1971

A class of projection methods for solving systems of n nonlinear equations in n unknowns allowing projections of order one through n

Michael William White
Iowa State University

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

Part of the Mathematics Commons

Recommended Citation
https://lib.dr.iastate.edu/rtd/5199

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.
WHITE, Michael William, 1943-
A CLASS OF PROJECTION METHODS FOR SOLVING
SYSTEMS OF N NONLINEAR EQUATIONS IN N
UNKNOWNs ALLOWING PROJECTIONS OF ORDER ONE
THROUGH N.

Iowa State University, Ph.D., 1971
Mathematics

University Microfilms, A XEROX Company, Ann Arbor, Michigan

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED
A class of projection methods for solving systems of $n$ nonlinear equations in $n$ unknowns allowing projections of order one through $n$

by

Michael William White

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

Major Subject: Mathematics

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 of Science and Technology Ames, Iowa

1971
PLEASE NOTE:

Some Pages have indistinct print. Filmed as received.

UNIVERSITY MICROFILMS
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>INTRODUCTION AND REVIEW</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>A. Introduction</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>B. Review of Progress and Literature Review</td>
<td>7</td>
</tr>
<tr>
<td>II.</td>
<td>DEVELOPMENT</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>A. Preliminaries</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>B. Development</td>
<td>32</td>
</tr>
<tr>
<td>III.</td>
<td>CONVERGENCE</td>
<td>39</td>
</tr>
<tr>
<td>IV.</td>
<td>GEOMETRICAL INTERPRETATION</td>
<td>49</td>
</tr>
<tr>
<td>V.</td>
<td>ALGORITHM</td>
<td>53</td>
</tr>
<tr>
<td>VI.</td>
<td>EXAMPLES AND COMPARISONS</td>
<td>55</td>
</tr>
<tr>
<td>VII.</td>
<td>CONCLUSIONS AND FUTURE RESEARCH</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>A. Conclusions</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>B. Future Research</td>
<td>91</td>
</tr>
<tr>
<td>VIII.</td>
<td>BIBLIOGRAPHY</td>
<td>94</td>
</tr>
<tr>
<td>IX.</td>
<td>ACKNOWLEDGEMENTS</td>
<td>99</td>
</tr>
</tbody>
</table>
I. INTRODUCTION AND REVIEW

A. Introduction

In order to introduce with due emphasis and proper motivation the subject to be considered in the succeeding pages, a few very interesting and provocative quotes are offered.

"Many problems from the nonlinear theory of elasticity and the theory of plasticity, etc. are reduced by various methods to the solution of nonlinear systems of equations of the form \( f_k(x_1, \ldots, x_n) = 0 \), \((k = 1, 2, \ldots, n)\), where the functions \( f_k(x), \) \((x = (x_1, x_2, \ldots, x_n))\), are given in the Euclidean space \( \mathbb{E}_n \)" (60, p. 29). "One of the fascinating aspects of nonlinear analysis is the unity it brings to a surprising variety of diverse problems in science. Nonlinear phenomenon arise [sic] naturally in geometry, physical science, economics, the life sciences, and ordinary and partial differential equations" (14, p. 25). "The solution of a set of nonlinear simultaneous equations is often the final step in the solution of practical problems arising in physics and engineering" (18, p. 577). "Nonlinear statistics may be expected to assume increasing importance in chemical engineering... A nonlinear statistical model, which, although requiring more computational effort, is likely to yield more meaningful results, and more rapid technical progress" (49, pp. 65-66). "Two-point boundary value problems for nonlinear ordinary differential equations occur naturally and frequently in applied mathematics, physics and engineering. For example, many problems in flight mechanics,
optimization, and control theory, when attacked by the calculus of variations, lead to two-point boundary value problems" (59, p. 402). "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" (17, p. 186).

Comments such as these are much the rule rather than the exception, particularly in the fields mentioned in the quotes above. Such comments provide sufficient motivation for investigating methods for solving systems of nonlinear equations. Now, in order to more graphically illustrate that which is to be attacked, and how they arise in some of the areas mentioned above, several typical problems which will be solved explicitly in Chapter VI are presented.

From an article by F. H. Deist and L. Sefor (27, p. 81) comes the following example.

"Consider the following set of equations:

\[ f_i = \sum_{j=1}^{n} F_{ij} = 0, \quad (i = 1, 2, \ldots, n), \]

where

\[ F_{ij} = \cot B_i x_j, \quad \text{for} \ i \neq j \]
\[ F_{ij} = 0 \quad \text{for} \ i = j. \]

These equations are used in the design of a VHF aerial feeder system. The \( x_j \)'s are the lengths associated with the
coaxial line connectors and the $B_i$'s are constants dependent on the carrier frequency" (see Chapter VI, Problem 4).

In a book published in 1928, Fr. A. Willers (64, p. 177) presents the following problem.

An elastic steel wire is stretched between two rigid supports which are $d = 2w$ meters apart. The wire has a sag of $h$ meters. What is its length $2L$ and the horizontal tension $H$?

Introducing the variable $a = H/g$ then the following equations hold:

$$w = a \ln(\frac{L}{a} + (1 + \frac{L^2}{a^2})^{\frac{1}{2}}) + \frac{agL}{E}$$
$$h = a((1 + \frac{L^2}{a^2})^{\frac{1}{2}} - 1) + g\frac{L^2}{2E}$$

where $E$ and $g$ are given elasticity constants, (see Chapter VI, Problem 3).

R. H. Moore in (7, p. 90) considers the equation for the $H$-function in the theory of radiative transfer:

$$\frac{1}{H(u')} = 1.0 - \frac{w}{2} \int_{0}^{1} \frac{u}{u+u'} H(u')du',$$

which he writes in the form

$$x(s) + \frac{w}{2} \sum_{0}^{1} \frac{s}{s+t} \frac{1}{x(t)} dt = 1.$$

Using $n$-point Gaussian quadrature this yields the system

$$f_i = x_i + \frac{w}{2} \sum_{j=1}^{n} \frac{s_i}{s_i+s_j} \frac{1}{x_j} A_j - 1.0 = 0, \ i = 1,2,\ldots,n,$$

where the $A_i$'s are the Gaussian weights, the $s_i$'s are the
Gaussian abscissas and \( x_i = x(s_i) \), \( i = 1, 2, ..., n \), (see Chapter VI, Problem 7).

A typical optimal control problem is outlined by Fournier and Groves (32) as follows. Consider the system defined by \( x = F(x,u,t) \) subject to initial conditions \( x = x_0 \) at \( t = t_0 \), where \( x \) is the state vector, \( u \) is the control vector and \( t \) is the time. Find the control \( u(t) \) that minimizes the performance index

\[
J = \int_0^T L(x,u,t) \, dt.
\]

We treat the system equations as constraints and using Lagrange multipliers \( \lambda(t) \), adjoin the system equations to the performance index to get a modified performance index

\[
K = \int_0^T (L(x,u,t) + \lambda(F(x,u,t) - x)) \, dt.
\]

Now the problem is to minimize \( K \). Discretizing the interval, approximating the integral and derivatives numerically we get for the performance index

\[
K = (L(x_0,u_0,t_0) + \lambda_0(F(x_0,u_0,t_0) - x_0)) \Delta t +
(L(x_1,u_1,t_1) + \lambda_1(F(x_1,u_1,t_1) - x_1)) \Delta t +
\cdots
+ (L(x_{n-1},u_{n-1},t_{n-1}) + \lambda_{n-1}(F(x_{n-1},u_{n-1},t_{n-1}) - x_{n-1})) \Delta t.
\]

The equations to be solved are then
Solution of these equations will yield a possible optimal control sequence \( u_0, u_1, \ldots, u_{n-1} \), along with the optimal state trajectory and Lagrange multipliers, (see Chapter VI, Problem 9).

T. A. Porsching (58) considers an interesting area of application called nonlinear network problems. He states, "Nonlinear network problems occur in several engineering areas - for example, in fluid mechanics and electrical circuit theory."

Underlying network problems is a set of nodes \( N = \{1, 2, \ldots, n\} \) and a set of pairs \((j, k)\), \(j, k \) in \( N\), called links representing the existing connections between nodes. Also, we recognize a nonvoid subset of \( N\) as the set of bound-nodes denoted by \( \partial N \).

For each link there is a pair of conductance functions \( G_{jk}(s,t) \) and \( G_{kj}(t,s) \) with certain properties. \( G_{jk}(s,t) \) may be regarded as the directed flow from node \( j \) to node \( k \) over link \((j, k)\).

Let \( u = (u_1, u_2, \ldots, u_n) \) be a state vector. Then the net efflux from node \( j \) is
The network problem of interest is to determine a state $u$ which satisfies

$$f_j(u) = q_j, \quad j \in \mathbb{N}$$

$$u_j = F_j(f_j(u)), \quad j \in \mathbb{N},$$

where the $q_j$'s are prescribed constants and the $F_j(s)$'s are continuous functions defined on $(-\infty, \infty)$, (see Chapter VI, Problem 5).

I offer one final example in which it becomes necessary to solve a system of nonlinear equations. It is, of course, obvious that these are but a touch on the surface of the vast number of problems which give rise to a system of nonlinear equations in their solution process. This example deals with a system of boundary value equations and comes from a paper by S. M. Roberts and J. S. Shipman (59, p. 406). Their problem deals with the two body equations of motion

$$\frac{d^2x}{dt^2} = -\frac{Kx(t)}{r^3},$$

$$\frac{d^2y}{dt^2} = -\frac{Ky(t)}{r^3},$$

$$\frac{d^2z}{dt^2} = -\frac{Kz(t)}{r^3},$$
where \( r = (x^2(t) + y^2(t) + z^2(t))^{\frac{1}{3}} \), \( k \) is a prescribed constant and

\[
\begin{align*}
x(0) &= x_0, & y(0) &= y_0, & z(0) &= z_0, \\
x(t_f) &= x_{t_f}, & y(t_f) &= y_{t_f}, & z(t_f) &= z_{t_f}.
\end{align*}
\]

By discretizing, the problem is reduced to solving a system of nonlinear difference equations

\[
\begin{align*}
a_kx_{n+k} + a_{k-1}x_{n+k-1} + \cdots + a_0x_n - h^2(b_k(-Kx_{n+k}/r_n^{3k} + \cdots + b_0(-Kx_n/r_n^{3k}) &= 0, \\
a_ky_{n+k} + a_{k-1}y_{n+k-1} + \cdots + a_0y_n - h^2(b_k(-Ky_{n+k}/r_n^{3k}) + \cdots + b_0(-Ky_n/r_n^{3k}) &= 0, \\
a_kz_{n+k} + a_{k-1}z_{n+k-1} + \cdots + a_0z_n - h^2(b_k(-Kz_{n+k}/r_n^{3k}) + \cdots + b_0(-Kz_n/r_n^{3k}) &= 0,
\end{align*}
\]

for \( n = 0, 1, \ldots, N \), and for some fixed choice of the \( a_i \)'s and \( b_i \)'s, (see Chapter VI, Problem 8).

B. Review of Progress and Literature Review

The problem of solving nonlinear equations and systems of nonlinear equations is certainly not a recent one. Although there were procedures for finding roots of single nonlinear equations prior to the time of Sir Isaac Newton, so very much of the work available to us today bears his name that his era seems the most logical place to begin in taking a brief
look at the efforts made in an assault on the problem.

Newton, in 1669, on pages 268 and 269 of a tract, De Analysi per Aequationes Numero Terminorum Infinitas, contained in Volume One of his Opera (edition by Horsley), explained his method of approximation to the real roots of equations. A study of Newton's works, however, reveals that his method of approximation is not the same as what is today called 'Newton's method', that is, if $r$ is the initial approximation to a root of the equation $f(x) = 0$, then

$$
r_1 = r - \frac{f(r)}{f'(r)}, \quad r_2 = r_1 - \frac{f(r_1)}{f'(r_1)}, \text{ etc.}
$$

This modification of Newton's process was first made by Joseph Raphson in 1690. "In view of the facts it is doubtful whether the method of approximation described by Raphson should be named after Newton alone. In the first place, the processes used by Newton, though not identical to that of Vieta, resembles [sic] it. Newton merely simplified the division used. Hence the honor of invention falls largely on Vieta. In the second place, Newton did not develop his method further than simply to solve the cubic $y^3 - 2y - 5 = 0$. That Raphson worked independently of Newton we doubt. But Raphson's version of the process has been accepted as an improvement. It would seem, therefore, that the 'Newton-Raphson method' would be a designation more nearly representing the facts of history than is 'Newton's method' " (21, p. 31).
Howsoever this may be, for his small effort Newton has had his name affixed to the method of the form

\[ x_{n+1} = x_n - (P'(x_n))^{-1}P(x_n) \]

whether for algebraic functions of single or vector variables, or functions on arbitrary Banach spaces or any other type.

There is very little mention of Newton's method from his time until shortly after the turn of the twentieth century. R. H. Moore comments, "The method has a very long history, with contributions by Cauchy, Runge, Faber, Blutel among others. The theorem of Fine (29) in 1919 seems to be the first in n-dimensional space which, under conditions given for an initial approximate solution, asserts the existence of a solution of \( P(x) = 0 \) to which the iterates in the Newton method converge. In the same year, Bennett (13) proved a convergence and existence theorem in more general spaces. Ostrowski (56) in 1937 gave error estimates in n-dimensional space. In 1939 Kantorovich [sic] (41) gave theorems for equations in a space with norm in a partially ordered space" (7, p. 66).

This roughly brings Newton's method up to 1940 and very nearly brings up to date all work on solving algebraic systems of nonlinear equations in this period. This period is ended and the 'modern' era begun by the classical theorem of Kantorovich (40, p. 167) often referred to as the Newton-Kantorovich theorem. In it he places conditions upon the
nonlinear operator $P$ mapping a linear normed space $X$ into a linear normed space $Y$ in order to guarantee convergence of Newton's method to a solution $s$ of the equation

$$P(x) = 0.$$ \hspace{1cm} 1.1

In relation to the first period noted above, results after Kantorovich's theorem begin to appear more frequently, "The literature on Newton's method after the fundamental paper of Kantorovich is large" (34, p. 391), however, the relative value of these results is not assessed. M. Altman (6) presented a generalization of the Newton-Kantorovich theorem for a nonlinear operator $P$ mapping a Banach space $X$ into a Banach space $Y$ and $P'(x)$ is the Frechet derivative of $P$ at $x$. Altman (5) also presents a reformation of the problem for nonlinear operators $P$ on arbitrary Hilbert spaces and presents the 'modified' Newton method

$$x_{n+1} = x_n - (P'(x_0))^{-1}P(x_n),$$ \hspace{1cm} 1.2

using the inverse Frechet derivative at the initial point only (4), and several other related results in other articles.

In 1954, M. I. Necepurenko (52) investigated an extension of Newton's method, sometimes referred to as the method of tangent parabolas, and presented a theorem for it similiar to the aforementioned Newton-Kantorovich theorem. The form of Necepurenko's iteration used to solve 1.1 is
\[ x_{n+1} = x_n - R_n P(x_n) - \frac{1}{2} R_n P''(x_n)(R_n P(x_n))^2, \]

where \( R_n = (P'(x_n))^{-1} \), and \( P \) maps the Banach space \( X \) into the Banach space \( Y \) such that its second Frechet derivative exists.

A process called the method of tangent hyperbolas was investigated by M. A. Mertvecova (51) in 1953. Again, the process is accompanied by a theorem similar to that of Kantorovich. The process presented by Mertvecova uses the following iteration

\[ x_{n+1} = x_n - Q_n R_n P(x_n) \]

where

\[ Q_n = (I - \frac{1}{2} R_n P''(x_n) R_n P(x_n))^{-1} \quad \text{and} \quad R_n = (P'(x_n))^{-1}. \]

T. I. Kogan (44) and M. Ya. Bartish (12) investigated interpolative methods. Bartish used the iteration

\[ x_{n+1} = x_n - R_n P'(x_n + \frac{1}{2} R_n P(x_n))R_n P(x_n), \]

and Kogan used

\[ x_{n+1} = x_n - P'(x_n - \frac{1}{2} R_n P(x_n))P(x_n), \]

both with \( R_n = (P'(x_n))^{-1} \), but Kaazik (39) investigates a method with the most complicated iterations

\[ x_{n+1} = x_n - (I + 2Q_n)^{-1}(I + 3Q_n)R_n P(x_n), \]
with $R_n$ as before and

$$Q_n = \frac{1}{2} R_n P''(x_n) R_n P(x_n).$$

Each of these methods, in addition to the basic Newtonian process has one inherent drawback which arises if, "... the function is so complicated that it is either impracticable or impossible to obtain explicit expressions for the elements of its Jacobian" (19, p. 94). In an effort to eliminate this problem, Broyden continues, "In recent years this problem has been pursued in two distinct but complementary ways. In the first, various modifications of Newton's method have been sought with a view to reducing the amount of labour involved in evaluating the Jacobian and solving the resulting set of linear equations. Most variations of Newton's method though have relied on using some approximation to the inverse Jacobian and modifying this 'iteration matrix' at each stage of the process" (19, p. 94).

And thus are introduced the so-called quasi-Newton methods. The very simplest such method has already been introduced as the modified Newton method with iteration given in 1.2.

Another such method investigated by Philip Wolfe (65) and J. G. P. Barnes (11) called the secant method involves arriving at some initial guess for the Jacobian $J^{(1)}$ and solving

$$x_{n+1} = x_n - (J^{(1)})^{-1} P(x_n).$$
The process also involves solving for a change matrix $D$ for the Jacobian approximation so that

$$J(i+1) = J(i) + D$$

and repeating.

Several methods have been proposed in which difference approximations are substituted for the elements of the Jacobian matrix. One such typical method is presented by V. E. Shamanskii (60) in which he replaces the Jacobian by

$$R(h,x) = \begin{pmatrix}
\frac{P(x+he_1) - P(x)}{h} & \frac{P(x+he_2) - P(x)}{h} & \cdots \\
\frac{P(x+he_n) - P(x)}{h}
\end{pmatrix}$$

where the $e_i$'s are the unit vectors of length $n$.

The ultimate generalization in methods of this type is investigated by H. A. Antosiewicz (8), for one, who asserts that for any sequence $(T_n)$ of linear homeomorphisms of $X$ onto $Y$ satisfying

$$||P'(x_0)^{-1}|| \cdot ||T_n - P'(x_0)|| \leq c$$

for a constant $c$ satisfying certain bounds the iteration

$$x_{n+1} = x_n - T_n^{-1}P(x_n)$$

converges. This idea was also extended by Dennis (28) to
the modified Newton's method using the iteration

\[ x_{n+1} = x_n - MP(x_n) \]

where \( M \) is not necessarily \((P'(x_0))^{-1}\) but convergence can be guaranteed under certain assumptions on \( M \).

Another approach to the problem deals with how one goes about solving the system

\[ (P'(x_n)) \Delta x_n = P(x_n) \]

and the methods resulting from this type of consideration are termed generalized linear methods.

These methods derive their more specific names from the familiar linear iteration techniques employed and are therefore more explicitly named the Newton-Gauss-Seidel, the Newton-Jacobi, the Newton-SOR and the Newton-alternating direction methods.

Write

\[ P'(x_n) = D - L - U \]

where \( D, L, U \) are diagonal, strictly lower and strictly upper triangular matrices, respectively. Using this decomposition the Newton-Jacobi iteration becomes

\[ x_{n+1} = D^{-1}(L + U)x_n + D^{-1}P(x_n). \]

The Newton-Gauss-Seidel method utilizes the following iteration
\[ x_{n+1} = (D - L)^{-1}(Ux_n + P(x_n)) \]

and the Newton-SOR process appears as

\[ x_{n+1} = (D - wL)^{-1}((1-w)D + wU)x_n + w(D - wL)^{-1}P(x_n) \]

where \( w \) is the relaxation parameter.

For the alternating direction method of Peaceman and Rachford the form is

\[ x_{n+1} = x_n - 2a(V + aI)^{-1}(H + aI)(P'(x_n)x_n - P(x_n)) \]

where \( P'(x_n) = H + V \) and \( a \) is a parameter. A good general discussion of all these methods is contained in chapter seven of a very recent book by Ortega and Rheinboldt (53). More specific work has been done by K. M. Brown (15) and Brown in conjunction with Samuel D. Conte (16), Charles A. Bryan (20), Ortega and Rheinboldt (54), Ortega and Maxine Rockoff (55) and T. A. Porsching (58).

I now return to the comment by Broyden on page 12 in which he states that the first technique used to circumvent the problem of finding the inverse Jacobian is to modify the Jacobian of the function in some way so that the Jacobian no longer explicitly appears. Broyden continues, "The second way in which the solution of nonlinear simultaneous equations has progressed in recent years is based on an idea due
originally to Davidenko (25) and rediscovered on a number of subsequent occasions. Essentially, a vector function $g(x, \theta)$, where $\theta$ is some scalar parameter, is constructed such that $g(x, 0) = f(x)$ and the equation $g(x, 1) = 0$ has a known solution.

If $x$ is a solution of $g(x, \theta) = 0$ then clearly $x$ is a function of $\theta$ and by reducing $\theta$ incrementally from 1 to 0 a series of intermediate problems are constructed which may be solved, it is argued more readily than the original macro-problem" (19, p. 94).

More specifically, for the problem 1.1, $P$ may depend naturally on a parameter $t$, such that when $t = 1$, the mapping $P$ results and when $t = 0$ the resulting system $P_0(x) = 0$ has a known solution $x_0^*$, and there is a family

$$H: \mathbb{D} \times [0,1] \subset \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$$

such that $H(x, 0) = P_0(x)$ and $H(x, 1) = P(x)$ for all $x \in \mathbb{D}$, where a solution $x_0^*$ of $H(x, 0) = 0$ is known, and the equation $H(x, 1) = 0$ is to be solved.

Even if $P$ does not depend on a parameter $t$ naturally we can define a family $H$ in various ways, for example

$$H(x, t) = tP(x) + (t-1)P_0(x), \quad x \in \mathbb{D}, \; t \in [0,1].$$

Considering

$$H(x, t) = 0, \quad t \in [0,1],$$
we suppose this equation has a solution $x = x(t)$ depending continuously on $t$ for each $t \in [0,1]$, or there is continuous $x: [0,1] \rightarrow D$ such that

$$H(x(t),t) = 0 \text{ for all } t \in [0,1].$$

Then $x(t)$ describes a curve in $\mathbb{R}^n$ with one end point at $x_0$ and the other at a solution $x^* = x(1)$ of $P(x) = H(x,1) = 0$ found by numerical integration.

Various investigations have been made on the choice of the function $H$ and the method of numerical integration to be used. See for example Davidenko (25), Ortega and Rheinboldt (53, p. 230), F. H. Deist and L. Sefor (27), Freudenstein and Roth (33), H. Kleinmichel (43) and William Kizner (42).

To this point we have examined Newton's method and its offshoots, the twists, turns and contortions performed to make the execution simpler. But basically they all stem from Newton's method. So one might reasonably ask if this is the only way to go. Thankfully, the answer is no. Another class of methods has been under investigation for some time. The methods of this class are called minimization methods due to the fact that the problem of finding $s$ such that $P(s) = 0$ is shifted to finding $s$ such that $Q(x)$, where $Q: \mathbb{R}^n \rightarrow \mathbb{R}$, has a minimum at $s$. That such $Q$ functions may be found related to $P$ is simply illustrated by choosing

$$Q(x) = \sum_{i=1}^{n} f_i^2(x) \quad 1.3$$
where \( f_i, i = 1, 2, \ldots, n \), are the component functions of \( P \).

Minimization methods may be divided into two basic groups, descent methods and conjugate-direction methods. The minimization concept was introduced in 1847 by M. Augustin Cauchy (22) when he introduced the descent method called the gradient method or method of steepest descent.

Cauchy began with the system \( p_i(x) = 0, i = 1, 2, \ldots, n \), and formed \( Q(x) \) as in 1.3. He considered a bounded region \( S \) of \( \mathbb{R}^n \) whose boundary is a Jordan curve \( \partial S \) such that \( Q(x) \), \( Q'(x) \) and \( Q''(x) \) are continuous on \( S \cup \partial S \) and \( Q(x) = c \) for all \( x \) in \( S \) and \( Q(x) < c \) for all \( x \) in \( S \). Now starting from a point \( x_0 \) on \( S \), Cauchy proposed going along the normal to \( S \) at \( x_0 \) into \( S \) a segment of length \( r_0 \) with end point \( x_1 \), the value of \( Q(x) \) diminishing from \( c = c_0 \) to \( c_1 < c_0 \). Now pass a level surface \( S_1 \) of \( Q(x) \) through \( x_1 \) such that \( Q(x) = c_1 \) for all \( x \) in \( S_1 \), and repeat the above process until the sequence \( x_n, n = 0, 1, \ldots, \) converges to a point \( s \) such that \( Q(s) \) is the minimum of \( Q(x) \) and

\[
\frac{\partial Q}{\partial x_i}(s) = 0, \quad i = 1, 2, \ldots, n.
\]

Cauchy's contentions of convergence of the method described were proven in 1944 by Haskell B. Curry (23).

All minimization methods are characterized by the formula

\[
x_{n+1} = x_n + c_n d_n,
\]
where \( c_n \) is a constant and \( d_n \) is a direction. The methods are then categorized upon how to choose the constants and the directions. Choosing \( d_n \) as the direction of the normal is the steepest descent method as described above. Choosing the directions along the coordinate axes produces a process known as the univariate or relaxation method.

The general form of descent methods comes from a generalization of methods due to W. Davidon (26). This method called the Davidon-Fletcher-Powell method admits iteration of the form

\[
x_{n+1} = x_n - a_n B_n P(x_n)
\]

\[
B_{n+1} = B_n + \frac{r_n (r_n)^T}{(r_n)^T q_n} - \frac{(B_n q_n) (B_n q_n)^T}{(q_n)^T (B_n q_n)}
\]

where \( B_0 \) is an arbitrary symmetric positive definite matrix (usually the identity matrix) and

\[
r_n = x_{n+1} - x_n,
\]

\[
q_n = P(x_{n+1}) - P(x_n).
\]

Conjugate gradient methods are characterized by the Daniel algorithm and the Fletcher-Reeves method. For these assume that \( g : D \subset \mathbb{R}^n \to \mathbb{R}^1 \) is twice Frechet-differentiable. The Daniel algorithm is then given by
\( x_{n+1} = x_n - a_n p_n \),

\( g(x_{n+1}) = \min(g(x_n - a p_n) \mid x_n - a p_n \in D) \),

\( p_{n+1} = (g'(x_{n+1}))^T - B_n p_n \),

\( B_n = g''(x_{n+1})(g'(x_n))^T p_n / g''(x_{n+1})p_n p_n \),

where \( p_0 = (g'(x_0))^T \).

The Fletcher-Reeves system differs only in the calculation of the \( B_n \)'s and this formula is

\[
B_n = g'(x_{n+1})(g'(x_{n+1}))^T / g'(x_n)(g'(x_n))^T
\]

which has the advantage of not requiring \( g'' \). Also see James W. Daniel (24) and R. Fletcher and C. M. Reeves (31).

As one can easily see, minimization processes can be somewhat complicated and in general are slow to converge. Efforts have, of course, been made to remedy these problems. See, for example, S. I. Al'ber and Ya. I. Al'ber (2 and 3), Yonathan Bard (9 and 10), M. N. Jakovlev (38) and David Luenberger (45).

The previous few paragraphs have been designed to give a quick review of the literature concerned with work in the area of solutions of nonlinear equations. It, of course, is not a complete view by any means but serves to rough out an idea of how we stand in the area. Unfortunately, most of the methods available have some major criticism directed
against them, too slow, too complicated - one thing or another. Consequently, much work yet needs to be done.

To complete this review there are several excellent bibliographic sources worthy of mention. These are Anselone (7), Householder (36 and 37), Kantorovich (40) and Ortega and Rheinboldt (53).

I do not propose the work to be presented in the following pages is the complete answer, however, it attacks the problem from a different angle and has certain advantages. It, unfortunately, has certain weaknesses also, but a new concept offers, in addition, the possibility of more new avenues being opened.

Alexander MacEachern, in his unpublished Ph.D. thesis (46), explores a minimization method such that \( Q(x_n^k) \) is chosen to be the norm of \( P(x_n^k) \) and exact minimization is achieved along each unit vector successively. This technique, therefore, changes a single element of the approximate solution vector at each step. The process, however, possesses the characteristic inherent in most minimization techniques, that of slowness to converge. The development to follow pursues the process of exact minimization but generalizes the basis chosen upon which minimization is to be made from one to \( k, 1 \leq k \leq n \).
II. DEVELOPMENT

A. Preliminaries

Before actually embarking on the process of development it will be necessary to set down a few definitions, notations and lemmas to be used on the following pages.

The problem, briefly stated once more, is to solve the equation \( P(x) = 0 \) where \( P \) maps Euclidean \( n \)-space \( \mathbb{R}^n \) into \( \mathbb{R}^n \). Therefore, the variable \( x \) is an \( n \)-dimensional vector, 
\[ x = (x_1, x_2, \ldots, x_n) \]
and \( x_i \) will denote the \( i \)th component of this vector. Similarly, \( P \) may be decomposed into component functions

\[
P(x) = \begin{pmatrix}
f_1(x) \\
f_2(x) \\
\vdots \\
f_n(x)
\end{pmatrix}
\]

where \( f_i(x) : \mathbb{R}^n \to \mathbb{R}, i = 1, 2, \ldots, n \). On this basis, solving 1.1 may now be written equivalently as solving the system

\[
\begin{align*}
f_1(x) &= 0 \\
f_2(x) &= 0 \\
&\quad \vdots \\
f_n(x) &= 0.
\end{align*}
\]

**Definition 2.1**

An inner product on a linear space is a complex valued
function of two variables selected from the space written
(a, b) and satisfying the conditions:

(i) (a, b) is linear as a function of a for fixed b,
(ii) (b, a) = \overline{(a, b)} (the complex conjugate),
(iii) (a, a) > 0 if a \neq 0.

For the spaces \( \mathbb{R}^n \),

\[
(x, y) = \sum_{i=1}^{n} x_i y_i, \quad (63, p. 120).
\]

A mapping \( \| \cdot \| \) from \( \mathbb{R}^n \) to \( \mathbb{R}^1 \) which satisfies:

(i) \( \| x \| \geq 0, \) for all \( x \in \mathbb{R}^n; \) \( \| x \| = 0 \) only if \( x = 0; \)
(ii) \( \| ax \| = |a| \| x \|, \) for all \( x \in \mathbb{R}^n, a \in \mathbb{R}^1; \)
(iii) \( \| x + y \| \leq \| x \| + \| y \|, \) for all \( x, y \in \mathbb{R}^n; \)

is called a norm. Well-known examples of norms on \( \mathbb{R}^n \) are the \( l_p \)-norms:

\[
\| x \|_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p}, \quad 1 \leq p.
\]

An inner product on \( \mathbb{R}^n \) defines a norm by means of

\[
\| x \| = (x, x)^{\frac{1}{2}},
\]

in particular, the \( l_2 \)-norm, also known as the Euclidean norm, is derived from the inner product given in 2.2 and will be used in the sequel, so that
\[ ||x|| = \left( \sum_{i=1}^{n} x_i^2 \right)^{\frac{1}{2}}. \]  \hspace{2cm} 2.3a

An additional connection between inner product and norm for vectors is

\[ (x,y) = ||x|| \cdot ||y|| \cdot \cos \theta_{x,y} \]  \hspace{2cm} 2.3b

where \( \theta_{x,y} \) is the angle between \( x \) and \( y \), (53, pp. 38-39). With the definition of norm given in 2.3a it is readily recognized that \( || \cdot || \) is a continuous function on \( \mathbb{R}^n \) to \( \mathbb{R}^1 \), (61, p. 75).

The notation to indicate differentiation will be set up in the following fashion

\[ f_{ix_j} = \frac{\partial f_i}{\partial x_j} \]  \hspace{2cm} 2.4

\[ f_{ix_j x_k} = \frac{\partial^2 f_i}{\partial x_j x_k} \]  \hspace{2cm} 2.5

To say that \( P \) is differentiable will mean \( P \) is differentiable in the sense of Frechet.

**Definition 2.2**

A mapping \( F : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m \) is Frechet differentiable at an interior point of \( D \) if there is a linear operator \( A : \mathbb{R}^n \rightarrow \mathbb{R}^m \) such that

\[ \lim_{h \rightarrow 0} \left( \frac{1}{||h||} \right) ||F(x+h) - F(x) - Ah|| = 0. \]  \hspace{2cm} 2.6
The linear operator $A$ is denoted $F'(x)$. The concrete representation for $F'(x)$ is the Jacobian matrix

$$J_{F}(x) = \begin{pmatrix} f_{1x_1}(x) & f_{1x_2}(x) & \cdots & f_{1x_n}(x) \\ f_{2x_1}(x) & \cdots & \cdots & f_{2x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ f_{mx_1}(x) & \cdots & \cdots & f_{mx_n}(x) \end{pmatrix} \tag{2.7}$$

$J_{F}^{i}(x)$ will be used to denote the $i^{th}$ column of $J_{F}(x)$.

**Definition 2.3**

If $F'$ has a Frechet derivative at $x$, then $F''$ is called the second Frechet derivative at $s$ and if $F_{i} : D \subset \mathbb{R}^{n} \to \mathbb{R}^{1}$ and $F''$ exists it will be represented by the Hessian matrix

$$H_{F_{i}}(x) = \begin{pmatrix} f_{ix_{1}x_{1}}(x) & f_{ix_{1}x_{2}}(x) & \cdots & f_{ix_{1}x_{n}}(x) \\ f_{ix_{2}x_{1}}(x) & \cdots & \cdots & f_{ix_{2}x_{n}}(x) \\ \vdots & \vdots & \ddots & \vdots \\ f_{ix_{n}x_{1}}(x) & \cdots & \cdots & f_{ix_{n}x_{n}}(x) \end{pmatrix} \tag{2.8}$$

**Definition 2.4**

Given $x,y$ in $\mathbb{R}^{n}$ the closed segment between $x$ and $y$ is

$$[x,y] = \{ z : z = tx + (1-t)y, \ 0 \leq t \leq 1 \} \tag{53, p. 68}$$

**Lemma 2.1**

$$\left( \sum_{m=1}^{p} a_{m} b_{m} \right)^{2} = \sum_{m=1}^{p} \sum_{s=1}^{p} a_{m} a_{s} b_{m} b_{s}, \ a_{i}, b_{i} \text{ real, } i = 1,2,\ldots,p.$$
Proof The proof is by induction. For \( p = 1 \),

\[
\left( \sum_{m=1}^{p} a_{m} b_{m} \right)^{2} = (a_{1} b_{1})^{2} = a_{1} b_{1} a_{1} b_{1} = \sum_{s=1}^{p} \sum_{m=1}^{p} a_{m} b_{m} a_{s} b_{s}.
\]

Assume

\[
\left( \sum_{m=1}^{t} a_{m} b_{m} \right)^{2} = \sum_{s=1}^{t} \sum_{m=1}^{t} a_{m} b_{m} a_{s} b_{s}.
\]

Examine

\[
\left( \sum_{m=1}^{t+1} a_{m} b_{m} \right)^{2} = \left( \sum_{m=1}^{t} a_{m} b_{m} + a_{t+1} b_{t+1} \right)^{2}
\]

\[
= \left( \sum_{m=1}^{t} a_{m} b_{m} \right)^{2} + 2a_{t+1} b_{t+1} \sum_{m=1}^{t} a_{m} b_{m} + a_{t+1}^{2} b_{t+1}^{2}
\]

\[
= \sum_{m=1}^{t} a_{m} b_{m} \left( \sum_{m=1}^{t} a_{m} b_{m} + a_{t+1} b_{t+1} \right)
\]

\[
= \sum_{s=1}^{t} a_{s} b_{s} a_{t+1} b_{t+1} + (a_{t+1} b_{t+1})^{2}
\]

\[
= \sum_{s=1}^{t} a_{s} b_{s} a_{t+1} b_{t+1} + \sum_{m=1}^{t} a_{m} b_{m} a_{t+1} b_{t+1} +
\]

\[
\sum_{s=1}^{t} a_{s} b_{s} a_{t+1} b_{t+1} + (a_{t+1} b_{t+1})^{2}
\]

\[
= \sum_{s=1}^{t} a_{s} b_{s} a_{t+1} b_{t+1} + \sum_{m=1}^{t} a_{m} b_{m} a_{t+1} b_{t+1} +
\]

\[
\sum_{s=1}^{t} a_{s} b_{s} a_{t+1} b_{t+1} + (a_{t+1} b_{t+1})^{2}
\]

\[
= \sum_{s=1}^{t} a_{s} b_{s} a_{t+1} b_{t+1} + \sum_{m=1}^{t} a_{m} b_{m} a_{t+1} b_{t+1} +
\]

\[
\sum_{s=1}^{t} a_{s} b_{s} a_{t+1} b_{t+1} + (a_{t+1} b_{t+1})^{2}
\]
Lemma 2.2

If $A$, an $n$ by $n$ nonsingular matrix, is symmetric then $A^{-1}$ is symmetric.

Proof

$$(A^{-1})^T = (AA^{-1})^T = I = (A^{-1}A)^T = A(A^{-1})^T$$

and the unique inverse property therefore implies $A^{-1} = (A^{-1})^T$.

Lemma 2.3

A real matrix $A$ is positive definite if and only if there is a nonsingular real matrix $P$ such that $A = P^TP$, (57, p. 94).

Definition 2.5

A submatrix of a square matrix $A$ is called principal if it is obtained from $A$ by deleting certain rows and the like-numbered columns. The determinant of a principal submatrix is called a principal subdeterminant or principal minor, (57, p. 94).

Lemma 2.4

If $A$ is positive definite, every principal submatrix is positive definite. Also the determinant of $A$ and all principal subdeterminants are positive, (57, p. 94).
Lemma 2.5

If A is a real positive definite matrix, then $A^{-1}$ is positive definite.

Proof

Lemma 2.3 allows $A = P^T P$ for some nonsingular real matrix $P$. By Lemma 2.4, $\text{Det}(A) \neq 0$ so $A^{-1}$ exists and $A^{-1} = (P^T P)^{-1} = P^{-1} (P^T)^{-1}$. Let $B = (P^T)^{-1}$. The transformation

$$P^T (P^{-1})^T = (P^{-1} P)^T = I = (P P^{-1})^T = (P^{-1})^T P^T$$

shows $B = (P^T)^{-1} = (P^{-1})^T$ and $B^T = P^{-1}$ so that $A^{-1} = B^T B$, $B$ a nonsingular real matrix. Therefore, by Lemma 2.3, $A^{-1}$ is positive definite.

Definition 2.6

The remainder upon division if $i$ by $k$, $i$ and $k$ integers, is called $i$ modulo $k$ and is denoted by $(i|k)$, (47, p. 49).

Definition 2.7

A real sequence $\{x_n\}$ is called monotonic if either $x_n \leq x_{n+1}$ for each $n$ or $x_n \leq x_{n+1}$ for each $n$, (62, p. 27).

Lemma 2.6

A monotonic sequence is convergent if it is bounded. For a nondecreasing sequence the limit is the least upper bound of the terms; for a nonincreasing sequence the limit is the greatest lower bound of the terms, (62, p. 27).

Lemma 2.7

Suppose that $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^1$ is Frechet differentiable
at each point of a convex set $D_0 \subset D$. Then, for any two points $x, y \in D_0$, there is a $t \in (0,1)$ such that

$$f(y) - f(x) = f'(x + t(y-x)) (y-x), \quad (53, \text{ p. 68}).$$

**Lemma 2.8**

If $S$ is a compact set on which the real-valued function $f$ is continuous when $S$ is regarded as its domain, then $f$ attains absolute minimum and maximum values $m, M$, respectively at certain points of $S$, (62, p. 73).

**Definition 2.8**

A mapping $F : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ is convex on a convex subset $D_0 \subset D$ if

$$F(tx + (1-t)y) \leq tF(x) + (1-t)F(y)$$

for all $x, y \in D_0$ and $t \in (0,1)$, (53, p. 448).

**Lemma 2.9**

Suppose $S$ is a bounded, infinite point set in $\mathbb{R}^k$. Then there is at least one point of accumulation of $S$, (62, p. 58).

**Lemma 2.10**

Let $y$ be an accumulation point of the set $S$ in $\mathbb{R}^k$. Then there exists a sequence $\{x_n\}$ such that each $x_n$ is in $S$ and $x_n \rightarrow y$, (62, p. 59).

Roughly speaking, in the following development a method will be devised in which a sequence of vectors $x^i, i = 0, 1, \ldots$, is found converging to a solution of 1.1. More specifically,
for a fixed $k$, $1 \leq k \leq n$, a new vector $y$ will be derived from $x^m$ by changing $k$ of its components, for example, by changing $x_i^m$, $x_{i+1}^m$, ..., $x_{i+k-1}^m$ for some $i$. The following is then the case

$$x^m = (x_1^m, x_2^m, ..., x_n^m), \quad 2.9$$

and

$$y = (x_1^m, x_2^m, ..., x_i^m + dx_i^m, x_{i+1}^m + dx_{i+1}^m, ..., x_{i+k-1}^m + dx_{i+k-1}^m, x_{i+k}^m, ..., x_n^m). \quad 2.10$$

In order to fit $y$ into the sequencing pattern a second superscript is introduced indicating the left-most element of the vector that has been changed in the computation. With this standardization

$$y = x^m_{i}. \quad 2.11$$

To complete this pattern two problems must be eliminated. First, the first superscript is incremented when $i = n$. That is,

$$x^m_{i}, n = x^{m+1}_{i}, 0 \quad 2.12$$

and when the second superscript is zero it will not be written.

The above notational conveniences allow a definition for
the change vector. In general, the vector being changed and the resultant vector will be clear from the context and therefore it is possible to write

$$dx = x^{p,q+1} - x^{p,q}.$$  \[2.13\]

and

$$dx_i = x^{p,q+1}_i - x^{p,q}_i.$$  \[2.14\]

If an ambiguity is possible the change vector and its components will be indicated with the superscript of the vector being changed, that is

$$dx^{p,q} = x^{p,q+1} - x^{p,q}.$$  \[\text{2.15}\]

The final problem concerns the situation occurring when \(q+k-1 \geq n\), and \(q \leq n\). Of course, when \(x^{p,n}\) has been found the next left-most element to be changed will be the first element. In line with this if \(q+j-1 \geq n\) then find \(dx(q+j-1|n)\), \(j = n-i+2, n-i+3, \ldots, k\), or, in other words, go back to the 'top' of the vector.

The following should help to illustrate the comments just made, the underlining indicating the changed components from the previous vector.

$$x^0 = x^{0,0} = (x_1^0, x_2^0, \ldots, x_n^0).$$
Using these basic elements of theory and notation the next step is then the development.

B. Development

Given \( P \) such that \( P : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n \) and each component function \( f_i \) is such that \( f_i' \) and \( f_i'' \) exist and are continuous for all \( x \) in a convex set \( D_0 \subset D \). Assuming without loss of generality that \( q+k \leq n \), expand \( f_j(x^{p,q+1}) \), \( j = 1, 2, \ldots, n \), in a Taylor's polynomial about the point \( x^{p,q} \) to get the following series of equations
\[ f_j(x^{p,q+1}) = f_j(x^{p,q}) + (dx, J_j(x^{p,q})) + (dx, f''(z), dx), \]
\[ j = 1, 2, \ldots, n; \quad z \in [x^{p,q}, x^{p,q+1}] \quad 2.15 \]

Forming an approximation for \( f_j(x^{p,q+1}) \) by truncating the second order term gives

\[ f_j(x^{p,q+1}) = f_j(x^{p,q}) + (dx, J_j(x^{p,q})), \]
\[ j = 1, 2, \ldots, n. \quad 2.16 \]

Expanding the inner product yields

\[ f_j(x^{p,q+1}) = f_j(x^{p,q}) + \sum_{m=1}^{n} dx_m f_j(x^{p,q}), \]
\[ j = 1, 2, \ldots, n. \quad 2.17 \]

However, \( dx_m = 0 \) for \( m = 1, 2, \ldots, q, q+k+1, q+k+2, \ldots, n \), so

\[ f_j(x^{p,q+1}) = f_j(x^{p,q}) + \sum_{m=q+1}^{q+k} dx_m f_j(x^{p,q}), \]
\[ j = 1, 2, \ldots, n. \quad 2.18 \]

The controlling characteristic is the desire to assure that \( ||P(x^{p,q+1})|| \) is reduced in relation to \( ||P(x^{p,q})|| \). As a matter of fact, it is to be required that the change is to be made in such a way as to maximize the difference between these two successive norms. In order to simplify calculations, the difference between the norms squared will be examined
rather than the difference between the norms. So the problem may be represented by examining

\[ ||P(x_{p,q})||^2 - ||P(x)||^2 \quad 2.19 \]

and finding \( x^{p,q+1} \) which maximizes the difference which is positive.

Using 2.2 in 2.19 gives

\[ \sum_{i=1}^{n} (f_i(x_{p,q}))^2 - \sum_{i=1}^{n} (f_i(x))^2. \quad 2.20 \]

Now introducing 2.18 in 2.20 yields

\[ \sum_{i=1}^{n} (f_i(x_{p,q}))^2 - \sum_{i=1}^{n} (f_i(x_{p,q})) + \]

\[ \sum_{m=q+1}^{q+k} \int_{x_{p,q}} f_{m,i}(x_{p,q})^2 \quad 2.21 \]

Since \( x \) no longer explicitly appears in 2.21 but rather appears in terms of \( dx \) (\( dx = x - x_{p,q} \)), define a function in terms of \( dx \) as

\[ G(dx) = \sum_{i=1}^{n} (f_i(x_{p,q}))^2 - \sum_{i=1}^{n} (f_i(x_{p,q})) + \]

\[ \sum_{m=q+1}^{q+k} \int_{x_{p,q}} f_{m,i}(x_{p,q})^2 \quad 2.22a \]
\[
= \sum_{i=1}^{n} (f_i(x^p,q))^2 - \sum_{i=1}^{n} (f'_i(x^p,q))^2 + \\
2f_i(x^p,q) \sum_{m=q+1}^{q+k} dx_m f_i x_m (x^p,q) + \\
\sum_{m=q+1}^{q+k} \left( \sum_{m=q+1}^{q+k} dx_m f_i x_m (x^p,q) \right)^2 \quad 2.22b
\]
\[
= -2 \sum_{i=1}^{n} (f_i(x^p,q)) \sum_{m=q+1}^{q+k} dx_m f_i x_m (x^p,q) - \\
\sum_{i=1}^{n} \sum_{m=q+1}^{q+k} dx_m f_i x_m (x^p,q) dx_s f_i x_s (x^p,q) \quad 2.22c
\]

Lemma 2.1 reduces 2.22c to
\[
= -2 \sum_{i=1}^{n} (f_i(x^p,q)) \sum_{m=q+1}^{q+k} dx_m f_i x_m (x^p,q) - \\
\sum_{i=1}^{n} \sum_{m=q+1}^{q+k} q+k \sum_{s=q+1}^{q+k} dx_m f_i x_m (x^p,q) dx_s f_i x_s (x^p,q) \quad 2.23
\]

Now, in order to find a candidate for a maximum point of \( G \), differentiate 2.23 with respect to \( dx_r \), \( r = q+1, q+2, \ldots, q+k \), and set the resulting expressions to zero to get the following equations
\[
-2 \sum_{i=1}^{n} f_i(x^p,q)f_i x_r (x^p,q) - \\
2 \sum_{i=1}^{n} \sum_{m=q+1}^{q+k} dx_m f_i x_m (x^p,q)f_i x_r (x^p,q), \\
r = q+1, q+2, \ldots, q+k \quad 2.24
\]
Since all values in the second term are real a rearrangement of the summation and multiplication by 1/2 gives

\[ - \sum_{i=1}^{n} f_i(x_{P,q}) f_{ix_r}(x_{P,q}) = \]

\[ - \sum_{m=q+1}^{q+k} \sum_{i=1}^{n} f_{ix_m}(x_{P,q}) f_{ix_r}(x_{P,q}) = 0, \]

\[ m = q+1, r = q+1, q+2, \ldots, q+k. \quad 2.25 \]

Recognizing from 2.7 that \( f_{ix_m}(x_{P,q}) = (J_P(x_{P,q}))_i \) and recalling that \( f_i(x_{P,q}) = (P(x_{P,q}))_i \), 2.25 becomes

\[- \sum_{i=1}^{n} (P(x_{P,q}))_i (J_P(x_{P,q}))_i \]

\[ - \sum_{m=q+1}^{q+k} \sum_{i=1}^{n} (J_P(x_{P,q}))_i (J_P(x_{P,q}))_i = 0, \]

\[ m = q+1, r = q+1, q+2, \ldots, q+k. \quad 2.26 \]

Simplifying 2.26 by use of 2.2 produces

\[ \sum_{m=q+1}^{q+k} (J_P(x_{P,q}), J_P(x_{P,q})) dx_m = -(P(x_{P,q}), J_P(x_{P,q})), \]

\[ r = q+1, q+2, \ldots, q+k. \quad 2.27 \]

2.27 may be represented by a k x k matrix system with the following definitions

\[ D_{q+1}^{x_k} = (dx_{q+1}, dx_{q+2}, \ldots, dx_{q+k})^T, \quad 2.28a \]
\[ V_{q+1}(x) = \begin{pmatrix} (P(x), J_P^{q+1}(x)) \\ (P(x), J_P^{q+2}(x)) \\ (P(x), J_P^{q+3}(x)) \\ \vdots \\ (P(x), J_P^{q+k}(x)) \end{pmatrix} \]

\[ K^{q+1}_{k}(x) = \begin{pmatrix} (J_P^{q+1}(x), J_P^{q+1}(x)) & (J_P^{q+1}(x), J_P^{q+2}(x)) & \cdots \\ (J_P^{q+2}(x), J_P^{q+1}(x)) & & \cdots \\ \vdots & & \ddots \\ (J_P^{q+k}(x), J_P^{q+1}(x)) & \cdots & (J_P^{q+k}(x), J_P^{q+k}(x)) \end{pmatrix} \]

System 2.27 now becomes

\[ K^{q+1}_{k}(x^{p,q})Dx^{q+1}_{k} = -V_{q+1}(x^{p,q}). \]  

Therefore, assuming \((K^{q+1}_{k}(x^{p,q}))^{-1}\) exists

\[ Dx^{q+1}_{k} = -(K^{q+1}_{k}(x^{p,q}))^{-1}V_{q+1}(x^{p,q}). \]

The contention then is that

\[ x^{p,q+1} = x^{p,q} + dx^{*} \]
with $dx^*$ defined as

$$dx^*_i = \begin{cases} 
0 & 1 \leq i \leq q, \quad q+k < i \leq n \\
(Dx^{q+1})_i & q+1 \leq i \leq q+k 
\end{cases}$$

forms a sequence $\{x^{p,q}\}$, $p = 0, 1, \ldots$, $q = 1, 2, \ldots, n$, that produces a nonincreasing sequence $\|P(x^{p,q})\|^2$. This sequence goes to zero and the original sequence converges to a solution $x^*$, $P(x^*) = 0$. 
III. CONVERGENCE

The methods have now been presented, however, before they are accepted for general use, there are several assumptions and contentions that must be solidified.

**Lemma 3.1**

\[ K_{k}^{q+1}(x^{p,q}) \text{ is positive definite if } J_{p}(x^{p,q}) \text{ is non-singular, } k = 1, 2, \ldots, n. \]

**Proof** From 2.28c, \( K_{k}^{q+1}(x) \) may be written as

\[ K_{k}^{q+1}(x) = (J_{p}^{q+1}(x) \ldots J_{p}^{q+k}(x))^{T} \]

\[ (J_{p}^{q+1}(x) \ldots J_{p}^{q+k}(x)), \]

recalling that \( J_{i}^{1}(x), i = 1, 2, \ldots, n, \) is a column of the Jacobian matrix \( P \) evaluated at \( x \). So, with \( k = n \) and arbitrary \( q, 1 \leq q \leq n, \)

\[ K_{n}^{q+1}(x) = (J_{p}^{(q+1|n)}(x) \ldots J_{p}^{(q+n|n)}(x))^{T} \]

\[ (J_{p}^{(q+1|n)}(x) \ldots J_{p}^{(q+n|n)}(x)), \]

Define a matrix \( R_{q+1}(x) \) in the following fashion

\[ R_{q+1}(x) = (J_{p}^{(q+1|n)}(x) \ldots J_{p}^{(q+n|n)}(x)) \]

\[ 3.3 \]

which allows \( K_{n}^{q+1}(x^{p,q}) = (R_{q+1}(x^{p,q}))^{T}R_{q+1}(x^{p,q}). \]

\[ 3.4 \]

It is easily recognized that \( R_{q+1}(x^{p,q}) \) is simply the
Jacobian matrix $J_p(x^P,q)$ with its columns rearranged which implies that $R_{q+1}(x^P,q)$ is also nonsingular. Therefore, by Lemma 2.3 $K_{q+1}^n(x^P,q)$ is positive definite.

It is also easily seen that $K_{q+1}^n(x^P,q)$ is obtained from $K_{q+1}^n(x^P,q)$ by deleting rows $(q+k+1|n),(q+k+2|n), \ldots, (q+n-1|n)$ and the like-numbered columns. Therefore, by Lemma 2.4, $K_{q+1}^n(x^P,q)$ is positive definite for $k = 1,2,\ldots,n$.

The assumption made relating to the existence of $(K_{q+1}^n(x^P,q))^{-1}$ may now be easily handled by the following lemma.

**Lemma 3.2**

$(K_{q+1}^n(x^P,q))^{-1}$ exists if $(J_p^q(x^P,q))$ is nonsingular, $k = 1,2,\ldots,n$.

**Proof** From Lemma 3.1, $K_{q+1}^n(x^P,q)$, $k = 1,2,\ldots,n$, is a principal submatrix of $K_{q+1}^n(x^P,q)$ which was shown to be positive definite. Lemma 2.4 then asserts that all principal subdeterminants are positive and therefore $K_{q+1}^n(x^P,q)$, $k = 1,2,\ldots,n$, is nonsingular.

**Lemma 3.3**

The function $G(dx)$ as defined by 2.22a is such that $G(dx^*) \geq 0$, $dx^*$ as given in 2.32, if $(J_p(x^P,q))^{-1}$ exists.

**Proof** The second term on the right side of 2.23 is

$$n \sum_{q+k}^q \sum_{q+k}^q dx f_{ix}(x^P,q)dx f_{ix}(x^P,q). \quad 3.5$$

By rearranging the summations this may be written as
By employing 2.2, the definition of inner product, 3.6 may be reduced to

\[ \sum_{s=q+1}^{q+k} \sum_{m=q+1}^{q+k} \int_{x_{s}}^{x_{m}} (J_{P}^{m}(x^{P},q),J_{P}^{S}(x^{P},q))dx_{s} \]  

3.7

The use of 2.28a and 2.28c allow 3.7 to be written as

\[ \sum_{s=q+1}^{q+k} (Dx_{s}+1)^{T}K_{k}^{q+1}(x^{P},q)dx_{s} \]  

3.8

and one further usage of 2.28a gives the form

\[ -(Dx_{k}^q)^{T}K_{k}^{q+1}(x^{P},q)Dx_{k}^{q+1} \]  

3.9

The first term on the right side of 2.23, after interchanging summations, may be written as

\[ -\sum_{m=q+1}^{q+k} \int_{m}^{n} \sum_{i=1}^{n} f_{i}(x^{P},q)f_{i}^{T}(x^{P},q)dx_{m} \]  

3.10

Once again utilizing 2.2, 3.10 is transformed into

\[ -\sum_{m=q+1}^{q+k} \int_{m}^{n} (P(x^{P},q),J_{P}^{q+1}(x^{P},q)) \]  

3.11

which further reduces to 3.12 by the use of 2.28a and 2.28b,
By using 3.9 and 3.12, \( G(dx^*) \) may now be written as

\[
G(dx^*) = -2(Dx_k^{q+1} )^T v_{q+1}(x^p, q) - \\
\left( Dx_k^{q+1} \right)^T k_k^{q+1}(x^p, q) Dk_k^{q+1} \]

3.13a

\[
= (Dx_k^{q+1} )^T ( -k_k^{q+1}(x^p, q) Dx_k^{q+1} - 2v_{q+1}(p, q)) . \]

3.13b

Inserting the value of \( Dx_k^{q+1} \) as given by 2.30 within the parentheses in 3.13b gives

\[
= (Dx_k^{q+1} )^T ( -k_k^{q+1}(x^p, q) Dx_k^{q+1})^{-1} v_{q+1}(x^p, q) - 2v_{q+1}(x^p, q) \]

3.13c

\[
= (Dx_k^{q+1} )^T ( v_{q+1}(x^p, q) - 2v_{q+1}(x^p, q) ) \]

3.13d

\[
= -(Dx_k^{q+1} )^T v_{q+1}(x^p, q). \]

3.13e

Once again using 2.30 in the first term this becomes

\[
= -(k_k^{q+1}(x^p, q))^{-1} v_{q+1}(x^p, q))^T v_{q+1}(x^p, q) \]

3.13f

\[
= (v_{q+1}(x^p, q))^T ( (k_k^{q+1}(x^p, q))^T )^{-1} v_{q+1}(x^p, q) . \]

3.14

Now, since

\[
(k_k^{q+1}(x))_{i,j} = (J_k^{q+1+i} (x), J_k^{q+1+j} (x)) \]

3.15
\[ n \sum_{m=1}^{n} f_{m}^{x_{q+1}^i} f_{m}^{x_{q+1}^j}(x) \]

of which all terms are real so this may be written as

\[ = \sum_{m=1}^{n} f_{m}^{x_{q+1}^i} f_{m}^{x_{q+1}^j}(x) \]

so that \( K^{q+1}(x) \) is symmetric. Using this fact, 3.14 becomes

\[ G(dx^*) = (V_{q+1}(x^{P,q}))^T(K^{q+1}(x^{P,q}))^{-1}V_{q+1}(x^{P,q}). \]

Lemma 3.1 asserted that \( K^{q+1}(x^{P,q}) \) is positive definite and therefore by Lemma 2.5 \( (K^{q+1}(x^{P,q}))^{-1} \) is also positive definite so that 3.16 now readily shows \( G(dx^*) > 0 \).

**Lemma 3.4**

The function \( G(dx) \) as defined by 2.22a has a maximum at \( dx^* \), with \( dx^* \) as defined in 2.32, if \( (J_{P}(x^{P,q})) \) is non-singular.

**Proof**

\[ \frac{\partial G}{\partial dx_r} = \sum_{m=q+1}^{q+k} (J_{P}^{m}(x^{P,q}),J_{P}^{r}(x^{P,q}))dx_m + (P(x^{P,q}),J_{P}^{r}(x^{P,q})) \]

\[ r = q+1, q+2, \ldots, q+k \]

from 2.27.
Differentiating again with respect to $dx^x$,

$$\frac{\partial^2 G}{\partial (dx^x)^2} = (J_p^r(x^{p,q}), J_p^r(x^{p,q})) \geq 0, r = q+1, q+2, \ldots, q+k$$

by part (iii) of Definition 2.1 which implies the quadratic function $G$ has a maximum at $dx^*$. Therefore, the desired result has been achieved, that is, forming $x^{p,q+1} = x^{p,q} + dx^*$, $dx^*$ as given by 2.32, assures that $||P(x^{p,q+1})||^2 \leq ||P(x^{p,q})||^2$. In addition, the difference between the norms squared is a maximum at this step. Therefore, generating the sequence

$$\Omega = (\{x^{p,q}\}_{q=1,2,\ldots,n})_{p=0}^\infty$$

by the process above results in the fact that the sequence

$$\phi = (\{||P(x^{p,q})||^2\}_{q=1,2,\ldots,n})_{p=0}^\infty$$

is a nonincreasing sequence bounded below by zero. Lemma 2.6 then guarantees that the sequence $\phi$ converges.

In order to simplify further considerations a renumbering of the sequence $\phi$ is in order. This may be carried out by defining

$$x^{np+q} = x^{p,q}, \quad q = 1, 2, \ldots, n; \quad p = 0, 1, \ldots, 3.17$$
Theorem 3.1

Given $P : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that each component function $f_i$, $i = 1, 2, \ldots, n$, is continuously differentiable on a closed and bounded convex subset $D_0 \subseteq D$ in which a solution exists. Then, if $P'(x)$ is nonsingular on $D_0$ and $|P(x)|$ is strictly convex, $\{x^i\}_{i=0}^\infty$ converges to an $x^*$ such that $P(x^*) = 0$.

Proof

Assume to the contrary that the sequence $\{x^i\}_{i=0}^\infty$ does not converge, that is, there is a subsequence $\{x^i\}_{i=0}^\infty$ such that $||x^{i+1} - x^i|| \geq \delta > 0$, for some $\delta$.

Let $x^{i+1} = x^{i+1}$ and $x^i = x^i$, $i = 0, 1, 2, \ldots$. Since $\{x^{i+1}\}_{i=0}^\infty$ and $\{x^i\}_{i=0}^\infty$ are contained in $D_0$ they are bounded and therefore have accumulation points $y$ and $z$, respectively, by Lemma 2.9 and since $D_0$ is closed, $y$ and $z$ are in $D_0$. Consequently, there are subsequences of these sequences converging, respectively, to $y$ and $z$ by Lemma 2.10. Rather than redefine subsequences assume that

$$\lim_{s \to \infty} x^{i+1} = y, \quad \text{and} \quad \lim_{s \to \infty} x^i = z.$$ 

Therefore,

$$\lim_{s \to \infty} ||x^{i+1} - x^i|| = ||\lim_{s \to \infty} (x^{i+1} - x^i)||$$
\[ \lim_{s \to \infty} x^{s+1} = \lim_{s \to \infty} x^s \]

implying \( \|y - z\| \geq \delta > 0 \).

Since the sequence \( \psi = \{\|P(x^i)\|^2\}_{i=0}^\infty \) converges, say, to a value \( c \), it is also true that \( \{\|P(x^s)\|^2\}_{s=0}^\infty \) and

\[ \{\|P(x^{s+1})\|^2\}_{s=0}^\infty \]

also converge to \( c \), therefore,

\[ \lim_{s \to \infty} (\|P(x^{s+1})\|^2 - \|P(x^s)\|^2) = 0 \]

and therefore \( \|P(y)\| = \|P(z)\| \).

Due to the process defined in Chapter II,

\[ \|P(x^k)\| \geq \|P(x^{k+1})\| \]

and since maximality was achieved

\[ \|P(x^k)\| \geq \|P(tx^k + (1-t)x^{k+1})\| \geq \|P(x^{k+1})\|, \]

for all \( t \), \( 0 < t < 1 \)

and the convexity of \( D_0 \) assures that \( tx^k + (1-t)x^{k+1} \) remains in \( D_0 \). So, since \( y \) and \( z \) are in \( D_0 \), it is true that \( ty + (1-t)z \in D_0 \) for all \( t \in (0,1) \). Applying 3.20 to the subsequence defined above and taking limits yields

\[ \|P(y)\| = \|P(ty + (1-t)z)\| = \|P(z)\| = a, \]

for all \( t \), \( 0 < t < 1 \)

Define \( \|P(y)\| = \|P(z)\| = a \) and examine
\[ |f_i(ty + (1-t)z)| < t|f_i(y)| + (1-t)|f_i(z)|, \]
\[ i = 1, 2, \ldots, n, \text{ for all } t \in (0,1) \] 3.22

implies
\[ ||P(ty + (1-t)z)|| < ||tP(y)| + (1-t)|P(z)|| ||, \]
\[ \text{for all } t \in (0,1) \]
\[ \leq ||tP(y)|| + ||(1-t)|P(z)|| ||, \]
\[ \text{for all } t \in (0,1) \]
\[ = t||P(y)|| + (1-t)||P(z)||, \]
\[ \text{for all } t \in (0,1) \]

so that the expression 3.20 yields
\[ a < ta + (1-t)a \] 3.23

which states that \( a < a \), which is a contradiction.

Therefore, the sequence \( \{x^i\}_{i=0}^{\infty} \) converges to some \( x^* \) and \( D_0 \) closed implies \( x^* \in D_0 \).

With this result it is then true that
\[ \lim_{i \to \infty} (x^{i+1} - x^i) = \lim_{i \to \infty} dx^i = 0. \] 3.24

Also, since each element of the K-matrices as defined by 2.28c is a continuous real-valued function, each is bounded in absolute value on \( D_0 \) by, say, \( M_{jm}, j = 1, 2, \ldots, n; \)
\[ m = 1, 2, \ldots, n, \text{ by Lemma 2.8. Therefore,} \]
\[- \sum_{m=1}^{n} M_{jm} \, dx_m^i \leq (P(x^i), J_P^j(x^i)) \leq \sum_{m=1}^{n} M_{jm} \, dx_m^i,\]

\[j = 1, 2, \ldots, n, \text{ for all } i. \quad 3.25\]

So that

\[\lim_{i \to \infty} (P(x^i), J_P^j(x^i)) = 0, \quad j = 1, 2, \ldots, n, \quad 3.26\]

and through continuity

\[(P(x^*), J_P^j(x^*)) = 0, \quad j = 1, 2, \ldots, n.\]

This is certainly true if \(||P(x^*)|| = 0\). Assume to the contrary that \(||P(x^*)|| \neq 0\). Then \(J_P^j(x^*), j = 1, 2, \ldots, n\), the column vectors of the Jacobian evaluated at \(x^*\) are orthogonal to the nonzero vector \(P(x^*)\) implying these columns are linearly dependent and therefore, \(P'(x^*)\) is singular which is a contradiction. Therefore, \(||P(x^*)|| = 0, \text{ and } x^*\) is the desired solution.
IV. GEOMETRICAL INTERPRETATION

In $\mathbb{R}^n$, assume an arbitrary set of $k$ linearly independent vectors, $u_1, u_2, \ldots, u_k$, is given. Let $S$ be the subspace spanned by these vectors so that an arbitrary element $u$ in $S$ may be represented as

$$u = \sum_{i=1}^{k} t_i u_i$$

for some choice of the scalars $t_i$, $i = 1, 2, \ldots, k$.

Now, the problem posed is that of finding, for an arbitrary $x \in \mathbb{R}^n$, $x \not\in S$, the element in $S$ 'closest' to $x$. The fact that such an element exists is handled by the following theorem.

Theorem 4.1

For any given $x \in \mathbb{R}^n$ there exists a $u \in S$ such that

$$||u - x|| \leq ||v - x||$$

for all $v \in S$, (50, p. 1).

In order to establish uniqueness the following definition is needed.

Definition 4.1

A space $V$ is called strictly convex if

$$||x + y|| < 2$$

whenever $||x|| = ||y|| = 1$ and $x \neq y$, (50, p. 2).

It is easily shown that $\mathbb{R}^n$ is strictly convex with the norm introduced in 2.3a by examining
\[ \|x + y\|^2 + \|x - y\|^2 = (x + y, x + y) + \\
(x - y, x - y) \quad 4.3 \]
\[ = (x, x + y) + (y, x + y) + \\
(x, x - y) - (y, x - y) \]
\[ = (x, x) + (x, y) + (y, x) + (y, y) + \\
(x, x) - (x, y) - (y, x) + (y, y) \]
\[ = 2(x, x) + 2(y, y) \]
\[ = 2\|x\|^2 + 2\|y\|^2. \]

Using 4.3 with \(x \neq y\) and \(\|x\| = \|y\| = 1\) then
\[ \|x + y\|^2 + \|x - y\|^2 = 4 \quad 4.4 \]

implying
\[ \|x + y\|^2 = 4 - \|x - y\|^2 \]

However, \(\|x - y\|^2 \neq 0\) so that \(\|x + y\|^2 < 4\) implying
\[ \|x + y\| < 2. \]

With this knowledge, the uniqueness of the best approximation comes from the following theorem.

**Theorem 4.2**

If \(V\) is strictly convex, then for each \(x \in V\) and each finite dimensional subspace \(S \subset V\) there exists precisely one best approximation of \(x\) with respect to \(S\), (50, p. 2).

The next problem is to actually find the best approximation for an arbitrary \(x\) or at least determine a representation of the best approximation. To do this it is necessary
to minimize $||x - \sum_{i=1}^{k} t_i u_i||^2$ with respect to $t_i$, $i = 1, 2, \ldots, k$.

$$||x - \sum_{i=1}^{k} t_i u_i||^2 = (x - \sum_{i=1}^{k} t_i u_i, x - \sum_{i=1}^{k} t_i u_i) = (x, x - \sum_{i=1}^{k} t_i u_i) - \sum_{i=1}^{k} t_i u_i, x + \sum_{i=1}^{k} t_i u_i, x - \sum_{i=1}^{k} t_i u_i, x + \sum_{i=1}^{k} t_i u_i, x$$

$$= (x, x) - 2 \sum_{i=1}^{k} t_i (x, u_i) + \sum_{j=1}^{k} \sum_{i=1}^{k} t_i t_j (u_j, u_i). \quad 4.6$$

Differentiating 4.6 with respect to $t_i$, $i = 1, 2, \ldots, k$, respectively and setting to zero yields

$$-2(x, u_i) + 2 \sum_{j=1}^{k} t_j (u_j, u_i) = 0, \quad i = 1, 2, \ldots, k. \quad 4.7$$

The vector $T = (t_1, t_2, \ldots, t_k)$ may be arrived at by solving the following matrix equation derived from 4.7.
From 4.8 it is easily seen that if the vector on the right hand side were changed to zero, the change required in $T$ to arrive at the solution of the homogeneous system would be $-T$, since the solution is the zero vector. Solving directly for $-T$ introduces a negative sign on the right side of 4.8.

Bearing this in mind and making the following assignments

$$P(x^{p,q}) = x, \quad j_{p}^{q+1}(x^{p,q}) = u_{j}, \quad j = 1, 2, \ldots, k,$$

the changed form of 4.8 becomes precisely 2.29.

To paraphrase this, $P(x^{p,q})$ is projected, hence the name 'projection method', onto the subspace of order $k$ formed from columns $q+1$ through $q+1+k$ of the Jacobian matrix of $P$ evaluated at $x^{p,q}$. The determination is then made as to the change necessary to reduce this projection to zero and the step is completed.
V. ALGORITHM

The actual implementation of the process described in Chapter II is basically simple. After deciding upon $k$, the dimension of the projection space, begin the process below at step 1.

1. Choose $x^0 = x^0, 0$, the initial vector.
2. Set $q = 0$ and $p = 0$.
3. Evaluate $J^i_p(x^{p,q})$ for $i = (q+1|n), (q+2|n), \ldots, (q+k|n)$.
4. Find $K_{k}^{q+1}(x^{p,q})$ by

$$K_{k}^{q+1}(x^{p,q})_{i,j} = (J^i_p(x^{p,q}), J^j_p(x^{p,q}))$$

$$i = (q+1|n), (q+2|n), \ldots, (q+k|n);$$

$$j = (q+1|n), (q+2|n), \ldots, (q+k|n).$$

5. Evaluate $P(x^{p,q})$.
6. Find $V_{q+1}(x^{p,q})$ such that

$$(V_{q+1}(x^{p,q})_i = (P(x^{p,q}), J^i_p(x^{p,q})),$$

$$i = (q+1|n), (q+2|n), \ldots, (q+k|n).$$

7. Solve

$$K_{k}^{q+1}(x^{p,q})Dx_{k}^{q+1} = -V_{q+1}(x^{p,q}) \text{ or}$$

$$Dx_{k}^{q+1} = -(K_{k}^{q+1}(x^{p,q}))^{-1}V_{q+1}(x^{p,q}).$$
8. Form

\[ dx = \begin{cases} 
0 & 1 \leq i \leq q, \ q+k < i \leq n, \\
(Dx^{q+1})_i & q+1 \leq i \leq q+k.
\end{cases} \]

9. \( x^{P,q+1} = x^{P,q} + dx \).

10. Test \( x^{P,q+1} \) to see if \( ||P(x^{P,q+1})|| \) is within desired tolerance. If so, the process is stopped. If not, continue with step 11.

11. \( q = q + 1 \). If \( q \) is now greater than \( n \) then set \( q = 1 \) and \( p = p + 1 \). Return to step 3 and continue.

The processes available for performing step 7 are many. In the examples presented in Chapter VI, the first approach was used and a partial-pivoting Gaussian elimination method was employed.
VI. EXAMPLES AND COMPARISONS

In this chapter we consider some interesting examples. In the following examples the method developed in Chapter II is employed with various values for $k$, the dimension of the projection space. In addition, Newton's method is applied where possible for comparison. In most cases computation (CPU time) in seconds is given.

Problem 6.1

This first example is a simple case generated to illustrate the desirability of having a choice available for the value $k$. The system generated was

\[
\begin{align*}
  x_2 x_3 - 1.0 &= 0 \\
  x_1 x_3 - 1.0 &= 0 \\
  x_1 x_2 - 1.0 &= 0
\end{align*}
\]

so that the Jacobian of the system could be represented as

\[
J_p(x) = \begin{pmatrix}
0 & x_3 & x_2 \\
\end{pmatrix} \begin{pmatrix}
x_3 & 0 & x_1 \\
x_2 & x_1 & 0
\end{pmatrix}
\]

The system has an obvious solution at $(1,1,1)$. However, choosing the initial point respectively as $x^0 = (0,0,0,6)$, $x^0 = (0,0,6,0)$, $x^0 = (0,6,0,0)$ makes $\det(J_p(x^0)) = 0$ so that immediately Newton's method is out of consideration. However,
choosing \( k = 2 \) it is seen that
\[
\begin{pmatrix} 0.36 & 0 \\ 0 & 0.36 \end{pmatrix}^{6.3}
\]
so that \( \det(K_2^i(x^0)) \neq 0, i = 1, 2, 3 \), for each \( x^0 \). Therefore, a \( dx^0 \) may be found. Indeed, the following results are obtained.

In all cases \( ||P(x^0)|| = 3 \).

Table 1. Convergence for Problem 6.1

<table>
<thead>
<tr>
<th>Iteration</th>
<th>((0,0,0.6),(0.6,0,0))</th>
<th>((0,0,6,0))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.777777777777777777</td>
<td>1.414213562373094900</td>
</tr>
<tr>
<td>2</td>
<td>0.319186719609867920</td>
<td>0.060216794359598582</td>
</tr>
<tr>
<td>3</td>
<td>0.035304985734810023</td>
<td>0.765488218757043000</td>
</tr>
<tr>
<td>4</td>
<td>0.003333656655319159</td>
<td>0.096839021564569180</td>
</tr>
<tr>
<td>5</td>
<td>0.0000001136518585480</td>
<td>0.004925192706578504</td>
</tr>
<tr>
<td>6</td>
<td>0.0</td>
<td>0.0000007955100091876</td>
</tr>
<tr>
<td>7</td>
<td>.......</td>
<td>0.000000000010463134</td>
</tr>
<tr>
<td>8</td>
<td>.......</td>
<td>0.000000000000000039</td>
</tr>
</tbody>
</table>

After 6 iterations for \( x^0 = (0,0,0.6) \) and \( x^0 = (0.6,0,0) \) the solution (to 18 calculated decimal places) is \( (1.0,1.0,1.0) \). The result for \( x^0 = (0,0.6,0) \) differs only in the third term by \( 2 \times 10^{-16} \).

This specifically points out the fact that possibly a normal sequential pattern can be continued in any usage of
the method if a column or columns of the Jacobian which are causing singularity can be isolated and circumvented for the next step.

**Problem 6.2**

Before proceeding to more practical applications the following contrived example was extracted from an article by Mancino (48, p.348).

\[20x_1 - \cos^2 x_2 + x_3 - \sin x_3 = 37\]
\[
\cos 2x_1 + 20x_2 + \log_{10}(1 + x_4^2) = -5
\]
\[
\sin(x_1 + x_2) - x_2 + 19x_3 + \arctan x_3 = 12
\]
\[
(-2x_2^2 + 0.5) + 21x_4 = 0
\]

The initial vector was \(x^0 = (1,1,1,1)\) resulting in \(||P(x^0)|| = 38.592927671717211000.\)

**Table 2. k = 1**

| Change Cycle Position | \(||P(x)||\) | Approximate Change | Per Cent Change |
|-----------------------|-------------|--------------------|-----------------|
| 1                     | 34.29498095504766500 | 4.2980             | 10.83           |
| 2                     | 22.69190916809457000 | 11.6030            | 33.83           |
| 3                     | 21.4431138320792000  | 1.2488             | 5.50            |
| 4                     | 1.76956190848165980  | 20.6736            | 96.41           |
| 1                     | 1.02518811899402610  | 0.74438            | 42.07           |
| 2                     | 0.92147568331129960  | 0.10371            | 10.12           |
| 3                     | 0.38564237723552512   | 0.63583            | 69.00           |
Table 2. (Continued)

| Change Cycle Position | $||P(x)||$ | Approximate Change | Per Cent Change |
|-----------------------|-----------|-------------------|-----------------|
| 4                     | 0.047593189042713141 | 0.33805 | 87.66 |
| 3                     | 0.041456285812162277 | 0.006137 | 12.89 |
| 2                     | 0.037990452580788012 | 0.003466 | 8.36 |
| 3                     | 0.003290396371959145 | 0.034700 | 91.34 |
| 4                     | 0.002423061482343607 | 0.000867 | 26.36 |
| 4                     | 1.002312265269975593 | 0.108x10^{-4} | 4.57 |
| 2                     | 0.000196701930407049 | 2.211x10^{-4} | 5.29 |
| 3                     | 0.000102810219735697 | 9.389x10^{-5} | 47.73 |
| 5                     | 0.000045877805396979 | 5.693x10^{-5} | 55.38 |
| 2                     | 0.000014930289320521 | 3.095x10^{-5} | 67.46 |
| 3                     | 0.000005626262426273 | 9.304x10^{-6} | 62.32 |
| 4                     | 0.000001839585845972 | 3.787x10^{-6} | 67.31 |
| 6                     | 0.0000000640961485063 | 1.198x10^{-6} | 65.16 |
| 2                     | 0.00000051675148409 | 8.919x10^{-8} | 13.92 |
| 3                     | 0.000000045390437265 | 5.063x10^{-7} | 91.77 |
| 4                     | 0.000000036909556111 | 8.490x10^{-9} | 18.70 |
| 12                    | 0.0000000000000003560 | ... | ... |

Final results for the case $k = 1$ to the accuracy available are presented in Table 3.
Table 3. Final results for $k = 1$

<table>
<thead>
<tr>
<th>Iterations</th>
<th>CPU Time</th>
<th>$x_{\text{final}}$</th>
<th>$P(x_{\text{final}})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>47</td>
<td>3.7</td>
<td>1.896513982697355700</td>
<td>3.553x10^{-15}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.210251668176657050</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.542087326060926580</td>
<td>2.220x10^{-16}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.023885953035348262</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 4. $k = 2$

| Change | Cycle Position | $||P(x)||$ | Approximate Change | Per Cent Change |
|--------|----------------|-----------|-------------------|-----------------|
| 1      | 1,2            | 22.74259491392678500 | 15.8504 | 41.07          |
|        | 2,3            | 21.326809798985982000 | 1.4157  | 6.22           |
|        | 3,4            | 2.370568514376673800 | 18.9563 | 88.88          |
|        | 4,1            | 2.297908197583011300 | 0.0726  | 3.06           |
| 2      | 1,2            | 0.231132228935475340 | 0.06677 | 2.90           |
|        | 2,3            | 0.208439918118389360 | 0.02260 | 9.78           |
|        | 3,4            | 0.022145208855534757 | 0.18629 | 89.38          |
|        | 4,1            | 0.018884430321294055 | 0.00326 | 14.72          |
| 3      | 1,2            | 0.002010392560066615 | 0.0168741 | 89.35       |
|        | 2,3            | 0.001586803373813616 | 0.0004235 | 21.07       |
|        | 3,4            | 0.0014054155364365 | 0.0014463 | 91.14       |
|        | 4,1            | 0.000140423928798925 | 0.0000001 | 0.07        |
| 4      | 1,2            | 0.000014899419065687 | 1.255x10^{-4} | 89.39     |
|        | 2,3            | 0.000011809080477263 | 3.090x10^{-6} | 20.74     |
|        | 3,4            | 0.000001045392030299 | 1.076x10^{-5} | 91.15     |
|        | 4,1            | 0.0000010447666360919 | 1.000x10^{-9} | 0.10      |
Table 4. (Continued)

<table>
<thead>
<tr>
<th>Cycle Position</th>
<th>||P(x)||</th>
<th>Approximate Change</th>
<th>Per Cent Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1,2</td>
<td>0.0000000110848745622</td>
<td>9.339x10^{-7}</td>
</tr>
<tr>
<td></td>
<td>2,3</td>
<td>0.000000087861415895</td>
<td>2.298x10^{-8}</td>
</tr>
<tr>
<td></td>
<td>3,4</td>
<td>0.00000007777852144</td>
<td>8.009x10^{-8}</td>
</tr>
<tr>
<td></td>
<td>4,1</td>
<td>0.00000007773210266</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.0000000824730151</td>
<td>6.948x10^{-9}</td>
</tr>
<tr>
<td></td>
<td>2,3</td>
<td>0.0000000653701451</td>
<td>1.710x10^{-10}</td>
</tr>
<tr>
<td></td>
<td>3,4</td>
<td>0.000000057868378</td>
<td>5.958x10^{-10}</td>
</tr>
<tr>
<td></td>
<td>4,1</td>
<td>0.000000057833951</td>
<td>3.000x10^{-14}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1,2</td>
<td>0.000000000000003560</td>
<td>.....</td>
</tr>
</tbody>
</table>

For k = 2, CPU time was 2.7 seconds and there were 37 iterations leading to the same solution as for k = 1.

Table 5. k = 3

<table>
<thead>
<tr>
<th>Cycle Position</th>
<th>||P(x)||</th>
<th>Approximate Change</th>
<th>Per Cent Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,2,3</td>
<td>21.450783805236491000</td>
<td>17.1422</td>
</tr>
<tr>
<td></td>
<td>2,3,4</td>
<td>2.122693170537455400</td>
<td>19.3281</td>
</tr>
<tr>
<td></td>
<td>3,4,1</td>
<td>2.056399443251645100</td>
<td>0.0663</td>
</tr>
<tr>
<td></td>
<td>4,1,2</td>
<td>0.121355116123677620</td>
<td>1.9350</td>
</tr>
<tr>
<td>2</td>
<td>1,2,3</td>
<td>0.014181262919804938</td>
<td>0.10717</td>
</tr>
<tr>
<td></td>
<td>2,3,4</td>
<td>0.000083657504891004</td>
<td>0.01409</td>
</tr>
<tr>
<td></td>
<td>3,4,1</td>
<td>0.000061175165666518</td>
<td>0.00002248</td>
</tr>
</tbody>
</table>
Table 5. (Continued)

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Change Position</th>
<th>|P(x)|</th>
<th>Approximate Change</th>
<th>Per Cent Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,1,2</td>
<td>0.0000003452365244130</td>
<td>0.000005772</td>
<td>94.36</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1,2,3</td>
<td>0.000000324676367283</td>
<td>3.128x10^{-6}</td>
<td>90.60</td>
</tr>
<tr>
<td></td>
<td>2,3,4</td>
<td>0.00000001794621762</td>
<td>3.229x10^{-7}</td>
<td>99.45</td>
</tr>
<tr>
<td></td>
<td>3,4,1</td>
<td>0.00000001435538333</td>
<td>3.591x10^{-10}</td>
<td>20.01</td>
</tr>
<tr>
<td></td>
<td>4,1,2</td>
<td>0.000000081013052</td>
<td>1.354x10^{-9}</td>
<td>94.36</td>
</tr>
<tr>
<td>4</td>
<td>1,2,3</td>
<td>0.00000000000007618954</td>
<td>7.339x10^{-11}</td>
<td>90.60</td>
</tr>
<tr>
<td></td>
<td>2,3,4</td>
<td>0.000000000000040685</td>
<td>7.578x10^{-12}</td>
<td>99.48</td>
</tr>
<tr>
<td></td>
<td>3,4,1</td>
<td>0.000000000000034158</td>
<td>6.527x10^{-15}</td>
<td>16.04</td>
</tr>
<tr>
<td></td>
<td>4,1,2</td>
<td>0.00000000000007328</td>
<td>2.683x10^{-14}</td>
<td>78.55</td>
</tr>
<tr>
<td>5</td>
<td>1,2,3</td>
<td>0.000000000000003567</td>
<td>4.761x10^{-15}</td>
<td>64.97</td>
</tr>
<tr>
<td></td>
<td>2,3,4</td>
<td>0.000000000000003560</td>
<td>7.000x10^{-18}</td>
<td>0.20</td>
</tr>
</tbody>
</table>

For k = 3, CPU time was 2.7 seconds and there were 18 iterations. The same solution as before was found.

Table 6. k = 4

<table>
<thead>
<tr>
<th>Cycle</th>
<th>|P(x)|</th>
<th>Approximate Change</th>
<th>Per Cent Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.492941552747714700</td>
<td>35.09998</td>
<td>90.95</td>
</tr>
<tr>
<td>2</td>
<td>0.035003848144719339</td>
<td>3.45795</td>
<td>99.0</td>
</tr>
<tr>
<td>3</td>
<td>0.000010123830252007</td>
<td>0.03498</td>
<td>99.94</td>
</tr>
<tr>
<td>4</td>
<td>0.000000004179305355</td>
<td>1.0119x10^{-5}</td>
<td>99.96</td>
</tr>
<tr>
<td>5</td>
<td>0.0000000001596945</td>
<td>4.1777x10^{-9}</td>
<td>99.96</td>
</tr>
<tr>
<td>6</td>
<td>0.00000000000003669</td>
<td>1.5933x10^{-12}</td>
<td>99.77</td>
</tr>
<tr>
<td>7</td>
<td>0.00000000000003560</td>
<td>1.0900x10^{-16}</td>
<td>2.97</td>
</tr>
</tbody>
</table>
For \( k = 4 \), CPU time was 1.3 seconds and 7 iterations were performed again yielding the same solution.

Newton's method, by comparison, took 2.1 seconds and 8 iterations to arrive at the same solution to 16 decimal positions.

In addition to the normal pattern, for \( k = 2 \) and \( k = 3 \) a different pattern was used to eliminate overlap and observe any possible effect. The results are presented in Table 7 and should be compared with the results in Table 4 and Table 5 respectively.

Table 7. \( k = 2 \) and \( k = 3 \) with varied change sequence

<table>
<thead>
<tr>
<th>( k )</th>
<th>Total Iterations</th>
<th>CPU Time</th>
<th>( | P(x^{\text{final}}) | )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>20</td>
<td>1.6</td>
<td>0.000000000000003560</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>1.5</td>
<td>0.000000000000003560</td>
</tr>
</tbody>
</table>

Of course, varying the order for \( k = 4 \) produces no change as all components are changed regardless.

From Table 7, for \( k = 2 \), the order of change proceeded 1 and 2, then 3 and 4, 1 and 2, then 3 and 4, etc., and the results indicate the number of iterations and time are more than halved. For \( k = 3 \), the order of change was 1, 2 and 3, then 4, 1 and 2, then 3, 4 and 1, then 2, 3 and 4, 1, 2 and 3 etc. Once again the number of iterations and time were reduced.
Denoting the standard sequential type change as type A and that presented in Table 7 as type B, Table 8 offers a cross-sectional comparison of the various methods as k varies.

### Table 8. Cross-sectional analysis

<table>
<thead>
<tr>
<th>Step</th>
<th>After</th>
<th>Norm of k-type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2-A</td>
</tr>
<tr>
<td>1</td>
<td>34.2949</td>
<td>22.7426</td>
</tr>
<tr>
<td>5</td>
<td>1.0252</td>
<td>2x10^-1</td>
</tr>
<tr>
<td>10</td>
<td>4x10^-2</td>
<td>2x10^-3</td>
</tr>
<tr>
<td>15</td>
<td>2x10^-4</td>
<td>1x10^-6</td>
</tr>
<tr>
<td>20</td>
<td>2x10^-6</td>
<td>8x10^-9</td>
</tr>
<tr>
<td>25</td>
<td>3x10^-8</td>
<td>6x10^-12</td>
</tr>
<tr>
<td>30</td>
<td>2x10^-10</td>
<td>4x10^-14</td>
</tr>
<tr>
<td>35</td>
<td>6x10^-13</td>
<td>7x10^-15</td>
</tr>
<tr>
<td>40</td>
<td>3x10^-14</td>
<td>...</td>
</tr>
<tr>
<td>45</td>
<td>4x10^-15</td>
<td>...</td>
</tr>
</tbody>
</table>

Viewing the various results presented might conceivably suggest future work in the area. First, the choice of k, second, given k how are the changes to be made. Checking the percentage changes given in Tables 2, 4 and 5 it is obvious that certain changes are much more efficient than others so that possibly an a priori determination to maximize effort might be possible.

### Problem 6.3

Consider next the problem presented by Willers (64, p. 177) introduced in Chapter I. Specifically, an elastic steel wire
is stretched between two rigid supports \( d = 2w = 100 \) meters apart. The wire has a sag of \( h = 2.5 \) meters. Determine its length \( 2L \) and the horizontal tension \( H \) given the elasticity constants \( E = 19,000 \text{ kg/mm}^2, \ g = 0.0078 \text{ kg/meter length}. \) With the more general variable \( a = H/g \) the equations to solve are

\[
\begin{align*}
w &= a \ln \left( \frac{L}{a} + \left(1 + \frac{L^2}{a^2}\right)^{\frac{1}{2}} \right) + agL/E, \\
h &= a(\left(1 + \frac{L^2}{a^2}\right)^{\frac{1}{2}} - 1) + \frac{gL^2}{2E}.
\end{align*}
\]

Inserting the known constants and \( x_1 = a, \ x_2 = L \) this becomes

\[
\begin{align*}
x_1 \ln \left(\frac{x_2}{x_1} + \sqrt{1 + (\frac{x_2}{x_1})^2}\right) + \\
0.0072x_1x_2/19,000 - 5.0 &= 0, \\
x_1 \sqrt{1 + (\frac{x_2}{x_1})^2} - x_1 + \\
0.0078x_2^2/38,000 - 2.5 &= 0
\end{align*}
\]

Initial values used for all cases were \( x_1^0 = 500.40, \ x_2^0 = 50.085. \)

**Table 9.** \( k = 1 \)

| \( i \) | \( ||P(x^i)|| \) | \( i \) | \( ||P(x^i)|| \) |
|---|---|---|---|
| 0 | 0.01210683086446 | 4 | 0.00006535649707 |
| 1 | 0.00008016430314 | 5 | 0.00006535620690 |
| 2 | 0.00007211180352 | 6 | 0.00005927633464 |
| 3 | 0.00007205986839 | 7 | 0.00005927607008 |
Table 9. (Continued)

| i  | $||P(x^i)||$ | i  | $||P(x^i)||$ |
|----|--------------|----|--------------|
| 8  | 0.00005376181689 | 17 | 0.00003299417688 |
| 9  | 0.00005376157694  | 18 | 0.00003299402966 |
| 10 | 0.00004876032166  | 19 | 0.00002992471565 |
| 11 | 0.00004876010396  | 20 | 0.00002992458206 |
| 12 | 0.00004422392388  | 21 | 0.00002714080859 |
| 13 | 0.00004010992839  | 22 | 0.00002714068746 |
| 14 | 0.00004010974942  | 23 | 0.00002461589008 |
| 15 | 0.00003637848286  | 24 | 0.00002461578028 |
| 16 | 0.00003637832043  | 25 | 0.00002232586646 |

After 24 more steps the following results are presented.

Table 10. Results for $k = 1$

| i  | $x_{49}^i$ | $P(x_{49}^i)$ | $||P(x_{49}^i)||$ | CPU Time |
|----|------------|--------------|------------------|----------|
| 1  | 500.41455957845740 | -7.595x10^{-7} |
| 2  | 50.07289807067660   | 7.589x10^{-6}  | 7.626x10^{-5}  | 2.15     |

Table 11. $k = 2$

| i  | $||P(x^i)||$ | i  | $||P(x^i)||$ |
|----|--------------|----|--------------|
| 0  | 0.01210683086446 | 3  | 0.00001507066163 |
| 1  | 0.00021539305280 | 4  | 0.00000399438629 |
| 2  | 0.00005686062428 | 5  | 0.00000105868638 |
Table 11. (Continued)

<table>
<thead>
<tr>
<th>i</th>
<th>|P(x^i)|</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0.000000028059802</td>
</tr>
<tr>
<td>7</td>
<td>0.00000007437051</td>
</tr>
<tr>
<td>8</td>
<td>0.00000001971165</td>
</tr>
<tr>
<td>9</td>
<td>0.00000000522445</td>
</tr>
<tr>
<td>10</td>
<td>0.00000000138453</td>
</tr>
<tr>
<td>11</td>
<td>0.00000000036700</td>
</tr>
<tr>
<td>12</td>
<td>0.00000000009733</td>
</tr>
<tr>
<td>13</td>
<td>0.00000000002588</td>
</tr>
<tr>
<td>14</td>
<td>0.00000000000668</td>
</tr>
<tr>
<td>15</td>
<td>0.00000000000186</td>
</tr>
<tr>
<td>16</td>
<td>0.00000000000063</td>
</tr>
<tr>
<td>17</td>
<td>0.000000000000010</td>
</tr>
</tbody>
</table>

Table 12. Results for \(k = 2\)

| i \(x_{i}^{17}\) | \(P(x_{i}^{17})\) | \(||P(x^{17})||\) | CPU Time |
|-----------------|-----------------|-----------------|-----------|
| 1 500.41609122936100 | -1.000x10^{-13} |
| 2 50.07289829279880 | 0.0 | 1.000x10^{13} | 1.9 |

Table 13. Newton's method

| i | \(||P(x^i)||\) |
|---|-----------------|
| 0 | 0.01199922704154 |
| 1 | 0.00033203850237 |
| 2 | 0.00068108750100 |
| 3 | 0.00010298899937 |
| 4 | 0.00001948202900 |
| 5 | 0.00001355684486 |
| 6 | 0.00002203448223 |
| 7 | 0.00003900469405 |
| 8 | 0.0006950943327  |
| 9 | 0.0006261227532  |
| 10| 0.00121779961396 |
| 11| 0.00056964240800 |
| 12| 0.00069289742893 |
| 13| 0.00121063007327 |
| 14| 0.00055714959304 |
| 15| 0.00133333324659 |
Table 13. (Continued)

| i  | ||P(x^i)|| | i  | ||P(x^i)|| |
|----|----------|----|----------|
| 16 | 0.00063571261228 | 22 | 0.00015847419092 |
| 17 | 0.00121235680609 | 23 | 0.00063987351394 |
| 18 | 0.00110072605465 | 24 | 0.00057863383967 |
| 19 | 0.00116204293142 | 25 | 0.00054674485510 |
| 20 | 0.00051925913943 | 26 | 0.00120242940653 |
| 21 | 0.00008496567336 | 27 | 0.00114290663467 |

Obviously, Newton's method does not converge, even for the initial values given which are quite close to the solution. For \( k = 2 \), 17 iterations produce the solution. For \( k = 1 \), convergence is occurring as evidenced by the fact that the norm is strictly reduced at every step, however, the convergence is very slow.

Problem 6.4

Next for consideration is the problem presented by F. H. Deist and L. Sefor (27, pp. 81-82) concerning the optimum design for a VHF aerial feeder system. The general system is defined by

\[
f_i = \sum_{j=1}^{n} F_{ij} = 0, \quad i = 1, 2, \ldots, n, \quad 6.7
\]

where

\[
F_{ij} = \cot B_i x_j, \quad i \neq j,
\]

\[
F_{ij} = 0 \quad \text{if} \quad i = j.
\]
The $B_i$'s are known constants dependent on the carrier frequency and the unknown terms $x_i$, $i = 1, 2, \ldots, n$, are the lengths of the connectors. In the particular example for consideration, $n = 6$. The specific system then becomes

\[
\begin{align*}
\sum_{j=2}^{6} \cot B_1 x_j &= 0 \\
\sum_{j=1, j \neq 2}^{6} \cot B_2 x_j &= 0 \\
\sum_{j=1, j \neq 3}^{6} \cot B_3 x_j &= 0 \\
\sum_{j=1, j \neq 4}^{6} \cot B_4 x_j &= 0 \\
\sum_{j=1, j \neq 5}^{6} \cot B_5 x_j &= 0 \\
\sum_{j=1}^{5} \cot B_6 x_j &= 0
\end{align*}
\]

The constants $B_i$, $i = 1, 2, \ldots, 6$ are given in the vector $B$:

\[
B = (0.02249, 0.02166, 0.02083, 0.02000, 0.01918, 0.01835)
\]

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

For this problem cross-sectional and final limit results, that is, results after a maximum number of iterations or maximum amount of CPU time, are given for $k = 1, 3$ and 4.
Results upon convergence are given for \( k = 5 \) and \( 6 \) and for Newton's method.

Table 14. \( k = 1 \)

| \( i \) | \( ||P(x^i)|| \) | \( i \) | \( ||P(x^i)|| \) |
|---|---|---|---|
| 0 | 0.004101046075 | 70 | 0.003751187125 |
| 1 | 0.004087141174 | 80 | 0.003726378144 |
| 2 | 0.004081325394 | 90 | 0.003706473471 |
| 3 | 0.004073343531 | 100 | 0.003681988850 |
| 4 | 0.004058970533 | 125 | 0.003616712771 |
| 5 | 0.004057925406 | 150 | 0.003548009857 |
| 10 | 0.003948295129 | 175 | 0.003478397736 |
| 15 | 0.003890729046 | 200 | 0.003410966744 |
| 20 | 0.003852978880 | 250 | 0.003287263211 |
| 25 | 0.003829149692 | 300 | 0.003163814099 |
| 30 | 0.003815186982 | 350 | 0.003043555568 |
| 40 | 0.003800205034 | 400 | 0.002935556653 |
| 50 | 0.003784123034 | 450 | 0.002827713979 |
| 60 | 0.003770333032 | 500 | 0.002722496584 |

The results after 558 steps are given in Table 15. Table 14 gives only a few cross-sectional values, however, it has been verified that the norm is strictly reduced at each step. Even with this strictly monotonic reduction of norm, the norm is reduced in 558 steps only to 0.001489177108. This is only about a 35 per cent reduction and efficiency is not very good for \( k = 1 \).
Table 15. Results for $k = 1$

| $i$ | $x_i^{558}$ | $(P(x_i^{558}))_i$ | $||P(x_i^{558})||$ | CPU Time |
|-----|-------------|--------------------|---------------------|----------|
| 1   | 126.167986410105 | -0.001251526938     | -0.001251526938     |          |
| 2   | 119.202322380015 | -0.001333925665     | -0.001333925665     |          |
| 3   | 96.260842934784  | 0.000106804978      | 0.000106804978      |          |
| 4   | 57.972572479273  | 0.001103784982      | 0.001103784982      |          |
| 5   | 36.275180788091  | 0.00144638598       | 0.00144638598       |          |
| 6   | 26.498193506654  | 0.00965221983       | 0.00965221983       | 0.002611868967 20.9 |

Table 16. $k = 3$

| $i$ | $||P(x^i)||$ | $i$ | $||P(x^i)||$ |
|-----|-------------|-----|-------------|
| 0   | 0.004101460750 | 70  | 0.002157388411 |
| 1   | 0.004042225518 | 80  | 0.001987026218 |
| 2   | 0.003925770540 | 90  | 0.001833961435 |
| 3   | 0.003850376647 | 100 | 0.001718444913 |
| 4   | 0.003748542803 | 125 | 0.001425125910 |
| 5   | 0.003671869757 | 150 | 0.001180269858 |
| 10  | 0.003493485303 | 175 | 0.000992160527 |
| 15  | 0.003327566638 | 200 | 0.000831529441 |
| 20  | 0.003192721768 | 250 | 0.000590964507 |
| 25  | 0.003066108450 | 300 | 0.000416315932 |
| 30  | 0.002927909054 | 350 | 0.000298280694 |
| 40  | 0.002730156168 | 400 | 0.000214496891 |
| 50  | 0.002506790608 | 450 | 0.000153845871 |
| 60  | 0.002307275798 | 500 | 0.000109845552 |

Table 17 gives the results after 502 steps. Once again
verification has been made of strict norm reduction at each step. Again, convergence is occurring and again very slowly, just slightly faster than for \( k = 1 \).

Table 17. Results for \( k = 3 \)

<table>
<thead>
<tr>
<th>( i )</th>
<th>( x_{i}^{502} )</th>
<th>( (P(x_{i}^{502}))_{i} )</th>
<th>( |P(x_{i}^{502})| )</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>125.993761018479</td>
<td>-0.000042050873</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>119.001437180246</td>
<td>-0.000054623259</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>96.148464679346</td>
<td>-0.000005745100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>58.157642921022</td>
<td>0.000042181148</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>36.484800580341</td>
<td>0.000051265365</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>26.664359041697</td>
<td>0.000051729252</td>
<td>0.000108937424</td>
<td>30.0</td>
</tr>
</tbody>
</table>

Table 18. \( k = 4 \)

<table>
<thead>
<tr>
<th>( i )</th>
<th>( |P(x_{i}^{4})| )</th>
<th>( i )</th>
<th>( |P(x_{i}^{4})| )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.004101046075</td>
<td>60</td>
<td>0.000419274168</td>
</tr>
<tr>
<td>1</td>
<td>0.003859687849</td>
<td>70</td>
<td>0.000301839292</td>
</tr>
<tr>
<td>2</td>
<td>0.003645746661</td>
<td>80</td>
<td>0.000201376219</td>
</tr>
<tr>
<td>3</td>
<td>0.003624573591</td>
<td>90</td>
<td>0.000147942746</td>
</tr>
<tr>
<td>4</td>
<td>0.003559567640</td>
<td>100</td>
<td>0.000107159343</td>
</tr>
<tr>
<td>5</td>
<td>0.003224698280</td>
<td>125</td>
<td>0.000043395992</td>
</tr>
<tr>
<td>10</td>
<td>0.002773334990</td>
<td>150</td>
<td>0.000019033392</td>
</tr>
<tr>
<td>15</td>
<td>0.002187682126</td>
<td>175</td>
<td>0.000007599995</td>
</tr>
<tr>
<td>20</td>
<td>0.001738214821</td>
<td>200</td>
<td>0.000003349755</td>
</tr>
<tr>
<td>30</td>
<td>0.001237732599</td>
<td>250</td>
<td>0.00000647420</td>
</tr>
<tr>
<td>40</td>
<td>0.000874968743</td>
<td>300</td>
<td>0.00000115804</td>
</tr>
<tr>
<td>50</td>
<td>0.000575115751</td>
<td>350</td>
<td>0.00000020406</td>
</tr>
</tbody>
</table>
The results for $k = 4$ after 397 steps, with norm reduction at each step, are given in Table 19.

Table 19. Results for $k = 4$

| i | $x_i^{397}$ | $(P(x_i^{397}))_i$ | $||P(x_i^{397})||$ | CPU Time |
|---|---|---|---|---|
| 1 | 125.985640488617 | -0.0000000001720 | |
| 2 | 118.992129411916 | -0.0000000001908 | |
| 3 | 96.143773915475 | 0.000000000172 | |
| 4 | 58.166445561470 | 0.0000000001817 | |
| 5 | 36.494448747726 | 0.0000000001799 | |
| 6 | 26.671908322816 | 0.0000000001682 | 0.0000000004000 | 30.0 |

For $k = 4$ the results are much more acceptable, however, 397 iterations and 30 seconds computing time is far from desirable. Norm reduction through 397 steps is 0.004101042075, however, this represents a 99 per cent reduction.

The next three calculations, for $k = 5$, the total step with $k = 6$ and Newton's method are presented in Table 20 since all three achieve convergence (of order $10^{-14}$) within the allotted computation time of 30 seconds.

Table 20. $k = 5$, $k = 6$ and Newton's method

| Method | $x_{\text{final}}$ | $||P(x_{\text{final}})||$ | Number of Iterations | CPU Time |
|---|---|---|---|---|
| $k = 5$ | 125.985640192000960 | |
| | 118.992129070154270 | |
Table 20. (Continued)

| Method  | $x^{\text{final}}$ | $||P(x^{\text{final}})||$ | Number of Iterations | CPU Time |
|---------|--------------------|--------------------------|----------------------|----------|
|         |                    |                          |                      |          |
| k = 6   | 125.985640192000770| 118.992129070154080      | 96.14373730789890    | 58.16645876236830 |
|         | 58.166445876237060 | 36.494449105591342       | 26.671908603892371   | 2.9x10^{-15}  |
|         |                    |                          |                      | 215      |
|         |                    |                          |                      | 15.1     |
| Newton  | 125.985640192000770| 118.002129070154060      | 96.14373730789810    | 58.16645876237050 |
|         | 58.166445876237060 | 36.494449105591342       | 26.671908603802524   | 9.7x10^{-16}  |
|         |                    |                          |                      | 4        |
|         |                    |                          |                      | 1.1      |
|         |                    |                          |                      | 1.1x10^{-15}  |
|         |                    |                          |                      | 7        |
|         |                    |                          |                      | 1.8      |

Table 20 illustrates the most spectacular result for this problem, that of the reduction of time and iterations between $k = 5$ and $k = 6$, the total step.

Problem 6.5

Nonlinear network problems occur in several engineering areas, for example, fluid mechanics and electrical circuit theory. The problem involves a set of nodes $N = 1, 2, \ldots, n$, and a set $L$ of unordered pairs $(j,k)$, $j, k \in N$, called links.
Only connected networks are of concern so all nodes have at least one path to any other node. There is a proper nonvoid subset of \( N \) as the set of boundary nodes \( \partial N \).

For each link there are conductance functions \( G_{jk}(s,t) \) and \( G_{kj}(t,s) \) with the properties:

(i) \( G_{jk}(s,t) \) and \( G_{kj}(t,s) \) are continuous in the plane,

(ii) \( G_{jk}(s,t) \) is strictly increasing in \( s \) and decreasing in \( t \). Similarly, \( G_{kj}(t,s) \) is increasing in \( t \), decreasing in \( s \),

(iii) \( G_{jk}(s,t) + G_{kj}(t,s) \) is increasing in \( s \) and \( t \),

(iv) \( G_{jk} \) and \( G_{kj} \) go to \( \infty \) as \( s \) or \( t \) go to \( \infty \).

Define \( G_{jk}(s,t) \) and \( G_{kj}(t,s) \) as 0 if \( (j,k) \not\in L \). If \( s \) and \( t \) are state variables, \( s \) the state at node \( j \) and \( t \) the state at node \( k \), then \( G_{jk}(s,t) \) is the directed flow from node \( j \) to node \( k \) over link \( (j,k) \).

Let \( u = (u_1, u_2, \ldots, u_n) \) be a state vector defined on \( N \). Then the net efflux from node \( j \) is

\[
\mathbf{f}_j(u) = \sum_{k} G_{jk}(u_j, u_k).
\]

The network problem of interest is to determine a state \( u \) which satisfies the equations

\[
\begin{align*}
\mathbf{f}_j(u) &= q_j, \quad j \not\in \partial N, \quad \text{6.10} \\
u_j &= \mathbf{F}_j(\mathbf{f}_j(u)), \quad j \in \partial N,
\end{align*}
\]
where the $q_j$'s are prescribed constants and the $F_j(s)$ functions are continuous decreasing functions defined on $(-\infty, \infty)$.

Consider the actual problem associated with the network shown below.

The set of boundary nodes is $\partial N = \{1,2,3,4\}$ and the conductance functions are

\[ G_{jk}(s,t) = -G_{kj}(t,s) = \text{sgn}(s-t)|s-t|^{\frac{3}{2}}, \quad 6.11 \]

a functional form which occurs in the flow of fluids. The appropriate $F_j$ and $q_j$ are $F_1(s) = F_2(s) = F_3(s) = 10$, $F_4(s) = -s$ and $q_5 = q_6 = q_7 = 0$. With these the system becomes

\[ u_1 - 10 = 0 \]
\[ u_2 - 10 = 0 \]
\[ u_3 - 10 = 0 \]
\[ u_4 + \text{sgn}(u_4-u_6)|u_4-u_6|^{\frac{3}{2}} = 0 \]
\[ \text{sgn}(u_5-u_1)|u_5-u_1|^\frac{1}{2} + \text{sgn}(u_5-u_2)|u_5-u_2|^\frac{1}{2} + \]
\[ \text{sgn}(u_5-u_6)|u_5-u_6|^\frac{1}{3} = 0 \]

\[ \text{sgn}(u_6-u_4)|u_6-u_4|^\frac{1}{2} + \text{sgn}(u_6-u_5)|u_6-u_5|^\frac{1}{2} + \]
\[ \text{sgn}(u_6-u_7)|u_6-u_7|^\frac{1}{3} = 0 \]

\[ \text{sgn}(u_7-u_3)|u_7-u_3|^\frac{1}{2} + \text{sgn}(u_7-u_6)|u_7-u_6|^\frac{1}{2} = 0 \]

The initial value chosen was \((20,1,10,0,0,0,0)\).

Table 21. Convergence results for Problem 6.5

| Method | ||P(x^{final})|| | Iterations | CPU Time |
|--------|-----------------|---------------|------------|
| k = 2  | 0.019039245272887451 | 70 | 2.3 |
| k = 3  | 0.001841797114303695 | 70 | 2.9 |
| k = 4  | 0.000000000000006366 | 70 | 3.2 |
| k = 5  | 0.000000000000000222 | 32 | 1.7 |
| k = 6  | 0.000000000000000222 | 24 | 2.1 |
| k = 7  | 0.0 | 21 | 3.3 |
| Newton | 0.0 | 23 | 2.1 |

Worthy of note in this example is first, the advantage demonstrated by moving from \(k = 3\) to \(k = 4\). The difference in convergence for the same number of iterations and 0.3 seconds computing time is of the order of \(10^{12}\), seemingly a reasonable profit for 0.3 seconds. Secondly, for \(k = 6\) there are 3 more iterations required than for \(k = 7\) but the time required is considerably less, 1.2 seconds. On this
basis, \( k = 6 \) would seem to be more efficient than \( k = 7 \) even though \( k = 7 \) converged faster. Also, Newton's method for this example is essentially the same in terms of iterations and time taken as \( k = 6 \). The solution vector for \( k = 7 \) is 
\[ u = (10.0, 10.0, 10.0, 2.3465995788574218, 9.5706257532103680, 7.8531293804322590, 8.9265648435886880). \]

**Problem 6.6**

This example will deal with the often encountered problem of solving a boundary value problem by numerical means. The attack involves generating a family of difference equations and solving this family. For the problem

\[ y'' = g(x, y), \quad y(a) = A, \quad y(b) = B \quad 6.13 \]

a division of the interval \([a, b]\) is made into \( n \) equal segments and the following definitions made

\[ h = \frac{b-a}{n}, \]
\[ y_i = y(x_i), \quad i = 0, 1, \ldots, n, \]
\[ g_i = g(x_i, y_i), \quad i = 0, 1, \ldots, n, \]
\[ x_0 = a \text{ and } x_n = b. \]

The difference equation used is taken from Henrici (35, p. 349) and has the following form

\[ -y_{i-1} + 2y_i - y_{i+1} + h^2g_i = 0. \quad 6.15 \]
The specific example to be solved also comes from Henrici (35, p. 356) and is given in 6.16.

\[ y'' = -2 + \sin hy, \quad y(0) = 0, \quad y(1) = 0. \]

The interval \([0,1]\) is to be divided into 10 parts so that \(h = 0.1\), and the resulting system, upon implementation of 6.15, is

\[ -y_{i-1} + 2y_i - y_{i+1} + 0.01(-2.0 + \sin hy_i) = 0, \]

\[ i = 1,2,\ldots,9, \]

with \(y_0 = 0\) and \(y_{10} = 0\), resulting in a system of 9 equations in the 9 unknowns \(y_i\), \(i = 1,2,\ldots,9\).

For a system of this size some very pleasing results are obtained. Newton’s method converges in 4 steps and 1.7 seconds but for \(k = 9\) convergence occurs in only 3 iterations and 2.0 seconds. These results in addition to the results for \(k = 3, 6\) and 8 are given in Table 22.

In all cases, the initial value chosen was

\[ y^0 = (0.09, 0.16, 0.21, 0.24, 0.25, 0.24, 0.21, 0.16, 0.09), \]

resulting in an initial norm of 0.005819177392818658.

Table 22. Convergence results for Problem 6.6

<table>
<thead>
<tr>
<th>Method</th>
<th>[||P(x^{\text{final}})|]</th>
<th>Iterations</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k = 3)</td>
<td>0.001422544717219208</td>
<td>223</td>
<td>5.5</td>
</tr>
<tr>
<td>(k = 6)</td>
<td>0.0000000032003971994</td>
<td>223</td>
<td>12.6</td>
</tr>
</tbody>
</table>
Table 22. (Continued)

| Method     | ||P(x\text{final})|| | Iterations | CPU Time |
|------------|----------------|--------------|------------|
| k = 8      | 0.00000000000000002226889 | 223         | 16.8       |
| k = 9      | 0.00000000000000000000019 | 3           | 2.0        |
| Newton     | 0.00000000000000000019  | 4           | 1.7        |

The solution as arrived at with k = 9 and Newton's method is the same to 16 decimal places and is given in 6.18.

\[
y_1 = 0.0824661688236101 \\
y_2 = 0.1457579343618698 \\
y_3 = 0.1905124458657656 \\
y_4 = 0.2171836271731653 \\
y_5 = 0.2260437588573645 \\
y_6 = 0.2171836271731653 \\
y_7 = 0.1905124458657656 \\
y_8 = 0.1457579343618698 \\
y_9 = 0.0824661688236101
\]

Problem 6.7

As given in Chapter I, Moore in (7, p. 90) attempts the solution of the H-function equation in the theory of radiation transfer:

\[
\frac{1}{H(u)} = 1.0 - \frac{\gamma}{w/2} \int_0^1 \frac{u}{u+u'} H(u') \, du'.
\]
Defining

\[ s = u, \]
\[ t = u', \]
\[ x(m) = \frac{1}{H(m)}, \]  

(6.20)

6.19 becomes

\[ x(s) + \frac{w}{2} \int_0^1 \frac{s}{s+t} \frac{1}{x(t)} \, dt - 1.0 = 0. \]  

(6.21)

Using \( w = 1.0 \) and 12-point Gaussian quadrature with weights and abscissas taken from the Handbook of Math Functions (1, p. 916), 6.21 becomes the system

\[ x_i + \frac{1}{2} \sum_{j=1}^{12} \frac{s_i}{s_i + s_j} \frac{1}{x_j} A_j - 1.0 = 0, \quad i = 1,2,\ldots,12, \]  

(6.22)

where \( A_i, \quad i = 1,2,\ldots,12, \) are the Gaussian weights, \( s_i, \quad i = 1,2,\ldots,12, \) are the Gaussian abscissas and \( x_i = x(s_i), \quad i = 1,2,\ldots,12. \) The initial vector is given in 6.23.

\[ x_i^0 = 1 + 2s_i, \quad i = 1,2,\ldots,12. \]  

(6.23)

Table 23. Convergence results for Problem 6.7

| Method | ||P(x_{final})|| | Iterations | CPU Time |
|--------|----------------|--------------|------------|
| k = 6  | 0.002152466150366219 | 249          | 30.0       |
| k = 8  | 0.000856471701662698  | 187          | 30.0       |
For this problem no great advantage seems to be gained by varying $k$ from 6 to 8 even to 11 but the switch from $k = 11$ to $k = 12$ produces a great change. The solution as arrived at by $k = 12$ is given in 6.24.

\[
\begin{align*}
x_1 &= 0.96925940052824170 \\
x_2 &= 0.8835893618693850 \\
x_3 &= 0.78183123310152430 \\
x_4 &= 0.6836340684786172 \\
x_5 &= 0.59778253314193189 \\
x_6 &= 0.52681974063682039 \\
x_7 &= 0.47024712179696811 \\
x_8 &= 0.42641515143645321 \\
x_9 &= 0.39348411378598124 \\
x_{10} &= 0.36984097030822567 \\
x_{11} &= 0.35424304610172028 \\
x_{12} &= 0.34584184421671715
\end{align*}
\]
Problem 6.8

The next problem for consideration involves a system of boundary value problems. Specifically, the two body equations of motion

\[ x''(t) = \frac{-kx(t)}{r^3}, \]
\[ y''(t) = \frac{-ky(t)}{r^3}, \]
\[ z''(t) = \frac{-kz(t)}{r^3}, \]

where \( r = (x^2(t) + y^2(t) + z^2(t))^{1/2}. \)

Specifically, consider the problem with \( k = 1 \) over the interval \( t_0 = 0 \) to \( t_f = 2. \) The following boundary conditions are given.

\[ x(0) = 1.07600, \quad x(2) = 0.0, \]
\[ y(0) = 0.0, \quad y(2) = 0.576, \quad 6.25 \]
\[ z(0) = 0.0, \quad z(2) = 0.997661. \]

The above set-up of the problem was taken from an article by S. M. Roberts and J. S. Shipman (59, p. 406).

The same difference equations are used as in Problem 6 with \( h = 0.2. \) The initial vector is

\[ x^0 = (1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9) \]

\[ x^0 = (1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9) \]
With this initial vector, $||P(x^0)|| = 0.18805605862117$. From this point Newton's method diverges uniformly and after 15 steps has norm greater than 46.

Due to the size of the problem the time involved is relatively large. Therefore, from the above initial point the method $k = 27$ was used to reduce the norm to 0.000066, this taking 26 iterations and 46.7 seconds CPU Time. The approximate solution at this point was then used for all further calculations resulting in the values given in Table 24. Newton's method, however, diverged even from this closer approximation uniformly for 40 iterations.

Table 24. Convergence results for Problem 6.8

| k  | $||P(x_{\text{final}})||$       | Iterations | CPU Time |
|----|--------------------------------|------------|----------|
| 27 | 0.000001825490390535           | 14         | 22.7     |
| 25 | 0.000001617956013108           | 14         | 22.0     |
| 20 | 0.000000728607822989           | 14         |          |
| 20 | 0.000000113997215976           | 19         | 17.7     |
| 15 | 0.000005049265469507           | 14         |          |
| 15 | 0.00000637708841962            | 25         | 13.2     |
| 10 | 0.000213663676466644          | 14         |          |
| 10 | 0.000021020529350444          | 40         | 10.1     |

Contrary to the last few examples the indication given here is that the optimum $k$ is considerably less than the maximum of 27.
The results for \( k = 20 \) after 19 iterations are presented in Table 25.

Table 25. Solution vector for Problem 6.8

<table>
<thead>
<tr>
<th>( i )</th>
<th>( x_i )</th>
<th>( f_i(x^{19}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.10823340983950760</td>
<td>1.83x10^{-9}</td>
</tr>
<tr>
<td>2</td>
<td>1.10117986749786460</td>
<td>1.63x10^{-9}</td>
</tr>
<tr>
<td>3</td>
<td>1.05671510326535230</td>
<td>5.06x10^{-13}</td>
</tr>
<tr>
<td>4</td>
<td>0.977569589571870</td>
<td>2.38x10^{-9}</td>
</tr>
<tr>
<td>5</td>
<td>0.86711473280483100</td>
<td>4.46x10^{-9}</td>
</tr>
<tr>
<td>6</td>
<td>0.72927782325415320</td>
<td>5.75x10^{-9}</td>
</tr>
<tr>
<td>7</td>
<td>0.56847020915975989</td>
<td>5.90x10^{-9}</td>
</tr>
<tr>
<td>8</td>
<td>0.38960146236367388</td>
<td>4.84x10^{-9}</td>
</tr>
<tr>
<td>9</td>
<td>0.19810834216605478</td>
<td>2.74x10^{-9}</td>
</tr>
<tr>
<td>10</td>
<td>0.10605056756460226</td>
<td>8.97x10^{-10}</td>
</tr>
<tr>
<td>11</td>
<td>0.20834163399096331</td>
<td>1.77x10^{-9}</td>
</tr>
<tr>
<td>12</td>
<td>0.30355454940697660</td>
<td>2.60x10^{-9}</td>
</tr>
<tr>
<td>13</td>
<td>0.38880423014980205</td>
<td>3.30x10^{-9}</td>
</tr>
<tr>
<td>14</td>
<td>0.46160344738762911</td>
<td>3.68x10^{-9}</td>
</tr>
<tr>
<td>15</td>
<td>0.51982458757329890</td>
<td>3.63x10^{-9}</td>
</tr>
<tr>
<td>16</td>
<td>0.56167235632897922</td>
<td>3.13x10^{-9}</td>
</tr>
<tr>
<td>17</td>
<td>0.58567497284455255</td>
<td>4.44x10^{-9}</td>
</tr>
<tr>
<td>18</td>
<td>0.59069978984625439</td>
<td>2.80x10^{-9}</td>
</tr>
<tr>
<td>19</td>
<td>0.18368470247636386</td>
<td>8.27x10^{-9}</td>
</tr>
<tr>
<td>20</td>
<td>0.36085777732136014</td>
<td>1.67x10^{-8}</td>
</tr>
<tr>
<td>21</td>
<td>0.5267117233426445</td>
<td>2.43x10^{-8}</td>
</tr>
<tr>
<td>22</td>
<td>0.67342779147695843</td>
<td>5.26x10^{-8}</td>
</tr>
<tr>
<td>23</td>
<td>0.79951966210173820</td>
<td>4.82x10^{-8}</td>
</tr>
<tr>
<td>24</td>
<td>0.90036164756001320</td>
<td>4.08x10^{-8}</td>
</tr>
<tr>
<td>25</td>
<td>0.97284419877298110</td>
<td>3.12x10^{-8}</td>
</tr>
</tbody>
</table>
Table 25. (Continued)

<table>
<thead>
<tr>
<th>i</th>
<th>$x_i$</th>
<th>$f_i(x^{19})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>1.01441810356874090</td>
<td>2.06x10^{-8}</td>
</tr>
<tr>
<td>27</td>
<td>1.02312154657473110</td>
<td>1.00x10^{-8}</td>
</tr>
</tbody>
</table>

Problem 6.9

The last problem is certainly not the least. It concerns a typical optimal control problem and comes from Fournier and Groves (32).

Consider the dynamic system

$$x = F(x,u,t), \quad x = x_0 \text{ at } t = t_0,$$

where $x$ is the state vector, $u$ is the control vector and $t$ is the time. The problem is to find the control $u(t)$ that minimizes the performance index

$$J = \int_0^t L(x,u,t) \, dt.$$

To apply finite difference methods, treat the system equations 6.28 as constraints. Using Lagrange multipliers $\lambda(t)$, adjoin the system equations to the performance index to get a modified performance index

$$K = \int_0^t \{ L(x,u,t) + \lambda[F(x,u,t) - x] \} \, dt.$$
Now the problem is to minimize $K$.

Divide the interval into $n$ equal parts $t_0, t_1, \ldots, t_{n-1}, t_f$. Corresponding to these points are $x_i = x(t_i)$, $u_i = u(t_i)$, $\lambda_i = \lambda(t_i)$, for $i = 1, 2, \ldots, n$. Approximate $K$ by rectangles and derivatives by forward difference quotients to get

$$K = \sum_{i=0}^{n-1} \left[ L(x_i, u_i, t_i) + \lambda_i [F(x_i, u_i, t_i) - x_i] \right] \Delta t. \quad 6.31$$

There are now $3n$ unknowns and $3n$ equations result from differentiation of $K$

$$\frac{\partial K}{\partial x_i} = 0, \quad i = 1, 2, \ldots, n, \quad 6.32$$
$$\frac{\partial K}{\partial u_i} = 0, \quad i = 0, 1, \ldots, n-1, \quad 6.32$$
$$\frac{\partial K}{\partial \lambda_i} = 0, \quad i = 0, 1, \ldots, n-1. \quad 6.32$$

The actual problem to which this is to be applied is

$$x' = -x^2(t) + u(t), \quad x(t=0) = 10.0. \quad 6.33$$

The index of performance is

$$J = \frac{1}{2} \int_0^1 (x^2(t) + u^2(t)) \, dt \quad 6.34$$

so the modified performance index becomes
\[ K = \int_0^1 \left[ x^2(t) + u^2(t) + \lambda(t) \left( -x^2(t) + u(t) - x(t) \right) \right] dt. \]  \[ 6.35 \]

Approximating \( K \) numerically yields

\[ K = \sum_{i=0}^{n-1} \left[ x_i^2 + u_i^2 + \lambda_i \left( -x_i^2 + u_i - \frac{x_{i+1} - x_i}{\Delta t} \right) \right] \Delta t, \]  \[ 6.36 \]

where \( \Delta t = 1/n \).

The equations for solution are

\[
\begin{align*}
\frac{\partial K}{\partial x_i} &= -\lambda_{i-1} + 2x_i \Delta t - 2\lambda_i \Delta t + \lambda_i = 0, \quad i = 1, 2, \ldots, n-1, \\
\frac{\partial K}{\partial x_n} &= -\lambda_{n-1} = 0, \\
\frac{\partial K}{\partial u_i} &= 2u_i \Delta t + \lambda_i \Delta t = 0, \quad i = 0, 1, \ldots, n-1, \\
\frac{\partial K}{\partial \lambda_i} &= -x_i^2 \Delta t + u_i \Delta t - (x_{i+1} - x_i) = 0, \quad i = 0, 1, \ldots, n-1.
\end{align*}
\]  \[ 6.37 \]

Take \( n = 10 \) to give a system of order 30. Initially all \( x \) values were 5, all \( u \) and \( \lambda \) values were 1.

Once again Newton's method begins to diverge from the first step and continues to diverge at each step (60 iterations were made). Newton's method was also attempted from an intermediate point arrived at through iteration with \( k = 30 \). The norm of this point was 0.00063489 as compared to the norm of the initial point of 6.9531642. However, even from this
approximation Newton's method diverged uniformly for 35 more iterations.

Table 26. Convergence results for Problem 6.9

| k  | ||P(x_{final})|| | Iterations | CPU Time |
|----|-----------------|-------------|-----------|
| 30 | 0.000000005464942980 | 13 | 30.0 |
| 28 | 0.000000016403722667 | 13 | |
| 28 | 0.000000007246461423 | 14 | 25.9 |
| 25 | 0.000000035760714926 | 13 | |
| 25 | 0.000000006800722561 | 15 | 24.6 |
| 20 | 0.000000173905094873 | 13 | |
| 20 | 0.00000005493098443 | 19 | 17.9 |
| 10 | 0.00025048136054844 | 13 | |
| 10 | 0.000000060887245110 | 43 | 11.9 |

The approximate solution for k = 30 after 13 iterations is given in Table 27.

Table 27. Solution vector for Problem 6.9

<table>
<thead>
<tr>
<th>i</th>
<th>x_{i}</th>
<th>(f_i(x_{13}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.15795808215396660</td>
<td>5.60\times10^{-17}</td>
</tr>
<tr>
<td>2</td>
<td>7.3313245726811300</td>
<td>2.80\times10^{-17}</td>
</tr>
<tr>
<td>3</td>
<td>8.04771331580550500</td>
<td>1.40\times10^{-17}</td>
</tr>
<tr>
<td>4</td>
<td>8.4569829398182200</td>
<td>2.08\times10^{-16}</td>
</tr>
<tr>
<td>5</td>
<td>8.68264305521561900</td>
<td>1.94\times10^{-16}</td>
</tr>
</tbody>
</table>
### Table 27. (Continued)

<table>
<thead>
<tr>
<th>(i)</th>
<th>(x_i)</th>
<th>(f_i(x^{13}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>8.80480799928205100</td>
<td>1.40x10^{-17}</td>
</tr>
<tr>
<td>7</td>
<td>8.8705088825104000</td>
<td>4.02x10^{-16}</td>
</tr>
<tr>
<td>8</td>
<td>8.90628630235709600</td>
<td>4.02x10^{-16}</td>
</tr>
<tr>
<td>9</td>
<td>8.92747179523254500</td>
<td>2.22x10^{-16}</td>
</tr>
<tr>
<td>10</td>
<td>8.9478673015014800</td>
<td>1.25x10^{-16}</td>
</tr>
<tr>
<td>11</td>
<td>-0.49997143689497536</td>
<td>0.0</td>
</tr>
<tr>
<td>12</td>
<td>-0.49993625881428781</td>
<td>1.00x10^{-18}</td>
</tr>
<tr>
<td>13</td>
<td>-0.49984279735006708</td>
<td>1.00x10^{-18}</td>
</tr>
<tr>
<td>14</td>
<td>-0.49958977297823808</td>
<td>0.0</td>
</tr>
<tr>
<td>15</td>
<td>-0.4988959163933944</td>
<td>1.00x10^{-18}</td>
</tr>
<tr>
<td>16</td>
<td>-0.49697864362138970</td>
<td>1.00x10^{-18}</td>
</tr>
<tr>
<td>17</td>
<td>-0.49165815105917572</td>
<td>1.00x10^{-18}</td>
</tr>
<tr>
<td>18</td>
<td>-0.47685886335906579</td>
<td>0.0</td>
</tr>
<tr>
<td>19</td>
<td>-0.43563854570184040</td>
<td>0.0</td>
</tr>
<tr>
<td>20</td>
<td>-0.32072153211244684</td>
<td>1.00x10^{-18}</td>
</tr>
<tr>
<td>21</td>
<td>0.99994287378995070</td>
<td>3.55x10^{-16}</td>
</tr>
<tr>
<td>22</td>
<td>0.99987251762857550</td>
<td>4.63x10^{-12}</td>
</tr>
<tr>
<td>23</td>
<td>0.99968559470013400</td>
<td>2.62x10^{-11}</td>
</tr>
<tr>
<td>24</td>
<td>0.99917954595647610</td>
<td>8.82x10^{-11}</td>
</tr>
<tr>
<td>25</td>
<td>0.99779183278667880</td>
<td>2.29x10^{-10}</td>
</tr>
<tr>
<td>26</td>
<td>0.99395728724277930</td>
<td>5.03x10^{-10}</td>
</tr>
<tr>
<td>27</td>
<td>0.98331630211835130</td>
<td>9.78x10^{-10}</td>
</tr>
<tr>
<td>28</td>
<td>0.95371772671813150</td>
<td>1.72x10^{-9}</td>
</tr>
<tr>
<td>29</td>
<td>0.87127709140368080</td>
<td>2.79x10^{-9}</td>
</tr>
<tr>
<td>30</td>
<td>0.64144306422489371</td>
<td>4.22x10^{-9}</td>
</tr>
</tbody>
</table>

Once again it must be recognized that to the positions indicated in Table 26, \(k = 20\) and \(k = 30\) have arrived at
essentially the same norm. For \( k = 20 \) there are 6 more iterations completed but 12.1 seconds less computing time. Once again the question of optimization of effort would indicate if measured in terms of iterations that \( k = 30 \) is best but if measured in terms of time, \( k = 20 \) is best.
VII. CONCLUSIONS AND FUTURE RESEARCH

A. Conclusions

The work presented in Chapters II through VI is most notable for the choice available in the projection space. Other considerations that must be made are that it is only slightly more complex than Newton's method, owing to the necessity of using a matrix formed from inner products of the columns of the Jacobian rather than just the Jacobian, but more explicit than other minimization methods. Also, the examples show that it has a wider range of application than Newton's method.

Returning to the question of choice of k, it is easily seen that such a choice can be indispensable and at least worthy of consideration in terms of locating the best approximation to the root in the least amount of time. This work does not attempt to answer questions concerning the optimum choice of k but these are strongly alluded to in the next section, Future Research.

B. Future Research

The computation time involved for any of the projection techniques is dependent to a large extent on the method of solving for the change vectors. As was pointed out in Chapter I, there have been many investigations into solving similar equations arising in Newton's method. The same type of investigations would be possible in the present context.
Also in Chapter I, it was shown that many modifications of the Jacobian matrix were made - the same modifications are applicable to the projection techniques on two levels. First, exactly the same modifications to the Jacobian would result in a modified Jacobian which could be used to derive a modified K-matrix. Secondly, the same principals of modification might be applied to the K-matrix directly, thereby generating a sequence of K-matrices in a prescribed fashion.

More specifically, the very form of the K-matrices suggests a possible simplification of technique. Each element of a K-matrix is an inner product of columns of a Jacobian or modified Jacobian, and the diagonal elements are the norm of these columns, respectively. If the columns of the Jacobian or modified Jacobian are orthogonal the K-matrix becomes a diagonal matrix and thereby produces the change vector immediately. Since, in general, the columns of the Jacobian will not be orthogonal it is possible that orthogonalization might be easier to accomplish than solution of the linear system as is presently necessary and indeed, might be used to predict a new projection space or modified K-matrix or modified Jacobian.

Also, in a broad context the actual minimizing function could be altered, however, the goals of minimizing the norm at each step and maximizing the difference would be retained. The alterations could range from a totally different function to a different expansion of the minimizing function used.
To recall the examples, many worthwhile investigations are suggested here. Primarily, is it possible to select an optimum \( k \) and is it possible, if \( k \) is not total, to select the Jacobian columns in other than the sequential manner used to optimize effort? Combinations of these two questions may then be entertained, for example, varying \( k \) and the choice of Jacobian columns at each step to maximize results.

As is readily evidenced by the previous paragraphs, the most important outcome of this work could be the fact that it is but a stepping stone on the way to bigger and better things. Of course, any progress or any opening made in the area of solution of systems of nonlinear equations is a big step in the right direction.
VIII. BIBLIOGRAPHY


64. Fr. A. Willers, Methoden der Praktischen Analysis, Walter de Gruyter and Co., Berlin, Germany, 1928.

IX. ACKNOWLEDGEMENTS

The author wishes to express his deepest appreciation to Dr. Roy Keller for his guidance and encouragement during the preparation of this dissertation and throughout my graduate program.

I would also like to express appreciation to Dr. Robert Lambert for his generous encouragement and assistance during my graduate work.

I am especially grateful to my family. To my parents for instilling in me the desire and initiative to pursue this course and to my wife and children for their patience and support through the sometimes difficult times associated with graduate study.