A Column-Secant Update Technique for Solving Systems of Nonlinear Equations

Guangye Li
Tingzhan Liu

CRPC-TR91182
September, 1991

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

A COLUMN-SECANT UPDATE TECHNIQUE FOR SOLVING SYSTEMS OF NONLINEAR EQUATIONS

GUANGYE LI * AND TINGZHAN LIU †

Abstract. This paper presents a QR update implementation of the successive column correction (SCC) method and a column-secant modification of the SCC method, which is called the CSSCC method. The computational cost of the QR update technique for the SCC method is much less than that for Broyden's method. The CSSCC method uses function values more efficiently than the SCC method, and it is shown that the CSSCC method has better local q-convergence and r-convergence rates than the SCC method. The numerical results show that the SCC method and the CSSCC method with the QR update technique are competitive with some well known methods for some standard test problems.

Key Words. nonlinear equatios, Jacobian, Newton's method, q-convergence.

AMS(MOS) subject classification. 65H10

1. Introduction.

In this paper, we consider the system of nonlinear equations

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

where $F : R^n \to R^n$ is continuously differentiable on an open convex set $D \subset R^n$. To solve (1.1), we consider the following Newton-like iterative method:

$$x_{k+1} = x_k - B_k^{-1} F(x_k), \quad k = 0, 1, ..., \tag{1.2}$$

where B_k is an approximation to $F'(x_k)$. A well known approach to obtain a good approximation to $F'(x_k)$ is to compute the columns of B_k by finite differences, i.e.

$$B_k e_j = \frac{F(x_k + h_k e_j) - F(x_k)}{h_k}, \quad j = 1, ..., n, \tag{1.3}$$

where e_j is the jth column of the identity matrix. The finite-difference method is shown to be locally q-quadratically convergent by properly choosing step length h_k (see Dennis and Schnabel[3]). However, at each iteration $n+1$ function values $(F(x))$ are needed to form matrix B_k, which is quite expensive for many problems.

Another well known method is Broyden's method (see Broyden [1]), in which B_k is updated at each iteration by the update formula

$$B_k = B_{k-1} + \frac{(y_{k-1} - B_{k-1} s_{k-1}) s_{k-1}^T}{s_{k-1}^T s_{k-1}}, \tag{1.4}$$

where $s_{k-1} = x_k - x_{k-1}$, $y_{k-1} = F(x_k) - F(x_{k-1})$. It is easy to verify that the updated matrix B_k satisfies the secant equation

$$B_k s_{k-1} = y_{k-1}. \tag{1.5}$$

* Computer and Information Technology Institute, Rice University, P. O. Box 1892, Houston, Texas 77251-1892.

† Computer Center, Jilin University Changchun, Jilin, P. R. China

It is shown that Broyden's method is locally q-superlinearly convergent (see Broyden, Dennis and More [2]). The most attractive advantage of Broyden's method is that at each iteration only one function value is needed. Gill, Golub, Murray and Saunders [5] proposed a QR update technique which can reduce the number of the arithmetic operations for solving the linear system

$$B_k s = -F(x_k) \tag{1.6}$$

from $O(n^3)$ to $O(n^2)$.

Polak [10] gave a successive column correction method for unconstrained optimization. Feng and Li [4] studied the successive column correction (SCC) method for systems of nonlinear equations. Liu and Li [7] further studied the SCC method and gave a secant modification of the SCC method (SSCC) by using Broyden's update.

In this paper, we introduce a QR implementation of the SCC method. Using this QR update technique the number of arithmetic operations for solving linear system (1.6) in the SCC method will be much less that in Broyden's method. Our numerical results also show that the SCC method with the QR update technique is better than the finite-difference method and Broyden's method for many well known test problems from the execution time point of view. The main work of this paper is to give a column-secant modification of the SCC method, which is called the CSSCC method. We will show that the CSSCC method has a better q-convergence rate and a better r-convergence order than both the SCC method and the SSCC method. Our numerical results show that the CSSCC method behaves well for many well known test problems.

This paper is organized in the following way: In section 2, we briefly describe the SCC method, the SSCC method and their convergence properties. In section 3, we present the QR implementation of the SCC method. In section 4, we give the CSSCC method and its local convergence results. In section 5, we give some numerical results and some comparisons. In section 6, we give some concluding remarks.

In this paper, $\|.\|_F$ denotes the Frobenius norm of matrix, and $\|.\|$ denotes the l_2-norm of a vector.

2. The SCC method and the SSCC method. The basic idea of the SCC method is that the columns of B_k are corrected by finite differences successively and periodically. This correction can be formulated by the following formula:

$$B_k = B_{k-1}(I - e_{i_k} e_{i_k}^T) + \frac{q_k e_{i_k}^T}{h_k}, \quad i_k = k(mod\, n), \tag{2.1}$$

$$q_k = F(x_k + p_k) - F(x_k), \tag{2.2}$$

$$p_k = h_k e_{i_k},$$

where h_k is a scalar. The SCC method can be formulated as follows:

ALGORITHM 2.1. *Given matrix B_0 and $x_0 \in R^n$, do the following:*

At the initial step:

1. Set $l = 0$.
2. Solve $B_0 s = -F(x_0)$.
3. Choose x_1 by $x_1 = x_0 + s$ or by a global strategy.

At each step $k \geq 1$:
1. Choose a scalar h_k.
2. If $l < n$ set $l = l + 1$, otherwise set $l = 1$.
3. Set

$$B_k e_l = \frac{F(x_k + h_k e_l) - F(x_k)}{h_k}, \tag{2.3}$$

and

$$B_k e_i = B_{k-1} e_i, \quad i = 1, ..., l-1, l+1, ..., n. \tag{2.4}$$

4. Solve $B_k s = -F(x_k)$.
5. Choose x_{k+1} by $x_{k+1} = x_k + s$ or a global strategy.
6. Check convergence.

Note that to form B_k in the SCC method, only two function values are needed at each iteration. To study the convergence properties, sometimes we assume the $F'(x)$ satisfies the following Lipschitz condition: For any $1 \leq i \leq n$ there exists $\alpha_i > 0$ such that

$$\|(F'(x) - F'(y))e_i\| \leq \alpha_i \|x - y\|, \quad \forall x, y \in D. \tag{2.5}$$

Let $\alpha = (\sum_{i=1}^n \alpha_i^2)^{1/2}$, then (2.5) implies that

$$\|F'(x) - F'(y)\|_F \leq \alpha \|x - y\|, \quad \forall x, y \in D. \tag{2.6}$$

LEMMA 2.1. *Let $\{x_k\}$, $\{B_k\}$ be generated by the SCC method. Assume that $F'(x)$ satisfies Lipschitz condition (2.5) and that $x_k \in D$ and $x_k + p_k \in D$. Then*

$$\|(B_k - F'(x_k))e_{i_k}\| \leq \frac{1}{2}\alpha_{i_k}|h_k|.$$

proof. Let

$$J_k = \int_0^1 F'(x_k + tp_k)dt.$$

Then, from Lipschitz condition (2.5),

$$\begin{aligned} \|(B_k - F'(x_k))e_{i_k}\| &= \|(J_k - F'(x_k))e_{i_k}\| \\ &= \|\int_0^1 (F'(x_k + tp_k) - F'(x_k))dt e_{i_k}\| \\ &\leq \alpha_{i_k}\|p_k\| \int_0^1 t dt \leq \frac{1}{2}\alpha_{i_k}|h_k|. \end{aligned}$$

LEMMA 2.2. *Assume that $F'(x)$ satisfies Lipschitz condition (2.5). Let $\{x_j\}_{j=1}^k$ and $\{B_j\}_{j=1}^k$ be generated by the SCC method, with B_0 satisfying $\|B_0 - F'(x_0)\|_F \le \delta$. If $\{x_j\}_{j=1}^k \subset D$ and $\{x_j + p_j\}_{j=1}^k \subset D$, then for $k < n$,*

$$\|B_k - F'(x_k)\|_F \le \alpha(2\bar{e}_k + \bar{h}_k) + \delta, \tag{2.7}$$

and for $k \ge n$,

$$\|B_k - F'(x_k)\|_F \le \alpha(\bar{e}_k + \bar{h}_k), \tag{2.8}$$

where

$$\bar{e}_k = \max_{1\le j\le m(k)} \{\|x_k - x_{k-j}\|\}, \ \bar{h}_k = \frac{1}{2} \max_{0\le j\le m(k)} \{h_{k-j}\},$$

$$m(k) = \min\{k,\ n-1\} \textit{ and } h_0 = 0.$$

Proof. We prove (2.8) first. By lemma 2.2 and Lipschitz condition (2.5) for $k \ge n$,

$$\begin{aligned}
\|B_k - F'(x_k)\|_F^2 &= \sum_{m=n-k+1}^{k} \|(B_k - F'(x_k))e_{i_m}\|^2 \\
&= \sum_{m=n-k+1}^{k} \|(B_m - F'(x_k))e_{i_m}\|^2 \\
&\le \sum_{m=n-k+1}^{k} (\|(B_m - F'(x_m))e_{i_m}\| + \|(F'(x_m) - F'(x_k))e_{i_m}\|)^2 \\
&\le \sum_{m=n-k+1}^{k} (\frac{1}{2}\alpha_{i_m}|h_m| + \alpha_{i_m}\|x_k - x_m\|)^2 \\
&\le (\bar{e}_k + \bar{h}_k)^2 \sum_{m=n-k+1}^{k} \alpha_{i_m}^2 = (\bar{e}_k + \bar{h}_k)^2\alpha^2, \qquad (2.9)
\end{aligned}$$

which implies (2.8).

New we consider the case where $1 \le k < n$. In this case,

$$\begin{aligned}
\|B_k - F'(x_k)\|_F &= \|\sum_{i=1}^{k}(B_k - F'(x_k))e_ie_i^T + \sum_{i=k+1}^{n} (B_0 - F'(x_k))e_ie_i^T\|_F \\
&\le \|\sum_{i=1}^{k}(B_k - F'(x_k))e_ie_i^T\|_F + \|B_0 - F'(x_k)\|_F.
\end{aligned}$$

Similar to (2.9),

$$\|\sum_{i=1}^{k}(B_k - F'(x_k))e_ie_i^T\|_F \le \alpha(\bar{e}_k + \bar{h}_k).$$

Thus,

$$\begin{aligned}\|B_k - F'(x_k)\|_F &\leq \alpha(\bar{e}_k + \bar{h}_k) + \|B_0 - F'(x_0)\|_F + \|F'(x_0) - F'(x_k)\|_F \\ &\leq \alpha(\bar{e}_k + \bar{h}_k) + \alpha\|x_k - x_0\| + \delta \leq \alpha(2\bar{e}_k + \bar{h}_k) + \delta.\end{aligned}$$

Applying Lemma 2.2, we have the following convergence results for the SCC method.

THEOREM 2.3. *Assume that $F(x)$ satisfies the following standard condition for local convergence:*

(2.10) *There is $x^* \in D$ such that $F(x^*) = 0$ and $F'(x^*)$ is nonsingular.*

Also assume that $F'(x)$ satisfies Lipschitz condition (2.5). Let $\{x_k\}$ be generated by the SCC method. Then there exist ϵ, δ, $h > 0$ such that if $0 < |h_k| < h$ and $x_0 \in D$, B_0 satisfy

$$\|x_0 - x^*\| \leq \epsilon,\ \|B_0 - F'(x_0)\|_F \leq \delta,$$

then $\{x_k\}$ is well defined and converges q-linearly to x^* . If $\lim_{k\to\infty} |h_k| = 0$, then the convergence is q-superlinear. If there exists some constant $C > 0$ such that $|h_k| \leq C\|F(x_k)\|$ then the convergence is n-step q-quadratic.

The proof of this theorem is similar to that of Theorem 4.2 in this paper.

THEOREM 2.4. *Assume that $F(x)$ and $F'(x)$ satisfy the hypotheses of Theorem 2.4 and that $|h_k| \leq C\|F(x_k)\|$, Then the r-convergence order of the SCC method is not less that τ , where τ is the unique positive root of the equation*

$$t^n - t^{n-1} - 1 = 0.$$

The proof of this theorem is similar to that of Theorem 4.3 in this paper.

The basic idea of the SSCC method is to have a secant modification on B_k by using the information $F(x_{k-1})$ we already have to get a better approximation to $F'(x_k)$. The SSCC method can be formulated as follows:

ALGORITHM 2.2. *Given matrix B_0 and $x_0 \in R^n$, do the following:*

At the initial step:

1. *Set $l = 0$ and $\bar{B}_0 = B_0$.*
2. *Solve $\bar{B}_0 s = -F(x_0)$.*
3. *Choose x_1 by $x_1 = x_0 + s$ or by a global strategy.*

At each step $k \geq 1$:

1. *Update B_{k-1} by Algorithm 2.1 to get B_k.*
2. *Update B_k by the Broyden update to get $\bar{B}_k$.*
3. *Solve $\bar{B}_k s = -F(x_k)$.*
4. *Choose x_{k+1} by $x_{k+1} = x_k + s$ or by a global strategy.*
5. *Check convergence.*

The following convergence results can be found in Liu and Li [7].

THEOREM 2.5. *The SSCC method has at least the same local convergence properties as the SCC method.*

3. A QR update technique for the SCC method. In Algorithm 2.1, solving the linear system needs $O(n^3)$ multiplications, which may be the main part of the whole computational cost for many problems. In this section, we discuss a QR update technique, which reduces the number of multiplications needed for solving the linear system to $O(n^2)$. This technique is similar to the one given by Gill, Golub, Murray and Saunders [5] for Broyden's method. However, it is much cheaper than the latter.

For convenience, in this section, we omit subscripts k in SCC update formula (2.1), and rewrite it as

$$\bar{B} = B(I - e_i e_i^T) + \frac{q e_i^T}{h}. \tag{3.1}$$

Suppose we already have $B = QR$, where Q is an orthogonal matrix and R is an upper triangular matrix. Now we want to obtain $\bar{Q}$ and $\bar{R}$ such that

$$\bar{B} = \bar{Q}\bar{R}. \tag{3.2}$$

From (3.1),

$$\bar{B} = Q[R(I - e_i e_i^T) + \frac{Q^T q e_i^T}{h}]. \tag{3.3}$$

Let

$$\bar{Q} = Q^T q/h = (\bar{q}_1, ..., \bar{q}_n)^T,$$

and

$$R_1 = \begin{bmatrix} r_{11} & \cdot & \cdot & \bar{q}_1 & \cdot & \cdot & \cdot & r_{1n} \\ & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & \bar{q}_i & \cdot & \cdot & \cdot & r_{in} \\ & & & \bar{q}_{i+1} & \cdot & \cdot & \cdot & r_{i+1,n} \\ & & & \cdot & & \cdot & \cdot & \cdot \\ & & & \cdot & & & \cdot & \cdot \\ & & & \bar{q}_n & & & & r_{nn} \end{bmatrix}.$$

Then (3.3) can be written as

$$\bar{B} = QR_1. \tag{3.4}$$

For $i < n$, using $n - i - 1$ Givens rotations, R_1 may be reduced to

$$\tilde{R} = G_{n-i-1} G_{n-i-2} ... G_1,$$

where

$$\tilde{R} = \begin{bmatrix} r_{11} & \cdot & \cdot & \bar{q}_1 & \cdot & \cdot & \cdot & r_{1n} \\ & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & \bar{q}_i & \cdot & \cdot & \cdot & r_{in} \\ & & & & \star & \cdot & \cdot & \star \\ & & & & & \cdot & \cdot & \cdot \\ & & & & & & \cdot & \cdot \\ & & & \star & \star & \cdot & \cdot & \star \end{bmatrix}.$$

Using another $n - i$ Givens rotations, we may reduce $\tilde{R}$ to an upper triangular matrix $\bar{R}$ i.e.,

$$\bar{R} = G_{2n-2i-1}G_{2n-2i-2}...G_2G_1R_1.$$

Let

$$\bar{Q} = QG_1^TG_2^T...G_{2n-2i-1}^T.$$

Then, $\bar{Q}$ is an orthogonal matrix and

$$\bar{B} = \bar{Q}\bar{R}.$$

Now we discuss the computational cost of the QR update technique mentioned above.

Note that the number of multiplications for this QR update varies from iteration to iteration. When the nth column of the approximate Jacobian is corrected, i.e. when $i = n$ in (3.1), then R_1 is already an upper triangular matrix, and the QR update costs nothing. When the $(n-1)$th column is corrected, only one Givens rotation is needed. When the ith column $(i > 1)$ is corrected $2n - 2i - 1$ Givens rotations are needed, and the number of multiplications for the QR update is

$$12n^2 - 16ni + 4i^2 + 12n - 16i - 9.$$

Note that at each iteration only one column is corrected in the SCC method. Thus the total number of multiplications needed for the QR updates in every n iterations for the SCC method is

$$\sum_{i=1}^{n-1} 12n^2 - 16ni + 4i^2 + 12n - 16i - 9.$$

On the other hand, for Broyden's method the number of multiplications for the QR update at each iteration is fixed. It is equal to that when the first column is corrected in the SCC method, i.e. it is $12n^2 - 4n - 21$.

From the discussion above, we have the following results:

THEOREM 3.1. *The total number of multiplications needed for the QR update in n iterations for the SCC method is*

$$\frac{16}{3}n^3 - 2n^2 - \frac{37}{3}n + 9. \tag{3.5}$$

THEOREM 3.2. *The total number of multiplications needed for the QR updates in n iterations for the Broyden's method is*

$$12n^3 - 4n^2 - 21n. \tag{3.6}$$

Comparing (3.5) with (3.6), we see that in every n steps the computational cost for QR updates in the SCC method is less than a half of those in Broyden's method.

In practice, the number of iterations needed to reach the desired accuracy is generally not a factor of n. Specially, when n is large, the desired accuracy may be reached in m iterations, where m is much less than n. Note that in the SCC method if we reverse the column correction order, i.e. we correct B_k from the nth column to the first column, then the convergence results will not be affected. If we do so, we may gain some savings (sometime great savings) on the QR updates. Therefore, we suggest to change column correction update formula (2.1) to

$$B_k = B_{k-1}(I - e_{i_k}e_{i_k}^T) + \frac{q_k e_{i_k}^T}{h_k}, \; i_k = n + 1 - k(mod\, n).$$

Now the SCC method should be formulated as follows:

ALGORITHM 3.1. *Given matrix B_0 and $x_0 \in R^n$, do the following:*

1. *Set $l = n$.*
2. *Solve $B_0 s = -F(x_0)$.*
3. *Choose x_1 by $x_1 = x_0 + s$ or by a global strategy.*

At each step $k \geq 1$:

1. *Compute B_k by (2.3) and (2.4).*
2. *Solve $B_k s = -F(x_k)$.*
3. *Choose x_{k+1} by $x_{k+1} = x_k + s$ or a global strategy.*
4. *If $l > 1$ set $l = l - 1$, otherwise set $l = n$.*
5. *Check convergence.*

4. The column-secant modification of the SCC method. The numerical results in [7] show that the secant modification of the SCC method using Broyden's update (SSCC) usually needs less number of iterations to get the solution, and therefore it uses less function values than the SCC method. However, theoretically we could not say that it is better than the SCC method. Moreover, the cost of the additional QR update for Broyden's update is significant, which makes the execution time longer than the SCC method and Broyden's method for many test problems. To overcome these drawbacks, we consider a column-secant update here. The basic idea of this technique is that after one column of B_k is corrected, the next column, which maintain the earliest information, is updated to make the updated matrix $\bar{B}_k$ satisfies the secant equation

$$\bar{B}_k s_{k-1} = y_{k-1}, \tag{4.1}$$

where $s_{k-1} = x_k - x_{k-1}$, $y_{k-1} = F(x_k) - F(x_{k-1})$. The column-secant update is formulated as follows: If

$$e_{i_{k+1}}^T s_{k-1} \geq \theta \|s_{k-1}\|_\infty, \tag{4.2}$$

then

$$\bar{B}_k = B_k + \frac{(y_{k-1} - B_k s_{k-1})e_{i_{k+1}}^T}{e_{i_{k+1}}^T s_{k-1}}. \tag{4.3}$$

Otherwise,

$$\bar{B}_k = B_k. \tag{4.4}$$

The column-secant modification of the SCC method (CSSCC) is as follows:

ALGORITHM 4.1. *Given matrix $B_0, x_0 \in R^n$ and a small scalar $\theta > 0$, do the following: At the initial step:*

1. *Set $l = n$ and $\bar{B}_0 = B_0$.*
2. *Solve $\bar{B}_0 s = -F(x_0)$.*
3. *Choose x_1 by $x_1 = x_0 + s$ or by a global strategy.*

At each step $k \geq 1$:

1. *Compute B_k by (2.3) and (2.4).*
2. *If $l - 1 \geq 1$ set $m = l - 1$, otherwise set $m = n$.*
3. *If $|e_m^T s_{k-1}| \geq \theta \|s_{k-1}\|_\infty$, then set*

$$\bar{B}_k = B_k + \frac{(y_{k-1} - B_k s_{k-1}) e_m^T}{e_m^T s_{k-1}},$$

otherwise, set

$$\bar{B}_k = B_k.$$

4. *Solve $\bar{B}_k s = -F(x_k)$.*
5. *Choose x_{k+1} by $x_{k+1} = x_k + s$ or a global strategy.*
6. *If $l > 1$ set $l = l - 1$, otherwise set $l = n$.*
7. *Check convergence.*

It can be see from Algorithm 4.1 that the number of the function evaluations needed to form $\bar{B}_k$ at each iteration in the CSSCC method is also two, the same as the SCC method, and the additional cost of the column-secant update is mainly a multiplication of a matrix and a vector plus the same QR update as that in the SCC method.

Now we study the convergence properties of the CSSCC method. The following lemma shows that we may have a better approximation to the Jacobian by using Algorithm 4.1 than that by Algorithm 2.1. Comparing the following lemma with Lemma 2.2, we see that $m(k)$ is small here. This means that the current approximation to the Jacobian obtained by Algorithm 4.1 only depends on the approximations in the previous $n-2$ steps in stead of $n-1$ steps required by Algorithm 2.1.

LEMMA 4.1. *Let $\{x_j\}_{j=1}^k$, $\{B_j\}_{j=1}^k$ and $\{\bar{B}_j\}_{j=0}^k$ are generated by Algorithm 4.1. Assume $F(x)$, $F'(x)$, B_0, $\{x_j\}_{j=1}^k$, $\{x_j + p_j\}_{j=1}^k$ satisfy the hypotheses of Lemma 2.2. If (4.2) is satisfied for all i_{k+1}, $k = 0, 1, 2, \ldots$, then there exists a constant $C_1 > 0$ such that for $k < n$*

$$\|\bar{B}_k - F'(x_k)\|_F \leq \alpha C_1 (2\hat{e}_k + \hat{h}_k) + \delta, \tag{4.5}$$

and for $k \geq n$,

$$\|\bar{B}_k - F'(x_k)\|_F \leq \alpha C_1 (\hat{e}_k + \hat{h}_k), \tag{4.6}$$

where

$$\hat{e}_k = \max_{1\le j\le m(k)}\{\|x_k - x_{k-j}\|\},\ \ \hat{h}_k = \frac{1}{2}\max_{0\le j\le m(k)}\{h_{k-j}\},$$

and $m(k) = \min\{k, n-2\},\ \ h_0 = 0.$

Proof. We prove (4.6) only. Inequality (4.5) can be obtained by applying (4.6) and using an argument similar to the second part of the proof of Lemma 2.2. Let

$$\bar{J}_{k-1} = \int_0^1 F'(x_{k-1} + ts_{k-1})dt. \tag{4.7}$$

Then

$$\bar{J}_{k-1}s_{k-1} = y_{k-1}. \tag{4.8}$$

Suppose that the lth column is corrected by the SCC method at the kth iteration then the mth column is updated by (4.3), where $m = l-1$ if $l > 1$, $m = n$ otherwise. Let

$$\Omega = \{1, 2, ..., n\},$$

and

$$\Omega_m = \{j \in \Omega:\ j \ne m\}.$$

From (4.3) and (4.8),

$$\begin{aligned}
\|(\bar{B}_k - \bar{J}_{k-1})e_m\| &= \|B_k e_m + \frac{1}{e_m^T s_{k-1}}(y_{k-1} - B_k s_{k-1}) - \bar{J}_{k-1}e_m\| \\
&= \|B_k e_m + \frac{1}{e_m^T s_{k-1}}(\bar{J}_{k-1} - B_k)\sum_{j=1}^{n} e_j^T s_{k-1}e_j - \bar{J}_{k-1}e_m\| \\
&= \|\sum_{j\in\Omega_m}(\bar{J}_{k-1} - B_k)e_j\frac{e_j^T s_{k-1}}{e_m^T s_{k-1}}\| \\
&\le \frac{1}{\theta}\sum_{j\in\Omega_m}\|(\bar{J}_{k-1} - B_k)e_j\|. \qquad (4.9)
\end{aligned}$$

Note that from (4.7) and Lipschitz condition (2.5), for $j \in \Omega$,

$$\begin{aligned}
\|(\bar{J}_{k-1} - F'(x_k))e_j\| &= \|\int_0^1 (F'(x_{k-1} + ts_{k-1}) - F'(x_k))e_j dt.\| \\
&\le \alpha_j\|s_{k-1}\|\int_0^1 (1-t)dt = \frac{\alpha_j}{2}\|s_{k-1}\|. \qquad (4.10)
\end{aligned}$$

Without loss of generality, we assume that $\theta \le 1$. Thus, from (4.9) and (4.10),

$$\begin{aligned}
\|(\bar{B}_k - F'(x_k))e_m\| &\le \|(\bar{B}_k - \bar{J}_{k-1})e_m\| + \frac{\alpha_m}{2}\|s_{k-1}\| \\
&\le \frac{1}{\theta}\sum_{j\in\Omega_m}(\|(F'(x_k) - B_k)e_j\| + \frac{\alpha_j}{2}\|s_{k-1}\|) + \frac{\alpha_m}{2}\|s_{k-1}\|
\end{aligned}$$

$$(4.11) \qquad \leq \frac{1}{\theta}\Big(\sum_{j\in\Omega_m} \|(F'(x_k)-B_k)e_j\| + \frac{1}{2}\|s_{k-1}\|\sum_{j=1}^{n}\alpha_j\Big).$$

Note that the mth column is the only column which maintain the information of the $(k-n+1)$th step. Therefore, it can be seen from the proof of Lemma 2.2 that for any $j \in \Omega_m$,

$$(4.12) \qquad \|(F'(x_k)-B_k)e_j\| \leq \alpha_j(\hat{e}_k + \hat{h}_k).$$

Thus, from (4.11),

$$\|(\bar{B}_k - F'(x_k))e_m\| \leq \frac{3}{2\theta}(\hat{e}_k+\hat{h}_k)\sum_{j=1}^{n}\alpha_j \leq \frac{3\sqrt{n}}{2\theta}\alpha(\hat{e}_k+\hat{h}_k).$$

Let $C_1 = (\frac{9n}{4\theta^2}+1)^{1/2}$. Then from (4.12),

$$\begin{aligned}
\|F'(x_k)-\bar{B}_k\|_F &= \sum_{j=1}^{n}\|(F'(x_k)-\bar{B}_k)e_j\|^2 \\
&= \|(F'(x_k)-\bar{B}_k)e_m\|^2 + \sum_{j\in\Omega_m}\|(F'(x_k)-B_k)e_j\|^2 \\
&\leq C_1^2\alpha^2(\hat{e}_k+\hat{h}_k)^2,
\end{aligned}$$

which implies (4.6).

THEOREM 4.2. *Assume that $F(x)$, $F'(x)$ satisfy the hypotheses of Theorem 2.3. Let $\{x_k\}$ be generated by Algorithm 4.1 without any global strategy. Then there exist ϵ, δ, $h > 0$ such that if $0 < |h_k| < h$ and $x_0 \in D$, B_0 satisfy*

$$\|x_0 - x^*\| \leq \epsilon, \quad \|B_0 - F'(x_0)\|_F \leq \delta,$$

then $\{x_k\}$ is well defined and converges q-linearly to x^*. If $\lim_{k\to\infty}|h_k| = 0$, then the convergence is q-superlinear. If there exists some constant C such that $|h_k| \leq C\|F(x_k)\|$, then the convergence is $(n-1)$-step q-quadratic.

Proof. Since $x^* \in D$ and D is an open convex set, we can choose ϵ so that $S(x^*, 2\epsilon) \equiv \{x : \|x - x^*\| < 2\epsilon\} \subset D$. Also, we can choose ϵ, δ and h so that

$$2\beta C_1(\alpha(\frac{9\epsilon}{2}+h)+\delta) < \frac{1}{2}, \quad h < \epsilon,$$

where C_1 is defined in Lemma 4.1, and $\beta > 0$ satisfies $\|F'(x^*)^{-1}\|_F \leq \beta$. Without loss of generality, we assume that $C_1 \geq 1$.

We first show by induction on k that

$$(4.13) \qquad \|x_{k+1} - x^*\| \leq \frac{1}{2}\|x_k - x^*\|, \; k = 0, 1, 2, \ldots.$$

It is easy to show that (4.13) holds for $k = 0$. Now suppose (4.13) holds for $k = 1, 2, \ldots, m-1$. We show that it also holds for $k = m$. By (4.13),

$$\|x_m + p_m - x^*\| \leq \|x_m - x^*\| + \|p_m\| \leq \|x_0 - x^*\| + h < 2\epsilon.$$

Thus, $\{x_k + p_k\}_{k=1}^{m} \subset S(x^*, 2\epsilon) \subset D$. By Lemma 4.1, there exists an integer $1 \le j_0 \le \max\{m, n-2\}$ such that

$$\begin{aligned} \|\bar{B}_m - F'(x_m)\|_F &\le C_1(\alpha(2\|x_m - x_{m-j_0}\| + \bar{h}_m) + \delta) \\ &\le C_1(\alpha(4\|x^* - x_{m-j_0}\| + \bar{h}_m) + \delta). \end{aligned} \tag{4.14}$$

Thus,

$$\begin{aligned} \|F'(x^*)^{-1}(\bar{B}_m - F'(x^*))\|_F &\le \|F'(x^*)^{-1}\|_F(\|\bar{B}_m - F'(x_m)\|_F + \|F'(x_m) - F'(x^*)\|_F) \\ &\le \beta C_1(\alpha(5\epsilon + h) + \delta) < \frac{1}{2}. \end{aligned}$$

Therefore, by Dennis and Schnabel's Theorem 3.1.4 [3],

$$\|(\bar{B}_m)^{-1}\|_F \le 2\beta, \tag{4.15}$$

which shows that x_{m+1} is well defined. From (4.14) and (4.15),

$$\begin{aligned} \|x_{m+1} - x^*\| &\le \|(\bar{B}_m)^{-1}\|_F[\|F(x^*) - F(x_m) - F'(x_m)(x^* - x_m)\| \\ &\quad + \|\bar{B}_m - F'(x_m)\|_F\|x^* - x_m\|] \\ &\le 2\beta[\frac{\alpha}{2}\|x_m - x^*\| + \|\bar{B}_m - F'(x_m)\|_F]\|x^* - x_m\| \\ &\le 2\beta C_1[\alpha(\frac{9}{2}\epsilon + h) + \delta]\|x^* - x_m\| \le \frac{1}{2}\|x_m - x^*\|, \end{aligned} \tag{4.16}$$

which completes the induction step. It follows from (4.13) that $\{x_k\}$ converges to x^* at least q-linearly. From (4.6) and (4.13), for $k \ge n$, inequality (4.14) is changed to

$$\|\bar{B}_k - F'(x_k)\|_F \le C_1\alpha(\|x_k - x^*\| + \|x^* - x_{k-n+2}\| + \bar{h}_k)$$

and therefore, (4.16) is changed to

$$\|x_{k+1} - x^*\| \le 2C_1\alpha\beta(\frac{5}{2}\|x^* - x_{k-n+2}\| + \bar{h}_k)\|x_k - x^*\|. \tag{4.17}$$

Since $\{\bar{h}_k\}$ is a sub-sequence of $\{h_k\}$, $h_k \to 0$ implies $\bar{h}_k \to 0$. Therefore, by (4.17), $\{x_k\}$ converges to x^* q-superlinearly if $h_k \to 0$. By Dennis and Schnabel's Lemma 4.1.16 [3]

$$|h_k| \le C\|F(x_k)\|$$

is equivalent to

$$|h_k| \le C_2\|x_k - x^*\|,$$

where $C_2 > 0$ is a constant. Therefore, if $|h_k| \le C\|F(x_k)\|$, inequality (4.17) can be rewritten as

$$\|x_{k+1} - x^*\| \le C_3\|x^* - x_{k-n+2}\|\|x_k - x^*\| \le C_3\|x^* - x_{k-n+2}\|^2, \tag{4.18}$$

where $C_3 > 0$ is a constant, which implies that $\{x_k\}$ converges to x^* at least $(n-1)$-step q-quadratically.

THEOREM 4.3. *Assume that $F(x), F'(x)$ satisfy the hypotheses of Theorem 2.3. Let $\{x_k\}$ be generated by Algorithm 4.1 without any global strategy. Then the r-convergence order of $\{x_k\}$ is not less than τ , where τ is the unique positive root of equation*

$$t^{n-1} - t^{n-2} - 1 = 0.$$

Proof. The desired results can be easily obtained by using (4.18) and applying Ortega and Reinboldt's Theorem 9.2.9 [9].

Comparing Theorem 4.2 and Theorem 4.3 with Theorem 2.3 and Theorem 2.4, we see that the CSSCC method has a better q-convergence rate and a better r-convergence order than the SCC method.

5. Numerical results. To see how our methods work in practizce, we computed nine examples by the Finite-difference (FD) method, Broyden's method, the SCC method (Algorithm 3.3), the SSCC method with the reversed column correction order and the CSSCC method (Algorithm 4.1).For Broyden's method, the well known QR update technique is used, and for the SCC method, the SSCC method and the CSSCC method, the QR update technique mentioned in section 3 is used.

For Broyden's method, the SCC method, the SSCC method and the CSSCC method, the initial approximations to the Jacobian were computed by the FD method. The 'global strategy' used to force convergence from far away points is the line search with backtracking (see Dennis and Schnabel [3, p.126]). If direction $p_k = -B_k^{-1}F(x_k)$ is not a descent direction, then we try $-p_k$. According to Dennis and Schnabel [3, p.79], instead of using a uniform step length h_k in finite differences at each step, we use

$$h_k^j = \sqrt{macheps}\,(x_k)_j$$

to perturb each component of x , where is the machine precision. The stopping test we used is given by Dennis and Schnabel [3], i.e.,

$$\max_{1 \le i \le n} \frac{|(x_{k+1})_i - (x_k)_i|)}{max\{(|(x_k)_i|), 1\}} \le steptol,$$

and we choose $steptol = 10^{-6}$. For Algorithm 4.1, we choose $\theta = 10^{-4}$. All tests were run on the Jilin University Honeywell DPS-8 in double precision. We compare the numerical results from the five methods in Table 1, Table 2 and Table 3, where NI is the number of iterations, NF is the number of function evaluations and TIME is the CPU time in seconds. The number of function evaluations includes the number of function values needed in line searches.

As we mentioned in section 3, the SCC method with the reversed column correction order (Algorithm 3.3) may be more efficient than the SCC method with the natural column correction order (Algorithm 2.1). To see this fact in practice, we compare CPU times for solving the nine problems mentioned above from Algorithm 2.1 and Algorithm 4.1 in Table 4.

The test problems are:

1. Discrete boundary value function in[8].

TABLE 1

Algorithms	Problem 1			Problem 2			Problem 3		
	NI	NF	TIME	NI	NF	TIME	NI	NF	TIME
FD	3	52	0.577	3	52	1.391	17	290	3.026
BROYDEN	4	21	0.445	4	21	0.783	23	40	1.925
SCC	5	26	0.320	5	26	0.739	73	162	3.341
SSCC	4	24	0.403	4	24	0.784	23	62	1.924
CSSCC	4	24	0.321	4	24	0.705	23	62	1.456

TABLE 2

Algorithms	Problem 4			Problem 5			Problem 6		
	NI	NF	TIME	NI	NF	TIME	NI	NF	TIME
FD	5	86	0.878	6	103	1.295	15	168	2.397
BROYDEN	9	26	0.815	15	32	1.352	20	61	1.646
SCC	12	40	0.594	19	54	1.076	76	698	3.064
SSCC	9	34	0.744	18	52	1.697	20	85	1.392
CSSCC	9	34	0.551	18	58	1.372	19	80	0.935

2. Discrete integral equation function in [8].
3. Trigonometric function in [8].
4. Variably dimensioned function in [8].
5. Broyden tridiagonal function in [8].
6. Broyden banded function in [8].
7. Example 6.2 in [6].
8. Example 6.3 in [6].
9. Example 6.4 in [6].

The starting points for the first six examples are the standard ones which can be found in Moré Garbow and Hillstrom [8], and the starting points for the last three examples are the first ones in Li [6], i.e. $x_0 = -2$. The dimension of all test problems is 16.

From Table 1, Table 2 and Table 3, we can see the following facts:

(1). The SCC method takes less execution time than the FD method in five out of nine cases, and it takes less execution time than Broyden's method in seven out of nine cases though sometimes it takes more iterations than the FD method and Broyden's method. This is due to the savings from the QR update technique mentioned in section 3.

(2). The CSSCC method takes the least execution time in seven out of all nine cases, it takes less execution time than Broyden's method in eight cases, and it takes less execution time than the SSCC method in all nine cases.

(3). The CSSCC method takes much less number of function evaluations than the FD method in eight cases, and it takes the least function evaluation number in two cases.

(4). The CSSCC method and the SSCC method take less iterations than the SCC

TABLE 3

Algorithms	Problem 7			Problem 8			Problem 9		
	NI	NF	TIME	NI	NF	TIME	NI	NF	TIME
FD	17	297	3.002	32	598	5.927	18	315	3.373
BROYDEN	78	184	6.460	79	180	6.570	119	337	10.15
SCC	77	538	4.641	66	308	3.656	71	428	4.285
SSCC	54	199	4.704	46	164	4.036	53	174	4.650
CSSCC	54	192	4.238	33	114	2.319	33	133	2.424

TABLE 4

Algorithms	Prob.1	Prob.2	Prob.3	Prob.4	Prob.5	Prob.6	Prob.7	Prob.8	Prob.9
Alg. 2.1	0.481	0.893	3.627	0.874	1.096	3.107	6.228	10.87	6.147
Alg. 3.1	0.320	0.739	3.341	0.594	1.076	3.063	4.641	3.565	4.285

method for all nine examples.

From Table 4, we see that the SCC method with the reversed column correction order (Algorithm 3.1) always takes less execution times than the SCC method with the natural order (Algorithm 2.1). This is mainly due to the QR update technique.

6. Concluding remarks. We have given a QR update technique for the SCC method, and our analysis and our numerical results have shown that this QR update technique may gain some savings from the computational cost. We have also given a column-secant modification for the SCC method (CSSCC method). We have shown that the CSSCC method has better local convergence rates than the SCC method. Our numerical results have verified this theoretical result.

When the function evaluation is not very expensive, one may consider a variation of the SCC method or the CSSCC method such that instead of correcting just one column of the approximate Jacobian, two or more columns may be corrected at each iteration. This technique may reduce the number of iterations needed for convergence. Of course, it will increase the number of function evaluations at each iteration than the SCC and the CSSCC method.

REFERENCES

[1] C. BROYDEN, *A class of methods for solving nonlinear simultaneous equations*, Mathematics of Computation, 19 (1965), pp. 577–593.

[2] C. BROYDEN, J. DENNIS, AND J. MORÉ, *On the lodal and superlinear convergence of quasi-newton methods*, Journal of the Institute of Mathematics and Its Application, 12 (1973), pp. 223–246.

[3] J. DENNIS AND R. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[4] G. FENG AND G. LI, *The column-row update method*, Numerical Mathematics, Journal of Chinese Universities, 5 (1983).

[5] P. GILL, G. GOLUB, W. MURRAY, AND M. SAUNDERS, *Methods for modifying matrix factorizations*, Mathematics of Computation, 28 (1974), pp. 505–535.

[6] G. Li, *Successive column correction algorithms for solving sparse nonlinear systems of equations*, Mathematical Programming, 43 (1989), pp. 187–207.

[7] T. Liu and G. Li, *Column correction methods for solving systems of nonlinear equations*, Technical Report 89-13, Mathematics Institute, Jilin University, (1989).

[8] J. Moré, B. Garbow, and K. Hillstrom, *Testing unconstrained optimization software*, ACM Transaction on Mathematical software, 7 (1981), pp. 17–41.

[9] J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations*, Academic Press, New York and London, 1970.

[10] E. Polak, *A modified secant method for unconstrained minimization*, Memorandum No. ERLM373, Electronics Research Lab, College of Enginerreing, University of California (Berkeley), (1973).