Projection-based methods for solving systems of n nonlinear equations in n unknowns

Long Vo Nguyen
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Part of the Computer Sciences Commons

Recommended Citation
https://lib.dr.iastate.edu/rtd/5433

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
INFORMATION TO USERS

This material was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.

4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

5. PLEASE NOTE: Some pages may have indistinct print. Filmed as received.

Xerox University Microfilms
300 North Zeib Road
Ann Arbor, Michigan 48106
NGUYEN, Long Vo, 1942-
PROJECTION-BASED METHODS FOR SOLVING SYSTEMS OF n NONLINEAR EQUATIONS IN n UNKNOWNS.

Iowa State University, Ph.D., 1975
Computer Science

Xerox University Microfilms, Ann Arbor, Michigan 48106

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED.
Projection-based methods for solving systems of n nonlinear equations in n unknowns

by

Long Vo Nguyen

A Dissertation Submitted to the Graduate Faculty in Partial Fulfillment of The Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major: Computer Science

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa
1975
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>AN INDEX TO SYMBOLS AND NOTATIONAL DEFINITIONS</td>
<td>1</td>
</tr>
<tr>
<td>II.</td>
<td>INTRODUCTION AND REVIEW OF MINIMIZATION METHODS</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>A. Paraboloid Methods</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>B. Descent Methods</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>C. Steplength Algorithms</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>D. Conjugate-Direction Methods</td>
<td>11</td>
</tr>
<tr>
<td>III.</td>
<td>THE NONLINEAR PROJECTION METHODS</td>
<td>13</td>
</tr>
<tr>
<td>IV.</td>
<td>NONLINEAR PROJECTION-BASED METHODS</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>A. 1-Dimensional Projection-Based Methods</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>B. 2-Dimensional Projection-Based Methods</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>C. 3-Dimensional Projection-Based Methods</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>D. M-Dimensional Projection-Based Methods</td>
<td>42</td>
</tr>
<tr>
<td>V.</td>
<td>CONVERGENCE</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>A. Proof of Convergence for the 1-Dimensional Projection-Based Methods</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>B. Extension to Higher Dimensions</td>
<td>55</td>
</tr>
<tr>
<td>VI.</td>
<td>COMPARISONS AND TEST PROBLEMS</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>A. Comparison with Basic Projection Methods</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>B. Comparison with Newton's Method</td>
<td>65</td>
</tr>
<tr>
<td>VII.</td>
<td>SUMMARY AND FUTURE RESEARCH</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>A. Summary</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>B. Future Research</td>
<td>68</td>
</tr>
<tr>
<td>VIII.</td>
<td>BIBLIOGRAPHY</td>
<td>70</td>
</tr>
<tr>
<td>IX.</td>
<td>ACKNOWLEDGEMENTS</td>
<td>72</td>
</tr>
</tbody>
</table>
I. AN INDEX TO SYMBOLS AND NOTATIONAL DEFINITIONS

\( a_{ij} \) \hspace{1cm} \text{Equation 4.29}

\( A^k \) \hspace{1cm} \text{Equation 3.12}

\( A_i^k \) \hspace{1cm} \text{Equation 5.3}

\( \alpha_k \) \hspace{1cm} \text{represents the basic steplength in iteration formulas}

\( \alpha_i^k \) \hspace{1cm} \text{Equation 4.9}

\( \alpha_{i}^{k'} \) \hspace{1cm} \text{Equation 4.12}

\( \beta_i^k \) \hspace{1cm} \text{Equation 4.33}

\( \beta_{i}^{k'} \) \hspace{1cm} \text{Equation 4.36}

\( d_{ijl} \) \hspace{1cm} \text{Equation 4.49}

\( dx^k \) \hspace{1cm} \text{represents the change vector} \ x^{k+1} - x^k

\( dx_i^k \) \hspace{1cm} \text{represents the ith component of} \ dx^k

\( f_i \) \hspace{1cm} \text{represents the ith component of the nonlinear system} \ F(x)

\( F(x^k) \) \hspace{1cm} \text{represents the residue vector at the kth iteration}

\( g(x) \) \hspace{1cm} \text{represents the square of the norm of} \ F(x) \text{ - Equation 3.13}

\( g'(x) \) \hspace{1cm} \text{represents the derivative of} \ g(x)

\( \gamma_i^k \) \hspace{1cm} \text{Equation 4.46}

\( \gamma_{i}^{k'} \) \hspace{1cm} \text{Equation 4.51}

\( J^k \) \hspace{1cm} \text{represents the Jacobian matrix evaluated at} \ x^k

\( J_i^k \) \hspace{1cm} \text{represents the ith column of} \ J^k

\( J_{ij}^k \) \hspace{1cm} \text{represents the inner product of} \ J_i^k \text{ and} \ J_j^k

\( L_i^k \) \hspace{1cm} \text{represents a lower triangular matrix - Equations 4.11, 4.35}

\( L_{i}^{k'} \) \hspace{1cm} \text{represents a lower triangular matrix - Equations 4.15, 4.41}

\( m \) \hspace{1cm} \text{represents the dimension of the projection subspace}
$M_i^k$ represents a band-diagonal matrix - Equation 4.37

$M_i^k'$ represents a band-diagonal matrix - Equation 4.42

$N_{ij}^k$ Equation 4.54

$n$ represents the dimension of the nonlinear system

$p^k$ represents the direction vector in iteration formulas

$P_i^k$ represents the $n \times n$ permutation matrix used at the $k$th iteration

$Q^k$ Equation 3.8

$\rho_i^k$ Equation 4.50

$S^k$ represents the $m \times n$ selection matrix used at the $k$th iteration

$w_k$ represents the relaxation parameter used at the $k$th iteration

$x^k$ represents the approximate solution vector at the $k$th iteration
II. INTRODUCTION AND REVIEW OF MINIMIZATION METHODS

The main motivation behind the present work is that nonlinear equations and systems of nonlinear equations arise naturally from studies of a large number of diverse problems in such fields as applied mathematics, physical sciences, engineering, economics, statistics, etc. Therefore, the solution of nonlinear systems of equations becomes very important to the understanding, development, and advances of these fields. Brown and Dennis (4, p. 186) stated that 'as more and more physical phenomena are being represented as nonlinear mathematical formulations, increasing attention is being paid to the solution of systems of nonlinear equations.'

There currently exist many methods of solving nonlinear systems of equations. They include, among others, generalizations of iterative methods for linear systems of equations such as the successive over-relaxation methods and generalizations of methods for the solution of one equation in one unknown, such as the n-dimensional extensions of Newton's method, the secant method and their variations. White (25, pp. 7-17) has provided a comprehensive review of these methods as well as of other Newton-related processes which have been developed since Newton (1669).

Another class of methods, namely minimization methods, for solving nonlinear equations has been studied for some time. These methods are reviewed in this chapter to better illustrate the similarities between them and the projection-based methods presented in this dissertation.
The problem of solving a nonlinear system of equations $F(x) = 0$ may be replaced by an equivalent problem of minimizing a functional $g: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^l$, where $g$, for instance, is a norm. Many basic ideas in minimization methods can be traced back to a paper written by Cauchy (6) in which he introduced the steepest descent method as well as some steplength algorithms. The convergence of this descent method was asserted by Cauchy himself and, 97 years later, was proven by Curry (7).

In minimization methods, the original nonlinear problem is reduced to that of finding minimizers or critical points of a given functional $g$. This leads to the following general form:

$$x^{k+1} = x^k + \alpha_k p^k,$$  \hspace{1cm} (2.1)

where

$$dx^k = -w_k \alpha_k p^k.$$ \hspace{1cm} (2.2)

$w_k$ is a relaxation parameter

$\alpha_k$ the basic steplength

and $p^k$ a direction vector.

The most commonly used methods for performing minimizations are the paraboloid methods, descent methods, conjugate-direction methods, and steplength algorithms.
Paraboloid methods approximate a nonquadratic functional \( g \) at the \( k \)th step of the iteration by a quadratic function \( g_k \) which can be minimized explicitly:

\[
g_k: \mathbb{R}^n \rightarrow \mathbb{R}, \quad g_k(x) = c - b^T x + \frac{1}{2} x^T A x
\]  

(2.3)

When \( A \) is a symmetric, positive definite matrix, \( g_k \) has a unique global minimizer \( x^* \). This \( x^* \) is the solution of the linear system

\[
g'(x)^T = Ax - b = 0
\]  

(2.4)

A few of the techniques of approximating \( g \) are listed as follows: By taking the Taylor expansion of \( g \) at \( x^k \), for \( x \) sufficiently close to \( x^k \), the following quadratic functional is obtained:

\[
g_k(x) \approx g(x^k) + g'(x^k)(x-x^k) + \frac{1}{2}(x-x^k)^T H_g(x^k)(x-x^k)
\]  

(2.5)

This represents an approximation of \( g \) near \( x^k \). Here \( H_g(x^k) \) denotes the Hessian matrix of \( g \) at \( x^k \). From equations 2.4 and 2.5, the unique global minimizer of \( g_k \) is the solution of the linear system

\[
H_g(x^k)(x-x^k) = g'(x^k)^T
\]

giving the iterative formula

\[
x^{k+1} = x^k - H_g(x^k)^{-1} g'(x^k)^T
\]  

(2.6)
This, in fact, is Newton's method for the equation \( F(x) = g'(x)^T = 0 \).

Another scheme for approximating \( g \) originally proposed by Levenberg (17) adds a scalar matrix \( \lambda_k I \) to \( H_g(x^k) \) so that \( H_g(x^k) + \lambda_k I \) is positive definite. This modification leads to the expression

\[
x^{k+1} = x^k - [H_g(x^k)^{-1} + \lambda_k I]^{-1} g'(x^k)^T\tag{2.7}
\]

which corresponds to the modified Newton method when \( F(x) = g'(x)^T \).

Schmidt and Trinkaus (24) have developed an interpolating quadratic functional for \( g \) leading to the iterative process

\[
x^{k+1} = x^k - A_k^{-1} b_k
\]

where \( A_k = ((h_i h_j)^{-1} \Delta_i \Delta_j g(x^k)) \)

\[
b_k^T = (h_1^{-1}[\Delta_1 g(x^k) - \frac{1}{2} \Delta_1^2 g(x^k)], \ldots, h_n^{-1}[\Delta_n g(x^k) - \frac{1}{2} \Delta_n^2 g(x^k)])
\]

\[
\Delta_i g(x^k) = g(x^k + h_i e^i) - g(x^k)
\]

\[
\Delta_i \Delta_j g(x^k) = g(x^k + h_i e^i + h_j e^j) - g(x^k + h_i e^i)
\]

\[
- g(x^k + h_j e^j) + g(x^k)
\]

the \( h_i \)'s and \( h_j \)'s define the \( 1+n+\frac{1}{2}n(n+1) \) interpolation points

\[
x^k, x^k + h_i e^i, i = 1, \ldots, n, x^k + h_i e^i + h_j e^j, i = 1, \ldots, n, j = 1, \ldots, n
\]
B. Descent Methods

Descent methods form a very broad and important class of minimization algorithms, in which the functional $g$ to be minimized is non-increasing, i.e.

$$g(x^{k+1}) \leq g(x^{k}), \quad k=0, 1, \ldots \quad (2.8)$$

The damped Newton method belongs to this class and has the form

$$x^{k+1} = x^k - \alpha_k F'(x^k)^{-1} F(x^k), \quad k=0, 1, \ldots \quad (2.9)$$

where $F'(x^k) \neq 0$ and the parameter $\alpha_k$ is chosen such that relation 2.8 is valid.

Univariate relaxation methods are descent methods which choose the direction vector $p^k$ in equation 2.2 from among the coordinate vectors $e^1, \ldots, e^n$ and change only 1 component of $x^k$ at each iteration such that the local decrease is maximal. Other relaxation schemes choose $p^k$ cyclically from among the coordinate vectors or, for that matter, from among any set $q^1, \ldots, q^m$ of nonzero vectors which span $\mathbb{R}^n$.

In gradient methods, the direction of $p^k$ is that of the gradient vector of $g$, i.e., $p^k = g'(x^k)^T$, while methods of steepest descent choose $-p^k$ as the direction of maximal local decrease of $g$, i.e., the direction for which $-g'(x^k)p/\|p\|$ takes on its minimum as a function of $p \neq 0$. 
The direction of steepest descent depends on the particular norm being used. The norm $\| x \| = \left( x^T C x \right)^{1/2}$ provides the iterative expression

$$x^{k+1} = x^k - \alpha_k C^{-1} g'(x^k)^T, \quad k = 0, 1, \ldots \quad (2.10)$$

If $C$ is the identity matrix, then $p = g'(x)^T$, i.e. the direction of steepest descent in the $l_2$-norm is the negative of the gradient vector.

Another descent method proposed originally by Davidon (9) and subsequently modified by Fletcher and Powell (10) has the form

$$x^{k+1} = x^k - \alpha_k B_k g'(x^k)^T \quad (2.11)$$

where the matrices $B_k$ are defined recursively by

$$B_{k+1} = B_k + \frac{r_k r_k^T}{(r_k)^T q_k} - \frac{(B_k q_k)(B_k q_k)^T}{(q_k)^T B_k q_k} \quad (2.12)$$

with $B_0$: an arbitrary, symmetric, positive definite matrix

$$r_k = x^{k+1} - x^k$$

$$q_k = g'(x^{k+1})^T - g'(x^k)^T, \quad k = 0, 1, \ldots$$
C. Steplength Algorithms

The steplength algorithms consider various ways of choosing the steplength $\alpha_k$ which is used in the general iteration

$$x^{k+1} = x^k - \alpha_k p^k, \quad k = 0, 1, \ldots$$  \hfill (2.13)

assuming that the direction vector $p^k$ is given.

Different steplength algorithms are obtained as a result of applying techniques such as the minimization principles, the Curry and Altman principles, the majorization principle, and the Goldstein principle.

If the minimization principle is applied, $\alpha_k$ is chosen so as to minimize $g$ according to one of the following three possibilities:

$$g(x^k - \alpha_k p^k) = \min \{g(x^k - \alpha_k p^k) \mid x^k - \alpha_k p^k \in D\}$$  \hfill (2.14)

$$g(x^k - \alpha_k p^k) = \min \{g(x^k - \alpha_k p^k) \mid x^k - \alpha_k p^k \in L_k^0\}$$  \hfill (2.15)

$$g(x^k - \alpha_k p^k) = \min \{g(x^k - \alpha_k p^k) \mid [x^k, x^k - \alpha_k p^k] \cap L_k^0\}$$  \hfill (2.16)

where $L_k^0$ contains $x^k$ and is the path-connected component of the level set $L_k = \{x \in D \mid g(x) \leq g(x^k)\}$.

Alternatively, for any one of the three choices, equations 2.14-2.16 above, if $x^k - \alpha_k p^k$ is an interior point of $D$, then $\alpha_k$ is a solution of the one-dimensional equation

$$g'(x^k - \alpha_k p^k)p^k = 0$$  \hfill (2.17)
If the Curry principle is applied, $\alpha_k$ is chosen to be the smallest positive root of equation 2.17. The method that results is a descent method.

The Altman principle, on the other hand, states that $\alpha_k$ is to be chosen as the smallest positive root of

$$[g'(x^k - \alpha p^k) - \mu g'(x^k)] p^k = 0 \quad (2.18)$$

where $\mu \in [0, 1]$ is fixed.

The majorization principle indicates that if there is a function $\psi: [0, \hat{a}] \rightarrow \mathbb{R}^1$ such that

$$g(x^k - \alpha p^k) \leq \psi(\alpha) < g(x^k), \forall \alpha \in (0, \hat{a}) \quad (2.19)$$

then any step length $\alpha \in (0, \hat{a})$ decreases $g$. Depending on the nonlinear system at hand, there are several ways of obtaining $\psi$. One possibility is:

$$\psi(\alpha) = g(x^k) - \alpha g'(x^k) p^k + \alpha \eta(\alpha) \quad (2.20)$$

where $\eta(\alpha) = ||p^k|| \int_0^1 w(\alpha t) ||p^k|| dt$

$w$ is the modulus of continuity of $g'$.

Furthermore, $\alpha_k$ may be chosen as that value of $\alpha \in (0, \hat{a})$ which minimizes $\psi$, that is

$$\psi(\alpha_k^*) = \min \{ \psi(\alpha) | \alpha \in (0, \hat{a}) \} \quad (2.21)$$
The Goldstein principle determines the admissible steplength set $J_k$ of $\alpha$, in which $g$ decreases:

$$J_k = \{ \alpha > 0 \mid \left[ x^k, x^k - \alpha p^k \right] \subseteq D, \sigma_2(\alpha) \leq g(x^k - \alpha p^k) \leq \sigma_1(\alpha) \} \quad (2.22)$$

where $\sigma_1(\alpha)$ and $\sigma_2(\alpha)$ are defined by:

$$\sigma_1(\alpha) = g(x^k) - \mu_1 g_i'(x^k) p^k, \ i = 1, 2$$

$$\sigma_2(\alpha) = g(x^k) - \mu_2 g_i'(x^k) p^k, \ i = 1, 2$$

and $\mu_1$, $\mu_2$ are arbitrary constants satisfying $0 < \mu_1 < \mu_2 < 1$.

Since any choice of $\alpha_k \in J_k$ will decrease $g$, the method resulting from an application of the Goldstein principle is a descent method.

D. Conjugate-Direction Methods

Conjugate-direction methods were first used to minimize the quadratic function in equation 2.3 above. A scheme to generate the $x^j$ and $p^j$ simultaneously is provided by the following conjugate-gradient algorithm

$$x^{k+1} = x^k - \alpha_k p^k, \ \alpha_k = (Ax^k - b)^T p^k / (Ap^k)^T p^k \quad (2.24)$$

$$p^0 = Ax^0 - b, \ p^{k+1} = Ax^{k+1} - b - \beta_k p^k$$

$$\beta_k = (Ax^{k+1} - b)^T (Ap^k)^T p_k$$

where the nonzero vectors $p^j, j = 0, 1, \ldots$ form a conjugate basis, and $\alpha_k$, the steplength, is obtained by choosing
The fundamental result of the conjugate-direction methods, when $g$ is quadratic, is that the iterates $\{x^k\}$ in equation 2.24 converge to the unique minimizer of $g$ in at most $n$ steps.

The conjugate-gradient algorithm defined in equations 2.24 and 2.25 has been extended to the case of nonquadratic functionals in the Daniel algorithm (8), the Fletcher-Reeves algorithm (11), and the method of Rosenbrook (23).

From the above review, we can easily see that particular minimization methods are quite complicated. The convergence of most of the methods available is slow compared to Newton's method (that is, when the latter converges). Efforts to remedy these shortcomings can be found in Al'ber and Al'ber (1), Bard (2), Luenberger (18), Broyden (5), and Brown (3).

To complete this review, there are several excellent bibliographic sources on minimization methods worth mentioning. They are Householder (14), Kantorovich (16), and Ortega and Rheinboldt (21).

The nonlinear projection methods introduced in the next chapter belong to the descent class of minimization methods. These projection methods serve as starting points for the development of projection-based methods which are presented in this dissertation. The objective of the projection-based methods is to develop a class of iterative methods which are relatively simple, fast to converge, and yet general enough to solve a large spectrum of nonlinear problems.
III. THE NONLINEAR PROJECTION METHODS

The projection methods are minimization methods first introduced by A. de la Garza (12) to solve linear equations and have recently been extended to nonlinear problems by MacEachern and Keller (19) and White (25).

Given a nonlinear system \( F(x) = 0 \), where \( F: D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n \) and each component function \( f_i \) of \( F(x) \) is such that the derivatives \( f'_i \) and \( f''_i \) exist and are continuous for all \( x \) in a compact set \( D_0 \subseteq D \), then an \( m \)-dimensional projection method (\( 1 \leq m \leq n \)) at step \( k \) of the iterative process changes \( m \) components of the approximate solution vector \( x^k \) and is developed as follows:

By using the Taylor's expansion about the point \( x^k \) and truncating all second and higher order terms, an approximation for each \( f_i(x^{k+1}) \) is obtained

\[
f_i(x^{k+1}) = f_i(x^k) + (dx^k, J_{f_i})_i, \quad i = 1, 2, \ldots, n
\]

(3.1)

where \( dx^k = x^{k+1} - x^k \)

(3.2)

\( J_{f_i} \) in the inner product \( (dx^k, J_{f_i})_i \) is the \( i \)th row of the Jacobian matrix (evaluated at \( x^k \)) of the system.

The \( m \) components of \( x^k \) which are to be modified can be denoted by the set \( T = \{i_1, i_2, \ldots, i_m\} \). Then by noting that \( dx^k_u = 0 \) for \( u \notin T \), equation 3.1 could be simplified further to give
\[ f_i(x^{k+1}) = f_i(x^k) + \sum_{v \in T} \frac{\delta f_i}{\delta x_v} \Delta x_v(x^k), \quad i = 1, 2, \ldots, n \] (3.3)

where \( \Delta x_v \) is the \( v \)th component of \( \Delta x \) and

\[ \frac{\delta f_i}{\delta x_v} = f_{ix_v} \]

\( \Delta x \) is then determined by maximizing the difference between the squares of the two Euclidean norms \( \| F(x^k) \|^2 \) and \( \| F(x^{k+1}) \|^2 \)

\[ \| F(x^k) \|^2 - \| F(x^{k+1}) \|^2 = \sum_{i=1}^{n} (f_i(x^k))^2 - \sum_{i=1}^{n} (f_i(x^{k+1}))^2 \]

Equation 3.4 is differentiated with respect to \( \Delta x_v \). After setting the resulting expressions to zero, \( \Delta x_v, \forall v \in T \), can be obtained directly by solving the following set of linear equations

\[ \sum_{v \in T} (J_v^k, J_s^k) \Delta x_v = -(F(x^k), J_s^k), \quad \forall s \in T \] (3.5)

where \( J_v^k \) and \( J_s^k \) are the \( v \)th and \( s \)th columns of the Jacobian matrix evaluated at \( x^k \). For all \( i \notin T \), \( \Delta x_i^k = 0 \).

The equation 3.5 is expanded and put into matrix notation to include the unchanged components of \( \Delta x \). This results in:

\[ \Delta x = x^{k+1} - x^k = -(S^k)^T(q^k)^{-1}V(x^k) \] or

\[ x^{k+1} = x^k - (S^k)^T(q^k)^{-1}V(x^k) \] (3.7)
where $S^k$ is an $m \times n$ selection matrix used at the $k$th iteration step. The $m$ rows of $S^k$ are those rows of the $n \times n$ identity matrix which correspond to the $m$ changed components of the approximate solution vector $x^k$. $Q^k$ is an $m \times m$ matrix evaluated at $x^k$. 

\[
Q^k = \begin{bmatrix}
(j^k_{i_1, j^k_{i_1}}, (j^k_{i_1, j^k_{i_2}}), \ldots, (j^k_{i_1, j^k_{i_m}}) \\
(j^k_{i_2, j^k_{i_1}}, ) \\
\vdots \\
(j^k_{i_m, j^k_{i_1}}, (j^k_{i_1, j^k_{i_m}})
\end{bmatrix}
\]

Each element of $Q^k$ is an inner product of two columns of the Jacobian matrix.

Each of the $m$ elements of the column vector $V(x^k)$ is an inner product of $F(x^k)$ with a column of the Jacobian:

\[
V(x^k) = (j^k_{i_1}, F(x^k))
\]

Equation 3.9 could be rewritten in another form:
\[ V(x^k) = S^k (j^k)^T F(x^k) \]  

(3.10)

where \( j^k \) is the Jacobian matrix evaluated at \( x^k \).

Using 3.10 in equation 3.7:

\[ x^{k+1} = x^k - (S^k)^T (Q^k)^{-1} S^k (j^k)^T F(x^k) \]

\[ x^{k+1} = x^k - A^k (j^k)^T F(x^k) \]  

(3.11)

where \( A^k = (S^k)^T (Q^k)^{-1} S^k \)  

(3.12)

Let

\[ g(x) = \| F(x) \|^2 \]  

(3.13)

Then

\[ g'(x^k) = 2F(x^k)^T j^k \]  

(3.14)

Equations 3.11 and 3.14 provide the iterative expression:

\[ x^{k+1} = x^k - \frac{1}{2} A^k g'(x^k)^T \]  

or

\[ x^{k+1} = x^k - \frac{1}{2} p^k \]  

(3.15)

(3.16)

where the direction vector \( p^k \) is assumed to be nonzero:

\[ p^k = A^k g'(x^k)^T \]  

(3.17)
Projection methods also have an interesting geometric interpretation. In addition to modifying $m$ components of the approximate solution vector $x^k$, the application of an $m$-dimensional projection forces the residue vector $F(x^{k+1})$ to be perpendicular to the $m$ columns $i_1, i_2, \ldots, i_m$ of the Jacobian matrix evaluated at $x^k$. The name projection method thus arises from the fact that during the application of an $m$-dimensional projection method, the residue is projected on a subspace of dimension $m$. 
IV. NONLINEAR PROJECTION-BASED METHODS

The rate of convergence of the projection methods is relatively slow. As equation 3.11 shows, they are generally more complex than Newton-type methods. Moreover, in order to change each component of the approximate solution vector at least once, an m-dimensional projection method requires that both the Jacobian matrix and the residue vector be evaluated \( \lceil \frac{n}{m} \rceil \) times, where the "ceiling function" \( \lceil \frac{n}{m} \rceil \), represents the smallest integer which is equal to or greater than \( \frac{n}{m} \).

The projection-based methods presented in this chapter calculate the Jacobian matrix and the residue vector only once per cycle; i.e., each component of the approximate solution vector is modified at least once before the Jacobian and the residue are evaluated again. The basic idea behind the projection-based methods is to initially proceed as in the regular projection methods. Then, instead of reevaluating the Jacobian and the residue at every iteration step, both the Jacobian matrix and the residue vector are approximated for the next steps until all n components of the approximate solution vector have been modified. This results in the class of iterative methods developed in sections IV. A., IV. B., IV. C., and IV. D. below. In these sections, without loss of generality, we initially assume that the successive components of the approximate solution vector which are to be changed follow the order 1, 2, 3, ..., n.
A. 1-Dimensional Projection-Based Methods

Using a 1-dimensional nonlinear projection method at step $k$, the change $dx^k_1$ in the component $x^k_1$ of the approximate solution vector $x^k$ is given by:

$$-dx^k_1 = \frac{(F(x^k), J^k_1)}{J^k_{11}}$$

(4.1)

where $J^k_1$ is the first column of the Jacobian matrix $J^k$ evaluated at $x^k$. $J^k_{11}$ is the inner product $(J^k_1, J^k_1)$.

Equation 4.1 could be rewritten as:

$$-dx^k_1 = \frac{(F(x^k), a^k_1)}{J^k_{11}} = \frac{(a^k_1)^T F(x^k)}{J^k_{11}}$$

(4.2)

where the column vector $a^k_1 = J^k_1$.

Without reevaluating $F(x^k)$ and $J^k$ and still using the 1-dimensional nonlinear projection method, the next component change $dx^k_2$ is:

$$-dx^k_2 = \frac{(F(x^k) - dx^k_1 J^k_1, J^k_2)}{J^k_{22}}$$

(4.4)

$$= \frac{(F(x^k), J^k_2) - (J^k_1, J^k_2) dx^k_1}{J^k_{22}}$$
\[ \begin{align*}
  &\left[(F(x^k), j^k_2) - \frac{\mathbf{J}^k_{12} \cdot \mathbf{F}(x^k), \alpha^k_1)}{\mathbf{J}^k_{11}} \right]/j^k_{22} \\
  &= \frac{(\mathbf{F}(x^k), \alpha^k_2)}{\mathbf{J}^k_{22}} = \frac{(\alpha^k_2)^T \mathbf{F}(x^k)}{\mathbf{J}^k_{22}} \\
  \text{where } \alpha^k_2 &= j^k_2 - \frac{\mathbf{J}^k_{12}}{\mathbf{J}^k_{11}} \alpha^k_1
\end{align*}\] (4.5)

\[ J^k_{ij} \] is the inner product of the \( i \)th column with the \( j \)th column of the Jacobian matrix \( J^k \). Note that \( J^k_{ij} = J^k_{ji} \), \( i=1, 2, \ldots, n \), \( j=1, 2, \ldots, n \).

Continuing in this manner, the component change \( dx^k_n \) is obtained. \( ^1 \)

\[ -dx^k_n = \frac{(\mathbf{F}(x^k) - dx^k_1 J^k_{11} - dx^k_2 J^k_{22} \cdots - dx^k_{n-1} J^k_{n-1} J^k_{n-1} J^k_{n})}{\mathbf{J}^k_{nn}} \] (4.7)

\[ = \frac{(\mathbf{F}(x^k) - (\mathbf{F}(x^k), \alpha^k_1) j^k_1 - (\mathbf{F}(x^k), \alpha^k_2) j^k_2}{\mathbf{J}^k_{11}} - \cdots \]

\( ^1 \)A comma sometimes separates the subscripts of \( J^k \). This is used only for clarity, i.e., \( j^k_{i,j} = J^k_{ij} \). This also applies to \( \alpha^k_{ij} \).
Equations 4.3, 4.6, and 4.9 could be rewritten as:

\[ \alpha_n^k = J_{nn}^k \]

\[ \frac{J_{12}^k}{J_{11}^k} \alpha_1^k + \alpha_2^k = J_{22}^k \]
\[
\frac{j_{1n}^k}{J_{11}^k} \alpha_1^k + \frac{j_{2n}^k}{J_{22}^k} \alpha_2^k + \ldots + \frac{j_{n-1,n}^k}{J_{n-1,n-1}^k} \alpha_{n-1}^k + \alpha_n^k = j_n^k
\]

In matrix notation (T denotes the transpose):

\[
\begin{pmatrix}
1 \\
\frac{j_{12}^k}{J_{11}^k} \\
\vdots \\
\frac{j_{1n}^k}{J_{11}^k}
\end{pmatrix}
\begin{pmatrix}
\phi
\end{pmatrix}
= 
\begin{pmatrix}
(\alpha_1^k)^T \\
(\alpha_2^k)^T \\
\vdots \\
(\alpha_n^k)^T
\end{pmatrix}
= 
\begin{pmatrix}
(j_1^k)^T \\
(j_2^k)^T \\
\vdots \\
(j_n^k)^T
\end{pmatrix}
\]

or

\[
\begin{pmatrix}
\frac{j_{11}^k}{J_{11}^k} \\
\frac{j_{12}^k}{J_{22}^k} \\
\vdots \\
\frac{j_{1n}^k}{J_{2n}^k}
\end{pmatrix}
\begin{pmatrix}
\phi
\end{pmatrix}
= 
\begin{pmatrix}
(\alpha_1^k)^T \\
(\alpha_2^k)^T \\
\vdots \\
(\alpha_n^k)^T
\end{pmatrix}
= (j^k)^T \quad (4.10)
\]

where \( j^k \) is the Jacobian matrix evaluated at \( x^k \).
Let

\[ L_{\perp}^k = \begin{bmatrix}
  j_{11}^k & \phi \\
  j_{21}^k & j_{22}^k \\
  \vdots & \vdots \\
  j_{n1}^k & j_{n2}^k & \cdots & j_{nn}^k
\end{bmatrix} \]  

(4.11)

\[ \alpha^{k'} = \begin{bmatrix}
  (\alpha_1^k)^T \\
  j_{11}^k \\
  \vdots \\
  (\alpha_{n'}^k)^T \\
  j_{nn}^k
\end{bmatrix} \]  

(4.12)

then

\[ \alpha^{k'} = (L_{\perp}^k)^{-1} (j^k)^T \]  

(4.13)
From equations 4.2, 4.5, and 4.8:

\[
-d\mathbf{x}^k = (\mathbf{L}_1^k)^{-1} (\mathbf{j}^k)^T \mathbf{F}(\mathbf{x}^k)
\]

or

\[
\mathbf{d}x^k = -(\mathbf{L}_1^k)^{-1} (\mathbf{j}^k)^T \mathbf{F}(\mathbf{x}^k)
\]  \quad (4.14)

where \(d\mathbf{x}^k\) is the column vector \(\mathbf{x}^{k+1} - \mathbf{x}^k\).

In the development above, the order of the changes in the components of \(\mathbf{x}^k\) is assumed to be 1, 2, ..., n. In the general case where that order is \(i_1, i_2, ..., i_n\), the matrix \(\mathbf{L}_1^k\) in equation 4.14 is replaced by \(\mathbf{L}_1^{i_1}\).
Also, a permutation matrix is needed to rearrange the rows of 
\((J^k)^T\) and to permute the components of \(dx^k\). Then, equation 4.14 becomes

\[
dx^k = -(P_{i_1}^k)^T (L_{i_1}^k)^{-1} P_{i_1}^k (J^k)^T F(x^k)
\]

or

\[
x^{k+1} = x^k - (P_{i_1}^k)^T (L_{i_1}^k)^{-1} P_{i_1}^k (J^k)^T F(x^k)
\]

(4.16)

where \(P_{i_1}^k\) is the \(n \times n\) permutation matrix used at step \(k\). The \(n\) rows of 
\(P_{i_1}^k\) are the \(n\) rows of the identity matrix, but correspond to the order 
\(i_1, i_2, \ldots, i_n\). That is, the first row of \(P_{i_1}^k\) is the \(i_1\)th row of the 
identity matrix, the second row of \(P_{i_1}^k\) is the \(i_2\)th row of the identity 
matrix, etc.

Equation 4.16 is the iterative expression for the 1-dimensional 
projection-based methods to solve nonlinear systems of equations. 
These methods are total step methods, i.e., one step of a 1-dimensional 
projection-based method modifies all \(n\) components of the approximate 
solution vector \(x^k\).
B. 2-Dimensional Projection-Based Methods

Using a 2-dimensional nonlinear projection method at step \( k \) and proceeding in a fashion similar to that in part IV. A., we obtain the following changes \( dx_1^k \) and \( dx_2^k \) in the first two components of \( x^k \):

\[
-dx_1^k = \frac{(F(x^k), J_1^k) - (F(x^k), J_2^k)}{J_{11}^k J_{22}^k - (J_{12}^k)^2}
\]

\[
= \frac{(F(x^k), \beta_{12}^k) - (\beta_{12}^k)^T F(x^k)}{a_{12}^k} (4.17)
\]

where the column vector \( \beta_{12}^k = J_{22}^k J_{11}^k - J_{12}^k J_{21}^k \) (4.18)

the scalar value \( a_{12}^k = J_{11}^k J_{22}^k - (J_{12}^k)^2 \) (4.19)

\[
-dx_2^k = \frac{(\beta_{22}^k)^T F(x^k)}{a_{12}^k} (4.20)
\]

where the column vector \( \beta_{22}^k = J_{12}^k J_{22}^k - J_{12}^k J_{21}^k \) (4.21)

The next component changes \( dx_3^k \) and \( dx_4^k \) are obtained without re-evaluating the Jacobian matrix and by approximating the next residue vector with \( F(x^k) - dx_1^k J_{11}^k - dx_2^k J_{22}^k \).
\[
-dx_k^3 = \frac{(F(x^k) - dx_4^k \cdot \frac{1}{a_{12}} - \frac{1}{a_{11}} - \frac{1}{a_{44}} - \frac{1}{a_{33}} - \frac{1}{a_{34}} - \frac{1}{a_{44}} - \frac{1}{a_{34}})}{a_{34}}
\]

where the scalar value \( a_{34} = \frac{1}{a_{33}} \cdot \frac{1}{a_{44}} - (\frac{1}{a_{34}})^2 \) \((4.23)\)

Using equations 4.17 and 4.20 to substitute for \(dx^4_1\) and \(dx^4_2\) in equation 4.22:

\[
-dx_k^3 = \frac{(F(x^k), J^k_2 - J^k_4, J^k_3 - J^k_4, J^k_3 - J^k_4)}{a_{12}}
\]

\[
-dx_k^3 = \frac{(F(x^k), J^k_4, J^k_3 - J^k_4)}{a_{12}} - \frac{(F(x^k), J^k_4, J^k_3 - J^k_4)}{a_{12}}
\]

\[
-dx_k^3 = \frac{(F(x^k), J^k_4, J^k_3 - J^k_4)}{a_{12}} - \frac{(F(x^k), J^k_4, J^k_3 - J^k_4)}{a_{12}}
\]

\[
-dx_k^3 = \frac{(F(x^k), J^k_4, J^k_3 - J^k_4)}{a_{12}} - \frac{(F(x^k), J^k_4, J^k_3 - J^k_4)}{a_{12}}
\]

\[
=d\frac{(F(x^k), J^k_4, J^k_3 - J^k_4)}{a_{12}}
\]

\[
= \frac{(F(x^k), J^k_4, J^k_3 - J^k_4)}{a_{12}}
\]

where the column vector \(\beta_3^k\) is given by

\[
= \frac{(\beta_3^k)^T F(x^k)}{a_{34}}
\]

\((4.24)\)
\[ \begin{align*}
\beta_3^k &= j_{44}^k j_{34}^k - j_{34}^k j_{44}^k - \frac{j_{23}^k}{a_{12}} - j_{34}^k j_{34}^k - \frac{j_{44}^k}{a_{12}} \beta_1^k \quad (4.25) \\
\end{align*} \]

Similarly:
\[ \begin{align*}
-dx_4^k &= (\beta_4^k)^T \frac{F(x^k)}{a_{34}} \\
&(4.26) \\
\end{align*} \]

where the column vector \( \beta_4^k \) is
\[ \begin{align*}
\beta_4^k &= j_{33}^k j_{34}^k - j_{34}^k j_{33}^k - \frac{j_{34}^k}{a_{12}} j_{23}^k - \frac{j_{34}^k}{a_{12}} \beta_2^k \quad (4.27) \\
\end{align*} \]

The expressions for the component changes \( dx_{n-1}^k \) and \( dx_n^k \) are obtained by continuing in the same manner and assuming that \( n \) is even:
\[ \begin{align*}
-dx_{n-1}^k &= \left( F(x^k) - dx_{n-1}^k j_{n-1}^k - \cdots - dx_{n-2}^k j_{n-2}^k \right) \left[ j_{nn}^k - j_{n-l,n}^k \right] \frac{j_{nn}^k}{a_{n-1,n}} \left( J_{n-1,n}^k \right)^2 \\
&(4.28) \\
\end{align*} \]

where the scalar value \( a_{n-1,n} = j_{n-1,n}^k - j_{n-1,nn}^k \left( J_{n-1,n}^k \right)^2 \) \( (4.29) \)
After substituting $dx^k_{n-2}$, ..., $dx^k_1$ with the corresponding expressions
\[ \frac{(F(x^k), \beta^k_{n-2})}{a_{n-3,n-2}}, \ldots, \frac{(F(x^k), \beta^k_1)}{a_{12}} \]
(note that $a_{ij} = a_{ji}$) and rearranging the terms, equation 4.28 gives:

\[ -dx^k_{n-1} = \left( \frac{F(x^k), \beta^k_{n-1}}{a_{n-1,n}} \right) = \frac{(\beta^k_{n-1})^T F(x^k)}{a_{n-1,n}} \quad (4.30) \]

where the column vector $\beta^k_{n-1}$ is

\[ \beta^k_{n-1} = J^k_n \beta^k_j - J^k_{n-1,n} \beta^k_j - \frac{J^k_{nn,n-2,n-1} - J^k_{n-1,n} J^k_{n,n-2,n}}{a_{n-3,n-2}} \beta^k_{n-2} \]
\[ \ldots - \frac{J^k_{nn,n,n-1} - J^k_{n-1,n} J^k_{n,n-1,n}}{a_{12}} \beta^k_1 \quad (4.31) \]

Also:

\[ -dx^k_n = \frac{(\beta^k_n)^T F(x^k)}{a_{n-1,n}} \quad (4.32) \]

with

\[ \beta^k_n = J^k_{n,n-1,n-1} \beta^k_n - J^k_{n-1,n} \beta^k_n - \frac{J^k_{n,n-1,n} J^k_{n-2,n} - J^k_{n-1,n} J^k_{n-2,n,n-1}}{a_{n-3,n-2}} \beta^k_{n-2} \]
The equations for the $\beta_k^1$'s 4.18, 4.21, 4.25, 4.27, 4.31, and 4.33 could be rewritten as:

$$\beta_k^1 = j_k^1 - j_k^2$$

$$\beta_k^2 = j_k^3 - j_k^4$$

$$\beta_k^3 = j_k^3 - j_k^4$$

$$\beta_k^4 = j_k^3 - j_k^4$$

$$\beta_k^5 = j_k^3 - j_k^4$$

$$\beta_k^n = j_k^n - j_k^n$$
The above equations could be put in matrix notation:

\[
L^k_{2} k' = N^k_2 \left( J^k \right)^T
\]

(4.34)

where

\[
L^k_2 = \begin{bmatrix}
a_{12} & 0 & \cdots & 0 \\
0 & a_{12} & \cdots & 0 \\
J^k_{1,13} - J^k_{3,14} & J^k_{1,23} - J^k_{3,24} & \cdots & 0 \\
J^k_{3,14} & J^k_{3,14} & \cdots & 0 \\
J^k_{n,n-1} & J^k_{n,n} & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots \\
J^k_{n-1,n} & J^k_{n-1,n} & \cdots & 0
\end{bmatrix}
\]

(4.35)
\[ \mathbf{y}^T = \begin{bmatrix} \frac{\mathbf{b}_1^T}{a_{12}} \\ \frac{\mathbf{b}_2^T}{a_{12}} \\ \vdots \\ \frac{\mathbf{b}_{n-1}^T}{a_{n-1,n}} \\ \frac{\mathbf{b}_n^T}{a_{n-1,n}} \end{bmatrix} \quad \text{(4.36)} \]
From equations 4.17, 4.20, 4.24, 4.26, 4.30, and 4.32:
Then the change vector $dx^k$ is obtained by combining equations 4.34 and 4.39:

$$dx^k = -(L_2^k)^{-1} M_2^k (J^k)^T F(x^k)$$  \hspace{1cm} (4.40)

Here $L_2^k$ is a triangular $n \times n$ matrix of which half of the number of elements are zero. $M_2^k$ is a sparse $n \times n$ matrix which has $2 \times n$ nonzero elements.

If the order of the changed components of $x^k$ is $i_1, i_2, \ldots, i_n$ instead of 1, 2, \ldots, $n$, then $L_2^k$ and $M_2^k$ in equation 4.40 are replaced by $L_2^{k'}$ and $M_2^{k'}$ respectively:
\[
\mathbf{L}_2^k = \begin{bmatrix}
- \mathbf{J}_{i_1 i_2} & \mathbf{J}_{i_1 i_2} \\
- \mathbf{J}_{i_1 i_2} & - \mathbf{J}_{i_1 i_2} \\
& \ddots \\
& & \mathbf{J}_{i_1 i_2} & \mathbf{J}_{i_1 i_2} \\
& & & \ddots \\
& & & & \mathbf{J}_{i_1 i_2} & \mathbf{J}_{i_1 i_2} \\
& & & & & \mathbf{J}_{i_1 i_2} \\
& & & & & & \mathbf{J}_{i_1 i_2} \\
\end{bmatrix}
\]

(4.41)

\[
\mathbf{M}_2^{k'} = \begin{bmatrix}
- \mathbf{J}_{i_1 i_2} & \mathbf{J}_{i_1 i_2} \\
- \mathbf{J}_{i_1 i_2} & - \mathbf{J}_{i_1 i_2} \\
& \ddots \\
& & \mathbf{J}_{i_1 i_2} & \mathbf{J}_{i_1 i_2} \\
& & & \ddots \\
& & & & \mathbf{J}_{i_1 i_2} & \mathbf{J}_{i_1 i_2} \\
& & & & & \mathbf{J}_{i_1 i_2} \\
& & & & & & \mathbf{J}_{i_1 i_2} \\
\end{bmatrix}
\]

(4.42)
Again a permutation matrix is needed to rearrange the rows of 
\((J^k)^T\) and to permute the components of \(dx^k\) in equation 4.40:

\[
dx^k = -(P_2^k)^T (L_2^k)^{-1} M_2^k P_2^k (J^k)^T F(x^k)
\]

or \[x^{k+1} = x^k - (P_2^k)^T (L_2^k)^{-1} M_2^k P_2^k (J^k)^T F(x^k)\] (4.43)

where \(P_2^k\) is the \(n \times n\) permutation matrix used at step \(k\). The \(n\) rows of \(P_2^k\) are the \(n\) rows of the identity matrix and correspond to the order \(i_1, i_2, \ldots, i_n\).

Note that the subscript \(i\) of \(P_i^k, L_i^k, L_{i1}\), \(M_i^k, M_{i1}\) is the dimension of the projection methods on which the new algorithms are based.

Equation 4.43 is the iteration formula for the 2-dimensional projection-based methods to solve a system of \(n\) nonlinear equations. These methods, like the 1-dimensional projection-based expressions, are total step methods. In fact, all the projection-based methods presented in this chapter are total step methods.

\[\text{C. 3-Dimensional Projection-Based Methods}\]

Here, the starting equations are those used in the 3-dimensional projection methods:

\[-dx_1^k = \frac{1}{d_{123}} \left[ (F(x^k), J_{11}^k, J_{12}^k, J_{13}^k, J_{21}^k, J_{22}^k, J_{23}^k, J_{31}^k, J_{32}^k, J_{33}^k) \right] + (F(x^k), J_2^k)\]
\[
\begin{align*}
(J_{13}^{j_k} j_{23}^{j_k} - J_{12}^{j_k} j_{33}^{j_k}) + (F(x^k), \gamma_{1}^{j_k}) (J_{23}^{j_k} j_{12}^{j_k} - J_{13}^{j_k} j_{22}^{j_k})
\end{align*}
\]

where the scalar value \(d_{123}\) is

\[
\begin{align*}
d_{123} &= j_{11}^{j_k} j_{22}^{j_k} j_{33}^{j_k} + 2 (j_{23}^{j_k} j_{13}^{j_k} j_{12}^{j_k} - (j_{13}^{j_k})^2 j_{22}^{j_k} - (j_{23}^{j_k})^2 j_{11}^{j_k} - \\
&\quad (j_{12}^{j_k})^2 j_{33}^{j_k}
\end{align*}
\]

and the column vector \(\gamma_{1}^{j_k}\)

\[
\begin{align*}
\gamma_{1}^{j_k} &= (j_{33}^{j_k} j_{22}^{j_k} - (j_{23}^{j_k})^2) j_{11}^{j_k} + (j_{13}^{j_k} j_{23}^{j_k} - j_{12}^{j_k} j_{33}^{j_k}) j_{22}^{j_k} \\
&\quad + (j_{23}^{j_k} j_{12}^{j_k} - j_{13}^{j_k} j_{23}^{j_k}) j_{33}^{j_k}
\end{align*}
\]

Expressions similar to equation 4.44 are obtained for \(d_{12}^{j_k}, \ldots, d_{n}^{j_k}\) by continuing as in sections IV. A. and IV. B. Also equations for \(\gamma_{2}^{j_k}, \ldots, \gamma_{9}^{j_k}\) similar to 4.46 are generated and lead to the following formula:

\[
\begin{align*}
L_{3}^{j_k} \gamma_{1}^{j_k} &= M_{3}^{j_k} (j_{3}^{j_k})^T
\end{align*}
\]
where, assuming that \( n \) is a multiple of 3:

\[
L^k_3 = \begin{bmatrix}
\begin{array}{ccc}
\d_{123} & & \\
0 & \d_{123} & \\
0 & 0 & \d_{123}
\end{array}
\end{bmatrix}
\begin{bmatrix}
\begin{array}{ccccccc}
(j^k_1, \rho_4) & (j^k_2, \rho_4) & (j^k_3, \rho_4) & 0 & 0 & 0 & d_{456} \\
(j^k_1, \rho_5) & (j^k_2, \rho_5) & (j^k_3, \rho_5) & 0 & d_{456} & & \\
(j^k_1, \rho_6) & (j^k_2, \rho_6) & (j^k_3, \rho_6) & 0 & 0 & d_{456} & \\
(j^k_1, \rho_7) & (j^k_2, \rho_7) & (j^k_3, \rho_7) & (j^k_4, \rho_7) & (j^k_5, \rho_7) & (j^k_6, \rho_7) & d_{789} \\
(j^k_1, \rho_8) & (j^k_2, \rho_8) & (j^k_3, \rho_8) & (j^k_4, \rho_8) & (j^k_5, \rho_8) & (j^k_6, \rho_8) & 0 & d_{789} \\
(j^k_1, \rho_9) & (j^k_2, \rho_9) & (j^k_3, \rho_9) & (j^k_4, \rho_9) & (j^k_5, \rho_9) & (j^k_6, \rho_9) & 0 & 0 & d_{789}
\end{array}
\end{bmatrix}
\]

etc.
\[ d_{ijl} = j_{il}^k j_{jj}^k j_{jj}^k + 2(j_{il}^k j_{jj}^k j_{jj}^k) - (j_{il}^k)^2 j_{jj}^k - (j_{il}^k j_{jj}^k)^2 j_{il}^k \]

\[-(j_{il}^k)^2 j_{il}^k, \quad ijl = 123, 456, \ldots \] (4.49)

\[ p_{ij}^k = (j_{il}^k j_{jj}^k - (j_{il}^k)^2) j_{il}^k + (j_{il}^k j_{jj}^k - j_{ij}^k j_{jl}^k) j_{il}^k \]
\[ + (j_{il}^k j_{jj}^k - j_{il}^k j_{jj}^k) j_{il}^k, \quad ijl = 456, 564, 645, 789, 897, \ldots \] (4.50)

\[ \gamma^{k'} = \begin{bmatrix}
\frac{(\gamma_{1}^k)^T}{d_{123}} \\
\frac{(\gamma_{2}^k)^T}{d_{123}} \\
\frac{(\gamma_{3}^k)^T}{d_{456}} \\
\frac{(\gamma_{n}^k)^T}{d_{n-2n-1,n}}
\end{bmatrix} \] (4.51)
\[ M^k_{ij} = J^k_{ij} - (J^k_{ij})^2, \quad i,j = 123, 231, 312, 456, \ldots \]  \hspace{1cm} (4.52)

\[ M^k_{ij} = J^k_{ij} - J^k_{i'j'}, \quad i,j = 123, 231, 312, 456, \ldots \]  \hspace{1cm} (4.53)
Next, the change vector $dx^k$ is derived from equations 4.44, 4.47, and 4.51:

$$dx^k = -(L_3^k)^{-1} M_3^k (J^k)^T F(x^k)$$  \hspace{1cm} (4.55)

Here, $L_3^k$ is a triangular $n \times n$ matrix with more than half of the number of its elements equal to zero. $M_3^k$ is a symmetric band-diagonal matrix which has $3 \ast n$ nonzero elements.

Again, if the order of the changes in $x^k$ is $i_1, i_2, \ldots, i_n$, then $L_3^k$ and $M_3^k$ in equation 4.55 are replaced by $L_3^{k'}$ and $M_3^{k'}$ respectively. $L_3^{k'}$ and $M_3^{k'}$ are obtained from $L_3^k$ and $M_3^k$ by replacing each subscript $v$ in equations 4.48 and 4.52 by $v_i$, where $v = 1, 2, \ldots, n$. Also, a permutation matrix $P_3^k$ is needed and the more general form of equation 4.55 is:

$$dx^k = -(P_3^k)^T (L_3^{k'})^{-1} M_3^{k'} P_3^k (J^k)^T F(x^k)$$

or

$$x^{k+1} = x^k - (P_3^k)^T (L_3^{k'})^{-1} M_3^{k'} P_3^k (J^k)^T F(x^k)$$  \hspace{1cm} (4.56)

where the $n$ rows of $P_3^k$ are obtained from the identity matrix and correspond to the order $i_1, i_2, \ldots, i_n$.

Equation 4.56 is the iterative expression for the 3-dimensional projection-based methods to solve nonlinear systems of equations.
D. M-Dimensional Projection-Based Methods

From the above development, it is clear that the iterative expression for total step methods which are based on m-dimensional projection methods is given by:

\[ x^{k+1} = x^k - (P^m)^T (L^m)^{-1} M^m P^m (J^m)^T F(x^k) \]  \hspace{1cm} (4.57)

where \( L^m \) is an \( n \times n \) triangular matrix similar to \( L^2 \) and \( L^3 \) in equations 4.41 and 4.48. \( M^m \) is a symmetric matrix which has \( m \times n \) nonzero elements. \( P^m \) is the permutation matrix reflecting the order in which the components of \( x^k \) are changed.

In particular, when \( m = n \) the iteration formula for the projection-based method is the same as that for the nonlinear projection method given by equation 3.11. To rewrite equation 3.11 in a more explicit form for the case \( m = n \), consider equation 3.8 for the matrix \( Q^k \).

When \( m = n \):

\[ Q^k = (J^k)^T J^k \] \hspace{1cm} (4.58)

and \( S^k \) in equation 3.12 becomes the identity matrix. Therefore, equation 3.11 gives:

\[ x^{k+1} = x^k - [(J^k)^T J^k]^{-1} (J^k)^T F(x^k) \] \hspace{1cm} (4.59)
As stated above, equation 4.59 is also the iteration formula for the n-dimensional projection-based method to solve systems of n nonlinear equations in n unknowns.
V. CONVERGENCE

A. Proof of Convergence for the 1-Dimensional Projection-Based Methods

Let \( g(x) = \|F(x)\|^2 \) as in equation 3.13, then

\[
g'(x^k) = 2F(x^k)^T J_k \quad (3.14)
\]

Equation 4.16 represents the iteration formula for the 1-dimensional projection-based methods and could be rewritten as follows:

\[
x^{k+1} = x^k - \frac{1}{\lambda_k} g'(x^k)^T
\]

or

\[
x^{k+1} = x^k - \frac{1}{\lambda_k} p^k
\]

where

\[
A_k^l \equiv (P_k^l)^T (I^l_k)^{-1} p_k^l
\]

\[
p^k = A_k^l g'(x^k)^T
\]

The direction vector \( p^k \) is assumed to be nonzero.

The proof of convergence of the expression 5.2 requires the following three intermediate results:

- \( g \) as defined by 3.13 is a strictly convex functional;

\[
-g'(x^k)p^k \geq 0 \quad \forall k \geq k_0 \quad (5.5)
\]

\[
\lim_{k \to \infty} \frac{g'(x^k)p^k}{\|p^k\|^2} = 0 \quad (5.6)
\]
These intermediate results are obtained respectively in definition 5.1 through theorem 5.6, theorem 5.9 through theorem 5.12, and definition 5.13 through theorem 5.16.

Definition 5.1:
A functional $g: D \subseteq \mathbb{R}^n \to \mathbb{R}$ is convex on a convex set $D_0 \subseteq D$ if, for all $x, y \in D_0$ and $0 < \alpha < 1$

$$g(\alpha x + (1-\alpha)y) \leq \alpha g(x) + (1-\alpha)g(y) \quad (5.7)$$

The functional $g$ is strictly convex on $D_0$ if strict inequality holds in equation 5.7 whenever $x \neq y$. Also, it is clear that strict convexity implies convexity.

Theorem 5.2:
Let $r(x)$ be the distance from the origin to the point $x$, where $x \in \mathbb{R}^n$. Then $r(x)$ is a strict convex functional on $\mathbb{R}^n$.

Proof:
The proof is straightforward and comes directly from the triangle inequality.

Definition 5.3:
A functional $\phi$ in $\mathbb{R}^m$ is said to be nondecreasing if

$$\phi(r_1, r_2, \ldots, r_m) \leq \phi(r_1^i, r_2^i, \ldots, r_m^i)$$
whenever $r_1 \leq r_1^i \quad \forall i$.

Theorem 5.4:
If $r_1, r_2, \ldots, r_m$ are convex functionals on $D \subseteq \mathbb{R}^n$ and if $\phi$ is a nondecreasing convex functional on $\mathbb{R}^m$, then the functional $g$ defined by

$$g(x) = \phi[r_1(x), r_2(x), \ldots, r_m(x)]$$
is convex on $D$. 
Proof:
Let \( x \) and \( y \in D \). Then, for all \( i \),

\[
r_{\perp}(\alpha x + (1-\alpha)y) \leq \alpha r_{\perp}(x) + (1-\alpha)r_{\perp}(y)
\]

Therefore:

\[
g(\alpha x + (1-\alpha)y) = \phi\left[r_{\perp}(\alpha x + (1-\alpha)y), \ldots \right]
\]

\[
\leq \phi\left[\alpha r_{\perp}(x) + (1-\alpha)r_{\perp}(y), \ldots \right]
\]

\[
\leq \alpha \phi\left[r_{\perp}(x), \ldots \right] + (1-\alpha) \left[r_{\perp}(y), \ldots \right]
\]

\[
\leq \alpha g(x) + (1-\alpha)g(y)
\]

So \( g \) is convex on \( D \).

Theorem 5.5:

Let \( \phi: \mathbb{R}^1 \to \mathbb{R}^1 \) be the functional defined by:

\[
\phi(r(x)) = r^2(x)
\]

where \( r(x) \) is the distance from the origin to the point \( x \in \mathbb{R}^n \).

Then \( \phi \) is a nondecreasing convex functional of \( r \).

Proof:
Clearly, \( \phi \) is a nondecreasing functional of \( r \). To prove that \( \phi \) is convex, let

\[
x, y \in \mathbb{R}^n
\]

and \( 0 < \alpha < 1 \).

Then

\[
[\alpha r(x) + (1-\alpha)r(y)]^2 = [\alpha r(y) - \alpha t + (1-\alpha)r(y)]^2
\]

\[
= [r(y) - \alpha t]^2
\]

\[
= r^2(y) - 2\alpha t r(y) + \alpha^2 t^2
\]

\[
= r^2(y) - \alpha r^2(y) + \alpha r^2(y) - 2\alpha t r(y) + \alpha^2 t^2
\]
\[ \leq (1-\alpha)r^2(y) + \alpha r^2(y) - 2\alpha tr(y) + \alpha t^2 \]
\[ \leq (1-\alpha)r^2(y) + \alpha [r(y) - t]^2 \]
\[ \leq (1-\alpha)r^2(y) + \alpha r^2(x) \]

i.e. \[ \phi[ar(x) + (1-\alpha)r(y)] \leq \alpha \phi[r(x)] + (1-\alpha) \phi[r(y)] \]

Therefore \( \phi \) is convex.

Note the equality holds if \( t = 0 \), i.e. if \( r(x) = r(y) \).

As a consequence of theorems 5.2, 5.4 and 5.5, the functional

\( g(x) = \|F(x)\|^2 \)

is convex. And since \( r(x) \) in theorem 5.2 is strictly convex, \( g(x) \) also is strictly convex. This result is summarized in the following theorem.

**Theorem 5.6:**

Let the functional \( G: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^1 \) be defined on a convex set \( D_0 \subset D \) such that \( g(x) = \|F(x)\|^2 \). Then \( g \) is strictly convex on \( D_0 \).

The following corollaries will be used to prove the final theorem of convergence.

**Corollary 5.7:**

Suppose that \( g: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^1 \) has a Gateaux-derivative on a convex set \( D_0 \subset D \). If \( g \) is convex on \( D_0 \), then

\[ [g'(y) - g'(x)](y-x) \geq 0, \forall x, y \in D_0 \]  \hspace{1cm} (5.8)

Moreover, if \( g \) is strictly convex on \( D_0 \), strict inequality holds in relation 5.8 whenever \( x \neq y \) and there is at most one critical point in \( D_0 \), (21, p. 86).

**Corollary 5.8:**

Suppose that \( g: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^1 \) has a G-derivative on a convex set \( D_0 \subset D \).
Then \( g \) is convex on \( D_0 \) if and only if
\[
g'(x)(y-x) \leq g(y) - g(x), \forall x, y \in D_0
\]
(5.9)

Moreover, \( g \) is strictly convex on \( D_0 \) if and only if strict inequality holds in relation 5.9 whenever \( x \neq y \), (21, p. 84).

Theorems 5.9 through 5.11 below are used to prove the intermediate result 5.5 in theorem 5.12.

**Theorem 5.9:**

A real matrix \( M \) is positive definite if and only if there is a non-singular real matrix \( N \) such that \( M = N^T N \). Also, if \( M \) is positive definite, (22, p. 94).

**Theorem 5.10:**

If a real matrix \( M \) is not symmetric, then \( M \) is positive definite if and only if the symmetric matrix \( M + M^T \) is positive definite, (22, p. 35).

**Theorem 5.11:**

If the Jacobian matrix \( J^k \) evaluated at \( x^k \) is nonsingular, then the matrix \( A^k_1 \) as defined in equation 5.3 and its inverse \( (A^k_1)^{-1} \) are positive definite.

**Proof:**

Consider the \( n \times n \) matrix \( L^k_1 \) given by equation 4.11 and its transpose,
\[
L^k_1 + (L^k_1)^T = (J^k)^T J^k + D^k
\]
where the diagonal matrix \( D^k \) is:
Since \( J^k \) is nonsingular, \((J^k)^T J^k\) is positive definite by theorem 5.9. Clearly, \( D^k \) is positive definite. Therefore \( L_1^k + (L_1^k)^T \) is positive definite. By theorem 5.10, \( L_1^k \) is also positive definite. It follows immediately that \( L_1 \) (equation 4.15), \( A_1^k \) and \((A_1^k)^{-1}\) are positive definite.

**Theorem 5.12:**

Let \( g: D \subset \mathbb{R}^n \to \mathbb{R}^1 \) be \( G \)-differentiable on a compact set \( D_0 \subset D \) and assume that \( A_1^{-1} : D_0 \to \mathbb{L}(\mathbb{R}^n) \) is a continuous mapping such that \( A_1(x)^{-1} \) is positive definite for each \( x \in D_0 \). Then there is a constant \( c > 0 \) such that:

\[
g'(x)p(x) \geq c \|g'(x)\| \|p(x)\| \quad \forall x \in D_0
\]  

(5.10)

where \( p(x) = A_1(x)g'(x)^T \), (21, p. 501).

Relation 5.5 comes directly from 5.10.

The following definitions are used in theorems 5.14 and 5.15 to prove the intermediate result:

\[
\lim_{k \to \infty} \frac{g'(x)^k p}{\|p\|} = 0
\]

**Definition 5.13:**

A mapping \( \sigma: [0, \infty) \to [0, \infty) \) is a forcing function (\( F \)-function) if for any sequence \( \{t_k\} \subset [0, \infty) \)

\[
\lim_{k \to \infty} \sigma(t_k) = 0 \implies \lim_{k \to \infty} t_k = 0.
\]
Note that any nondecreasing function $\sigma: [0, \infty) \to [0, \infty)$ such that $\sigma(0) = 0$ and $\sigma(t) > 0$ for $t > 0$ is necessarily an $F$-function.

**Definition 5.14:**

Let $g: D \subset \mathbb{R}^n \to \mathbb{R}_+$ be continuously differentiable, and assume that on some $D_0 \subset D$

$$\alpha = \sup \{ \|g'(x) - g'(y)\| : x, y \in D_0 \} > 0.$$ 

Then the mapping $\delta: [0, \infty) \to [0, \infty)$ defined by

$$\delta(t) = \inf \{\|x - y\| : x, y \in D_0, \|g'(x) - g'(y)\| \geq t\}, \quad t \in [0, \alpha),$$

$$\lim_{s \to \alpha} \delta(s), \quad t \in [\alpha, +\infty),$$

is the reverse modulus of continuity of $g': D \subset \mathbb{R}^n \to L(\mathbb{R}^n, \mathbb{R}^l)$ on $D_0$.

Note that $\delta$ is always well defined and isotone, and that $\delta(0) = 0$.

Theorem 5.15 shows that for $g$ defined in equation 3.13, $\delta$ of $g'$ is an $F$-function.

**Theorem 5.15:**

Let $g: D \subset \mathbb{R}^n \to \mathbb{R}_+$ be a strictly convex differentiable functional on $D_0 \subset D$. Then the reserve modulus of continuity $\delta$ of $g'$ is an $F$-function.

**Proof:**

Since $g$ is strictly convex:

$$\langle g'(x) - g'(y), (x - y) \rangle > 0 \quad \forall x, y \in D_0, x \neq y$$

$$\|g'(x) - g'(y)\| \|x - y\| \geq \|g'(x) - g'(y)\| \|x - y\| > 0$$

Hence

$$\|g'(x) - g'(y)\| > 0 \quad \text{and}$$

$$\|x - y\| > 0 \quad \forall x, y \in D_0, x \neq y$$
Therefore the quantity $\alpha$ in definition 5.14 is positive, and $\delta(t) > 0$ for all $t > 0$. Thus $\delta$ is an F-function.

Theorem 5.16:

Consider the iteration 5.2 where $\{p^k\}$ is any sequence such that $g'(x)p^k \geq 0$, $p^k \neq 0$, and $g$ is defined in equation 3.13 such that $g(x^{k+1}) \leq g(x^k)$. Then

1. $\{x^k\} \subset L (g(x^0))$

where the level set $L (g(x^0)) = \{x \in D \mid g(x) \leq g(x^0)\}$ of $g$ is path connected;

2. $\lim_{k \to \infty} \frac{g'(x^k)p^k}{\|p^k\|} = 0$

Proof:

Since $g$ is convex, every level set of $g$ is path connected (see, for example, 21, p. 99). Also since $g(x^{k+1}) \leq g(x^k)$, $\{x^k\} \subset L (g(x^0))$.

To prove $\lim_{k \to \infty} \frac{g'(x^k)p^k}{\|p^k\|} = 0$, relation 5.9 ($g'(x)(y - x) \leq g(y) - g(x)$, $\forall x, y \in L (g(x^0))$) is used twice to obtain

$$g'(x^{k+1}) p^k \leq g(x^k) - g(x^{k+1}) \leq g'(x^k)p^k$$

Then two situations are considered:

a/ Assume $g'(x^{k+1}) p^k \leq 0$, then
 $$g'(x^k)p^k - g'(x^{k+1})p^k \geq g'(x^k)p^k$$

Cauchy-Schwarz inequality gives:

$$\|g'(x^k) - g'(x^{k+1})\| \|p^k\| \geq \|g'(x^k)p^k\|$$

$$\geq g'(x^k)p^k$$
\[ \|g'(x^k) - g'(x^{k+1})\| \geq \frac{g'(x^k)p^k}{\|p^k\|} \]

It then follows from the definition of the reverse modulus of continuity function \( \delta \) that

\[ \delta(g'(x^k)p^k) \leq \|x^k - x^{k+1}\| \]

\[ \leq \frac{1}{2} \|p^k\| \]

Since \( g'(x^k)p^k \geq g(x^k) - g(x^{k+1}) > 0 \), there exists a \( \mu_1 \in (0, 1] \) such that

\[ g(x^k) - g(x^{k+1}) \geq \mu_1 g'(x^k)p^k \]

\[ \geq \mu_1 g'(x^k)p^k \frac{1}{2} \|p^k\| \]

\[ \geq \mu_1 g'(x^k)p^k \delta(g'(x^k)p^k) \]

\[ \geq \mu_1 t \delta(t) \]

where \( t = \frac{g'(x^k)p^k}{\|p^k\|} \geq 0 \)

\[ g(x^k) - g(x^{k+1}) \geq \sigma_1(t) = \sigma_1(\frac{g'(x^k)p^k}{\|p^k\|}) \quad (5.11) \]

where \( \sigma_1(t) = \mu_1 t \delta(t) \)

From theorem 5.15, the reverse modulus of continuity function \( \delta \) of \( g' \) is an \( F \)-function. It is immediate that \( \sigma_1(t) \) in relation 5.11 is also an \( F \)-function.

Since \( g(x^k) \geq g(x^{k+1}) \), \( g \) is bounded below on \( L(g(x^0)) \) and the sequence \( \{x^k\} \) remains in \( L(g(x^0)) \), it follows that
\[
\lim_{k \to \infty} [g(x^k) - g(x^{k+1})] = 0
\]

Hence
\[
\lim_{k \to \infty} \sigma_1 \left( \frac{g'(x^k)}{||p_k||} \right)^k = 0
\]

and
\[
\lim_{k \to \infty} \frac{g'(x^k)p_k}{||p_k||} = 0
\]

b/ Assume \( g'(x^{k+1})p_k > 0 \).

Since \( g'(x^{k+1})p_k \geq g'(x^k)p_k > 0 \), there exists a \( \mu_2 \in (0, 1) \) such that \( g'(x^{k+1})p_k = \mu_2 g'(x^k)p_k \)

Then
\[
(1 - \mu_2)g'(x^k)p_k = [g'(x^k) - g'(x^{k+1})]p_k \\
\leq \|g'(x^k) - g'(x^{k+1})\|_p \|p_k\|
\]
or
\[
\|g'(x^k) - g'(x^{k+1})\|_p \geq (1 - \mu_2)\frac{g'(x^k)p_k}{\|p_k\|}
\]

Then the definition of the reverse modulus of continuity function is used to obtain:
\[
\frac{1}{\|p_k\|} \geq \delta[(1 - \mu_2)\frac{g'(x^k)p_k}{\|p_k\|}] = \delta[(1 - \mu_2)t]
\]

Repeating the steps in part a/ above:
\[
g(x^k) - g(x^{k+1}) \geq \mu_1 \frac{g'(x^k)p_k}{\|p_k\|} - \delta[(1 - \mu_2)t] \\
\geq \sigma_2(t) = \mu_1 + \delta[(1 - \mu_2)t]
\]

where \( \sigma_2(t) \) is an \( F \)-function. Therefore
Theorem 5.17 given below is used with corollary 5.7 to show that \( g \) has only one critical point in \( D_0 \).

**Theorem 5.17:**

Let \( g: D \subset \mathbb{R}^n \rightarrow \mathbb{R} \) be continuously differentiable on a compact set \( D_0 \subset D \) and suppose that \( \{x^k\} \subset D_0 \) is any sequence which satisfies

\[
\lim_{k \to \infty} g'(x^k)^T = 0
\]

Then the set \( \Omega = \{x \in D_0 \mid g'(x)^T = 0\} \) of critical points of \( g \) in \( D_0 \) is not empty and

\[
\lim_{k \to \infty} \left[ \inf_{x \in \Omega} ||x^k - x|| \right] = 0
\]

In particular, if \( \Omega \) consists of a single point \( x^* \), then

\[
\lim_{k \to \infty} x^k = x^* \quad \text{and} \quad g'(x^*)^T = 0 \quad (21, \text{p. 475}).
\]

Finally the results of corollary 5.7, theorems 5.12, 5.16 and 5.17 are combined to prove the following theorem of convergence.

**Theorem 5.18:**

Given \( F: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n \) such that each component function \( f_i, i = 1, 2, \ldots, n \) is continuously differentiable on a closed and bounded convex subset \( D_0 \subset D \) and \( g \) is defined in equation 3.13 such that \( g(x^{k+1}) \leq g(x^k) \), then if the Jacobian matrix \( J(x) \) is nonsingular on \( D_0 \), the sequence \( \{x^k\}, k = 0, 1, \ldots, \infty \) generated by equation 5.2 converges to an \( x^* \) such that the residue \( F(x^*) = 0 \).
Proof:

Theorems 5.12 and 5.16 give:

\[
\lim_{k \to \infty} g'(x^k) p^k = 0, \quad k = 0, 1, \ldots
\]

\[
g'(x^k) p^k \geq c \|g'(x^k)\|^T \|p\|, \quad c > 0
\]

which imply

\[
\lim_{k \to \infty} g'(x^k) = 0
\]

From corollary 5.7, there is at most one critical point of \(g\) in \(D_0\). This together with theorem 5.17 proves that \(g\) has only one critical point in \(D_0\) and

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

Therefore:

\[
g'(x^\ast) = 0 = 2F(x^\ast)^T J(x^\ast)
\]

Assume that \(F(x^\ast) \neq 0\). Then the column vectors of the Jacobian evaluated at \(x^\ast\) are orthogonal to the nonzero vector \(F(x^\ast)^T\). This implies these columns are linearly dependent and therefore, that \(J(x^\ast)\) is singular, which is a contradiction.

So \(F(x^\ast) = 0\) and \(x^\ast\) is the solution of the system of equations.

B. Extension to Higher Dimensions

The above proof of convergence could be readily carried over to the case of higher dimensional projection-based methods. The iteration formula for all the projection-based methods can be placed in the form of the general expression 5.1. The only difference lies in the matrices \(A^k_i\) (equation 5.3). For the 2-dimensional projection-based methods:
\[ A_2^k = (P_2^k)^T (L_2^{k_1})^{-1} M_2^{k_1} P_2^k \]  \hspace{1cm} (5.12)

For the 3-dimensional projection-based methods:
\[ A_3^k = (P_3^k)^T (L_3^{k_1})^{-1} M_3^{k_1} P_3^k \]  \hspace{1cm} (5.13)

where \( P_2^k, L_2^{k_1}, M_2^{k_1}, P_3^k, L_3^{k_1}, M_3^{k_1} \) are defined in sections IV B and IV C.

From the development of the methods in Chapter IV, the matrices \( (L_2^{k_1})^{-1} M_2^{k_1} \) and \( (L_2^{k_1})^{-1} M_3^{k_1} \) are positive definite. Then theorem 5.11 shows that \( A_2^k, (A_2^k)^{-1}, A_3^k \) and \( (A_3^k)^{-1} \) are also positive definite when the Jacobian matrix \( J^k \) evaluated at \( x^k \) is nonsingular. Theorem 5.12 uses this property of positive definiteness to obtain the intermediate result
\[ g'(x^k) P^k \geq 0 \quad \forall k \geq k_o \]  \hspace{1cm} (5.5)

The rest of the proof of convergence follows as in section V A.

For the \( m \)-dimensional \( (1 \leq m \leq n) \) projection-based methods:
\[ A_m^k = (P_m^k)^T (L_m^{k_1})^{-1} M_m^{k_1} P_m^k \]  \hspace{1cm} (5.14)

as shown in equation 4.57.

The proof of convergence for this general case then consists in showing that \( (L_m^{k_1})^{-1} M_m^{k_1} \) is positive definite for each value of \( m \). The remaining results can be obtained from section V A.

In particular, the iteration expression for the \( n \)-dimensional projection-based method is the same as that for the \( n \)-dimensional basic nonlinear projection method. Convergence of the \( m \)-dimensional \( (1 \leq m \leq n) \) basic nonlinear projection methods has been proven in Nguyen, Georg and Keller (20).
VI. COMPARISONS AND TEST PROBLEMS

A. Comparison with Basic Projection Methods

From the development of the projection-based methods in chapter IV, it can be seen that both the Jacobian matrix and the residue vector are not reevaluated until all \( n \) components of the approximate solution vector have been modified at least once. That is, the Jacobian and the residue need be evaluated only once per cycle (in iterative methods, a cycle is defined to be composed of one or more iteration steps which change each component of the approximate solution vector at least once). This advantage of the projection-based methods over the basic nonlinear projection methods is very significant and sometimes leads to considerably improved rates of convergence, specially when the dimension \( n \) of the nonlinear system is very large or when the evaluation of the components \( f_i^k, i = 1, 2, \ldots, n \) of \( F(x) \) requires much computational time.

Tables 6.1 to 6.3 below are used to compare the 1-dimensional basic projection methods with the 1-dimensional projection-based methods.
Table 6.1. Number of operations needed for each iteration step of 1-dimensional basic projection methods.

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Number of Multiplications</th>
<th>Number of Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Evaluate $F(x^k)$ once</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) Evaluate one column $J^k_i$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) $(J^k_i, J^k_{ii}) = J^k_{ii}$</td>
<td>n</td>
<td>n-1</td>
</tr>
<tr>
<td>(d) $(F(x^k), J^k_i)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(e) $x^{k+1} = x^k - \frac{(F(x^k), J^k_i)}{J^k_{ii}}$</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6.2 corresponds to 1 cycle which has n iteration steps.

Table 6.2. Number of operations needed for each cycle of 1-dimensional basic projection methods.

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Number of Multiplications</th>
<th>Number of Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Evaluate $F(x^k)$ n times</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) Evaluate matrix $J^k$ once</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) Iteration expressions</td>
<td>$n(2n + 1)$</td>
<td>$n(2n - 1)$</td>
</tr>
</tbody>
</table>
Table 6.3. Number of operations needed for each cycle of 1-dimensional projection-based methods.

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Number of Multiplications</th>
<th>Number of Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Evaluate (F(x^k)) once</td>
<td>(n^2 (n + 2))</td>
<td>(n(n^2 + \frac{n}{2} - \frac{1}{2}))</td>
</tr>
<tr>
<td>(b) Evaluate matrix (J^k) once</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) Iteration expression</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tables 6.2 and 6.3 show that 1-dimensional projection-based methods evaluate the residue vector \(F(x^k)\) only once per cycle, thus need less computational time for part (a) of the evaluation and more computational time for part (c) than 1-dimensional basic projection methods.

In order to carry the comparisons further to specific test problems, the following timing estimates are obtained from the reference manual (15).

Table 6.4. Timing estimates for IBM S/360 Model 65 (Fortran IV.)

<table>
<thead>
<tr>
<th>Operation/Subprogram</th>
<th>Time (μsec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>3.5</td>
</tr>
<tr>
<td>Multiplication</td>
<td>6.5</td>
</tr>
<tr>
<td>DCOTAN</td>
<td>153</td>
</tr>
<tr>
<td>DEXP</td>
<td>138</td>
</tr>
<tr>
<td>DSIN</td>
<td>133</td>
</tr>
<tr>
<td>DSQRT</td>
<td>85.3</td>
</tr>
<tr>
<td>DLOG</td>
<td>161</td>
</tr>
</tbody>
</table>

To compare 1-dimensional projection-based methods with 1-dimensional basic projection methods, test problems are obtained from White's dissertation (25) on basic projection methods.
Problem 6.1

This problem (25, p. 67) concerns the optimum design for a VHF aerial feeder system. With the dimension \( n = 6 \), the nonlinear equations are:

\[
\begin{align*}
\sum_{j=2}^{6} \cot B_1 x_j &= 0 \quad \text{where } B_1 = 0.02249 \\
\sum_{j=1}^{6} \cot B_2 x_j &= 0 \quad \text{where } B_2 = 0.02166 \\
\sum_{j=2}^{6} \cot B_3 x_j &= 0 \quad \text{where } B_3 = 0.02083 \\
\sum_{j=3}^{6} \cot B_4 x_j &= 0 \quad \text{where } B_4 = 0.02000 \\
\sum_{j=4}^{6} \cot B_5 x_j &= 0 \quad \text{where } B_5 = 0.01918 \\
\sum_{j=1}^{6} \cot B_6 x_j &= 0 \quad \text{where } B_6 = 0.01835
\end{align*}
\]

The initial approximate solution vector \( x^0 \) is chosen as:

\[
x^0 = (126.245, 120.262, 96.292, 57.864, 36.170, 26.443)
\]

Problem 6.2

The following problem illustrates the case where the expressions for the components of the residue \( F(x) \) look simple; but the amount of computational time needed to evaluate these expressions is large enough to make the projection-based methods more efficient than the basic projection methods.

\[
\begin{align*}
n &= 6 \\
\sum_{j=1}^{i} \left( \frac{x_j^2}{2} - \frac{x_j}{2} \right) &= 0, \quad i = 1, 2, ..., 6
\end{align*}
\]

The solution vector for this problem is obviously \( x = (1, 1, 1, 1, 1, 1) \).
The chosen initial approximate solution vector is
\[ x^0 = (0.9, 0.9, 0.9, 0.9, 0.9, 0.9). \]

**Problem 6.3**

This problem (25, p. 63) is used to determine the length and the horizontal tension of an elastic steel wire which is stretched between two rigid supports. The final nonlinear system with two equations (\( n = 2 \)) is:

\[
\begin{align*}
&x_1 \ln\left(x_2/x_1 + \sqrt{1 + (x_2/x_1)^2}\right) + 0.0072x_1x_2/19,000 - 50.0 = 0 \\
&x_1 \sqrt{1 + (x_2/x_1)^2} - x_1 + 0.0078x_2^2/38,000 - 2.5 = 0
\end{align*}
\]

Here, the initial approximate solution vector is \( x^0 = (500.40, 50.085). \)

**Problem 6.4**

In this problem, the evaluation of the components of the residue \( F(x) \) requires relatively little computational time.

\[
\begin{align*}
n &= 4 \\
x_1x_2 - 1 &= 0 \\
x_1x_3 - 1 &= 0 \\
x_1x_4 - 1 &= 0 \\
x_2x_3 - 1 &= 0
\end{align*}
\]

The initial approximate solution vector is \( x = (0.5, 0.5, 0.5, 0.5). \)

For each of the test problems shown above, the computational time needed for each iteration cycle could be determined from the timing estimates in table 6.4 and from tables 6.2, 6.3. This is summarized in table 6.5.
Table 6.5. Estimated computational time needed for each iteration cycle.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Dimension of Nonlinear System</th>
<th>Time (μsec) 1-Dim Basic Projection Methods</th>
<th>Time (μsec) 1-Dim Projection-Based Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>6</td>
<td>34,956</td>
<td>12,974</td>
</tr>
<tr>
<td>6.2</td>
<td>6</td>
<td>5,778</td>
<td>3,521</td>
</tr>
<tr>
<td>6.3</td>
<td>2</td>
<td>1,112</td>
<td>818</td>
</tr>
<tr>
<td>6.4</td>
<td>4</td>
<td>492</td>
<td>909</td>
</tr>
</tbody>
</table>

The timing estimates in table 6.5 do not include the overhead costs such as the initialization of various identifiers in the programs, the use of a vector in the programs to handle the different orders of projection, the time to print out intermediate results, etc. These overhead costs are higher for 1-dimensional projection-based methods than for 1-dimensional basic projection methods due to the use of more arrays in the programs. Therefore, table 6.5 actually shows a rough comparison between the two indicated classes of methods.

Table 6.6 lists the results of the runs used to test the comparison in table 6.5. The Fortran programs used were executed on an IBM S360/65 under HASP. The same programs were also run on an IBM S360/65 user ASP with no significant difference in the results. Double precision was used.
Table 6.6. Test results for 1-dimensional basic projection methods (1DBPM) and for 1-dimensional projection-based methods (1DPBM).

<table>
<thead>
<tr>
<th>Problem</th>
<th>Norm of Residue</th>
<th>Order of Projection</th>
<th>1DBPM Cycles</th>
<th>1DBPM Time</th>
<th>1DPBM Cycles</th>
<th>1DPBM Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>0.006050</td>
<td>6, 1, 2, 3, 4, 5</td>
<td>19</td>
<td>3.69</td>
<td>16</td>
<td>2.63</td>
</tr>
<tr>
<td>6.2</td>
<td>0.002610</td>
<td>1, 2, 3, 4, 5, 6</td>
<td>20</td>
<td>1.83</td>
<td>8</td>
<td>1.10</td>
</tr>
<tr>
<td>6.3</td>
<td>&lt; 10^{-13}</td>
<td>1, 2</td>
<td>5</td>
<td>0.08</td>
<td>4</td>
<td>0.07</td>
</tr>
<tr>
<td>6.4</td>
<td>&lt; 10^{-7}</td>
<td>1, 2, 4, 3</td>
<td>27</td>
<td>0.99</td>
<td>27</td>
<td>1.73</td>
</tr>
</tbody>
</table>

For each class of methods, the number of cycles and the CPU time (in seconds) are given. These denote the number of cycles necessary to reduce the norm of the residue to the values shown in the second column of table 6.6, and the corresponding CPU time.

There is an obvious correlation between table 6.5 and table 6.6. Table 6.5 shows theoretical estimates, while table 6.6 indicates the experimental results. As expected, the 1-dimensional projection based-methods (1DPBM) converge faster, i.e. in less time and less number of cycles, than the 1-dimensional basic projection methods (1DBPM) for problems 6.1 through 6.3. For problem 6.4, 1DBPM require less time than 1DPBM, again as expected from table 6.5. In problem 6.4, the expression for each component of F(x) involves only one multiplication and one addition. Therefore, table 6.5 and the result in table 6.6 show that the use of 1DPBM to solve such a system is not advantageous. Table 6.6 indicates that the rate of convergence of 1DPBM is faster than that of 1DBPM, for problems 6.1 through 6.3. The equations in these problems
are slightly more complex than those in problem 6.4. In actual practice, where a nonlinear system may have several hundred of computationally complex equations, the advantages of IDPBM over IDBPM would become much more significant.

Fortran programs also were written to test 2-dimensional basic projection methods(2DBPM) and 2-dimensional projection-based methods(2DPBM). These programs were also executed on an IBM 330/65 under HASP and used double precision. Results are shown in table 6.7.

Table 6.7. Test results for 2DBPM and 2DPBM.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Norm of Residue</th>
<th>Order of Projection</th>
<th>2DBPM Cycles</th>
<th>2DBPM Time</th>
<th>2DPBM Cycles</th>
<th>2DPBM Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>0.003596</td>
<td>1 2, 3 4, 5 6</td>
<td>35</td>
<td>5.32</td>
<td>18</td>
<td>3.49</td>
</tr>
<tr>
<td>6.2</td>
<td>0.002610</td>
<td>1 2, 3 4, 5 6</td>
<td>15</td>
<td>2.15</td>
<td>13</td>
<td>1.43</td>
</tr>
<tr>
<td>6.3</td>
<td>&lt; 10^{-13}</td>
<td>1 2</td>
<td>3</td>
<td>0.06</td>
<td>3</td>
<td>0.06</td>
</tr>
<tr>
<td>6.4</td>
<td>&lt; 10^{-7}</td>
<td>1 2, 3 4</td>
<td>24</td>
<td>1.13</td>
<td>24</td>
<td>1.60</td>
</tr>
</tbody>
</table>

Again, 2DPBM are superior to 2DBPM for nonlinear systems which do not have very simple equations, as is the case of problems 6.1 and 6.2. For problem 6.3, the speed of convergence of 2DPBM is about the same as that of 2DBPM. The equations of the nonlinear system in problem 6.4 are so simple that 2DEPM converge faster than 2DPBM in this case. In practical nonlinear systems with complex equations, 2DPBM would have a higher speed of convergence than 2DEPM and could be used as accelerations to 2DBPM.

Finally, test results have also shown that, like the basic projec-
tion methods, the rate of convergence of projection-based methods depends on the order of the projection.

B. Comparison with Newton's Method

Table 6.8 below is used to compare Newton's method with projection-based methods.

Table 6.8. Number of operations needed for each cycle of Newton's method.

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Number of Multiplications</th>
<th>Number of Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Evaluate $F(x^k)$ once</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) Evaluate $J^k$ once</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) Evaluate $(J^k)^{-1}$</td>
<td>$\frac{n^4}{4} + n^3 - \frac{n^2}{3}$</td>
<td>$\frac{n^4}{3} + \frac{n^3}{2} - \frac{5n^2}{6}$</td>
</tr>
<tr>
<td>(d) Iterative expression</td>
<td>$n^2$</td>
<td>$n^2$</td>
</tr>
</tbody>
</table>

Tables 6.3 and 6.8 show that during each iteration cycle, both LDPEM and Newton's method evaluate the residue vector once and the Jacobian matrix once. However, the number of the remaining operations required during each cycle of Newton's method is of the order $n^4$ (n being the dimension of the nonlinear system of equations) while that of LDPEM is of the order $n^3$. Therefore, when $n$ is large, one iteration cycle of LDPEM requires less computation than one iteration cycle of Newton's method. Low-dimensional projection-based methods also hold this computational advantage over Newton's method. For high-dimensional pro-
jection-based methods, the iteration expressions become more complex and the computational time may exceed that of Newton's method.

In many problems, Newton's method does not converge to a solution if the choice of the initial approximate solution vector is not 'right'. For instance, for problem 6.4 above, White (25, p. 67) stated that 'obviously, Newton's method does not converge, even for the initial values given which are quite close to the solution'. Other examples in White (25) show that Newton's diverges under similar conditions. Therefore, these examples indicate that projection-based methods, which could be considered as accelerations to basic projection methods, have a wider range of applications than Newton's method.
A new class of algorithms to solve nonlinear systems of equations has been presented. This class is based on the nonlinear projection methods. The main advantage of the new methods is that the residue vector and the Jacobian need be calculated only once per cycle. As a consequence, the number of arithmetic operations for the projection-based methods is less than that for Newton's method and less than that for basic projection methods, when low-dimensional projections are used. Also converging conditions have been established and are less stringent than either those of Newton's method or those of the basic projection methods. Computational results from test problems are given to illustrate the above observations.

As the dimension of the projection subspace increases, the computational overhead incurred in the projection-based methods could become significant. In fact, both the n-dimensional projection-based method and the n-dimensional basic projection method are computationally more complex than Newton's method. Georg in Nguyen, Georg and Keller (20) has proved that Newton's method is mathematically equivalent to the above n-dimensional methods.

For best results it is suggested that, given a nonlinear system of equations, some criterion be used to select an optimal projection subspace at each iteration step. If the dimension of this optimal projection subspace is n, Newton's method should be used, otherwise
the projection-based method with the proper dimension is used.

B. Future Research

Some of the possible topics for future research are listed below.

1. In the author's opinion, the condition \( g(x^{k+1}) \leq g(x^k) \) in theorem 5.16 could be considerably weakened, perhaps by using the convexity property of \( g(x) \). The condition \( g(x^{k+1}) \leq g(x^k) \) is satisfied in all attempted test problems and initial conditions, including many of those presented in White (25). Further study is necessary to show that the assertion \( g(x^{k+1}) \leq g(x^k) \) is true under weak conditions.

2. The iteration expressions for projection-based methods include sparse matrices such as \( L^k_1 \) in equation 4.16, \( L^k_2 \) and \( H^k_2 \) in equation 4.43, \( L^k_3 \) and \( M^k_3 \) in equation 4.56. For sparse matrices, the number of nonzero elements is small, and hence the number of arithmetic operations to be performed per iteration is small. This advantage could be used to further reduce the computational time needed when using projection-based methods to solve nonlinear systems.

3. The choice of the initial approximate solution vector \( x^0 \) affects the rate of convergence to the desired solution. Further work is necessary to develop initial vector algorithms which significantly reduce the number of cycles normally required. These results may also be used to obtain a good initial approximate vector \( x^0 \) for Newton's method, since Newton's method is a member of the class of projection methods.

4. Georg and Keller (13) have developed criteria to select quasi-optimal projection subspaces. Further work could be done to improve
existing selection criteria and to develop new optimal or quasi-optimal criteria. These new criteria should be analyzed to determine their characteristic as well as their effects on the convergence of the projection-based methods.

5. Also further acceleration schemes should be considered to increase the speed of convergence. It is recommended that mathematical results on the rate of convergence of projection-based methods be obtained. These mathematical results could be used to compare the speed of convergence of projection-based methods with that of other iterative schemes.

As stated in the introduction (Chapter II), the objective of this research work is to develop a class of methods to solve nonlinear systems, which require fewer conditions for guaranteed convergence, are fast to converge and yet general enough to cover a large number of nonlinear problems.
VIII. BIBLIOGRAPHY


15. IBM S/360 Fortran IV Library Subprograms, Form No. GN 28-0588.


IX. ACKNOWLEDGEMENTS

The author wishes to express his deepest appreciation to Dr. Roy Keller for his constant guidance and encouragement during the preparation of this dissertation and throughout the author's graduate program at Iowa State University.

I would also like to express my sincere appreciation to Drs. Roger Homer, Robert Lambert, Arthur Pohm and Charles Wright who have accepted to serve on my advisory committee and to Iowa State University for the financial support during my graduate program.

I am specially grateful to my family. To my parents for instilling in me the desire and initiative to pursue this academic course and to my wife Lien for her unrelenting support and patience throughout my graduate study years.