27
6-23-75
-cpt to NTLS
IS-3546

PROJECTION-BASED METHODS
FOR SOLVING SYSTEMS OF
N NONLINEAR EQUATIONS
IN N UNKNOWN

Long Vo Nguyen and Roy F. Keller

AMES LABORATORY, USERDA
IOWA STATE UNIVERSITY
AMES, IOWA

MASTER

Date Transmitted: May 1975

PREPARED FOR THE U. S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
UNDER CONTRACT W-7405-eng-82

DISTRIBUTION OF THIS DOCUMENT UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

PROJECTION-BASED METHODS FOR SOLVING
SYSTEMS OF N NONLINEAR EQUATIONS IN N UNKNOWN

Long Vo Nguyen and Roy F. Keller

Based on Ph.D. Thesis Submitted to Iowa State University

Ames Laboratory, ERDA
Iowa State University
Ames, Iowa 50010

Date Transmitted: May 1975

PREPARED FOR THE U. S. ENERGY RESEARCH AND
DEVELOPMENT ADMINISTRATION UNDER CONTRACT W-7405-eng-82

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

DISTRIBUTION OF THIS DOCUMENT UNLIMITED

fy

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Available from: National Technical Information Service
U. S. Department of Commerce
P.O. Box 1553
Springfield, VA 22161

Price: Microfiche	\$2.25
Paper Copy	\$4.25

TABLE OF CONTENTS

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

ABSTRACT

Projection methods were first introduced to solve systems of linear equations and have recently been extended to nonlinear problems. The basic idea behind the nonlinear projection methods is to minimize the square of the norm of the residue vector at each iteration step. These belong to the descent class of minimization methods. They are not strict total step methods and are used to change from 1 to n (n being the dimension of the nonlinear system of equations) components of the approximate solution vector at any given iteration step.

The class of projection-based methods presented in this dissertation are total step methods. They change all n components of the approximate solution vector at every iteration. The main advantage of these methods over the basic projection methods is that both the residue vector and the Jacobian matrix of the nonlinear system need be evaluated only once per iteration cycle. In general, the rate of convergence of projection-based methods is faster than that of basic projection methods. Projection-based methods use less computational time and fewer number of cycles to reduce the norm of the residue to a given tolerance. Therefore, they could be considered as accelerations to basic projection methods.

A proof of convergence follows the development of the projection-based methods. The conditions for convergence are more relaxed than

the original convergence conditions of the basic projection methods. A comparison is made between 1-dimensional projection-based methods and Newton's method. It shows that low-dimensional projection-based methods are computationally more efficient than Newton's method.

Also computational results from test problems are given to illustrate the above observations. In particular, examples show that projection-based methods have a wider range of applications than Newton's methods.

Distribution List

Chicago Patent Office	1
OR	27
Ames Laboratory Library	5
R. F. Keller	27
	—
Total	60

I. AN INDEX TO SYMBOLS AND NOTATIONAL DEFINITIONS

a_{ij}	Equation 4.29
A^k	Equation 3.12
A_i^k	Equation 5.3
α_k	represents the basic steplength in iteration formulas
α_i^k	Equation 4.9
$\alpha^{k'}$	Equation 4.12
β_i^k	Equation 4.33
$\beta^{k'}$	Equation 4.36
d_{ijl}	Equation 4.49
dx^k	represents the change vector $x^{k+1} - x^k$
dx_i^k	represents the i th component of dx^k
f_i	represents the i th component of the nonlinear system $F(x)$
$F(x^k)$	represents the residue vector at the k th iteration
$g(x)$	represents the square of the norm of $F(x)$ - Equation 3.13
$g'(x)$	represents the derivative of $g(x)$
γ_i^k	Equation 4.46
$\gamma^{k'}$	Equation 4.51
J^k	represents the Jacobian matrix evaluated at x^k
J_i^k	represents the i th column of J^k
J_{ij}^k	represents the inner product of J_i^k and J_j^k
L_i^k	represents a lower triangular matrix - Equations 4.11, 4.35
$L_i^{k'}$	represents a lower triangular matrix - Equations 4.15, 4.41
m	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
M_{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
ρ_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 $g: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^1$, 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:

$$x^{k+1} = x^k + dx^k, \quad k = 0, 1, \dots \quad (2.1)$$

$$\text{where } dx^k = -w_k \alpha_k p^k \quad (2.2)$$

w_k is a relaxation parameter

α_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.

A. Paraboloid Methods

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:

$$g_k: \mathbb{R}^n \rightarrow \mathbb{R}^1, g_k(x) = c - b^T x + \frac{1}{2} x^T A x \quad (2.3)$$

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

$$g'_k(x)^T = Ax - b = 0 \quad (2.4)$$

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:

$$g_k(x) \approx g(x^k) + g'(x^k)(x - x^k) + \frac{1}{2}(x - x^k)^T H_g(x^k)(x - x^k) \quad (2.5)$$

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

$$x^{k+1} = x^k - H_g(x^k)^{-1} g'(x^k)^T \quad (2.6)$$

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

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

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

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

$$\Delta_j g(x^k) = g(x^k + h_j e^j) - g(x^k)$$

$$\begin{aligned} \Delta_i \Delta_j g(x^k) &= g(x^k + h_i e^i + h_j e^j) - g(x^k + h_i e^i) \\ &\quad - g(x^k + h_j e^j) + g(x^k) \end{aligned}$$

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

$$x^k, x^k + h_i e^i, i=1, \dots, n, x^k + h_i e^i + h_j e^j, i=1, \dots, n, j=1, \dots, n$$

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.

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

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

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

where $F(x^k) \neq 0$ and the parameter α_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 $e^1, \dots, e^n$ 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 $q^1, \dots, q^m$ of nonzero vectors which span 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 $\|x\| = (x^T C x)^{\frac{1}{2}}$ provides the iterative expression

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

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

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

$$k = 0, 1, \dots$$

with B_0 : an arbitrary, symmetric, positive definite matrix

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

$$q^k = g'(x^{k+1})^T - g'(x^k)^T, \quad k = 0, 1, \dots$$

C. Steplength Algorithms

The steplength algorithms consider various ways of choosing the steplength α_k which is used in the general iteration

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

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, α_k is chosen so as to minimize g according to one of the following three possibilities:

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

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

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

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 α_k is a solution of the one-dimensional equation

$$g'(x^k - \alpha_k p^k) p^k = 0 \quad (2.17)$$

If the Curry principle is applied, α_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 α_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 $\psi: [0, \hat{\alpha}] \rightarrow \mathbb{R}^1$ such that

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

then any steplength $\alpha_k \in (0, \hat{\alpha})$ decreases g . Depending on the nonlinear system at hand, there are several ways of obtaining ψ . 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, α_k may be chosen as that value of $\alpha \in (0, \hat{\alpha})$ which minimizes ψ , that is

$$\psi(\alpha_k) = \min \{\psi(\alpha) | \alpha \in (0, \hat{\alpha})\} \quad (2.21)$$

The Goldstein principle determines the admissible steplength set J_k of α , in which g decreases:

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

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

$$\sigma_i(\alpha) = g(x^k) - \mu_i \alpha g'(x^k) p^k, \quad i = 1, 2 \quad (2.23)$$

and μ_1, μ_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

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

$$p^0 = Ax^0 - b, \quad p^{k+1} = Ax^{k+1} - b - \beta_k p^k \quad (2.25)$$

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

where the nonzero vectors p^j , $j = 0, 1, \dots$ form a conjugate basis, and α_k , the steplength, is obtained by choosing

$$g(x^k - \alpha_k p^k) = \min \{g(x^k - \beta p^k) \mid \beta \in (-\infty, \infty)\} \quad (2.26)$$

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 $F: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ 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 $D_0 \subset D$, 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

$$f_i(x^{k+1}) = f_i(x^k) + (dx^k, J_{f_i}), \quad i = 1, 2, \dots, n \quad (3.1)$$

$$\text{where } dx^k = x^{k+1} - x^k \quad (3.2)$$

J_{f_i} in the inner product (dx^k, J_{f_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 $T = \{i_1, i_2, \dots, i_m\}$. Then by noting that $dx^k_u = 0$ for $u \notin T$, equation 3.1 could be simplified further to give

$$f_i(x^{k+1}) = f_i(x^k) + \sum_{v \in T} dx_v^k f_{ix_v}(x^k), \quad i = 1, 2, \dots, n \quad (3.3)$$

where dx_v^k is the v th component of dx^k and

$$f_{ix_v} = \frac{\delta f_i}{\delta x_v}$$

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

$$\begin{aligned} \|F(x^k)\|^2 - \|F(x^{k+1})\|^2 &= \sum_{i=1}^n (f_i(x^k))^2 - \sum_{i=1}^n (f_i(x^{k+1}))^2 \\ &\approx \sum_{i=1}^n (f_i(x^k))^2 - \sum_{i=1}^n (f_i(x^k) + \sum_{v \in T} dx_v^k f_{ix_v}(x^k))^2 \end{aligned} \quad (3.4)$$

Equation 3.4 is differentiated with respect to dx_v^k . After setting the resulting expressions to zero, dx_v^k , $v \in T$, can be obtained directly by solving the following set of linear equations

$$\sum_{v \in T} (J_v^k, J_s^k) dx_v^k = -(F(x^k), J_s^k), \quad v \in T \quad (3.5)$$

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 $i \notin T$, $dx_i^k = 0$.

The equation 3.5 is expanded and put into matrix notation to include the unchanged components of dx^k . This results in:

$$dx^k = x^{k+1} - x^k = -(S^k)^T (Q^k)^{-1} V(x^k) \quad \text{or} \quad (3.6)$$

$$x^{k+1} = x^k - (S^k)^T (Q^k)^{-1} V(x^k) \quad (3.7)$$

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

$$Q^k = \begin{bmatrix} (J_{i_1}^k, J_{i_1}^k) & (J_{i_1}^k, J_{i_2}^k) & \dots & (J_{i_1}^k, J_{i_m}^k) \\ (J_{i_2}^k, J_{i_1}^k) & & & \\ \vdots & & & \\ (J_{i_m}^k, J_{i_1}^k) & \dots & \dots & (J_{i_m}^k, J_{i_m}^k) \end{bmatrix} \quad (3.8)$$

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:

$$V(x^k) = \begin{bmatrix} (J_{i_1}^k, F(x^k)) \\ (J_{i_2}^k, F(x^k)) \\ \vdots \\ (J_{i_m}^k, F(x^k)) \end{bmatrix} \quad (3.9)$$

Equation 3.9 could be rewritten in another form:

$$V(x^k) = S^k (J^k)^T F(x^k) \quad (3.10)$$

where J^k is the Jacobian matrix evaluated at x^k .

Using 3.10 in equation 3.7:

$$x^{k+1} = x^k - (S^k)^T (Q^k)^{-1} S^k (J^k)^T F(x^k)$$

$$x^{k+1} = x^k - A^k (J^k)^T F(x^k) \quad (3.11)$$

$$\text{where } A^k = (S^k)^T (Q^k)^{-1} S^k \quad (3.12)$$

Let

$$g(x) = \|F(x)\|^2 \quad (3.13)$$

Then

$$g'(x^k) = 2F(x^k)^T J^k \quad (3.14)$$

Equations 3.11 and 3.14 provide the iterative expression:

$$x^{k+1} = x^k - \frac{1}{2} A^k g'(x^k)^T \quad \text{or} \quad (3.15)$$

$$x^{k+1} = x^k - \frac{1}{2p} p^k \quad (3.16)$$

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

$$p^k = A^k g'(x^k)^T \quad (3.17)$$

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 $i_1, i_2, \dots, i_m$ 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_1^k in the component x_1^k of the approximate solution vector x^k is given by:

$$-dx_1^k = \frac{(F(x^k), J_1^k)}{J_{11}^k} \quad (4.1)$$

where J_1^k is the first column of the Jacobian matrix J^k evaluated at x^k . J_{11}^k is the inner product (J_1^k, J_1^k) .

Equation 4.1 could be rewritten as:

$$-dx_1^k = \frac{(F(x^k), \alpha_1^k)}{J_{11}^k} = \frac{(\alpha_1^k)^T F(x^k)}{J_{11}^k} \quad (4.2)$$

$$\text{where the column vector } \alpha_1^k = J_1^k \quad (4.3)$$

Without reevaluating $F(x^k)$ and J^k and still using the 1-dimensional nonlinear projection method, the next component change dx_2^k is:

$$\begin{aligned} -dx_2^k &= \frac{(F(x^k) - dx_1^k J_1^k, J_2^k)}{J_{22}^k} \\ &= \frac{(F(x^k), J_2^k) - (J_1^k, J_2^k) dx_1^k}{J_{22}^k} \end{aligned} \quad (4.4)$$

$$\begin{aligned}
&= [(F(x^k), \alpha_2^k) - \frac{J_{12}^k (F(x^k), \alpha_1^k)}{J_{11}^k}] / J_{22}^k \\
&= \frac{(F(x^k), \alpha_2^k)}{J_{22}^k} = \frac{(\alpha_2^k)^T F(x^k)}{J_{22}^k} \quad (4.5)
\end{aligned}$$

$$\text{where } \alpha_2^k = J_2^k - \frac{J_{12}^k}{J_{11}^k} \alpha_1^k \quad (4.6)$$

J_{ij}^k is the inner product of the i th column with the j th column of the Jacobian matrix J^k . Note that $J_{ij}^k = J_{ji}^k$, $i=1, 2, \dots, n$, $j=1, 2, \dots, n$. Continuing in this manner, the component change dx_n^k is obtained.¹

$$-dx_n^k = \frac{(F(x^k) - dx_1^k J_{11}^k - dx_2^k J_{21}^k - \dots - dx_{n-1}^k J_{n-1,1}^k, J_n^k)}{J_{nn}^k} \quad (4.7)$$

$$= (F(x^k) - \frac{(F(x^k), \alpha_1^k)}{J_{11}^k} J_1^k - \frac{(F(x^k), \alpha_2^k)}{J_{22}^k} J_2^k - \dots$$

¹A comma sometimes separates the subscripts of J^k . This is used only for clarity, i.e., $J_{i,j}^k = J_{ij}^k$. This also applies to α_{ij} .

$$\begin{aligned}
& - \frac{(F(x^k), \alpha_{n-1}^k)}{J_{n-1,n-1}^k} J_{n-1}^k, J_n^k) / J_{nn}^k \\
& = [(F(x^k), J_n^k) - (F(x^k), \alpha_1^k) \frac{J_{1n}^k}{J_{11}^k} - (F(x^k), \alpha_2^k) \frac{J_{2n}^k}{J_{22}^k} - \dots \\
& \quad - (F(x^k), \alpha_{n-1}^k) \frac{J_{n-1,n}^k}{J_{n-1,n-1}^k}] / J_{nn}^k \\
& - \alpha_n^k = \frac{(F(x^k), \alpha_n^k)}{J_{nn}^k} = \frac{(\alpha_n^k)^T F(x^k)}{J_{nn}^k} \tag{4.8}
\end{aligned}$$

$$\text{where } \alpha_n^k = J_n^k - \frac{J_{1n}^k}{J_{11}^k} \alpha_1^k - \frac{J_{2n}^k}{J_{22}^k} \alpha_2^k - \dots - \frac{J_{n-1,n}^k}{J_{n-1,n-1}^k} \alpha_{n-1}^k \tag{4.9}$$

Equations 4.3, 4.6, and 4.9 could be rewritten as:

$$\begin{aligned}
\alpha_1^k & = J_1^k \\
\frac{J_{12}^k}{J_{11}^k} \alpha_1^k + \alpha_2^k & = J_2^k
\end{aligned}$$

$$\frac{J_{1n}^k}{J_{11}^k} \alpha_1^k + \frac{J_{2n}^k}{J_{22}^k} \alpha_2^k + \dots + \frac{J_{n-1,n}^k}{J_{n-1,n-1}^k} \alpha_{n-1}^k + \alpha_n^k = J_n^k$$

In matrix notation (T denotes the transpose):

$$\begin{bmatrix} 1 & & & & \\ & \frac{J_{12}^k}{J_{11}^k} & & & \\ & \vdots & & & \\ & \frac{J_{1n}^k}{J_{11}^k} & & & \\ & & \frac{J_{2n}^k}{J_{22}^k} & \dots & \frac{J_{n-1,n}^k}{J_{n-1,n-1}^k} & \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} (\alpha_1^k)^T \\ (\alpha_2^k)^T \\ \vdots \\ (\alpha_n^k)^T \end{bmatrix} = \begin{bmatrix} (J_1^k)^T \\ (J_2^k)^T \\ \vdots \\ (J_n^k)^T \end{bmatrix} = (J^k)^T$$

or

$$\begin{bmatrix} J_{11}^k & & & & \\ & J_{12}^k & & & \\ & \vdots & & & \\ & J_{1n}^k & & & \\ & & J_{22}^k & & \\ & & \vdots & & \\ & & J_{2n}^k & & \\ & & & J_{n-1,n}^k & \\ & & & & J_{nn}^k \end{bmatrix} \begin{bmatrix} \frac{(\alpha_1^k)^T}{J_{11}^k} \\ \frac{(\alpha_2^k)^T}{J_{22}^k} \\ \vdots \\ \frac{(\alpha_n^k)^T}{J_{nn}^k} \end{bmatrix} = (J^k)^T \quad (4.10)$$

where J^k is the Jacobian matrix evaluated at x^k .

Let

$$L_1^k = \begin{bmatrix} J_{11}^k & & & \phi \\ & J_{21}^k & & \\ & \vdots & J_{22}^k & \\ & J_{n1}^k & \vdots & J_{nn}^k \end{bmatrix} \quad (4.11)$$

$$\alpha^{k'} = \begin{bmatrix} \frac{(\alpha_1^k)^T}{J_{11}^k} \\ \frac{(\alpha_2^k)^T}{J_{22}^k} \\ \vdots \\ \frac{(\alpha_n^k)^T}{J_{nn}^k} \end{bmatrix} \quad (4.12)$$

then

$$\alpha^{k'} = (L_1^k)^{-1} (J^k)^T \quad (4.13)$$

From equations 4.2, 4.5, and 4.8:

$$-dx^k = \begin{bmatrix} -dx_1^k \\ -dx_2^k \\ \vdots \\ -dx_n^k \end{bmatrix} = \begin{bmatrix} \frac{(\alpha_1^k)^T}{J_{11}^k} F(x^k) \\ \frac{(\alpha_2^k)^T}{J_{22}^k} F(x^k) \\ \vdots \\ \frac{(\alpha_n^k)^T}{J_{nn}^k} F(x^k) \end{bmatrix} = \alpha^{k'} F(x^k)$$

$$-dx^k = (L_1^k)^{-1} (J^k)^T F(x^k)$$

$$\text{or } dx^k = -(L_1^k)^{-1} (J^k)^T F(x^k) \quad (4.14)$$

where dx^k is the column vector $x^{k+1} - x^k$.

In the development above, the order of the changes in the components of x^k is assumed to be 1, 2, ..., n. In the general case where that order is $i_1, i_2, \dots, i_n$, the matrix L_1^k in equation 4.14 is replaced by $L_1^{k'}$:

$$L_1^{k'} = \begin{bmatrix} J_{i_1 i_1}^k & & & \\ & J_{i_2 i_1}^k & & \\ & | & & \\ & | & & \\ & J_{i_n i_1}^k & & \\ & & J_{i_2 i_2}^k & \\ & & | & \\ & & J_{i_n i_2}^k & \\ & & & J_{i_n i_n}^k \end{bmatrix} \quad (4.15)$$

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

$$dx^k = -(P_1^k)^T (L_1^{k'})^{-1} P_1^k (J^k)^T F(x^k)$$

$$\text{or } x^{k+1} = x^k - (P_1^k)^T (L_1^{k'})^{-1} P_1^k (J^k)^T F(x^k) \quad (4.16)$$

where P_1^k is the $n \times n$ permutation matrix used at step k . The n rows of P_1^k are the n rows of the identity matrix, but correspond to the order $i_1, i_2, \dots, i_n$. That is, the first row of P_1^k is the i_1 th row of the identity matrix, the second row of P_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 :

$$\begin{aligned} -dx_1^k &= \frac{(F(x^k), J_1^k) J_{22}^k - (F(x^k), J_2^k) J_{12}^k}{J_{11}^k J_{22}^k - (J_{12}^k)^2} \\ &= \frac{(F(x^k), \beta_1^k)}{a_{12}} = \frac{(\beta_1^k)^T F(x^k)}{a_{12}} \end{aligned} \quad (4.17)$$

$$\text{where the column vector } \beta_1^k = J_{22}^k J_1^k - J_{12}^k J_2^k \quad (4.18)$$

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

$$-dx_2^k = \frac{(\beta_2^k)^T F(x^k)}{a_{12}} \quad (4.20)$$

$$\text{where the column vector } \beta_2^k = J_{11}^k J_2^k - J_{12}^k J_1^k \quad (4.21)$$

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 $F(x^k) - dx_1^k J_1^k - dx_2^k J_2^k$:

$$-dx_3^k = \frac{(F(x^k) - dx_2^k J_2^k - dx_1^k J_1^k, J_{44}^k J_3^k - J_{34}^k J_4^k)}{a_{34}} \quad (4.22)$$

$$\text{where the scalar value } a_{34} = J_{33}^k J_{44}^k - (J_{34}^k)^2 \quad (4.23)$$

Using equations 4.17 and 4.20 to substitute for dx_1^k and dx_2^k in equation 4.22:

$$-dx_3^k = \frac{(F(x^k) - \frac{(F(x^k), \beta_2^k)}{a_{12}} J_2^k - \frac{(F(x^k), \beta_1^k)}{a_{12}} J_1^k, J_{44}^k J_3^k - J_{34}^k J_4^k)}{a_{34}}$$

$$-dx_3^k = [(F(x^k), J_{44}^k J_3^k - J_{34}^k J_4^k) - (F(x^k), \beta_2^k) \frac{J_{44}^k J_{23}^k - J_{34}^k J_{24}^k}{a_{12}}]$$

$$- (F(x^k), \beta_1^k) \frac{J_{44}^k J_{13}^k - J_{34}^k J_{14}^k}{a_{12}}] / a_{34}$$

$$= \frac{(F(x^k), \beta_3^k)}{a_{34}}$$

$$= \frac{(\beta_3^k)^T F(x^k)}{a_{34}} \quad (4.24)$$

where the column vector β_3^k is given by

$$\beta_3^k = J_{44}^k J_3^k - J_{34}^k J_4^k - \frac{J_{44}^k J_{23}^k - J_{34}^k J_{24}^k}{a_{12}} - \frac{J_{44}^k J_{13}^k - J_{34}^k J_{14}^k}{a_{12}} \beta_1^k \quad (4.25)$$

Similarly:

$$-dx_4^k = \frac{(\beta_4^k)^T F(x^k)}{a_{34}} \quad (4.26)$$

where the column vector β_4^k is

$$\begin{aligned} \beta_4^k = & J_{33}^k J_4^k - J_{34}^k J_3^k - \frac{J_{33}^k J_{24}^k - J_{34}^k J_{23}^k}{a_{12}} \beta_2^k \\ & - \frac{J_{33}^k J_{14}^k - J_{34}^k J_{13}^k}{a_{12}} \beta_1^k \end{aligned} \quad (4.27)$$

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:

$$-dx_{n-1}^k = \frac{(F(x^k) - dx_{n-2}^k J_{n-2}^k - \dots - dx_1^k J_1^k, J_{nn}^k J_{n-1}^k - J_{n-1,n}^k J_n^k)}{a_{n-1,}} \quad (4.28)$$

$$\text{where the scalar value } a_{n-1,n} = J_{n-1, n-1}^k J_{nn}^k - (J_{n-1,n}^k)^2 \quad (4.29)$$

After substituting $dx_{n-2}^k, \dots, dx_1^k$ with the corresponding expressions $\frac{(F(x^k), \beta_{n-2}^k)}{a_{n-3,n-2}}, \dots, \frac{(F(x^k), \beta_1^k)}{a_{12}}$ (note that $a_{ij} = a_{ji}$) and

rearranging the terms, equation 4.28 gives:

$$-dx_{n-1}^k = \frac{(F(x^k), \beta_{n-1}^k)}{a_{n-1,n}} = \frac{(\beta_{n-1}^k)^T F(x^k)}{a_{n-1,n}} \quad (4.30)$$

where the column vector β_{n-1}^k is

$$\begin{aligned} \beta_{n-1}^k = & J_{nn}^k J_{n-1}^k - J_{n-1,n}^k J_n^k - \frac{J_{nn}^k J_{n-2,n-1}^k - J_{n-1,n}^k J_{n-2,n}^k}{a_{n-3,n-2}} \beta_{n-2}^k \\ & - \dots - \frac{J_{nn}^k J_{1,n-1}^k - J_{n-1,n}^k J_{1n}^k}{a_{12}} \beta_1^k \end{aligned} \quad (4.31)$$

Also:

$$-dx_n^k = \frac{(\beta_n^k)^T F(x^k)}{a_{n-1,n}} \quad (4.32)$$

with

$$\beta_n^k = J_{n-1,n-1}^k J_n^k - J_{n-1,n}^k J_{n-1}^k - \frac{J_{n-1,n-1}^k J_{n-2,n}^k - J_{n-1,n}^k J_{n-2,n-1}^k}{a_{n-3,n-2}} \beta_{n-2}^k$$

$$- \dots - \frac{J_{n-1,n-1}^k J_{1,n}^k - J_{n-1,n}^k J_{1,n-1}^k}{a_{12}} \beta_1^k \quad (4.33)$$

The equations for the β_i^k 's 4.18, 4.21, 4.25, 4.27, 4.31, and 4.33 could be rewritten as:

$$\beta_1^k = J_{22}^k J_1^k - J_{12}^k J_2^k$$

$$\beta_2^k = J_{11}^k J_2^k - J_{12}^k J_1^k$$

$$\frac{J_{44}^k J_{13}^k - J_{34}^k J_{14}^k}{a_{12}} \beta_1^k + \frac{J_{44}^k J_{23}^k - J_{34}^k J_{24}^k}{a_{12}} \beta_2^k + \beta_3^k = J_{44}^k J_3^k - J_{34}^k J_4^k$$

$$\frac{J_{33}^k J_{14}^k - J_{34}^k J_{13}^k}{a_{12}} \beta_1^k + \frac{J_{33}^k J_{24}^k - J_{34}^k J_{23}^k}{a_{12}} \beta_2^k + \beta_4^k = J_{33}^k J_4^k - J_{34}^k J_3^k$$

$$\frac{J_{nn}^k J_{1,n-1}^k - J_{n-1,n}^k J_{1n}^k}{a_{12}} \beta_1^k + \dots + \frac{J_{nn}^k J_{n-2,n-1}^k - J_{n-1,n}^k J_{n-2,n}^k}{a_{n-3,n-2}} \beta_{n-2}^k$$

$$+ \beta_{n-1}^k = J_{nn}^k J_{n-1}^k - J_{n-1,n}^k J_n^k$$

$$\frac{J_{n-1,n-1}^k J_{1,n-1}^k - J_{n-1,n}^k J_{1,n-1}^k}{a_{12}} \beta_1^k + \dots +$$

$$\frac{J_{n-1,n-1}^k J_{n-2,n}^k - J_{n-1,n}^k J_{n-2,n-1}^k}{a_{n-3,n-2}} \beta_{n-2}^k$$

$$+ \beta_n^k = J_{n-1,n-1}^k J_n^k - J_{n-1,n}^k J_{n-1}^k$$

The above equations could be put in matrix notation:

$$L_2^k \beta^k = M_2^k (J^k)^T \quad (4.34)$$

where

$$L_2^k = \begin{bmatrix} a_{12} & 0 & & & \\ 0 & a_{12} & & & \\ J_{44}^k J_{13}^k - J_{34}^k J_{14}^k & J_{44}^k J_{23}^k - J_{34}^k J_{24}^k & a_{34} & & \phi \\ J_{33}^k J_{14}^k - J_{34}^k J_{13}^k & J_{33}^k J_{24}^k - J_{34}^k J_{23}^k & 0 & a_{34} & \\ J_{nn}^k J_{1,n-1}^k - J_{n-1,n}^k J_{1n}^k & J_{nn}^k J_{n-2,n-1}^k - J_{n-1,n}^k J_{n-2,n}^k & a_{n-1,n} & 0 & \\ J_{n-1,n-1}^k J_{1,n}^k - J_{n-1,n}^k J_{1,n-1}^k & - & - & - & \\ - & - & - & J_{n-1,n-1}^k J_{n-2,n}^k - J_{n-1,n}^k J_{n-2,n-1}^k & 0 \ a_{n-1,n} \end{bmatrix} \quad (4.35)$$

$$\beta^{k'} =$$

$$\begin{bmatrix} \frac{(\beta_1^k)^T}{a_{12}} \\ \frac{(\beta_2^k)^T}{a_{12}} \\ \frac{(\beta_3^k)^T}{a_{34}} \\ \vdots \\ \frac{(\beta_{n-1}^k)^T}{a_{n-1,n}} \\ \frac{(\beta_n^k)^T}{a_{n-1,n}} \end{bmatrix}$$

(4.36)

$$M_2^k = \begin{bmatrix} J_{22}^k & -J_{12}^k & & & & \\ -J_{12}^k & J_{11}^k & & & & \\ & & J_{44}^k & -J_{34}^k & & \\ & & -J_{34}^k & J_{33}^k & & \\ & & & & J_{nn}^k & -J_{n-1,n}^k \\ & & & & -J_{n-1,n}^k & J_{n-1,n-1}^k \end{bmatrix} \quad (4.37)$$

$$(J^k)^T = \begin{bmatrix} (J_1^k)^T \\ (J_2^k)^T \\ \vdots \\ (J_n^k)^T \end{bmatrix} \quad (4.38)$$

From equations 4.17, 4.20, 4.24, 4.26, 4.30, and 4.32:

$$\begin{aligned}
 -dx^k &= \begin{bmatrix} -dx_1^k \\ -dx_2^k \\ \vdots \\ -dx_n^k \end{bmatrix} = \begin{bmatrix} \frac{(\beta_1^k)^T}{a_{12}} F(x^k) \\ \frac{(\beta_2^k)^T}{a_{12}} F(x^k) \\ \vdots \\ \frac{(\beta_n^k)^T}{a_{n-1,n}} F(x^k) \end{bmatrix} \\
 &= \beta^{k'} F(x^k)
 \end{aligned} \tag{4.39}$$

Then the change vector dx^k is obtained by combining equations 4.34 and 4.39:

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

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 $2 * n$ nonzero elements.

If the order of the changed components of x^k is $i_1, i_2, \dots, i_n$ instead of $1, 2, \dots, n$, then L_2^k and M_2^k in equation 4.40 are replaced by $L_2^{k'}$ and $M_2^{k'}$ respectively:

$$L_2^k = \begin{bmatrix} a_{i_1 i_2} & 0 & & \emptyset \\ 0 & a_{i_1 i_2} & & \\ J_{i_4 i_4}^k J_{i_1 i_3}^k - J_{i_3 i_4}^k J_{i_1 i_4}^k & J_{i_4 i_4}^k J_{i_2 i_3}^k - J_{i_3 i_4}^k J_{i_2 i_4}^k & a_{i_3 i_4} & \\ \vdots & \vdots & \vdots & \vdots \\ J_{i_{n-1} i_{n-1}}^k J_{i_1 i_n}^k - J_{i_{n-1} i_n}^k J_{i_1 i_{n-1}}^k & \text{---} & 0 & a_{i_{n-1} i_n} \end{bmatrix} \quad (4.41)$$

$$M_2^{k'} = \begin{bmatrix} J_{i_2 i_2}^k & -J_{i_1 i_2}^k & & \emptyset \\ -J_{i_1 i_2}^k & J_{i_1 i_1}^k & & \\ & \text{---} & J_{i_n i_n}^k & -J_{i_{n-1} i_n}^k \\ \emptyset & & -J_{i_{n-1} i_n}^k & J_{i_{n-1} i_{n-1}}^k \end{bmatrix} \quad (4.42)$$

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:

$$dx^k = -(P_2^k)^T (L_2^{k'})^{-1} M_2^{k'} P_2^k (J^k)^T F(x^k)$$

$$\text{or } x^{k+1} = x^k - (P_2^k)^T (L_2^{k'})^{-1} M_2^{k'} P_2^k (J^k)^T F(x^k) \quad (4.43)$$

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 $i_1, i_2, \dots, i_n$.

Note that the subscript i of $P_i^k, L_i^{k'}, L_i^k, M_i^{k'}, M_i^k$ 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.

C. 3-Dimensional Projection-Based Methods

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

$$-dx_1^k = \frac{1}{d_{123}} [(F(x^k), J_1^k) (J_{33}^k J_{22}^k - (J_{23}^k)^2) + (F(x^k), J_2^k)]$$

$$(J_{13}^k J_{23}^k - J_{12}^k J_{33}^k) + (F(x^k), J_3^k) (J_{23}^k J_{12}^k - J_{13}^k J_{22}^k)]$$

$$-dx_1^k = \frac{1}{d_{123}} (F(x^k), \gamma_1^k) = \frac{(\gamma_1^k)^T F(x^k)}{d_{123}} \quad (4.44)$$

where the scalar value d_{123} is

$$d_{123} = J_{11}^k J_{22}^k J_{33}^k + 2 (J_{23}^k J_{13}^k J_{12}^k) - (J_{13}^k)^2 J_{22}^k - (J_{23}^k)^2 J_{11}^k - (J_{12}^k)^2 J_{33}^k \quad (4.45)$$

and the column vector γ_1^k :

$$\begin{aligned} \gamma_1^k = & (J_{33}^k J_{22}^k - (J_{23}^k)^2) J_1^k + (J_{13}^k J_{23}^k - J_{12}^k J_{33}^k) J_2^k \\ & + (J_{23}^k J_{12}^k - J_{13}^k J_{22}^k) J_3^k \end{aligned} \quad (4.46)$$

Expressions similar to equation 4.44 are obtained for $dx_2^k, \dots, dx_n^k$ by continuing as in sections IV. A. and IV. B. Also equations for $\gamma_2^k, \dots, \gamma_9^k$ similar to 4.46 are generated and lead to the following formula:

$$L_3^k \gamma^{k'} = M_3^k (J^k)^T \quad (4.47)$$

where, assuming that n is a multiple of 3:

$$L_3^k = \begin{bmatrix} d_{123} & & & & & & & & & & \\ 0 & d_{123} & & & & & & & & & \\ 0 & 0 & d_{123} & & & & \emptyset & & & & \\ (J_{1,\rho_4}^k) & (J_{2,\rho_4}^k) & (J_{3,\rho_4}^k) & d_{456} & & & & & & & \\ (J_{1,\rho_5}^k) & (J_{2,\rho_5}^k) & (J_{3,\rho_5}^k) & 0 & d_{456} & & & & & & \\ (J_{1,\rho_6}^k) & (J_{2,\rho_6}^k) & (J_{3,\rho_6}^k) & 0 & 0 & d_{456} & & & & & \\ (J_{1,\rho_7}^k) & (J_{2,\rho_7}^k) & (J_{3,\rho_7}^k) & (J_{4,\rho_7}^k) & (J_{5,\rho_7}^k) & (J_{6,\rho_7}^k) & d_{789} & & & & \\ (J_{1,\rho_8}^k) & (J_{2,\rho_8}^k) & (J_{3,\rho_8}^k) & (J_{4,\rho_8}^k) & (J_{5,\rho_8}^k) & (J_{6,\rho_8}^k) & 0 & d_{789} & & & \\ (J_{1,\rho_9}^k) & (J_{2,\rho_9}^k) & (J_{3,\rho_9}^k) & (J_{4,\rho_9}^k) & (J_{5,\rho_9}^k) & (J_{6,\rho_9}^k) & 0 & 0 & d_{789} & & \\ & & & \text{etc.} & & & & & & & \end{bmatrix}$$

(4.48)

$$d_{ijl} = J_{ii}^k J_{jj}^k J_{ll}^k + 2 (J_{jl}^k J_{il}^k J_{ij}^k) - (J_{il}^k)^2 J_{jj}^k - (J_{jl}^k)^2 J_{ii}^k - (J_{ij}^k)^2 J_{ll}^k, \quad ij1 = 123, 456, \dots \quad (4.49)$$

$$\begin{aligned} \rho_i^k &= (J_{ll}^k J_{jj}^k - (J_{jl}^k)^2) J_i^k + (J_{il}^k J_{jl}^k - J_{ij}^k J_{ll}^k) J_j^k \\ &+ (J_{ij}^k J_{jl}^k - J_{il}^k J_{jj}^k) J_l^k, \quad ij1 = 456, 564, 645, 789, 897, \dots \end{aligned} \quad (4.50)$$

$$\gamma^{k'} = \begin{bmatrix} \frac{(\gamma_1^k)^T}{d_{123}} \\ \frac{(\gamma_2^k)^T}{d_{123}} \\ \frac{(\gamma_3^k)^T}{d_{123}} \\ \frac{(\gamma_4^k)^T}{d_{456}} \\ \vdots \\ \frac{(\gamma_n^k)^T}{d_{n-2,n-1,n}} \end{bmatrix} \quad (4.51)$$

$$M_3^k = \begin{bmatrix} M_{11}^k & -M_{12}^k & -M_{13}^k & & & & & & & \\ -M_{12}^k & M_{22}^k & -M_{23}^k & & & & & & & \\ -M_{13}^k & -M_{23}^k & M_{33}^k & & & & & & & \\ & & & M_{44}^k & -M_{45}^k & -M_{46}^k & & & & \\ & & & -M_{45}^k & M_{55}^k & -M_{56}^k & & & & \\ & & & -M_{46}^k & -M_{56}^k & M_{66}^k & & & & \\ & & & & & & M_{77}^k & -M_{78}^k & -M_{79}^k & \\ & & \emptyset & & & & -M_{78}^k & M_{88}^k & -M_{89}^k & \\ & & & & & & -M_{79}^k & -M_{89}^k & M_{99}^k & \\ & & & & & & & & & \text{etc.} \end{bmatrix}$$

(4.52)

$$M_{ii}^k = J_{jj}^k J_{11}^k - (J_{jl}^k)^2, \quad ij1 = 123, 231, 312, 456, \dots \quad (4.53)$$

$$M_{ij}^k = J_{ij}^k J_{11}^k - J_{il}^k J_{jl}^k, \quad ij1 = 123, 231, 312, 456, \dots \quad (4.54)$$

Next, the change vector dx^k is derived from equations 4.44, 4.47, and 4.51:

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

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 $3 * n$ nonzero elements.

Again, if the order of the changes in x^k is $i_1, i_2, \dots, i_n$, 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 i_v , where $v = 1, 2, \dots, n$. Also, a permutation matrix P_3^k is needed and the more general form of equation 4.55 is:

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

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

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

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:

$$x^{k+1} = x^k - (P_m^k)^T (L_m^{k'})^{-1} M_m^{k'} P_m^k (J^k)^T F(x^k) \quad (4.57)$$

where $L_m^{k'}$ is an $n \times n$ triangular matrix similar to $L_2^{k'}$ and $L_3^{k'}$ in equations 4.41 and 4.48. $M_m^{k'}$ is a symmetric matrix which has $m * n$ nonzero elements. P_m^k 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 $m = n$:

$$Q^k = (J^k)^T J^k \quad (4.58)$$

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

$$x^{k+1} = x^k - [(J^k)^T J^k]^{-1} (J^k)^T F(x^k) \quad (4.59)$$

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:

$$x^{k+1} = x^k - \frac{1}{2} A_1^k g'(x^k)^T \quad (5.1)$$

$$\text{or } x^{k+1} = x^k - \frac{1}{2} p^k \quad (5.2)$$

$$\text{where } A_1^k = (P_1^k)^T (L_1^{k'})^{-1} P_1^k \quad (5.3)$$

$$p^k = A_1^k g'(x^k)^T \quad (5.4)$$

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;

$$-g'(x^k) p^k \geq 0, \quad \forall k \geq k_0; \quad (5.5)$$

$$-\lim_{k \rightarrow \infty} \frac{g'(x^k) p^k}{\|p^k\|} = 0 \quad (5.6)$$

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 $g: D \subset R^n \rightarrow R^1$ is convex on a convex set $D_0 \subset D$ 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 $x \in R^n$. Then $r(x)$ is a strict convex functional on R^n .

Proof:

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

Definition 5.3:

A functional ϕ in R^m is said to be nondecreasing if

$$\phi(r_1, r_2, \dots, r_m) \leq \phi(r_1^i, r_2^i, \dots, r_m^i)$$

whenever $r_1 \leq r_1^i \quad \forall i.$

Theorem 5.4:

If $r_1, r_2, \dots, r_m$ are convex functionals on $D \subset R^n$ and if ϕ is a nondecreasing convex functional on R^m , then the functional g defined by

$$g(x) = \phi[r_1(x), r_2(x), \dots, r_m(x)]$$

is convex on D .

Proof:

Let x and $y \in D$. Then, for all i ,

$$r_1(\alpha x + (1-\alpha)y) \leq \alpha r_1(x) + (1-\alpha)r_1(y)$$

Therefore:

$$\begin{aligned} g(\alpha x + (1-\alpha)y) &= \phi[r_1(\alpha x + (1-\alpha)y, \dots)] \\ &\leq \phi[\alpha r_1(x) + (1-\alpha)r_1(y), \dots] \\ &\leq \alpha \phi[r_1(x), \dots] + (1-\alpha) \phi[r_1(y), \dots] \\ &\leq \alpha g(x) + (1-\alpha)g(y) \end{aligned}$$

So g is convex on D .

Theorem 5.5:

Let $\phi: \mathbb{R}^1 \rightarrow \mathbb{R}^1$ 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 ϕ is a nondecreasing convex functional of r .

Proof:

Clearly, ϕ is a nondecreasing functional of r . To prove that ϕ is convex, let

$$r(x) = r(y) - t \quad \text{where } t \in \mathbb{R}^1$$

$$x, y \in \mathbb{R}^n$$

$$\text{and } 0 \leq \alpha < 1.$$

Then

$$\begin{aligned} [\alpha r(x) + (1-\alpha)r(y)]^2 &= [\alpha r(y) - \alpha t + (1-\alpha)r(y)]^2 \\ &= [r(y) - \alpha t]^2 \\ &= r^2(y) - 2\alpha t r(y) + \alpha^2 t^2 \\ &= r^2(y) - \alpha r^2(y) + \alpha r^2(y) - 2\alpha t r(y) + \alpha^2 t^2 \end{aligned}$$

$$\begin{aligned}
&\leq (1-\alpha)r^2(y) + \alpha r^2(y) - 2\alpha tr(y) + \alpha t^2 \\
&\leq (1-\alpha)r^2(y) + \alpha[r(y) - t]^2 \\
&\leq (1-\alpha)r^2(y) + \alpha r^2(x)
\end{aligned}$$

$$\text{i.e. } \phi[\alpha r(x) + (1-\alpha)r(y)] \leq \alpha \phi[r(x)] + (1-\alpha)\phi[r(y)]$$

Therefore ϕ 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 $G: D \subset R^n \rightarrow R^1$ 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 $g: D \subset R^n \rightarrow R^1$ has a G(Gateaux)-derivative on a convex set $D_0 \subset D$. If g is convex on D_0 , then

$$[g'(y) - g'(x)](y-x) \geq 0, \quad \forall x, y \in D_0. \quad (5.8)$$

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 $g: D \subset R^n \rightarrow R^1$ has a G-derivative on a convex set $D_0 \subset D$.

Then g is convex on D_0 if and only if

$$g'(x)(y-x) \leq g(y) - g(x), \quad \forall x, y \in D_0 \quad (5.9)$$

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_1^k as defined in equation 5.3 and its inverse $(A_1^k)^{-1}$ are positive definite.

Proof:

Consider the $n \times n$ matrix L_1^k given by equation 4.11 and its transpose.

$$L_1^k + (L_1^k)^T = (J^k)^T J^k + D^k$$

where the diagonal matrix D^k is:

$$D^k = \begin{bmatrix} J_{11}^k & & & \\ & J_{22}^k & & \\ & & \phi & \\ & & & \phi \\ & & & & J_{nn}^k \end{bmatrix}$$

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 $L_1^{k'}$ (equation 4.15), A_1^k and $(A_1^k)^{-1}$ are positive definite.

Theorem 5.12:

Let $g: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^1$ be G -differentiable on a compact set $D_0 \subset D$ and assume that $A_1^{-1}: D_0 \rightarrow L(\mathbb{R}^n)$ 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:

$$g'(x)p(x) \geq c \|g'(x)^T\| \|p(x)\| \quad \forall x \in D_0 \quad (5.10)$$

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:

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

Definition 5.13:

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

$$\lim_{k \rightarrow \infty} \sigma(t_k) = 0 \text{ implies } \lim_{k \rightarrow \infty} t_k = 0.$$

Note that any nondecreasing function $\sigma: [0, \infty) \rightarrow [0, \infty)$ such that $\sigma(0) = 0$ and $\sigma(t) > 0$ for $t > 0$ is necessarily an F-function.

Definition 5.14:

Let $g: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^1$ be continuously differentiable, and assume that on some $D_0 \subset D$

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

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

$$\begin{aligned} \delta(t) &= \inf \{ \|x - y\| \mid x, y \in D_0, \|g'(x) - g'(y)\| \geq t \}, \quad t \in [0, \alpha), \\ &= \lim_{s \rightarrow \alpha^-} \delta(s), \quad t \in [\alpha, +\infty), \end{aligned}$$

is the reverse modulus of continuity of $g': D \subset \mathbb{R}^n \rightarrow L(\mathbb{R}^n, \mathbb{R}^1)$ on D_0 .

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

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

Theorem 5.15:

Let $g: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^1$ be a strictly convex differentiable functional on $D_0 \subset D$. Then the reserve modulus of continuity δ of g' is an F-function.

Proof:

Since g is strictly convex:

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

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

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 α in definition 5.14 is positive, and $\delta(t) > 0$ for all $t > 0$. Thus δ is an F-function.

Theorem 5.16:

Consider the iteration 5.2 where $\{p^k\}$ is any sequence such that $g'(x^k)p^k \geq 0$, $p^k \neq 0$, 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;

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

Proof:

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

$$\text{To prove } \lim_{k \rightarrow \infty} \frac{g'(x^k)p^k}{\|p^k\|} = 0,$$

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

$$g'(x^{k+1})\frac{1}{2}p^k \leq g(x^k) - g(x^{k+1}) \leq g'(x^k)\frac{1}{2}p^k$$

Then two situations are considered:

a/ Assume $g'(x^{k+1})\frac{1}{2}p^k \leq 0$, then

$$g'(x^k)\frac{1}{2}p^k - g'(x^{k+1})\frac{1}{2}p^k \geq g'(x^k)\frac{1}{2}p^k$$

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

Cauchy-Schwarz inequality gives:

$$\begin{aligned} \|g'(x^k) - g'(x^{k+1})\| \|p^k\| &\geq [g'(x^k) - g'(x^{k+1})]p^k \\ &\geq g'(x^k)p^k \end{aligned}$$

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

It then follows from the definition of the reverse modulus of continuity function δ that

$$\begin{aligned} \delta\left(\frac{g'(x^k)p^k}{\|p^k\|}\right) &\leq \|x^k - x^{k+1}\| \\ &\leq \frac{1}{2}\|p^k\| \end{aligned}$$

Since $g'(x^k)\frac{1}{2}\|p^k\| \geq g(x^k) - g(x^{k+1}) > 0$, there exists a $\mu_1 \in (0, 1]$ such that

$$\begin{aligned} g(x^k) - g(x^{k+1}) &\geq \mu_1 g'(x^k)\frac{1}{2}\|p^k\| \\ &\geq \mu_1 \frac{g'(x^k)p^k}{\|p^k\|} \cdot \frac{1}{2}\|p^k\| \\ &\geq \mu_1 \frac{g'(x^k)p^k}{\|p^k\|} \delta\left(\frac{g'(x^k)p^k}{\|p^k\|}\right) \\ &\geq \mu_1 t \delta(t) \end{aligned}$$

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

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

$$\text{where } \sigma_1(t) = \mu_1 t \delta(t)$$

From theorem 5.15, the reverse modulus of continuity function δ 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

$$\lim_{k \rightarrow \infty} [g(x^k) - g(x^{k+1})] = 0$$

Hence

$$\lim_{k \rightarrow \infty} \sigma_1 \left(\frac{g'(x^k)p^k}{\|p^k\|} \right) = 0$$

and

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

b/ Assume $g'(x^{k+1})_{\frac{1}{2}p^k} > 0$.

Since $g'(x^k)_{\frac{1}{2}p^k} \geq g'(x^{k+1})_{\frac{1}{2}p^k} > 0$, there exists a $\mu_2 \in (0, 1]$ such that $g'(x^{k+1})_{p^k} = \mu_2 g'(x^k)_{p^k}$.

Then

$$\begin{aligned} (1 - \mu_2)g'(x^k)_{p^k} &= [g'(x^k) - g'(x^{k+1})]_{p^k} \\ &\leq \|g'(x^k) - g'(x^{k+1})\| \|p^k\| \end{aligned}$$

or

$$\|g'(x^k) - g'(x^{k+1})\| \geq (1 - \mu_2) \frac{g'(x^k)_{p^k}}{\|p^k\|}$$

Then the definition of the reverse modulus of continuity function is used to obtain:

$$\frac{1}{2}\|p^k\| \geq \delta \left[(1 - \mu_2) \frac{g'(x^k)_{p^k}}{\|p^k\|} \right] = \delta[(1 - \mu_2)t]$$

Repeating the steps in part a/ above :

$$\begin{aligned} g(x^k) - g(x^{k+1}) &\geq \mu_1 \frac{g'(x^k)_{p^k}}{\|p^k\|} \delta[(1 - \mu_2)t] \\ &\geq \sigma_2(t) = \mu_1 t \delta[(1 - \mu_2)t] \end{aligned}$$

where $\sigma_2(t)$ is an F-function. Therefore

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

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 $g: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^1$ be continuously differentiable on a compact set $D_0 \subset D$ and suppose that $\{x^k\} \subset D_0$ is any sequence which satisfies

$$\lim_{k \rightarrow \infty} g'(x^k)^T = 0$$

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

$$\lim_{k \rightarrow \infty} [\inf_{x \in \Omega} \|x^k - x\|] = 0$$

In particular, if Ω consists of a single point x^* , then

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

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 , $i = 1, 2, \dots, n$ 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 $\{x^k\}$, $k = 0, 1, \dots, \infty$ generated by equation 5.2 converges to an x^* such that the residue $F(x^*) = 0$.

Proof:

Theorems 5.12 and 5.16 give:

$$\lim_{k \rightarrow \infty} \frac{g'(x^k)p^k}{\|p^k\|} = 0, \quad k = 0, 1, \dots$$

$$g'(x^k)p^k \geq c \|g'(x^k)\|^T \|p^k\|, \quad c > 0 \quad \text{which imply}$$

$$\lim_{k \rightarrow \infty} g'(x^k) = 0$$

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

$$\lim_{k \rightarrow \infty} x^k = x^*$$

Therefore:

$$g'(x^*) = 0 = 2F(x^*)^T J(x^*)$$

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

So $F(x^*) = 0$ and x^* 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_i^k (equation 5.3). For the 2-dimensional projection-based methods:

$$A_2^k = (P_2^k)^T (L_2^{k'})^{-1} M_2^{k'} P_2^k \quad (5.12)$$

For the 3-dimensional projection-based methods:

$$A_3^k = (P_3^k)^T (L_3^{k'})^{-1} M_3^{k'} P_3^k \quad (5.13)$$

where P_2^k , $L_2^{k'}$, $M_2^{k'}$, P_3^k , $L_3^{k'}$, $M_3^{k'}$ are defined in sections IV B and IV C.

From the development of the methods in Chapter IV, the matrices

$(L_2^{k'})^{-1} M_2^{k'}$ and $(L_3^{k'})^{-1} M_3^{k'}$ 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

$$g'(x^k) p^k \geq 0, \quad \forall k \geq k_0 \quad (5.5)$$

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:

$$A_m^k = (P_m^k)^T (L_m^{k'})^{-1} M_m^{k'} P_m^k \quad (5.14)$$

as shown in equation 4.57.

The proof of convergence for this general case then consists in showing that $(L_m^{k'})^{-1} M_m^{k'}$ 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 , $i = 1, 2, \dots, n$ 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.

Evaluation	Number of Multiplications	Number of Additions
(a) Evaluate $F(x^k)$ once		
(b) Evaluate one column J_i^k		
(c) $(J_i^k, J_i^k) = J_{ii}^k$	n	n-1
(d) $(F(x^k), J_i^k)$	n	n-1
(e) $x^{k+1} = x^k - \frac{(F(x^k), J_i^k)}{J_{ii}^k}$	1	1

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.

Evaluation	Number of Multiplications	Number of Additions
(a) Evaluate $F(x^k)$ n times		
(b) Evaluate matrix J^k once		
(c) Iteration expressions	$n(2n + 1)$	$n(2n - 1)$

Table 6.3. Number of operations needed for each cycle of 1-dimensional projection-based methods.

Evaluation	Number of Multiplications	Number of Additions
(a) Evaluate $F(x^k)$ once		
(b) Evaluate matrix J^k once		
(c) Iteration expression	$n^2(n + 2)$	$n(n^2 + \frac{n}{2} - \frac{1}{2})$

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.)

Operation/Subprogram	Time (μ sec)
Addition	3.5
Multiplication	6.5
DCOTAN	153
DEXP	138
DSIN	133
DSQRT	85.3
DLOG	161

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:

$$\begin{aligned} \sum_{j=2}^6 \cot B_1 x_j &= 0 & \text{where } B_1 &= 0.02249 \\ \sum_{\substack{j=1 \\ j \neq 2}}^6 \cot B_2 x_j &= 0 & \text{where } B_2 &= 0.02166 \\ \sum_{\substack{j=1 \\ j \neq 3}}^6 \cot B_3 x_j &= 0 & \text{where } B_3 &= 0.02083 \\ \sum_{\substack{j=1 \\ j \neq 4}}^6 \cot B_4 x_j &= 0 & \text{where } B_4 &= 0.02000 \\ \sum_{\substack{j=1 \\ j \neq 5}}^6 \cot B_5 x_j &= 0 & \text{where } B_5 &= 0.01918 \\ \sum_{j=1}^5 \cot B_6 x_j &= 0 & \text{where } B_6 &= 0.01835 \end{aligned}$$

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.

$$\begin{aligned} n &= 6 \\ \sum_{j=1}^i \left(\frac{x_j^2}{2} - \frac{1}{2} \right) &= 0, \quad i = 1, 2, \dots, 6 \end{aligned}$$

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:

$$\begin{aligned} x_1 \ln(x_2/x_1 + \text{SQRT}(1 + (x_2/x_1)^2)) + \\ 0.0072x_1x_2/19,000 - 50.0 = 0 \\ x_1 \text{SQRT}(1 + (x_2/x_1)^2) - x_1 + 0.0078x_2^2/38,000 - 2.5 = 0. \end{aligned}$$

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.

$$n = 4$$

$$x_1x_2 - 1 = 0$$

$$x_1x_3 - 1 = 0$$

$$x_1x_4 - 1 = 0$$

$$x_2x_3 - 1 = 0$$

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.

Problem	Dimension of Nonlinear System	Time (μ sec)	
		1-Dim Basic Projection Methods	1-Dim Projection- Based Methods
6.1	6	34,956	12,974
6.2	6	5,778	3,521
6.3	2	1,112	818
6.4	4	492	909

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 under ASP with no significant difference in the results. Double precision was used.

Table 6.6. Test results for 1-dimensional basic projection methods (LDBPM) and for 1-dimensional projection-based methods(LDPBM).

Problem	Norm of Residue	Order of Projection	LDBPM Cycles Time		LDPBM Cycles Time	
6.1	0.006050	6,1,2,3,4,5	19	3.69	16	2.63
6.2	0.002610	1,2,3,4,5,6	20	1.83	8	1.10
6.3	$< 10^{-13}$	1,2	5	0.08	4	0.07
6.4	$< 10^{-7}$	1,2,4,3	27	0.99	27	1.73

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(LDPBM) converge faster, i.e. in less time and less number of cycles, than the 1-dimensional basic projection methods(LDBPM) for problems 6.1 through 6.3. For problem 6.4, LDBPM require less time than LDPBM, 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 LDPBM to solve such a system is not advantageous. Table 6.6 indicates that the rate of convergence of LDPBM is faster than that of LDBPM, 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 2DPBM over 2DBPM 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 S360/65 under HASP and used double precision. Results are shown in table 6.7.

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

Problem	Norm of Residue	Order of Projection	2DBPM Cycles Time		2DPBM Cycles Time	
6.1	0.003596	1 2,3 4,5 6	35	5.32	18	3.49
6.2	0.002610	1 2,3 4,5 6	15	2.15	13	1.43
6.3	$< 10^{-13}$	1 2	3	0.06	3	0.06
6.4	$< 10^{-7}$	1 2,3 4	24	1.13	24	1.60

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 2DBPM converge faster than 2DPBM in this case. In practical nonlinear systems with complex equations, 2DPBM would have a higher speed of convergence than 2DBPM 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.

Evaluation	Number of Multiplications	Number of Additions
(a) Evaluate $F(x^k)$ once		
(b) Evaluate J^k once		
(c) Evaluate $(J^k)^{-1}$	$\frac{n^4}{4} + n^3 - \frac{n^2}{3}$	$\frac{n^4}{3} + \frac{n^3}{2} - \frac{5n^2}{6}$
(d) Iterative expression	n^2	n^2

Tables 6.3 and 6.8 show that during each iteration cycle, both LDPBM 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 LDPBM is of the order n^3 . Therefore, when n is large, one iteration cycle of LDPBM 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.

VII. SUMMARY AND FUTURE RESEARCH

A. Summary

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 $L_1^{k'}$ in equation 4.16, $L_2^{k'}$ and $M_2^{k'}$ in equation 4.43, $L_3^{k'}$ and $M_3^{k'}$ 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

1. S. I. Al'ber and Ya. I. Al'ber, A method of differential descent for solving nonlinear systems, USSR Comput. Math. and Math. Phys., 7(1967), 15-40.
2. Y. Bard, Comparison of gradient methods for the solution of nonlinear parameter estimation problems, SIAM J. Numer. Anal., 7(1970) 157-186.
3. K. M. Brown, A quadratically convergence Newton-like method based upon Gaussian elimination, SIAM J. Numer. Anal., 6(1969), 560-569.
4. K. M. Brown and J. E. Dennis, Jr., On Newton-like iteration functions: general convergence theorems and a specific algorithm, Numer. Math., 12(1968), 186-191.
5. C.G. Broyden, A new method of solving nonlinear simultaneous equations, Comput. J., 12(1969), 94-99.
6. M. A. Cauchy, Méthode générale pour la résolution des systèmes d'équations simultanées, C. R. Acad. Sci., 25(1847), 536-538.
7. H. B. Curry, The method of steepest descent for nonlinear minimization problems, Quart. Appl. Math., 2(1944), 258-261.
8. J. Daniel, Convergence of the conjugate gradient method with computationally convenient modification, Numer. Math., 10(1967), 125-131.
9. W. Davidon, Variable metric methods for minimization, A.E.C. Res. and Develop. Rept. ANL-5990(1959), Argonne Nat'l Lab., Argonne, Illinois.
10. R. Fletcher and M. Powell, A rapidly convergent descent method for minimization, Comput. J., 6(1963), 163-168.
11. R. Fletcher and C. Reeves, Function minimization by conjugate gradients, Comput. J., 7(1964), 149-154.
12. A. de la Garza, An iterative method for solving systems of linear equations, Union Carbide and Carbon Corp. K-25 Plant, Oak Ridge, Tenn., Report K-731, 1951.
13. D. D. Georg, and R. F. Keller, Determining nonlinear projection subspace, National Technical Information Service, USAEC Report No. IS-3545, 1975.

14. A. S. Householder, KWIC index for the numerical treatment of nonlinear equations, Oak Ridge National Laboratory, Oak Ridge Tenn., Report ORNL-4595, 1970.
15. IBM S/360 Fortran IV Library Subprograms, Form No. GN 28-0588.
16. L. V. Kantorovich, Functional analysis and applied mathematics, Uspehi Mat. Nauk, 3(1948), 89-185.
17. K. Levenberg, A method for the solution of certain nonlinear problems in least squares, Quart. Appl. Math., 2(1944), 164-168.
18. D. G. Luenberger, The conjugate residual method for constrained minimization problems, SIAM J. Numer. Anal., 7(1970), 390-398.
19. A. MacEachern and R. F. Keller, A projection method for solving nonlinear systems of equations, National Technical Information Service, USAEC Report No. IS-3040, 1972.
- 20. L. V. Nguyen, D. D. Georg, and R. F. Keller, The projection methods: A class of minimization methods for solving nonlinear systems of equations, paper submitted to SIAM J. Numer. Anal., 1975.
21. J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.
22. S. Perlis, Theory of Matrices, Addison-Wesley Publishing Co., Inc., Reading, Mass., 1952.
23. H. Rosenbrock, An automatic method for finding the greatest or least valued of a function, Comput. J., 3(1960), 175-184.
24. J. Schmidt and H. Trinkaus, Extremwertermittlung mit Funktionswerten bei Funktion von mehreren Veranderlichen, Computing, 1(1966) 224-232.
25. M. White, A class of projection methods for solving systems of n nonlinear equations in n unknowns allowing projections of order one through n, unpublished Ph.D. thesis, Ames, Iowa Library, Iowa State University of Science and Technology, 1971.