DETERMINING NONLINEAR PROJECTION SUBSPACES

Dennis Daniel Georg

NOTICE

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

Ph.D. Thesis Submitted to Iowa State University

Ames Laboratory, ERDA

Iowa State University

Ames, Iowa 50010

Date Transmitted: September 1975

PREPARED FOR THE U. S. ENERGY RESEARCH AND DEVELOPMENT
ADMINISTRATION UNDER CONTRACT NO. W-7405-eng-82

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

RB

DISCLAIMER

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

DISCLAIMER

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

—NOTICE—

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

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

Price: Microfiche \$2.25

Determining nonlinear
projection subspaces

Dennis Daniel Georg

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

Major: Computer Science

Approved:

In Charge of Major Work

For the Major Department

For the Graduate College

Iowa State University
Ames, Iowa

1975

Copyright (C) Dennis Daniel Georg, 1975. All rights reserved.

TABLE OF CONTENTS

	Abstract	Page v
I.	INTRODUCTION	1
II.	PROJECTION SUBSPACE SELECTION CRITERIA	20
	A. τ Criteria	20
	B. Coefficient Criteria	30
	C. Angle Criteria	36
	D. φ Criteria	37
III.	CLASSIFICATION OF NONLINEAR PROBLEMS	45
IV.	APPLICATION OF THE PROJECTION METHOD TO NEARLY LINEAR SYSTEMS OF EQUATIONS	53
V.	COMPARISONS OF SUBSPACE SELECTION CRITERIA	59
VI.	CONCLUSIONS AND FUTURE RESEARCH	96
	A. Conclusions	96
	B. Future Research	98
VII.	ACKNOWLEDGEMENTS	101
VIII.	BIBLIOGRAPHY	102

Determining nonlinear
projection subspaces

Dennis Daniel Georg

Under the supervision of R. F. Keller
From the Department of Computer Science
Iowa State University of Science and Technology

Recently, the projection method for solving linear systems of equations has been extended to nonlinear systems of equations.. The nonlinear projection method is an iterative method which allows its user to modify from one to n , the dimension of the system, components at each iteration step. If n components are modified, then this paper shows that the iteration is mathematically equivalent to Newton's method. The potential advantage offered by the projection method over the total step methods such as Newton's method is that it may be computationally more effective to modify less than n components at certain iteration steps. If m components of the approximate solution vector are modified using the nonlinear projection method, where $1 \leq m \leq n$, then the residue vector associated with the new approximation to the solution vector is projected on a subspace of dimension m . Hence, determining a set of m components to be modified at an iteration step is equivalent to choosing an m dimensional projection subspace. One of the goals of this dissertation is to develop criteria for selecting a computationally efficient sequence of projection subspaces.

In order to meet the above mentioned goal, this paper develops and analyzes four separate criteria for determining projection subspaces. Then, after certain subsets of nonlinear problems are distinguished and classified by extending the class of almost linear problems, the analysis for subspace selection criteria is extended and alternative computational algorithms are developed for these subsets. Finally, several example problems are solved using the projection method with the various criteria as well as Newton's method to illustrate the significant advantages offered by the nonlinear projection method.

Copyright by
DENNIS DANIEL GEORG
1975

I. INTRODUCTION

Obtaining the solutions of nonlinear systems of equations is a nontrivial mathematical and computational task. However, according to Broyden (4), "The solution of a set of nonlinear simultaneous equations is often the final step in the solution of practical problems arising in engineering and physics". That the results obtained analytically and numerically in the area of obtaining solutions to nonlinear systems of equations are immediately applicable to practical problems, motivates the following study.

Newton's method is perhaps the most widely known and applied method for obtaining the solutions of nonlinear systems of equations. Computationally and mathematically, Newton's method is often preferable because of its quadratic convergence. However, Newton's method does have one important practical limitation in that it does not generally converge to some solution from an arbitrary starting point.

The above limitation motivates the consideration and development of alternative solution methods. Recently, the projection method for solving linear systems of equations was developed by MacEachern (22) for nonlinear systems of equations and extended by White (37). The nonlinear projection method fits into the broad classification of methods known as minimization methods.

A nonlinear system of equations is usually represented by the equation $F(x) = 0$ where F maps Euclidean n -space R^n into itself.

If f_i denotes the i th component function, then F can be decomposed as follows:

$$F(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad 1.1$$

Each component function f_i maps Euclidean n -space R^n into R^1 .

For convenience, we let x^k denote the k th approximation to the solution vector. In order to obtain a solution to the system $F(x) = 0$, it is necessary to choose an arbitrary initial approximation x^0 , and then successively calculate new approximations by determining change vectors dx^k at each step according to the scheme:

$$x^{k+1} = x^k + dx^k \quad 1.2$$

Newton's method is considered to be a total step method since all the components of the approximate solution vector are modified at each iteration step. The method developed by MacEachern (22) is a single step method since only one component of the approximate solution vector is changed at any step. His method, which is now known as the one-dimensional nonlinear projection method, can best

be explained by examining a modified form of 1.2:

$$x^{k+1} = x^k + \alpha_k p_k \quad 1.2'$$

α_k is a scalar value and p_k is a vector. The p_k 's used are the columns of an $n \times n$ identity matrix when an n th order system is being solved. The stepsize α_k , calculated in the one-dimensional nonlinear projection method, is optimal in the sense that the difference between the two Euclidean norms $\|F(x^k)\|$ and $\|F(x^{k+1})\|$ is maximized.

If J_i^k denotes the i th column of the Jacobian matrix of $F(x)$ evaluated at x^k , then α_k is calculated as follows when the i th component of the approximate solution vector is being modified.

$$\alpha_k = - \frac{(F(x^k), J_i^k)}{(J_i^k, J_i^k)} \quad 1.3$$

(a,b) denotes the inner product of the two similarly dimensioned vectors a and b .

MacEachern (22) shows that the nonsingularity of the Jacobian matrix is not necessary for convergence for the one-dimensional nonlinear projection method. However, Newton's method requires a nonsingular Jacobian at each iteration step. On some of the examples described by MacEachern (22), the one-dimensional nonlinear projection method competed computationally with Newton's method. The fact that Newton's method performed significantly better on other examples motivated the work done by White (37).

White (37) developed the computational formulas for what will henceforth be referred to as the nonlinear projection method. When solving an n -dimensional nonlinear system using the nonlinear projection method, n_k components of the approximate solution vector can be modified at step k where $1 \leq n_k \leq n$. If j components are modified at step k , then step k is said to have been a j -dimensional projection. The following is a more general form of the development originally done by White (37).

Suppose that a j -dimensional projection is used to modify x^k to form x^{k+1} . The j components which are to be modified can be denoted by $B = \{i_1, i_2, \dots, i_j\}$. Each component function $f_m(x^{k+1})$, $m = 1, 2, \dots, n$ can be expanded in a Taylor's polynomial about the point x^k . After truncating the second order term, the following series of equations are obtained:

$$f_m(x^{k+1}) = f_m(x^k) + (dx^k, J_{f_m}) \quad 1.4$$

$$m = 1, 2, \dots, n$$

Expanding the inner product of the equations in 1.4 yields:

$$f_m(x^{k+1}) = f_m(x^k) + \sum_{v=1}^n dx_v f_{mx_v}(x^k) \quad 1.5$$

$$m = 1, 2, \dots, n$$

where dx_v is the v^{th} component of dx and $f_{mx_v} = \partial f_m / \partial x_v$. Note

that $dx_v = 0$ for all $v \notin B$ so

$$f_m(x^{k+1}) = f_m(x^k) + \sum_{v \in B} dx_v f_{mx_v}(x^k)$$

$$m = 1, 2, \dots, n$$

White's (37) goal in choosing the stepsizes was exactly the same as MacEachern's (22). That is, dx^k is calculated in such a way as to maximize the difference between the two Euclidean norms $||F(x^k)||$ and $||F(x^{k+1})||$. To simplify the procedure, the difference between the norms squared is considered. Define

$$G(dx) = ||F(x^k)||^2 - ||F(x^{k+1})||^2$$

Using the Euclidean norm, we find that

$$G(dx) = \sum_{m=1}^n (f_m(x^k))^2 - \sum_{m=1}^n (f_m(x^k) + \sum_{v \in B} dx_v f_{mx_v}(x^k))^2 \quad 1.6$$

In order to find a candidate for a maximum point of G , we need to

1. expand the second term in 1.6
2. differentiate the resulting expression with respect to dx_r , $r \in B$; set these expressions to 0, and solve for dx .

The result of this manipulation produces the following set of equations:

$$\sum_{m \in B} (J_m^k, J_r^k) dx_m = - (F(x^k), J_r^k) \quad 1.7$$

$$r \in B$$

In order to solve for the change components for a j -dimensional projection, it is necessary to solve the $j \times j$ linear system defined in 1.7.

White (37) proves that under certain conditions a sequence of successive j -dimensional projections will force the sequence of approximate solution vectors $\{x^i\}_{i=0}^{\infty}$ to converge to an x^* such that $F(x^*) = 0$. His result is contained in the following Theorem (37).

Theorem 1.1

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

In addition to modifying j components of the approximate solution vector $B_k = \{x_{i_1}^k, x_{i_2}^k, \dots, x_{i_j}^k\}$, a j -dimensional projection also forces the residue vector, $r^{k+1} = F(x^{k+1})$, to be perpendicular to the j columns, $i_1, i_2, \dots, i_j$, of the Jacobian matrix evaluated at the previous approximate solution vector x^k . The name projection method thus arises from the fact that one step of a j -dimensional projection method results in the residue being projected on a subspace of dimension j .

Consider the following example (Broyden (5)):

Example 1.1.

$$f_1(x) = 0.1x_1^2 - 3.0x_1 + 2.0x_2 - 1.0$$

$$f_2(x) = x_1 + 0.1x_2^2 - 3.0x_2 + 2.0x_3 - 1.0$$

$$f_3(x) = x_2 + 0.1x_3^2 - 3.0x_3 - 1.0$$

$$x^0 = (1.0, 1.0, 1.0)$$

$$r^0 = (-1.9, -.9, -2.9)$$

$$||r^0|| = 3.58190$$

Table 1.1. Results of various single projections on Example 1.1

Dimension	Subspace	$ r^1 $
1	(1)	3.25509
1	(2)	3.38097
1	(3)	3.06222
2	(1, 2)	3.25504
2	(1, 3)	2.83568
2	(2, 3)	3.05882
3	(1, 2, 3)	1.04890

As can be seen in the example above, the amount of reduction obtained in the norm of the residue vector at each iteration step is dependent not only on the dimension of the projection subspace, but also on the columns contained in the subspace. Therefore,

the sequence of projections used in solving a nonlinear system is important.

In using one-dimensional projections a cycle is said to have been completed when all the components have been modified at least one time. The notion of a cycle can be extended to include j -dimensional projections although the definition is not as strict. A cycle, in this case, is said to have been completed when each component of the approximate solution vector has been modified at least once by a fixed sequence of projections. The projections in the cycle may or may not be of the same dimension. Cycles that contain t projections are called t -step cycles. A t -step cycle must necessarily include each column of the Jacobian at least one time. The repeated use of a fixed t -step cycle results in a stationary iterative method.

Nonstationary projection methods for solving nonlinear systems of equations use projection subspaces which are dynamically chosen based on criteria which are calculated at each iteration step. Even though the evaluation of these criteria introduces additional computational complexity to the solution process, the use of the correct criteria might significantly reduce the total amount of computation that is needed to solve a given problem. One of the objectives of this work is to develop and analyze subspace selection criteria for the nonlinear projection method.

A subspace is said to be optimal if projecting on that subspace produces the maximum reduction in the norm of the residue vector. Pyron (30), Georg (13), and Tokko (36) have investigated the determination of optimal and quasi-optimal 2-dimensional and 3-dimensional linear projection subspaces. Their work can be extended to obtain criteria for determining optimal 1, 2, or 3 dimensional nonlinear projection subspaces.

In the computational and mathematical analysis of methods for solving nonlinear systems of equations there are two significant problems. The first problem is that we are dealing with an approximation to the change vector dx^k at each iteration step. Secondly, in most cases, the linear system that must be solved in order to obtain dx^k dynamically depends on x^k .

Throughout the remainder of this paper, it will be convenient to use the following notation:

1. $\theta_{\delta\beta}$ refers to the angle between vectors δ and β
2. $f_{mx_v} = \partial f_m / \partial x_v$

The problems mentioned above are both present even in the analysis of determining an optimal one-dimensional subspace.

Consider the following lemma:

Lemma 1.1

If the i th column of the Jacobian is projected on at step k (i.e., $B_k = \{i\}$), then

$$\begin{aligned}
1. \quad \Delta r^k &= ||r^k||^2 - ||r^{k+1}||^2 \\
&= (r^k, r^k) - (r^{k+1}, r^{k+1}) \\
&= \frac{(r^k, J_i^k)^2}{(J_i^k, J_i^k)}
\end{aligned}$$

$$2. \quad ||r^{k+1}||^2 = ||r^k||^2 \sin^2 \theta_{r^k J_i^k}$$

Proof:

1. Each component function evaluated at x^{k+1} can be expressed as a truncated Taylor's polynomial:

$$f_m(x^{k+1}) = f_m(x^k) + \sum_{v \in B_k} dx_v f_{mx_v}(x^k) \quad 1.8$$

$$m = 1, 2, \dots, n$$

2. 1.8 can be re-written as follows

$$f_m(x^{k+1}) = f_m(x^k) - \sum_{v \in B_k} - dx_v f_{mx_v}(x^k) \quad 1.9$$

3. After squaring both sides of 1.9 and summing over all m , the following expression results:

$$\begin{aligned}
\sum_{m=1}^n (f_m(x^{k+1}))^2 &= \sum_{m=1}^n [(f_m(x^k))^2 - 2dx_i f_m(x^k) f_{mx_i}(x^k) \\
&\quad + (dx_i)^2 f_{mx_i}(x^k) f_{mx_i}(x^k)] \\
&= \sum_{m=1}^n (f_m(x^k))^2 - 2dx_i (F(x^k), J_i^k) \\
&\quad + dx_i^2 (J_i^k, J_i^k)
\end{aligned} \quad 1.10$$

4. Using 1.3, we know that $dx_i = - \frac{(F(x^k), J_i^k)}{(J_i^k, J_i^k)}$.

Thus, since $\|r^k\|^2 = \sum_{m=1}^n (f_m(x^k))^2$, 1.10 can be simplified

$$\|r^{k+1}\|^2 = \|r^k\|^2 - \frac{(r^k, J_i^k)^2}{(J_i^k, J_i^k)} \quad 1.11$$

5. From 1.11, the results are obtained directly:

$$\|r^k\|^2 - \|r^{k+1}\|^2 = \frac{(r^k, J_i^k)^2}{(J_i^k, J_i^k)} \quad 1.12$$

$$\begin{aligned} \|r^{k+1}\|^2 &= \|r^k\|^2 - \|r^k\|^2 \cos^2 \theta_{rk, J_i^k} \\ &= \|r^k\|^2 (1 - \cos^2 \theta_{rk, J_i^k}) \\ &= \|r^k\|^2 (\sin^2 \theta_{rk, J_i^k}) \end{aligned} \quad 1.13$$

Lemma 1.1 illustrates that we are making an approximation to dx^k and that the expressions which are evaluated to determine an optimal subspace do depend on x^k . In addition, Lemma 1.1 does demonstrate how it would be possible to determine an optimal one-dimensional subspace.

Choosing optimal one-dimensional subspaces may mean that certain columns would be projected on more often than others. However, since all the columns of the Jacobian must be projected on, it may not be computationally advantageous to select optimal one-dimensional

subspaces. Experimentally, it has been shown, when larger subspaces are considered, that choosing optimal or quasi-optimal subspaces can be advantageous. The following two lemmas illustrate the work that must be done in order to choose an optimal 2-dimensional subspace.

Lemma 1.2

If the i th and j th columns are projected on at step k (i.e., $B_k = \{i, j\}$), then

$$dx_i^k = \frac{c_j^i}{(J_i^k, J_i^k)(J_j^k, J_j^k)} [(r^k, J_i^k)(J_j^k, J_j^k) - (r^k, J_j^k)(J_i^k, J_j^k)]$$

$$dx_j^k = \frac{c_i^j}{(J_i^k, J_i^k)(J_j^k, J_j^k)} [(r^k, J_j^k)(J_i^k, J_i^k) - (r^k, J_i^k)(J_j^k, J_i^k)]$$

$$\text{where } c_j^i = \frac{1}{1 - \cos^2 \theta_{J_i^k J_j^k}}$$

Proof:

1. Using 1.7, we have

$$(J_i^k, J_i^k)dx_i + (J_j^k, J_i^k)dx_j = - (F(x^k), J_i^k)$$

$$(J_i^k, J_j^k)dx_i + (J_j^k, J_j^k)dx_j = - (F(x^k), J_j^k)$$

2. After applying Cramer's rule to the above equations,
the result is

$$dx_i = \frac{(r^k, J_i^k)(J_j^k, J_j^k) - (r^k, J_j^k)(J_i^k, J_j^k)}{(J_i^k, J_i^k)(J_j^k, J_j^k) - (J_i^k, J_j^k)^2}$$

$$dx_j = \frac{(r^k, J_j^k)(J_i^k, J_i^k) - (r^k, J_i^k)(J_i^k, J_j^k)}{(J_i^k, J_i^k)(J_j^k, J_j^k) - (J_i^k, J_j^k)^2}$$

3. Now, using $1 - \frac{(J_i^k, J_j^k)^2}{(J_i^k, J_i^k)(J_j^k, J_j^k)} = 1 - \cos^2 \theta_{J_i^k J_j^k}$

$$\begin{aligned} dx_i &= \frac{1}{1 - \cos^2 \theta_{J_i^k J_j^k}} \left[\frac{(r^k, J_i^k)}{(J_i^k, J_i^k)} - \frac{(r^k, J_j^k)(J_i^k, J_j^k)}{(J_i^k, J_i^k)(J_j^k, J_j^k)} \right] \\ &= \frac{C_j^i}{(J_i^k, J_i^k)(J_j^k, J_j^k)} [(r^k, J_i^k)(J_j^k, J_j^k) - (r^k, J_j^k)(J_i^k, J_j^k)] \end{aligned}$$

4. Similarly

$$dx_j = \frac{C_i^j}{(J_i^k, J_i^k)(J_j^k, J_j^k)} [(r^k, J_j^k)(J_i^k, J_i^k) - (r^k, J_i^k)(J_i^k, J_j^k)]$$

Lemma 1.3

If the i th and j th columns of the Jacobian are projected on
at step k (i.e., $B_k = \{i, j\}$), then

$$\begin{aligned}
\Delta r^k &= (r^k, r^k) - (r^{k+1}, r^{k+1}) \\
&= (r^k, r^k) C_j^i [\cos^2 \theta_{r^k J_i^k} + \cos^2 \theta_{r^k J_j^k} \\
&\quad - 2 \cos \theta_{J_i^k J_j^k} \cos \theta_{r^k J_i^k} \cos \theta_{r^k J_j^k}] \quad 1.14
\end{aligned}$$

$$\text{where } C_j^i = \frac{1}{1 - \cos^2 \theta_{J_i^k J_j^k}}$$

Proof:

1. $f_m(x^{k+1})$ can be expanded as a truncated Taylor's polynomial:

$$\begin{aligned}
f_m(x^{k+1}) &= f_m(x^k) + \sum_{v \in B_k} dx_v f_{mx_v}(x^k) \quad 1.15 \\
&= f_m(x^k) - [dx_i f_{mx_i}(x^k) - dx_j f_{mx_j}(x^k)]
\end{aligned}$$

2. After squaring both sides of 1.16 and summing over all m , the following expression results:

$$\begin{aligned}
\sum_{m=1}^n (f_m(x^{k+1}))^2 &= \sum_{m=1}^n [(f_m(x^k))^2 \\
&\quad - 2 f_m(x^k) \{-dx_i f_{mx_i}(x^k) - dx_j f_{mx_j}(x^k)\} \\
&\quad + \{-dx_i f_{mx_i}(x^k) - dx_j f_{mx_j}(x^k)\}^2] \\
&= \sum_{m=1}^n (f_m(x^k))^2 \\
&\quad - 2 [dx_i (F(x^k), J_i^k) - dx_j (F(x^k), J_j^k)] \\
&\quad + dx_i^2 (J_i^k, J_i^k) + 2 dx_i dx_j (J_i^k, J_j^k) + dx_j^2 (J_j^k, J_j^k) \quad 1.16
\end{aligned}$$

3. Now, after rearranging 1.16 and substituting the values for dx_i and dx_j calculated in Lemma 1.2, the desired expression results.

Using Lemmas 1.2 and 1.3 it would be possible to determine an optimal two dimensional subspace. Quasi-optimal projection subspaces can be determined by considering only parts of the expression 1.15. For example, it may be advantageous to choose i and j which maximizes C_j^i . Pyron (30) makes a strong case for this criteria in the case of linear projection subspaces. Georg (13), on the other hand, investigated the results of choosing quasi-optimal subspaces on the basis of the direction cosines associated with θ_{rkj_i} and θ_{rkj_j} .

The determination of quasi-optimal 2-dimensional projection subspaces would require less computation than the determination of optimal 2-dimensional subspaces. These criteria, however, are still dynamically dependent on the current approximation to the solution vector, and therein hangs part of the problem mentioned above.

Similar analysis, to that carried out in Lemmas 1.1, 1.2, and 1.3, has been done for the determination of optimal and quasi-optimal linear 3-dimension projection subspaces (Tokko (36)). A parallel result can be obtained for nonlinear systems but the form is prohibitive. In any event, the important observation to make is that the choice of an optimal 2-dimensional or 3-dimensional nonlinear projection subspace depends on the angles between the

columns of the Jacobian, as well as the angles between the residue vector and the columns of the Jacobian. It appears that the choice of optimal higher dimensional projection subspaces might also depend on these angles.

In order to determine a quasi-optimal or optimal 2-dimensional projection subspace, it would be necessary to calculate 1.15 for $\frac{n(n-1)}{2}$ pairs of columns if the system being solved is of dimension n , $n \geq 2$. The number of triples of columns that would need to be considered in order to determine a quasi-optimal or optimal 3-dimensional subspace would be $\frac{n(n-1)(n-2)}{6}$, $n \geq 3$. This fact, in addition to the experimental observation that projections on smaller subspaces result in slow convergence, motivates the consideration of larger dimensioned projection subspaces. The slow convergence results from r^k becoming almost perpendicular to all the columns of the Jacobian while $\|r^k\|$ is still large.

The largest projection subspace would contain all n columns of the Jacobian. Theorem 1.2 proves that one n -dimensional projection is mathematically equivalent to one step of Newton's method.

Theorem 1.2

Consider solving the n -dimensional nonlinear system $F(x) = 0$. Assume x^{k+1} is being calculated from x^k . One n -dimensional projection is mathematically equivalent to one step of Newton's method.

Proof:

1. The n-dimensional system that must be solved in order to obtain the change vector dx^k for an n-dimensional projection can be determined from 1.7. In matrix notation this system is:

$$J^T J dx^k = - J^T F \quad 1.17$$

2. For Newton's method, the change vector dx^k can be obtained by solving the linear system

$$J dx^k = - F \quad 1.18$$

3. After multiplying both sides of 1.17 by $(J^T J)^{-1}$ and simplifying,

$$\begin{aligned} dx^k &= - (J^T J)^{-1} J^T F \\ &= - J^{-1} (J^T)^{-1} J^T F \\ &= - J^{-1} F \end{aligned} \quad 1.19$$

4. After multiplying both sides of 1.19 by J, the desired result 1.18 is obtained.

It is important to stress that while the n-dimensional projection method and Newton's method are mathematically equivalent, they are not computationally equivalent.

White (37) notes in comparing various subspace dimensions that projecting on less than n columns may be more efficient computationally than projecting on all n columns. This efficiency is important, but equally significant is the fact that the projection method provides alternatives when the linear systems that must be solved for an

n-dimensional projection or Newton's method are singular. The following two examples illustrate these points.

Example 1.2.

$$f_1(x) = x_1 * x_2 / x_3 - x_4 * x_5$$

$$f_2(x) = x_1 * x_3 / x_2 + x_4 + x_5 - 2.0$$

$$f_3(x) = x_2 * x_3 / x_1 - x_4^2$$

$$f_4(x) = x_1 / (x_2 * x_3) - x_5^2$$

$$f_5(x) = x_2 / (x_1 * x_3) - x_4^2 + x_5^2 - 1.0$$

$$x^0 = (1.0, 1.0, 1.0, -2.0, 2.0)$$

$$r^0 = (5.0, -5.0, -3.0, -3.0, 0.0)$$

$$||r^0|| = 8.24621$$

Table 1.2. Result of single projections on Example 1.2

Dimension	Subspace	$ r^1 $
5	(1,2,3,4,5)	7.71497
4	(1,2,3,4)	7.80230
4	(1,2,3,5)	8.90330
4	(1,2,4,5)	6.20509
4	(1,3,4,5)	2.35483
4	(2,3,4,5)	7.71499
3	(1,3,5)	11.69840
3	(1,4,5)	2.82843
3	(3,4,5)	2.51791

The results in Table 1.2 clearly indicates that projecting on less than n columns may be computationally more efficient. In the following example one step of Newton's method requires the solution of a system of linear equations that is singular. In this case, the alternatives offered by the projection method are clearly advantageous.

Example 1.3.

$$f_1(x) = x_1^2 + x_1 x_2 + x_1$$

$$f_2(x) = x_1 x_2^2 - x_1$$

$$J = \begin{bmatrix} 2x_1 + x_2 + 1 & x_1 \\ x_2^2 - 1 & 2x_1 x_2 \end{bmatrix}$$

$$x^0 = (-1.0, 1.0)$$

$$r^0 = (-1.0, 0)$$

$$J(x^0) = \begin{bmatrix} 0.0 & -1.0 \\ 0.0 & -2.0 \end{bmatrix}$$

At each iteration step, the projection method requires the specification of the subspace dimension and Jacobian columns which will make up the projection subspace. The results above motivate the consideration of effective and efficient criteria for determining these projection subspaces. Such criteria are presented and analyzed in the remaining chapters.

II. PROJECTION SUBSPACE SELECTION CRITERIA

Projection subspace selection criteria, henceforth referred to as criteria, will be the main subject of this chapter. Although the principal use of the criteria developed in this chapter will be to dynamically determine projection subspaces, the criteria may also be used to help determine cycles of projections.

A. τ Criteria

In Chapter I, expressions were obtained for Δr^k for both one and two-dimensional projections. Similar analysis could be carried out in order to obtain expressions for Δr^k for subspaces with dimensions larger than two. However, as Example 1.1 illustrated, it may be advantageous to consider varying the projection subspace dimensions. Thus, it may be helpful to analyze an approximation for Δr^k that is independent of the subspace dimension.

In the analysis to follow, these definitions will be used:

Definition 2.1:

Let n_k denote the dimension of the projection subspace at iteration step k .

Definition 2.2:

Let $B_k = \{i_1^k, i_2^k, \dots, i_{n_k}^k\}$ denote the columns of the Jacobian which determine the projection subspace at iteration step k .

Definition 2.3:

Let S_k denote a $n_k \times n$ selection matrix used at the k th iteration step. S_k is comprised of those rows of the $n \times n$ identity matrix which

correspond to the columns of the Jacobian which are contained in B_k .

Definition 2.4:

Let d_k denote the n_k dimensional vector of change components that are solved for at the k th iteration step.

Definition 2.5:

Let J_k denote the Jacobian matrix associated with F evaluated at x^k .

The selection matrix can be used to obtain a matrix expression for the linear system that must be solved to obtain d_k at step k . This result is contained in the following lemma:

Lemma 2.1

At the k th iteration step, if $S_k J_k^T J_k S_k^T$ is nonsingular, then the $n_k \times n_k$ linear system that must be solved in order to obtain d_k is

$$(S_k J_k^T J_k S_k^T) d_k = - S_k (J_k^T F_k)$$

$$\text{where } F_k = F(x^k).$$

Proof:

This result follows directly from 1.7.

Remark 2.1

It follows directly from Lemma 2.1 that the n_k component vector d_k can be obtained directly:

$$d_k = - (S_k J_k^T J_k S_k^T)^{-1} (S_k (J_k^T F_k))$$

Theorem 2.1 gives a bound for $\Delta r^k = ||r^k|| - ||r^{k+1}||$.

Theorem 2.1

Assume that each component function f_i , $i = 1, 2, \dots, n$ of the nonlinear system $F(x) = 0$ is at least twice continuously differentiable. Then $\Delta r^k \leq || - J_k S_k^T d_k + \theta_k ||$ where

$$\theta_k = \begin{bmatrix} (S_k^T d_k, (f''_1(z_1), S_k^T d_k)) \\ (S_k^T d_k, (f''_2(z_2), S_k^T d_k)) \\ \vdots \\ (S_k^T d_k, (f''_n(z_n), S_k^T d_k)) \end{bmatrix}$$

and $z_i \in [x^k, x^{k+1}]$, $i = 1, 2, \dots, n$.

Proof:

1. Each component function can be expanded in a Taylor's polynomial about x^k .

$$f_j(x^{k+1}) = f_j(x^k) + (S_k^T d_k, J_{f_j}(x^k)) + (S_k^T d_k, (f''_j(z), S_k^T d_k))$$

$$j = 1, 2, \dots, n$$

$$\text{where } z \in [x^k, x^{k+1}]$$

(note: $J_{f_j}(x^k)$ denotes the j th row of the Jacobian evaluated at x^k)

$$\begin{aligned} 2. \quad \Delta r^k &= ||F(x^k)|| - ||F(x^k + S_k^T d_k)|| \\ &= ||F(x^k)|| - ||(f_1(x^k) + (S_k^T d_k, J_{f_1}(x^k)) \\ &\quad + (S_k^T d_k, (f''_1(z_1), S_k^T d_k)), f_2(x^k) \end{aligned}$$

$$\begin{aligned}
& + (S_{k,k}^T d_k, J_{f_2}(x^k)) + (S_{k,k}^T d_k, (f_2''(z_2), S_{k,k}^T d_k)), \\
& \dots, f_n(x^k) + (S_{k,k}^T d_k, J_{f_n}(x^k)) \\
& + (S_{k,k}^T d_k, (f_n''(z_n), S_{k,k}^T d_k)) ||
\end{aligned} \tag{2.1}$$

where $z_i \in [x^k, x^{k+1}]$, $i = 1, 2, \dots, n$

3. 2.1 can be rewritten using matrix notation

$$\Delta r^k = ||F(x^k)|| - ||F(x^k) + J_k S_{k,k}^T d_k - \theta_k|| \tag{2.2}$$

4. Using the fact that $||A|| - ||B|| \leq ||A - B||$, the following result can be obtained from 2.2:

$$\Delta r^k \leq || - J_k S_{k,k}^T d_k + \theta_k || \tag{2.3}$$

The form of the approximation to Δr^k generated in Theorem 2.1 is still dependent on the subspace dimension, since this factor necessarily affects S_k . A further analysis, however, will develop a criteria independent of n_k . The following remark defines the approximation that will be analyzed:

Remark 2.2

Dropping the second order terms in $(d_k)^2$ from 2.3 generates the following approximation for Δr^k

$$\Delta r^k \doteq \hat{\Delta r}^k = || - J_k S_{k,k}^T d_k || \tag{2.4}$$

Now, instead of trying to maximize Δr^k , suppose instead that subspaces are chosen which maximize $\hat{\Delta r}^k$. Assuming that a reasonable criteria can be developed, this may be a computationally advantageous

goal. Theorem 2.2 develops $\hat{\Delta r}^k$ into a form that can be examined for a criteria.

Theorem 2.2

$\hat{\Delta r}^k$ can be maximized at step k by selecting S_k in such a way as to maximize

$$\left\{ \sum_{m=1}^n \left(\sum_{q=1}^n \sum_{p=1}^{n_k} \sum_{j=1}^{n_k} J_{mq}^T q_i J_j (S_k J_k^T J_k S_k^T)^{-1} \sum_{r=1}^n \sum_{s=1}^n S_{i_r} J_{sr} F_s \right)^2 \right\}^{\frac{1}{2}} \quad 2.5$$

Proof:

1. $\hat{\Delta r}^k = || - J_k S_k^T d_k ||$
2. Using the definition of d_k from Remark 2.1

$$\begin{aligned} \hat{\Delta r}^k &= || J_k S_k^T (S_k J_k^T J_k S_k^T)^{-1} (S_k (J_k^T F_k)) || \\ &= || J_k^k T^k Z^k Y^k || \end{aligned}$$

$$\text{where } Z^k = (S_k J_k^T J_k S_k^T)^{-1}$$

$$Y^k = (S_k (J_k^T F_k))$$

$$T^k = S_k^T$$

$$J^k = J_k$$

3. Before finding the norm, consider the expansion of

$$J_k^k T^k Z^k Y^k$$

$$4. \quad J_T^k Z_Y^k = J_T^k \begin{bmatrix} \sum_{j=1}^{n_k} z_{1j}^k y_j^k \\ \sum_{j=1}^{n_k} z_{2j}^k y_j^k \\ \vdots \\ \sum_{j=1}^{n_k} z_{n_k j}^k y_j^k \end{bmatrix}$$

5. For convenience, drop the dependence on k . Then

$$J_T Z Y = \begin{bmatrix} J_{11} J_{12} \cdots J_{1n} \\ J_{12} J_{22} \cdots J_{2n} \\ \vdots \\ J_{1n} J_{2n} \cdots J_{nn} \end{bmatrix} \begin{bmatrix} \sum_{p=1}^{n_k} \sum_{j=1}^{n_k} T_{1i_j} z_{jp} y_p \\ \sum_{p=1}^{n_k} \sum_{j=1}^{n_k} T_{2i_j} z_{jp} y_p \\ \vdots \\ \sum_{p=1}^{n_k} \sum_{j=1}^{n_k} T_{ni_j} z_{jp} y_p \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{q=1}^n J_{1q} \sum_{p=1}^{n_k} \sum_{j=1}^{n_k} T_{qi_j} Z_{jp} Y_p \\ \sum_{q=1}^n J_{2q} \sum_{p=1}^{n_k} \sum_{j=1}^{n_k} T_{qi_j} Z_{jp} Y_p \\ \vdots \\ \sum_{q=1}^n J_{nq} \sum_{p=1}^{n_k} \sum_{j=1}^{n_k} T_{qi_j} Z_{jp} Y_p \end{bmatrix}$$

6. Now, applying the definition of the Euclidean norm to the vector determined in (5),

$$\begin{aligned} \hat{\Delta r}^k &= \left\{ \sum_{m=1}^n \left(\sum_{q=1}^n \sum_{p=1}^{n_k} \sum_{j=1}^{n_k} J_{mq} T_{qi_j} Z_{jp} Y_p \right)^2 \right\}^{\frac{1}{2}} \\ &= \left\{ \sum_{m=1}^n \left(\sum_{q=1}^n \sum_{p=1}^{n_k} \sum_{j=1}^{n_k} J_{mq} T_{qi_j} (S_k^J T_k^J T_k^T)^{-1} \sum_{r=1}^n \sum_{s=1}^n S_{i_p r}^J S_{sr}^F Y_p \right)^2 \right\}^{\frac{1}{2}} \\ &\quad \text{since } Y_p = (S_k^J T_k^F)_p = \sum_{r=1}^n \sum_{s=1}^n S_{i_p r}^J S_{sr}^F \end{aligned}$$

It is not apparent from a direct examination of 2.5 that $\hat{\Delta r}^k$ will assume a maximum value when all the columns are projected on. The exclusion of certain columns, however, may have a small effect on $\hat{\Delta r}^k$ at step k . Hence, it may be computationally advantageous to select a smaller subspace and project on it and yet achieve a very good reduction in the norm of the residue vector. Computational

savings are realized since the linear system that must be solved for d_k would be smaller.

An optimal criteria for choosing projection subspaces on the basis of $\hat{\Delta}r^k$ would consider the entire expression in 2.5. Such a procedure would require the calculation of $\sum_{i=1}^n \binom{n}{i}$ inverses of matrices. Because of this, a sub-expression of 2.5 is examined in order to determine a quasi-optimal criteria. The goal in choosing a projection subspace would then be to exclude those columns which may have only a minor effect on the magnitude of $\hat{\Delta}r^k$. One estimate of the effect of the inclusion or exclusion of a particular column in the projection subspace can be obtained by examining:

$$\tau_{i_p} = \left(\sum_{r=1}^n \sum_{s=1}^n S_{i_p r} J_{sr} F_s \right)^2$$

The following lemma reduces the expression for τ_t to a simpler form.

Lemma 2.3

$$\tau_t = \left(\sum_{s=1}^n J_{st} F_s \right)^2$$

Proof:

$$\begin{aligned} 1. \quad \tau_t &= \left(\sum_{r=1}^n \sum_{s=1}^n S_{tr} J_{sr} F_s \right)^2 \\ &= \left(\sum_{r=1}^n S_{tr} \sum_{s=1}^n J_{sr} F_s \right)^2 \end{aligned}$$

2. Now using an algebraic result proven by White (37)

$$\begin{aligned}\tau_t &= \sum_{i=1}^n \sum_{r=1}^n S_{tr} \left(\sum_{s=1}^n J_{sr} F_s \right) S_{ti} \left(\sum_{s=1}^n J_{si} F_s \right) \\ &= \sum_{i=1}^n \sum_{r=1}^n S_{tr} S_{ti} \left(\sum_{s=1}^n J_{sr} F_s \right) \left(\sum_{s=1}^n J_{si} F_s \right)\end{aligned}$$

3. Now, since by definition only one element from each row of S will be nonzero, each summand above will be zero unless $i = r = t$

$$\tau_t = \left(\sum_{s=1}^n J_{st} F_s \right)^2$$

Using the results of Lemma 2.3, the procedure that would be followed in order to determine a projection subspace would be as follows. At each iteration step calculate

$$\tau_i = \left(\sum_{m=1}^n J_{mi} F_m \right)^2 \quad i = 1, 2, \dots, n$$

Then if τ_i is greater than some tolerance (dependent on k) column i should be included in the projection subspace. Note that calculating the τ_i 's is not much extra work since $\sum_{m=1}^n J_{mj} F_m$, $j \in B_k$ make up the right sides of the system being solved for d_k .

Determining the tolerance that is to be used in deciding whether a particular column is to be included or excluded from the projection subspace is the most difficult part of using the criteria developed from Theorem 2.2. The following lemma motivates the use of $\|r^k\|^2$ as a viable tolerance.

Lemma 2.4

Assume $J_{mt}^k \geq 0 \forall m, t$ and let

$$\max_t = \max_m (J_{mt}) \quad \min_t = \min_m (J_{mt})$$

$$\text{Then } (\min_t)^2 \{ ||r^k|| \}^2 \leq \tau_t \leq (\max_t)^2 \{ ||r^k|| \}^2$$

Proof:

$$1. \left(\sum_{m=1}^n J_{mt} F_m(x^k) \right) \leq \left(\sum_{m=1}^n \max_t F_m(x^k) \right)^2 = (\max_t)^2 ||r^k||^2$$

$$2. \text{ Similarly } \left(\sum_{m=1}^n J_{mt} F_m(x^k) \right) \geq (\min_t)^2 \{ ||r^k|| \}^2 \text{ and the}$$

result follows immediately.

There are two main difficulties with the use of the τ_i as a criteria. These difficulties are (1) that we are working with $\hat{\Delta}r^k$ instead of Δr^k and (2) the determination of a universal tolerance. However, the criteria is very consistent with strict angle criteria which indicates not to project on any column to which the residue is almost perpendicular. In fact, if $\theta_{rkJ_i^k}$ is the angle between the residue vector r^k and the i th column of the Jacobian J_i^k , then

$$\cos^2 \theta_{rkJ_i^k} = \frac{\tau_i}{||r^k|| ||J_i^k||}.$$

The goal of obtaining a criteria independent of the subspace dimension has been achieved. Each column is examined individually to determine whether or not it should be included in the projection subspace.

B. Coefficient Criteria

An analysis of an approximation to Δr^k resulted in the development of the τ criteria. Another criteria can be developed by examining the linear approximations to the component functions. As part of this analysis, the information contained in the Hessian matrices associated with F will be used.

The first step in the development of the projection method was to express each component function $f_m(x^{k+1})$, $m = 1, 2, \dots, n$, in a Taylor's polynomial. Then, all second as well as higher ordered terms were truncated. This results in the establishment of a linear approximation to the change vector at each iteration step. The fact that this linear approximation may be good or bad could be an important consideration in determining an effective projection subspace. Suppose, for example, that each component of the change vector dx was known to satisfy an equation of the form:

$$d_i = a_i dx_i + b_i dx_i^2 + c_i dx_i^3 \quad 2.6$$

$$i = 1, 2, \dots, n$$

If this were the case, large values of b_i and c_i would tend to indicate that the linear approximation was not very good.

The present goal then is to develop a criteria which will take advantage of good linear approximations to the components of the change vector. In the first step of this analysis, the projection

method is redeveloped using the second order terms of the Taylor expansions to the component functions. The result is a set of cubic equations which can be solved for dx . Even though it may not be computationally desirable to solve such a cubic system, the coefficients from the cubic equation that must be solved for a one-dimensional projection can be examined to determine the goodness of the linear approximation for a change vector with one nonzero component.

The following work results in the development of the system of cubic equations in dx_i , $i = 1, 2, \dots, n$. Each component function f_i can be expressed in a truncated Taylor's polynomial:

$$f_i(x^{k+1}) = f_i(x^k) + (dx^k, J_{f_i}(x^k)) + (dx^k, (f_i''(x^k), dx^k)) \quad 2.7$$

In order to choose dx^k in such a way as to maximize the difference between the two successive norms, we consider the following function:

$$\begin{aligned} H(dx) &= ||F(x^k)||^2 - ||F(x^{k+1})||^2 \\ &= \sum_{i=1}^n (f_i(x^k))^2 - \sum_{i=1}^n [f_i(x^k) + (dx^k, J_{f_i}(x^k)) \\ &\quad + (dx^k, (f_i''(x^k), dx^k))]^2 \\ &= \sum_{i=1}^n (f_i(x^k))^2 - \left[\sum_{i=1}^n (f_i(x^k))^2 \right. \\ &\quad \left. + 2 \sum_{i=1}^n f_i(x^k) (dx^k, J_{f_i}(x^k)) \right] \end{aligned}$$

$$\begin{aligned}
& + 2 \sum_{i=1}^n F_i(x^k)(dx^k, (f_i''(x^k), dx^k)) \\
& + \sum_{i=1}^n (dx^k, J_{f_i}(x^k))(dx^k, (f_i''(x^k), dx^k)) \\
& + \sum_{i=1}^n (dx^k, J_{f_i}(x^k))^2 \\
& + \sum_{i=1}^n (dx^k, (f_i''(x^k), dx^k))^2] \quad 2.8
\end{aligned}$$

Now, in order to simplify the analysis, the dependence on k of dx and x will be dropped. Assume that B is the dimension of the projection subspace which consists of the set of columns $\{i_1, i_2, \dots, i_B\}$. After multiplying both sides of 2.8 by -1 and expanding the inner products, the following equation is obtained

$$\begin{aligned}
-H(dx) &= 2 \sum_{i=1}^n f_i(x) \sum_{J=1}^n dx_J f_{i x_J}(x) \\
&+ 2 \sum_{i=1}^n f_i(x) \sum_{J=1}^n \sum_{k=1}^n dx_J dx_k \partial_J \partial_k f_i(x) \\
&+ 2 \sum_{i=1}^n \left[\sum_{J=1}^n dx_J f_{i x_J}(x) \right] \left[\sum_{J=1}^n \sum_{k=1}^n dx_J dx_k \partial_J \partial_k f_i(x) \right] \\
&+ \sum_{i=1}^n \left(\sum_{J=1}^n dx_J f_{i x_J}(x) \right)^2 \\
&+ \sum_{i=1}^n \left(\sum_{J=1}^n \sum_{k=1}^n dx_J dx_k \partial_J \partial_k f_i(x) \right)^2 \quad 2.9
\end{aligned}$$

To find a maximum point of H , 2.9 is differentiated with respect to dx_r , $r \in \{i_1, i_2, \dots, i_B\}$ and the resulting expressions are set equal to zero to get the following equations:

$$\begin{aligned}
 0 = & 2 \sum_{i=1}^n f_i(x) f_{ix_r}(x) \\
 & + 2 \sum_{i=1}^n f_i(x) \left(\sum_{j=1}^n dx_j \partial_j \partial_r f_i(x) + \sum_{j=1}^n dx_j \partial_r \partial_j f_i(x) \right) \\
 & + 2 \sum_{i=1}^n [(f_{ix_r}(x)) \left(\sum_{j=1}^n \sum_{k=1}^n dx_j dx_k \partial_j \partial_k f_i(x) \right) \\
 & \quad + (2 \sum_{k=1}^n dx_k \partial_k \partial_r f_i(x)) \left(\sum_{j=1}^n dx_j f_{ix_j}(x) \right)] \\
 & + \sum_{i=1}^n 2 \sum_{j=1}^n dx_j f_{ix_j} f_{ix_r} \\
 & + \sum_{i=1}^n 2 \left[\sum_{j=1}^n \sum_{k=1}^n dx_j dx_k \partial_j \partial_k f_i(x) \right] \left(2 \sum_{k=1}^n dx_k \partial_k \partial_r f_i(x) \right) \quad 2.10
 \end{aligned}$$

$$r \in \{i_1, i_2, \dots, i_B\}$$

Let $f_i = f_i(x)$ and $f_{ix_r} = f_{ix_r}(x)$. Then, after grouping like terms together, 2.10 can be rewritten using inner product notation:

$$\begin{aligned}
 0 = & 2 (f_i, f_{ix_r} + 2 \sum_{j=1}^n dx_j \partial_j \partial_r f_i) \\
 & + 2 (f_{ix_r}, \sum_{j=1}^n dx_j f_{ix_j} + \sum_{j=1}^n \sum_{k=1}^n dx_j dx_k \partial_j \partial_k f_i)
 \end{aligned}$$

$$\begin{aligned}
& + 4 \left(\sum_{k=1}^n dx_k \partial_k \partial_r f_i, \sum_{j=1}^n dx_j f_{ix_j} + \sum_{j=1}^n \sum_{k=1}^n dx_j dx_k \partial_j \partial_k f_i \right) \\
& r \in \{i_1, i_2, \dots, i_B\} \quad 2.11
\end{aligned}$$

After multiplying both sides of 2.11 by $-\frac{1}{2}$ and rearranging the terms, the following equations are obtained:

$$\begin{aligned}
& \sum_{j=1}^n (f_{ix_j}, f_{ix_r}) dx_j = \quad 2.12 \\
& - \sum_{i=1}^n f_i (f_{ix_r} + 2 \sum_{j=1}^n dx_j \partial_j \partial_r f_i) \\
& - \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n f_{ix_r} dx_j dx_k \partial_j \partial_k f_i \\
& - 2 \left(\sum_{k=1}^n dx_k \partial_k \partial_r f_i, \sum_{j=1}^n dx_j f_{ix_j} + \sum_{j=1}^n \sum_{k=1}^n dx_j dx_k \partial_j \partial_k f_i \right) \\
& r \in \{i_1, i_2, \dots, i_B\}
\end{aligned}$$

With the use of the following notation 2.12 can finally be reduced to a compact form which is the system of cubic equations desired in this analysis. Let

$$\begin{aligned}
\alpha_{ir} &= \sum_{k=1}^n dx_k \partial_k \partial_r f_i \\
\beta_i &= \sum_{j=1}^n \sum_{k=1}^n dx_j dx_k \partial_j \partial_k f_i \\
\gamma_{ir} &= f_{ix_r} + 2\alpha_{ir} \\
g_i &= f_i + \beta_i
\end{aligned}$$

$$\text{Then, } \sum_{j=1}^n (f_{ix_j}, \gamma_{ir}) dx_j = - \sum_{i=1}^n g_i \gamma_{ir} \quad 2.13$$

$$r \in \{i_1, i_2, \dots, i_B\}$$

Note the similarity between the system of cubic equations in 2.13 and the system of linear equations that are solved in the regular projection method shown in Chapter I (see 1.7).

Suppose now that we are just projecting on the a^{th} column. Then, $dx_i = 0$ for $i \neq a$. In this case, the single cubic equation in dx_a can be obtained directly from 2.13:

$$\begin{aligned} & [(f_i, \frac{\partial f_i}{\partial x_a})] \\ & + [(\frac{\partial f_i}{\partial x_a}, \frac{\partial f_i}{\partial x_a}) + 2 (f_i, \frac{\partial^2 f_i}{\partial^2 x_a})] dx_a \\ & + 3 [(\frac{\partial^2 f_i}{\partial^2 x_a}, \frac{\partial f_i}{\partial x_a})] dx_a^2 \\ & + 2 [(\frac{\partial^2 f_i}{\partial^2 x_a}, \frac{\partial^2 f_i}{\partial^2 x_a})] dx_a^3 \\ & = 0 \end{aligned} \quad 2.14$$

If the coefficients of dx_a^2 and dx_a^3 are small, then it would be expected that a good approximation to dx_a could be obtained by solving the single linear equation, derived from Equation 1.7 when $B = \{a\}$, for dx_a . The main use of this criteria is to determine

which one-dimensional projections would result in good linear approximations to the components of the change vector. In order to use the criteria, it would be necessary to either explicitly or numerically calculate the coefficients of dx_a , $a = 1, 2, \dots, n$ in 2.14 at each iteration step. For the general nonlinear system of equations this is a formidable task. However, in the case of a system which has constant second partials, this criteria may be valuable as an aid to determining effective cycles of projections. This usage will be exploited further in Chapter 4.

C. Angle Criteria

Another criteria can be developed by examining an approximation to the angle criteria. As mentioned previously, the angle criteria includes or excludes a column from the projection subspace on the basis of the angle between the column and the current residue vector. The reasoning being that if the angle between the residue vector and the i th column of the Jacobian, denoted by θ_{rkjk_i} , is approximately 90° , then projecting on column i will not produce a good contribution to the reduction of the norm of the residue vector.

Working directly with the angles is not computationally convenient. Consequently, values of the cosines of the angles squared are evaluated. By definition

$$\cos^2 \theta_{rkjk_i} = \frac{(r^k, j_i^k)^2}{(r^k, r^k)(j_i^k, j_i^k)} \quad 2.15$$

To use the strict angle criteria, the values of $\gamma_i = \cos^2 \theta_{rkj_i^k}$, $i = 1, 2, \dots, n$ are calculated at each iteration step. Then, if γ_i is greater than some predetermined tolerance, column i is included in the projection subspace. The determination of a tolerance is one of the drawbacks of the strict angle criteria. Another criticism of the angle criteria is that for a system of dimension n , it requires $2n^2 + n + 3$ multiplications at each step in order to evaluate the γ_i 's.

D. φ Criteria

An alternative to the angle criteria can be obtained by using an approximation to the γ_i 's. This criteria is motivated by the fact that if column i was contained in the subspace that was projected on in order to obtain x^{k+1} , then $\theta_{rk+1j_i^k} = 90^\circ$. Let $\Delta_i^k = J_i^{k+1} - J_i^k$. Now, if the change in the i th column of the Jacobian, Δ_i^k , is small it would be expected that $\theta_{rk+1j_i^{k+1}} \approx 90^\circ$. If this were the case, column i should be, using the same reasoning motivating the angle criteria, excluded from the next projection subspace. The result of the following theorem is an expression for γ_i in terms of Δ_i^k .

Theorem 2.3

If $J_i^{k+1} = J_i^k + \Delta_i^k$, then

$$\cos^2 \theta_{rk+1j_i^{k+1}} = \cos^2 \theta_{rk+1\Delta_i^k} \left[1 - \frac{Z_i^k}{(\Delta_i^k, \Delta_i^k) + Z_i^k} + Q_i^k \right]$$

$$\text{where } z_i^k = [2(j_i^k, \Delta_i^k) + (j_i^k, j_i^k)]$$

$$Q_i^k = \frac{(\Delta_i^k, \Delta_i^k)}{(r^{k+1}, \Delta_i^k)^2} \left[\frac{(r^{k+1}, j_i^k)(r^{k+1}, j_i^k + 2\Delta_i^k)}{(\Delta_i^k, \Delta_i^k) + (j_i^k, j_i^k + 2\Delta_i^k)} \right]$$

Proof:

1. By definition

$$\cos^2 \theta_{r^{k+1} j_i^{k+1}} = \frac{(r^{k+1}, j_i^{k+1})^2}{(r^{k+1}, r^{k+1})(j_i^{k+1}, j_i^{k+1})} \quad 2.16$$

2. After introducing the fact that $j_i^{k+1} = j_i^k + \Delta_i^k$ in

2.16 and expanding the result

$$\cos^2 \theta_{r^{k+1} j_i^{k+1}} = \frac{\{(r^{k+1}, j_i^k)^2 + 2(r^{k+1}, j_i^k)(r^{k+1}, \Delta_i^k) + (r^{k+1}, \Delta_i^k)^2\}}{\{(r^{k+1}, r^{k+1})(j_i^k, j_i^k) + 2(r^{k+1}, j_i^k)(j_i^k, \Delta_i^k) + (r^{k+1}, r^{k+1})(\Delta_i^k, \Delta_i^k)\}}$$

3. Let

$$a = (r^{k+1}, \Delta_i^k)^2$$

$$b = (r^{k+1}, j_i^k)^2 + 2(r^{k+1}, j_i^k)(r^{k+1}, \Delta_i^k)$$

$$c = (r^{k+1}, r^{k+1})(\Delta_i^k, \Delta_i^k)$$

$$d = (r^{k+1}, r^{k+1})(j_i^k, j_i^k) + 2(r^{k+1}, r^{k+1})(j_i^k, \Delta_i^k)$$

$$\begin{aligned}
\text{then } \cos^2 \theta_{r^{k+1} J_1^{k+1}} &= \frac{a+b}{c+d} \\
&= a/c \left[1 + \frac{bc - ad}{ac - ad} \right] \\
&= \cos^2 \theta_{r^{k+1} \Delta_1^k} [1 + h(a, b, c, d)] \quad 2.17
\end{aligned}$$

4. Consider $h(a, b, c, d)$

$$h(a, b, c, d) =$$

$$\begin{aligned}
&\{[(r^{k+1}, J_1^k)^2 + 2(r^{k+1}, J_1^k)(r^{k+1}, \Delta_1^k)](r^{k+1}, \Delta_1^k) \\
&\quad - [(r^{k+1}, r^{k+1})(J_1^k, J_1^k) + 2(r^{k+1}, r^{k+1})(J_1^k, \Delta_1^k)](r^{k+1}, \Delta_1^k)^2\} \\
&\frac{\{[(r^{k+1}, \Delta_1^k)^2[(r^{k+1}, r^{k+1})(\Delta_1^k, \Delta_1^k)] + (r^{k+1}, \Delta_1^k)^2[(r^{k+1}, r^{k+1})(J_1^k, J_1^k) \\
&\quad + 2(r^{k+1}, r^{k+1})(J_1^k, \Delta_1^k)]\}}{(r^{k+1}, \Delta_1^k)^2[(r^{k+1}, r^{k+1})(\Delta_1^k, \Delta_1^k)] + (r^{k+1}, \Delta_1^k)^2[(r^{k+1}, r^{k+1})(J_1^k, J_1^k) \\
&\quad + 2(r^{k+1}, r^{k+1})(J_1^k, \Delta_1^k)]\}} \quad 2.18
\end{aligned}$$

5. After removing the factor $\frac{(r^{k+1}, r^{k+1})}{(r^{k+1}, r^{k+1})}$ from 2.18

$$h(a, b, c, d) =$$

$$\begin{aligned}
&\{[(r^{k+1}, J_1^k)^2 + 2(r^{k+1}, J_1^k)(r^{k+1}, \Delta_1^k)](\Delta_1^k, \Delta_1^k) \\
&\quad - [(J_1^k, J_1^k) + 2(J_1^k, \Delta_1^k)](r^{k+1}, \Delta_1^k)^2\} \\
&\frac{\{[(r^{k+1}, \Delta_1^k)^2(\Delta_1^k, \Delta_1^k) + (r^{k+1}, \Delta_1^k)^2[(J_1^k, J_1^k) + 2(J_1^k, \Delta_1^k)]\}}{(r^{k+1}, \Delta_1^k)^2[(J_1^k, J_1^k) + 2(J_1^k, \Delta_1^k)]\}} \quad 2.19
\end{aligned}$$

6. Remove the factor $\frac{(r^{k+1}, \Delta_1^k)^2}{(r^{k+1}, \Delta_1^k)^2}$ from 2.19 and rearrange the terms

$$h(a,b,c,d) =$$

$$\frac{\left\{ \left[\frac{(r^{k+1}, J_i^k)^2}{(r^{k+1}, \Delta_i^k)^2} + \frac{2(r^{k+1}, J_i^k)}{(r^{k+1}, \Delta_i^k)} \right] (\Delta_i^k, \Delta_i^k) - [(J_i^k, J_i^k) + 2(J_i^k, \Delta_i^k)] \right\}}{\{(\Delta_i^k, \Delta_i^k) + [(J_i^k, J_i^k) + 2(J_i^k, \Delta_i^k)]\}}$$

$$= \frac{\left\{ \left[\frac{(r^{k+1}, J_i^k)}{(r^{k+1}, \Delta_i^k)} \right] (\Delta_i^k, \Delta_i^k) \left[2 - \frac{(r^{k+1}, J_i^k)}{(r^{k+1}, \Delta_i^k)} \right] - Z_i^k \right\}}{(\Delta_i^k, \Delta_i^k) + Z_i^k}$$

$$= \frac{-Z_i^k}{(\Delta_i^k, \Delta_i^k) + Z_i^k} + P_i^k$$

$$\text{where } P_i^k = \frac{\left[\frac{(r^{k+1}, J_i^k)}{(r^{k+1}, \Delta_i^k)} \right] (\Delta_i^k, \Delta_i^k) \left[2 - \frac{(r^{k+1}, J_i^k)}{(r^{k+1}, \Delta_i^k)} \right]}{(\Delta_i^k, \Delta_i^k) + Z_i^k}$$

7. Now, reduce P_i^k to the form of Q_i^k and the desired result is obtained.

$$\begin{aligned} P_i^k &= \frac{\left[\frac{(r^{k+1}, J_i^k)^2}{(r^{k+1}, \Delta_i^k)^2} + \frac{2(r^{k+1}, J_i^k)}{(r^{k+1}, \Delta_i^k)} \right] (\Delta_i^k, \Delta_i^k)}{\{(\Delta_i^k, \Delta_i^k) + 2(J_i^k, \Delta_i^k) + (J_i^k, J_i^k)\}} \\ &= \frac{\frac{(r^{k+1}, J_i^k)^2}{(r^k, \Delta_i^k)^2} + 2(r^{k+1}, J_i^{k+1})(r^{k+1}, \Delta_i^k)}{(\Delta_i^k, \Delta_i^k) + 2(J_i^k, \Delta_i^k) + (J_i^k, J_i^k)} \\ &= \frac{(r^{k+1}, J_i^k)^2 + 2(r^{k+1}, J_i^{k+1})(r^{k+1}, \Delta_i^k)}{(\Delta_i^k, \Delta_i^k) + 2(J_i^k, \Delta_i^k) + (J_i^k, J_i^k)} \end{aligned}$$

$$\begin{aligned}
&= \frac{(\Delta_i^k, \Delta_i^k)}{(r^{k+1}, \Delta_i^k)^2} \left[\frac{(r^{k+1}, J_i^k)^2 + 2(r^{k+1}, J_i^k)(r^{k+1}, \Delta_i^k)}{(\Delta_i^k, \Delta_i^k) + 2(J_i^k, \Delta_i^k) + (J_i^k, J_i^k)} \right] \\
&= \frac{(\Delta_i^k, \Delta_i^k)}{(r^{k+1}, \Delta_i^k)^2} = \left[\frac{(r^{k+1}, J_i^k)(r^{k+1}, J_i^k + 2\Delta_i^k)}{(\Delta_i^k, \Delta_i^k) + (J_i^k, J_i^k + 2\Delta_i^k)} \right] \\
&= Q_i^k
\end{aligned}$$

The fact that $\cos^2 \theta_{r^{k+1} J_i^k} = 0$ if column i was contained in the subspace used to obtain x^{k+1} allows an extension to be made to the result of Theorem 2.3.

Corollary 2.1

Assume that column i was contained in the projection subspace at step k , then

$$\begin{aligned}
\cos^2 \theta_{r^{k+1} J_i^{k+1}} &= \cos^2 \theta_{r^{k+1} \Delta_i^k} \left[1 - \frac{Z_i^k}{(\Delta_i^k, \Delta_i^k) + Z_i^k} \right] \\
\text{where } Z_i^k &= [2(J_i^k, \Delta_i^k) + (J_i^k, J_i^k)]
\end{aligned}$$

Proof:

1. From Theorem 2.3, it is known that

$$\begin{aligned}
\cos^2 \theta_{r^{k+1} J_i^{k+1}} &= \cos^2 \theta_{r^{k+1} \Delta_i^k} \left[1 - \frac{Z_i^k}{(\Delta_i^k, \Delta_i^k) + Z_i^k} \right] \\
&+ \frac{(\Delta_i^k, \Delta_i^k)}{(r^{k+1}, \Delta_i^k)^2} \left[\frac{(r^{k+1}, J_i^k)(r^{k+1}, J_i^k + 2\Delta_i^k)}{(\Delta_i^k, \Delta_i^k) + (J_i^k, J_i^k + 2\Delta_i^k)} \right]
\end{aligned}$$

2. If $\cos^2 \theta_{r,k+1,j_i^k} = 0$, then it follows from the definition of the cosine that $(r^{k+1}, j_i^k) = 0$. Hence, $Q_i^k = 0$ and the result follows directly.

For convenience of the discussion to follow let,

$$\varphi_i^k = [1 - \frac{z_i^k}{(\Delta_i^k, \Delta_i^k) + z_i^k} + Q_i^k]$$

Then, it follows from Theorem 2.3 that

$$\cos^2 \theta_{r,k+1,j_i^{k+1}} = (\cos^2 \theta_{r,k+1,\Delta_i^k}) \varphi_i^k$$

An optimal angle criteria would calculate $\cos^2 \theta_{r,k+1,j_i^{k+1}}$, $i = 1, 2, \dots, n$, at each iteration step. A candidate for a quasi-optimal angle criteria would be the φ_i^k , $i = 1, 2, \dots, n$. This criteria is attractive because of the general form of φ_i^k .

If column i was contained in the previous projection subspace, then we know from Corollary 2.1 that $Q_i^k = 0$. Thus, $0 \leq \varphi_i^k$ and the magnitude of φ_i^k is determined by the relative magnitudes of z_i^k and (Δ_i^k, Δ_i^k) . The smaller the magnitude of (Δ_i^k, Δ_i^k) relative to z_i^k , the closer φ_i^k will be to zero. So, using the φ criteria, column i would be reincluded in the projection subspace if (Δ_i^k, Δ_i^k) 's magnitude was relatively large compared to cz_i^k where c is a constant.

On the other hand, if column i was not included in the previous subspace, Q_i^k would not necessarily be zero. In this case the direct use of the φ criteria would not be computationally convenient because

of the form of Q_1^k . However, if it is assumed that the residue has been significantly reduced so that Q_1^k is small, then the φ criteria may still be a viable criteria. In this case the contribution of the Q_1^k term to φ_1^k is neglected.

The φ criteria does require less computation than the strict angle criteria and the requirement for a tolerance is more easily met since a constant c need only be specified once. These two factors are the main motivations for its use. In Chapter 4, the φ criteria will be further exploited for a special class of nonlinear problems.

In this chapter, four criteria were presented for determining nonlinear projection subspaces. These criteria included:

1. τ - criteria
2. angle or γ - criteria
3. coefficient criteria
4. φ - criteria

These criteria, with the exception of the coefficient criteria are mainly intended to be used as dynamic subspace selection guides. In all cases, the criteria are quasi-optimal in the sense that they do not guarantee the determination of the optimal subspace sequence. However, the use of these criteria should result in greater computational efficiency than the selection of arbitrary projection subspaces. Chapter 5 contains examples which compare the use of these criteria.

The next two chapters define a classification of nonlinear problems and examine the use of the coefficient and ϕ criteria on a special class of nonlinear problems.

III. CLASSIFICATION OF NONLINEAR PROBLEMS

A mathematical investigation which considers the set of all nonlinear systems of equations is generally more pleasing than one concerning only certain types of nonlinear problems. In many cases, however, the investigation can only be extended for special classes of problems. This is presently the case when subspace selection criteria are being examined.

In this chapter, the classification of nonlinear systems of equations is considered. The classification is made on the properties of the Jacobian and Hessian matrices associated with the nonlinear system $F(x) = 0$. These matrices are important, since it is the information they contain that describe how the columns of the Jacobian are changing.

A system of dimension n has n associated Hessian matrices; one matrix associated with each component function. The components of a Hessian matrix are the second partial derivatives of the component function the matrix is associated with. Let H_a^k denote the Hessian matrix corresponding to the a^{th} component function evaluated at x^k .

$$H_a^k = \begin{bmatrix} \frac{\partial f_a}{\partial x_1 \partial x_2} & \frac{\partial f_a}{\partial x_2 \partial x_1} & \dots & \frac{\partial f_a}{\partial x_n \partial x_1} \\ \frac{\partial f_a}{\partial x_1 \partial x_2} & \frac{\partial f_a}{\partial x_2 \partial x_2} & \dots & \frac{\partial f_a}{\partial x_n \partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_a}{\partial x_1 \partial x_n} & \dots & \dots & \frac{\partial f_a}{\partial x_n \partial x_n} \end{bmatrix}$$

Rheinboldt and Ortega (26) distinguish a class of nonlinear problems which they classify as being almost linear. The almost linear class of problems is defined below in Definition 3.2. This definition makes use of the concept of a diagonal mapping described in Definition 3.1.

Definition 3.1:

A nonlinear mapping $L : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ is diagonal if for $i = 1, 2, \dots, n$, the (i,i) component, $L_{i,i}$, is a function of only the i th variable x_i and $L_{i,j} = 0$ for $i \neq j$.

Definition 3.2:

A mapping $F : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ is almost linear if F can be written in the form $F = Ax + L$ where A is an $n \times n$ matrix and L is diagonal.

The essence of this definition is that the a^{th} component function can be written in the form:

$$f_a(x^k) = \sum_{\substack{i=1 \\ i \neq a}}^n c_{ai} x_i^k + L_{a,a}$$

where the c_{ai} , $i = 1, 2, \dots, n$ are constants. Thus for an almost linear problem, the off-diagonal components of the Jacobian are constants and the diagonal components are functions of at most one variable. The Jacobian matrix for an almost linear problem has the form:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & c_{12} & c_{13} & \dots & c_{1n} \\ c_{21} & \frac{\partial f_2}{\partial x_2} & \dots & \dots & c_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{n1} & \dots & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

where $\partial f_i / \partial x_i$ is a function of only one variable x_i . Each of the Hessian matrices associated with an almost linear problem have at most one nonzero element. The nonzero component of the Hessian matrix associated with the a th component is either constant or a function of x_a .

The following are examples of systems which are almost linear:

Example 3.1. Almost Linear Problem

$$f_1(x) = (x_1)^2 - x_2$$

$$f_2(x) = (x_2)^2 + x_1$$

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$L_{1,1} = x_1^2$$

$$L_{2,2} = x_2^2$$

$$H_1 = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

$$H_2 = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}$$

Example 3.2. Almost Linear Problem

$$f_1(x) = \sin x_1 + 3x_2$$

$$f_2(x) = x_1 - e^{-x_2}$$

$$A = \begin{bmatrix} 0 & 3 \\ 1 & 0 \end{bmatrix}$$

$$L_{1,1} = \sin x_1$$

$$L_{2,2} = e^{-x_2}$$

$$H_1 = \begin{bmatrix} -\sin x_1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$H_2 = \begin{bmatrix} 0 & 0 \\ 0 & e^{-x_2} \end{bmatrix}$$

Further analysis on subspace selection criteria can be done for a much larger class of problems than the almost linear set. Specifically, the set of problems which the almost linear class omits are

those which a.) have component function which contain constant terms, b.) have terms of the form $x_i x_j$, $i \neq j$, or c.) have off-diagonal Jacobian components which are linear functions of the unknowns. Of special interest are those problems which have Jacobian matrices whose components are all linear functions of the unknowns. This set of problems, as well as the almost linear set are subsets of what we will call a "nearly linear class" of nonlinear problems. A nearly linear problem is one in which the mapping F satisfies the following definition:

Definition 3.3:

A mapping $F : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ is nearly linear if the off-diagonal elements of the Jacobian associated with F are linear functions of the unknown variables.

It follows directly from the definitions that the nearly linear class of problems completely contains the almost linear class. For a nearly linear problem, the off-diagonal components of the Jacobian are linear functions of the unknowns and the diagonal components are arbitrary. The Hessian matrix for a component function of a nearly linear system has constants for off-diagonal elements and arbitrary diagonal components.

The following are examples of problems which are nearly linear.

Example 3.3. Nearly Linear Problem

$$f_1(x) = x_1^2 - x_2^2$$

$$f_2(x) = x_1 x_2 - x_2^2$$

$$J = \begin{bmatrix} 2x_1 & -2x_2 \\ x_2 & x_1 - 2x_2 \end{bmatrix}$$

$$H_1 = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} \quad H_2 = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix}$$

Example 3.4. Nearly Linear Problem

$$f_1(x) = \sin x_1 - x_2^2$$

$$f_2(x) = 2x_1^2 - e^{-x_2}$$

$$J = \begin{bmatrix} \cos x_1 & -2x_2 \\ 4x_1 & e^{-x_2} \end{bmatrix}$$

$$H_1 = \begin{bmatrix} -\sin x_1 & 0 \\ 0 & -2 \end{bmatrix} \quad H_2 = \begin{bmatrix} 4 & 0 \\ 0 & -e^{-x_2} \end{bmatrix}$$

A comparison of Examples 3.1 and 3.2 to 3.3 and 3.4 shows that the latter exhibit a stronger degree of nonlinearity. This is the motivation for the descriptive word "nearly" since this word implies a greater distance or degree than almost.

The almost linear and the nearly linear problems can be divided into two distinct subclasses based on the form of the Hessian matrices associated with the problem. Nearly linear problems, which include the almost linear problems, can be described as having either constant Hessian matrices or almost constant Hessian matrices. These two bases for classification are defined as follows.

Definition 3.4:

A mapping $F : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ is said to have constant Hessian matrices if all the components of the Jacobian matrix are linear functions of the components of x .

Definition 3.5:

A mapping $F : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ is said to have almost constant Hessian matrices if the off-diagonal elements of the Jacobian are linear and the diagonal elements are arbitrary.

Examples 3.1 and 3.3 have constant Hessian matrices and Examples 3.2 and 3.4 have almost constant Hessian matrices.

The following diagram summarizes the classification of nonlinear problems that will be used in Chapter 4.

Figure 3.1. Nearly linear systems of equations

IV. APPLICATION OF THE PROJECTION METHOD TO NEARLY LINEAR SYSTEMS OF EQUATIONS

In Chapter 2 subspace selection criteria for the nonlinear projection method were discussed for the general nonlinear problem. Chapter 4 will make use of the classification of nonlinear problems discussed in Chapter 3 in order to consider the determination of projection subspaces for particular classes of nonlinear problems. Furthermore, computational alternative for certain classes of nonlinear problems will be discussed.

First, consideration will be given to the use of the ϕ criteria to the solution of nearly linear systems of equations with constant Hessian matrices. The main motivation for using this criteria on these problems is that the information required to compute the criteria can be obtained as a by-product of the information used to update the Jacobian matrix. For a nearly linear problem with constant Hessians, each of the components of the Jacobian matrix can be written as a linear combination of the unknowns. That is,

$$J_{ab} = c_{ab}^1 x_1 + c_{ab}^2 x_2 + \dots + c_{ab}^n x_n \quad 4.1$$

$$a = 1, 2, \dots, n$$

$$b = 1, 2, \dots, n$$

The i th column of the Jacobian can be expressed as the product of a matrix and a vector

$$J_1^k = \begin{bmatrix} c_{11}^1 c_{11}^2 \dots c_{11}^n \\ c_{21}^1 c_{21}^2 \dots c_{21}^n \\ \vdots \\ c_{n1}^1 c_{n1}^2 \dots c_{n1}^n \end{bmatrix} \begin{bmatrix} x_1^k \\ x_2^k \\ \vdots \\ x_n^k \end{bmatrix} = C_1 x^k \quad 4.2$$

Using this notation, a new expression can be developed for Δ_1^k which is done in the following lemma.

Lemma 4.1

For a nearly linear system of equations with constant Hessian matrices

$$\Delta_1^k = C_1(dx^k) \quad 4.3$$

Proof:

1. By definition, $\Delta_1^k = J_1^{k+1} - J_1^k$
2. The result follows directly from an application of 4.2.

$$\begin{aligned} \Delta_1^k &= J_1^{k+1} - J_1^k \\ &= C_1 x^{k+1} - C_1 x^k \\ &= C_1 (x^{k+1} - x^k) \\ &= C_1 dx^k \end{aligned}$$

The use of the φ criteria requires the calculation of (Δ_i^k, Δ_i^k) , $i = 1, 2, \dots, n$ at each iteration step. For nearly linear systems with constant Hessian matrices, Lemma 4.1, gives a convenient way of updating the Jacobian matrix. As a by-product, if the Jacobian matrix were modified using the result of Lemma 4.1, then the information required to determine a projection subspace with the φ criteria would be at hand with very little extra work required.

Note that if the C_i matrices were used to initially evaluate and successively update the Jacobian matrix, no explicit subroutine or subroutine call would be needed to evaluate the Jacobian. There is a trade off however, since constants contained in the C_i matrix as well as any constants contained in the Jacobian components must be read into and maintained by the computer program solving the nonlinear system. For large systems this is no small problem. A comparison between explicit and implicit Jacobian calculation will be made in Chapter 5.

Nearly linear systems of equations with almost constant Hessian matrices can take advantage of the result of Lemma 4.1 in order to update the off-diagonal components of the Jacobian matrix. For these problems, the diagonal components of the Jacobian can be obtained by explicitly calculating n functions for an n -dimensional system. Lemma 4.2 indicates how the data needed for the φ criteria can be obtained for these problems.

Lemma 4.2

Let g_i be an n -dimensional vector whose only nonzero component is in the i th position and is equal to $\partial f_i / \partial x_i(x^{k+1}) - \partial f_i / \partial x_i(x^k)$.

Then, assuming that the i th row of C_i were zero,

$$\Delta_i^k = C_i dx^k + g_i \quad 4.4$$

Proof:

1. Since the nearly linear system has almost constant Hessian matrices, the off-diagonal components can be computed using Lemma 4.1.

That is

$$\Delta_i^{k'} = C_i dx^k$$

2. The g_i vectors represent the change in the diagonal components. The result follows directly:

$$\begin{aligned} \Delta_i^k &= \Delta_i^{k'} + g_i \\ &= C_i dx^k + g_i \end{aligned}$$

As an alternative to the computational scheme described in Lemma 4.2, the Jacobian columns can be calculated explicitly and the differences, Δ_i^k , $i = 1, 2, \dots, n$, can be calculated directly.

That is

$$\Delta_i^k = J_i^{k+1} - J_i^k \quad 4.5$$

A comparison between the use of 4.4 and 4.5 will be made in Chapter 5.

Since the almost linear problems constitute a subset of the nearly linear problems, the data required to calculate the φ criteria and to update the Jacobian can be calculated using the schemes developed in Lemma 4.1 and Lemma 4.2. The computational process is considerably simpler, however, because the almost linear problems have Jacobians with constant off-diagonal components. Thus, in order to update the Jacobian matrix it is only necessary to modify the diagonal components. These changes are reflected in the use of the g_i in Lemma 4.2.

For the almost linear problems, the use of the φ criteria, introduces only minimal additional computational requirements. Since each vector Δ_i^k has only one nonzero component, only n multiplications are needed to calculate (Δ_i^k, Δ_i^k) at each iteration step. Consequently, the determination of a projection subspace for the almost linear problems can be done at a very small cost.

The coefficient criteria developed in Chapter 2 can be used most easily to determine a fixed cycle of projections for nearly linear problems with constant Hessian matrices. For these problems, the coefficient C_a of dx_a^3 , $a = 1, 2, \dots, n$, in 2.14 is constant. It would be expected that the linear approximation for the a th component of x^{k+1} would be better if C_a were small. One example

problem in Chapter 5 will examine alternate cycles of projections and illustrate the effect of taking advantage of the better linear approximations.

Because of the form of the nearly linear problems and subsets of them, the φ criteria and the coefficient criteria are of special interest for determining projection subspaces. These criteria as well as the τ criteria and γ criteria will be compared in Chapter 5 on their ability to determine effective projection subspaces for nearly linear systems of equations.

V. COMPARISONS OF SUBSPACE SELECTION CRITERIA

The purpose of this chapter is to examine and compare the four criteria developed in Chapter II. This is done by inspecting the subspace sequences chosen as well as comparing the computational requirements recorded in CPU seconds for the example problems.

The γ , τ , and φ criteria are compared in Examples 1 through 10. Each criteria was embedded in a similarly constructed computer program in order to compare the resulting rates of convergence. New approximations to the solution vector are successively computed by each routine until the Euclidean norm of $F(x)$ is reduced to a value less than 0.00001. For all algorithms, including the program which uses Newton's method, the linear systems generated are solved by the same routine. This routine solves a linear system using Gaussian elimination with full pivoting.

The first eight example problems were chosen to specifically illustrate the type of nonlinear problems that were discussed in Chapter III. Example problems 5.6 and 5.8 compare different computational algorithms for successively updating the Jacobian matrix that were discussed in Chapter IV.

The γ , τ , and φ criteria each require a tolerance value which can be used to determine if a particular column should or should not be contained in a projection subspace. Unless it is noted otherwise, the tolerances used in the solution process for the first

ten example problems will be calculated by the procedures discussed below.

The γ criteria (or angle-criteria) involves the calculation of the cosine squared of the angles between the residue vector and the columns of the Jacobian. Column i is included in the projection subspace if $\gamma_i = \cos^2 \theta_{rkj_i^k}$ is greater than some Tolerance T . For example if $T = .25$ then no column would be projected on if the residue was within 30° of being perpendicular to it.

The τ criteria uses the Euclidean norm of the residue vector as a tolerance. This tolerance is motivated by the result obtained in Lemma 2.4.

When the φ criteria is used, column i is included or excluded from the projection subspace on the basis of the relative magnitudes of (Δ_i^k, Δ_i^k) and Z_i^k where

$$Z_i^k = 2(J_i^k, \Delta_i^k) + (J_i^k, J_i^k) .$$

Since the (J_i^k, J_i^k) term will, in most cases, tend to dominate Z_i^k , the tolerance $(J_i^k, J_i^k)V$ will be used for the φ criteria. Thus, column i is included in the projection subspace to be used at step $k+1$ when $(\Delta_i^k, \Delta_i^k) > (J_i^k, J_i^k)V$. Note that the φ criteria always use the γ criteria to choose its initial subspace.

In the event that a null subspace is generated, a default subspace made up of all the columns of the Jacobian is used.

A general nonlinear problem is one that cannot be classified by any of the definitions given in Chapter IV. Table 5.1 below specifies the types of the ten example problems.

Table 5.1. Problem types

Example	Type
1	Almost Linear, Constant Hessian
2	Almost Linear, Constant Hessian
3	Almost Linear, Almost Constant Hessian
4	Almost Linear, Almost Constant Hessian
5	Nearly Linear, Constant Hessian
6	Nearly Linear, Constant Hessian
7	Nearly Linear, Almost Constant Hessian
8	Nearly Linear, Almost Constant Hessian
9	General Nonlinear Problem
10	General Nonlinear Problem

The first example is a contrived 3-dimensional problem which is almost linear and has constant Hessian matrices. For this system six of the nine components of the Jacobian matrix are constants.

Example 5.1.

$$f_1(x) = x_1^2 + x_2 + x_3 - 3.0$$

$$f_2(x) = x_1 + 2x_2^2 - x_3 - 3.0$$

$$f_3(x) = x_1 - x_2 + 3x_3^2 - 3.0$$

Table 5.2. Comparisons of criteria on Example 5.1 using:

$$\mathbf{x}^0 = (0, 0, 0)$$

$$\mathbf{r}^0 = (-3, -3, -3)$$

$$||\mathbf{r}^0|| = 5.19615$$

Method	Criteria	Tolerance	Steps	$ \mathbf{r}^f ^a$	CPU Sec.
Newton	-	-	7	0.0000052	0.57
Projection	γ	$T = .25$	9	0.0000010	0.28
Projection	φ	$V = .1$	8	0.0000010	0.39
Projection	τ	-	31	0.0000094	0.50

^aEuclidean norm of the residue vector following the number of steps specified.

For this problem, the use of the γ and φ criteria both required significantly less processor time. A substantial portion of the difference in processor requirements is due to

1. the number of components which are modified, and
2. the sizes of the systems which must be solved to obtain the change components.

Newton's method required seven steps to solve the problem to the desired accuracy. Hence, Newton's method had to solve seven 3-dimensional systems and modify 21 individual components. On the other hand the projection method with the γ and φ criteria required

16 and 17 individual component modifications respectively. Note that of the 17 subspaces chosen by the φ and γ criteria only six of them were three dimensional. Table 5.3 lists the subspaces chosen by the γ and φ criteria.

Table 5.3. Subspaces chosen for Example 5.1 when $x^0 = (0,0,0)$:

Step	γ Criteria	φ Criteria
1	(1)	(1)
2	(1,2,3)	(1)
3	(1,2,3)	(1,2,3)
4	(2,3)	(2,3)
5	(1)	(3)
6	(2,3)	(1,2,3)
7	(1)	(1,2,3)
8	(2,3)	(1,2,3)
9	(1)	-

Example 5.2 is another almost linear problem which has constant Hessian matrices. The problem was originally proposed by Broyden (4) to test his method and has been used by many others including Brown (1) to test their algorithms. The Jacobian for this problem is a banded matrix with at most three nonzero entries in each row.

Example 5.2

$$f_1(x) = - (3 + \alpha x_1)x_1 + 2x_2 - B$$

$$f_i(x) = x_{i-1} - (3 + \alpha x_i)x_i + 2x_{i+1} - B \quad i = 2, 3, \dots, n-1$$

$$f_n(x) = x_{n-1} - (3 + \alpha x_n)x_n - B$$

$$n = 10$$

$$\alpha = .5$$

$$B = 1.0$$

One of the difficulties of comparing iterative methods such as Newton's method or the projection method is that there are infinitely many starting vectors which could be used. Tables 5.4 - 5.9 contain the results of comparing criteria using three different initial starting vectors on Example 5.2.

Table 5.4. Results for Example 5.2 when:

$$x^0 = (-1, 1, 0, -1, 1, 0, -1, 1, 0, -1)$$

$$r^0 = (4.5, -4.5, -2.0, 4.5, -4.5, -2.0, 4.5, -4.5, -2.0, 2.5)$$

$$||r^0|| = 11.82159$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	8	0.0000023	0.59
Projection	γ	$T = .25$	15	0.0000052	0.74
Projection	φ	$V = .1$	9	0.0000037	0.59
Projection	τ	-	18	0.0000043	0.67

Table 5.5. Subspaces chosen for Example 5.2 when:

$$\mathbf{x}^0 = (-1, 1, 0, -1, 1, 0, -1, 1, 0, -1)$$

Step	γ Criteria	φ Criteria
1	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
2	(2)	(1, 2, 3, 4, 5, 6, 7, 8, 9)
3	(3, 5)	(1, 2, 3, 4, 5, 6, 7, 8)
4	(4)	(1, 2, 3, 4, 5, 6)
5	(1)	(2)
6	(2)	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
7	(6)	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
8	(5)	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
9	(3)	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
10	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	
11	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	
12	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	
13	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	
14	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	
15	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	

Table 5.6. Results for Example 5.2 when:

$$\mathbf{x}^0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

$$\mathbf{r}^0 = (-1, -1, -1, -1, -1, -1, -1, -1, -1, -1)$$

$$||\mathbf{r}^0|| = 3.16228$$

Method	Criteria	Tolerance	Steps	$ \mathbf{r}^f $	CPU Sec.
Newton	-	-	6	0.0000025	0.62
Projection	γ	T = .25	6	0.0000052	0.59
Projection	φ	V = .1	7	0.0000040	0.59
Projection	τ	-	31	0.0000055	0.70

Table 5.7. Subspaces chosen for Example 5.2 when:

$$x^0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

Step	γ Criteria	φ Criteria
1	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
2	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
3	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	(2, 3, 4)
4	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
5	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
6	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
7	-	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

Table 5.8. Results for Example 5.2 when:

$$x^0 = (-1, 1, -1, 1, -1, 1, -1, 1, -1, 1)$$

$$r^0 = (4.5, -6.5, 5.5, -6.5, 5.5, -6.5, 5.5, -6.5, 5.5, -4.5)$$

$$||r^0|| = 18.17966$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	9	0.0000039	0.54
Projection	γ	$T = .25$	8	0.0000037	0.33
Projection	φ	$V = .1$	7	0.0000037	0.32
Projection	τ	-	43	0.0000053	0.85

Table 5.9. Subspaces chosen for Example 5.2 when:

$$\mathbf{x}^0 = (-1, 1, -1, 1, -1, 1, -1, 1, -1, 1)$$

Step	γ Criteria	φ Criteria
1	(2,4,6,8)	(2,4,6,8)
2	(10)	(2,4,6,8)
3	(7,8)	(1,2,3,4,5,6,7,8,9,10)
4	(2,3)	(10)
5	(5,10)	(1,2,3,4,5,6,7,8,9,10)
6	(1,2,3,4,5,6,7,8,9,10)	(1,2,3,4,5,6,7,8,9,10)
7	(1,2,3,4,5,6,7,8,9,10)	(1,2,3,4,5,6,7,8,9,10)
8	(1,2,3,4,5,6,7,8,9,10)	

For the first two starting vectors used in solving Example 5.2, Newton's method compared quite favorably with the projection method. However, for $\mathbf{x}^0 = (-1, 1, -1, 1, -1, 1, -1, 1, -1, 1)$ the use of the γ and φ criteria not only required fewer CPU seconds to solve the problem to a similar accuracy but also required fewer iterations. In fact, for this $\mathbf{x}^0$, Newton's method required 90 single component modifications while the projection method using the γ and φ criteria required 41 and 49 single component modifications respectively.

Example 5.3 is a nine dimensional almost linear problem which has almost constant Hessian matrices. The solution of this system is the numerical solution to a boundary value problem suggested by Henrici (15).

Example 5.3.

$$y_0 = 0$$

$$-y_{i-1} + 2y_i - y_{i+1} + 0.01(-2 + \sinh y_i) = 0 \quad i = 1, \dots, 9$$

$$y_{10} = 0$$

For this example, the use of the γ and φ criteria with the projection method are compared to Newton's method. The results for two different values of x^0 are contained in Tables 5.10 - 5.13.

Table 5.10. Results for Example 5.3 with:

$$x^0 = (8, 6, 4, 2, 0, 2, 4, 6, 8)$$

$$r^0 = (24.88478, 1.99713, 0.25290, 0.01627, -4.02, \\ 0.01627, 0.25290, 1.99713, 24.88478)$$

$$||r^0|| = 35.53546$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	6	0.0000002	.64
Projection	γ	T = .25	7	0.0000002	.44
Projection	φ	V = .1	6	0.0000001	.43

Table 5.11. Subspaces chosen for Example 5.3 when:

$$x^0 = (8, 6, 4, 2, 0, 2, 4, 6, 8)$$

Step	γ Criteria	φ Criteria
1	(1, 9)	(1, 9)
2	(1, 2, 8, 9)	(1, 2, 3, 4, 5, 6, 7, 8, 9)
3	(5)	(1, 2, 3, 4, 5, 6, 7, 8, 9)
4	(3, 7)	(1, 2, 3, 4, 5, 6, 7, 8, 9)
5	(1, 2, 3, 4, 5, 6, 7, 8, 9)	(1, 2, 3, 4, 5, 6, 7, 8, 9)
6	(1, 2, 3, 4, 5, 6, 7, 8, 9)	(1, 2, 3, 4, 5, 6, 7, 8, 9)
7	(1, 2, 3, 4, 5, 6, 7, 8, 9)	-

Table 5.12. Results for Example 5.3 with:

$$x^0 = (-8, 6, -4, 2, 0, -2, 4, -6, 8)$$

$$r^0 = (-36.92477, 25.99712, -16.29289,$$

$$8.01627, -0.02, -8.05627, 16.25288,$$

$$-26.03712, 36.88478)$$

$$||r^0|| = 68.82213$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	5	0.0000001	.59
Projection	γ	T = .25	6	0.0000002	.50
Projection	φ	V = .1	4	0.0000001	.37

Table 5.13. Subspaces chosen for Example 5.3 when:

$$\mathbf{x}^0 = (-8, 6, -4, 2, 0, -2, 4, -6, 8)$$

Step	γ Criteria	φ Criteria
1	(1, 2, 8, 9)	(1, 2, 8, 9)
2	(1, 9)	(1, 2, 3, 4, 5, 6, 7, 8, 9)
3	(3, 4, 6, 7)	(1, 2, 3, 4, 5, 6, 7, 8, 9)
4	(1, 2, 3, 4, 5, 6, 7, 8, 9)	(1, 2, 3, 4, 5, 6, 7, 8, 9)
5	(1, 2, 3, 4, 5, 6, 7, 8, 9)	
6	(1, 2, 3, 4, 5, 6, 7, 8, 9)	

Note particularly, the comparison of Newton's method to the projection method with the φ criteria for $\mathbf{x}^0 = (-8, 6, -4, 2, 0, -2, 4, -6, 8)$. In this case, the use of the φ criteria resulted in four projections, the last three of which were nine dimensional projections. Newton's method, on the other hand, started off with a nine dimensional projection and required four additional iterations to achieve the same accuracy.

Example 5.4 is another almost linear problem which has almost constant Hessian matrices. The problem was suggested as an example by Scarborough (31) and was also used by MacEachern (22) to test the one-dimensional nonlinear projection method.

Example 5.4.

$$f_1(\mathbf{x}) = \sin x_1 - x_2 + 1.32$$

$$f_2(\mathbf{x}) = x_1 + \cos x_2 + 0.85$$

Results of comparing Newton's method to the projection method with four different values of x^0 are given in Tables 5.14 - 5.17.

Table 5.14. Results for Example 5.4 with:

$$x^0 = (1.0, 1.0)$$

$$r^0 = (1.16147, 0.39030)$$

$$||r^0|| = 1.2253$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	3	0.0000001	0.39
Projection	γ	$T = .25$	4	0.0000002	0.35
Projection	ϕ	$V = .1$	6	0.0000042	0.30
Projection	τ	-	6	0.0000042	0.40

Table 5.15. Results for Example 5.4 with:

$$x^0 = (-1.0, -1.0)$$

$$r^0 = (1.47853, 2.39030)$$

$$||r^0|| = 2.81062$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	13	.0000001	.52
Projection	γ	$T = .25$	20	.0000047	.49
Projection	ϕ	$V = .1$	14	.0000016	.49
Projection	τ	-	10	.0000044	.45

Table 5.16. Results for Example 5.4 with:

$$x^0 = (0.0, 0.0)$$

$$r^0 = (1.32, 1.85)$$

$$||r^0|| = 2.27264$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	6	.0000001	.44
Projection	γ	$T = .25$	5	.0000002	.44
Projection	ϕ	$V = .1$	6	.0000025	.47
Projection	τ	-	8	.0000038	.49

Table 5.17. Results for Example 5.4 with:

$$x^0 = (1.0, 0.0)$$

$$r^0 = (2.16147, 0.85)$$

$$||r^0|| = 2.32260$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	6	.0000013	.37
Projection	γ	$T = .25$	5	.0000002	.37
Projection	ϕ	$V = .1$	6	.0000012	.50
Projection	τ	-	7	.0000001	.35

For Example 5.4, the projection method with the ϕ criteria worked as well as or better than Newton's method in all four cases.

Example 5.5 is a contrived problem which is nearly linear and has constant Hessian matrices. Results of comparing Newton's method to the projection method with two different values of x^0 are given in Tables 5.18 - 5.21.

Example 5.5.

$$f_1(x) = x_1x_2 - 1.0$$

$$f_2(x) = x_1x_3 - 1.0$$

$$f_3(x) = x_1x_4 - 1.0$$

$$f_4(x) = x_2x_3 - 1.0$$

Table 5.18. Results for Example 5.5 with:

$$x^0 = (2.0, 1.0, -1.0, -2.0)$$

$$r^0 = (1.0, -3.0, -5.0, -2.0)$$

$$||r^0|| = 6.245$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	5	0.0000052	0.34
Projection	γ	$T = .25$	15	0.0000007	0.39
Projection	ϕ	$V = .1$	9	0.0000001	0.39
Projection	τ	-	8	0.0	0.47

Table 5.19. Subspaces chosen for Example 5.5 when:

$$x^0 = (2.0, 1.0, -1.0, -2.0)$$

Step	γ Criteria	ϕ Criteria	τ Criteria
1	(1, 3, 4)	(1, 2, 3)	(1, 3, 4)
2	(3, 4)	(1, 2, 4)	(4)
3	(1, 2)	(1)	(3)
4	(3)	(1, 2, 3, 4)	(1)
5	(4)	(1, 2, 3, 4)	(1, 2, 3, 4)
6	(1)	(1, 2, 3, 4)	(1)
7	(4)	(1, 3)	(1, 2, 3, 4)
8	(3)	(1, 2, 3, 4)	(1, 2, 3, 4)
9	(1)	(1, 2, 3, 4)	
10	(1, 2, 3, 4)		
11	(1, 4)		
12	(2, 3)		
13	(1)		
14	(4)		
15	(1, 2, 3, 4)		

Table 5.20. Results for Example 5.5 with:

$$x^0 = (1.1, 1.2, 1.3, 1.4)$$

$$r^0 = (0.32, 0.43, 0.54, 0.56)$$

$$||r^0|| = 0.94472$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	3	0.0000003	0.35
Projection	γ	$T = .25$	3	0.0000003	0.37
Projection	ϕ	$V = .1$	5	0.0	0.32
Projection	τ	-	5	0.0	0.45

Table 5.21. Subspaces chosen for Example 5.5 when:

$$\mathbf{x}^0 = (1.1, 1.2, 1.3, 1.4)$$

Step	γ Criteria	φ Criteria	τ Criteria
1	(1,2,3,4)	(1,2,3,4)	(1,2,3)
2	(1,2,3,4)	(1,2,3,4)	(1,2,3,4)
3	(1,2,3,4)	(2,3)	(1)
4		(1)	(1,2,3,4)
5		(1,2,3,4)	(1,2,3,4)

In the case where $\mathbf{x}^0 = (2.0, 1.0, -1.0, -2.0)$ for Example 5.5, Newton's method required 20 single component modifications and 5 steps while the projection method with the φ criteria required 27 single component modifications and 15 steps. Despite these differences, the projection method with the φ criteria did not require significantly greater CPU time. This is probably due to the fact that the average dimension of the linear system that was solved at each step of the projection method with the φ criteria was less than two.

Example 5.6 is also a nearly linear problem with constant Hessian matrices.

Example 5.6.

$$f_1(x) = x_1^2 - x_2^2 + x_3^2 - x_2x_3 + 4.0$$

$$f_2(x) = x_1^2 + x_2^2 - x_3^2 - x_1x_3 - 3.0$$

$$f_3(x) = -x_1^2 + x_2^2 + x_3^2 - x_1x_2 - 2.0$$

As part of the comparisons made of this problem, two different computational schemes for the use of the φ criteria were considered. The results of the comparisons are contained in Tables 5.22 and 5.23.

Table 5.22. Results for Example 5.6 with:

$$x^0 = (.1, .2, .3)$$

$$r^0 = (4.0, -3.07, -1.9)$$

$$||r^0|| = 5.38840$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	11	0.0000013	0.42
Projection	γ	$T = .25$	15	0.0000061	0.44
Projection	φ^a	$V = .1$	9	0.0000013	0.39
Projection	φ^b	$V = .1$	9	0.0	0.34
Projection	τ	-	11	0.0000067	0.35

^aUses computational scheme 4.5.

^bUses computational scheme 4.3.

Table 5.23. Results for Example 5.6 with:

$$x^0 = (1.1, 1.2, 1.3)$$

$$r^0 = (3.9, -3.47, -1.4)$$

$$||r^0|| = 5.40471$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	13	0.0	0.34
Projection	γ	$T = .25$	10	0.0000013	0.32
Projection	ϕ^a	$V = .1$	5	0.0000013	0.32
Projection	ϕ^b	$V = .1$	5	0.0000079	0.29
Projection	τ	-	12	0.0000013	0.37

^aUses computational scheme 4.5.

^bUses computational scheme 4.3.

On Example 5.6 the projection method with any of the three criteria performed as well as Newton's method. Also, on this example, there appears to be a slight computational edge when using the ϕ criteria, in computing

$$\Delta_i^k = C_i(dx^k) \text{ rather than } \Delta_i^k = J_i^{k+1} - J_i^k.$$

Example 5.7 is a nearly linear system with almost constant Hessian matrices. This example as well as Example 5.8 have both been suggested and used by Broyden (5) for testing algorithms on sparse nonlinear systems.

Example 5.7.

$$f_i = (k_1 + k_2 x_i^2) x_i + 1.0 - k_3 \sum_{\substack{j=i-r_1 \\ j \neq i}}^{i+r_2} (x_j + x_j^2) \quad i = 1, 2, \dots, n$$

$$n = 8$$

$$k_1 = k_2 = k_3 = 1$$

$$r_1 = r_2 = 1$$

The general form of this problem is interesting because the parameters k_1, k_2, k_3, r_1 , and r_2 permit the nonlinearity of the system and the bandwidth of the Jacobian to be varied. Tables 5.24 - 5.27 contain the results of comparing the projection method to Newton's method for two initial starting vectors.

Table 5.24. Results for Example 5.7 with:

$$x^0 = (1, 2, 2, 1, 1, 2, 2, 1)$$

$$r^0 = (-3, 3, 3, -5, -5, 3, 3, -3)$$

$$||r^0|| = 10.19804$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	17	0.0000069	0.52
Projection	γ	$T = .25$	23	0.0000069	0.43
Projection	ϕ	$V = .1$	22	0.0000071	0.42
Projection	τ	-	130	0.0000064	1.29

Table 5.25. Subspaces chosen for Example 5.7 when:

$$x^0 = (1, 2, 2, 1, 1, 2, 2, 1)$$

Step	γ Criteria	φ Criteria	τ Criteria
1	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)
2	(4, 5)	(1, 2, 3, 4, 5, 6, 7, 8)	(4, 5)
3	(4, 5)	(1, 2, 3, 4, 5, 6, 7, 8)	(4, 5)
4	(4, 5)	(1, 2, 3, 4, 5, 6, 7, 8)	
5	(4, 5)	(1, 2, 3, 4, 5, 6, 7, 8)	(4, 5)
6	(4, 5)	(1, 3, 4, 5, 6, 7, 8)	(2, 4, 5, 7)
7	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 3, 4, 5, 6, 8)	(1, 2, 3, 4, 5, 6, 7, 8)
8	(1, 8)	(3, 4, 5, 6)	(1, 2, 3, 6, 7, 8)
9	(1, 8)	(3, 4, 5, 6)	(1, 2, 3, 6, 7, 8)
10	(3, 6)	(3, 4, 5, 6)	(1, 2, 3, 6, 7, 8)
11	(1, 2, 3, 4, 5, 6, 7, 8)	(4, 5, 6)	(1, 8)
12	(4, 5)	(4, 5)	(1, 2, 7, 8)
13	(4, 5)	(4, 5)	(2, 3, 4, 5, 6, 7)
14	(4, 5)	(5)	(1, 3, 6, 8)
15	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(2, 7)
16	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 3, 6, 8)
17	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 6, 7, 8)	(2, 7)
18	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 7, 8)	(1, 3, 6, 8)
19	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(2, 7)
20	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 3, 6, 8)
21	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(2, 7)
22	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 3, 6, 8)
23	(1, 2, 3, 4, 5, 6, 7, 8)		(2, 7)
24			(1, 3, 6, 8)
25			(2, 7)

Table 5.26. Results for Example 5.7 with:

$$x^0 = (1, 1, 1, 1, 1, 1, 1, 1)$$

$$r^0 = (1, -1, -1, -1, -1, -1, -1, 1)$$

$$||r^0|| = 2.82843$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	500	0.0000164	4.34
Projection	γ	$T = .25$	25	0.0000065	0.89
Projection	ϕ	$V = .1$	34	0.0000069	0.95
Projection	τ	-	26	0.0000028	0.60

Table 5.27. Subspaces chosen for Example 5.7 when:

$$x^0 = (1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0)$$

Step	γ Criteria	φ Criteria	τ Criteria
1	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 7, 8)
2	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 6, 7, 8)	(3, 6)
3	(1, 2, 3, 4, 5, 6, 7, 8)	(2, 7)	(1, 2, 3, 4, 5, 6, 7, 8)
4	(3, 6)	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 3, 4, 5, 6, 8)
5	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 4, 5, 7, 8)	(1, 2, 4, 5, 7, 8)
6	(4, 5)	(1, 2, 4, 5, 7, 8)	(1, 4, 5, 8)
7	(4, 5)	(1, 2, 4, 5, 7, 8)	(3, 6)
8	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 8)	(4, 5)
9	(3, 6)	(1, 2, 3, 4, 5, 6, 7, 8)	(3, 6)
10	(1, 2, 3, 4, 5, 6, 7, 8)	(2, 3, 4, 5, 6, 7)	(4, 5)
11	(4, 5)	(3, 4, 5, 6)	(3, 6)
12	(4, 5)	(1, 2, 3, 4, 5, 6, 7, 8)	(4, 5)
13	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 4, 5, 8)	(3, 6)
14	(1, 2, 3, 4, 5, 6, 7, 8)	(4, 5)	(4, 5)
15	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(3, 6)
16	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(4, 5)
17	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(3, 6)
18	(1, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)
19	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 8)
20	(1, 2, 3, 4, 5, 6, 7, 8)	(1, 2, 3, 4, 5, 6, 7, 8)	(4, 5)
21	(1)	(1, 2, 3, 4, 5, 6, 7, 8)	(3, 6)
22	(8)	(3, 4, 5, 6)	(4, 5)
23	(1, 2, 3, 4, 5, 6, 7, 8)	(3, 4, 5, 6)	(3, 6)
24	(1)	(4, 5)	(4, 5)
25	(1, 2, 3, 4, 5, 6, 7, 8)	(4, 5)	(3, 6)
26		(4, 5)	(1, 2, 3, 4, 5, 6, 7, 8)
27		(4, 5)	
28		(4, 5)	
29		(4, 5)	
30		(1, 2, 3, 4, 5, 6, 7, 8)	
31		(3, 4, 5, 6)	
32		(1, 2, 3, 4, 5, 6, 7, 8)	
33		(1, 2, 3, 4, 5, 6, 7, 8)	
34		(1, 2, 3, 4, 5, 6, 7, 8)	

For $x^0 = (1, 2, 2, 1, 1, 2, 2, 1)$, Newton's method required 136 single component modifications while the projection method with the φ criteria and the projection method with the φ criteria both required 122 single component modifications. Newton's method failed to converge to the specified accuracy in 500 iterations when $x^0 = (1, 1, 1, 1, 1, 1, 1, 1)$. However, the projection method converged successfully to the desired accuracy with any of its subspace selection criteria.

Example 5.8 is also a nearly linear problem with almost constant Hessian matrices.

Example 5.8.

$$f_i = (k_1 + k_2 x_i^2) x_i + 1.0 - k_3 \sum_{\substack{j=i-r_1 \\ j \neq 1 \\ j > 0}}^{i+r_2} (x_j + x_j^2) \quad i = 1, 2, \dots, n$$

$$n = 6$$

$$k_1 = k_2 = k_3 = 1.0$$

$$r_1 = r_2 = 3$$

As part of the comparisons made on this problem two different computational schemes for the use of the φ criteria were considered. The results of comparing Newton's method to the projection method on Example 5.8 are contained in Table 5.28 and Table 5.29.

Table 5.28. Results for Example 5.8 with:

$$x^0 = (0, 0, 0, 0, 0, 0)$$

$$r^0 = (1, 1, 1, 1, 1, 1)$$

$$||r^0|| = 2.44949$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	4	0.0000012	0.42
Projection	γ	$T = .25$	10	0.0000010	0.37
Projection	φ^a	$V = .1$	5	0.0000011	0.42
Projection	φ^b	$V = .1$	5	0.0000014	0.59
Projection	τ	-	37	0.0000086	0.57

Table 5.29. Results for Example 5.8 with:

$$x^0 = (-1, 1, 0, -1, 1, 0)$$

$$r^0 = (-3, 1, -3, -5, 1, -1)$$

$$||r^0|| = 6.78233$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	500	0.0000682	2.47
Projection	γ	$T = .25$	18	0.0000011	0.50
Projection	φ^a	$V = .1$	12	0.0000018	0.52
Projection	φ^b	$V = .1$	12	0.0000005	0.80
Projection	τ	-	24	0.0000015	0.54

^aUses computational scheme 4.5.

^bUses computational scheme 4.4.

For both initial vectors x^0 used to solve Example 5.8, the computational scheme indicated by 4.5 performed significantly better than the computational scheme indicated by 4.4. With $x^0 = (-1, 1, 0, -1, 1, 0)$ Newton's method failed to solve Example 5.8 to the desired accuracy within 500 iterations.

Examples 5.9 and 5.10 are examples of nonlinear problems that cannot be classified by the definitions given in Chapter III. Example 5.9 is a two-dimensional system of transcendental equations that were used by Brown (1) in testing his method. Example 5.10 is a four-dimensional system that was used by White (37).
Example 5.9.

$$f_1(x) = 1/2 \sin(x_1 x_2) - \frac{x_2}{4\pi} - \frac{x_1}{2}$$

$$f_2(x) = (1 - \frac{1}{4\pi})(e^{2x_1} - e) + \frac{ex_2}{\pi} - 2ex_1$$

Table 5.30. Results for Example 5.9 with:

$$x^0 = (0.6, 3.0)$$

$$r^0 = (-0.05181, -0.11223)$$

$$||r^0|| = .12361$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	4	0.0000053	.35
Projection	γ	$T = .25$	15	0.0000086	.44
Projection	ϕ	$V = .1$	7	0.0000021	.34
Projection	τ	-	4	0.0000053	.39

Table 5.31. Results for Example 5.9 with:

$$x^0 = (0.0, 0.0)$$

$$r^0 = (0.0, 1.58154)$$

$$||r^0|| = 1.58154$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	8	0.0000091	.42
Projection	γ	$T = .25$	8	0.0000092	.37
Projection	ϕ	$V = .1$	10	0.0000057	.39
Projection	τ	-	10	0.0000034	.32

Table 5.32. Results for Example 5.9 with:

$$x^0 = (-1.0, -1.0)$$

$$r^0 = (1.00031, 2.19391)$$

$$||r^0|| = 2.41119$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	9	0.0000088	.44
Projection	γ	$T = .25$	18	0.0000023	.42
Projection	ϕ	$V = .1$	10	0.0000064	.44
Projection	τ	-	11	0.0000060	.45

Table 5.33. Results for Example 5.9 with:

$$x^0 = (1.0, 1.0)$$

$$r^0 = (-0.15884, -0.27222)$$

$$||r^0|| = 0.31518$$

Method	Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
Newton	-	-	500	0.0000375	1.25
Projection	γ	$T = .25$	162	0.0000085	.64
Projection	ϕ	$V = .1$	31	0.0000045	.40
Projection	τ	-	500	0.0001057	1.29

Table 5.34. Subspaces chosen for Example 5.9 when:

$$x^0 = (0.6, 3.0)$$

Step	γ Criteria	φ Criteria	τ Criteria
1	(2)	(2)	(1,2)
2	(1)	(1,2)	(1,2)
3	(1,2)	(1)	(1,2)
4	(1,2)	(1)	(1,2)
5	(1)	(1,2)	
6	(2)	(1,2)	
7	(1)	(1,2)	
8	(2)		
9	(1)		
10	(2)		
11	(1)		
12	(2)		
13	(1)		
14	(2)		
15	(1)		

Table 5.35. Subspaces chosen for Example 5.9 when:

$$x^0 = (0.0, 0.0)$$

Steps	γ Criteria	φ Criteria	τ Criteria
1	(1,2)	(1,2)	(1)
2	(1,2)	(1)	(1)
3	(1,2)	(1,2)	(1,2)
4	(1,2)	(1,2)	(1,2)
5	(1,2)	(1,2)	(1,2)
6	(1,2)	(1,2)	(1,2)
7	(1,2)	(1,2)	(1,2)
8	(1,2)	(1,2)	(1,2)
9		(1,2)	(1,2)
10		(1,2)	(1,2)

Table 5.36. Subspaces chosen for Example 5.9 when:

$$\mathbf{x}^0 = (-1.0, -1.0)$$

Steps	γ Criteria	φ Criteria	τ Criteria
1	(1)	(1)	(1)
2	(2)	(1,2)	(1)
3	(1)	(1,2)	(1,2)
4	(2)	(1,2)	(1,2)
5	(1)	(1,2)	(1,2)
6	(2)	(1,2)	(1,2)
7	(1)	(1,2)	(1,2)
8	(2)	(1,2)	(1,2)
9	(1)	(1,2)	(1,2)
10	(2)	(1,2)	(1,2)
11	(1,2)		(1,2)
12	(1,2)		
13	(1,2)		
14	(1,2)		
15	(1,2)		
16	(1,2)		
17	(1,2)		
18	(1,2)		

Table 5.37. Subspaces chosen for Example 5.9 when:

$$\mathbf{x}^0 = (1.0, 1.0)$$

Step	γ Criteria	φ Criteria	τ Criteria
1	(1,2)	(1,2)	(1)
2	(1,2)	(2)	(1,2)
3	(1,2)	(1,2)	(1)
4	(1)	(2)	(1,2)
5	(1,2)	(1,2)	(1)
6	(1,2)	(1,2)	(1,2)
7	(1,2)	(2)	(1)
8	(1)	(1,2)	(1,2)
9	(2)	(1,2)	(1)
10	(1)	(1)	(1,2)
11	(2)	(1,2)	(1)
12	(1)	(1,2)	(1,2)
13	(2)	(2)	(1)
14	(1)	(1,2)	(1,2)
15	(2)	(1,2)	(1)
16	(1)	(1,2)	(1)
17	(2)	(1,2)	(2)
18	(1,2)	(1,2)	(1)
19	(1,2)	(1,2)	(1,2)
20	(2)	(1,2)	(1,2)
21	(1,2)	(1,2)	(1,2)
22	(1)	(1,2)	(1)
23	(2)	(1,2)	(1,2)
24	(1)	(1,2)	(1)
25	(2)	(1,2)	(1,2)
26	(1)	(1,2)	(1)
27	(2)	(1,2)	(1)
28	(1)	(1,2)	(1,2)
29	(2)	(1,2)	(1)
30	(1)	(1,2)	(1,2)
31	(2)	(1,2)	(1)
32	(1,2)		(1,2)
33	(1,2)		(1)
34	(1,2)		(1,2)
35	(1)		(1)

Thus, with any of the four starting vectors used on Example 5.9, the projection method performed as well as or better than Newton's method. In fact, when $x^0 = (1.0, 1.0)$ the difference in performance was significant.

Example 5.10.

$$20x_1 - \cos^2 x_2 + x_3 - \sin x_3 - 37 = 0$$

$$\cos 2x_1 + 20x_2 + \log_{10} (1 + x_4^2) + 5 = 0$$

$$\sin (x_1 + x_2) - x_2 + 19x_3 + \arctan x_3 - 12 = 0$$

$$2 \tanh x_2 + e^{(-2x_3^2 + 0.5)} + 21x_4 = 0$$

Tables 5.38 and 5.39 summarize the comparison of Newton's method and the projection method on Example 5.10.

Table 5.38. Steps to solve Example 5.10.

x^0	Newton	ϕ Criteria	γ Criteria	τ Criteria
(1,1,1,1)	4	5	11	4
(0,0,0,0)	4	5	12	5
(-1,-1,-1,-1)	4	5	11	4
(-1,1,-1,1)	4	6	9	4
(-1,-1,1,1)	4	5	11	8
(-1,-.5,.5,1)	5	5	13	8

Table 5.39. CPU seconds to solve Example 5.10.

x^0	Newton	ϕ Criteria	γ Criteria	τ Criteria
(1,1,1,1)	.34	.47	.30	.35
(0,0,0,0)	.35	.49	.45	.49
(-1,-1,-1,-1)	.37	.28	.37	.42
(-1,1,-1,1)	.32	.37	.40	.27
(-1,-1,1,1)	.34	.40	.39	.40
(-1,-.5,.5,1)	.42	.34	.30	.30

The results contained in Table 5.38 dramatically point out the dependence that the projection method has on the starting vector for Example 5.10. Different starting vectors do result in different numbers of and sequences of projections.

Both the γ criteria and the ϕ criteria require a tolerance to be specified in order to determine projection subspaces. For the first ten examples these tolerances have remained fixed. Table 5.40 summarizes the results of using different values of the tolerance parameters when solving Example 5.8.

Table 5.40. Results for Example 5.8 with:

$$x^0 = (-1, 1, 0, -1, 1, 0)$$

Criteria	Tolerance	Steps	$ r^f $	CPU Sec.
φ	$V = 0.01$	15	0.0000012	0.34
φ	$V = 0.05$	12	0.0000018	0.40
φ	$V = 0.1$	12	0.0000018	0.47
φ	$V = 0.2$	500	0.7439035	2.10
φ	$V = 0.4$	70	0.0000013	0.77
φ	$V = 0.8$	7	0.0000017	0.44
φ	$V = 1.0$	8	0.0000010	0.27
γ	$T = 0.05$	14	0.0000015	0.50
γ	$T = 0.10$	10	0.0000021	0.40
γ	$T = 0.25$	18	0.0000011	0.52
γ	$T = 0.40$	13	0.0000028	0.49
γ	$T = 0.60$	14	0.0000004	0.50
γ	$T = 0.80$	14	0.0000015	0.59
γ	$T = 1.00$	500	0.0000682	6.42

When the φ criteria is used with the projection method, column i is included in the projection subspace when $(\Delta_i^k, \Delta_i^k) > (J_i^k, J_i^k) * V$. Consequently, larger values of V , increase the strictness of the criteria. On Example 5.8, the projection method with the φ criteria was most successful when criteria was the most stringent and when the criteria was the least stringent.

Column i , is included in the projection subspace when the φ criteria is used with the projection method when $\gamma_i = \cos^2 \theta_{rkj_i^k} > T$. So, the smaller the value of T , the more perpendicular J_i^k is allowed to be to r^k and still be included in the projection subspace. As, a

result, the larger values of T make the criteria more strict. On Example 5.8, the projection method with the γ criteria was most successful when the criteria $T \leq 0.60$.

In all of the examples to this point, subspaces have been chosen dynamically based on criteria which were calculated at each step. Now, consider the use of the coefficient criteria as a basis for determining a fixed cycle of projections. The subspaces that are included in a cycle are determined on the basis of an estimate of whether or not individual one-dimensional projections would result in good linear approximations to the single nonzero components of the change vectors. Measures of the goodness of the linear approximations are obtained by examining the coefficients of dx_a^2 and dx_a^3 , $a = 1, 2, \dots, n$, in 2.14. Table 5.41 contains the numerical approximations to these coefficients for Example 5.10 with $x^0 = (1, 1, 1, 1)$.

Table 5.41. Coefficients of dx_a^2 and dx_a^3 for Example 5.10.

a	Coefficient of dx_a^2	Coefficient of dx_a^3
1	- 2.71535	3.71524
2	- 0.55790	3.16669
3	- 11.71503	8.03225
4	- 0.04691	0.04667

One strategy that can be used to take advantage of the coefficient criteria is to project on the one-dimensional subspaces where the linear approximations to the change components are the best and to group the remaining columns into one subspace. If this strategy were used on Example 5, then the subspace sequence for a cycle would be: $\{(4), (1, 2, 3)\}$. Another strategy that can be used with the coefficient criteria is to group into one subspace those columns whose corresponding change components have the best linear approximation. In this case, the subspace sequence for a cycle of projections to be used on Example 5.10 would also be: $\{(4), (1, 2, 3)\}$.

Table 5.42 contains the results of comparing the above cycles to others on Example 5.10.

Table 5.42. Results for Example 5.10 with:

$$x^0 = (1, 1, 1, 1)$$

Cycle	Number of Cycles	$ r^f $	CPU Sec.
(4)(1,2,3)	5	0.0000019	0.35
(2,4)(1,3)	5	0.0000010	0.45
(1,2)(3,4)	5	0.0000010	0.50
(1,4)(2,3)	5	0.0000010	0.40
(1)(2,3,4)	4	0.0000010	0.39
(2)(1,3,4)	5	0.0000010	0.39
(3)(1,2,4)	4	0.0000010	0.37
(1)(3)(2,4)	5	0.0000021	0.49
(2)(4)(1,3)	5	0.0000020	0.47

Conclusions and further remarks concerning the examples used in this chapter are in Chapter VI.

VI. CONCLUSIONS AND FUTURE RESEARCH

A. Conclusions

By allowing a variable number of components of the approximate solution vector to be modified at each iteration step, the nonlinear projection method allows greater flexibility than the total step methods such as Newton's method. In order to take advantage of this flexibility, criteria were developed in Chapter II to dynamically determine quasi-optimal projection subspaces. The nonstationary projection methods that result from using the γ and φ subspace selection criteria are shown through the examples in Chapter V to be in most cases as efficient or more efficient than the unmodified Newton's method. In fact, on most of the examples in Chapter V, the use of the projection method with the γ and the φ criteria required fewer single component modifications as well as fewer CPU seconds to solve the same problem.

The four criteria developed in Chapter II are all independent of the projection subspace dimension. This represents a distinct advantage over those criteria developed for linear systems of equations that depend on the subspace dimension.

The effective results obtained when the projection method is used with the γ and φ criteria are strong evidence supporting the contention that projection subspaces with dimensions greater than three also depend on the angles between the residue vector and the

columns of the Jacobian. It is important to note that the tolerances used by these criteria affects the rate of convergence.

The use of the coefficient criteria requires too much calculation to be used as a criteria for dynamically determining subspaces. However, the coefficient criteria can be used to determine a cycle of projections. As an example in Chapter V illustrates, this use of the projection method does not produce results which are significantly more efficient than Newton's method.

Theorem 2.2 dramatically illustrates the computational inefficiency of trying to maximize Δr^k at each iteration step. However, the approximation to Δr^k that is developed in Chapter II reinforces the fact that Δr^k may or may not be at a maximum when less than an n-dimensional projection is made. An attempt to use part of the expression that represented an approximation to Δr^k resulted in the development of the τ criteria. This criteria, did not in general, work as well as the γ or φ criteria.

The classification of the nonlinear problems that was presented in Chapter III does classify in a reasonable and consistent way a larger subset of the nonlinear problems than had previously been classified. This classification made it possible to consider alternative computational schemes for distinguishable subsets of the nonlinear problems in Chapter IV.

B. Future Research

The following areas are worthy of future consideration:

1. For the examples in Chapter V, the projection method always calculated the Jacobian exactly. In order to make the projection method computationally competitive with modifications to Newton's method, it will be necessary to develop and analyze projection algorithms which take advantage of approximations to the Jacobian. The same analysis might also consider for the projection method, the desirability of only recomputing the Jacobian at fixed or varying intervals of iterations.
2. The linear systems that were generated in the computations for the examples in Chapter V were solved using Gaussian elimination with full pivoting. For the projection method, it may be possible to take advantage of the fact that all the linear systems are symmetric and positive definite by using a linear system solver that is tailored to these characteristics.
3. For many of the examples in Chapter V, the total space was projected on continuously after several iteration steps projected on varying subspaces with dimension less than the dimension of the total space. A criteria needs to be developed which would signal these successive total step projections and stop further subspace selection computation.

4. Since one step of Newton's method is mathematically equivalent to one n -dimensional projection, it might be advantageous to use Newton's method computationally instead of an n -dimensional projection. This kind of algorithm acceleration might significantly reduce the cost of making an n -dimensional projection.

5. This paper develops four criteria for determining projection subspaces. The γ , φ and τ criteria are intended to be used for nonstationary projection methods. The coefficient criteria was developed to determine cycles of projections for stationary projection methods. It may be ultimately more efficient to have an algorithm which uses both stationary and nonstationary techniques. For such an algorithm to become a reality it is necessary to develop some criteria for determining which of the techniques should be used.

6. It may be possible to refine the classification system proposed in Chapter III. This refinement should go along with the further development of specific criteria and computational alternatives for subsets of the nonlinear systems of equations.

7. The γ , φ and τ criteria all require the use of a tolerance value to determine a projection subspace. The data contained in Table 5.39 clearly indicate the difference in the convergence rates that resulted from the use of different criteria tolerances. Some

work should be done to determine if it is possible to find a single optimal criteria for a problem or whether the criteria should be varied during the problem solution.

VII. ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Dr. Roy F. Keller, my Major Professor, for the constant support, patience, guidance, and encouragement that he has given me throughout my research efforts and graduate program. I also wish to express my appreciation to Dr. Robert J. Lambert and to Dr. Arthur E. Oldehoeft. Their stimulating conversations, seminars, and courses have greatly enriched my graduate program. In addition, I would also like to thank Dr. James A Dyer and Dr. Clair G. Maple who served on my graduate committee along with Dr. Keller, Dr. Lambert, and Dr. Oldehoeft.

I am especially grateful to my parents who have instilled in me the ability to set goals and the ethic to work diligently toward those goals. My relatives and friends have been most generous in their understanding and kindness throughout my graduate study. Their support and encouragement has been genuinely appreciated.

Last, but certainly not least, I wish to thank Miss Wanda Lembke for her patience and thorough attention to detail in the typing of this dissertation.

VIII. BIBLIOGRAPHY

1. Kenneth M. Brown, A Quadratically Convergent Method for Solving Simultaneous Nonlinear Equations, Unpublished Ph.D. thesis, Lafayette, Indiana, Library, Purdue University, 1966.
2. Kenneth M. Brown, A Quadratically Convergent Newton-like Method Based Upon Gaussian Elimination, SIAM J. Numer. Anal. 6(4):560-569, December, 1969.
3. Kenneth M. Brown and Samuel D. Conte, The Solution of Simultaneous Nonlinear Equations, A.C.M. National Meeting Proceedings 22:111-114, 1967.
4. C. G. Broyden, A Class of Methods for Solving Nonlinear Simultaneous Equations, Math. Comp. 19:577-593, 1965.
5. C. G. Broyden, The Convergence of an Algorithm for Solving Sparse Nonlinear Systems, Math. Comp. 25:285-294, April, 1971.
6. C. G. Broyden, The Convergence of Single-Rank Quasi-Newton Methods, Math. Comp. 24:365-381, April, 1970.
7. C. G. Broyden, A New Method of Solving Nonlinear Simultaneous Equations, The Computer Journal 12(8):94-99, February, 1969.
8. J. E. Dennis, Jr., On the Convergence of Broyden's Method for Nonlinear Systems of Equations, Math. Comp. 25:559-567, July, 1971.
9. R. Fletcher and M. J. D. Powell, A Rapidly Convergent Descent Method for Minimization, Comput. J. 6:163-168, 1963.
10. L. Fox, An Introduction to Numerical Linear Algebra, Oxford University Press, New York, 1956.
11. Ferdinand Freudenstein and Bernard Roth, Numerical Solution of Systems of Nonlinear Equations, J. Assoc. Comput. Mach. 10:550-556, 1963.
12. D. D. Georg and R. F. Keller, A Computer Program for Solving Nonlinear Systems of Equations, National Technical Information Service, USAEC Report No. IS-3174, 1973.

13. D. D. Georg, Criteria for Ordering Columns in Two-Dimensional Projection Methods, Unpublished M.S. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1973.
14. D. D. Georg and R. F. Keller, A Projection Algorithm for Solving Nonlinear Systems of Equations, Unpublished paper presented at the Mathematical Software Conference II at Purdue University, May, 1974, National Technical Information Service, USAEC Report No. IS-M-16, 1974.
15. Peter Henrici, Elements of Numerical Analysis, John Wiley and Sons, New York, 1964.
16. Alston Householder, Principles of Numerical Analysis, McGraw-Hill Book Company, Inc., New York, N.Y., 1953.
17. Alston Householder, The Theory of Matrices in Numerical Analysis, Blaisdell Publishing Company, New York, 1964.
18. R. F. Keller and I-Ming Shen, Acceleration of Projection Methods for Solving Systems of Linear Algebraic Equations, Unpublished paper presented at the SIAM National Meeting, Denver, Colorado, June, 1970, Mimeo, Ames, Iowa, Iowa State University of Science and Technology, Department of Computer Science, 1970.
19. R. F. Keller, H. D. Pyron, and I-Ming Shen, On Determining an Optimal Two-Dimensional Projection Method, National Technical Information Service, USAEC Report No. IS-2755, 1970.
20. R. F. Keller, A Single Step Gradient Method for Solving Systems of Linear Equations, University of Missouri at Columbia, Math. Sci. Tech. Report No. 8, 1964.
21. David G. Luenberger, The Conjugate Residual Method for Constrained Minimization Problems, SIAM J. Numer. Anal. 7(3):390-398, September, 1970.
22. A. MacEachern, A Generalized Projection Method for Systems of Nonlinear Equations, Unpublished Ph.D. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1971.
23. A. MacEachern and R. F. Keller, A Projection Method for Solving Nonlinear Systems of Equations, National Technical Information Service, USAEC Report No. IS-3040, 1972.

24. O. G. Mancino, Resolution by Iteration of Some Nonlinear Systems, J. ACM 14(2):171-190, June, 1967.
25. J. M. Ortega and W. C. Rheinboldt, A General Convergence Result for Unconstrained Minimization Methods, SIAM J. Numer. Anal. 9(1):40-43, March, 1972.
26. J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.
27. A. M. Ostrowski, Solutions of Equations and Systems of Equations, Academic Press, New York, 1966.
28. Sam Perlis, Theory of Matrices, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts, 1952.
29. M. J. D. Powell, A Method for Minimizing a Sum of Squares of Nonlinear Functions Without Calculating Derivatives, The Computer Journal 7:303-307, 1965.
30. H. D. Pyron, A Nonstationary Two-Dimensional Acceleration for the One-Dimensional Projection Method, Unpublished Ph.D. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1970.
31. James B. Scarborough, Numerical Mathematical Analysis, Fifth ed., The John Hopkins Press, Baltimore, Maryland, 1962.
32. L. K. Schubert, Modification of a Quasi-Newton Method for Nonlinear Equations with a Sparse Jacobian, Math. Comp. 25:27-30, 1970.
33. J. A. Schmitz, D. D. Georg, and R. F. Keller, Interactive Computer Programs for Linear and Nonlinear Projection Methods, National Technical Information Service, USAEC Report No. IS-3179, 1973.
34. I-Ming Shen, Acceleration of the Projection Method for Solving Systems of Linear Equations, Unpublished Ph.D. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1970.

35. G. B. Thomas, Calculus and Analytic Geometry, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1960.
36. M. Tokko, Optimal Three-Dimensional Projection Method for Solving Linear Algebraic Equations, Unpublished Ph.D. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1972.
37. M. White, A Class of Projection Methods for Solving Systems of n Nonlinear Equations in n Unknowns Allowing Projections of Order One Through n , Unpublished Ph.D. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1971.
38. Frank J. Zeleznik, Quasi-Newton Methods for Nonlinear Equations, J. Assoc. Comput. Mach. 15(2):265-271, 1968.