INFORMATION TO USERS

This reproduction was made from a copy of a manuscript sent to us for publication and microfilming. While the most advanced technology has been used to photograph and reproduce this manuscript, the quality of the reproduction is heavily dependent upon the quality of the material submitted. Pages in any manuscript may have indistinct print. In all cases the best available copy has been filmed.

The following explanation of techniques is provided to help clarify notations which may appear on this reproduction.

1. Manuscripts may not always be complete. When it is not possible to obtain missing pages, a note appears to indicate this.

2. When copyrighted materials are removed from the manuscript, a note appears to indicate this.

3. Oversize materials (maps, drawings, and charts) are photographed by sectioning the original, beginning at the upper left hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is also filmed as one exposure and is available, for an additional charge, as a standard 35mm slide or in black and white paper format. *

4. Most photographs reproduce acceptably on positive microfilm or microfiche but lack clarity on xerographic copies made from the microfilm. For an additional charge, all photographs are available in black and white standard 35mm slide format. *

*For more information about black and white slides or enlarged paper reproductions, please contact the Dissertations Customer Services Department.

UMI Dissertation Information Service
University Microfilms International
A Bell & Howell Information Company
300 N. Zeeb Road, Ann Arbor, Michigan 48106
Li, Guangye

ALGORITHMS FOR SOLVING SPARSE NONLINEAR SYSTEMS OF EQUATIONS

Rice University

Ph.D. 1986

University Microfilms
International

300 N. Zeeb Road, Ann Arbor, MI 48106
RICE UNIVERSITY

ALGORITHMS FOR SOLVING SPARSE NONLINEAR SYSTEMS OF EQUATIONS

by

GUANGYE LI

A Thesis Submitted
In Partial Fulfillment Of The
Requirements For The Degree

DOCTOR OF PHILOSOPHY

Approved, Thesis Committee:

John E. Dennis Jr.,
Professor of Mathematical Sciences
Chairman

Richard A. Tapia,
Professor of Mathematical Sciences

Mary F. Wheeler,
Professor of Mathematical Sciences

Deane Yang,
Assistant Professor of Mathematics

Houston, Texas
May, 1986
ALGORITHMS FOR SOLVING SPARSE NONLINEAR SYSTEMS OF EQUATIONS

by Guangye Li

Abstract

In this thesis, we present four algorithms for solving sparse nonlinear systems of equations: the partitioned secant algorithm, the CM-successive displacement algorithm, the modified CM-successive displacement algorithm and the combined secant algorithm. The partitioned secant algorithm is a combination of a finite difference algorithm and a secant algorithm which requires one less function evaluation at each iteration than Curtis, Powell and Reid's algorithm (the CPR algorithm). The combined secant algorithm is a combination of the partitioned secant algorithm and Schubert's algorithm which incorporates the advantages of both algorithms by considering some special structure of the Jacobians to further reduce the number of function evaluations. The CM-successive displacement algorithm is based on Coleman and More's partitioning algorithm and a column update algorithm, and it needs only two function values at each iteration. The modified CM-successive displacement algorithm is a combination of the CM-successive displacement algorithm and
Schubert's algorithm. It also needs only two function values at each iteration, but it uses the information at every step more effectively. The locally $q$-superlinear convergence results, the $r$-convergence order estimates and the Kantorovich-type analyses show that these four algorithms have good local convergence properties. The numerical results indicate that the partitioned secant algorithm and the modified CM-successive displacement algorithm are probably more efficient than the CPR algorithm and Schubert's algorithm.

In addition to the four algorithms, we give a local convergence result for the CPR algorithm, and we sharpen error estimates and improve Kantorovich-type analyses for both Broyden's algorithm and Schubert's algorithm.
Acknowledgement

I would like to express my deepest thanks to my advisor, Professor John Dennis, for his generous support, guidance and encouragement. I would also like to thank Ann Dennis for her kindness and help. I am very grateful to Professor Richard Tapia and Professor Mary Wheeler for their assistance and advice. I would like to thank Professor Deane Yang for taking time to be on my thesis committee. I would like to express my appreciation to the Chairman of our department, Professor Chao-Cheng Wang, for his help and encouragement throughout my years at Rice.

I would especially like to thank Karen Williamson for her help in writing my thesis. I also thank Daniel Woods and Vivian Chio for their help in typing this thesis. Lastly, I would like to thank my wife Zeyun Song for her understanding and encouragement.

This work was supported by the government of the People’s Republic of China and the Rice University Mathematical Sciences Department Student Assistance Fund.
# TABLE OF CONTENTS

Chapter 1  Introduction ......................................................... 1

Chapter 2  Newton's Method and its Variations .......................... 6
  2.1. Introduction ............................................................. 6
  2.2. Newton's Method ....................................................... 9
  2.3. The Column-by-Column Finite Difference Algorithm .......... 11
  2.4. The CPR Algorithm .................................................... 14
  2.5. Broyden's Algorithm .................................................. 23
  2.6. Schubert's Algorithm ............................................... 33
  2.7. The Column-Update Quasi-Newton Algorithm .................. 38

Chapter 3  The Partitioned Secant Algorithm ......................... 42
  3.1. Introduction ............................................................ 42
  3.2. The Partitioned Secant Algorithm and its Properties .......... 43
  3.3. A Kantorovich-Type Analysis ...................................... 50
  3.4. Local Convergence Properties ..................................... 51
  3.5. Numerical Results .................................................... 57

Chapter 4  Successive Displacement Algorithms ....................... 62
  4.1. Introduction ............................................................ 62
  4.2. The CM-successive Displacement Algorithm and its Properties 63
  4.3. A Kantorovich-Type Analysis ...................................... 67
  4.4. Local Convergence Properties ..................................... 68
  4.5. The Modified CM-Successive Displacement Algorithm .......... 71
  4.6. Numerical Results .................................................... 74

Chapter 5  The Combined Secant Algorithm .............................. 76
  5.1. Introduction ............................................................ 76
  5.2. The Combined Secant Algorithm and its Properties .......... 77
  5.3. A Kantorovich-Type Analysis ...................................... 82
  5.4. Local Convergence Properties ..................................... 89

Chapter 6  Concluding Remarks ............................................. 92

Bibliography ................................................................. 95
CHAPTER 1

Introduction

This thesis is concerned with the numerical solution of nonlinear system of the equations

$$F(x) = 0,$$  \hspace{1cm} (1.1)

where $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is continuously differentiable on an open convex set $D \subset \mathbb{R}^n$.

The most popular method for solving this problem is Newton's method, which can be formulated as the iteration

$$x^{k+1} = x^k - F'(x^k)^{-1}F(x^k), \quad k=0,1,..., \hspace{1cm} (1.2)$$

where $F'(x^k)$ is the Jacobian of $F$ at $x^k$.

For many problems the Jacobian matrix is sparse. For example, most problems arising from discretizations of nonlinear differential equations have sparse Jacobians. In this thesis the emphasis will be on sparse problems.

Newton's method has good local convergence properties. However, at every iterative step we have to compute the Jacobian, and in most cases this is expensive. Therefore, we are very interested in inexpensively finding an $n \times n$ matrix $B_k$ which is a good approximation to the Jacobian $F'(x^k)$. With an approximation to the Jacobian we have the iteration

1
\[ x^{k+1} = x^k - B_k^{-1}F(x^k), \quad k=0,1,... \] (1.3)

For convenience, sometimes we rewrite (1.3) as
\[ \bar{x} = x - B^{-1}F(x), \] (1.4)
where \( x \) and \( \bar{x} \) indicate the current iterate and the new iterate respectively, and
\( B \) is an approximation to the Jacobian.

Currently, there are several ways to get \( B_k \). Here, we will mention 3 types of them.

(1). Finite difference algorithms. Usually finite difference algorithms have good convergence properties since \( B_k \) can be made to be a very good approximation to \( F'(x^k) \) by properly choosing the step size. However, finite difference algorithms are still expensive when it is expensive to compute the value of \( F(x) \). For sparse problems, if we can compute the function values element by element, then some finite difference algorithms can take advantage of the sparsity to reduce the number of function evaluations. However, in practice it is usually expensive to evaluate the function values element by element. This would certainly be true if the components of \( F(x) \) have expensive common sub-expressions. Based on this consideration, we will assume in this thesis that it is desirable to evaluate the function value as a single entity rather than to evaluate it element by element. In this case, to reduce the number of function evaluations, Curtis, Powell and Reid (1974) proposed a difference algorithm, called the CPR algorithm, which is based on a partition of the columns of the Jacobian. Coleman and Moré (1981), connected the partition
problem to a graph coloring problem and gave some partitioning algorithms which can make the number of function evaluations optimal or nearly optimal. We call the CPR algorithm based on Coleman and More's algorithm the CPR-CM algorithm.

(2). Secant (or Quasi-Newton) algorithms. This type of algorithm was introduced first by Davidon (1959) for unconstrained optimization problems, and then by Broyden (1965) for systems of equations. The most attractive advantage of Broyden's algorithm is that only one function value is needed at each iterative step.

Since Broyden's algorithm can not take advantage of the sparsity of the Jacobian, Schubert (1970) gave a sparse modification of Broyden's algorithm.


In Chapter 2 we will discuss in greater detail all the algorithms mentioned above. Also we will give a local convergence result for the CPR algorithm, and we will sharpen error estimates and improve Kantorovich-type theorems for Broyden's algorithm and Schubert's algorithm. In Chapter 3 we present the partitioned secant 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 order estimate
show that this algorithm has good local convergence properties. The numerical results indicate that the this algorithm is probably more efficient than the CPR algorithms. In Chapter 4 we present two algorithms for solving sparse nonlinear systems of equations: the CM-successive displacement algorithm and the modified CM-successive displacement algorithm. A q-superlinear convergence theorem and an r-convergence order estimate are given for both algorithms. The numerical results indicate that these two algorithms, especially the modified algorithm, are probably more efficient than the CPR algorithm and Schubert’s algorithm. In Chapter 5 we present the combined secant algorithm for solving some sparse nonlinear systems of equations. This algorithm is a combination of the partitioned secant algorithm and Schubert’s algorithm which incorporates the advantages of both algorithms by considering some special structure of the Jacobian. A q-superlinear convergence result for this algorithm is given.

In this thesis, $\mathbb{R}^n$ denotes $n$-dimensional real Euclidean space with the usual inner product $\langle x , y \rangle = x^T y$, and $L(\mathbb{R}^n)$ denotes the linear space of all real $n \times n$ matrices. We use $||.||$ to denote the $l_2$ vector norm $||x|| = \langle x , x \rangle^{\frac{1}{2}}$ or any matrix norm which is consistent with the $l_2$ norm in the sense that $||Ax|| \leq ||A|| \cdot ||x||$ for each $x \in \mathbb{R}^n$ and $A \in L(\mathbb{R}^n)$. In particular, we use $||.||_F$ to denote the Frobenius norm which is also consistent with the $l_2$ vector norm. It can be computed by
\[ \| A \|_F^2 = \sum_{i=1}^{n} \| A v_i \|^2 = tr(A^T A), \]

where \{v_1, \ldots, v_n\} is any orthonormal set in \( \mathbb{R}^n \). Moreover, we use \( N(y, \epsilon) \) to denote the set \( \{ x \in \mathbb{R}^n : \| x - y \| < \epsilon, \ y \in \mathbb{R}^n \} \), and we use \( \overline{N}(y, \epsilon) \) to denote the closure of \( N(y, \epsilon) \). For a matrix \( B \), we use \( b_{ij} \) to denote the element in the \( i \)th row and \( j \)th column, and for a vector \( v \in \mathbb{R}^n \), we use \( v_i \) to denote the \( i \)th component. We use \( \setminus \) to denote the subtraction of two sets; that is,

\[ A \setminus B = \{ v : v \in A \text{ and } v \notin B \}. \]
CHAPTER 2

Newton's Method and its Variations

2.1. Introduction.

In this chapter, we discuss Newton's method and some of its variations. In Section 2.2 we review Newton's method and its local convergence properties. In Section 2.3 we discuss the column-by-column finite difference algorithm and its local convergence properties. In Section 2.4 we discuss the CPR algorithm, and give a local convergence result for this algorithm. In Section 2.5 we discuss Broyden's algorithm and its local convergence properties, and we give a Kantorovich-type analysis for this algorithm, which is sharper than that given by Dennis(1971). In Section 2.6 we discuss Schubert's algorithm and its local convergence properties. Also, we give a Kantorovich-type analysis for this algorithm, which is sharper than that given by Marwil(1979). In Section 2.7 we discuss the column-update algorithm.

To study local convergence properties of an algorithm, it is assumed that

\[ F: \mathbb{R}^n \rightarrow \mathbb{R}^n \text{ is continuously differentiable, and there is an } x^* \in D, \text{ such that } F(x^*) = 0 \text{ and } F'(x^*) \text{ is nonsingular.} \]  \hspace{1cm} (2.1.1)

In addition, we assume that \( F' \) satisfies the following Lipschitz condition at \( x^* \):

There exists a constant \( \gamma > 0 \) such that
\[ ||F'(x) - F'(x^*)|| \leq \gamma ||x - x^*||, \quad \forall x \in D. \quad (2.1.2) \]

Sometimes we assume that \( F' \) satisfies a stronger Lipschitz condition: There exists \( \rho > 0 \) such that

\[ ||F'(x) - F'(y)|| \leq \rho ||x - y||, \quad \forall x, y \in D. \quad (2.1.3) \]

For some algorithms, we assume that \( F' \) satisfies the following Lipschitz condition: There exist \( \alpha_i > 0, i = 1, 2, \ldots, n \) such that

\[ ||(F'(x) - F'(y))e_i|| \leq \alpha_i ||x - y||, \quad \forall x, y \in D. \quad (2.1.4) \]

Let \( \alpha = (\sum_{i=1}^{n} \alpha_i^2)^{\frac{1}{2}}. \) Then, (2.1.4) implies that

\[ ||F'(x) - F'(y)|| \leq \alpha ||x - y||, \quad \forall x, y \in D. \quad (2.1.5) \]

The following lemmas are very useful for investigating an algorithm's convergence properties.

**Lemma 2.1.1.** Let \( x, x+p \in D. \) Then,

\[ F(x+p) - F(x) = \int_0^1 F'(x+tp)p \ dt. \quad (2.1.6) \]

The proof of Lemma 2.1.1 can be found in Dennis and Schnabel (1983 p.75).

**Lemma 2.1.2.** Let \( x, \bar{x} \in D. \) Given \( \epsilon > 0 \) there exists \( \delta > 0 \) such that if

\[ ||\bar{x} - x|| < \delta, \] 

\[ ||F(x) - F(\bar{x}) - F'(\bar{x})(x - \bar{x})|| \leq \epsilon ||x - \bar{x}||. \quad (2.1.7) \]

If, in addition, \( F' \) satisfies Lipschitz condition (2.1.3), then

\[ ||F(x) - F(\bar{x}) - F(\bar{x})(x - \bar{x})|| \leq \frac{L}{2} ||x - \bar{x}||^2. \quad (2.1.8) \]

The proof of this lemma can be found in Ortega and Rheinboldt (1970 p.75).
Lemma 2.1.3. Let $A, B \in L(R^n)$. If $A$ is nonsingular and $||A^{-1}(B-A)|| < 1$, then $B$ is nonsingular and

$$||B^{-1}|| \leq \frac{||A^{-1}||}{1 - ||A^{-1}(B-A)||}.$$  \hspace{1cm} (2.1.9)

The proof of Lemma 2.1.3 can be found in Dennis and Schnabel (1983 p.45).

To evaluate algorithms, the following concepts concerning the convergence rate of an algorithm are very important.

Definition 2.1.4. A sequence $\{x^k\}$ is $q$-linearly convergent to $x^*$ if $\{x^k\}$ converges to $x^*$ and there exist a constant $c \in [0, 1)$ and an integer $k_0$ such that for all $k \geq k_0$,

$$||x^{k+1} - x^*|| \leq c \ ||x^k - x^*||.$$  

Definition 2.1.5. A sequence $\{x^k\}$ is $q$-superlinearly convergent to $x^*$ if for some sequence $\{c_k\}$ that converges to 0,

$$||x^{k+1} - x^*|| \leq c_k \ ||x^k - x^*||.$$  

Definition 2.1.6. A sequence $\{x^k\}$ is convergent to $x^*$ with $q$-order at least $p$ if $\{x^k\}$ converges to $x^*$ and there exist constants $p > 1$, $c \geq 0$ and $k_0 \geq 0$ such that for all $k \geq k_0$,

$$||x^{k+1} - x^*|| \leq c \ ||x^k - x^*||^p.$$  

Definition 2.1.7. A sequence $\{x^k\}$ is convergent to $x^*$ with $r$-order at least $p$ if

$$||x^k - x^*|| \leq b_k,$$

where the scalar sequence $\{b_k\}$ is convergent to 0 with $q$-order $p$. 
A useful algorithm should at least be locally $q$-linearly convergent. A good algorithm should be locally $q$-superlinearly convergent. The following result given by Dennis and More' (1974) is very useful for investigating the $q$-superlinear convergence of an algorithm.

**Theorem 2.1.8.** Let $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ satisfy assumption (2.1.1), and let \( \{B_k\}_{k \in \mathbb{N}} \subset L(\mathbb{R}^n) \) be a sequence of nonsingular matrices. Suppose that for some $x^0 \in D$, the sequence

$$
x^{k+1} = x^k - B_k^{-1}F(x^k), \quad k = 0, 1, \ldots,
$$

remains in $D$, $x^k \neq x^*$ for $k \geq 0$, and converges to $x^*$. Then \( \{x^k\} \) converges $q$-superlinearly to $x^*$ if and only if

$$
\lim_{k \to \infty} \frac{|| (B_k - F'(x^*)(x^{k+1} - x^k)) ||}{|| x^{k+1} - x^k ||} = 0. \tag{2.1.10}
$$

2.2. Newton's Method.

Newton's method for a nonlinear systems of equations is an extension of Newton's method for one dimensional equations to $n$-dimensional problems. Recall Lemma 2.1.1, for $x, x + p \in D$, we have that

$$
F(x + p) - F(x) = \int_0^1 F'(x + tp)p \, dt.
$$

Therefore, it is reasonable to model $F(x^k + p^k)$ by the affine model

$$
M(x^k + p^k) = F(x^k) + F'(x^k)p^k.
$$

Solving the equation $M(x^k + p^k) = 0$, we have

$$
p^k = -F'(x^k)^{-1}F(x^k).
$$

Then, it is likely that $x^{k+1} = x^k + p^k$ is a better approximation to $x^*$ than $x^k$. 
This is the basic idea of Newton's method. We have the following local convergence result for Newton's method.

**Theorem 2.2.1.** Let $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ satisfy assumption (2.1.1). Then there is an open set $S$ which contains $x^*$ such that for any $x^0 \in S$ the Newton iterates are well-defined, remain in $S$ and converge $q$-superlinearly to $x^*$. If in addition, $F'$ satisfies the Lipschitz condition (2.1.2) then there is a constant $\beta$ such that

$$||x^{k+1} - x^*|| \leq \beta ||x^k - x^*||^2, \quad k = 0, 1, \cdots. \quad (2.2.2)$$

The proof of this theorem can be found in Ortega and Rheinboldt (1970, p.312).

Newton's method has several advantages. (1). Under weak assumptions, it is locally $q$-superlinearly convergent. Moreover, if Lipschitz condition (2.1.2) is satisfied, then it is $q$-quadratically convergent. (2). There exists a domain of attraction $S$. If the Newton iterates ever land in $S$, then they will remain in $S$ and eventually converge to $x^*$. (3). It is self-corrective; that is, $x^{k+1}$ only depends on $F$ and $x^k$. Therefore, the bad information from previous iterative steps can not affect the new iterate. (4). For large scale problems, the Jacobian may have some special structure such as symmetry or sparsity. The special structure of the Jacobian can be incorporated into Newton's method to reduce the cost.

Unfortunately, Newton's method also has several disadvantages. For many problems, the domain of attraction $S$ guaranteed by Theorem 2.2.1 is quite small. To guarantee the convergence, one has to choose a starting point $x^0$ sufficiently close to $x^*$. Sometimes, this is very difficult. Therefore,
globalization strategies for Newton's method are very important, and much work has been done in this area since the 1950's in this area. In recent years, globalizing Newton's method for the unconstrained minimization problem has been a very active field. Two of the most successful approaches are line searches and the model-trust region approach (see More and Sorenson (1982) or Dennis and Schnabel (1983 p.111)). These two approaches can also be extended to the problem of nonlinear system of equations (see Dennis and Schnabel (1983 p.155)).

Another disadvantage of Newton's method is that the Jacobian has to be computed at every iterative step. For many problems, it is not easy to obtain the analytic expression of the Jacobian. Moreover, the computation of the Jacobian at each step involves the evaluation of $n^2$ scalar functions, and for many functions this is very expensive. An important way to overcome this weakness is to find a matrix $B_k$ which is an approximation to $F'(x^k)$, and then use iteration (1.3) instead of (1.2). This is the topic of the following sections and chapters.

2.3. The column-by-column finite difference algorithm.

In this section, we study a finite difference algorithm for general nonlinear systems. Consider the $j$th column of $F'(x)$:

$$F'(x)e_j = \frac{\partial F(x)}{\partial x_j}, \quad j=1, 2, \ldots, n,$$

where $F(x) = (f_1(x), f_2(x), \ldots, f_n(x))^T$, and $e_j$ is the $j$th column of the $n \times n$
identity matrix. If we approximate \( \frac{\partial F(x)}{\partial x_j} \) by the finite differences

\[
\frac{F(x+he_j) - F(x)}{h}, \quad j=1, 2, \ldots, n,
\]

then we obtain a matrix \( B \), which satisfies

\[
Be_j = \frac{F(x+he_j) - F(x)}{h}, \quad j=1, 2, \ldots, n. \tag{2.3.1}
\]

This is the motivation of the column-by-column finite difference algorithm. This algorithm with a global strategy is stated below:

**Algorithm 2.3.1.** Given \( x^0 \in D \). At each iteration \( k \geq 0 \):

1. Choose a scalar \( h^k \).
2. Set

\[
b_{ij}^k = \frac{1}{h^k} e_i^T (F(x^k + h^k e_j) - F(x^k)), \tag{2.3.2}
\]

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

The column-by-column finite difference algorithm has the following local convergence property:

**Theorem 2.3.2.** Let \( F \) satisfy the assumption (2.1.1) and the \( F' \) satisfy Lipschitz condition (2.1.2). Then, there exist \( \epsilon, h > 0 \) such that if \( \{h^k\} \) is a real sequence with \( 0 < |h^k| \leq h \) and \( x^0 \in N(x^*, \epsilon) \), then the sequence \( \{x^k\} \) generated by Algorithm 2.3.1 without any global strategy is well defined and converges at
least \( q \)-linearly to \( x^* \). If, in addition,

\[
\lim_{k \to \infty} h^k = 0,
\]

then the convergence is \( q \)-superlinear. Furthermore, if there exists some constant \( c_1 \) such that

\[
|h^k| \leq c_1 ||x^k - x^*||, 
\]

or equivalently a constant \( c_2 \) such that

\[
|h^k| \leq c_2 ||F(x^k)||, 
\]

then the convergence is \( q \)-quadratic.

The proof of this theorem can be found in Dennis and Schnabel (1983 p.95). Theorem 2.2.1 shows that the column-by-column finite difference algorithm can retain the good local convergence properties of Newton's method by properly choosing the step length \( h^k \). However, in practice, we can not choose the step length too small since, as \( h^k \) becomes small, the errors from inaccuracies in the function values and cancellation errors in subtracting them become more significant. Dennis and Schnabel (1983 p.98) gave a way to choose the step length that works well in practice. They use a different \( h^k_j \) for each component of \( x^k \) instead of one uniform \( h^k \), and they choose

\[
h^k_j = \sqrt{\eta} \max\{ |x^k_j|, \text{typx}_j \} \text{sign}(x^k_j),
\]

(2.3.5)

where \( \eta \) is the relative error in computing \( F(x^k) \), and \( \text{typx}_j \) is a typical size of \( x^k_j \) provided by the user (see Dennis Schnabel (1893 p.155)). In cases when \( F(x) \) is given by a simple formula, it is reasonable that \( \eta \approx \text{macheps} \), where \( \text{macheps} \) is the machine precision.
The most attractive advantage of finite difference algorithms is that the analytic Jacobian is not needed. In addition, since $B_k$ generated by Algorithm 2.3.1 can be a very good approximation to $F'(x^k)$, the column-by-column finite difference algorithm can have almost the same local convergence properties as Newton's method. The disadvantage of the column-by-column finite difference algorithm is that $n+1$ potentially expensive values of $F(x)$ must be computed at every iterative step and that the cancellation errors we mentioned above may affect the convergence rate.

2.4. The CPR algorithm.

If $F(x)$ can be evaluated component by component, then $B_k$ can be obtained by an element-by-element finite difference algorithm. Using this approach, $B_k$ is given by

$$b_{ij}^k = \frac{1}{h^k}(f_i(x^k + h^k e_j) - f_i(x^k)).$$

The element-by-element algorithm can take advantage of sparsity to reduce the number of function evaluations. Unfortunately, for most problems, especially for large systems, it is difficult to implement the element-by-element finite difference algorithm. Moreover, for many problems it is desirable to evaluate $F(x)$ as a single entity rather than evaluate it element by element. Therefore, we would like to have some finite difference algorithms which can take advantage of the sparsity of the Jacobian to reduce the number of function evaluations and which uses only single entity function evaluations. This is the
motivation of the CPR algorithm, which was given by Curtis, Powell and Reid (1974).

Now, we derive the CPR algorithm. Let

$$J = \int_0^1 F'(x + td)dt.$$  \hspace{1cm} (2.4.1)

Then, $J \in L(R^n)$, and we have the following result:

**Lemma 2.4.1.** Let $F'$ satisfy Lipschitz condition (2.1.3). If $x + d \in D$, then

$$||J - F'(x)|| \leq \frac{\rho}{2} ||d||.$$  \hspace{1cm} (2.4.2)

**Proof.** By (2.4.1) and Lipschitz condition (2.1.3),

$$||J - F'(x)|| = ||\int_0^1 (F'(x + td) - F'(x))dt||$$

$$\leq \rho \int_0^1 ||d||t \, dt = \frac{\rho}{2} ||d||.$$  \hspace{1cm} (2.4.2)

Lemma 2.4.1 implies that if $||d||$ is small, then $J$ is a good approximation to $F'(x)$. By Lemma 2.1.1,

$$Jd = F(x + d) - F(x).$$

Therefore, the following conjecture should be reasonable. If there exist a matrix $B$ and directions $d_1, d_2, \ldots, d_p$, where $||d_i||$, $i=1, 2, \ldots, p$, are small, such that $B$ is determined uniquely by the equations

$$Bd_i = F(x + d_i) - F(x), \hspace{1cm} i=1, 2, \ldots, p,$$  \hspace{1cm} (2.4.3)

then $B$ is a good approximation to $F'(x)$. Notice that if we take $p=n$ and $d_i=he_i$, then (2.4.3) is equivalent to (2.3.1), which leads to the column-by-column finite difference algorithm. If the special structure is ignored, then $p=n$
is necessary for uniquely determining $B$. However, if the sparsity of $B$ is considered, then $p$ can be much smaller than $n$. This can be done by means of a proper partition of the columns of $B$. Following Coleman and More (1983), we give some definitions concerning a partition of the columns of the Jacobian.

**Definition 2.4.2.** A partition of the columns of a matrix $B$ is a division of the columns into groups $c_1, c_2, \ldots, c_p$ such that each column belongs to one and only one group.

**Definition 2.4.3.** A partition of the columns of a matrix $B$ is consistent with the direct determination of $B$ if whenever $b_{ij}$ is a nonzero element of $B$, then the group containing column $j$ has no other column with a nonzero element in row $i$.

As an example we consider the matrix with a tridiagonal structure

$$
\begin{bmatrix}
\times & \times & 0 & 0 & 0 & 0 \\
\times & \times & \times & 0 & 0 & 0 \\
0 & \times & \times & \times & 0 & 0 \\
0 & 0 & \times & \times & \times & 0 \\
0 & 0 & 0 & \times & \times & \times \\
0 & 0 & 0 & 0 & \times & \times \\
\end{bmatrix}.
$$

(2.4.4)

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

Given a sparse matrix $B$, let $M$ denote the set of pairs of indices $(i, j)$, where $b_{ij}$ is a structured nonzero of $B$, i.e.
\[ M = \{(i, j): \ b_{ij} \neq 0\} \] .

This is to distinguish structured nonzeros from arithmetic nonzeros.

The CPR algorithm now can be formulated as follows: for a given consistent partition of the columns of the Jacobian, obtain vectors \( d_1, d_2, \ldots, d_p \) such that \( B \) can be determined uniquely by the equations

\[ Bd_i = F(x + d_i) - F(x) = y_i \quad i = 1, 2, \ldots, p \] . \hspace{1cm} (2.4.5)

The CPR algorithm with a global strategy is given below.

**Algorithm 2.4.4.** Given a consistent partition of the columns of the Jacobian, which divides the set \( \{1, 2, \ldots, n\} \) into \( p \) subsets \( c_1, c_2, \ldots, c_p \) (for convenience, \( c_i, i = 1, 2, \ldots, p \), indicates both the sets of the columns and the sets of the indices of these columns), and given an \( x^0 \in \mathbb{R}^n \), at each iteration \( k \geq 0 \):

1. Choose a scalar \( h^k \).
2. For \( i = 1, 2, \ldots, p \), set
   \[ d_i^k = \sum_{j \in c_i} h^k e_j \] . \hspace{1cm} (2.4.6)
3. For \( i = 1, 2, \ldots, p \) and \( m \in c_i \), if \( (i, m) \in M \), then set
   \[ b_{im}^k = \frac{1}{h^k} e_i^T(F(x^k + d_i^k) - F(x^k)), \] \hspace{1cm} (2.4.7)
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.
Notice that for the CPR algorithm, the number of function evaluations at each iterative step is $p+1$.

For the consistent partition given in example (2.4.4), 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 $B$ is determined uniquely, and the number of function evaluations required at each iteration is 4.

Coleman and More (1983) associated the partition problem with a graph coloring problem and gave some partitioning algorithms which can make the number of function evaluations optimal or nearly optimal. Since the partition of the columns of the Jacobian plays an important role in the CPR algorithm, we call the CPR algorithm based on Coleman and More's algorithm the CPR-CM algorithm.

The following local convergence results for the CPR algorithm are given here for completeness. They do not appear elsewhere to our knowledge, and our proof illustrates the utility of Lemma 2.4.1. Let

$$J_i = \int_0^1 F'(x + td_i)dt,$$  \hspace{1cm} (2.4.8)

where

$$d_i = \sum_{j \in c_i} he_j.$$  \hspace{1cm} (2.4.9)

Then, we have the following results.
Lemma 2.4.5. Let $B \in L(R^n)$ satisfy (2.4.5). Then

$$B = \sum_{i=1}^{p} \sum_{j \in c_i} J_i e_j e_j^T.$$  \hfill (2.4.10)

Proof. By Lemma 2.1.1,

$$J_i d_i = y_i, \quad i = 1, \ldots, p,$$  \hfill (2.4.11)

where $y_i$ is defined by (2.4.5). Let $J_i = [J_{im}^j]$. By (2.4.8), $J_i$ has the same sparsity as the Jacobian. Therefore, for $i = 1, 2, \ldots, p$ and $m \in c_i$, if $(l,m) \in M$,

then

$$J_{im} = \frac{e_l^T y_i}{h}.$$  

Ignoring the superscript $k$ in (2.4.7), we have

$$b_{lm} = \frac{e_l^T y_i}{h} = J_{lm}^i,$$

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

$$J_i e_j = B e_j,$$  \hfill (2.4.12)

where $j \in c_i$, $i = 1, \ldots, p$. Thus, (2.4.10) follows from (2.4.12) and

$$I = \sum_{i=1}^{p} \sum_{j \in c_i} e_j e_j^T.$$

Lemma 2.4.6. Let $F'$ satisfy Lipschitz condition (2.1.4), and $B \in L(R^n)$ satisfy (2.4.5), where $d_i$ is defined by (2.4.9). Then

$$||B - F'(x)||_F \leq \frac{\sqrt{n} \alpha |h|}{2}.$$  \hfill (2.4.13)

Proof. By Lemma 2.4.5 and Lipschitz condition (2.1.4),
\[ ||F'(x) - B||^2 \]
\[ = \sum_{i=1}^{p} \sum_{j \in c_i} ||(F'(x) - B)e_j||^2 \]
\[ = \sum_{i=1}^{p} \sum_{j \in c_i} ||(F'(x) - J_i)e_j||^2 \]
\[ = \sum_{i=1}^{p} \sum_{j \in c_i} \left| \left| \int_0^1 (F'(x) - F'(x+td_i))dt \right| e_j \right| ^2 \]
\[ \leq \sum_{i=1}^{p} \sum_{j \in c_i} \left( \int_0^1 \left| (F'(x) - F'(x+td_i))e_i \right| dt \right)^2 \]
\[ \leq \sum_{i=1}^{p} \sum_{j \in c_i} \left( \frac{\alpha_j}{2} \right)^2 \]
\[ \leq \frac{nh^2}{4} \sum_{i=1}^{p} \sum_{j \in c_i} (\alpha_j)^2 \]
\[ = \frac{nh^2\alpha^2}{4} \]

Thus, (2.4.13) follows from (2.4.14).

**Theorem 2.4.7.** Let \( F \) satisfy assumption (2.1.1) and \( F' \) satisfy Lipschitz condition (2.1.4). Then, given any \( r \in (0,1) \) there exist \( \epsilon_r, h_r > 0 \) such that if \( \{h^k\} \) is a real sequence with \( 0 < |h^k| \leq h_r \) and \( x^0 \in N(x^*, \epsilon_r) \), then the sequence \( \{x^k\} \) generated by Algorithm 2.4.4 without a global strategy is well defined and converges at least \( q \)-linearly to \( x^* \) with \( ||x^{k+1} - x^*|| \leq \frac{r}{2} ||x^k - x^*|| \). If, in addition,

\[ \lim_{k \to \infty} h^k = 0, \]

then the convergence is \( q \)-superlinear. Furthermore, if there exists some
constant

c_1 such that

\[ |h^k| \leq c_1 ||x^k - x^*||, \quad (2.4.15) \]

or equivalently a constant \( c_2 \) such that

\[ |h^k| \leq c_2 ||F(x^k)||, \quad (2.4.16) \]

then the convergence is \( q \)-quadratic.

**Proof.** By (2.1.1), \( F^r(x^*)^{-1} \) exists. Thus, there exists \( \beta > 0 \) such that

\[ ||F^r(x^*)^{-1}||_F \leq \beta. \]

We choose \( \epsilon_r \) and \( h_r \) so that

\[ \epsilon_r + \sqrt{n} h_r \leq \frac{r}{\alpha \beta(1+r)}. \]

Now we show by induction that

\[ ||B_{k-1}||_F \leq (1+r) \beta, \quad k=0, 1, 2, \ldots, \quad (2.4.17) \]

and that

\[ ||x^{k+1} - x^*|| \leq r ||x^k - x^*||. \quad (2.4.18) \]

By Lemma 2.4.6,

\[ ||B_0 - F^r(x^0)||_F \leq \frac{\sqrt{n} \alpha h_r}{2}. \quad (2.4.19) \]

Using (2.4.19) and Lipschitz condition (2.1.4), we have

\[ ||F^r(x^*)^{-1}[B_0 - F^r(x^*)]||_F \]
\[ \leq ||F^r(x^*)^{-1}||_F ||[B_0 - F^r(x^0)] + [F^r(x^0) - F^r(x^*)]||_F \]
\[ \leq \beta(\frac{\sqrt{n} \alpha h_r}{2} + \alpha ||x^0 - x^*||) \]
\[ \leq \beta(\frac{\sqrt{n} \alpha h_r}{2} + \alpha \epsilon_r) \leq \frac{r}{1+r}. \]
By Lemma 2.1.3, \( B_0 \) is nonsingular and
\[
|| B_0^{-1} ||_F \leq (1+r)\beta,
\]
which proves (2.4.17) for \( k=0 \), and therefore, \( x^1 \) is well defined. Now using (2.4.19), Lemma 2.1.2 and Lipschitz condition (2.1.4), we see that
\[
\begin{align*}
|| x^1 - x^* || \\
\leq || B_0^{-1} \{ [F(x^*) - F(x^0) - F'(x^0)(x^* - x^0)] + [(F'(x^0) - B_0)(x^* - x^0)] \} || \\
\leq || B_0^{-1} ||_F \{ || F(x^*) - F(x^0) - F'(x^0)(x^* - x^0) || \\
+ || F'(x^0) - B_0 ||_F || x^* - x^0 || \} \\
\leq (1+r)\beta \left( \frac{\alpha}{2} || x^* - x^0 ||^2 + \frac{\sqrt{n} \alpha h_r}{2} || x^* - x^0 || \right) \\
\leq (1+r)\beta \left( \frac{\varepsilon_r + \sqrt{n} h_r}{2} || x^0 - x^* || \right) \\
\leq \frac{r}{2} || x^0 - x^* ||,
\end{align*}
\]
which proves (2.4.18) for \( k=0 \). The proof of the induction step is identical. By (2.4.18), we know that \( \{x^k\} \) converges at least \( q \)-linearly to \( x^* \).

Now we prove the \( q \)-superlinear convergence result. Using almost the same argument as (2.4.21), we have that
\[
\begin{align*}
|| x^{k+1} - x^* || \\
\leq \frac{1}{2}(1+r)\alpha\beta \left( || x^* - x^k || + \sqrt{n} | h_k | \right) || x^* - x^k ||.
\end{align*}
\]
Let
\[
c_k = || x^k - x^* || + \sqrt{n} | h_k |.
\]
Thus, \( \lim_{k \to \infty} c_k = 0 \) provided that \( \lim_{k \to \infty} h_k = 0 \). This shows that \( \{x^k\} \) is \( q \)-superlinearly convergent.
If

\[ |h^k| \leq c_1||x^k - x^*||, \]

then (2.4.22) implies

\[ ||x^{k+1} - x^*|| \leq \frac{1}{2}(1+r)\alpha\beta\{1 + \sqrt{n(c_1)}\}||x^* - x^k||^2. \]

This proves that \( \{x^k\} \) is \( q \)-quadratically convergent. The equivalence of (2.4.15) and (2.4.16) is given in Theorem 2.3.2.

To implement the CPR algorithm, we use different \( h^k_f \), given by (2.3.5), for each different component of \( x^k \) instead of one uniform \( h^k \).

2.5. Broyden's algorithm.

In this section, we will investigate Broyden's algorithm and its convergence behavior. The convergence theory of Broyden's algorithm is already well developed. We will give the main theorems without proofs. However, we will improve an error estimate for \( B_k \) and a Kantorovich-type analysis, and we will give the proofs for them.

To motivate Broyden's algorithm, first we consider the one dimensional secant method. Recall Newton's method for one dimensional problems:

\[ x^{k+1} = x^k - \frac{1}{f'(x^k)}f(x^k). \]  \hspace{1cm} (2.5.1)

If we approximate \( f'(x^k) \) by the secant factor

\[ b_k = \frac{f(x^k) - f(x^{k-1})}{x^k - x^{k-1}}. \]  \hspace{1cm} (2.5.2)

then we obtain the secant method:
\[ x^{k+1} = x^k - b_k^{-1} f(x^k). \]

Notice that we have already calculated the value of \( f(x^{k-1}) \) at the previous step. Therefore, we need only to evaluate one function value \( f(x^k) \) at step \( k \). The number of function evaluations for the secant method is one less than that for Newton's method at each iterative step.

We can rewrite (2.5.2) as

\[ b(x - x) = f(x) - f(x), \quad (2.5.3) \]

where \( b \) is the new secant factor. Equation (2.5.3) is called the secant equation. For one dimensional problems, \( b \) is determined uniquely by the secant equation.

Now we consider \( n \)-dimensional problems. Let

\[ J = \int_0^1 F'(x+t(x-x))dt. \quad (2.5.4) \]

By Lemma 2.1.1,

\[ J(x-x) = F(x)-F(x). \]

Since \( J \) is a good approximation to \( F'(x) \) when \( |x-x| \) is small, it is reasonable to find a matrix \( B \) to be an approximation to \( F'(x) \) which satisfies the secant equation:

\[ B(x-x) = F(x)-F(x). \quad (2.5.5) \]

Let \( s = x-x \) and \( y = F(x)-F(x) \). Then equation (2.5.5) can be rewritten as

\[ Bs = y. \quad (2.5.6) \]

This is the well known secant (or quasi-Newton) equation. However, when \( n>1 \), \( B \) is not determined uniquely by the secant equation since we have \( n^2 \) unknowns, but we have only \( n \) equations. Remember that the information we have so far is: \( x, x, F(x), F(x), \) and \( B \). We would like to use this information
as much as possible. Therefore, $\bar{B}$ should not be far away from $B$. Let

$$Q_{y,s} = \{ \hat{B} \in L(R^n) : \hat{B}s = y \} .$$ (2.5.7)

Then, we will take $\bar{B}$ to be the projection of $B$ onto $Q_{y,s}$. In other words, $\bar{B}$ is the unique solution of

$$\min_{B \in Q_{y,s}} ||\hat{B} - B||_F .$$ (2.5.8)

The solution of this problem can be obtained from the following lemma:

**Lemma 2.5.1.** Let $B \in L(R^n)$, $s$, $y \in R^n$, $s \neq 0$. Then the solution to (2.5.8) is

$$\bar{B} = B + \frac{(y - Bs)s^T}{s^Ts} .$$ (2.5.9)

The proof of the lemma can be found in Dennis and Moré (1977).

Update (2.5.9) was given by Broyden (1965). Because of (2.5.8), we say Broyden's update is a least change secant update. Now we state Broyden's algorithm with a global strategy.

**Algorithm 2.5.2.** Given $x^0 \in R^n$ and a nonsingular matrix $B_0$, at each step $k \geq 0$:

1. Solve

$$B_k s^k = -F(x^k)$$

2. Choose $x^{k+1}$ by $x^{k+1} = x^k + s^k$, or by a global strategy.

3. Check for convergence.

4. Set

$$y^k = F(x^{k+1}) - F(x^k)$$
(5). Set

\[ B_{k+1} = B_k + \frac{(y^k - B_k s^k)(s^k)^T}{(s^k)^T(s^k)}. \] (2.5.10)

For implementation, one may consider an inverse update for (2.5.10):

\[ B_{k+1}^{-1} = B_k^{-1} + \frac{(s^k - B_k^{-1} y^k)(s^k)^T B_k^{-1}}{(s^k)^T B_k^{-1} y^k}. \] (2.5.11)

(see Dennis and Moré (1977)), and Broyden's direction is given by

\[ s^k = -B_k^{-1} F(x^k). \] (2.5.12)

By using (2.5.11) and (2.5.12), the total work at each iteration only involves matrix-vector multiplications which require \( O(n^2) \) operations. One may also implement Broyden's algorithm by a sequence of QR factorizations (see Gill and Murray (1972)) which also requires \( O(n^2) \) operations.

Now we investigate the local convergence behavior of Broyden's algorithm. Dennis (1971) gave a local \( q \)-linear convergence result for Broyden's algorithm. Actually, this result can be obtained by applying a general theorem given by Broyden, Dennis and Moré (1973). We will also use this theorem to prove the local convergence results of our new algorithms given in Chapter 3, Chapter 4 and Chapter 5. To present this important theorem, we need the concept of an update function. Notice that in (2.5.10), \( B_{k+1} \) can be seen as a function of \( B_k \) and \( x^k \), where the matrices \( \{B_k\} \) lie in a set \( D_M \subset L(R^n) \) and \( F \) is defined on \( D \). Broyden's algorithm for generating \( \{B_k\} \) can be described by specifying for each \((x^k, B_k)\) a nonempty set \( U(x^k, B_k) \) of possible candidates for \( B_{k+1} \). That is,
Broyden's algorithm can be rewritten as

$$x^{k+1} = x^k - B_k^{-1}F(x^k),$$

$$B_{k+1} \in U(x^k, B_k), \quad k = 0, 1, \ldots,$$

where $U$ is a set-valued mapping whose domain (we say $\text{dom} U$) is a subset of $D \times D_M$ and whose range is contained in $D_M$. The general local convergence theorem is given below.

**Theorem 2.5.3.** Let $F$ satisfy assumption (2.1.1) and $F'$ satisfy Lipschitz condition (2.1.2). Also let $U$ be a update function for $F$ such that for all $(x, B) \in \text{dom} U$ and $\overline{B} \in U(x, B),$$$

$$||\overline{B} - F'(x^*)||$$

$$\leq [1 + \alpha_1 \sigma(x, \overline{x})]|B - F'(x^*)|| + \alpha_2 \sigma(x, \overline{x}),$$

for some constants $\alpha_1$ and $\alpha_2$, where

$$\overline{x} = x - B^{-1}F(x),$$

and

$$\sigma(x, \overline{x}) = \max\{||\overline{x} - x^*||, ||x - x^*||\}.$$  

Then there exist positive constants $\epsilon$ and $\delta$ such that if $x^0 \in D$ and $B_0 \in D_M$ satisfy $||x^0 - x^*|| < \epsilon$ and $||B_0 - F'(x^*)|| < \delta$, then iteration (2.5.13) is well defined and converges $q$-linearly to $x^*$.

Applying Theorem 2.5.3 and Theorem 2.1.8, we have the following local convergence result for Broyden's algorithm.
Theorem 2.5.4. Let $F$ satisfy assumption (2.1.1) and $F'$ satisfy Lipschitz condition (2.1.2), and let $\{x^k\}$ and $\{B_k\}$ be generated by Algorithm 2.5.2 without a global strategy. Then, there exist $\epsilon, \delta > 0$ such that if $x^0 \in D$ satisfies $||x^0 - x^*|| < \epsilon$, and a nonsingular matrix $B_0$ satisfies $||B_0 - F'(x^*)|| < \delta$, then $\{x^k\}$ converges $q$-superlinearly to $x^*$.

The proof of this theorem can be found in Broyden, Dennis and Moré (1973).

The following Kantorovich-type analysis for Broyden’s algorithm that we present is sharper than that given by Dennis (1971), and the proof of the theorem is also simpler than that given by Dennis. To develop a Kantorovich-type analysis for Broyden’s algorithm, Dennis (1971) gave an error estimate for $B_{k+1}$:

$$||B_{k+1} - F'(x^{k+1})|| \leq ||B_0 - F'(x^0)|| + \frac{3\rho}{2} \sum_{i=0}^{k} ||s^i|| .$$

Now under the same assumptions, we have a sharper error estimate for $B_{k+1}$:

Lemma 2.5.5. Let $F$ satisfy assumption (2.1.1) and $F'$ satisfy Lipschitz condition (2.1.3). Also let $\{x^k\}$ and $\{B_k\}$ be generated by Broyden’s algorithm. If $\{x_i\}_{i=0}^{k+1} \subset D$, then

$$||B_{k+1} - F'(x^{k+1})|| \leq ||B_0 - F'(x^0)|| + \rho \sum_{i=0}^{k} ||s^i|| . \quad (2.5.15)$$

Proof. Let

$$J_k = \int_{0}^{1} F'(x^k + ts^k)dt . \quad (2.5.16)$$

By Lemma 2.1.1,
\[ J_k s^k = y^k \equiv F(x^{k+1}) - F(x^k). \tag{2.5.17} \]

It follows from (2.5.10) and (2.5.17) that

\[
F'(x^{j+1}) - B_{j+1} \\
= F'(x^{j+1}) - B_j - \frac{(J_j - B_j)(s^j)(s^j)^T}{(s^j)^T(s^j)}. \\
= F'(x^{j+1}) - J_j + (J_j - B_j)[I - \frac{(s^j)(s^j)^T}{(s^j)^T(s^j)}].
\]

Hence,

\[
\| F'(x^{j+1}) - B_{j+1} \| \\
\leq \| F'(x^{j+1}) - J_j \| + \| J_j - B_j \| \tag{2.5.18} \\
\leq \| F'(x^{j+1}) - J_j \| + \| J_j - F'(x^j) \| + \| F'(x^j) - B_j \|. \\
\]

Using Lipschitz condition (2.1.3), we obtain

\[
\| F'(x^{j+1}) - J_j \| \\
= \| \int_0^1 [F'(x^j + t s^j) - F'(x^j)] dt \| \\
= \int_0^1 \rho \| x^{j+1} - (x^j + t s^j) \| dt \tag{2.5.19} \\
= \rho \| s^j \| \int_0^1 (1-t) dt = \frac{\rho}{2} \| s^j \| ,
\]

and

\[
\| J_j - F'(x^j) \| \\
= \rho \| x^{j+1} - x^j \| \int_0^1 t dt = \frac{\rho}{2} \| s^j \|. \tag{2.5.20}
\]

Substituting (2.5.19) and (2.5.20) in (2.5.18), we have

\[
\| F'(x^{j+1}) - B_{j+1} \| \leq \| F'(x^j) - B_j \| + \rho \| x^{j+1} - x^j \|. \tag{2.5.21}
\]

Applying (2.5.21) for \( j = k, k-1, \ldots, 0 \), we obtain (2.5.15).

Now using this estimate for \( B_{k+1} \), we give a Kantorovich-type analysis for Broyden's algorithm.
Theorem 2.5.6. Assume that $F'$ satisfies Lipschitz condition (2.1.3). Let $x^0 \in D$, and let $B_0$ be a nonsingular $n \times n$ matrix such that

$$||B_0 - F'(x^0)|| \leq \delta, \quad ||B_0^{-1}|| \leq \beta, \quad ||B_0^{-1}F(x^0)|| \leq \eta,$$

$$h = \frac{\rho \beta \eta}{(1 - 3\beta \delta)^2} \leq \frac{1}{6}, \quad \beta \delta < \frac{1}{3}.$$

If $S(x^0, t^*) \subset D$, where

$$t^* = \frac{1 - 3\beta \delta}{3\rho \beta} (1 - \sqrt{1 - 6h}),$$

then $\{x^k\}$, generated by Broyden's algorithm, converges to $x^*$, which is the unique root of $F(x)$ in $S(x^0, t) \cap D$, where

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

Before we give the proof of this theorem, we will give some comments on the improvement of the theorem. In the Kantorovich-type theorem given by Dennis (1971), the restriction on $h$ is that $h \leq \frac{1}{8}$, and in Theorem 2.5.6, we have that $h \leq \frac{1}{6}$. Notice that the restriction on $h$ is essentially the restriction on the value of $||F(x^0)||$. Therefore, to relax the restriction is to widen the region in which the initial guess $x^0$ can be chosen.

Proof. Consider the scalar iteration

$$t_{k+1} - t_k = \beta f(t_k), \quad t_0 = 0, \quad k = 0, 1, 2, \ldots,$$

where

$$t_{k+1} - t_k = \beta f(t_k), \quad t_0 = 0, \quad k = 0, 1, 2, \ldots, \quad (2.5.22)$$
\[ f(t) = \frac{3}{2} \rho t^2 - \left( \frac{1 - 3 \beta \delta}{\beta} \right) t + \frac{\eta}{\beta}. \]  
(2.5.23)

It is easy to show that the sequence \( \{ t_k \} \) satisfies the difference equation

\[ t_{k+1} - t_k = 3 \beta \left[ \frac{\rho}{2} (t_k - t_{k-1}) + \rho t_{k-1} + \delta \right] (t_k - t_{k-1}). \]  
(2.5.24)

Equation (2.5.24) can be rewritten as

\[ t_{k+1} - t_k = 3 \beta \left[ \frac{\rho}{2} (t_k + t_{k-1}) + \delta \right] (t_k - t_{k-1}). \]  
(2.5.25)

From (2.5.25), we see that \( \{ t_k \} \) is a strictly increasing sequence. Since \( t^* \) is the smallest root of \( f(t) \),

\[
\begin{align*}
    t^* - t_{k+1} \\
    = t^* - t_k - \beta f(t_k) \\
    = \beta \left[ f(t^*) - f(t_k) - f'(t_k)(t^* - t_k) \right] + \beta \left( f'(t_k) + \frac{1}{\beta} \right) (t^* - t_k) \\
    = \beta \left( \frac{3}{2} \rho (t^* - t_k) + 3 \delta \right) (t^* - t_k).
\end{align*}
\]

Noticing \( t_0 = 0 \), by induction, we obtain

\[ t_k \leq t^*, \quad k = 0, 1, \ldots. \]

Hence, \( \{ t_k \} \) has a limit point. By (2.5.22),

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

Now we will prove the following inequality by induction.

\[ ||x^{k+1} - x^k|| \leq t_{k+1} - t_k, \quad k = 0, 1, 2, \ldots. \]  
(2.5.26)

For \( k = 0 \), we have

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

Suppose that (2.5.26) holds for \( k = 0, 1, 2, \ldots m - 1 \). Then
\[ ||x^m - x^0|| \leq \sum_{i=0}^{m-1} ||s_i|| \leq t_m \leq t^* . \]  

(2.5.27)

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

Using Lemma 2.5.5, we have

\[ ||B_0^{-1}(B_m - B_0)|| \]
\[ \leq ||B^{-1}||(||B_m - F'(x^m)|| + ||F'(x^m) - F'(x^0)|| ||F + ||F'(x^0) - B_0||) \]
\[ \leq \beta(2\rho \sum_{i=0}^{m-1} ||x^{i+1} - x^i|| + 2\delta) \leq \beta(2\rho t^* + 2\delta) \leq \beta(\frac{2t^*}{\beta}) = \frac{2}{3} . \]

Thus, By Lemma 2.1.3,

\[ ||B_1^{-1}|| \leq \frac{\beta}{1 - 2/3} = 3\beta . \]

Hence,

\[ ||x^{m+1} - x^m|| \]
\[ \leq ||B_m^{-1}||F||F(x^m) - F(x^{m-1}) - B_{m-1}(x^m - x^{m-1})|| \]
\[ \leq 3\beta(\frac{\beta}{2} ||x^m - x^{m-1}|| + \rho \sum_{i=0}^{m-2} ||x^{i+1} - x^i|| + \delta ||x^m - x^{m-1}||) \]
\[ \leq 3\beta(\frac{\beta}{2}(t_m - t_{m-1}) + \rho t_{m-1} + \delta(t_m - t_{m-1}) = t_{m+1} - t_m . \]

This completes the induction step. By (2.5.26), it is easy to show that \( \{x^k\} \) is a Cauchy sequence so that there exists an \( x^* \in D \) such that

\[ \lim_{k \to \infty} x^k = x^* . \]

The uniqueness of \( x^* \) in \( \mathcal{S}(x^0, \bar{t}) \cap D \) can be obtained from Ortega and Rheinboldt's (1970) Theorem 12.6.4 by setting \( A(x) \equiv B_0 . \)
Broyden's algorithm has the following advantages. (1). The number of function evaluations at each iteration is only one. (2). It is q-superlinearly convergent. (3). Each iteration can be implemented by $O(n^2)$ operations if inverse update (2.5.11) or Gill and Murray's strategy is used. The weakness of Broyden's algorithm is that it is not self-corrective. That is, $B_k$ depends on not only $B_{k-1}$ but all $B_i, i=0, 1, ..., k-1$. This can be seen from estimate (2.5.15). Therefore, the bad information from the initial step may be retained until the current step. This could explain why it frequently requires more iterations than finite difference algorithms.

2.6. Schubert's algorithm.

In this section, we will investigate Schubert's algorithm and its convergence behavior. As in section 2.6, we will give the local convergence results without proofs. Also, we will sharpen an error estimate for $B_k$ and improve a Kantorovich-type analysis for Schubert's algorithm.

Broyden's update can not maintain the sparsity of $B_k$. To deal with the sparse case, Schubert (1970) proposed a sparse modification of Broyden's update. Broyden (1971) also gave this algorithm independently. In order to represent Schubert's algorithm, we introduce the following notation concerning the sparsity pattern of the Jacobian:

Definition 2.6.1. For $j=1,2,\ldots$, define the subspace $Z_j \subset \mathbb{R}^n$ determined by the sparsity pattern of the $j$th row of the Jacobian:
Define the set of matrices $Z$ that preserve the sparsity pattern of the Jacobian:

$$Z = \{ A \in L(R^n) : A^T e_j \in Z_j \text{ for } j = 1, 2, \ldots, n \}.$$ 

**Definition 2.6.2.** For $j = 1, 2, \ldots, n$, define the projection operator, $D_j \in L(R^n)$, that maps $R^n$ onto $Z_j$:

$$D_j = \text{diag} (d_{j_1}, d_{j_2}, \ldots, d_{j_n}),$$

where

$$d_{ji} = \begin{cases} 1, & \text{if } e_i \in Z_j, \\ 0, & \text{otherwise.} \end{cases}$$

For a scalar $\alpha \in R$, define the pseudo-inverse:

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

Now Schubert's update can be written as

$$\overline{B} = B + \sum_{j=1}^{n} ([s]_j^T [s]_j)^+ e_j e_j^T (y - B s)[s]_j^T,$$  \hspace{1cm} (2.6.1)

where $[s]_j = D_j s$, $s = \overline{x} - x$ and $y = F(\overline{x}) - F(x)$.

Schubert's update is also a least change update. That is, $\overline{B}$ is a projection of $B$ onto $Q_{y,s} \cap Z$. This can be seen from the following theorem.

**Theorem 2.6.3.** Let $B \in Z$; $y, s \in R^n$ with $s \neq 0$. Define $\overline{B}$ by (2.6.1). Then $\overline{B}$ is the unique solution to

$$\min \{ \| \hat{B} - B \|_F : \hat{B} \in Q_{y,s} \cap Z \}.$$  \hspace{1cm} (2.6.2)
This theorem was proved independently by Reid (1973) and Marwil (1979).

Schubert’s algorithm with a global strategy is given below:

**Algorithm 2.6.4.** Given an \( x^0 \in \mathbb{R}^n \) and a nonsingular matrix \( B_0 \) with the same sparsity as the Jacobian, at each step \( k \geq 0 \):

1. Solve
   \[
   B_k s^k = -F(x^k).
   \]
2. Choose \( x^{k+1} \) by \( x^{k+1} = x^k + s^k \), or by a global strategy.
3. Check for convergence.
4. Set
   \[
   y^k := F(x^{k+1}) - F(x^k).
   \]
5. Set
   \[
   B_{k+1} = B_k + \sum_{j=1}^{n} \left[ (s^k)_j^T (s^k)_j \right] e_j e_j^T (y - Bs)[s^k]_j^T. \quad (2.6.3)
   \]

Marwil (1979) also gave the following local convergence result for Schubert’s algorithm.

**Theorem 2.6.4.** Let \( F \) satisfy assumption (2.1.1) and \( F' \) satisfy the following Lipschitz condition: There exists \( K = (k_1, k_2, \ldots, k_n) \in \mathbb{R}^n \) with \( k_i \geq 0 \) for \( i = 1, 2, \ldots, n \), such that

\[
|| e_j^T F' (x) - F' (y) || \leq k_i || x - y ||, \quad \forall x, y \in D.
\]

Then, there exist \( \epsilon, \delta > 0 \) such that if \( x^0 \in D \) satisfies \( || x^0 - x^* || < \epsilon \), and a
nonsingular matrix $B_0$ satisfies $\|B_0 - F'(x^*)\| < \delta$, then Schubert’s algorithm converges $q$-superlinearly to $x^*$.

The following property of Schubert’s algorithm is useful for investigating its convergence properties.

**Lemma 2.6.5.** Let $B, A \in \mathbb{Z}$ and $s, y \in \mathbb{R}^n$ with $s \neq 0$. If $\overline{B}$ is defined by (2.6.1), then

$$
\| \overline{B} - A \|_F^2 \leq \| B - A \|_F^2 - \frac{\| (B - A)s \|_2^2}{\| s \|_2^2} + \sum_{j=1}^{n} (\langle s, T^*[s] \rangle + [e_j^T(y - As)]^2).
$$

(2.6.4)

The proof of this Lemma can be found in Marwil (1979).

By using Lemma 2.6.5, Marwil gave an error estimate for $B_{k+1}$ generated by Schubert’s update:

$$
\| B_{k+1} - F'(x^{k+1}) \|_F \leq \| B_0 - F'(x^0) \|_F + \frac{3\alpha}{2} \sum_{i=0}^{k} \| s^i \|.
$$

(2.6.5)

He also gave a Kantorovich-type analysis for Schubert’s algorithm. As we did for Broyden’s algorithm, we will also sharpen the Kantorovich-type analysis for Schubert’s algorithm. First, under the same assumptions, we give a sharper estimate than (2.6.5).

**Lemma 2.6.6.** Let $F$ satisfy assumption (2.1.1) and $F'$ satisfy Lipschitz condition (2.1.5). Also let $\{x^k\}$ and $\{B_k\}$ be generated by Schubert’s algorithm. If $\{x^i\}_{i=0}^{k+1} \subset D$, then
\begin{equation}
\|B_{k+1} - F'(x^{k+1})\|_F \leq \|B_0 - F'(x^0)\|_F + \alpha \sum_{i=0}^{k} \|s^i\| .
\end{equation}

\textbf{Proof.} Let \(J_k\) be defined as in 2.5.16. Then \(J_k \in \mathbb{Z}\). Applying Lemma 2.6.5 with \(B = B_{k+1}\) and \(A = J_j\), and noticing that \(y^j - J_j s^j = 0\), we have

\[\|B_{j+1} - J_j\|_F^2 \leq \|B_j - J_j\|_F^2 - \frac{\|(B_j - J_j)s^j\|^2}{\|s^j\|^2} .\]

Therefore,

\[\|B_{j+1} - J_j\|_F \leq \|B_j - J_j\|_F .\]

Using Lipschitz condition (2.1.5), we obtain

\[\|F'(x^{j+1}) - J_j\|_F \]
\[= \|(\int_0^1 F'(x^{j+1}) - F'(x^j + ts^j)dt\|_F \]
\[= \int_0^1 \alpha \|x^{j+1} - (x^j + ts^j)\| dt \]
\[= \alpha \|s^j\| \int_0^1 (1-t)dt = \frac{\alpha}{2} \|s^j\| ,\]

and

\[\|J_j - F'(x^j)\|_F \]
\[= \alpha \|x^{j+1} - x^j\| \int_0^1 tdt = \frac{\alpha}{2} \|s^j\| .\]

It follows from (2.6.7), (2.6.8) and (2.6.9) that

\[\|B_{j+1} - F'(x^{j+1})\|_F \]
\[\leq \|B_{j+1} - J_j\|_F + \|J_j - F'(x^{j+1})\|_F \]
\[\leq \|B_j - J_j\|_F + \|J_j - F'(x^{j+1})\|_F \]
\[\leq \|B_j - F'(x^j)\|_F + \|F'(x^j) - J_j\|_F + \|J_j - F'(x^{j+1})\|_F \]
\[\leq \|B_j - F'(x^j)\|_F + \alpha \|s^j\| .\]

Thus, (2.6.6) follows from (2.6.10).

Now we give a Kantorovich-type analysis for Schubert's algorithm which is
sharper than that given by Marwil (1979).

**Theorem 2.6.7.** Assume that $F'$ satisfies Lipschitz condition (2.1.5). Let $x^0 \in D$, and let $B_0$ be a nonsingular $n \times n$ matrix with the same sparsity as the Jacobian such that

$$||B_0 - F'(x^0)||_F \leq \delta, \quad ||B_0^{-1}||_F \leq \beta, \quad ||B_0^{-1}F(x^0)|| \leq \eta,$$

$$h = \frac{\alpha \beta \eta}{(1-3\beta \delta)^2} \leq \frac{1}{6}, \quad \beta \delta < \frac{1}{3}.$$

If $\mathcal{S}(x^0, t^*) \subset D$, where

$$t^* = \frac{1-3\beta \delta}{3\alpha \beta} \left(1 - \sqrt{1 - 6h} \right),$$

then $\{x^k\}$, generated by Schubert's algorithm, converges to $x^*$, which is the unique root of $F(x)$ in $\mathcal{S}(x^0, \bar{t}) \cap D$, where

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

The proof of this theorem is almost the same as that for Broyden's algorithm.

Like Broyden's algorithm, Schubert's algorithm also needs only one function value at each iteration, and it is also $q$-superlinearly convergent. However, to implement Schubert's algorithm, we can not use the inverse update or the $QR$ factorization technique to reduce the arithmetic cost from $O(n^3)$ to $O(n^2)$ as we do for Broyden's algorithm because the inverse of a sparse matrix is rarely sparse and the $QR$ factorization technique in sparse case is unavailable. On the other hand, these two techniques are not important in the sparse case.
because the factorization and the solution of a sparse linear system often
requires only $O(n)$ operations.

2.7. The column-update quasi-Newton method.

Polak (1973) proposed a successive displacement algorithm for solving
unconstrained optimization problems. Using the same idea, Feng and Li (1983)
developed a successive displacement algorithm for nonlinear systems of
equations. It is called the column-update algorithm. Using this algorithm,
columns of $B_k$ are displaced by differences successively and periodically. This
algorithm is given below:

*Algorithm 2.7.1.* Given an $x^0 \in R^n$ and a nonsingular matrix $B_0$ with the same
sparsity as the Jacobian, at the initial step:

1. Set $k = 0$, $l = 0$.

2. Solve $B_0 s^0 = -F(x^0)$.

3. Choose $x^1$ by $x^1 = x^0 + s^0$, or by a global strategy.

At each iteration $k > 0$:

1. If $l < n$, set $l = l + 1$, else set $l = 1$.

2. Choose $h^k \neq 0$.

3. If $j = l$ and $(i, l) \in M$, then set

$$b_{ij}^k = \frac{1}{h^k} e_i^T (F(x^k + h^k e_l) - F(x^k)), \tag{2.7.1}$$
otherwise set

\[ b_{ij}^k = b_{ij}^{k-1}. \]

(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.

The update of the column-update algorithm can be formulated as:

\[ B_k = B_{k-1}(I - e_i e_i^T) + \frac{q_k e_i}{h_k^k}, \tag{2.7.2} \]

\[ i_k = k \mod n, \quad k = 1, 2, \ldots, \]

where \( q_k = F(x^k + h^k e_i) - F(x^k) \).

Notice that the column-update algorithm can maintain the sparsity of \( B_k \) and that at every iterative step only two function values are needed. Moreover, the column-update algorithm has the following property:

**Theorem 2.7.2.** Let \( F \) satisfy assumption (2.1.1) and \( F' \) satisfy Lipschitz condition (2.1.3). Also let \( \{x^k\} \) and \( \{B_k\} \) be generated by the column-update algorithm with \( |h^i| \leq ||s^j||, \quad j = 1, 2, \ldots \). If \( \{x_i\}_{i=0}^{k+1} \subset D \) and \( \{x^j + h^j e_i\}_{j=0}^{k+1} \subset D \), then

\[ ||B_{k+1} - F'(x^{k+1})|| \leq \frac{3p}{2} \sum_{i=k-n+1}^{k} ||s^i||. \tag{2.7.3} \]

Comparing (2.7.3) with (2.5.15), we see that the 'tail' of the summation in (2.7.3) has been cut off, i.e. in (2.7.3) the summation has only a finite number of
terms. This means that the bad information from the initial step can no longer affect the current step when $k \geq n$. Therefore, this algorithm is 'semi-self-corrective'.

The column-update algorithm has the following local convergence property:

**Theorem 2.7.3.** Let $F$ satisfy assumption (2.1.1) and $F'$ satisfy Lipschitz condition (2.1.3). Then the column-update algorithm is locally $q$-superlinearly convergent. Moreover, the $r$-convergence order of this algorithm is not less than $r$, where $r$ is the unique positive root of

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

From Theorem 2.7.3, it can be seen that the convergence rate of the column-update algorithm depends on $n$. When $n$ is large, the convergence may be not good. To overcome this weakness, one may displace a group of columns of $B_k$ at every step instead of displacing just one column.
CHAPTER 3

The Partitioned Secant Algorithm

3.1. Introduction.

In this chapter, we propose a new algorithm, called the partitioned secant 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. The secant equation is also satisfied by the partitioned secant algorithm. Therefore, this algorithm can be seen as a combination of the CPR-CM algorithm and a secant algorithm. The partitioned secant 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 partitioned secant algorithm is probably more efficient than the CPR-CM algorithm.

The partitioned secant algorithm and some of its properties are given in Section 3.2. A Kantorovich-type analysis for the partitioned secant algorithm is given in Section 3.3. A q-superlinear convergence result and an r-convergence order estimation of the partitioned secant algorithm are given in Section 3.4. Some numerical results are given in Section 3.5.
3.2. The Partitioned Secant Algorithm and its Properties.

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

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

and

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

If \( s_j \neq 0, j = 1, \ldots, n \), then \( B \) is determined uniquely by the equations

\[
B d_1 = B (\bar{x} - (\bar{x} - g_1)) = F(\bar{x}) - F(\bar{x} - g_1) \equiv y_1,
\]

\[
\ldots \quad B d_i = B (\bar{x} - g_{i-1} - (\bar{x} - g_i)) = F(\bar{x} - g_{i-1}) - F(\bar{x} - g_i) \equiv y_i, \quad (3.2.1)
\]

\[
\ldots \quad B d_p = B (\bar{x} - g_{p-1} - x) = F(\bar{x} - g_{p-1}) - F(x) \equiv y_p.
\]

Let \( \bar{B} = [b_{lm}] \). By (3.2.1), if \( (l, m) \in M \), then

\[
b_{lm} = \frac{e_l^T y_i}{s_m}, \quad (3.2.2)
\]

where \( m \in c_i, \quad i = 1, 2, \ldots, p \).

Notice that by (3.2.1), 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 is one less than the CPR-CM algorithm. For example (2.4.4), 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 3 per iteration instead of 4 required by the CPR-CM algorithm.

Let
\[ J_i = \int_0^1 F'((\bar{x} - g_i + t d_i) dt, \quad i = 1, \ldots, p. \quad (3.2.3) \]

By Lemma 2.1.1,
\[ J_i d_i = y_i, \quad i = 1, \ldots, p. \quad (3.2.4) \]

Let \( J_i = [J_{im}]. \) Since \( J_i \) has the same sparsity as the Jacobian, by (3.2.4), we have that if \((l, m) \in M, \) then
\[ J_{im}^i = \frac{e_i^T y_i}{s_m}, \quad (3.2.5) \]

where \( m \in c_i, \quad i = 1, 2, \ldots, p. \) Comparing (3.2.5) with (3.2.2), we have
\[ J_i e_j = \bar{B} e_j, \quad (3.2.6) \]

where \( j \in c_i, \quad i = 1, \ldots, p. \) Therefore, \( \bar{B} \) can be written as
\[ \bar{B} = \sum_{i=1}^p \sum_{j \in c_i} J_i e_j e_j^T. \quad (3.2.7) \]
Now we have the following estimate for $\overline{B}$:

**Theorem 3.2.1.** Suppose $F'(x)$ satisfies Lipschitz condition (2.1.4), and $\overline{B}$ is determined by (3.2.1). If $\bar{x} \in D$, $\bar{x} - g_i \in D$, $i = 1, \ldots, p$, and $s_i \neq 0$, $i = 1, 2, \ldots, n$, then

$$
||F'(\bar{x}) - \overline{B}||_F \leq \alpha ||\bar{x} - x||.
$$

(3.2.8)

**Proof.** By (3.2.6) and (3.2.7),

$$
||F'(\bar{x}) - \overline{B}||_F^2 = \sum_{i=1}^{n} ||(F'(\bar{x}) - \overline{B})e_i||^2
$$

$$
= \sum_{i=1}^{p} \sum_{j \in \varepsilon_i} ||(F'(\bar{x}) - \overline{B})e_j||^2
$$

$$
= \sum_{i=1}^{p} \sum_{j \in \varepsilon_i} ||(F'(\bar{x}) - J_i)e_j||^2.
$$

(3.2.9)

Using (3.2.3) and Lipschitz condition (2.1.4), we obtain

$$
\sum_{j \in \varepsilon_i} ||(F'(\bar{x}) - J_i)e_j||^2
$$

$$
= \sum_{j \in \varepsilon_i} ||(F'(\bar{x}) - \int_0^1 F'(\bar{x} - g_i + t(g_i - g_{i-1}))dt)e_j||^2
$$

$$
\leq \sum_{j \in \varepsilon_i} (\alpha_j \int_0^1 ||g_i - t(g_i - g_{i-1})||dt)^2
$$

$$
\leq \sum_{j \in \varepsilon_i} \alpha_j^2 (\int_0^1 (1 - t)||g_i||dt + \int_0^1 ||g_{i-1}||t dt)^2
$$

$$
\leq \sum_{j \in \varepsilon_i} \alpha_j^2 (\frac{1}{2}||s|| + \frac{1}{2}||s||)^2 = ||s||^2 \sum_{j \in \varepsilon_i} \alpha_j^2.
$$

(3.2.10)
It follows from (3.2.9) and (3.2.10) that

$$||F'(x) - B||^2 \leq ||s||^2 \sum_{i=1}^{p} \sum_{j \in c_i} \alpha_j^2 = \alpha^2 ||s||^2. \quad (3.2.11)$$

Then (3.2.8) follows from (3.2.11).

In (3.2.1), to determine $B$ uniquely, we assume that $s_j \neq 0$, $j = 1, ..., n$. 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 $B$ can not be determined uniquely by (3.2.1). In this case, let

$$\Omega_1 = \{i \in \{1, 2, ..., n\}: \ s_i \neq 0\},$$

and let

$$\Omega_2 = \{1, 2, ..., n\} \setminus \Omega_1.$$

Now we deal with the general case in such a way that if $j \in \Omega_1$, then the $j$th column of $B$ is determined uniquely by (3.2.1). If $j \in \Omega_2$, then we let the $j$th column of $B$ be equal to the $j$th column of $B$. In practice, if $|s_j|$ is too close to zero the cancellation errors will become significant. Therefore, there should be a lower bound $\theta$ for $|s_j|$. Now the partitioned secant algorithm with a global strategy can be stated as follows:

**Algorithm 3.2.2.** Given a consistent partition of the columns of the Jacobian, which divides the set $\{1, 2, ..., n\}$ into $p$ subsets $c_1, c_2, ..., c_p$, and given $x^{-1}, x^0 \in \mathbb{R}^n$ such that $s_i^{-1} \equiv x_i^0 - x_i^{-1} \neq 0$, $i = 1, 2, \ldots, n$, at each step $k \geq 0$: 
(1). Set

\[ g_i^{k-1} = \sum_{j=1}^{i} g_{i}^{k-1} e_i , \quad i=1, 2, \ldots, p-1, \quad g_0^{k-1} = 0 . \]

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

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

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

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

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

\[ b_{lm}^k = \frac{e_l^T y_i^{k-1}}{s_m^{k-1}} , \quad (3.2.12) \]

otherwise set

\[ b_{lm}^k = b_{lm}^{k-1} , \]

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.

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

\[ \bar{B} = B \sum_{j \in \Omega_2} e_j e_j^T + \sum_{i=1}^{p} \sum_{j \in c_i \cap \Omega_1} J_i e_j e_j^T , \quad (3.2.13) \]

where \( J_i \) is defined by (2.4.8).
The following result shows that the partitioned secant algorithm is a secant algorithm.

Lemma 3.2.3. $B$ satisfies the secant equations

$$
B d_i = y_i , \quad i = 1, \ldots, p ,
$$

and (3.2.14) implies

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

The proof of Lemma 3.2.3 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

$$
J_t^{k+1} = \int_0^1 F'(x^{k+1} - g_i^k + t d_i^k) dt , \quad i = 1, \ldots, p .
$$

Then by (3.2.6),

$$
J_t^{k+1} e_j = B_{k+1} e_j ,
$$

where $j \in c_i , \quad i = 1, 2, \ldots, p$.

Theorem 3.2.4. Assume that $F'$ satisfies Lipschitz condition (2.1.4). Let $\{x^j\}_{j=0}^{k+1}$ and $\{B_j\}_{j=0}^{k+1}$ be generated by Algorithm 3.2.2. Suppose that $\{x^{j+1} - g_i^j , \quad i = 0, 1, 2, \ldots, p\}_{j=0}^k \subset D$. If $s_i^k = 0$ appears consecutively in at most $m$ steps for any specific $1 \leq i \leq n$, then for $k \geq m$,
\[ \| F'(x^{k+1}) - B_{k+1} \|_F \leq \alpha \sum_{j=k-m}^{k} \| x^{j+1} - x^j \|. \quad (3.2.17) \]

**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 \( s_i^{k-j} \neq 0 \). 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,

\[ B_{k-j(k,i)+1} e_i = J_{k-j(k,i)+1} e_i, \]

by (3.2.16). Therefore,

\[ \| (F'(x^{k+1}) - B_{k+1}) e_i \| \]
\[ = \| (F'(x^{k+1}) - B_{k-j(k,i)+1}) e_i \| \]
\[ \leq \| (F'(x^{k+1}) - F'(x^{k-j(k,i)+1})) e_i \| + \| (F'(x^{k-j(k,i)+1}) - B_{k-j(k,i)+1}) e_i \| \]
\[ = \| (F'(x^{k+1}) - F'(x^{k-j(k,i)+1})) e_i \| + \| (F'(x^{k-j(k,i)+1}) - J_{k-j(k,i)+1}) e_i \| \]
\[ \leq \alpha_i \| x^{k+1} - x^{k-j(k,i)+1} \| + \alpha_i \| x^{k-j(k,i)+1} - x^{k-j(k,i)} \| \]
\[ \leq \alpha_i \sum_{l=k-j(k,i)}^{k} \| x^{l+1} - x^l \| \]
\[ \leq \alpha_i \sum_{l=k-m}^{k} \| x^{l+1} - x^l \|. \]

Hence,

\[ \| F'(x^{k+1}) - B_{k+1} \|_F^2 = \sum_{i=1}^{n} \| (F'(x^{k+1}) - B_{k+1}) e_i \|^2 \]
\[ \leq (\sum_{j=k-m}^{k} \| x^{j+1} - x^j \|)^2 \sum_{i=1}^{n} \alpha_i^2. \quad (3.2.18) \]
Then, (3.2.17) follows from (3.2.18).

3.3. A Kantorovich-Type Analysis.

Theorem 3.3.1. Assume that $F'$ satisfies Lipschitz condition (2.1.4) and that 
\{x^k\} and \{B_k\} are generated by Algorithm 3.2.2 with \[ ||x^{-1}-x^0|| \leq \delta \ . \] If \[ \{x^{j+1}-g_i^j, i = 0, 1, ..., p\} \_{j=0}^k \subset D, \] then
\[ ||F'(x^{k+1})-B_{k+1}||_F \leq \alpha \sum_{j=0}^k ||x^{j+1}-x^j|| + \alpha \delta \ . \quad (3.3.1) \]

Proof. Inequality (3.3.1) can be obtained immediately by setting $m = k+1$ in (3.2.17).

Theorem 3.3.2. Let $F'$ satisfy Lipschitz condition (2.1.4). 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
\[ ||x^{-1}-x^0|| \leq \delta, \quad ||B_0^{-1}||_F \leq \beta, \quad ||B_0^{-1}F(x^0)|| \leq \eta, \]
\[ h = \frac{\alpha^2 \beta \eta}{(1 - 3\alpha \beta \delta)^2} \leq \frac{1}{6}, \quad (3.3.2) \]
and
\[ \alpha \beta \delta < \frac{1}{3} . \]

If $\bar{S}(x^0, 2t^*) \subset D$, where
\[ t^* = \frac{1 - 3\alpha \beta \delta}{3\alpha \beta} \left(1 - \sqrt{1 - 6h}\right) , \quad (3.3.3) \]
then \{x^k\}, generated by Algorithm 3.2.2 without any global strategy, converges to $x^*$, which is the unique root of $F(x)$ in $\bar{S}(x^0, \bar{t}) \cap D$, where
\[
\bar{t} = \frac{1 - \alpha \beta \delta}{\alpha \beta} \left( 1 + \left( 1 - \frac{2 \alpha^2 \beta \eta}{(1 - \alpha \beta \delta)^2} \right)^{\frac{1}{2}} \right).
\]

**Proof.** The proof of this theorem is almost the same as that of Theorem 2.5.6.

However, when we prove the inequality
\[
||x^{k+1} - x^k|| \leq t_{k+1} - t_k, \quad k = 0, 1, 2, \ldots
\]
(3.3.4)
in the induction step \(k = m\), we must prove that
\[
\{x^m - g_i^{m-1}, i = 1, \ldots, p\} \in \mathcal{S}(x^0, 2t^*).
\]
(3.3.5)
Notice that from (2.5.27), it follows that
\[
||x^m - g_i^{m-1} - x^0|| \leq ||x^m - x^0|| + ||g_i^{m-1}||
\leq 2||x^m - x^0|| \leq 2t_m \leq 2t^*.
\]
(3.3.6)
Then, (3.3.5) follows from (3.3.6).

### 3.4. Local Convergence Properties.

**Theorem 3.4.1.** Assume that \(F: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n\) satisfies (2.1.1) and that \(F'\) satisfies Lipschitz condition (2.1.4). Let \(\{x^k\}\) be generated by Algorithm 3.2.2. without any global strategy. Then there exist \(\epsilon, \delta > 0\) such that if \(x^{-1}, x^0 \in D\) satisfy
\[
||x^0 - x^*|| < \epsilon, \quad ||x^{-1} - x^0|| \leq \delta,
\]
then \(\{x^k\}\) is well defined and converges \(q\)-superlinearly to \(x^*\).

**Proof.** Notice that we can choose \(\epsilon\) small enough so that \(||B_0^{-1}F(x^0)||\) is also small such that \(h < \frac{1}{6}\) and that \(\mathcal{S}(x^0, 2t^*) \subset D\), where \(h\) and \(t^*\) are defined by (3.3.2) and (3.3.3) respectively. When \(\delta\) is small enough, we also have
\[ \alpha \beta \delta < \frac{1}{3} , \]

where \( \beta \) is defined in Theorem 3.3.2. Therefore, by Theorem 3.3.2,

\[ \{ x^{k+1} + g^k, i=1,2,\ldots,p \} \subset D , \quad k=0,1,\ldots . \]

Thus, from (3.2.13) and the proof of Theorem 3.2.1, we have

\[
\begin{align*}
&\| F'(x^*) - \overline{B} \|_F^2 \\
= &\sum_{i \in \Omega_1} \| (F'(x^*) - \overline{B})e_i \|^2 + \sum_{i \in \Omega_2} \| (F'(x^*) - B)e_i \|^2 \\
= &\sum_{i=1}^p \sum_{j \in \xi \cap \Omega_1} \| (F'(x^*) - J_i)e_j \|^2 + \sum_{i \in \Omega_2} \| (F'(x^*) - B)e_i \|^2 \\
= &\sum_{i=1}^p \sum_{j \in \xi \cap \Omega_1} \left[ \int_0^1 (F'(x^*) - F'(x - g_i + t(g_i - g_{i-1})))e_i \, dt \right]^2 \\
&+ \sum_{i \in \Omega_2} \| (F'(x^*) - B)e_i \|^2 \\
\leq &\alpha^2 (\| x^* - \overline{x} \| + \| \overline{x} - x \|)^2 + \| F'(x^*) - B \|_F^2 \\
\leq &\alpha^2 (2\| x^* - \overline{x} \| + \| x^* - x \|)^2 + \| F'(x^*) - B \|_F^2 .
\end{align*}
\]

Therefore,

\[
\| F'(x^*) - \overline{B} \|_F \leq \| F'(x^*) - B \|_F + 3\alpha \sigma(x, \overline{x}),
\]

where

\[
\sigma(x, \overline{x}) = \max\{ \| \overline{x} - x^* \|, \| x - x^* \| \}.
\]

Notice that by Theorem 3.2.1 and Lipschitz condition (2.1.4),
\[ ||F'(x^*) - B_0||_F \]
\[ \leq ||F'(x^*) - F'(x^0)||_F + ||F'(x^0) - B_0||_F \]
\[ \leq \alpha ||x^* - x^0|| + \alpha ||x^0 - x^{-1}|| \]
\[ \leq \alpha(\epsilon + \delta) \]

Thus, by Theorem 2.5.3, we know that \( \{x^k\} \) converges at least \( q \)-linearly to \( x^* \).

According to Theorem 2.1.8, to get \( q \)-superlinear convergence, we need only to prove that

\[
\lim_{k \to \infty} \frac{||(B_k - F'(x^*))(x^{k+1} - x^k)||}{||x^{k+1} - x^k||} = 0 .
\] (3.4.2)

If for all \( 1 \leq i \leq n \), \( s_i^k = 0 \) appears consecutively in at most \( m \) steps, then by Theorem 3.2.4, then it is easy to show that

\[
\lim_{k \to \infty} ||B_k - F'(x^*)||_F = 0 .
\] (3.4.3)

Thus, (3.4.2) follows immediately from (3.4.3).

Otherwise, let

\[
A_2 = \{i \in \{1, 2, ..., n\} : \text{For any } k > 0, \text{ there exists at least one integer } m > k \text{ such that } s_i^m \neq 0 \} ,
\]

and let \( A_1 = \{1, ..., n\} \setminus A_2 \). 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,
\[ \sum_{i \in A_1} (B_k - F'(x^*)) e_i e_i^T (x^{k+1} - x^k) = 0, \]  

for \( k > k_0 \). Now we show that

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

(3.4.5)

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

\[ \| x^k - x^* \| < \frac{\epsilon}{3\alpha}, \quad \forall \ k > K. \]

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 \( k > K \) and \( i \in A_2 \), define

\[ j(k,i) = \min \{ j : s_i^{k-j} \neq 0 \}. \]

Then \( k - j(i,k) > K \). Let \( i \in \mathcal{c}_l \), \( 1 \leq l \leq p \), we have that

\[ B_k e_i = B_{k-j(i,k)+1} e_i = J_{l}^{k-j(i,k)+1} e_i. \]

Thus, by Lipschitz condition (2.1.4),

\[ \| (B_k - F'(x^*)) e_i \|^2 \]

\[ = \| (J_{l}^{k-j(i,k)+1} - F'(x^*)) e_i \|^2 \]

\[ = \int_0^1 \| F'(x^{k-j(i,k)+1}) - g_i^{k-j(i,k)} + t(g_{i-1}^{k-j(i,k)} - g_i^{k-j(i,k)}) - F'(x^*) e_i \| dt \|

\[ \leq (\int_0^1 \| x^{k-j(i,k)+1} - g_i^{k-j(i,k)} + t(g_{i-1}^{k-j(i,k)} - g_i^{k-j(i,k)}) - x^* \| dt)^2 \]

\[ \leq \alpha_i^2 (\| x^{k-j(i,k)+1} - x^* \| + \frac{1}{2} \| g_i^{k-j(i,k)} \| + \frac{1}{2} \| g_{i-1}^{k-j(i,k)} \|)^2 \]
\[ \leq \alpha_i^2 \left( \| x^{k-j(k,i)+1} - x^* \| + \| x^{k-j(k,i)+1} - x^{k-j(k,i)} \| \right)^2 \]

\[ \leq \alpha_i^2 \left( 2 \| x^{k-j(k,i)+1} - x^* \| + \| x^{k-j(k,i)} - x^* \| \right)^2 \]

\[ < \alpha_i^2 \left( \frac{\epsilon}{3\alpha} + \frac{\epsilon}{3\alpha} \right)^2 = \alpha_i^2 \frac{\epsilon^2}{\alpha^2} . \]

Then

\[ \| \sum_{i \in A_2} (B_k - F'(x^*))e_i e_i^T \|_F^2 \]

\[ = \sum_{i \in A_2} \| (B_k - F'(x^*))e_i \|^2 \]

\[ < \frac{\epsilon^2}{\alpha^2} \sum_{i \in A_2} \alpha_i^2 \leq \epsilon^2 . \]

Therefore,

\[ \| \sum_{i \in A_2} (B_k - F'(x^*))e_i e_i^T \|_F < \epsilon . \]

This completes the proof of (3.4.5).

By (3.4.4) and (3.4.5),

\[ \lim_{k \to \infty} \frac{\| (B_k - F'(x^*)) (x^{k+1} - x^k) \|}{\| x^{k+1} - x^k \|} \]

\[ = \lim_{k \to \infty} \frac{\| \sum_{i \in A_2} (B_k - F'(x^*))e_i e_i^T (x^{k+1} - x^k) \|}{\| x^{k+1} - x^k \|} \]

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

**Theorem 3.4.2.** Assume that $F$, $x^{-1}$, $x^0$ and \{ $x^k$ \} satisfy the hypotheses of Theorem 3.4.1. If, for any $1 \leq i \leq n$, $e_i^k = 0$ appears consecutively in at most $m$
steps, then the r-convergence order is not less than \( \tau \), where \( \tau \) is the unique positive root of

\[
0 \leq t^{m+2} - t^{m+1} - 1.
\]

In particular, if \( s_i^k \neq 0 \), \( i = 1, \ldots, n \), \( k = 1, 2, 3, \ldots \), then \( \tau = \frac{1 + \sqrt{5}}{2} \approx 1.618 \).

**Proof.** Notice that (3.4.3) implies that there exist \( k_0 \) and \( \beta > 0 \) such that

\[ ||B_k^{-1}|| \leq \beta \] for all \( k \geq k_0 \). Thus, by Theorem 3.2.4,

\[
||x^{k+1} - x^*|| = ||x^{k} - x^* - B_k^{-1}F(x^k)|| \leq \beta \left( \frac{3}{2} \alpha ||x^{k} - x^*|| + \alpha \sum_{j=k-m-1}^{k-1} ||x^{j+1} - x^{j}|| \right) \leq \frac{5}{2} \alpha \beta \left( \sum_{j=k-m-1}^{k} ||x^{j} - x^*|| \right) ||x^{k} - x^*|| .
\]

Let \( \varepsilon_j = \frac{5}{2} (m + 2) \alpha \beta ||x^{j} - x^*|| \). Then (3.4.7) can be rewritten as

\[
\varepsilon_{k+1} \leq \varepsilon_k \sum_{j=k-m-1}^{k} \frac{1}{m+2} \varepsilon_j .
\]

Since \( \varepsilon_k \to 0 \), there exist a \( k_1 \) and an \( \varepsilon \) such that for all \( k \geq \max \{k_1, m+1\} \), \( \varepsilon_k \leq \varepsilon < 1 \). Therefore,

\[
\varepsilon_{k_1+m+2} \leq \varepsilon_{k_1+m+1} \sum_{j=k_1}^{k_1+m+1} \frac{1}{m+2} \varepsilon_j \leq \varepsilon^2 ,
\]
\[
\varepsilon_{k_1+m+3} \leq \varepsilon_{k_1+m+2} \sum_{j=k_1+1}^{k_1+m+2} \frac{1}{m+2} \varepsilon_j \leq \varepsilon^3.
\]

By induction, it is easy to show that

\[
\varepsilon_{k_1+i} \leq \varepsilon^{\mu_i}, \quad i = 0, 1, \ldots,
\]  

(3.4.8)

where

\[
\mu_{i+1} = \mu_i + \mu_{i-m-1}, \quad i = m+1, m+2, \ldots,
\]

\[
\mu_0 = \mu_1 = \ldots = \mu_{m+1} = 1.
\]

Next, we show that \( \mu_i \) satisfies

\[
\mu_i \geq \Theta r^i, \quad i = 0, 1, \ldots,
\]

(3.4.9)

where \( \Theta = r^{(m+1)} \). In fact, \( r > 1 \) since \( r \) is the unique positive root of (3.4.6).

So, (3.4.9) holds for \( i = 0, 1, \ldots, m+1 \). Notice that \( r^{-1} + r^{-m-2} = 1 \). Therefore, if (3.4.9) is valid up to some \( i \geq m+1 \), then

\[
\mu_{i+1} = \mu_i + \mu_{i-m-1} \geq \Theta r^i + \Theta r^{i-m-1}
\]

\[
= \Theta r^{i+1}(r^{-1} + r^{-m-2}) = \Theta r^{i+1},
\]

which completes the induction.

From (3.4.8) and (3.4.9), it follows that

\[
||x^{k_1+i} - x^*|| \leq \frac{1}{\frac{5}{2}(m+2) \alpha \beta} \varepsilon^{\Theta r^i}.
\]

(3.4.10)

Hence, the \( r \)-convergence order of Algorithm 3.2.2 is not less than \( r \).

3.5. Numerical Results.

We computed some examples with tridiagonal Jacobians by the CPR algorithm and Algorithm 3.2.2. In this section, we compare the numerical
results from the two algorithms. The global strategy we used in computing the
examples is the line search with backtracking strategy (see Dennis and Schnabel
(1983, p. 126)). 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

\[
\max_{1 \leq i \leq n} \frac{|x_i^{k+1} - x_i^k|}{\max\{|x_i^{k+1}|, \text{typ}x_i\}} \leq \epsilon, \tag{3.5.1}
\]

and we choose \( \text{typ}x_i = 10^{-8} \) and \( \epsilon = 10^{-5} \). For the lower bound of \( |s_j| \), we
choose

\[
\theta = \sqrt{\text{macheps}} \ ||s||.
\]

We used double precision, and the machine precision is 2.22\(d\)-16.

Example 3.5.1 is new, and it can be seen to be an extension of the
Rosenbrock (1962) function (also see Moré, Garbow and Hillstrom (1981)) to
nonlinear system of equations with tridiagonal structure. Example 3.5.2 was
given by Broyden (1965) (also see Moré, Garbow and Hillstrom (1981)). Example
3.5.3 was given by Moré and Cosnard (1979) (also see Moré, Garbow and
Hillstrom (1981)).

The results are shown in the tables below, 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. ZR is the number of the iterations that there exists an
integer \( j \) such that \( |s_j| < \theta \). \( x_0 \) is the initial guess.
Example 3.5.1.

\[ f_1(x) = 8(x_1-x_2^2), \]
\[ f_j(x) = 16x_j(x_j^2 - x_{j-1}) - 2(1 - x_j) + 8(x_j - x_{j+1}^2), \quad j = 2, \ldots, n-1, \]
\[ f_n(x) = 16x_n(x_n^2 - x_{n-1}) - 2(1 - x_n), \]

\[ n = 9 \]

\[ x_1 = (-1, -1, \ldots, -1)^T, \quad x_2 = (-0.5, -0.5, \ldots, -0.5)^T, \quad x_3 = (2, 2, \ldots, 2)^T. \]

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>( x_0 = x_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IT</td>
</tr>
<tr>
<td>CPR Alg. 3.2.2</td>
<td>22</td>
</tr>
<tr>
<td>CPR Alg. 3.2.2</td>
<td>24</td>
</tr>
</tbody>
</table>

*Table 3.5.1 (a).*

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>( x_0 = x_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IT</td>
</tr>
<tr>
<td>CPR Alg. 3.2.2</td>
<td>22</td>
</tr>
<tr>
<td>CPR Alg. 3.2.2</td>
<td>22</td>
</tr>
</tbody>
</table>

*Table 3.5.1 (b).*

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>( x_0 = x_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IT</td>
</tr>
<tr>
<td>CPR Alg. 3.2.2</td>
<td>8</td>
</tr>
<tr>
<td>CPR Alg. 3.2.2</td>
<td>10</td>
</tr>
</tbody>
</table>

*Table 3.5.1 (c).*
Example 3.5.2 (Broyden tridiagonal function).

\[ f_i(x) = (3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1, \]

\[ x_0 = x_{n+1} = 0, \]

\[ n = 9, \]

\[ x_1 = (-1, -1, \ldots, -1)^T, \quad x_2 = (-0.3, 0.3, \ldots, -0.3, 0.3)^T, \]

\[ x_3 = (-10, -10, \ldots, -10)^T. \]

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>( x_0=x_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IT</td>
</tr>
<tr>
<td>CPR Alg. 3.2.2</td>
<td>5</td>
</tr>
<tr>
<td>CPR Alg. 3.2.2</td>
<td>6</td>
</tr>
</tbody>
</table>

*Table 3.5.2 (a).*

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>( x_0=x_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IT</td>
</tr>
<tr>
<td>CPR Alg. 3.2.2</td>
<td>5</td>
</tr>
<tr>
<td>CPR Alg. 3.2.2</td>
<td>6</td>
</tr>
</tbody>
</table>

*Table 3.5.2 (b).*

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>( x_0=x_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IT</td>
</tr>
<tr>
<td>CPR Alg. 3.2.2</td>
<td>8</td>
</tr>
<tr>
<td>CPR Alg. 3.2.2</td>
<td>10</td>
</tr>
</tbody>
</table>

*Table 3.5.2 (c).*
Example 3.5.3 (Discrete boundary value function).

\[ f_i(x) = 2x_i - x_{i-1} - x_{i+1} + \frac{h^2}{2} (x_i + t_i + 1)^3 \]

\[ h = \frac{1}{n+1}, \quad t_i = ih, \quad x_0 = x_{n+1} = 0. \]

\[ n = 9, \]

\[ x1 = (\eta_j)^T, \quad \eta_j = t_j(t_j - 1), \quad x2 = (-1, -1, ..., -1)^T, \]

\[ x3 = (10, 10, ..., 10)^T. \]

\[ \begin{array}{|c|c|c|c|c|c|} \hline \text{Algorithms} & \text{\( x0=x1 \)} \\ \cline{2-6} & IT & NF & LN & ND & ZR \\ \hline \text{CPR} & 3 & 12 & 0 & 0 & 0 \\ \text{Alg. 3.2.2} & 4 & 13 & 0 & 0 & 0 \\ \hline \end{array} \]

Table 3.5.3 (a).

\[ \begin{array}{|c|c|c|c|c|c|} \hline \text{Algorithms} & \text{\( x0=x2 \)} \\ \cline{2-6} & IT & NF & LN & ND & ZR \\ \hline \text{CPR} & 4 & 16 & 0 & 0 & 0 \\ \text{Alg. 3.2.2} & 5 & 16 & 0 & 0 & 0 \\ \hline \end{array} \]

Table 3.5.3 (b).

\[ \begin{array}{|c|c|c|c|c|c|} \hline \text{Algorithms} & \text{\( x0=x3 \)} \\ \cline{2-6} & IT & NF & LN & ND & ZR \\ \hline \text{CPR} & 8 & 32 & 0 & 0 & 0 \\ \text{Alg. 3.2.2} & 10 & 31 & 0 & 0 & 0 \\ \hline \end{array} \]

Table 3.5.3 (c).
CHAPTER 4

Successive Displacement Algorithms

4.1. Introduction.

In this chapter, we propose two algorithms: the CM-successive displacement algorithm and the modified CM-successive displacement algorithm. The former is based on Coleman and More's algorithm and the column-update algorithm. The latter is a combination of the CM-successive displacement algorithm and Schubert's algorithm. Both algorithms need only two function values at each iterative step. Our numerical results show that the new algorithms, especially the modified one, are probably more efficient than the CPR algorithm and Schubert's algorithm.

The CM-successive displacement algorithm is given in Section 4.2. A Kantorovich-type analysis for this algorithm is given in Section 4.3. A q-superlinear convergence result and an r-convergence order estimation of the CM-successive displacement algorithm are given in Section 4.4. The modified CM-successive displacement algorithm is given in Section 4.5. Some numerical results are given in Section 4.6.
4.2. The CM-successive Displacement Algorithm and its Properties.

Given a consistent partition of the columns of the Jacobian, which divides the set \{1, 2, ..., n\} into \(p\) subsets \(c_1, c_2, ..., c_p\), let

\[
d^k = \sum_{j \in c_k} h^k e_j,
\]

where

\[
i_k = k \pmod{p}, \quad k = 1, 2, ..., \]

and let

\[
y^k = F(x^k + d^k) - F(x^k).
\]

The CM-successive displacement algorithm can be formulated as follows: If \(k \leq p\), then for \(j \in c_k\), the \(j\)th column of \(B_k\) is determined uniquely by the equations

\[
B_k d^k = y^k,
\]

and the other columns of \(B_k\) are equal to the corresponding columns of \(B_{k-1}\). If \(k > p\), the columns of \(B_k\) are displaced as described above successively and periodically. In other words, for \(j \in c_i\), the \(j\)th column of \(B_k\) is determined uniquely by (4.2.3), and the other columns of \(B_k\) are equal to the corresponding columns of \(B_{k-1}\).

For example (2.4.4), at the first iteration we displace the first group \(c_1 = \{1, 4\}\). At the second iteration we displace the second group \(c_2 = \{2, 4\}\). At the third iteration we displace the third group \(c_3 = \{3, 6\}\), and then we displace the three groups successively and periodically.
The CM-successive displacement algorithm with a global strategy is given below.

Algorithm 4.2.1. Given a consistent partition of the columns of the Jacobian, which divides the set \( \{1, 2, \ldots, n\} \) into \( p \) subsets \( c_1, c_2, \ldots, c_p \), and given an \( x^0 \in \mathbb{R}^n \) and a nonsingular matrix \( B_0 \), which has the same sparsity as the Jacobian, at the initial step:

1. Set \( k = 0, l = 0 \).
2. Solve \( B_0 s^0 = -F(x^0) \).
3. Choose \( x^1 \) by \( x^1 = x^0 + s^0 \), or by a global strategy.

At each iteration \( k > 0 \):

1. Choose a scalar \( h^k \).
2. If \( l < p \), then set \( l = l + 1 \), otherwise set \( l = 1 \).
3. Set
   \[
   d^k = \sum_{j \in c_l} h^k e_j.
   \]
4. If \( j \in c_l \) and \( (i, j) \in M \), then set
   \[
   b^k_{ij} = \frac{1}{h^k} e_i^T(F(x^k + d^k) - F(x^k)),
   \]
   otherwise set
   \[
   b^k_{ij} = b^{k-1}_{ij},
   \]
   where \( B_k = [b^k_{ij}] \).
(5). Solve $B_k s^k = -F(x^k)$.

(6). Choose $x^{k+1}$ by $x^{k+1} = x^k + s^k$, or by a global strategy.

(7). Check for convergence.

Let

$$J_k = \int_0^1 F'(x^k + td^k) dt.$$  

(4.2.5)

Then

$$J_k d^k = y^k,$$  

(4.2.6)

by Lemma 2.1.1. Similar to (2.4.12), we have

$$B_k e_j = J_k e_j,$$  

(4.2.7)

for $j \in c_i$. 

The CM-successive displacement algorithm is also an update algorithm, and the update formula can be written as:

$$B_k = B_{k-1}(I - \sum_{j \in c_i} e_j e_j^T) + \sum_{j \in c_i} J_k e_j e_j^T.$$

(4.2.8)

From (4.2.8), it is easy to get the following result:

**Lemma 4.2.2.** Let $B_k$, $k=1,2,...$, be generated by Algorithm 4.2.1. If $k \geq p$, then

$$B_k = \sum_{j=k-p+1}^{k} \sum_{l \in c_i} J_{j} e_l e_l^T.$$  

(4.2.9)

**Theorem 4.2.3.** Let $F'$ satisfy Lipschitz condition (2.1.4). Also let \{x_j\}_{j=0}^{k} \subset D and \{B_j\}_{j=0}^{k} be generated by Algorithm 4.2.1 with $|h^{k}| \leq \frac{2}{\sqrt{n}} ||x^{k} - x^{k-1}||$. 

If
\[ \{x^j + d^j\}_{j=1}^k \subset D, \text{ then for } k \geq p, \]
\[ ||B_k - F'(x_k)||_F \leq \alpha \sum_{j=k-p+1}^k ||x^j - x^{j-1}||. \quad (4.2.10) \]

Proof. By (4.2.5), (4.2.1) and Lipschitz condition (2.1.4),
\[ ||(F'(x^m) - J_m)e_j|| \]
\[ = ||(\int_0^1 (F'(x^m + td^m) - F'(x^m))dt)e_j|| \]
\[ \leq \alpha_j \int_0^1 ||d^m|| t dt = \frac{\alpha_j}{2} ||d^m|| \]
\[ = \frac{\alpha_j}{2} ||\sum_{j \in c_m} h^m e_j|| \]
\[ \leq \frac{\alpha_j}{2} \sqrt{n} ||h^m|| \leq \alpha_j \|x^m - x^{m-1}\|, \]
where \( k-p+1 \leq m \leq k \). It follows from (4.2.9) and (4.2.11) that
\[ ||F'(x^k) - B_k||_F \]
\[ = \sum_{m=k-p+1}^k ||\sum_{j \in c_m} (F'(x^k) - B_k)e_j||_F^2 \]
\[ = \sum_{m=k-p+1}^k \sum_{j \in c_m} ||(F'(x^k) - J_m)e_j||^2 \]
\[ \leq \sum_{m=k-p+1}^k \sum_{j \in c_m} (||F'(x^k) - F'(x^m)||e_j|| + ||(F'(x^m) - J_m)e_j||)^2 \]
\[ \leq \sum_{m=k-p+1}^k \sum_{j \in c_m} \alpha_j^2 (||x^k - x^m|| + ||x^m - x^{m-1}||)^2 \quad (4.2.12) \]
\[
\begin{align*}
&\leq \sum_{m=k-p+1}^{k} \sum_{j \in \mathcal{E}_m} \alpha_j^2 \left( \sum_{l=1-p+1}^{k} ||x^l - x^{l-1}|| \right)^2 \\
&= \alpha^2 \left( \sum_{l=k-p+1}^{k} ||x^l - x^{l-1}|| \right)^2.
\end{align*}
\]

Then, (4.2.10) follows from (4.2.12).

From Theorem 4.2.3, it can be seen that the CM-successive displacement algorithm is also semi-self-corrective, and there are only \( p \) terms in the summation. Since \( p \) is much smaller than \( n \) for large sparse problems, \( B_k \) may only retain the information from a few previous steps. Therefore, \( B_k \) generated by the CM-successive displacement algorithm, may be a much better approximation to \( F'(x^k) \) than the approximation generated by the column-update algorithm.

To start iteration (1.3), for a given \( x^0 \in D \), an initial matrix \( B_0 \) is needed. We suggest using the CPR-CM algorithm to get \( B_0 \) since it is easy to implement after we have a consistent partition of the columns of the Jacobian.

### 4.3. A Kantorovich-Type Analysis.

By means of Theorem 4.2.3, we have the following Kantorovich-type analysis for the CM-successive displacement algorithm.

**Theorem 4.3.1.** Assume that \( F'(x) \) satisfies Lipschitz condition (2.1.4). Let \( x^0 \in D \), and let \( B_0 \) be a nonsingular \( n \times n \) matrix such that
\[ ||B_0 - F'(x^0)||_F \leq \delta, \ ||B_0^{-1}||_F \leq \beta, \ ||B_0^{-1}F(x^0)|| \leq \eta, \]
\[ h = \frac{\alpha \beta \eta}{(1-3\beta\delta)^2} \leq \frac{1}{6}, \]  
(4.3.1)

and
\[ \beta \delta < \frac{1}{3}. \]

If \( \overline{S}(x^0, 2t^*) \subset D \), where
\[ t^* = \frac{1-3\beta\delta}{3\alpha\beta} (1 - \sqrt{1-6h}), \]  
(4.3.2)
then \( \{x^k\} \), generated by the CM-successive displacement algorithm with
\[ ||h^k|| \leq \frac{2}{\sqrt{n}} ||x^k - x^{k-1}|| \]
and without any global strategy, converges to \( x^* \),
which is the unique root of \( F(x) \) in \( \overline{S}(x^0, \overline{t}) \cap D \), where
\[ \overline{t} = \frac{1-\beta\delta}{\alpha\beta} \left[ 1 + \left( 1 - \frac{2\alpha\beta\eta}{(1-\beta\delta)^2} \right)^\frac{1}{2} \right]. \]

**Proof.** The proof of this theorem is almost the same as that of Theorem 3.3.2.

However, instead of showing (3.3.3), we must show that
\[ \{x^m + d^m\} \in \overline{S}(x^0, 2t^*). \]  
(4.3.3)

This can be obtained from
\[ ||x^m + d^m - x^0|| \leq ||x^m - x^0|| + ||d^m|| \]
\[ \leq ||x^m - x^0|| + ||x^{m-1} - x^m|| \leq 2||x^m - x^0|| \leq 2t^*. \]

**4.4. Local Convergence Properties.**

**Theorem 4.4.1.** Let \( F:D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n \) satisfy (2.1.1), and let \( F' \) satisfy Lipschitz condition (2.1.4). Also let \( \{x^k\} \) be generated by Algorithm 4.2.1 with
\[ |h^k| \leq \frac{2}{\sqrt{n}} ||x^k - x^{k-1}|| \] and without any global strategy. Then, there exist \( \epsilon, \delta > 0 \) such that if \( x^0 \in D \) and \( B_0 \) satisfy

\[ ||x^0 - x^*|| < \epsilon, \quad ||F'(x^0) - B_0||_F \leq \delta, \]

then \( \{x^k\} \) is well defined and converges q-superlinearly to \( x^* \).

**Proof.** As we discussed in the proof of Theorem 3.4.1, when \( \epsilon \) and \( \delta \) are small enough, we have that \( h \leq \frac{1}{6}, \beta \delta < \frac{1}{3} \) and that \( \bar{S}(x^0,2t^*) \subset D \) where \( h, \beta \) and \( t^* \) are defined in Theorem 4.3.1. Therefore, by Theorem 4.3.1,

\[ x^k + d^k \in D, \quad k=0, 1, \ldots \]

Notice that by (4.2.8),

\[
B_k - F'(x^*) \\
= (B_{k-1} - F'(x^*)) \left( I - \sum_{j \in c_i} e_j e_j^T \right) + \sum_{j \in c_i} (J_k - F'(x^*)) e_j e_j^T.
\]

(4.4.1)

Thus,

\[
||J_k F'(x^*)||_F \\
= ||\int_0^1 (F'(x^k - td^k) - F'(x^*)) dt ||_F \\
\leq \alpha(||x^k - x^*|| + \frac{1}{2} ||d^k||) \\
\leq \alpha(||x^k - x^*|| + ||x^k - x^{k-1}||) \\
\leq \alpha(2||x^k - x^*|| + ||x^{k-1} - x^*||).
\]

(4.4.2)

Let \( \sigma(x^{k-1}, x^k) = \max \{ ||x^k - x^*||, ||x^{k-1} - x^*|| \} \). Then it follows from (4.4.1) and (4.4.2) that
\[ ||B_k - F'(x^*)||_F \leq ||B_{k-1} - F'(x^*)||_F + ||J_k - F'(x^*)||_F \]
\[ \leq ||B_{k-1} - F'(x^*)||_F + 3\alpha \sigma(x^{k-1}, x^k). \]

Also notice that by Lipschitz condition (2.1.4),
\[ ||F'(x^*) - B_0||_F \]
\[ \leq ||F'(x^*) - F'(x^0)||_F + ||F'(x^0) - B_0||_F \]
\[ \leq \alpha ||x^* - x^0|| + \delta \]
\[ \leq \alpha \epsilon + \delta \]

Thus, by Theorem 2.5.3, \( \{x^k\} \) converges at least \( q \)-linearly to \( x^* \).

According to Theorem 2.1.8, to show the \( q \)-superlinear convergence we need only to prove

\[ \lim_{k \to \infty} \frac{||(B_k - F'(x^*)(x^{k+1} - x^k)||}{||x^{k+1} - x^k||} = 0. \] (4.4.3)

From (4.2.10), it follows that

\[ \lim_{k \to \infty} ||B_k - F'(x^*)|| = 0. \] (4.4.4)

(4.4.4) implies (4.4.3).

**Theorem 4.4.2.** Assume that \( F \) satisfies the hypotheses in Theorem 4.4.1. Then the \( r \)-convergence order of Algorithm 4.2.1 is not less than \( r \), where \( r \) is the unique positive root of

\[ t^{p+1} - t^p - 1 = 0. \] (4.4.5)

The proof of this theorem is almost the same as that of Theorem 3.4.2.
4.5. The Modified CM-Successive Displacement Algorithm.

Estimate (4.2.10) shows that when \( p \) is small, \( B_k \) is a good approximation to \( F'(x^k) \). However, \( B_k \) still retains information from the previous \( p \) steps. Therefore, the following question is reasonable: Can we have a better approximation to \( F'(x^k) \) without more function evaluations? Notice that when we get \( B_k \) by Algorithm 4.2.1, we did not use the value of \( F(x^k) \). The main idea of the modified CM-successive displacement algorithm stated below is to use all the information we already have to improve our approximation to \( F'(x^k) \).

Algorithm 4.5.1. Given a consistent partition of the columns of the Jacobian, a vector \( x^0 \) and a nonsingular matrix \( B_0 \) with the same sparsity as the Jacobian, at the initial step:

1. Set \( k = 0 \), \( l = 0 \) and \( \overline{B}_0 = B_0 \).

2. Solve \( \overline{B}_0 s^0 = -F(x^0) \).

3. Choose \( x^1 \) by \( x^1 = x^0 + s^0 \), or by a global strategy.

At each iteration \( k > 0 \):

1. Update \( B_{k-1} \) by Algorithm 4.2.1 to get \( B_k \).

2. Update \( B_k \) by Schubert's update to get \( \overline{B}_k \).

3. Solve \( \overline{B}_k s^k = -F(x^k) \).
(4). Choose \( x^{k+1} \) by \( x^{k+1} = x^k + s^k \), or by a global strategy.

(5). Check for convergence.

Our numerical results show that Algorithm 4.5.1 usually converges faster than Algorithm 4.2.1. When the problem is not well behaved, and a global strategy is used, it behaves significantly better than Algorithm 4.2.1. The cost of the improvement is the computation of Schubert's update. However, since the Jacobian is sparse, Schubert's update requires only \( O(n) \) operations. We feel that it is worth doing this rather than computing more function values and solving more linear systems.

Now we will briefly discuss the convergence properties of Algorithm 4.5.1. Let

\[
\overline{J}_k = \int_0^1 F'(x^{k-1} + t(x^k - x^{k-1}))dt. \tag{4.5.1}
\]

Since \( \overline{J}_k \) performs exactly as the secant factor \( \frac{f(x^k) - f(x^{k-1})}{x^k - x^{k-1}} \) in one dimensional problems, we call \( \overline{J}_k \) the secant operator. Notice that by Lemma 2.4.1, if \( F' \) satisfies Lipschitz condition (2.1.4), then

\[
|| \overline{J}_k - F'(x^k)||_F \leq \frac{\alpha}{2} ||x^k - x^{k-1}||. \tag{4.5.2}
\]

Therefore, \( \overline{J}_k \) is a good approximation to \( F'(x^k) \) when \( ||x^k - x^{k-1}|| \) is small.

**Theorem 4.5.1.** Let \( F' \) satisfy Lipschitz condition (2.1.4). If \( \{B_k\} \) and \( \{\overline{B}_k\} \) are generated by Algorithm 4.5.1, then

\[
|| \overline{B}_k - \overline{J}_k ||_F \leq ||B_k - \overline{J}_k ||_F. \tag{4.5.3}
\]
If, in addition, $\overline{B}_k \neq B_k$, then the strict inequality holds.

**Proof.** Since $\overline{J}_k \in Q(x) \cap Z$, by Theorem 2.6.3, we have

$$||\overline{B}_k - \overline{J}_k||_F^2 + ||\overline{B}_k - B_k||_F^2 = ||B_k - \overline{J}_k||_F^2. \tag{4.5.4}$$

Then, the result of this theorem follows immediately from (4.5.4).

Notice that in general, $\overline{B}_k \neq B_k$. Therefore, by Theorem 4.5.1, $\overline{B}_k$ is usually closer to the secant operator $\overline{J}_k$ than $B_k$. Thus, $\overline{B}_k$ should be a better approximation to the Jacobian than $B_k$ when $B_k$ retains some information from the previous steps. But theoretically, we can not get a better estimate for $||\overline{B}_k - F'(x_k)||_F$ than that for $||B_k - F'(x_k)||_F$. However, we can get the following result:

**Theorem 4.5.2.** Let $F : R^n \to R^n$ satisfy Lipschitz condition (2.1.4). Also let $\{B_k\}$ and $\{x^k\}$ be generated by Algorithm 4.5.1. Then,

$$||\overline{B}_k - F'(x_k)||_F \leq 2\alpha \sum_{j=k-p+1}^{k} ||x^j - x^{j-1}||. \tag{4.5.5}$$

**Proof.** By (4.5.3),

$$||\overline{B}_k - F'(x_k)||_F$$

$$\leq ||\overline{B}_k - \overline{J}_k||_F + ||\overline{J}_k - F'(x_k)||_F$$

$$\leq ||B_k - \overline{J}_k||_F + ||\overline{J}_k - F'(x_k)||_F$$

$$\leq ||B_k - F'(x_k)||_F + 2 ||\overline{J}_k - F'(x_k)||_F$$

Then, from (4.2.10) and (4.5.2), we obtain (4.5.5).
From estimate (4.5.5), it is easy to prove that Algorithm 4.5.1 has at least the same local convergence properties as Algorithm 4.2.1.

4.6. Numerical Results.

We computed the examples given in Section 3.5 by the CPR algorithm, Schubert's algorithm, Algorithm 4.2.1, and Algorithm 4.5.1. In this section, we compare the numerical results from these four algorithms. The global strategy we used in computing the example s is the line search with the backtracking strategy (see Dennis and Schnabel (1983 p.126)). For the CPR algorithm, 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. For the other algorithms, if \( p^k \) is not a descent direction, then we try \(-p_k\). If it is not a descent direction either, then we try the CPR direction. If the CPR direction fails, then the algorithm fails. In the CPR algorithm, Algorithm 4.2.1 and Algorithm 4.5.1, at step \( k \), we use different \( h_j^k \) for each component of \( x^k \) instead of one uniform \( h^k \), and \( h_j^k \) is defined by (2.3.5). The stopping test, \( \epsilon \) and the machine precision are the same as those in section 3.5.

Example 4.6.1, Example 4.6.2, Example 4.6.3 and the initial guesses are also the same as those in Section 3.5. The results are shown in the tables below, 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. \( z_0 \) is the initial guess.
**Example 4.6.1.**

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>$x_0 = x_1$</th>
<th>$x_0 = x_2$</th>
<th>$x_0 = x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IT</td>
<td>NF</td>
<td>LN</td>
</tr>
<tr>
<td>CPR</td>
<td>22</td>
<td>88</td>
<td>15</td>
</tr>
<tr>
<td>Schubert</td>
<td>38</td>
<td>41</td>
<td>21</td>
</tr>
<tr>
<td>Alg. 4.2.1</td>
<td>fail</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alg. 4.5.1</td>
<td>24</td>
<td>50</td>
<td>14</td>
</tr>
</tbody>
</table>

*Table 4.6.1.*

**Example 4.6.2 (Broyden tridiagonal function).**

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>$x_0 = x_1$</th>
<th>$x_0 = x_2$</th>
<th>$x_0 = x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IT</td>
<td>NF</td>
<td>LN</td>
</tr>
<tr>
<td>CPR</td>
<td>5</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Schubert</td>
<td>7</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Alg. 4.2.1</td>
<td>6</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>Alg. 4.5.1</td>
<td>6</td>
<td>14</td>
<td>0</td>
</tr>
</tbody>
</table>

*Table 4.6.2*

**Example 4.6.3 (Discrete boundary value function).**

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>$x_0 = x_1$</th>
<th>$x_0 = x_2$</th>
<th>$x_0 = x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IT</td>
<td>NF</td>
<td>LN</td>
</tr>
<tr>
<td>CPR</td>
<td>3</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>Schubert</td>
<td>4</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Alg. 4.2.1</td>
<td>4</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Alg. 4.5.1</td>
<td>4</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

*Table 4.6.3.*
CHAPTER 5

The Combined Secant Algorithm

5.1. Introduction.

In this chapter, we consider nonlinear system of equations with some special sparsity structures. For example, consider the Jacobian with the following sparsity structure:

\[
\begin{bmatrix}
\times & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \times & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \times & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \times & 0 & 0 & 0 \\
\times & \times & \times & 0 & 0 & \times & 0 & 0 \\
\times & \times & \times & 0 & 0 & 0 & \times & 0 \\
\times & \times & \times & 0 & 0 & 0 & 0 & \times \\
\end{bmatrix}
\]

(5.1.1)

The partition \( c_1 = \{1\} \), \( c_2 = \{2\} \), \( c_3 = \{3\} \), \( c_4 = \{4, 5, 6, 7\} \) is an optimal consistent partition of the columns of the Jacobian. For this problem, the partitioned secant algorithm requires 4 function values at each iterative step.

In this chapter, we propose a combined secant algorithm, which is a combination of Algorithm 3.3.2 and Schubert's algorithm (including Broyden's algorithm). This algorithm can save more function values than the partitioned secant algorithm by considering special structures of the Jacobians of some
problems. For example (5.1.1), the number of function evaluations is 2 instead of the 4 required by the partitioned secant algorithm.

The combined secant algorithm and its properties are given in Section 5.2. A Kantorovich-type analysis for this algorithm is given in Section 5.3. A $q$-superlinear convergence result is given in Section 5.4.

5.2. The Combined Secant Algorithm and its Properties.

Consider example (5.1.1). The first 3 columns of the matrix are denser than the other columns, and this makes $p$, the number of the groups in the partition, at least 4. The combined secant algorithm allows us to divide the columns of the Jacobian into two parts, and to use different algorithms on each part.

We say a group of the columns of a matrix has 'good sparsity' if the columns in this group have few nonzeros in the same row position. Otherwise, we say the group of the columns has 'bad sparsity'.

If some columns of the Jacobian have bad sparsity, and others have good sparsity, then we can divide all columns into two groups — the good sparsity group and the bad sparsity group. In other words, the set $\{1, \ldots, n\}$ is divided into 2 subsets $c_1$ and $c$ such that $c$ contains the good sparsity columns, and $c_1$ contains the bad sparsity columns.

For $B \in L(R^n)$, let

$$B_1 = B \sum_{j \in c_1} e_j e_j^T, \quad B_2 = B \sum_{j \in c} e_j e_j^T.$$
Then $B = B_1 + B_2$. The main idea of the combined algorithm is to use Schubert's update (including Broyden's update) on $B_1$ and to use Algorithm 3.2.2 on $B_2$.

*Algorithm 5.2.1.* Given a consistent partition of $B_2$, which divides $c$ into $p - 1$ subsets $c_2, c_3, \ldots, c_p$, and given an $x^0 \in \mathbb{R}^n$ and a nonsingular matrix $B_0$ with the same sparsity as the Jacobian, at each step $k \geq 0$:

1. Solve

   $$B_k s^k = -F(x^k)$$

2. Choose $x^{k+1}$ by $x^{k+1} = x^k + s^k$ or by a global strategy.

3. Check for convergence.

4. Update $B_{k,1}$ by Schubert's update to get $B_{k+1,1}$, and update $B_{k,2}$ by Algorithm 3.2.2 to get $B_{k+1,2}$.

5. Set

   $$B_{k+1} = B_{k+1,1} + B_{k+1,2}.$$

Let

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

(5.2.1)

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

(5.2.2)

$$y_i = F(\bar{x} - g_{i-1}) - F(\bar{x} - g_i), \quad i = 1, 2, \ldots, p,$$

and
\[ J_i = \int_0^1 F''(x - g_i + t(g_i - g_{i-1})) dt, \quad i = 1, 2, ..., p. \quad (5.2.3) \]

Then,

\[ J_i d_i = y_i, \quad i = 1, 2, ..., p \quad (5.2.4) \]

by Lemma 2.1.1, and the update of Algorithm 5.2.1 can be formulated as

\[
\begin{align*}
\bar{B}_1 &= B_1 + \sum_{i=1}^{n} ([d_1]_i^T[d_1]_i)^{+} e_i e_i^T(y_1 - B_1 d_1)[d_1]_i^T, \\
\bar{B}_2 &= B_2 + \sum_{i=2}^{p} \sum_{j \in e_i} s_j e_j (J_i - B) e_j e_j^T, \\
\bar{B} &= \bar{B}_1 + \bar{B}_2.
\end{align*} \quad (5.2.5)
\]

Now we give some of the properties of \( \bar{B} \) obtained from (5.2.5).

**Lemma 5.2.1.** \( \bar{B} \) satisfies the secant equations

\[ \bar{B} d_i = y_i, \quad i = 1, ..., p, \quad (5.2.6) \]

and (5.2.6) implies that

\[ \bar{B}_s = F(x) - F(x) = y. \quad (5.2.7) \]

**Lemma 5.2.2.** \( \bar{B} \) is the unique solution to

\[ \min \{ ||\hat{B} - B||_F; \hat{B} d_i = y_i, \quad i = 1, ..., p, \text{ and } \hat{B} \in Z \}. \quad (5.2.8) \]

The proof is similar to that of Theorem 2.6.3.

**Theorem 5.2.3.** If \( A \in L(R^n) \) has the same sparsity as the Jacobian, then
\[
\|B_1 - A_1\|_F^2 \leq \|B_1 - A_1\|_F^2 - \frac{1}{\|s\|^2} \|B_1 - A_1s\|^2
+ \sum_{i=1}^n \left( \|d_i\|^2 + [v_i]^T [d_i]_i + \|e_i^T (y_1 - A d_1)\|^2 \right).
\]  

(5.2.9)

\textbf{Proof.} Let \( E_1 = B_1 - A_1 \), and \( d_1 = B_1 - A_1 \). From (5.2.5), we have

\[
e_i^T \bar{E}_1 = e_i^T B_1 + ([d_i]^T [d_i]_i + e_i^T (y_1 - B_1 d_1) [d_i]_i)^T.
\]  

(5.2.10)

Subtracting \( e_i^T A d_1 \) from both sides of (5.2.10), and noticing that \( e_i^T B_1 d_1 = e_i^T B_1 [d_i]_i \) and that \( e_i^T A_1 d_1 = e_i^T A_1 [d_i]_i \), we obtain

\[
e_i^T \bar{E}_1 = e_i^T E_1 + ([d_i]^T [d_i]_i + e_i^T (y_1 - B_1 d_1) [d_i]_i)^T
\]  

(5.2.11)

\[= e_i^T E_1 (I - ([d_i]^T [d_i]_i + e_i^T (y_1 - A d_1) [d_i]_i)^T
\]  

\[+ ([d_i]^T [d_i]_i + e_i^T (y_1 - A d_1) [d_i]_i)^T.
\]

Since the first and second terms on the right of (5.2.11) are perpendicular to each other, we have

\[
\|e_i^T \bar{E}_1\|^2 = \|e_i^T E_1 (I - ([d_i]^T [d_i]_i + e_i^T (y_1 - A d_1) [d_i]_i)^T\|^2
\]  

\[+ ([d_i]^T [d_i]_i + e_i^T (y_1 - A d_1) [d_i]_i)^2
\]  

\[= \|e_i^T E_1\|^2 - \|([d_i]^T [d_i]_i + e_i^T (y_1 - A d_1) [d_i]_i)^2
\]  

\[+ ([d_i]^T [d_i]_i + e_i^T (y_1 - A d_1) [d_i]_i)^2.
\]

Therefore,

\[
\|B_1 - A_1\|_F^2 = \sum_{i=1}^n \|e_i^T \bar{E}_1\|^2
\]  

\[\leq \|B_1 - A_1\|_F^2 - \frac{1}{\|s\|^2} \|B_1 - A_1 d_1\|^2
\]  

\[+ \sum_{i=1}^n \left( \|d_i\|^2 + [v_i]^T [d_i]_i + \|e_i^T (y_1 - A d_1)\|^2 \right).
\]
Theorem 5.2.4. If $A \in L(R^n)$ has the same sparsity as the Jacobian, then

$$|| \overline{E}_2 - A_2 ||^2_F \leq || B_2 - A_2 ||^2_F - \frac{1}{|| s ||^2} || (B_2 - A_2)s ||^2$$

$$+ \sum_{i=2}^{p} \sum_{j \in c_i} s_i^+ s_j || (J_i - A)e_j ||^2 . \tag{5.2.12}$$

Proof. Let $\overline{E}_2 = \overline{E}_2 - A_2$, and $E_2 = B_2 - A_2$. It follows from (5.2.5) that if $j \in c_i, i = 2, \ldots, p$, then

$$\overline{E}_2 e_j = B_2 e_j + s_i^+ s_j (J_i - B_2) e_j . \tag{5.2.13}$$

Subtracting $A_2 e_j$ from both sides of (5.2.13), we obtain

$$\overline{E}_2 e_j = (1 - s_i^+ s_j) E_2 e_j + s_i^+ s_j (J_i - A_2) e_j .$$

Since $(1 - s_i^+ s_j) s_i^+ s_j = 0$, we have

$$|| \overline{E}_2 e_j ||^2 = (1 - s_i^+ s_j) || E_2 e_j ||^2 + s_i^+ s_j || (J_i - A_2) e_j ||^2$$

$$= || E_2 e_j ||^2 - s_i^+ s_j || E_2 e_j ||^2 + s_i^+ s_j || (J_i - A_2) e_j ||^2 .$$

Therefore,

$$|| \overline{E}_2 ||^2_F = \sum_{j \in c} || \overline{E}_2 e_j ||^2$$

$$= || E_2 ||^2_F - \sum_{j \in c} s_i^+ s_j || E_2 e_j ||^2 + \sum_{i=2}^{p} \sum_{j \in c_i} s_i^+ s_j || (J_i - A)e_j ||^2 . \tag{5.2.14}$$
In addition,

\[
\sum_{j \in C} s_j^+ s_j \| E_2 e_j \|^2 = \| E_2 \sum_{j \in C} s_j^+ s_j e_j e_j^T \|^2 \geq \frac{\| E_2 \sum_{j \in C} s_j^+ s_j e_j e_j^T s \|^2}{\| s \|^2} = \frac{\| E_2 s \|^2}{\| s \|^2}.
\]

Thus, (5.2.12) follows from (5.2.14).

5.3. A Kantorovich-type Analysis.

To study the convergence properties of Algorithm 5.2.1, we assume that \( F' \) satisfies the following Lipschitz condition: For every \( i \in C \), there exists \( \gamma_i > 0 \), such that

\[
\| (F'(x) - F'(y)) e_i \| \leq \gamma_i \| x - y \| , \quad \forall \ x, y \in D , \tag{5.3.1}
\]

and there exists \( \Theta_i > 0, \ i = 1, 2, ..., n \), such that

\[
\| e_i^T (F'(x)_1 - F'(y)_1) \| \leq \Theta_i \| x - y \| , \quad \forall \ x, y \in D . \tag{5.3.2}
\]

Let \( \gamma = (\sum_{i \in C} \gamma_i^2)^{\frac{1}{2}}, \ \Theta = (\sum_{i=1}^n \Theta_i^2)^{\frac{1}{2}}, \ \alpha = (\gamma^2 + \Theta^2)^{\frac{1}{2}}. \) If \( F' \) satisfies this Lipschitz condition, then the following are true:

\[
\| F'(x)_1 - F'(y)_1 \|_F \leq \Theta \| x - y \| , \quad \forall \ x, y \in D , \tag{5.3.3}
\]

\[
\| F'(x)_2 - F'(y)_2 \|_F \leq \gamma \| x - y \| , \quad \forall \ x, y \in D , \tag{5.3.4}
\]

and

\[
\| F'(x) - F'(y) \|_F \leq \alpha \| x - y \| , \quad \forall \ x, y \in D . \tag{5.3.5}
\]
Lemma 5.3.1. Let $F'$ satisfy (5.3.1) and (5.3.2), and let $B$ be generated by Algorithm 5.2.1. If $\bar{z} \in D$ and $\bar{z} - d_1 \subset D$, then for any $z \in D$,

$$\| B_1 - F'(z_1) \|_F^2 \leq \| B_1 - F'(z_1) \|_F^2 - \frac{1}{\| s \|_2^2} \| (B_1 - F'(z_1))s \|_2^2$$

$$+ \Theta^2(\| \bar{z} - z \| + \frac{1}{2} \| d_1 \|)^2.$$  

(5.3.6)

Proof. Substituting $F'(z)$ for $A$ in (5.2.9), we obtain

$$\| B_1 - F'(z_1) \|_F^2 \leq \| B_1 - F'(z_1) \|_F^2 - \frac{1}{\| s \|_2^2} \| (B_1 - F'(z_1))s \|_2^2$$

$$+ \sum_{i=1}^n (|d_i|^T[d_1]^i |e_i^T(y_1 - F'(z)d_i)|^2).$$  

(5.3.7)

By (5.2.3), (5.2.4), (5.3.3), and Cauchy-Schwarz inequality we have

$$\sum_{i=1}^n (|d_i|^T[d_1]^i |e_i^T(y_1 - F'(z)d_i)|^2)$$

$$= \sum_{i=1}^n (|d_i|^T[d_1]^i |e_i^T(J_1 - F'(z))_i[d_1]^i)|^2)$$

$$\leq \sum_{i=1}^n (|d_i|^T[d_1]^i |e_i^T(J_1 - F'(z))_i||[d_1]^i|^2$$

$$\leq \sum_{i=1}^n \| e_i^T(J_1 - F'(z))_i \|_2^2$$

$$= \| (J_1 - F'(z))_1 \|_F^2$$

$$= \frac{1}{\| J_1 - F'(z) - (1 - t)d_1 \|_F} \| F'(\bar{z} - (1 - t)d_1 - F'(z))_1 dt \|_F^2$$

$$\leq \Theta^2(\| \bar{z} - z \| + \frac{1}{2} \| d_1 \|)^2.$$
Then (5.3.6) follows from (5.3.7) and (5.3.8).

**Lemma 5.3.2.** Let $F'$ satisfy (5.3.1) and (5.3.2), and let $\bar{B}$ be generated by Algorithm 5.2.1. If $\bar{x} \in D$ and $\{\bar{x} - g_i, i = 2, \ldots, p\} \subset D$, then for any $z \in D$,

$$||\bar{B}_2 - F'(z)_2||^2 \leq ||B_2 - (F'(z)_2)||^2 - \frac{1}{||s||^2} ||(B_2 - F'(z)_2)s||^2$$

$$+ \gamma^2(||\bar{x} - z|| + ||s||)^2.$$  

(5.3.9)

**Proof.** Substituting $F'(z)$ for $A$ in (5.2.12), we obtain

$$||\bar{B}_2 - F'(z)_2||^2 \leq ||B_2 - F'(z)_2||^2 - \frac{1}{||s||^2} ||(B_2 - F'(z)_2)s||^2$$

$$+ \sum_{i=2}^{p} \sum_{j \in c_i} s_j^+ s_j ||(J_i - F'(z))e_j||^2.$$  

(5.3.10)

It follows from (5.2.3) and (5.3.1) that

$$\sum_{i=2}^{p} \sum_{j \in c_i} s_j^+ s_j ||(J_i - F'(z))e_j||^2$$

$$\leq \sum_{i=2}^{p} \sum_{j \in c_i} ||(J_i - F'(z))e_j||^2$$

$$= \sum_{i=2}^{p} \sum_{j \in c_i} \frac{1}{0} \int (F'(\bar{x} - g_i + t(g_i - g_{i-1})) - F'(z)) dt \ e_j ||^2$$  

(5.3.11)

$$\leq \sum_{i=2}^{p} \sum_{j \in c_i} \gamma^2(f(||\bar{x} - z|| + (1 - t)||g_i|| + t ||g_i - g_{i-1}||) dt)^2$$

$$\leq \sum_{i=2}^{p} \sum_{j \in c_i} \gamma^2(||\bar{x} - z|| + ||s||)^2$$

$$= \gamma^2(||\bar{x} - z|| + ||s||)^2.$$  


Thus, (5.3.9) follows from (5.3.10) and (5.3.11).

Let

\[ d^k_i = \sum_{j \in c_i} s^k_{ij} e_j , \]

and

\[ g^k_i = \sum_{j=1}^i d^k_j, \quad i = 1, 2, \ldots, p, \quad g^k_0 = 0. \]

We have the following estimate for \( B_{k+1} \).

**Theorem 5.3.3.** Let \( F' \) satisfy (5.3.1) and (5.3.2), and let \( \{x^k\} \) and \( \{B_k\} \) be generated by Algorithm 5.2.1. If \( \{x^{i+1}_j\}_{j=0}^{k+1} \subset D \) and \( \{x^i - g^i_j, \; i = 1, 2, \ldots, p \}_{j=1}^{k+1} \subset D \), then

\[
||B_{k+1} - F'(x^{k+1})||_F \\
\leq ||B_0 - F'(x_0)||_F + 2\alpha \sum_{i=0}^k ||x^{i+1} - x^i||. \quad (5.3.12)
\]

**Proof.** Substituting \( z \) for \( \overline{x} \) in (5.3.6) and (5.3.9), we have

\[
||\overline{B}_1 - F'(\overline{x})_1|| \leq ||B_1 - F'(\overline{x})_1|| + \left( \frac{\Theta}{2} ||d_1|| \right)^2
\]

and

\[
||\overline{B}_2 - F'(\overline{x})_2|| \leq ||B_2 - F'(\overline{x})_2|| + (\gamma ||s||)^2.
\]

Therefore

\[
||\overline{B} - F'(\overline{x})|| \leq ||\overline{B}_1 - F'(\overline{x})_1|| + ||\overline{B}_2 - F'(\overline{x})_2|| \leq ||B - F'(\overline{x})|| + (\Theta^2 + \gamma^2)||s||^2
\]

\[
= ||B - F'(\overline{x})|| + \alpha^2 ||s||^2.
\]
Then
\[ \|\overline{B} - F'(\overline{x})\|_F \leq \|B - F'(\overline{x})\|_F + \alpha \|\overline{x} - x\| \]
\[ \leq \|B - F'(x)\|_F + 2\alpha \|\overline{x} - x\|. \]
(5.3.13)

Thus, (5.3.12) follows (5.3.13).

From (5.3.12), we have the following Kantorovich-type theorem for Algorithm 5.2.1. Since the proof of this theorem is slightly different from that of Theorem 2.5.6, we will give the proof in detail.

**Theorem 5.3.4.** Assume that \( F' \) satisfies (5.3.1) and (5.3.2). Also assume that \( x_0 \in D \) and \( B_0 \in L(R^n) \) satisfy
\[ \|B_0 - F'(x_0)\|_F \leq \delta, \quad \|B_0^{-1}\|_F \leq \beta, \quad \|B_0^{-1}F(x_0)\| \leq \eta \]
and
\[ h = \frac{\alpha \beta \eta}{(1-3\beta \delta)^2} \leq \frac{1}{10}, \quad \beta \delta < \frac{1}{3}. \]

If \( \overline{S}(x_0, 2t^*) \subset D \), where
\[ t^* = \frac{1-3\beta \delta}{5\alpha \beta} \left(1-(1-10^\frac{1}{2})\right), \]
then \( \{x^k\} \), generated by Algorithm 5.2.1 without any global strategy, converges to \( x^* \), which is the unique root of \( F(x) \) in \( \overline{S}(x_0, \overline{t}) \cap D \), where
\[ \overline{t} = \frac{1-\beta \delta}{\alpha \beta} \left[1 + \left(1-\frac{2\alpha \beta \eta}{(1-\beta \delta)^2}\right)^{\frac{1}{2}}\right]. \]
Proof. Consider the scalar iteration

\[ t_{k+1} - t_k = \frac{2\beta}{2 - \beta \delta} f(t_k), \quad t_0 = 0, \quad k = 1, 2, \ldots \]  \hspace{1cm} (5.3.14)

where

\[ f(t) = \frac{5}{2} \alpha t^2 - \frac{1 - 3\beta \delta}{\beta} t + \frac{n}{\beta}. \]  \hspace{1cm} (5.3.15)

It is easy to show that \( \{t_k\} \) satisfies the difference equation

\[ t_{k+1} - t_k = \frac{\beta}{1 - \phi} \left[ \alpha(t_k - t_{k-1}) + 2\alpha t_{k-1} + \delta(t_k - t_{k-1}) \right], \]  \hspace{1cm} (5.3.16)

where \( \phi = \frac{3 + \beta \delta}{5} < \frac{2}{3} \). From (5.3.16), we see that \( \{t_k\} \) is a monotonically increasing sequence and that

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

where \( t^* \) is the smallest root of (5.3.15).

Now, by induction, we will prove that

\[ ||x^{k+1} - x^k|| \leq t_{k+1} - t_k, \quad k = 1, 2, \ldots, \]  \hspace{1cm} (5.3.17)

\[ \{x^k\} \subset \overline{S}(x^0, t^*), \]  \hspace{1cm} (5.3.18)

\[ \{x^{k+1} + g^k, i = 1, 2, \ldots, p\} \subset \overline{S}(x^0, 2t^*), \]  \hspace{1cm} (5.3.19)

and

\[ ||B_k^{-1}|| \leq \frac{\beta}{1 - \phi} \leq 3\beta, \quad k = 1, 2, \ldots. \]  \hspace{1cm} (5.3.20)

For \( k = 0 \), we have

\[ ||x^1 - x^0|| \leq \eta \leq \frac{2}{2 - \beta \delta} \eta = t_1 - t_0 \leq t^*. \]
Thus,

$$||x^1-g_i^0-x^0|| \leq ||x^1-x^0|| + ||g_i^0|| \leq 2||x^1-x^0|| \leq 2t^*.$$  

Suppose (5.3.17) holds for $k = 0, 1, \ldots, m - 1$. Then,

$$||x^m-x^0|| \leq \sum_{i=0}^{m-1} (t_{i+1} - t_i) = t_m \leq t^*.$$  

Therefore, $x^m \in \mathcal{S}(x^0, t^*)$, and

$$\{x^m - g_i^{m-1}, i = 1, \ldots, p \} \subset \mathcal{S}(x^0, 2t^*).$$  

By Theorem 5.3.3,

$$||B_0^{-1}(B_m - B_0)|| \leq ||B_0^{-1}||_F ||B_m - F'(x^m)||_F + ||F'(x^m) - F'(x^0)||_F + ||F'(x^0) - B_0||_F \leq \beta(3\alpha \sum_{i=0}^{m-1} ||x^{i+1} - x^i|| + 2\delta) \leq \beta(3\alpha t^* + 2\delta) \leq \frac{3 + \beta \delta}{5} = \phi.$$  

Thus, by Lemma 2.1.3,

$$||B_m^{-1}|| \leq \frac{\beta}{1 - \phi} \leq 3\beta.$$  

Therefore,

$$||x^{m+1} - x^m|| \leq \frac{\beta}{1 - \phi} ||B_m^{-1}||_F \leq 3\beta.$$  

This completes the induction step. The rest of the proof is the same as that of Theorem 2.5.6.
5.4. Local Convergence properties.

**Theorem 5.4.1.** Let \( F \) satisfy (2.1.1), and let \( F' \) satisfy (5.3.1) and (5.3.2). Also, let \( \{x^k\} \) be generated by Algorithm 5.2.1 without any global strategy. Then, there exist \( \epsilon, \delta > 0 \), such that if \( x_0 \in D \) and \( B_0 \), a nonsingular \( n \times n \) matrix, satisfy

\[
||x^0 - x^*|| < \epsilon, \quad ||B_0 - F'(x^0)||_F \leq \delta,
\]

then \( \{x^k\} \) is well defined and converges \( q \)-superlinearly to \( x^* \).

**Proof.** As we discussed in the proof of Theorem 3.4.1, when \( \epsilon \) and \( \delta \) are small enough, we have that \( h \leq \frac{1}{10}, \beta \delta < \frac{1}{3} \) and that \( \bar{S}(x^0, 2t^*) \subset D \), where \( h, \beta \) and \( t^* \) are defined in theorem 5.3.4. Therefore, by Theorem 5.3.4,

\[
\{x^k + g_i^k, i = 1, 2, ..., p\} \subset D.
\]

Thus, substituting \( x^* \) for \( x \) in (5.3.6) and (5.3.9), we have

\[
||B_1 - F'(x^*)_1||^2 \leq ||B_1 - F'(x^*)_1||^2 - \frac{1}{||s||^2} ||(B_1 - F'(x^*)_1)s||^2
\]

\[+ \Theta^2(||\bar{x} - x^*|| + ||s||)^2, \tag{5.4.2}\]

and

\[
||B_2 - F'(x^*)_2||^2 \leq ||B_2 - F'(x^*)_2||^2 - \frac{1}{||s||^2} ||(B_2 - F'(x^*)_2)s||^2
\]

\[+ \gamma^2(||\bar{x} - x^*|| + ||s||)^2. \tag{5.4.3}\]

Then,
\[
\| \overline{B} - F'(x^*) \|_F^2 = \| \overline{B}_1 - F'(x^*)_1 \|_F^2 + \| \overline{B}_2 - F'(x^*)_2 \|_F^2 \\
\leq \| \overline{B} - F'(x^*) \|_F^2 + 2^2(\| \overline{x} - x^* \| + \| s \|)^2 \\
\leq \| B - F'(x^*) \|_F^2 + (3\sigma(\overline{x}, \overline{x}))^2
\]
where \( \sigma(x, \overline{x}) = \max\{ \| \overline{x} - x^* \|, \| x - x^* \| \} \). Therefore,

\[
\| \overline{B} - F'(x^*) \|_F \leq \| B - F'(x^*) \|_F + 3\sigma(\overline{x}, \overline{x})
\]

Notice that by (5.3.5),

\[
\| F'(x^*) - B_0 \|_F \\
\leq \| F'(x^*) - F'(x^0) \|_F + \| F'(x^0) - B_0 \|_F \\
\leq \alpha \| x^* - x^0 \| + \delta \\
\leq \alpha \epsilon + \delta
\]

Thus, by Theorem 2.5.3, \( \{x^k\} \) converges at least \( q \)-linearly to \( x^* \).

From Theorem 2.1.8, we need only to show

\[
\lim_{k \to \infty} \frac{\| (B_k - F'(x^*))s^k \|}{\| s^k \|} = 0 , \quad (5.4.4)
\]
to prove \( q \)-superlinear convergence.

Let \( \overline{E} = \overline{B} - F'(x^*) \) and \( E = B - F'(x^*) \). Then, it follows from (5.4.2) and (5.4.3), that

\[
\| \overline{E}_1 \|_F \leq (\| E_1 \|_F^2 - \frac{\| E_1 s \|_F^2}{\| s \|_F^2})^{\frac{1}{2}} + 3\Theta(x, \overline{x}) , \quad (5.4.5)
\]

and that

\[
\| \overline{E}_2 \|_F \leq (\| E_2 \|_F^2 - \frac{\| E_2 s \|_F^2}{\| s \|_F^2})^{\frac{1}{2}} + 3\gamma(x, \overline{x}) . \quad (5.4.6)
\]
From (5.4.5) and (5.4.6), using the same argument for proving the \( q \)-superlinear convergence property of Broyden's algorithm (see Dennis and Moré (1977)), we obtain

\[
\lim_{k \to \infty} \frac{\| (B_k - F'(x^*))_1 s^k \|}{\| s^k \|} = 0, \tag{5.4.7}
\]

and

\[
\lim_{k \to \infty} \frac{\| (B_k - F'(x^*))_2 s^k \|}{\| s^k \|} = 0. \tag{5.4.8}
\]

Notice that

\[
\| (B_k - F'(x^*))s^k \| \leq \| (B_k - F'(x^*))_1 s^k \| + \| (B_k - F'(x^*))_2 s^k \|. \]

Thus, (5.4.4) follows from (5.4.7) and (5.4.8).
CHAPTER 6

Concluding Remarks

We have presented four algorithms for solving sparse nonlinear systems of equations. These algorithms are all based on consistent partitions of the columns of the Jacobians. The partitioned secant algorithm (Algorithm 3.2.2) 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 iterative step, and it has good local convergence properties. We have shown that the partitioned secant algorithm is locally $q$-superlinearly convergent, and that under reasonable assumptions, the $r$-convergence order of the partitioned secant algorithm is not less than $\frac{1+\sqrt{5}}{2}$, 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, the partitioned secant algorithm is probably more efficient than the CPR-CM algorithm.

The CM-successive displacement algorithm (Algorithm 4.2.1) is based on Coleman and More's partitioning algorithm and the column-update algorithm. This algorithm uses only two function values at each iterative step, and it is $q$-
superlinearly convergent. Using this algorithm, one group of the columns of $B_k$ is displaced at each step. However, it is not necessary to update just one group at each iterative step. Actually, we can displace several groups at each step, and this gives the algorithm a faster convergence rate. However, if one more group is displaced, then one more function evaluation is needed. Therefore, the efficiency of the algorithm depends on the number of the groups displaced at each iterative step. The number of groups that should be displaced at each step to make the efficiency of the algorithm optimal is an unsolved problem.

The modified CM-successive displacement algorithm is a combination of the CM-successive displacement algorithm and Schubert's algorithm. It is also $q$-superlinearly convergent. Our numerical results indicate that the modified CM-successive displacement algorithm usually behaves much better than the CM-successive displacement algorithm. However, we have not been able to prove better theoretical convergence results for the modified CM-successive displacement algorithm than those for the CM-successive displacement algorithm. We will leave this problem for a future study. Additional numerical results indicate that the modified CM-successive displacement algorithm is also usually more efficient than the CPR-CM algorithm and Schubert's algorithm. When the problem is not well behaved, or the initial guess is far away from the solution, the modified CM-successive displacement algorithm is much more efficient than Schubert's algorithm.
The combined secant algorithm is a combination of the partitioned secant algorithm and Schubert's algorithm. It can save more function values than the partitioned secant algorithm by considering special structures of the Jacobians. This algorithm is locally $q$-superlinearly convergent.

The idea of the partitioned secant algorithm and the CM-successive displacement algorithm can be used on Powell and Toint's (1979) algorithms by means of a symmetrically consistent partition of the columns of the Hessian, or by means of a consistent partition of the columns of the lower (or upper) triangular part of the Hessian (see Coleman and Moré (1984)). These will lead to the partitioned secant algorithm and the partitioned successive displacement algorithm for unconstrained optimization problems. This will be our future work.

In addition to the four algorithms, we have given a local convergence result for the CPR algorithm, and we have sharpened error estimates and improved Kantorovich-type analyses for both Broyden's algorithm and Schubert's algorithm.
BIBLIOGRAPHY


