Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2014, Article ID 251587, 12 pages
http://dx.doi.org/10.1155/2014/251587

Research Article

Global Convergence of Schubert's Method for Solving Sparse Nonlinear Equations

Huiping Cao

College of Mathematics and Econometrics, Hunan University, Changsha 410082, China

Correspondence should be addressed to Huiping Cao; huiping_cao@hnu.edu.cn

Received 7 May 2014; Accepted 28 July 2014; Published 14 October 2014

Academic Editor: Shuangjie Peng

Schubert's method is an extension of Broyden's method for solving sparse nonlinear equations, which can preserve the zero-nonzero structure defined by the sparse Jacobian matrix and can retain many good properties of Broyden's method. In particular, Schubert's method has been proved to be locally and q-superlinearly convergent. In this paper, we globalize Schubert's method by using a nonmonotone line search. Under appropriate conditions, we show that the proposed algorithm converges globally and superlinearly. Some preliminary numerical experiments are presented, which demonstrate that our algorithm is effective for large-scale problems.

1. Introduction

In this paper, we consider the quasi-Newton method [1] for solving the general nonlinear equation

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

where the function $F : R^n \to R^n$ is continuously differentiable. The ordinary quasi-Newton method for solving (1) generates a sequence $\{x_k\}$ by the following iterative scheme:

$$x_{k+1} = x_k + d_k, \tag{2}$$

where the quasi-Newton direction d_k is obtained by solving the system of linear equations

$$B_k d + F(x_k) = 0. \tag{3}$$

Here the matrix B_k is an approximation to the Jacobian matrix of F at x_k which usually satisfies the quasi-Newton condition (i.e., the secant condition):

$$B_{k+1} s_k = y_k, \tag{4}$$

where $s_k = x_{k+1} - x_k$ and $y_k = F(x_{k+1}) - F(x_k)$. The matrix B_k can be updated by different quasi-Newton update formulae. However, quasi-Newton method is not applicable for solving large-scale problems due to the density of B_k. Fortunately, a large-scale problem usually has the property of sparsity, and then it is natural to extend some known quasi-Newton methods based on this property. Early in 1970, Schubert [2] modified Broyden's method and proposed a sparse Broyden's method, that is, the Schubert's method [3] with B_k defined in the following way:

$$B_{k+1} = B_k + \sum_{i=1, s(i)_k \neq 0}^{n} e_i e_i^T \frac{(y_k - B_k s_k)\, s(i)_k^T}{s(i)_k^T s(i)_k}, \tag{5}$$

where

$$s(i)_k(j) = \begin{cases} 0, & \text{if } (i,j) \in E, \\ s_k(j), & \text{if } (i,j) \in J, \end{cases} \tag{6}$$

where E and J are the sparsity patterns of the Jacobian matrix $F'(x)$ and e_i denotes the ith column of the $n \times n$ identity matrix. It has been proved by Broyden [4] that the Schubert's method is locally convergent when the Jacobian satisfies a Lipschitz condition. Lam [5] further showed the local and superlinear convergence of Schubert's method in the special case when $s(i)_k \neq 0$, $i = 1, 2, \ldots, n$ at each iteration. As an improvement, Marwil considered the following updated formula:

$$B_{k+1} = B_k + \sum_{i=1}^{n} \left(s(i)_k^T s(i)_k \right)^+ e_i^T (y_k - B_k s_k)\, e_i s(i)_k^T, \tag{7}$$

where, for a scalar $\alpha \in R$,

$$\alpha^{+} = \begin{cases} \alpha^{-1}, & \text{if } \alpha \neq 0, \\ 0, & \text{if } \alpha = 0. \end{cases} \quad (8)$$

Marwil established stronger local and superlinear convergence, which contains the results in [5] as a special case.

Note that the updated formula (7) is not symmetric; therefore, its use is restricted to problems where the symmetry of the updated matrix is not important. The sparse and symmetric quasi-Newton update has attracted much attention [6–8]. Toint [6] and Fletcher [8] previously proposed symmetric updates which met the sparsity and secant conditions simultaneously. Yamashita [9] proposed a new sparse quasi-Newton update, called Matrix Completion Quasi-Newton (MCQN) [10], which exploited the sparsity of the Hessian and guaranteed positive definiteness.

So far, most studies in the convergence of sparse quasi-Newton methods have focused on their local behaviors. Seldom studies are concerned with the global convergence of those methods. It is a relatively more difficult research topic than optimization. To the author's knowledge, the main work related to the general global convergence of sparse quasi-Newton methods is based on the work [10–12]. The purpose of this paper is to study the global and superlinear convergence of the Schubert's method.

The remainder of this paper is organized as follows. In Section 2, we review some properties of Schubert's method. In Section 3, by using a nonmonotone line search [13], we globalize the Schubert's method and prove its global and superlinear convergence under appropriate conditions. In particular, we will show that after finitely many iterations, the unit step length will be accepted. In Section 4, some preliminary numerical results are presented. Finally, we provide some remarks in Section 5.

2. Schubert's Update

In this section, we present some useful properties of the Schubert's update. For the sake of convenience, we introduce some notations.

(1) The subspace $Z_i \subset R^n$ that identifies the sparsity structure of the ith row of Jacobian matrix $F'(x)$ is defined as $Z_i = \{v \in R^n \mid v^T F'(x) e_i = 0, \forall x \in D\}$;

(2) $Z \subset R^n \times R^n$ denotes the sparsity structure of the Jacobian matrix $F'(x)$ and it is defined as $Z = \{A \in R^{n\times n} \mid A^T e_i \in Z_i, i = 1, 2, \ldots, n\}$;

(3) the projection operators S_i, $i = 1, 2, \ldots, n$ project vectors onto the subspaces Z_i, $i = 1, 2, \ldots, n$, which particularly makes $S_i s_k = s(i)_k$;

(4) $Q(y, s) = \{B \in R^{n\times n} \mid Bs = y\}$.

Schubert's update (7) is the unique solution to the following minimization problem [3]:

$$\min \left\{ \left\| \widehat{B} - B \right\|_F : \widehat{B} \in Q(y, s) \cap Z \right\}. \quad (9)$$

Specifically, the following inequality [3] holds for any $J \in Z$, $y, s \in R^n$ with $s \neq 0$:

$$\begin{aligned} \left\| B_{k+1} - J \right\|_F^2 & \\ \leq \left\| B_k - J \right\|_F^2 - \frac{\left\| (B_k - J) s_k \right\|_2^2}{\left\| s_k \right\|_2^2} & \\ + \sum_{i=1}^{n} \left(s(i)_k^T s(i)_k \right)^{+} \left(e_i^T (y_k - J s_k) \right)^2 . & \end{aligned} \quad (10)$$

The following theorem states the local and superlinear convergence of Schubert's method, which has been proved by Marwil [3].

Theorem 1. *Suppose that $F : R^n \to R^n$ satisfies the following conditions.*

(1) *F is continuously differentiable in an open convex set D_0.*

(2) *There exists an $x^* \in D_0$ such that $F(x^*) = 0$ and $F'(x^*)$ is nonsingular.*

(3) *There exists $K = (k_1, k_2, \ldots, k_n)^T \in R^n$ with $k_i \geq 0$ for $i = 1, 2, \ldots, n$, such that*

$$\left\| e_i^T \left(F'(x) - F'(y) \right) \right\|_2 \leq k_i \|x - y\|_2, \quad \forall x, y \in D_0. \quad (11)$$

Then there exist constants ϵ, $\delta > 0$ and a nonsingular matrix B_0 such that if $x_0 \in D_0$ satisfies $\|x_0 - x^*\|_2 < \epsilon$ and $\|B_0 - F'(x^*)\|_F < \delta$, then

(i) *Schubert's method generates $\{B_k\}$ with B_k nonsingular for all $k \geq 0$;*

(ii) *the sequence $\{x_k\}$ converges to x^*;*

(iii) *the convergence is superlinear.*

It is noticed that the matrix B_{k+1} determined by (7) may be singular even if B_k is nonsingular. To overcome such a difficulty, Marwil [3] proposed a nonsingular schubert's method by using

$$\overline{B} = B + \sum_{i=1}^{n} \theta^i \left(s(i)^T s(i) \right)^{+} e_i^T (y - Bs) e_i s(i)^T. \quad (12)$$

Here, we have omitted the subscript k and used $\overline{B}$ to represent B_{k+1} and B to represent B_k. The parameters θ^i are chosen so that $\overline{B}$ is nonsingular when B is nonsingular and the details are given below.

Set $C_0 = B$ and define for $i = 1, 2, \ldots, n$ as follows:

$$\begin{aligned} C_i &= C_0 + \sum_{j=1}^{i} \theta^j \left(s(j)^T s(j) \right)^{+} e_j^T (y - Bs) e_j s(j)^T \\ &= C_{i-1} + \theta^i \left(s(i)^T s(i) \right)^{+} e_i^T (y - Bs) e_i s(i)^T. \end{aligned} \quad (13)$$

Note that $e_i^T C_0 = e_i^T C_1 = \cdots = e_i^T C_{i-1}$, and then

$$C_i = C_{i-1} + \theta^i \left(s(i)^T s(i) \right)^{+} e_i^T (y - C_{i-1} s) e_i s(i)^T. \quad (14)$$

For a scalar $\alpha \in (0,1)$, θ^i can be chosen to satisfy

$$\left|\det C_i\right| \geq \sqrt[n]{\alpha}\left|\det C_{i-1}\right|, \qquad \theta^i \in \left[\frac{1-\sqrt[n]{\alpha}}{1+\sqrt[n]{\alpha}}, 1\right]. \tag{15}$$

Therefore, $|\det(\overline{B})| \geq \alpha|\det B|$, and θ^i can be chosen so that

$$\overline{B} \text{ is nonsingular}, \quad \left|\theta^i - 1\right| \leq \widehat{\theta} < 1. \tag{16}$$

The dependence of θ^i on the iteration k is suppressed, but $\widehat{\theta}$ is independent of k [3].

3. The Algorithm

In this section, we will globalize Schubert's method. To this end, we introduce a derivative-free line search proposed by Li and Fukushima [13] to determine a step length α_k.

Algorithm 2. Given constants $\sigma_1 > 0$ and $\beta \in (0,1)$. Let $\alpha_k = \beta^{i_k}$, where i_k is the smallest nonnegative integer such that

$$\left\|F\left(x_k + \beta^{i_k} d_k\right)\right\| \leq \left\|F\left(x_k\right)\right\| - \sigma_1 \left\|\beta^{i_k} d_k\right\|^2 + \varepsilon_k \left\|F\left(x_k\right)\right\|, \tag{17}$$

and $\{\varepsilon_k\}$ is a given positive sequence satisfying

$$\sum_{k=0}^{\infty} \varepsilon_k \leq \varepsilon < \infty. \tag{18}$$

It is not difficult to see that Algorithm 2 is well defined. Moreover, for each k, we have

$$\left\|F\left(x_k + \alpha_k d_k\right)\right\| \leq \left(1+\varepsilon_k\right)\left\|F\left(x_k\right)\right\|. \tag{19}$$

By using Algorithm 2, we give our algorithm as follows.

Algorithm 3. Consider the following.

Step 0. Given constants $\rho, \beta, \overline{\theta} \in (0,1)$, $\sigma_1, \sigma_2 > 0$, select a positive sequence $\{\varepsilon_k\}$ satisfying (18). Then choose an initial point $x_0 \in R^n$ and a nonsingular matrix $B_0 \in R^{n\times n}$. Let $k := 0$.

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

$$B_k d + F\left(x_k\right) = 0 \tag{20}$$

to get d_k.

Step 2. If

$$\left\|F\left(x_k + d_k\right)\right\| \leq \rho\left\|F\left(x_k\right)\right\| - \sigma_2\left\|d_k\right\|^2, \tag{21}$$

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

Step 3. Let α_k be determined by Algorithm 2.

Step 4. Set $x_{k+1} := x_k + \alpha_k d_k$.

Step 5. Compute B_{k+1} by (12).

Step 6. Set $k := k+1$. Go to Step 1.

We then show some useful properties of Algorithm 3.

Lemma 4. *Let the level set $\Omega = \{x \in R^n \mid \|F(x)\| \leq e^{\varepsilon}\|F(x_0)\|\}$ be bounded and let $\{x_k\}$ be generated by Algorithm 3. Then $\{x_k\}$ is contained in Ω. Moreover, it holds that*

$$\sum_{k=0}^{\infty}\left\|s_k\right\|^2 < \infty. \tag{22}$$

Proof. By the line search (17), we have for any k

$$\begin{aligned}
\left\|F\left(x_{k+1}\right)\right\| &\leq \left(1+\varepsilon_k\right)\left\|F\left(x_k\right)\right\| \\
&\ \vdots \\
&\leq \left\|F\left(x_0\right)\right\|\left[\prod_{j=0}^{k}\left(1+\varepsilon_j\right)\right] \\
&\leq \left\|F\left(x_0\right)\right\|\left[\frac{1}{k+1}\sum_{j=0}^{k}\left(1+\varepsilon_j\right)\right]^{k+1} \\
&= \left\|F\left(x_0\right)\right\|\left[1+\frac{1}{k+1}\sum_{j=0}^{k}\varepsilon_j\right]^{k+1} \\
&\leq \left\|F\left(x_0\right)\right\|\left[\left(1+\frac{\varepsilon}{k+1}\right)^{(k+1)/\varepsilon}\right]^{\varepsilon} \\
&\leq e^{\varepsilon}\left\|F\left(x_0\right)\right\|.
\end{aligned} \tag{23}$$

This implies that the sequence $\{x_k\}$ generated by Algorithm 3 is contained in Ω and the sequence $\{\|F(x_k)\|\}$ is bounded. Moreover, combined with (17) and (21), we can get for each k

$$\begin{aligned}
\sigma_0\left\|s_k\right\|^2 &= \sigma_0\left\|x_{k+1} - x_k\right\|^2 \\
&\leq \left\|F\left(x_k\right)\right\| - \left\|F\left(x_{k+1}\right)\right\| + \varepsilon_k\left\|F_k\right\|,
\end{aligned} \tag{24}$$

where $\sigma_0 = \min\{\sigma_1, \sigma_2\}$. Making summation on both sides for k from 0 to ∞, we obtain (22). □

Similar to Lemma 2.4 in [13], we have the following result.

Lemma 5. *Let the level set Ω be bounded and let $\{x_k\}$ be generated by Algorithm 3. Then the sequence $\{\|F(x_k)\|\}$ is convergent.*

In order to establish the global convergence of Algorithm 3, we need the following assumptions.

Assumption 6. (i) The level set Ω defined in Lemma 4 is bounded.

(ii) $F(x)$ is continuously differentiable in an open set $D \subseteq \Omega$, and there exists an $x^* \in \Omega$ such that $F(x^*) = 0$ and $F'(x^*)$ is nonsingular.

(iii) $F'(x)$ is Lipschitz continuous on Ω; that is, there exists an $L > 0$ such that

$$\left\|F'(x) - F'(y)\right\| \leq L\left\|x - y\right\|, \quad \forall x, y \in \Omega. \tag{25}$$

(iv) $F'(x)$ is nonsingular for any $x \in \Omega$.

Assumption 6 (iv) is the same as that in [13], which is not as strong as the assumption adopted in [14], where the uniform nonsingularity of $F'(x)$ is assumed.

We first introduce some notations. Define

$$A_{k+1} = \int_0^1 F'(x_k + ts_k)\,dt, \tag{26}$$

and then we have $y_k = A_{k+1}s_k$. Let

$$\begin{aligned}\eta_k &= \left\|\sum_{i=1}^{n}\left(s(i)_k^T s(i)_k\right)^+ e_i^T (y_k - B_k s_k) e_i s(i)_k^T\right\|,\\ \delta_k &= \frac{\|y_k - B_k s_k\|}{\|s_k\|} = \frac{\|(A_{k+1} - B_k)s_k\|}{\|s_k\|}\\ &= \frac{1}{\|s_k\|}\left\|\sum_{i=1}^{n}\left(s(i)_k^T s(i)_k\right)^+ e_i^T (y_k - B_k s_k) e_i s(i)_k^T s_k\right\|.\end{aligned} \tag{27}$$

The following lemma is an extension of Lemma 2.5 in [13].

Lemma 7. *Let the sequence $\{x_k\}$ be generated by Algorithm 3. Suppose that the conditions (i)–(iii) in Assumption 6 hold. If*

$$\sum_{k=0}^{\infty}\|s_k\|^2 < \infty, \tag{28}$$

then one has

$$\lim_{l\to\infty}\frac{1}{l}\sum_{k=0}^{l-1}\delta_k^2 = 0. \tag{29}$$

In particular, there is a subsequence of $\{\delta_k\}$ tending to zero. If

$$\sum_{k=0}^{\infty}\|s_k\| < \infty, \tag{30}$$

then one has

$$\sum_{k=0}^{\infty}\delta_k^2 < \infty. \tag{31}$$

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

Proof. By the Lipschitz continuity of F', we have

$$\|A_{k+1} - A_k\| \le \frac{1}{2}L\left(\|s_k\| + \|s_{k-1}\|\right). \tag{32}$$

Denote

$$a_k = \|B_k - A_k\|_F, \qquad b_k = \|A_{k+1} - A_k\|_F. \tag{33}$$

According to the updated (12), we have

$$e_i^T B_{k+1} = e_i^T B_k + \left(s(i)_k^T s(i)_k\right)^+ \theta_k^i e_i^T (y_k - B_k s_k) s(i)_k^T. \tag{34}$$

Subtracting $e_i^T A_{k+1}$ from both sides of the above equality gives

$$\begin{aligned}&e_i^T (B_{k+1} - A_{k+1})\\ &\quad= e_i^T (B_k - A_{k+1}) + \theta_k^i e_i^T\left(s(i)_k^T s(i)_k\right)^+ (y_k - B_k s_k) s(i)_k^T\\ &\quad= e_i^T (B_k - A_{k+1})\left(I - \theta_k^i\left(s(i)_k^T s(i)_k\right)^+ s(i)_k s(i)_k^T\right).\end{aligned} \tag{35}$$

Taking norms yields

$$\begin{aligned}&\left\|e_i^T (B_{k+1} - A_{k+1})\right\|^2\\ &\quad= \left\|e_i^T (B_k - A_{k+1})\left(I - \theta_k^i\left(s(i)_k^T s(i)_k\right)^+ s(i)_k s(i)_k^T\right)\right\|^2\\ &\quad= \left\|e_i^T (B_k - A_{k+1})\right\|^2\\ &\qquad - \theta_k^i\left(2 - \theta_k^i\right)\left(s(i)_k^T s(i)_k\right)^+ \left\|e_i^T (B_k - A_{k+1}) s(i)_k s(i)_k^T\right\|^2.\end{aligned} \tag{36}$$

Making summation on both sides, $i = 1, \ldots, n$, yields

$$\begin{aligned}\|B_{k+1} - A_{k+1}\|_F^2 &= \sum_{i=1}^{n}\left\|e_i^T (B_{k+1} - A_{k+1})\right\|^2\\ &= \sum_{i=1}^{n}\left\|e_i^T (B_k - A_{k+1})\right\|^2\\ &\quad - \sum_{i=1}^{n}\theta_k^i\left(2 - \theta_k^i\right)\left(s(i)_k^T s(i)_k\right)^+\\ &\quad \times \left\|e_i^T (B_k - A_{k+1}) s(i)_k s(i)_k^T\right\|^2\\ &= \left\|(B_k - A_{k+1})\right\|_F^2\\ &\quad - \left\|\sum_{i=1}^{n}\theta_k^i\left(2 - \theta_k^i\right) e_i e_i^T\left(s(i)_k^T s(i)_k\right)^+\right.\\ &\qquad \left.\times (y_k - B_k s_k) s(i)_k^T\right\|^2\\ &\le \left\|(B_k - A_{k+1})\right\|_F^2 - \left(1 - \widehat{\theta}^2\right)\\ &\quad \times \left\|\sum_{i=1}^{n} e_i e_i^T\left(s(i)_k^T s(i)_k\right)^+ (y_k - B_k s_k) s(i)_k^T\right\|^2\\ &= \left\|(B_k - A_{k+1})\right\|_F^2 - \left(1 - \widehat{\theta}^2\right)\eta_k^2.\end{aligned} \tag{37}$$

Then it follows that

$$\begin{aligned}\left(1 - \widehat{\theta}^2\right)\eta_k^2 &\le \|B_k - A_{k+1}\|_F^2 - a_{k+1}^2\\ &\le (a_k + b_k)^2 - a_{k+1}^2\\ &= a_k^2 - a_{k+1}^2 + 2a_k b_k + b_k^2,\\ a_{k+1}^2 &\le (a_k + b_k)^2 - \left(1 - \widehat{\theta}^2\right)\eta_k^2.\end{aligned} \tag{38}$$

According to Lemma 2.5 of [13], we get

$$\begin{aligned}&\lim_{l\to\infty}\frac{1}{l}\sum_{k=0}^{l-1}\eta_k^2 = 0,\\ &\sum_{k=0}^{\infty}\eta_k^2 < \infty.\end{aligned} \tag{39}$$

Moreover, for each k we have

$$\delta_k = \frac{1}{\|s_k\|}\left\|\sum_{i=1}^{n}\left(s(i)_k^T s(i)_k\right)^+ e_i^T\left(y_k - B_k s_k\right) e_i s(i)_k^T s_k\right\| \leq \left\|\sum_{i=1}^{n}\left(s(i)_k^T s(i)_k\right)^+ e_i^T\left(y_k - B_k s_k\right) e_i s(i)_k^T\right\| = \eta_k. \tag{40}$$

This completes the proof. □

According to Algorithm 3, we have the following lemma.

Lemma 8. *Let $\{x_k\}$ be generated by Algorithm 3. If there are finitely many k for which α_k is determined by* (21)*, then one has*

$$\lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t}\sum_{i=1}^{m}\delta_k^2 = 0. \tag{41}$$

In particular, there is an infinity index set K such that, for each $i = 1, 2, \ldots, m$, the subsequence $\{\delta_k\}_{k\in K}$ converges to zero.

Proof. Since there are finitely many k for which α_k is determined by (21), we can know that there exists an index $\hat{k}$ such that, for $k > \hat{k}$,

$$\left\|F\left(x_k + \rho^i d_k\right)\right\| \leq \|F(x_k)\| - \sigma_1\left\|\rho^i d_k\right\|^2 + \epsilon_k\|F(x_k)\|. \tag{42}$$

This implies

$$\sum_{k=\hat{k}}^{\infty}\|s_k\|^2 = \sum_{k=\hat{k}}^{\infty}\|x_{k+1} - x_k\|^2 = \sum_{k=\hat{k}}^{\infty}\|\alpha_k d_k\|^2 < \infty. \tag{43}$$

According to Lemma 7, we can easily prove the result. □

Lemma 9. *Suppose that $\sum_{k=1}^{\infty}\|s_k\|^2 < \infty$ and that there is an accumulation point x^* of $\{x_k\}_{k\in K}$ at which $F'(x^*)$ is nonsingular. Then there exists a constant $M_1 > 0$ such that the following inequality holds for all $k \in K$ sufficiently large:*

$$\|d_k\| \leq M_1\|F(x_k)\|. \tag{44}$$

Proof. Without loss of generality, we suppose $\{x_k\}_K \to x^*$. Since $\sum_{k=1}^{\infty}\|s_k\|^2 < \infty$, it is clear that when $k \in K$ is sufficiently large, A_{k+1} is nonsingular. Moreover, there is a constant $C_1 > 0$ such that the inequality $\|A_{k+1}^{-1}\| \leq C_1$ holds for all $k \in K$ sufficiently large. It then follows from Lemma 8 that $\{\delta_k\}_K \to 0$. Therefore, there exists an index $N > 0$ such that the inequality $\delta_k \leq 1/(2C_1)$ holds for all $k \in K$ with $k \geq N$. Consequently, we get from the definition of δ_k that for all $k \in K$ with $k > N$ sufficiently large

$$\begin{aligned}\|d_k\| &\leq \left\|A_{k+1}^{-1}\right\|\left(\|A_{k+1}d_k + g(x_k)\| + \|g(x_k)\|\right)\\ &\leq C_1\left(\|g(x_k)\| + \delta_k\|d_k\|\right) \leq C_1\left(\|g(x_k)\| + \frac{1}{2C_1}\|d_k\|\right)\\ &\leq C_1\|g(x_k)\| + \frac{1}{2}\|d_k\|.\end{aligned} \tag{45}$$

The last inequality implies (44) with $M_1 = 2C_1$. □

We show the global convergence of Algorithm 3 in the following section.

Theorem 10. *Let Assumption 6 hold and let index set K be specified by Lemma 8. Then the sequence $\{x_k\}$ converges to the unique solution x^* of* (1).

Proof. We first verify

$$\lim_{k\to\infty}\inf\|F(x_k)\| = 0. \tag{46}$$

If there are infinitely many k for which α_k is determined by (21), then $\|F(x_{k+1})\| \leq \rho\|F(x_k)\|$ holds for infinitely many k. Let I be the index set for which (21) holds and let i_k be the number of index j, where $j \leq k$ and $j \in I$. Then we can know that $i_k \to \infty$, when $k \to \infty$. For each $k \notin I$, we have $\|F(x_{k+1})\| \leq (1+\epsilon_k)\|F(x_k)\|$, and then for all sufficiently large k we have

$$\|F(x_{k+1})\| \leq (\rho)^{i_{k+1}}\prod_{i=0}^{k}(1+\epsilon_i)\|F(x_0)\| \leq \rho^{i_{k+1}}e^{\epsilon}\|F(x_0)\|, \tag{47}$$

where $\rho \in (0,1)$. This implies $\lim\inf_{k\to\infty}\|F(x_k)\| = 0$, and hence the conclusion is true.

If there are finitely many k for which α_k is determined by (21), by Lemma 8, there exists a subsequence $\{\delta_k\}_{k\in K}$ that converges to zero. Similar to the proof of Lemma 9, it is not difficult to show that (44) holds for all $k \in \overline{K}$ sufficiently large, where $\overline{K}$ denotes the index set of $k > \hat{k}$, $k \in K$. In particular, the subsequence $\{d_k\}_{\overline{K}}$ is bounded. Without loss of generality, we suppose that $\{d_k\}_{\overline{K}}$ converges to some d^*.

Denote $\overline{\alpha} = \lim\sup_{k\to\infty}\alpha_k$. It is clear that $\overline{\alpha} \geq 0$. If $\overline{\alpha} > 0$, then $d^* = 0$, and hence it follows from (20) that $F(x^*) = 0$. Suppose $\overline{\alpha} = 0$, or equivalently $\lim_{k\to\infty}\alpha_k = 0$. By the line search rule, when $k \in \overline{K}$ is sufficiently large, $\alpha_k < 1$ and hence

$$\left\|F\left(x_k + \rho^{-1}\alpha_k d_k\right)\right\| - \|F(x_k)\| > -\sigma_1\left\|\rho^{-1}\alpha_k d_k\right\|^2. \tag{48}$$

Multiplying both sides by $(\|F(x_k + \rho^{-1}\alpha_k d_k)\| + \|F(x_k)\|)/(\rho^{-1}\alpha_k)$ and then taking limit as $k \to \infty$ with $k \in \overline{K}$, we obtain

$$F(x^*)^T F'(x^*)d^* \geq 0. \tag{49}$$

On the other hand, taking the limit in (20) as $k \to \infty$ with $k \in \overline{K}$ yields

$$F'(x^*)d^* + F(x^*) = 0. \tag{50}$$

This together with (49) implies $F(x^*) = 0$. □

In the latter part of this section, we give the superlinear convergence of Algorithm 3.

Theorem 11. *Let the conditions in Theorem 10 hold. Then there exist a constant $\delta > 0$ and an index $\overline{k}$ such that $\alpha_k = 1$ whenever $\delta_k \leq \delta$ and $k \geq \overline{k}$, the inequality*

$$\|F(x_k + d_k)\| \leq \rho\|F(x_k)\| - \sigma_1\|d_k\|^2 < \rho\|F(x_k)\| \tag{51}$$

holds for all $k \geq \overline{k}$ and $\delta_k \leq \delta$.

Proof. By Theorem 10, $\{x_k\}$ converges to the solution of (1), say, x^*, and there exists a constant $C_2 > 0$ such that $\|A_{k+1}^{-1}\| \le C_2$ for all k sufficiently large. Similar to the proof of Lemma 9, when k is large enough we can show that

$$\|d_k\| \le M_2 \|F(x_k)\|. \tag{52}$$

And from (20) we have

$$\begin{aligned} A_{k+1}(x_k + d_k - x^*) &= A_{k+1}(x_k - x^*) + (A_{k+1} - B_k) d_k - F(x_k) \\ &= \left(A_{k+1} - F'(x^*)\right)(x_k - x^*) + (A_{k+1} - B_k) d_k \\ &\quad - F(x_k) + F(x^*) + F'(x^*)(x_k - x^*), \end{aligned} \tag{53}$$

and this implies

$$\begin{aligned} \|x_k + d_k - x^*\| &\le \left\|A_{k+1}^{-1}\right\| \\ &\quad \times \Big(\left\|A_{k+1} - F'(x^*)\right\| \\ &\quad \times \|x_k - x^*\| + \|(A_{k+1} - B_k) d_k\| \\ &\quad + \big\|F(x_k) - F(x^*) \\ &\quad + F'(x^*)(x_k - x^*)\big\|\Big) \\ &\le C_2\left(o\|x_k - x^*\|^2 + \delta_k \|d_k\|\right) \\ &\le C_2\big(o\|x_k - x^*\| \\ &\quad + M_2\delta_k \|F(x_k) - F(x^*)\|\big) \\ &\le C_2\left(o\|x_k - x^*\| + M_2 M \delta_k \|x_k - x^*\|\right), \end{aligned} \tag{54}$$

where M is an upper bound of $F'(x)$ in Ω. The second and third inequalities follow from the definition of δ_k and (52), respectively. It then follows that

$$\begin{aligned} \|F(x_k + d_k)\| &= \|F(x_k + d_k) - F(x^*)\| \\ &\le M\|x_k + d_k - x^*\| \\ &\le MC_2\left(o\|x_k - x^*\| + M_2 M \delta_k \|x_k - x^*\|\right). \end{aligned} \tag{55}$$

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

$$\|F(x_k)\| = \|F(x_k) - F(x^*)\| \ge m\|x_k - x^*\| \tag{56}$$

holds for all k sufficiently large. Therefore, we deduce from (52) and (56) that when $\delta_k \le \delta'$

$$\begin{aligned} &\|F(x_{k+1})\| - \rho\|F(x_k)\| + \sigma_2\|d_k\|^2 \\ &\quad \le MC_2\left(o\|x_k - x^*\| + M_2 M \delta_k \|x_k - x^*\|\right) \\ &\qquad - \rho m\|x_k - x^*\| + \sigma_2 M_2^2 \|F(x_k)\|^2 \\ &\quad \le \left(M_2 C_2 M^2 \delta_k - \rho m\right)\|x_k - x^*\| \\ &\qquad + o\left(\|x_k - x^*\|\right) + \sigma_2 M_2^2 M^2 \|x_k - x^*\|^2 \\ &\quad \le -\left(\rho m - M_2 C_2 M^2 \delta_k\right)\|x_k - x^*\| \\ &\qquad + o\left(\|x_k - x^*\|\right). \end{aligned} \tag{57}$$

Let $\delta = \min\{\delta', (1/2)\rho m (M_2 C_2 M^2)^{-1}\}$. Then we know that when $\delta_k < \delta$, (51) holds for all k sufficient large. □

The following theorem establishes the superlinear convergence of Algorithm 3.

Theorem 12. *Let the conditions in Theorem 10 hold. Then the sequence $\{x_k\}$ generated by Algorithm 3 converges to the unique solution x^* of* (1) *superlinearly.*

Proof. By Theorem 11, it suffices to show $\{\delta_k\} \to 0$ as $k \to \infty$.

Let δ and $\bar{k}$ be as specified by Theorem 11. It follows from Lemma 8 that there is an index $\tilde{k}$ such that the following inequality holds for all $k \ge \tilde{k}$:

$$\frac{1}{k}\sum_{j=0}^{k-1}\delta_j^2 \le \frac{1}{2}\delta^2. \tag{58}$$

This shows that, for any $k \ge \tilde{k}$, there are at least $\lceil k/2 \rceil$ many indices $j \le k$ such that $\delta_j \le \delta$. Let $k' = \max\{\bar{k}, \tilde{k}\}$. By Theorem 11, for any $k \ge 2k'$, there are at least $\lceil k/2 \rceil - k'$ many indices $j \le k$ such that $\alpha_j = 1$ and

$$\left\|F(x_{j+1})\right\| = \left\|F(x_j + d_k)\right\| \le \rho\left\|F(x_j)\right\|. \tag{59}$$

Let J_k be the set of indices for which (21) holds and let j_k be the number of elements in J_k. Then $j_k \ge (k/2) - k' - 1$. On the other hand, for each $j \notin J_k$, we have

$$\left\|F(x_{j+1})\right\| \le (1 + \varepsilon_k)\left\|F(x_j)\right\|. \tag{60}$$

Multiplying inequalities (21) with $j \in J_k$ and (60) with $j \notin J_k$, we can obtain for any $k > 2k'$

$$\begin{aligned} \|F(x_{k+1})\| &\le (\rho)^{j_k}\|F(x_{k'})\|\left[\prod_{j=k'}^{k}(1 + \varepsilon_j)\right] \\ &\le \|F(x_{k'})\|\rho^{(1/2)k - k' - 1} e^{\varepsilon}, \end{aligned} \tag{61}$$

or equivalently

$$\left\|F\left(x_{k+1}\right)\right\| \leq \left\|F\left(x_{k'}\right)\right\| \rho^{(1/2)k-k'-1} e^{\varepsilon}. \tag{62}$$

So, we have

$$\sum_{k=0}^{\infty} \left\|F\left(x_k\right)\right\| < \infty. \tag{63}$$

This together with (56) implies

$$\sum_{k=0}^{\infty} \left\|s_k\right\| < \infty. \tag{64}$$

It then follows from Lemma 7 that $\{\delta_k\} \to 0$ as $k \to \infty$. The proof is completed. □

4. Numerical Experiments

In this section, we will present some numerical results to show the efficiency of Algorithm 3 for a class of sparse nonlinear equations.

In each experiment, we employ the following termination criterion:

$$\left\|F\left(x_k\right)\right\|_2 \leq 10^{-5}. \tag{65}$$

The parameters in Algorithm 3 are specified as follows:

$$\rho = 0.9, \qquad \sigma_1 = \sigma_2 = 0.001, \qquad \beta = 0.45, \qquad \eta_k = \frac{1}{(k+1)^2}, \tag{66}$$

see [13].

The numerical experiments are done by using MATLAB version 7.10 on a Core (TM) 2 PC with Windows XP. The details of the problems are given as follows, where x_0 denotes the initial point.

Problem 1 (Broyden tridiagonal function [15]). The elements of $F(x)$ are

$$\begin{aligned} F_1(x) &= -\left(3-0.5x_1\right)x_1 + 2x_2 - 1, \\ F_i(x) &= x_{i-1} - \left(3-0.5x_i\right)x_i + 2x_{i+1} - 1, \\ &\qquad i = 2,3,\ldots,n-1, \\ F_n(x) &= x_{n-1} - \left(3-0.5x_n\right)x_n - 1, \\ x_0 &= (-3,-3,\ldots,-3)^T. \end{aligned} \tag{67}$$

Problem 2 (Trigexp function [16]). The elements of $F(x)$ are

$$\begin{aligned} F_1(x) &= 3x_1^2 + 2x_2 - 5 + \sin\left(x_1 - x_2\right)\sin\left(x_1 + x_2\right), \\ F_i(x) &= 3x_i^2 + 2x_{i+1} - 5 + \sin\left(x_i - x_{i+1}\right) \\ &\quad \times \sin\left(x_i + x_{i+1}\right) + 4x_i \\ &\quad - x_{i-1}\exp\left(x_i - 1\right) - x_i - 3, \\ &\qquad i = 2,3,\ldots,n-1, \\ F_n(x) &= 4x_n - x_{n-1}\exp\left(x_{n-1} - x_n\right) - 3, \\ x_0 &= (-2,-2,\ldots,-2)^T. \end{aligned} \tag{68}$$

Problem 3 (tridiagonal exponential problem [17]). The elements of $F(x)$ are

$$\begin{aligned} F_1(x) &= x_1 - \exp\left(\cos\left(h\left(x_1 + x_2\right)\right)\right), \\ F_i(x) &= x_i - \exp\left(\cos\left(h\left(x_{i-1} + x_i + x_{i+1}\right)\right)\right), \\ &\qquad i = 2,3,\ldots,n-1, \\ F_n(x) &= x_n - \exp\left(\cos\left(h\left(x_{n-1} + x_n\right)\right)\right), \\ h &= \frac{1}{(n+1)}, \\ x_0 &= (1.5,1.5,\ldots,1.5)^T. \end{aligned} \tag{69}$$

Problem 4 (discrete boundary value problem [18]). The elements of $F(x)$ are

$$\begin{aligned} F_1(x) &= 2x_1 + 0.5h^2\left(x_1 + h\right)^3 - x_2, \\ F_i(x) &= 2x_i + 0.5h^2\left(x_i + hi\right)^3 - x_{i-1} + x_{i+1}, \\ &\qquad i = 2,3,\ldots,n-1, \\ F_n(x) &= 2x_n + 0.5h^2\left(x_n + hn\right)^3 - x_{n-1}, \\ h &= \frac{1}{n+1}, \\ x_0 &= \left(h(h-1), h(2h-1), \ldots, h(nh-1)\right)^T. \end{aligned} \tag{70}$$

Problem 5 (exponentional problem 1 [19]). The elements of $F(x)$ are

$$\begin{aligned} F_1(x) &= \exp\left(x_1 - 1\right) - 1, \\ F_i(x) &= i\left(\exp\left(x_i - 1\right) - x_i\right), \quad i = 2,3,\ldots,n, \\ x_0 &= \left(\frac{n}{n-1},\ldots,\frac{n}{n-1}\right)^T. \end{aligned} \tag{71}$$

Table 1: Results of Algorithm 3 for Problems 1–7.

Pro	n	$\|F(x_0)\|$	$B_0 = I$			$B_0 = F'(x_0)$		
			$\|F(x_k)\|$	Iter	Time (s)	$\|F(x_k)\|$	Iter	Time (s)
1	50	26.8421	$4.2686e-06$	17	0.0000	$3.0281e-06$	12	0.0000
	100	36.5103	$6.6863e-06$	17	0.0000	$3.0284e-06$	12	0.0000
	200	50.5767	$3.4126e-06$	19	0.0313	$3.0284e-06$	12	0.0313
	500	78.9494	$3.5829e-06$	19	0.0313	$3.0284e-06$	12	0.0313
	1000	111.1665	$6.7616e-06$	20	0.0938	$3.0284e-06$	12	0.0625
	3000	156.8247	$4.6028e-06$	21	0.1780	$3.0284e-06$	12	0.1563
	5000	247.7055	$5.0272e-06$	22	0.2063	$3.0284e-06$	12	0.2031
	10000	256.8910	$5.6234e-06$	25	0.2028	$3.0284e-06$	12	0.3125
	20000	289.2351	$5.4512e-06$	25	0.3276	$3.0284e-06$	12	0.6563
2	50	6.9282	$1.3431e-06$	10	0.0000	$1.0235e-06$	4	0.0000
	100	9.8995	$1.8437e-06$	11	0.0000	$1.0235e-06$	4	0.0000
	200	14.0712	$6.2185e-06$	11	0.0313	$1.0235e-06$	4	0.0000
	500	22.3159	$2.6805e-06$	12	0.0313	$1.0235e-06$	4	0.0000
	1000	31.5911	$7.8148e-06$	11	0.0625	$1.0235e-06$	4	0.0313
	3000	54.7540	$8.5561e-06$	18	0.7188	$1.0235e-06$	4	0.0625
	5000	70.6965	$8.3005e-06$	12	0.2500	$1.0235e-06$	4	0.0938
	10000	75.0126	$8.2145e-06$	14	0.2808	$1.0235e-06$	4	0.1406
	20000	81.2659	$8.5179e-06$	14	0.5304	$1.0235e-06$	4	0.2500
3	50	8.5416	$8.7650e-08$	3	0.0000	$1.1795e-07$	5	0.0000
	100	12.1562	$6.8406e-10$	3	0.0000	$5.3904e-06$	4	0.0000
	200	17.2195	$8.0105e-06$	2	0.0000	$3.6741e-07$	3	0.0000
	500	27.2392	$3.3130e-07$	2	0.0000	$1.2337e-06$	3	0.0000
	1000	38.5246	$2.9461e-08$	2	0.0000	$1.5498e-07$	3	0.0313
	3000	66.7279	$6.3233e-10$	3	0.0313	$3.0014e-06$	2	0.0313
	5000	86.1455	$1.0586e-10$	2	0.0313	$1.0804e-06$	2	0.0625
	10000	121.8282	$9.3691e-12$	2	0.0938	$2.7008e-07$	2	0.0938
	20000	172.2911	$8.7919e-13$	2	0.1563	$6.7517e-08$	2	0.1719
4	50	0.1511	$8.7479e-06$	19	0.0000	$2.4065e-06$	13	0.0000
	100	0.1111	$8.4206e-06$	16	0.0000	$5.7592e-06$	12	0.0000
	200	0.0801	$9.4752e-06$	18	0.0313	$9.6492e-06$	10	0.0313
	500	0.0512	$8.2454e-06$	14	0.0313	$9.0492e-06$	10	0.0313
	1000	0.0364	$7.7380e-06$	13	0.0625	$4.0496e-06$	10	0.0625
	3000	0.0211	$7.6951e-06$	13	0.1719	$9.7135e-06$	8	0.1250
	5000	0.0163	$9.8163e-06$	10	0.1875	$8.4494e-06$	7	0.1563
	10000	0.0115	$8.4212e-06$	10	0.3125	$4.1718e-06$	7	0.2344
	20000	0.0082	$6.9072e-06$	4	0.2656	$2.0897e-06$	7	0.3906
5	50	0.0481	$7.6492e-06$	9	0.0000	$5.5212e-06$	9	0.0000
	100	0.0315	$5.3122e-06$	9	0.0000	$9.8754e-06$	8	0.0000
	200	0.0213	$9.7441e-06$	8	0.0000	$6.8642e-06$	8	0.0000
	500	0.0131	$6.1289e-06$	8	0.0000	$4.2970e-06$	8	0.0000
	1000	0.0092	$4.3258e-06$	8	0.0000	$7.9238e-06$	7	0.0000
	3000	0.0053	$6.5316e-06$	7	0.0000	$4.5644e-06$	7	0.0000
	5000	0.0041	$5.0582e-06$	7	0.0000	$9.2636e-06$	6	0.0000
	10000	0.0029	$9.3577e-06$	6	0.0000	$6.5482e-06$	6	0.0313
	20000	0.0020	$6.6163e-06$	6	0.0313	$4.6295e-06$	6	0.0313

TABLE 1: Continued.

Pro	n	$\|F(x_0)\|$	$B_0 = I$ $\|F(x_k)\|$	Iter	Time (s)	$B_0 = F'(x_0)$ $\|F(x_k)\|$	Iter	Time (s)
6	50	0.0166	$6.6166e-06$	7	0.0000	$6.5364e-06$	5	0.0000
	100	0.0116	$7.5062e-06$	6	0.0000	$8.5483e-06$	2	0.0000
	200	0.0082	$9.4436e-06$	4	0.0000	$2.1372e-06$	2	0.0000
	500	0.0052	$9.8417e-06$	18	0.0406	$5.2204e-06$	1	0.0000
	1000	0.0037	$9.6163e-06$	24	0.0313	$3.6586e-06$	3	0.0313
	3000	0.0021	$8.3375e-06$	17	0.0625	$2.1088e-06$	3	0.0313
	5000	0.0016	$9.7458e-06$	15	0.0938	$1.6333e-06$	3	0.0313
	10000	0.0012	$9.9565e-06$	11	0.1563	$1.1548e-06$	3	0.0313
	20000	0.0082	$8.8706e-06$	9	0.2031	$8.1653e-06$	2	0.0625
7	50	0.2271	$1.4914e-14$	2	0.0000	$5.9494e-06$	53	0.1250
	100	0.2319	$1.4792e-14$	2	0.0000	$5.6147e-06$	58	0.0938
	200	0.2413	$1.4960e-14$	2	0.0000	$6.6486e-06$	77	0.2031
	500	0.2675	$1.5760e-14$	2	0.0000	$7.4199e-06$	70	0.7500
	1000	0.3062	0	2	0.0313	$8.2082e-06$	35	0.3906
	3000	0.4274	0	4	0.0625	$5.7188e-06$	53	10.5469
	5000	0.5211	0	7	2.5000	$6.7092e-06$	41	19.0625
	10000	0.7027	0	9	10.3125	$6.8420e-06$	45	25.0156
	20000	0.9686	0	11	43.7500	$7.1086e-06$	51	30.0156

Problem 6 (exponentional problem 2 [19]). The elements of $F(x)$ are

$$F_1(x) = \exp(x_1) - 1,$$
$$F_i(x) = \frac{i}{10}\left(\exp(x_i) + x_{i-1} - 1\right), \quad i = 2, 3, \ldots, n, \tag{72}$$
$$x_0 = \left(\frac{1}{n^2}, \ldots, \frac{1}{n^2}\right)^T.$$

Problem 7 (penalty I function [19]). The elements of $F(x)$ are

$$F_i(x) = \sqrt{10^{-5}}\left(x_i - 1\right), \quad i = 1, \ldots, n-1,$$
$$F_n(x) = \frac{1}{4n}\sum_{j=1}^{n} x_j^2 - \frac{1}{4}, \tag{73}$$
$$x_0 = \left(\frac{1}{3}, \ldots, \frac{1}{3}\right)^T.$$

Problem 8 (exponential function [19]). The elements of $F(x)$ are

$$F_i(x) = \frac{i}{10}\left(1 - x_i^2 - \exp\left(-x_i^2\right)\right), \quad i = 1, 2, \ldots, n-1,$$
$$F_n(x) = \frac{n}{10}\left(1 - \exp\left(-x_n^2\right)\right),$$
$$x_0 = \left(\frac{1}{4n^2}, \frac{2}{4n^2}, \ldots, \frac{n}{4n^2}\right)^T. \tag{74}$$

Problem 9 (minimal function [19]). The elements of $F(x)$ are

$$F_i(x) = \frac{\left(\ln x_i - \exp(x_i)\right) - \sqrt{\left(\ln x_i - \exp(x_i)\right)^2 + 10^{-10}}}{2},$$
$$i = 1, \ldots, n,$$
$$x_0 = (1, 1, \ldots, 1)^T. \tag{75}$$

Problem 10 (extended Rosenbrock function (n is even) [20]). The elements of $F(x)$ are

$$F_{2i-1}(x) = 10\left(x_{2i} - x_{2i-1}^2\right),$$
$$F_{2i}(x) = 1 - x_{2i-1}, \quad i = 1, 2, \ldots, \frac{n}{2}, \tag{76}$$
$$x_0 = (5, 1, 5, 1, \ldots, 5, 1)^T.$$

Problem 11 (logarithmic function [19]). The elements of $F(x)$ are

$$F_i(x) = \ln\left(x_i + 1\right) - \frac{x_i}{n}, \quad i = 1, 2, \ldots, n, \tag{77}$$
$$x_0 = (1, 1, \ldots, 1)^T.$$

Problem 12 (strictly convex function 1 [21]). $F(x)$ is the gradient of $f(x) = \sum_{i=1}^{n}(\exp(x_1) - x_i)$. The elements of $F(x)$ are

$$F_i(x) = \exp(x_i) - 1, \quad i = 1, 2, \ldots, n.$$
$$x_0 = \left(\frac{1}{n}, \frac{2}{n}, \ldots, 1\right)^T. \tag{78}$$

Table 2: Results of Algorithm 3 for Problems 8–14.

Pro	n	$\|F(x_0)\|$	$B_0 = I$			$B_0 = F'(x_0)$		
			$\|F(x_k)\|$	Iter	Time (s)	$\|F(x_k)\|$	Iter	Time (s)
8	52	1.2019e − 04	4.7426e − 06	4	0.0000	4.8145e − 06	6	0.0000
	100	6.2500e − 05	6.8402e − 06	3	0.0000	6.9537e − 06	4	0.0000
	200	3.1250e − 05	7.7243e − 06	2	0.0000	7.8200e − 06	2	0.0000
	500	1.2500e − 05	9.8751e − 06	11	0.0000	3.1279e − 06	2	0.0000
	1000	6.2500e − 06	6.2500e − 06	0	0.0000	6.2500e − 06	0	0.0000
	3000	2.0833e − 06	2.0833e − 06	0	0.0000	2.0833e − 06	0	0.0000
	5000	1.2500e − 06	1.2500e − 06	0	0.0000	1.2500e − 06	0	0.0000
	10000	6.2500e − 07	6.2500e − 07	0	0.0000	6.2500e − 07	0	0.0000
	20000	3.1250e − 07	3.1250e − 07	0	0.0000	3.1250e − 07	0	0.0000
9	50	19.2212	4.7930e − 11	4	0.0000	4.3678e − 11	2	0.0000
	100	27.1828	6.7783e − 11	4	0.0000	6.1770e − 11	2	0.0000
	200	38.4423	9.5860e − 11	4	0.0000	9.8689e − 11	2	0.0000
	500	60.7826	1.5157e − 10	4	0.0000	1.5604e − 10	2	0.0000
	1000	85.9596	2.1435e − 10	4	0.0313	2.2068e − 10	2	0.0000
	3000	148.8864	4.1536e − 10	4	0.0313	3.8222e − 10	2	0.0313
	5000	192.2116	5.3623e − 10	4	0.0625	4.9345e − 10	2	0.0625
	10000	271.8282	7.5834e − 10	4	0.1250	7.7298e − 10	2	0.0938
	20000	384.4231	1.0725e − 09	4	0.2500	1.0932e − 09	2	0.1563
10	50	1200	8.3426e − 06	33	0.0000	4.2841e − 09	19	0.0000
	100	1697.1	2.2053e − 07	34	0.0000	6.0586e − 09	19	0.0000
	200	2400	2.4538e − 07	34	0.0000	3.5281e − 06	23	0.0000
	500	3794.7	1.3524e − 06	35	0.0313	5.5783e − 06	23	0.0000
	1000	5366.6	1.0785e − 06	33	0.0313	7.8890e − 06	23	0.0000
	3000	9295.2	2.3859e − 07	33	0.0625	1.1640e − 07	24	0.0313
	5000	12000	3.0801e − 07	33	0.0938	1.5028e − 07	24	0.0625
	10000	16971	4.1301e − 06	29	0.1563	2.1252e − 07	24	0.1250
	20000	24000	5.3904e − 06	25	0.2813	1.4378e − 07	24	0.2500
11	50	4.7599	1.2153e − 07	6	0.0000	1.4006e − 06	6	0.0000
	100	6.8315	1.1647e − 07	6	0.0000	1.4743e − 06	6	0.0000
	200	9.7319	1.3534e − 07	6	0.0000	1.8019e − 06	6	0.0000
	500	15.4545	1.9008e − 07	6	0.0000	2.6117e − 06	6	0.0000
	1000	21.8876	2.5837e − 07	6	0.0000	3.5883e − 06	6	0.0000
	3000	37.9470	4.3584e − 07	6	0.0000	6.0967e − 06	6	0.0000
	5000	48.9988	5.5970e − 07	6	0.0000	2.6853e − 06	6	0.0000
	10000	69.3047	7.8840e − 07	6	0.0313	3.7947e − 06	6	0.0313
	20000	98.0187	1.1128e − 06	6	0.0313	5.3645e − 06	6	0.0469
12	50	6.2761	1.5871e − 08	7	0.0000	2.7576e − 08	6	0.0000
	100	8.7909	1.9036e − 08	7	0.0000	3.6023e − 08	6	0.0000
	200	12.3723	2.4543e − 08	7	0.0000	4.8866e − 08	6	0.0000
	500	19.5054	3.6586e − 08	7	0.0000	7.5309e − 08	6	0.0000
	1000	27.5580	5.0705e − 08	7	0.0000	1.0559e − 07	6	0.0000
	3000	47.7008	8.6635e − 08	7	0.0000	1.8182e − 07	6	0.0000
	5000	61.5735	1.1154e − 07	7	0.0000	2.3446e − 07	6	0.0000
	10000	87.0696	1.5742e − 07	7	0.0000	3.3129e − 07	6	0.0000
	20000	123.1291	2.2239e − 07	7	0.0313	4.6831e − 07	6	0.0313

Table 2: Continued.

Pro	n	$\|F(x_0)\|$	$B_0 = I$			$B_0 = F'(x_0)$		
			$\|F(x_k)\|$	Iter	Time (s)	$\|F(x_k)\|$	Iter	Time (s)
13	51	771.2349	$8.4607e-09$	10	0.0000	$1.6413e-08$	9	0.0000
	99	1089.8	$1.1644e-08$	10	0.0000	$2.3210e-08$	9	0.0000
	201	1564.5	$1.6479e-08$	10	0.0000	$3.3331e-08$	9	0.0000
	501	2480.5	$2.5914e-08$	10	0.0000	$5.2860e-08$	9	0.0000
	999	3507.7	$3.6544e-08$	10	0.0000	$7.4754e-08$	9	0.0000
	3000	6087.3	$6.3242e-08$	10	0.0313	$1.2974e-07$	9	0.0000
	5001	7857.2	$8.1675e-08$	10	0.0625	$1.6746e-07$	9	0.0313
	9999	11112	$1.1547e-07$	10	0.0938	$2.3682e-07$	9	0.0625
	20001	15717	$1.6330e-07$	10	0.1563	$3.3496e-07$	9	0.1563
14	50	147.1394	$2.8039e-06$	7	0.0000	$1.4943e-06$	8	0.0000
	100	208.0865	$3.9654e-06$	7	0.0000	$2.1133e-06$	8	0.0000
	200	294.2788	$5.6079e-06$	7	0.0000	$2.9887e-06$	8	0.0313
	500	465.2956	$8.8669e-06$	7	0.0313	$4.7255e-06$	8	0.0625
	1000	658.0274	$3.5104e-07$	8	0.0625	$6.6829e-06$	8	0.1250
	3000	1139.7	$6.0803e-07$	8	0.1250	$3.1348e-08$	9	0.3438
	5000	1471.4	$7.8496e-07$	8	0.2969	$4.0470e-08$	9	0.4219
	10000	2080.9	$1.1101e-06$	8	0.3594	$5.7234e-08$	9	0.8438
	20000	2942.8	$1.5699e-06$	8	0.7656	$8.0941e-08$	9	1.5000

Problem 13 (tridimensional valley function (n is a multiple of 3) [22]). The elements of $F(x)$ are

$$\begin{aligned} F_{3i-2}(x) &= \left(c_2 x_{3i-2}^3 + c_1 x_{3i-2}\right)\exp\left(\frac{-x_{3i-2}^2}{100}\right) - 1, \\ F_{3i-1}(x) &= 10\left(\sin\left(x_{3i-2} - x_{3i-1}\right)\right), \\ F_{3i}(x) &= 10\left(\cos\left(x_{3i-2} - x_{3i}\right)\right), \quad i = 1, 2, \ldots, \frac{n}{3}, \end{aligned} \tag{79}$$

and then denote

$$\begin{aligned} c_1 &= 1.003344481605351, \\ c_2 &= -3.344481605351171 \times 10^{-3}, \\ x_0 &= (2, 1, 2, 1, 2, \ldots, 1)^T. \end{aligned} \tag{80}$$

Problem 14 (extended Freudenstein and Roth function (n is even) [17]). The elements of $F(x)$ are

$$\begin{aligned} F_{2i-1}(x) &= x_{2i-1} + \left(\left(5 - x_{2i}\right)x_{2i} - 2\right)x_{2i} - 13, \\ F_{2i}(x) &= x_{2i-1} + \left(\left(x_{2i} + 1\right)x_{2i} - 14\right)x_{2i} - 29, \\ & \qquad i = 1, 2, \ldots, \frac{n}{2}, \\ x_0 &= (6, 3, 6, 3, \ldots, 6, 3)^T. \end{aligned} \tag{81}$$

The sparsity patterns of most of the problems are tridiagonal and the dimension of problems varies from 50 to 20000. The results are given in Tables 1 and 2, and each column is specified as follows:

Pro: the problem;

n: the dimension of the problem;

$\|F(x_0)\|$: the initial Euclidean norms of $F(x)$;

$\|F(x_k)\|$: the final Euclidean norms of $F(x)$;

Iter: the total number of iterations;

Times: the CPU time in second.

It can be seen from the tables that, for all tested problems, Algorithm 3 terminated successfully. The numerical results show that Algorithm 3 becomes increasingly desirable as n increases.

In Tables 1 and 2, we list the results of Algorithm 3 for solving Problems 1 to 14 with $B_0 = I$. Because B_0 is very important for the performance of Broyden's method, we also present the results with $B_0 = F'(x_0)$. We can see that Algorithm 3 can be applied to solve a class of nonlinear equations, where the dimension of which can be up to 20000. Since the Schubert's update formula (7) can maintain the sparsity pattern of Jacobian matrix exactly, so Algorithm 3 is especially effective for solving large-scale nonlinear equations with sparse Jacobian matrix, such as tridiagonal or block diagonal Jacobian matrix.

5. Remarks

In this paper, based on the work of Schubert, Broyden, and Marwil, we have globalized Schubert's method and proposed a global algorithm by using a nonmonotone line search. We have established the global and superlinear convergence. Numerical results showed that the algorithm is especially effective for large-scale problems.

Conflict of Interests

The author declares that there is no conflict of interests regarding to the publication of this paper.

Acknowledgment

The work is supported by the National Science Foundation of China through Grant 11371154.

References

[1] Y. Yuan, "Recent advances in numerical methods for nonlinear equations and nonlinear least squares," *Numerical Algebra, Control and Optimization*, vol. 1, no. 1, pp. 15–34, 2011.

[2] L. K. Schubert, "Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian," *Mathematics of Computation*, vol. 24, pp. 27–30, 1970.

[3] E. Marwil, "Convergence results for Schubert's method for solving sparse nonlinear equations," *SIAM Journal on Numerical Analysis*, vol. 16, no. 4, pp. 588–604, 1979.

[4] C. G. Broyden, "The convergence of an algorithm for solving sparse nonlinear systems," *Mathematics of Computation*, vol. 25, no. 114, pp. 285–294, 1971.

[5] B. Lam, "On the convergence of a quasi-Newton method for sparse nonlinear systems," *Mathematics of Computation*, vol. 32, no. 142, pp. 447–451, 1978.

[6] P. L. Toint, "On sparse and symmetric matrix updating subject to a linear equation," *Mathematics of Computation*, vol. 31, no. no 140, pp. 954–961, 1977.

[7] D. F. Shanno, "On variable-metric methods for sparse Hessians," *Mathematics of Computation*, vol. 34, no. 150, pp. 499–514, 1980.

[8] R. Fletcher, "An optimal positive definite update for sparse Hessian matrices," *SIAM Journal on Optimization*, vol. 5, no. 1, pp. 192–218, 1995.

[9] N. Yamashita, "Sparse quasi-Newton updates with positive definite matrix completion," *Mathematical Programming*, vol. 115, no. 1, pp. 1–30, 2008.

[10] Y. H. Dai and N. Yamashita, "Analysis of sparse quasi-Newton updates with positive definite matrix completion," *Journal of the Operations Research Society of China*, vol. 2, no. 1, pp. 39–56, 2014.

[11] P. L. Toint, "Global convergence of the partitioned BFGS algorithm for convex partially separable optimization," *Mathematical Programming*, vol. 36, no. 3, pp. 290–306, 1986.

[12] A. Griewank, "The global convergence of partitioned BFGS on problems with convex decompositions and Lipschitzian gradients," *Mathematical Programming*, vol. 50, no. 1–3, pp. 141–175, 1991.

[13] D. Li and M. Fukushima, "A derivative-free line search and global convergence of Broyden-like method for nonlinear equations," *Optimization Methods and Software*, vol. 13, no. 3, pp. 181–201, 2000.

[14] A. Griewank, "The "global" convergence of Broyden-like methods with a suitable line search," *Australian Mathematical Society. Journal. Series B. Applied Mathematics*, vol. 28, no. 1, pp. 75–92, 1986.

[15] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Mathematics of Computation*, vol. 19, no. 92, pp. 577–593, 1965.

[16] M. A. Gomes-Ruggiero, J. M. Martínez, and A. C. Moretti, "Comparing algorithms for solving sparse nonlinear systems of equations," *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 2, pp. 459–483, 1992.

[17] B. Yang and L. Gao, "An efficient implementation of Merrill's method for sparse or partially separable systems of nonlinear equations," *SIAM Journal on Optimization*, vol. 1, no. 2, pp. 206–221, 1991.

[18] J. J. Moré, B. S. Garbow, and K. E. Hillström, "Testing unconstrained optimization software," *ACM Transactions on Mathematical Software (TOMS)*, vol. 7, no. 1, pp. 17–41, 1981.

[19] W. La Cruz, J. M. Martnez, and M. Raydan, "Spectral residual method without gradient information for solving large-scale nonlinear systems of equations," *Mathematics of Computation*, vol. 75, no. 255, pp. 1429–1448, 2006.

[20] M. G. Gasparo, "A nonmonotone hybrid method for nonlinear systems," *Optimization Methods and Software*, vol. 13, no. 2, pp. 79–94, 2000.

[21] M. Raydan, "The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem," *SIAM Journal on Optimization*, vol. 7, no. 1, pp. 26–33, 1997.

[22] A. Friedlander, M. A. Gomes-Ruggiero, D. N. Kozakevich, J. M. Martínez, and S. A. Santosa, "Solving nonlinear systems of equations by means of quasi-newton methods with a nonmonotone strategy," *Optimization Methods and Software*, vol. 8, no. 1, pp. 25–51, 1997.