-006
	100	10	6.1476e-006	5	8.1998e-006	5	8.1998e-006
	200	8	5.9290e-006	5	1.0671e-006	5	1.0671e-006
	500	8	8.3964e-007	5	1.6041e-006	5	1.6041e-006
5	10	23	1.1951e-006	24	5.1698e-006	21	1.2487e-006
	50	25	1.2672e-006	74	8.5351e-006	55	2.1606e-006
	100	25	1.2672e-006	133	8.2265e-006	71	1.8686e-006
	200	25	1.2672e-006	247	2.6519e-006	62	2.0457e-006
	500	25	1.2672e-006	768	5.6172e-006	63	2.5031e-006
6	10	73	2.6993e-006	12	7.6348e-007	76	2.1456e-006
	50	73	6.0359e-006	15	1.0587e-006	71	1.2235e-006
	100	73	8.5360e-006	20	1.2388e-006	73	6.5660e-006
	200	73	1.2072e-006	14	1.4580e-006	77	1.3194e-006
	500	73	1.9087e-007	17	4.6552e-006	76	2.3098e-006
7	10	38	1.5645e-006	28	1.3713e-006	68	1.4533e-006
	50	30	1.4713e-006	63	1.1735e-006	130	1.3065e-006
	100	27	1.5720e-006	72	2.8461e-006	158	1.8402e-006
	200	27	1.2690e-006	52	3.1887e-006	145	2.0925e-006
	500	27	1.4797e-007	42	3.5336e-006	83	2.0047e-006
8	10	27	1.7294e-006	254	9.8492e-006	53	3.8359e-006
	50	54	6.1756e-006	310	9.8170e-006	57	9.4744e-006
	100	84	2.5792e-006	296	9.9157e-006	156	8.2504e-006
	200	166	4.4425e-006	473	9.9803e-006	183	5.0366e-006
	500	419	5.7041e-006	689	9.5861e-006	465	8.1274e-006
9	10	2	7.2341e-006	2	6.6658e-007	2	6.6667e-007
	50	2	1.3121e-007	2	1.4100e-008	2	1.4101e-008
	100	1	1.4018e-006	1	1.4018e-006	1	1.4018e-006
	200	1	4.9502e-006	1	4.9502e-006	1	4.9502e-006
	500	1	1.2514e-006	1	1.2514e-006	1	1.2514e-006
10	10	19	1.3284e-006	18	8.5437e-007	18	2.7926e-006
	50	19	1.3285e-006	29	1.9860e-006	22	1.6419e-006
	100	19	1.3285e-006	29	1.8687e-006	22	1.4387e-006
	200	19	1.3285e-006	29	1.8150e-006	22	1.3489e-006
	500	19	1.3285e-006	29	1.8311e-006	22	1.4541e-006

Table 3: Results for Problem 11

Init	Broyden-like method			SAB method		
	$n=9$	$n=49$	$n=99$	$n=9$	$n=49$	$n=99$
x_0	14	61	127	6	26	75
x_1	13	66	122	9	40	99
x_2	16	69	124	9	43	97
x_3	16	66	127	9	54	171
x_4	19	70	186	9	51	101
x_5	14	66	123	8	39	74
x_6	15	68	124	9	37	100
x_7	16	67	127	10	53	169
x_8	19	74	187	10	53	99
x_9	15	61	116	12	52	117
x_{10}	15	61	116	13	52	123
x_{11}	15	61	116	12	52	117
x_{12}	15	61	116	13	52	123

a nonmonotone line search, we proposed a symmetric adjoint Broyden algorithm for solving symmetric nonlinear equations and established its global and superlinear convergence. Our limited numerical experiments verified the efficiency of the proposed method.

Acknowledgement

The authors would like to thank anonymous referees for their valuable comments and suggestions which improve the paper.

References

[1] J.E. Dennis and J.J. Moré, Quasi-Newton methods, motivation and theory, *SIAM rev.* 19 (1977) 46–89.

[2] A. Griewank and A. Walther, On constrained optimization by adjoint based quasi-Newton methods, *Optim. Methods Softw.* 17 (2002) 869–889.

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

[4] W. La Cruz, J. Martínez, and M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, *Math. Comp.* 75 (2006) 1429–1448.

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

[6] D.H. Li and M. Fukushima, A derivative-free line search and DFP method for symmetric equations with global and superlinear convergence, *Numer. Funct. Anal. Optim.* 20 (1999) 59–77.

[7] D.H. Li and M. Fukushima, A derivative-free line search and global convergence of Broyden-like method for nonlinear equations, *Optim. Methods Softw.* 13 (2000) 181–201.

[8] D.H. Li and X.L. Wang, A modified Fletcher-Reeves-type derivative-free method for symmetric nonlinear equations, *Numer. Algebra Control Optim.* 1 (2011) 71–82.

[9] J. J. Moré and J. A. Trangenstein, On the global convergence of Broyden's method, *Math. Comp.* 30 (1976) 523–540.

[10] M.J.D. Powell, A new algorithm for unconstrained optimization, in: J.B. Rosen, O.L. Mangasarian and K. Ritter, eds., *Nonlinear Programming*, Academic Press, New York, 1970, pp. 31–66.

[11] S.M. Roberts and J.S. Shipman, On the closed form solution of Troesch's problem, *J. Comput. Phys.* 21 (1976) 291–304.

[12] S. Schlenkrich, A. Griewank and A. Walther, On the local convergence of adjoint Broyden methods, *Math. Program.* 121 (2010) 221–247.

[13] S. Schlenkrich and A. Walther, Adjoint-based quasi-Newton methods for partially separable problems, *PAMM* 7 (2007) 2020091–2020092.

[14] S. Schlenkrich and A. Walther, Global convergence of quasi-Newton methods based on adjoint Broyden updates, *Appl. Numer. Math.* 59 (2009) 1120–1136.

[15] Y.X. Yuan and W.Y. Sun, *Optimization Theory and Methods*, Science Press, Beijing, 1997.

[16] G.L. Yuan and S.W. Yao, A BFGS algorithm for solving symmetric nonlinear equations, *Optimization* 62 (2013) 85–99.

[17] L. Zhang, W.J. Zhou and D.H. Li, Global convergence of a modified Fletcher Reeves conjugate gradient method with Armijo-type line search, *Numer. Math.* 104 (2006) 561–572.

[18] W.J. Zhou and D.M. Shen, An inexact PRP conjugate gradient method for symmetric nonlinear equations, *Numer. Funct. Anal. Optim.* 35 (2014) 370–388,

Manuscript received 5 April 2014
revised 2 September 2014
accepted for publication 29 September 2014

Hui-Ping Cao
College of Science
Central South University of Forestry and Technology
Changsha 410004, China
E-mail address: huiping_cao@hnu.edu.cn

Dong-Hui Li
School of Mathematical Sciences
South China Normal University
Guangzhou 510631, China
E-mail address: dhli@scnu.edu.cn