THE SECANT/FINITE DIFFERENCE ALGORITHM FOR SOLVING SPARSE NONLINEAR SYSTEMS OF EQUATIONS*

GUANGYE LI†

Abstract. This paper presents an algorithm, the secant/finite difference algorithm, for solving sparse nonlinear systems of equations. This algorithm is a combination of a finite difference method and a secant method. A $q$-superlinear convergence result and an $r$-convergence rate estimate show that this algorithm has good local convergence properties. The numerical results indicate that this algorithm is competitive with some currently used algorithms.

Key words. algorithm, finite difference, Jacobian, nonlinear system of equations, numerical example, superlinear convergence, secant method, sparsity

AMS(MOS) subject classification. 65H10

1. Introduction. Consider the nonlinear system of equations

\[ F(x) = 0, \]

where $F: R^n \to R^n$ is continuously differentiable on an open convex set $D \subset R^n$, and the Jacobian matrix $F'(x)$ is sparse. To solve the system, we consider the iteration

\[ \bar{x} = x - B^{-1}F(x), \]

where $x$ is the current iterate, $\bar{x}$ is the new iterate and $B$ is an approximation to the Jacobian $F'(x)$ with the same sparsity as the Jacobian. After we finish this step, we have the information: $x$, $\bar{x}$, and $B$. Our purpose is to get a $\bar{B}$, a good approximation to $F'(\bar{x})$, by using as little effort as possible.

Schubert [13] gave a secant method for solving sparse nonlinear systems of equations which is a sparse modification of Broyden’s [1] update. It is called the sparse Broyden algorithm. Broyden [2] also gave this algorithm independently. The most attractive advantage of the sparse Broyden algorithm is that it uses only one function value at each iteration and that it is locally $q$-superlinearly convergent (see Marwil [8]).

Curtis, Powell, and Reid [4] also gave an algorithm for sparse problems called the CPR algorithm, which is a finite difference algorithm, but which can take the advantage of the sparsity to make the number of the function evaluations smaller than general finite difference algorithms. To describe the CPR algorithm, Coleman and Moré [3] gave some definitions concerning a partition of the columns of a matrix $B$.

**Definition 1.1.** A partition of the columns of $B$ is a division of the columns into groups $c_1, c_2, \cdots, c_p$ such that each column belongs to one and only one group.

**Definition 1.2.** A partition of the columns such that columns in a given group do not have a nonzero element in the same row position is consistent with the direct determination of $B$.

In order to have a good partition of the columns of $B$, Coleman and Moré [3] associated the partition problem with a certain graph coloring problem and gave some

---

* Received by the editors October 21, 1986; accepted for publication (in revised form) July 28, 1987.
† Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1. Permanent address, Computer Center, Jilin University, Changchun, Jilin, People’s Republic of China. This research was partially supported by Jilin University, People’s Republic of China, by Department of Energy grant DE-AS05-82ER1-13016, and Air Force Office of Scientific Research grant 85-0243. This work was done while the author was a visiting scholar in the Mathematical Sciences Department, Rice University, Houston, Texas 77251.
partitioning algorithms which can make $p$ optimal or nearly optimal. For convenience, we call the CPR algorithm based on Coleman and Moré’s algorithms the CPR-CM algorithm.

The CPR algorithm can be formulated as follows: For a given consistent partition of the columns of $B$, obtain vectors $d_1, d_2, \ldots, d_p$ such that $\bar{B}$ is determined uniquely by the equations

$$\bar{B}d_i = F(\bar{x} + d_i) - F(\bar{x}) = y_i, \quad i = 1, \ldots, p.$$  

Notice that at each iterative step we need only to compute $p + 1$ function values rather than the $n + 1$ values required by a straightforward column-by-column finite-difference algorithm.

As an example we consider the following matrix with a tridiagonal structure:

$$
\begin{bmatrix}
X & X & 0 & 0 & 0 \\
X & X & X & 0 & 0 \\
0 & X & X & X & 0 \\
0 & 0 & X & X & X \\
0 & 0 & 0 & X & X
\end{bmatrix}
$$

(1.3)

A consistent partition of the columns of the matrix is $c_1 = \{1, 4\}$, $c_2 = \{2, 5\}$, and $c_3 = \{3, 6\}$. If we take

$$
d_1 = (h, 0, 0, h, 0, 0)^T, \\
d_2 = (0, h, 0, 0, h, 0)^T, \\
d_3 = (0, 0, h, 0, 0, h)^T,
$$

then $\bar{B}$ is determined uniquely, and the number of function evaluations required at each iteration is 4.

In this paper, we propose an algorithm called the secant/finite difference (SFD) algorithm for solving sparse nonlinear systems of equations. This algorithm is also based on a consistent partition of the columns of the Jacobian. However, it uses the information we already have at every iterative step more efficiently than the CPR algorithm. This algorithm can be seen as a combination of the CPR-CM algorithm and a secant algorithm. The SFD algorithm reduces the number of function evaluations required by the CPR-CM algorithm by one, and it has good local convergence properties. Our numerical results show that the SFD algorithm is competitive with the CPR-CM algorithm and the sparse Broyden algorithm for some problems. The SFD algorithm and some of its properties are given in § 2. A Kantorovich-type analysis for the SFD algorithm is given in § 3. A $q$-superlinear convergence result and an $r$-convergence order estimate of the SFD algorithm are given in § 4. Some numerical results are given in § 5.

In this paper, $\| \cdot \|_F$ indicates the Frobenius norm of a matrix, and $\| \cdot \|$ indicates the $l_2$-vector norm. We use \"\" to denote the subtraction of two sets; that is,

$$A \backslash B = \{ v : v \in A \text{ and } v \notin B \}.$$  

For a sparse matrix $B$, we use $M_B$ to denote the set of pairs of indices $(i, j)$, where $b_{ij}$ is a structurally nonzero element of $B$, i.e.,

$$M_B = \{(i, j) : b_{ij} \neq 0 \}.$$
Since $M_{F(x)}$ is a constant set for a given $F$, we ignore the subscript throughout the paper. Moreover, we use $S(y, \delta)$ to denote the set $\{x \in \mathbb{R}^n : \|x - y\| < \delta, y \in \mathbb{R}^n\}$ and use $\overline{S}(y, \delta)$ to denote the closure of $S(y, \delta)$.

2. The SFD algorithm and its properties. Given a consistent partition of the columns of the Jacobian, which divides the set $\{1, \cdots, n\}$ into $p$ subsets $c_1, \cdots, c_p$ (for convenience, $c_i$, $i = 1, 2, \cdots, p$, indicates both the sets of the columns and the sets of the indices of these columns), also given $x, \bar{x} \in \mathbb{R}^n$, let

$$d_i = \sum_{j \in c_i} s_j e_j, \quad i = 1, \cdots, p,$$

where $e_j$ is the $j$th column of the unit matrix and let

$$g_i = \sum_{j=1}^i d_j, \quad i = 1, \cdots, p, \quad g_0 = 0,$$

where $s = \bar{x} - x$, and $s_j$ is the $j$th component of $s$. If $s_j \neq 0, j = 1, \cdots, n$, then $\bar{B}$ is determined uniquely by the equations

$$\bar{B}d_i = \bar{B}(\bar{x} - (\bar{x} - g_i)) = F(\bar{x}) - F(\bar{x} - g_i) \equiv y_i,$$

$$\vdots$$

$$\bar{B}d_p = \bar{B}(\bar{x} - g_{p-1} - x) = F(\bar{x} - g_{p-1}) - F(\bar{x} - g_p) \equiv y_p.$$

Since

$$F(\bar{x} - g_p) = F(x),$$

the last equation in (2.1) can be written as

$$\bar{B}d_p = F(\bar{x} - g_{p-1}) - F(x) \equiv y_p.$$

Let $\bar{B} = [\bar{b}_{lm}]$. By (2.1), if $(l, m) \in M$, then

$$\bar{b}_{lm} = \frac{e_l^T y_m}{s_m},$$

where $m \in c_i, i = 1, 2, \cdots, p$.

Notice that by (2.1), to get $\bar{B}$, we need only to compute $p$ function values since we already have the value $F(x)$ at the current step. The number of function evaluations at each iteration is one less than the CPR-CM algorithm. For example (1.3), we take

$$d_1 = (s_1, 0, 0, s_4, 0, 0)^T,$$

$$d_2 = (0, s_2, 0, 0, s_5, 0)^T,$$

$$d_3 = (0, 0, s_3, 0, 0, s_6)^T,$$

and

$$g_1 = (s_1, 0, 0, s_4, 0, 0)^T,$$

$$g_2 = (s_1, s_2, 0, s_4, s_5, 0)^T.$$
Then, we need only to compute the values of \( F(\bar{x}), F(\bar{x} - g_1), \) and \( F(\bar{x} - g_2), \) and the number of function evaluations is three per iteration instead of the four required by the CPR-CM algorithm.

Let

\[
J_i = \int_0^1 F'(\bar{x} - g_i + t d_i) \, dt, \quad i = 1, \cdots, p.
\]

Then

\[
J_i d_i = y_i, \quad i = 1, \cdots, p.
\]

Let \( J_i = [J_{im}^i]. \) Since \( J_i \) has the same sparsity as the Jacobian, by (2.4), we have that if \( (l, m) \in M, \) then

\[
J_{im}^i = \frac{e_i^T y_i}{s_m},
\]

where \( m \in c_i, \ i = 1, 2, \cdots, p. \) Comparing (2.5) with (2.2), we have

\[
J_i e_j = \tilde{B} e_j,
\]

where \( j \in c_i, \ i = 1, \cdots, p. \) Therefore, \( \tilde{B} \) can be written as

\[
\tilde{B} = \sum_{i=1}^p \sum_{j \in c_i} J_i e_j e_j^T.
\]

To study the convergence properties of the SFD algorithm, sometimes we assume that \( F'(x) \) satisfies the following Lipschitz condition: There exist \( \alpha_i > 0, \ i = 1, \cdots, n \) such that

\[
\| (F'(x) - F'(y)) e_i \| \leq \alpha_i \| x - y \|, \quad i = 1, 2, \cdots, n, \ x, y \in D.
\]

Let \( \alpha = (\sum_{i=1}^n \alpha_i^2)^{1/2}, \) then

\[
F'(x) - F'(y) \|_F \leq \alpha \| x - y \|, \quad x, y \in D.
\]

Notice that in practice we do not check this condition and we only assume that \( F \) is continuously differentiable on \( D. \)

Now we have the following estimate for \( \tilde{B}. \)

**Theorem 2.1.** Suppose \( F'(x) \) satisfies Lipschitz condition (2.8), and \( \tilde{B} \) is determined by (2.1). If \( \bar{x} \in D, \ \bar{x} - g_i \in D, \ i = 1, \cdots, p, \) and \( s_i \neq 0, \ i = 1, 2, \cdots, n, \) then

\[
\| F'(\bar{x}) - \tilde{B} \|_F \leq \alpha \| \bar{x} - x \|.
\]

**Proof.** By (2.6) and (2.7),

\[
\| F'(\bar{x}) - \tilde{B} \|_F^2 = \sum_{i=1}^p \| (F'(\bar{x}) - \tilde{B}) e_i \|^2
\]

\[
= \sum_{i=1}^p \sum_{j \in c_i} \| (F'(\bar{x}) - \tilde{B}) e_j \|^2
\]

\[
= \sum_{i=1}^p \sum_{j \in c_i} \| (F'(\bar{x}) - J_i) e_j \|^2.
\]
Using (2.3) and Lipschitz condition (2.8), we obtain

\[
\sum_{j \in c_i} \| (F'((\tilde{x}) - J_i) e_j \|^2 = \sum_{j \in c_i} \left\| \left( F'(\tilde{x}) - \int_0^1 F'(\tilde{x} - g_i + t(g_i - g_{i-1})) \, dt \right) e_j \right\|^2 \\
\leq \sum_{j \in c_i} \left( \alpha_j \int_0^1 \| g_i - t(g_i - g_{i-1}) \| \, dt \right)^2 \\
\leq \sum_{j \in c_i} \alpha_j^2 \left( \int_0^1 (1 - t) \| g_i \| \, dt + \int_0^1 \| g_{i-1} \| \, dt \right)^2 \\
\leq \sum_{j \in c_i} \alpha_j^2 \left( \frac{1}{2} \| s \| + \frac{1}{2} \| s \| \right)^2 = \| s \|^2 \sum_{j \in c_i} \alpha_j^2.
\]

(2.11)

It follows from (2.10) and (2.11) that

\[
\| F'(\tilde{x}) - B \|^2 \leq \| s \|^2 \sum_{i=1}^{p} \sum_{j \in c_i} \alpha_j^2 = \alpha^2 \| s \|^2.
\]

(2.12)

Then (2.9) follows from (2.12).

In (2.1), to determine \( \bar{B} \) uniquely, we assume that \( s_j \neq 0 \), \( j = 1, \cdots, n \). However, sometimes it may happen that \( s_i = 0 \) for some \( 1 \leq i \leq n \). If this happens, then the \( i \)-th column of \( \bar{B} \) cannot be determined uniquely by (2.1). In this case, let

\[
\Omega_1 = \{ i \in \{1, 2, \cdots, n\} : s_i \neq 0 \},
\]

and let

\[
\Omega_2 = \{1, 2, \cdots, n\} \setminus \Omega_1.
\]

Now we deal with the general case in such a way that if \( j \in \Omega_1 \), then the \( j \)-th column of \( \bar{B} \) is determined uniquely by (2.1). If \( j \in \Omega_2 \), then we let the \( j \)-th column of \( \bar{B} \) be equal to the \( j \)-th column of \( B \) if the current step is not the first step \( (k = 0) \). For the first step we can choose \( x^{-1} \) such that \( \Omega_2 \) is empty. Now the SFD algorithm with a global strategy can be stated as follows.

**Algorithm 2.2.** Given a consistent partition of the columns of the Jacobian, which divides the set \( \{1, 2, \cdots, n\} \) into \( p \) subsets \( c_1, c_2, \cdots, c_p \), and given \( x^{-1}, x^0 \in \mathbb{R}^n \) such that \( s_i^{-1} = x_i^0 - x_i^{-1} \neq 0 \), \( i = 1, 2, \cdots, n \), at each step \( k \geq 0 \):

1. Set

\[
g_i^{k-1} = \sum_{l=1}^{i} \sum_{l \in c_i} s_i^{k-1} e_l, \quad i = 1, 2, \cdots, p-1, \quad g_0^{k-1} = 0,
\]

where \( s_i^{k-1} = x_i^k - x_i^{k-1} \).

2. Compute \( F(x^k - g_i^{k-1}) \), \( i = 0, 1, \cdots, p-1 \), and set

\[
y_i^{k-1} = F(x^k - g_i^{k-1}) - F(x^k - g_i^{k-1}), \quad i = 1, \cdots, p,
\]

where \( F(x^k - g_p^{k-1}) = F(x^{k-1}) \).

3. If \( (l, m) \in M \) and \( |s_m^{k-1}| \neq 0 \), then set

\[
b_{lm}^{k} = \frac{e_l^T y_i^{k-1}}{s_m^{k-1}},
\]

(2.13)
otherwise set
\[ b_{im}^k = b_{im}^{k-1}, \]
where \( m \in c_i, i = 1, 2, \ldots, p. \)

4. Solve \( B_k s^k = -F(x^k). \)
5. Choose \( x^{k+1} \) by \( x^{k+1} = x^k + s^k \), or by a global strategy.
6. Check for convergence.

Note that in practice, if \(|s_i^{k-1}|\) in step (3) is too close to zero the cancellation errors will become significant. Therefore, there should be a lower bound \( \theta_i^{k-1} > 0 \) for \(|s_i^{k-1}|\).

We suggest choosing
\[ \theta_i^{k-1} = \sqrt{\text{macheps}} \max \{ \text{typ} x_i, |x_i^k| \}, \]
where \text{macheps} is the machine precision and \text{typ} \( x_i \) is a typical value of \( x_i \) given by users. Also for the first step, one should choose \( |s_i^{-1}| \approx \theta_i^{-1}, i = 1, 2, \ldots, n \) instead of \( |s_i^{-1}| \neq 0, i = 1, \ldots, n. \) We suggest choosing
\[ x_i^{-1} = x_i^0 + \sqrt{\text{macheps}} x_i^0. \]

The SFD algorithm is also an update algorithm, and the update can be written as

\[ (2.14) \quad B = B + \sum_{j \in \Omega_2} e_j e_j^T + \sum_{i=1}^{p} \sum_{j \in c_i \cap \Omega_1} J_i e_j e_j^T. \]

The following result shows that the SFD algorithm is a secant algorithm.

**Lemma 2.3.** \( B \) satisfies the secant equations

\[ (2.15) \quad B d_i = y_i, \quad i = 1, \ldots, p, \]
and (2.15) implies

\[ B s = F(\bar{x}) - F(x) = y. \]

The proof of Lemma 2.3 is straightforward. Suppose that we have finished the \( k \)th step of the iteration. Then the information we have is \( x_k, F(x_k), B_k, \) and \( x^{k+1}. \) Let

\[ d_i^k = \sum_{j \in c_i} s_j^k e_j, \quad i = 1, \ldots, p, \]

\[ g_i^k = \sum_{j=1}^{i} d_j^k, \quad i = 1, \ldots, p, \quad g_0^k = 0, \]
and

\[ (2.16) \quad f_i^{k+1} = \int_0^1 F(x^{k+1} - g_i^k + t d_i^k) \, dt, \quad i = 1, \ldots, p. \]

By (2.6), if \( s_j^k \neq 0 \) then

\[ (2.17) \quad f_i^{k+1} e_j = B_{k+1} e_j, \]
where \( j \in c_i, i = 1, 2, \ldots, p. \)

**Theorem 2.4.** Assume that \( F' \) satisfies Lipschitz condition (2.8). Let \( \{ x^i \}_{j=0}^{k+1} \) and \( \{ B_j \}_{j=0}^{k+1} \) be generated by Algorithm 2.2. Suppose that \( \{ x^{i+1} - g_i^j, i = 0, 1, 2, \ldots, p \}_{j=0}^{k+1} \subset D. \) If \( s_j^0 = 0, j = 0, 1, \ldots, k \) appears consecutively in at most \( m \) steps for any specific \( 1 \leq i \leq n, \) then for \( k \geq m - 1, \)

\[ (2.18) \quad \| F'(x^{k+1}) - B_{k+1} \|_F \leq \alpha \sum_{j=k-m}^{k} \| x^{j+1} - x^j \|. \]
Proof. By the hypothesis of the theorem, given $k$, for any $1 \leq i \leq n$, there exists at least one integer $0 \leq j \leq m$ such that $s_i^k \neq 0$. Let $j(k, i)$ be the smallest one of these integers. Then
\[ B_{k+1} e_i = B_{k-j(k,i)+1} e_i. \]
Let $i \in c_i$. Then
\[ B_{k-j(k,i)+1} e_i = J_{I_i}^{k-j(k,i)+1} e_i \]
by (2.17). Therefore,
\[
\| (F'(x^{k+1}) - B_{k+1}) e_i \| \\
\leq \| (F'(x^{k+1}) - F'(x^{k-j(k,i)+1})) e_i \| + \| (F'(x^{k-j(k,i)+1}) - B_{k-j(k,i)+1}) e_i \| \\
\leq \alpha_i \| x^{k+1} - x^{k-j(k,i)+1} \| + \alpha_i \| x^{k-j(k,i)+1} - x^{k-j(i,i)} \| \\
\leq \alpha_i \sum_{i=k-m}^{k} \| x^{i+1} - x^i \|,
\]
Hence,
\[
\| F'(x^{k+1}) - B_{k+1} \|_F^2 = \sum_{i=1}^{n} \| (F'(x^{k+1}) - B_{k+1}) e_i \|^2 \\
\leq \left( \sum_{j=k-m}^{k} \| x^{j+1} - x^j \| \right)^2 \sum_{i=1}^{n} \alpha_i^2.
\] (2.19)
Then (2.18) follows from (2.19).

3. A Kantorovich-type analysis. The following estimate for the SFD algorithm is sharper than that of Broyden’s algorithm given by Dennis [5].

**Theorem 3.1.** Assume that $F'$ satisfies Lipschitz condition (2.8) and that $\{x^k\}$ and $\{B_k\}$ are generated by Algorithm 2.2 with $\| x^{-1} - x^0 \| \leq \delta$. If $\{x^{j+1} - g_i^j, i = 0, 1, \cdots, p\}_{j=0}^{k} \subset D$, then
\[
\| F'(x^{k+1}) - B_{k+1} \|_F \leq \alpha \sum_{j=0}^{k} \| x^{j+1} - x^j \| + \alpha \delta.
\] (3.1)

**Proof.** Noticing that by Algorithm 2.2, $s_i^{-1} \neq 0$, $i = 1, 2, \cdots, n$, inequality (3.1) can be obtained immediately by setting $m = k+1$ in (2.18).

**Theorem 3.2.** Let $F'$ satisfy Lipschitz condition (2.8). Suppose that $x^{-1}, x^0 \in D$, and that $B_0$, generated by $x^{-1}$ and $x^0$, is a nonsingular $n \times n$ matrix such that
\[
\| x^{-1} - x^0 \| \leq \delta, \quad \| B_0^{-1} \| \leq \beta, \quad \| B_0^{-1} F(x^0) \| \leq \eta,
\] (3.2)
and
\[
\alpha \beta \delta < \frac{1}{3}.
\]
If $S(x^0, 2t^*) \subset D$, where
\[
i^* = \frac{1 - 3 \alpha \beta \delta}{3 \alpha \beta} (1 - \sqrt{1 - 6h}),
\] (3.3)
then \( \{x^k\} \), generated by Algorithm 2.2 without any global strategy, converges to \( x^* \), which is the unique root of \( F(x) \) in \( S(x^0, \bar{t}) \cap D \), where

\[
\bar{t} = \frac{1 - \alpha \beta \delta}{\alpha \beta} \left( 1 + \left( 1 - \frac{2 \alpha \beta \eta}{(1 - \alpha \beta \delta)^2} \right)^{1/2} \right).
\]

Proof. Consider the scalar iteration

\[
t_{k+1} - t_k = \beta f(t_k), \quad t_0 = 0, \quad k = 0, 1, 2, \cdots,
\]

where

\[
f(t) = \frac{3}{2} \alpha t^2 - \left( \frac{1 - 3 \alpha \beta \delta}{\beta} \right) t + \frac{\eta}{\beta}.
\]

By Taylor expansion,

\[
f(t_k) = f(t_{k-1}) + f'(t_{k-1})(t_k - t_{k-1}) + \frac{1}{2} f''(t_{k-1})(t_k - t_{k-1})^2.
\]

From (3.4),

\[
f(t_{k-1}) = \frac{1}{\beta} (t_k - t_{k-1}).
\]

Substituting (3.6) into (3.5), we have

\[
f(t_k) = 3 \left[ \frac{\alpha}{2} (t_k - t_{k-1}) + \alpha t_{k-1} + \alpha \delta \right] (t_k - t_{k-1}).
\]

Therefore, (3.4) can be rewritten as

\[
t_{k+1} - t_k = 3 \beta \left[ \frac{\alpha}{2} (t_k - t_{k-1}) + \alpha t_{k-1} + \alpha \delta \right] (t_k - t_{k-1}), \quad k = 1, 2, \cdots.
\]

From this equation, it follows that \( \{t_k\} \) is a monotonically increasing sequence and

\[
\lim_{k \to \infty} t_k = t^*,
\]

where \( t^* \) is the smallest root of \( f(t) \).

Now, by induction we will prove the following estimate:

\[
x^{k+1} - x^k \preceq t_{k+1} - t_k, \quad k = 0, 1, 2, \cdots.
\]

For \( k = 0 \), we have

\[
x^1 - x^0 \preceq \eta = t_1 - t_0.
\]

Suppose that (3.7) holds for \( k = 0, 1, 2, \cdots, m-1 \). Then

\[
x^m - x^0 \preceq t^*.
\]

Therefore, \( \{x^k\} \subset \tilde{S}(x^0, t^*) \). Notice that

\[
x^m - g_i^m - x^0 \preceq \|x^m - x^0\| + \|x^m - x^{m-1}\|
\]

\[
\preceq 2(t_k - t_0) = 2t^m \preceq 2t^*.
\]

Thus,

\[
\{x^k - g_i^k, i = 1, \cdots, p\} \subset \tilde{S}(x^0, 2t^*), \quad k = 0, 1, \cdots, m.
\]
Noticing that \( s_i^{-1} \neq 0, \ i = 1, 2, \cdots, n \), by Theorem 3.1, we have
\[
\|B_m - B_0\| \leq \|B_m - F'(x^m)\|_F + \|F'(x^m) - F'(0)\|_F + \|F'(0) - B_0\|_F \\
\leq 2\alpha \sum_{i=0}^{m-1} \|x^{i+1} - x^i\| + 2\alpha \delta \leq 2\alpha \delta + 2\alpha \frac{2/3}{\beta}.
\]

Then by Ortega and Rheinboldt’s Perturbation Lemma [10, p. 45],
\[
\|B_m^{-1}\| \leq \frac{\beta}{1 + 2/3} = 3\beta.
\]

Noticing that
\[
F(x^{m-1}) + B_{m-1}(x^m - x^{m-1}) = 0,
\]
we have
\[
F(x^m) = F(x^m - F(x^{m-1}) - F'(x^{m-1})(x^m - x^{m-1}) + (F'(x^{m-1}) - B_{m-1})(x^m - x^{m-1}).
\]

Therefore, using (3.1) and Lemma 4.1.12 in [7] we get
\[
\|x^{m+1} - x^m\| \leq \|B_m^{-1}\|_F \|F(x^m)\| \\
\leq \|B_m^{-1}\|_F [\|F(x^m) - F(x^{m-1}) - F'(x^{m-1})(x^m - x^{m-1})\| \\
+ \|F'(x^{m-1}) - B_{m-1}\|_F \|x^m - x^{m-1}\|] \\
\leq 3\beta \left[ \frac{\alpha}{2} \|x^m - x^{m-1}\| + \alpha \sum_{i=0}^{m-2} \|x^{i+1} - x^i\| + \alpha \delta \right] \|x^m - x^{m-1}\| \\
\leq 3\beta \left[ \frac{\alpha}{2} (t_m - t_{m-1}) + \alpha t_{m-1} + \alpha \delta \right] (t_m - t_{m-1}) = t_{m+1} - t_m.
\]

This completes the induction. By (3.7), it is easy to show that there is an \( x^* \in \bar{S}(x^0, t^*) \) such that
\[
\lim_{k \to \infty} x^k = x^*.
\]

The uniqueness of \( x^* \) in \( \bar{S}(x^0, t) \cap D \) can be obtained from Ortega and Rheinboldt’s Theorem 12.5.5 [10, p. 418] by setting \( G(x) = x - B_0^{-1}F(x) \).

4. Local convergence properties. To study the local convergence of the SFD algorithm, we assume that \( F: D \subset \mathbb{R}^n \to \mathbb{R}^n \) has the following property:

(4.1) There is an \( x^* \in D \), such that \( F(x^*) = 0 \) and \( F'(x^*) \) is nonsingular.

**Theorem 4.1.** Assume that \( F: D \subset \mathbb{R}^n \to \mathbb{R}^n \) satisfies (4.1) and that \( F' \) satisfies Lipschitz condition (2.8). Let \( \{x^k\} \) be generated by Algorithm 2.2 without any global strategy. Then there exist \( \varepsilon, \delta > 0 \) such that if \( x^{-1}, x^0 \in D \) satisfy
\[
\|x^0 - x^*\| < \varepsilon, \quad \|x^{-1} - x^0\| \leq \delta,
\]
then \( \{x^k\} \) is well defined and converges q-superlinearly to \( x^* \).

**Proof.** Notice that we can choose \( \varepsilon \) small enough so that \( \|B_0^{-1}F(x^0)\| \) is also small such that \( h < \beta \) and that \( \bar{S}(x^0, 2t^*) \subset D \), where \( h \) and \( t^* \) are defined by (3.2) and (3.3), respectively. When \( \delta \) is small enough, we also have
\[
\alpha \beta \delta < \frac{1}{2},
\]
where \( \beta \) is defined in Theorem 3.2. Therefore, by Theorem 3.2,
\[
\{x^{k+1} - g^k, i = 1, 2, \cdots, p \} \subset D, \quad k = 0, 1, \cdots.
\]
Thus, from (2.14) and the proof of Theorem 2.1, we have
\[
\|F'(x^*) - B_{k+1}\|_F^2 \\
= \sum_{i \in \Omega_1} \| (F'(x^*) - B_{k+1}) e_i \|^2 + \sum_{i \in \Omega_2} \| (F'(x^*) - B_k) e_i \|^2 \\
= \sum_{i=1}^p \sum_{j \in \mathcal{E}_i \cap \Omega_1} \| (F'(x^*) - F_i^{k+1}) e_j \|^2 + \sum_{i \in \Omega_2} \| (F'(x^*) - B_k) e_i \|^2 \\
= \sum_{i=1}^p \sum_{j \in \mathcal{E}_i \cap \Omega_1} \left[ \int_0^1 [F'(x^*) - F'(x^{k+1}) - g_i^k + t(g_i^k - g_i^{k-1})] e_j \, dt \right]^2 \\
+ \sum_{i \in \Omega_2} \| (F'(x^*) - B_k) e_i \|^2 \\
\leq \alpha^2 \left( \|x^* - x^{k+1}\| + \|x^{k+1} - x^k\| \right)^2 + \| F'(x^*) - B_k \|_F^2 \\
\leq \alpha^2 \left( 2\|x^* - x^{k+1}\| + \|x^* - x^k\| \right)^2 + \| F'(x^*) - B_k \|_F^2.
\]
Therefore,
\[
\|F'(x^*) - B_{k+1}\|_F \leq \|F'(x^*) - B_k\|_F + 3\alpha\sigma(x^k, x^{k+1}),
\]
where
\[
\sigma(x^k, x^{k+1}) = \max \{ \|x^{k+1} - x^*\|, \|x^k - x^*\| \}.
\]
Notice that by Theorem 2.1 and Lipschitz condition (2.8),
\[
\|F'(x^*) - B_0\|_F \leq \|F'(x^*) - F'(x^0)\|_F + \|F'(x^0) - B_0\|_F \\
\leq \alpha \|x^* - x^0\| + \alpha \|x^0 - x^{-1}\| \\
\leq \alpha (\varepsilon + \delta).
\]
Thus, by Dennis and Moré [6, Thm. 5.1], we know that \{x^k\} converges at least \(q\)-linearly to \(x^*\).

According to Dennis and Moré [6, Thm. 3.1], to get \(q\)-superlinear convergence, we need only to prove that
\[
\lim_{k \to \infty} \frac{\|(B_k - F'(x^*))(x^{k+1} - x^k)\|}{\|x^{k+1} - x^k\|} = 0.
\]
If for all \(1 \leq i \leq n\), \(s_i^k = 0\) appears consecutively in at most \(m\) steps, then by Theorem 2.4 it is easy to show that
\[
\lim_{k \to \infty} \|B_k - F'(x^*)\|_F = 0.
\]
Thus, (4.2) follows immediately from (4.3).

Otherwise, let
\[
A_2 = \{ i \in \{ 1, 2, \ldots, n \} : \text{For any } k > 0, \text{there exists at least one integer } m > k \text{ such that } s_i^m \neq 0 \},
\]
and let \(A_1 = \{ 1, \ldots, n \} \setminus A_2 \). Then
\[
B_k - F'(x^*) = \sum_{i \in A_1} (B_k - F'(x^*)) e_i e_i^T + \sum_{i \in A_2} (B_k - F'(x^*)) e_i e_i^T.
\]
From the definition of $A_1$, there exists a large integer $k_0$ such that $s_i^{K} = 0$ for all $i \in A_1$ and $k > k_0$. Therefore,

$$\sum_{i \in A_1} (B_k - F'(x^*))e_ie_i^T(x^{k+1} - x^k) = 0,$$

for $k > k_0$. Now we show that

$$\lim_{k \to \infty} \left\| \sum_{i \in A_2} (B_k - F'(x^*))e_i e_i^T \right\|_F = 0.$$

In the first part of the proof, we proved that $\lim_{k \to \infty} \|x^k - x^*\| = 0$. This implies that given $\varepsilon > 0$, there exists an integer $K$ such that

$$\|x^k - x^*\| < \frac{\varepsilon}{3\alpha} \quad \forall k > K.$$

By the definition of $A_2$, there exists an integer $K_1$, which depends on $K$, such that for every $i \in A_2$, there exists at least one integer $0 < j < K_1$ such that $s_i^{K+j} \neq 0$. Let $K = K + K_1$. For $k > \tilde{K}$ and $i \in A_2$, define

$$j(k, i) = \min \{ j : s_i^{K+j} \neq 0 \}.$$

Then $k - j(i, k) > K$. Let $i \in c_l$, $1 \leq l \leq p$; then we have that

$$B_k e_i = B_{k-j(i,k)} e_i = J_{k-j(i,k)} e_i.$$

Thus, by Lipschitz condition (2.8),

$$\left\| (B_k - F'(x^*))e_i \right\|^2 = \left\| (J_{k-j(k,i)} e_i) \right\|^2 = \left\| \int_0^1 (F'(x^{k-j(k,i)}) - g_i^{k-j(k,i)} + \alpha \alpha e_i e_i^T (g_i^{k-j(k,i)} - g_i^{k-j(k,i)}) - F'(x^*) e_i dt \right\|^2$$

$$= \left\| \alpha \int_0^1 \alpha e_i e_i^T (g_i^{k-j(k,i)} - g_i^{k-j(k,i)}) - F'(x^*) e_i dt \right\|^2$$

$$\leq \alpha^2 \int_0^1 \alpha e_i e_i^T (F'(x^*) e_i)^2$$

$$< \alpha^2 \int_0^1 \alpha e_i e_i^T (F'(x^*) e_i)^2$$

$$< \alpha^2 \left( \frac{\varepsilon}{3\alpha} + \frac{\varepsilon}{3\alpha} \right)^2 = \alpha^2 \frac{\varepsilon^2}{\alpha^2}.$$
By (4.4) and (4.5)
\[
\lim_{k \to \infty} \frac{\| (B_k - F'(x^*)) (x^{k+1} - x^k) \|}{\| x^{k+1} - x^k \|} = \lim_{k \to \infty} \frac{\| \sum_{i \in A_2} (B_k - F'(x^*)) e_i e_i^T (x^{k+1} - x^k) \|}{\| x^{k+1} - x^k \|} \\
\leq \lim_{k \to \infty} \left\| \sum_{i \in A_2} (B_k - F'(x^*)) e_i e_i^T \right\| = 0.
\]

**Theorem 4.2.** Assume that $F$, $x^{-1}$, $x^0$ and $\{x^k\}$ satisfy the hypotheses of Theorem 4.1. If, for any $1 \leq i \leq n$, $s_i^k = 0$ appears consecutively in at most $m$ steps, then the $r$-convergence order is not less than $r$, where $r$ is the unique positive root of
\[
(4.6)
\]
\[t^{m+2} - t^{m+1} - 1 = 0.\]

In particular, if $s_i^k \neq 0$, $i = 1, \cdots, n$, $k = 1, 2, 3, \cdots$, then $r = (1 + \sqrt{5})/2 \approx 1.618$.

**Proof.** Notice that (4.3) implies that there exist $k_0$ and $\beta_0 > 0$ such that $\| B_k^{-1} \| \leq \beta$ for all $k \leq k_0$. Thus, by Theorem 2.4,
\[
\| x^{k+1} - x^* \| = \| x^k - x^* - B_k^{-1} F(x^k) \| \\
\leq \| B_k^{-1} \|_F \left( \| F(x^k) - F(x^*) - F'(x^*)(x^k - x^*) \| \\
+ \left( \| F'(x^*) \|_F + \| F'(x^k) - B_k \|_F \right) \| x^k - x^* \| \right) \\
\leq \beta \left( \frac{3}{2} \alpha \| x^k - x^* \| + \alpha \sum_{j=k-m-1}^{k-1} \| x^{j+1} - x^j \| \right) \| x^k - x^* \|
\]
\[
\leq \frac{5}{2} \alpha \beta \left( \sum_{j=k-m-1}^{k} \| x^j - x^* \| \right) \| x^k - x^* \|.
\]

Thus, the desired result follows from Ortega and Rheinboldt's Theorem 9.2.9 [10, p. 291].

**5. Numerical results.** We computed some examples by the CPR algorithm, sparse Broyden (SB) algorithm, and the SFD algorithm. In this section, we compare the numerical results from the three algorithms to show roughly how the SFD algorithm works. The global strategy we used in computing the examples is the line search with backtracking strategy (see Dennis and Schnabel [7]). If $p^k = -B_k^{-1} F(x^k)$ is not a descent direction, then we try $-p^k$. If it is not a descent direction either, then the algorithm fails. The stopping test we used is
\[
\max_{1 \leq i \leq n} \frac{|x_i^{k+1} - x_i^k|}{\max \{ |x_i^{k+1}|, \text{typx}_i \}} \leq \varepsilon,
\]
with $\text{typx}_i = 10^{-8}$ and $\varepsilon = 10^{-5}$. The merit function we used is the $l_2$ norm of $F(x)$, i.e., $\| F(x) \|_2$. According to Dennis and Schnabel [7, p. 79], for CPR algorithm, instead of using a uniform step size $h_k$ at each step, we use
\[
h_k^i = \sqrt{\text{macheps}} x_i^k
\]
to perturb each component of $x$. We used double precision, and the machine precision was $2.22d - 16$.

Examples 5.1, 5.2, 5.3, and 5.4 are new, and they can be seen to be the extensions of the Rosenbrock [11] function (also see Moré, Garbow, and Hillstrom [9]) to nonlinear system of equations with tridiagonal, five-diagonal and seven-diagonal structures. Example 5.5 was given by Broyden [1] (also see Moré, Garbow, and Hillstrom [9]).
### Table 5.1

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>IT</th>
<th>NF</th>
<th>LN</th>
<th>ND</th>
<th>NC</th>
<th>ZR</th>
<th>IT</th>
<th>NF</th>
<th>LN</th>
<th>ND</th>
<th>NC</th>
<th>ZR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPR</td>
<td>22</td>
<td>88</td>
<td>15</td>
<td>0</td>
<td>22</td>
<td></td>
<td>22</td>
<td>88</td>
<td>15</td>
<td>0</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>SB</td>
<td>44</td>
<td>50</td>
<td>28</td>
<td>6</td>
<td>2</td>
<td></td>
<td>53</td>
<td>56</td>
<td>47</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SFD</td>
<td>24</td>
<td>73</td>
<td>13</td>
<td>4</td>
<td>0</td>
<td></td>
<td>22</td>
<td>67</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 5.2

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>IT</th>
<th>NF</th>
<th>LN</th>
<th>ND</th>
<th>NC</th>
<th>ZR</th>
<th>IT</th>
<th>NF</th>
<th>LN</th>
<th>ND</th>
<th>NC</th>
<th>ZR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPR</td>
<td>16</td>
<td>64</td>
<td>7</td>
<td>0</td>
<td>16</td>
<td></td>
<td>41</td>
<td>164</td>
<td>28</td>
<td>0</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>SB</td>
<td>40</td>
<td>46</td>
<td>16</td>
<td>5</td>
<td>2</td>
<td></td>
<td>319</td>
<td>325</td>
<td>257</td>
<td>22</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>SFD</td>
<td>17</td>
<td>52</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>40</td>
<td>121</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
</tbody>
</table>

### Table 5.3

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>IT</th>
<th>NF</th>
<th>LN</th>
<th>ND</th>
<th>NC</th>
<th>ZR</th>
<th>IT</th>
<th>NF</th>
<th>LN</th>
<th>ND</th>
<th>NC</th>
<th>ZR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPR</td>
<td>13</td>
<td>78</td>
<td>5</td>
<td>0</td>
<td>13</td>
<td></td>
<td>17</td>
<td>102</td>
<td>7</td>
<td>0</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>SB</td>
<td>31</td>
<td>36</td>
<td>11</td>
<td>2</td>
<td>1</td>
<td></td>
<td>93</td>
<td>103</td>
<td>47</td>
<td>14</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SFD</td>
<td>15</td>
<td>76</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td></td>
<td>17</td>
<td>86</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 5.4

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>IT</th>
<th>NF</th>
<th>LN</th>
<th>ND</th>
<th>NC</th>
<th>ZR</th>
<th>IT</th>
<th>NF</th>
<th>LN</th>
<th>ND</th>
<th>NC</th>
<th>ZR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPR</td>
<td>16</td>
<td>128</td>
<td>7</td>
<td>0</td>
<td>16</td>
<td></td>
<td>17</td>
<td>136</td>
<td>6</td>
<td>0</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>SB</td>
<td>65</td>
<td>72</td>
<td>23</td>
<td>4</td>
<td>1</td>
<td></td>
<td>61</td>
<td>75</td>
<td>25</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>SFD</td>
<td>16</td>
<td>113</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td></td>
<td>20</td>
<td>141</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 5.5

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>IT</th>
<th>NF</th>
<th>LN</th>
<th>ND</th>
<th>NC</th>
<th>ZR</th>
<th>IT</th>
<th>NF</th>
<th>LN</th>
<th>ND</th>
<th>NC</th>
<th>ZR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPR</td>
<td>5</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td></td>
<td>9</td>
<td>36</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>SB</td>
<td>7</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td>23</td>
<td>29</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>SFD</td>
<td>6</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>11</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The results are shown in Tables 5.1-5.5, where IT is the number of iterations, NF is the number of function \( F(x) \) evaluations, and LN is the number of line searches in which the step length \( \lambda < 1 \). ND is the number of nondecrease directions. NC is the number of the CPR steps. ZR is the number of the iterations that have an integer \( j \) such that \( |s_j^k| < \theta_j^k \). \( x^0 \) is the initial guess.

**Example 5.1** (tridiagonal).

\[
\begin{align*}
  f_1(x) &= 8(x_1 - x_2^2), \\
  f_j(x) &= 16x_j(x_j^2 - x_{j-1}) - 2(1 - x_j) + 8(x_j - x_{j+1}^2), \quad j = 2, \ldots, n - 1, \\
  f_n(x) &= 16x_n(x_n^2 - x_{n-1}) - 2(1 - x_n), \\
  n &= 9, \\
  x_1 &= (-1, -1, \ldots, -1)^T, \quad x_2 = (-0.5, -0.5, \ldots, -0.5)^T.
\end{align*}
\]

**Example 5.2** (tridiagonal).

\[
\begin{align*}
  f_1(x) &= 4(x_1 - x_2^2), \\
  f_j(x) &= 8x_j(x_j^2 - x_{j-1}) - 2(1 - x_j) + 4(x_j - x_{j+1}^2), \quad j = 2, \ldots, n - 1, \\
  f_n(x) &= 8x_n(x_n^2 - x_{n-1}) - 2(1 - x_n), \\
  n &= 36, \\
  x_1 &= (-2, -2, \ldots, -2)^T, \quad x_2 = (12, 12, \ldots, 12)^T.
\end{align*}
\]

**Example 5.3** (five-diagonal).

\[
\begin{align*}
  f_1(x) &= 4(x_1 - x_2^2) + x_2 - x_3^2, \\
  f_2(x) &= 8x_2(x_2^2 - x_1) - 2(1 - x_2) + 4(x_2 - x_3^2) + x_3 - x_4^2, \\
  f_j(x) &= 8x_j(x_j^2 - x_{j-1}) - 2(1 - x_j) + 4(x_j - x_{j+1}^2) \\
  &\quad + x_{j-1}^2 - x_{j-2} + x_{j+1} - x_{j+2}^2, \quad j = 3, \ldots, n - 2, \\
  f_{n-1}(x) &= 8x_{n-1}(x_{n-1}^2 - x_{n-2}) - 2(1 - x_{n-1}) + 4(x_{n-1} - x_n^2) + x_{n-2}^2 - x_{n-3}, \\
  f_n(x) &= 8x_n(x_n^2 - x_{n-1}) - 2(1 - x_n) + x_{n-1}^2 - x_{n-2}, \\
  n &= 36, \\
  x_1 &= (-1, -1, \ldots, -1)^T, \quad x_2 = (-3, -3, \ldots, -3)^T.
\end{align*}
\]

**Example 5.4** (seven-diagonal).

\[
\begin{align*}
  f_1(x) &= 4(x_1 - x_2^2) + x_2 - x_3^2 + x_3 - x_4^2, \\
  f_j(x) &= 8x_j(x_j^2 - x_{j-1}) - 2(1 - x_j) + 4(x_j - x_{j+1}^2) + x_{j-1}^2 - x_{j-2} \\
  &\quad + x_{j+1}^2 - x_{j+2} + x_{j-2}^2 - x_{j-3} + x_{j+2} - x_{j+3}, \quad j = 2, \ldots, n - 1, \\
  f_n(x) &= 8x_n(x_n^2 - x_{n-1}) - 2(1 - x_n) + x_{n-1}^2 - x_{n-2} + x_{n-2}^2 - x_{n-3}, \\
  n &= 36, \\
  x_1 &= (-2, -2, \ldots, -2)^T, \quad x_2 = (-3, -3, \ldots, -3)^T.
\end{align*}
\]
Example 5.5 (Broyden tridiagonal function).
\[ f_i(x) = (3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1, \]
\[ x_0 = x_{n+1} = 0, \]
\[ n = 36, \]
\[ x_1 = (-1, -1, \cdots, -1)^T, \quad x_2 = (-15, -15, \cdots, -15)^T. \]

From the numerical results it can be seen that the SFD algorithm makes some compromise between the CPR algorithm and the SB algorithm. That is, the number of iterations required by the SFD algorithm is close to the CPR algorithm and it is much less than that required by the sparse Broyden algorithm when the problem is badly nonlinear and/or when the starting point is far away from the solution. On the other hand, for some problems, the number of function evaluations for the SFD algorithm is less than that for the CPR algorithm and more than that for the SB algorithm, and when the problem is badly nonlinear and the starting point is far away from the solution it is even less than that for the SB algorithm. Moreover, it seems that much more savings on function evaluations are obtained whenever line searches are used. It is interesting that for the test problems the CPR algorithm never takes nondecrease directions. The SFD algorithm sometimes takes some nondecrease directions. However, the number of nondecrease directions for the SFD algorithm is usually much less than that for the SB algorithm. We also see that when the number of the groups in a partition increases the efficiency of the SFD algorithm decreases.

6. Concluding remarks. We have presented an algorithm for solving sparse nonlinear systems of equations. This algorithm is based on consistent partitions of the columns of the Jacobians, and it is a combination of the CPR-CM algorithm and a secant algorithm. This algorithm incorporates the advantages of the CPR-CM algorithm and secant algorithms in such a way that it reduces by one the number of function evaluations required by the CPR-CM algorithm at each iteration, and it has good local convergence properties. We have shown that the SFD algorithm is locally \( q \)-superlinearly convergent, and that under reasonable assumptions, the \( r \)-convergence order of the SFD algorithm is not less than \((1 + \sqrt{5})/2\), which is the \( r \)-convergence order of the one dimensional secant algorithm. Our numerical results indicate that when \( p \), the number of the groups in a partition of the columns of the Jacobian, is not large, especially when the problem to be solved is badly nonlinear, the SFD algorithm is competitive with the CPR-CM algorithm and the sparse Broyden algorithm.

The idea exploited here can also be used with Powell and Toint's [12] work, which will lead to a method for unconstrained optimization problems. This will be our future work.

Acknowledgments. This paper is part of my Ph.D. thesis. I would like to express my deepest thanks to my advisor John Dennis. He read this paper several times and made many helpful suggestions and corrections which improved this paper a lot. I am also very grateful to him for his support, his guidance and his encouragement. I would also like to thank the referees for their helpful suggestions and corrections.

REFERENCES


