TECHNIQUES FOR ACCELERATING ITERATIVE METHODS FOR THE SOLUTION OF MATHEMATICAL PROBLEMS

By

STEVEN RUSSELL CAPEHART

Bachelor of Arts
Arkansas Polytechnic College
Russellville, Arkansas
1971

Master of Arts
University of Arkansas
Fayetteville, Arkansas
1975

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF EDUCATION
July, 1989

TECHNIQUES FOR ACCELERATING ITERATIVE METHODS FOR THE SOLUTION OF MATHEMATICAL PROBLEMS

Thesis Approved:

John P Chandler
Thesis Advisor

Marvin L Kemer

Alan V. Nolle

John Jobe

John E Gardiner

Norman N. Durham
Dean of the Graduate College

ACKNOWLEDGEMENTS

This thesis discusses the solving of mathematical problems by accelerating sequences generated by a numerically derived itertive scheme. Acceleration methods are applied to these sequences to attempt to speed up their convergence or to force the convergence of a divergent sequence. The thesis presents the derivation of some of these acceleration methods and then compares the methods theoretically and numerically. Numerical results are presented in the form of tables and/or graphs.

I would like to express my deepest appreciation to the many people who have played major roles in my completing this project. First and foremost, I thank my Lord and Savior, Jesus Christ. Without the extra strength that He has provided me at critical times during these three years, I would not be at this point in the program. Second, I thank Professor Chandler, my thesis advisor. His continual support, encouragement, and assistance have been vital. In addition, he was willing to be my advisor even though I was from a different department than his. Thank you, Dr. Chandler, your help will always be greatly appreciated. The other members of my committee: Professors Gardiner, Jobe, Keener, and Noell; each provided important pieces in my finishing what seemed like one big "jig-saw" puzzle. Their willingness to help in any way, at any time has demonstrated their "true" character.

The list of Oklahoma State University professors to thank does not stop at my committee. Professors Alspach, Burchard, Conrey, Haack, Powell, Webster, and Wolfe provided help and encouragement. I will never be able to express my deepest thanks to my fellow graduate students. At the top of the list are Ivan and Nora Schukei. The many long hours of study and support we had together will never be forgotten. In addition, a special thanks goes to Debbie Carment. Though we

worked together only one semester, her help that first semester means a great deal to me. I also thank Ken Harrelson for taking away from his busy time as student and TA in assisting me with the typing of this thesis.

There are many other people I need to thank for their prayer support. First, my parents. They have always believed in me and have continually lifted me up in prayer. Next, I thank my brother and the many friends in Arkansas, Colorado, and at Sunnybrook Christian Church who have continued to support me with telephone calls, cards, and words of encouragement.

I thank my two sons, Stony and Shay. They have understood why I was not available for many of their activities and not home many of the nights. Their love is forever. I have saved the best for last. How can I even begin to say thank you to my wife, Betty Lue. She has endured much during these three years and I know that her love for me is stronger now than it was when I began this adventure. The fire of trials will purify the gold, and my wife is golden. Thank you, Betty Lue, for your love and support.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

NOMENCLATURE

A	Matrix
A^+	Generalized inverse of A
A^*	Complex conjugate transpose of A
$\vec{a}_{i,j}$	Inner product of $\vec{a}_i$ and $\vec{a}_j$
det(A)	Determinant of A
$\vec{\epsilon}_k^n$	Element of Wynn's epsilon arrangement corresponding to the k^{th} column and the n^{th} diagonal
e_n	The nth error vector: $\vec{x}_{n+1} - \vec{s}$
$\mathrm{EXT}(x_p, \ldots, x_k)$	Extrapolation applied to terms x_p through x_k
F	Operator on x
G	Iteration function
I_k	Identity matrix of rank k
$\vec{s}$	Solution of the problem or the limit of the sequence $\{\vec{x}_n\}$
$(1,...,1)^T$	Transpose of (1,...,1)
$[t_p, \ldots, t_k]$	Matrix with t_p through t_k as first row
$\vec{u}_n$	The n^{th} difference vector: $\vec{x}_{n+1} - \vec{x}_n$
$\vec{v}_n$	The n^{th} second difference vector: $\vec{u}_{n+1} - \vec{u}_n$
$\vec{x}_n$	The n^{th} term of the sequence $\{\vec{x}_n\}$
$\{\vec{x}_n\}$	Sequence $\{\vec{x}_0, \vec{x}_1, \ldots\}$
$(\vec{x}, \vec{y})$	Inner product of $\vec{x}$ and $\vec{y}$

CHAPTER I

INTRODUCTION

Solving large scale mathematical problems has been and will always be a vital concern in many areas of life. This concern has led to an increased popularity and explosive growth of numerical analysis. Numerical analysis is the theory of constructing methods for approximating, in an efficient manner, the solutions to mathematical problems. Existing methods are grouped into two types: direct methods and iterative methods (Golub and Van Loan, 1983). Direct methods will determine a solution exactly, up to the precision of accuracy of the computer, in a finite number of steps. The goal of iterative methods is to start with an initial guess of the solution and then improve the guess by the use of an updating step which is called an iteration. A succession of iterations will produce a sequence of scalars - real or complex numbers - or vectors, as appropriate, which converges to a limit, the solution. If the limit is not obtained in a finite number of iterations, it may be approximated to a desired accuracy after a finite number of iterations. This study will be concerned only with iterative methods.

Before the age of digital computers, methods requiring a large amount of computational effort were impractical, if not totally unreasonable, to apply. However, we now have the high speed computers available that allow us to solve large scale problems to a high degree of accuracy. But we still have a problem: computer time costs. Therefore, it was essential that the subject area of numerical analysis respond to the overwhelming need for faster computation and reduced computer time. The result was the development of sophisticated numerical methods called acceleration

techniques. These techniques use the original sequence to produce a new sequence which converges to the same limit as the original one, only faster.

An acceleration technique is presented in the form of what is called an extrapolation algorithm (Skelboe, 1980). An extrapolation algorithm determines an element of the second sequence from some desired number, say k, of consecutive elements of the original sequence. In other words, assume our original sequence is

$$\{x_n\} = \{x_0, x_1, x_2, \ldots\}.$$

We then extrapolate (compute) a second sequence, $\{y_n\}$, by applying the algorithm to the k terms and represent an extrapolation by the notation

$$y_{p+k-1} = \mathrm{EXT}(x_p, x_{p+1}, x_{p+2}, \ldots, x_{p+k-1}), \quad p = 0, 1, 2, \ldots$$

The purpose of this study is to compare several of these acceleration techniques. The techniques studied will include Wynn's (1956) epsilon algorithm, the modified epsilon algorithm (Cheng and Hafez, 1959), Cabay and Jackson's (1976) Minimal Polynomial Extrapolation method (MPE), a matrix Full Rank Extrapolation (FRE) originally developed by Henrici (1964) and modified to a Reduced Rank Extrapolation (RRE) by Eddy (1979), and Anderson's (1965) generalized secant methods. In addition, a method derived by Aitken (1936-37) for scalars and modified for vectors by Jennings (1971) will be studied in the early chapters to help establish notation and to set the pattern of how extrapolation algorithms will be developed.

The focus of this study will be on acceleration techniques for finding the solution of the problem

$$F(x) = 0, \tag{1}$$

where F is an operator on a scalar x or on a m-dimensional vector $\vec{x}$. The theory will be developed in the beginning with the use of scalars; however, all theory will

quickly be related to vectors and by the end of the study our only concern will be solving (1) operating on m-dimensional vectors.

For convenience, Equation (1) will be rewritten as

$$x = G(x), \tag{2}$$

where $G(x) = x - F(x)H(x)$ for any H such that, if s is a solution of (1), then $H(s)$ is finite and nonzero (Traub, 1964). If s is a solution of (1), then s is also a solution of (2) and we have converted our problem to determining a fixed point s of Equation (2). A sequence $\{x_n\}$ is produced from Equation (2) by the iterative scheme

$$x_{n+1} = G(x_n), \quad n = 0, 1, \ldots, \tag{3}$$

where $\lim_{n\to\infty} x_n = s$ for a convergent sequence $\{x_n\}$. The function G is referred to as an iteration function. An iteration is, therefore, defined as computing x_{n+1} by Equation (3) for some non-negative integer n. An acceleration technique will produce a new sequence $\{y_n\}$ which also converges to s, but faster than the original sequence $\{x_n\}$.

Sequences are generated several different ways. For scalars, there exist the well-studied sequences produced by the partial sums of a series. Perhaps the best known method for producing vector sequences is numerically solving the system of linear equations

$$\vec{x} = A\vec{x} + \vec{b}, \tag{4}$$

where A is an m x m matrix, $\vec{x}$ and $\vec{b}$ are m-dimensional column vectors with $\vec{b}$ constant. Hence, the basic iteration equation becomes

$$\vec{x}_{n+1} = A\vec{x}_n + \vec{b} = G(\vec{x}_n). \tag{5}$$

Vector sequences can also be generated by nonlinear problems: integral equations and ordinary differential equations. Examples and test problems of varied form will be presented throughout the thesis.

Equations (4) and (5) will be used extensively in later chapters in the development of the acceleration methods. However, in practice, problems of this type would normally be solved by other methods that will not be discussed in this paper. The small linear problems, where the value of m in Equation (4) is small, would be solved by a direct method called Gaussian elimination. There also exist efficient special methods that will solve sparse linear problems of large dimension. However, linear problems are quite useful for designing extrapolation models and testing the algorithms. The small nonlinear problems would be solved, in practice, by Newton's method or an optimization method, for example, a quasi-Newton method. These problems are most useful as test problems where the limit can only be estimated and not found exactly. The domain of problems that will be most practical for the methods presented in this study is medium-to-large-sized nonlinear problems. Fox (1965), Ortega and Rheinboldt (1970), Varga (1962), and Young (1971) provide further background material on these other methods.

CHAPTER II

MOTIVATIONAL EXAMPLES

Using Acceleration Techniques

Before developing the acceleration techniques, this chapter will be used to help motivate interest. The motivation will come in two parts. First, it will be shown how an acceleration technique can be used to find the limit of a sequence. It will then be shown how the eigenvalues of the matrix A of the iteration Equation (5) effect the convergence of the vector sequence $\{\vec{x}_n\}$.

To start with, consider the scalar problem

$$F(x) = x - e^{-0.5x} = 0, \tag{6}$$

with a solution of 0.7034674 for seven place accuracy. Figure 1 (page 6) is the graph of $F(x)$ on the interval $[0,1]$. The solution is found by solving the fixed point solution of (2), where

$$x = e^{-0.5x} = G(x).$$

This will give an iteration equation of

$$x_{n+1} = e^{-0.5x_n} = G(x_n). \tag{7}$$

Figure 2 (page 6) shows the graph of $G(x)$ and the equation $y = x$ on the interval [0,1]. In addition, the figure illustrates graphically the convergence of the fixed point problem (Conte and de Boor, 1980) with an initial value of $x_0 = 0.5$. If a solution of Equation (6) exists, then it will be the intersection of $y = x$ and $y = G(x)$. To

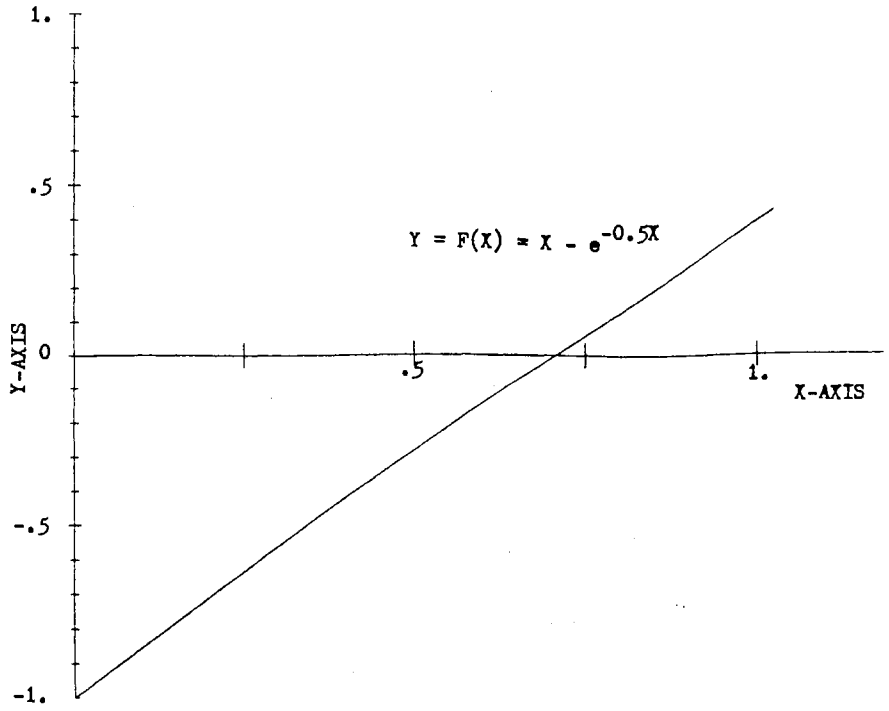

Figure 1. Graph of $F(x) = x - e^{-0.5x}$

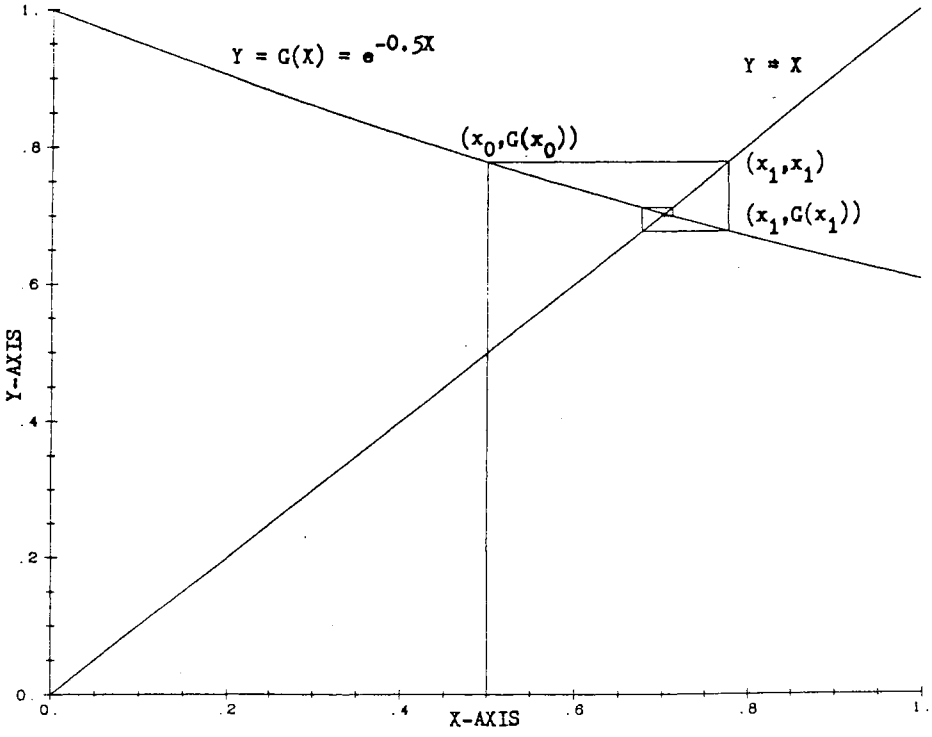

Figure 2. Graphs of $y = G(x) = e^{-0.5x}$ and $y = x$

find this point graphically, it is known that the point (x_{n-1}, x_n) must lie on the graph of $G(x)$. The next point, (x_n, x_{n+1}), can be found by drawing a line through (x_{n-1}, x_n) parallel to the x axis. This line intersects the graph of $y = x$ at (x_n, x_n). Now draw through this point the line parallel to the y axis. The intersection of this line and the graph $y = G(x)$ will be the point $(x_n, G(x_n))$, or (x_n, x_{n+1}). Continuing this procedure will eventually lead us to the desired solution.

It is a fact, however, that not all fixed point iterations converge. It often occurs that a mathematical problem has a unique and reasonable solution, but when a numerical algorithm is devised to solve the problem, like the fixed point iteration scheme, the resulting sequence of approximations diverges. However, there are certain criteria that will insure convergence. It is not the intent of this paper to discuss such criteria, but one can find good discussion for the linear case in texts written by Henrici (1964) or by Conte and de Boor (1980). We will consider divergent sequences after determining the solution of Equation (6).

Using (7) and $x_0 = 0.5$, a sequence $\{x_n\}$ is derived whose limit is the solution. As shown in Table 1 (page 8), it requires 16 iterations before the sequence converges to the correct value with seven decimal place accuracy. The correct digits for each iterate are underlined. In addition, Table 1 gives the differences between consecutive iterates and also the ratios of consecutive differences, denoted by $u_n = x_{n+1} - x_n$ and $r_n = u_n/u_{n-1}$, respectively. This information will help develop an acceleration technique to apply to Equation (7) for comparison (Atkinson, 1972).

A closer look at the first few ratios of Table 1 shows that they seem to be converging to a value of approximately -0.3517. The ratios for $n \geq 9$ no longer converge due to rounding errors in determining x_n and u_n to seven places. Assuming that the ratio is approximately constant after the fifth iteration, estimates for u_5 through u_{11}, denoted by $\tilde{u}_n, n = 5, \ldots, 11$, can be made by $\tilde{u}_{i+1} = \tilde{u}_i(\tilde{r}), i = 4, \ldots, 10$, where $\tilde{r} = r_4 = -0.3512072$ and $\tilde{u}_4 = u_4$. These estimates are given in Table 2 (page 9)

along with the actual values from Table 1.

TABLE 1

ITERATED VALUES FOR EQUATION (7)

n	x_n	$u = x_{n+1} - x_n$	$r_n = u_n/u_{n-1}$
0	0.5000000	0.2788008	
1	0.7788008	−0.1013378	−0.3634773
2	0.6774630	0.0352108	−0.3474596
3	0.7126738	−0.0124371	−0.3532183
4	0.7002367	0.0043680	−0.3512072
5	0.7046047	−0.0015372	−0.3519230
6	0.7030675	0.0005406	−0.3516783
7	0.7036081	−0.0001902	−0.3518312
8	0.7034179	0.0000669	−0.3517350
9	0.7034848	−0.0000235	−0.3512705
10	0.7034613	0.0000083	−0.3531914
11	0.7034696	−0.0000029	−0.3493975
12	0.7034667	0.0000010	−0.3448275
13	0.7034677	−0.0000004	−0.4000000
14	0.7034673	0.0000002	−0.5000000
15	0.7034675	−0.0000001	−0.5000000
16	0.7034674	0.0000000	0.0000000
17	0.7034674		

TABLE 2

ESTIMATED AND ACTUAL VALUES FOR CONSECUTIVE DIFFERENCES FOR EQUATION (7)

n	$\tilde{u}_n = \tilde{u}_{n-1}(\tilde{r})$	$u_n(r)$
5	$(0.0043680)\tilde{r} = -0.0015340$	-0.0015372
6	$(-0.0015340)\tilde{r} = 0.0005388$	0.0005406
7	$(0.0005388)\tilde{r} = -0.0001892$	-0.0001902
8	$(-0.0001892)\tilde{r} = 0.0000664$	0.0000669
9	$(0.0000664)\tilde{r} = -0.0000233$	-0.0000235
10	$(-0.0000233)\tilde{r} = 0.0000082$	0.0000083
11	$(0.0000082)\tilde{r} = -0.0000029$	-0.0000029

It is true that

$$\begin{aligned} x_{11} &= x_5 + (x_6 - x_5) + (x_7 - x_6) + \cdots + (x_{11} - x_{10}) \\ &= x_5 + u_5 + u_6 + u_7 + u_8 + u_9 + u_{10}. \end{aligned}$$

Therefore, we can estimate x_{11} by

$$\begin{aligned} x'_{11} &\approx 0.7046047 - 0.001534 + 0.0005388 - 0.0001892 + \\ &\quad 0.0000664 - 0.0000233 + 0.0000082 - 0.0000029 \\ &= 0.7034687. \end{aligned}$$

Hence, using only information obtained from x_3, x_4, and x_5; x'_{11} has been determined more accurately than the actual iterated x_{11}. Even though an estimate for x_k, for some positive integer k, may not always be more accurate than the iterated x_k, the estimate, x'_k, will usually be a better approximation of the solution than x_p, for some p, $p \leq k$.

Generalizing the concept, assume x_{n-2}, x_{n-1}, and x_n have been computed. Also assume r_n is approximately constant with $\tilde{r} = r_{n-1} = u_{n-1}/u_{n-2}$. Then the solution can be approximated by viewing it as the limit, x_∞, of the sequence:

$$\begin{aligned} x_\infty &\approx x_n + u_n + u_{n+1} + u_{n+2} + \cdots \\ &\approx x_n + u_{n-1}\tilde{r} + u_n\tilde{r} + u_{n+1}\tilde{r} + \cdots \\ &\approx x_n + u_{n-1}\tilde{r} + u_{n-1}\tilde{r}^2 + u_{n-1}\tilde{r}^3 + \cdots \\ &= x_n + u_{n-1}(\tilde{r} + \tilde{r}^2 + \tilde{r}^3 + \cdots) \\ &= x_n + u_{n-1}\left(\tilde{r}/(1-\tilde{r})\right), \quad |\tilde{r}| < 1, \end{aligned} \tag{8}$$

since the series in parentheses is a geometric series. Substituting $\tilde{r} = u_{n-1}/u_{n-2}$ and simplifying give

$$\begin{aligned} x_\infty &\approx x_n - u_{n-1}\left([u_{n-1}/u_{n-2}]/[(u_{n-1}/u_{n-2}) - 1]\right) \\ &= x_n - (u_{n-1})^2/(u_{n-1} - u_{n-2}) \\ &= x_n - \frac{(x_n - x_{n-1})^2}{(x_n - x_{n-1}) - (x_{n-1} - x_{n-2})}. \end{aligned} \tag{9}$$

This formula is Aitken's Δ^2 formula (1936-37) for accelerating a convergent sequence. More information will be discussed concerning Aitken's formula in Chapter IV. However, based on the above development, one may assume that if Aitken's method is applied to Equation (7) after the n^{th} iteration, a better estimate of the solution can often be found with no additional iterations.

Aitken's formula can be used as a sequence generator to derive a new sequence $\{y_n\}$. The new sequence is generated by Formula (9) rewritten as

$$y_{n+2} = x_{n+2} - \frac{(x_{n+2} - x_{n+1})^2}{x_{n+2} - 2x_{n+1} + x_n}, \quad n = 0, 1, \ldots \tag{10}$$

In fact, (10) can be used to produce even a third sequence $\{z_n\}$ from $\{y_n\}$. Applying this technique to the sequence derived from Equation (7), the resulting three

sequences are shown in Table 3. One can see that the solution, to seven place accuracy, is found after only five iterations. Generating sequences in this fashion is called a static model and Table 3 is referred to as a static display.

TABLE 3

AITKEN'S COMPUTED VALUES FOR EQUATION (7)

n	x_n	y_n	z_n
0	0.5000000		
1	0.7788008		
2	0.6774630	0.7044777	
3	0.7126738	0.7035942	
4	0.7002367	0.7034830	0.7034669
5	0.7046047	0.7034693	0.7034674

Equation (10) is written differently than the way it is given in many texts. The difference is that the new term in $\{y_n\}$ is denoted by the subscript $n+2$ instead of its usual subscript n. The reason for this change is that if we want to compare terms of the two sequences for error, that is, their closeness to the exact answer, then to compare the n^{th} term of $\{y_n\}$, call it y for the moment, to the n^{th} term of $\{x_n\}$, call it x, is unfair. This y cannot be computed until x_{n+2} is available. If the original sequence converges, not only should y have a smaller error than x, but so should the next two terms following x in the original sequence. Therefore, what is valuable is to compare y with x_{n+2} in error. Hence, the n^{th} term of $\{y_n\}$ is referred

to as y_{n+2} so that when comparisons are made, the elements being compared will have the same subscripts.

Now consider the geometric series

$$1+2+4+\ldots$$

Then the sequence of partial sums is

$$1,3,7,\ldots, \quad \text{or} \quad x_n = -1+2^n, \quad n=1,2,\ldots \tag{11}$$

If Aitken's method is applied to Sequence (11), the resulting sequence is

$$-1,-1,-1,\ldots$$

Hence, the acceleration technique determined the value -1 as the "limit" of a divergent sequence. Shanks (1955) says the divergent sequence is "diverging from" -1 and calls the value -1 the "antilimit" of (11).

As stated earlier, iterated sequences of some mathematical problems do not converge. However, if the sequence has an antilimit, the antilimit is usually unique, equal to the solution of the problem, and can usually be found by applying an acceleration method to the original divergent sequence (Sidi, Ford, and Smith, 1986). There are cases known where this is not true (Shanks, 1955); therefore, one must be careful to ensure that the computed antilimit is, in fact, the solution. An example where Aitken's method gives erroneous results will be discussed in Chapter IV.

The early leader in the use of divergent sequences to derive correct answers was Euler (1707-1783). He maintained that if a function f gave rise to a series, then the "sum" of the series should be $f(x)$ for any x, even when the series diverged. Even though his definition of the word "sum" extended the normal definition of a sum of a series, he felt quite comfortable with it since "the new definition ... coincides with the ordinary meaning when a series converges..." (Bromwich, 1926, p. 322).

One may look at antilimits as the assigning of a number to a divergent sequence. This has been around for quite some time in mathematics in the form of summability methods. There exists several methods of summability. Excellent material on the subject matter may be found in texts written by Hardy (1949), Lubkin (1952), Moore (1938), and Zygmund (1959).

Illustrations of Convergence in Two Dimensions

Let us now look at the second motivational factor of this chapter. Consider Equation (4) with dimension $m = 2$. Define A and $\vec{b}$ by

$$A = \begin{bmatrix} 1.0 & 0.1 \\ -0.5 & 0.4 \end{bmatrix} \quad \text{and} \quad \vec{b} = \begin{bmatrix} 1.2 \\ -2.0 \end{bmatrix}. \tag{12}$$

Using Equation (5) and the initial vector (1,1), a convergent sequence $\{\vec{x}_n\}$ of two-dimensional vectors is generated that converges to the solution $\vec{s} = (10.4, -12.0)^T$. Table 4 shows the first six terms of the sequence. The path of convergence of $\{\vec{x}_n\}$ to $\vec{s}$ is shown in Figure 3 (page 14). The graph can be considered as a trajectory of a moving particle originating at $\vec{x}_0$ and terminating at the solution.

TABLE 4

FIRST SIX TERMS OF THE GENERATED

SEQUENCE OF PROBLEM (12)

n	x_n	n	x_n
0	(1.000000, 1.000000)	3	(4.091000, −5.241000)
1	(2.300000, −2.100000)	4	(4.766900, −6.141900)
2	(3.290000, −3.990000)	5	(5.352710, −6.840210)

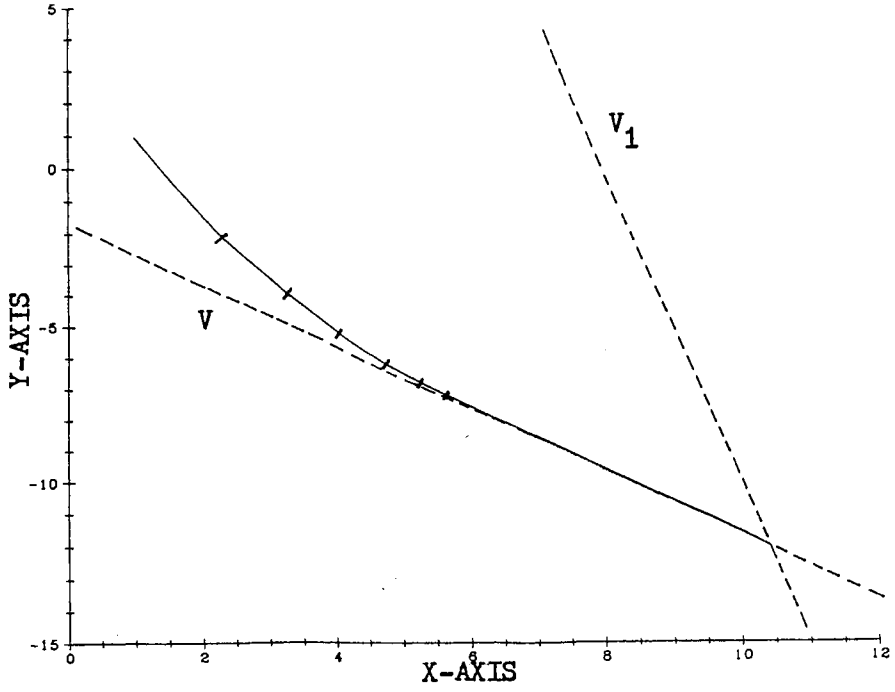

Figure 3. Trajectory for Eigenvalues 0.9 and 0.5

The matrix A has eigenvalues $q = 0.9$ and $q_1 = 0.5$ with associated eigenvectors $\vec{v} = (1, -1)^T$ and $\vec{v}_1 = (1, -5)^T$, respectively, which are also shown in Figure 3. The trajectory of $\{\vec{x}_n\}$ shows that as $\vec{x}_n$ converges to $\vec{s}$, the convergence is asymptotic along the vector $\vec{v}$. Since $\vec{v}$ and $\vec{v}_1$ are linearly independent, the vector $\vec{x}_0 - \vec{s}$ can be written as a linear combination of these vectors. Hence,

$$\begin{aligned} \vec{x}_0 - \vec{s} &= (-9.4, 13.0)^T = -8.5(1, -1)^T - 0.9(1, -5)^T. \\ &= -8.5\vec{v} - 0.9\vec{v}_1. \end{aligned}$$

Using the fact that $\vec{v}$ and $\vec{v}_1$ are eigenvectors, it follows that

$$\begin{aligned} \vec{x}_1 - \vec{s} &= A(\vec{x}_0 - \vec{s}) = -8.5(0.9)\vec{v} - 0.9(0.5)\vec{v}_1 \\ &= (2.3, -2.1)^T, \quad \text{and} \\ \vec{x}_2 - \vec{s} &= A(\vec{x}_1 - \vec{s}) = -8.5(0.9)^2\vec{v} - 0.9(0.5)^2\vec{v}_1 \end{aligned}$$

$$= (3.29, -3.99)^T.$$

Continuing, it can be seen that the sequence $\{\vec{x}_n\}$ can be generated by the equation

$$\vec{x}_n = \vec{s} + aq^n\vec{v} + a_1q_1^n\vec{v}_1,$$

where $a = -8.5$ and $a_1 = -0.9$. Assume that $n = 100$. Then

$$\begin{aligned} \vec{x}_{100} &= \vec{s} + a(0.9)^{100}\vec{v} + a_1(0.5)^{100}\vec{v}_1 \\ &\approx \vec{s} + (2.66 \text{ x } 10^{-5})a\vec{v} + (7.89 \text{ x } 10^{-31})a_1\vec{v}_1. \end{aligned}$$

Since the last term is approximately zero, $\vec{x}_{100}$ is primarily the sum of $\vec{s}$ and a multiple of the eigenvector $\vec{v}$. So, as n increases, the convergence of the sequence is controlled by the dominant eigenvalue, 0.9, resulting in the asymptotic convergence along $\vec{v}$. In addition, for n greater than 200, the coefficient of $\vec{v}$ is less than $1 \text{ x } 10^{-8}$ which implies $\{\vec{x}_n\}$ has converged to $\vec{s}$ with seven place accuracy.

The problem can be generalized for dimension m where $q_1, q_2, \ldots, q_m$ are the eigenvalues of A with corresponding eigenvectors $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_m$; and with the assumption that unity is not an eigenvalue of A so that the problem has the unique solution $\vec{s}$. Then for some scalars a_i, the sequence $\{\vec{x}_n\}$ can be generated by

$$\vec{x}_n = \vec{s} + \sum_{i=1}^{m} a_i \, \vec{v}_i \, q_i^n, \quad n = 0, 1, \ldots \tag{13}$$

If q_1 is the dominant eigenvalue and $a_1 \neq 0$, which will usually be true for the given $\vec{x}_0$, then the limit of $\{\vec{x}_n\}$ is $\vec{s}$ provided $|q_1| < 1$. If $|q_1| \geq 1$, then $\{\vec{x}_n\}$ is a divergent sequence and $\vec{s}$ is its antilimit. Therefore, assuming that $\{\vec{x}_n\}$ converges, it is the dominant eigenvalue, call it q, that is of importance. The smaller the modulus of q, the faster the convergence. However, if one or more eigenvalues have modulus close to unity, then the convergence will be slow.

Figures 4 through 6 (pages 16 and 17) show other trajectories of the sequence $\{\vec{x}_n\}$ generated by Equation (5) with dimension two. The matrix A and associated

eigenvalues for each figure are

Figure 5	Figure 4	Figure 6
$A = \begin{bmatrix} -1.0 & -0.2 \\ 0.75 & 0.6 \end{bmatrix}$	$A = \begin{bmatrix} 1.0 & -0.2 \\ 0.75 & -0.6 \end{bmatrix}$	$A = \begin{bmatrix} 1.0 & -1.0 \\ 1.0 & -0.2 \end{bmatrix}$
$q_i = -0.9, 0.5$	$q_i = 0.9, -0.5$	$q_i = 0.4 \pm 0.8i$

In addition, the eigenvectors are graphed and labeled as $\vec{v}$, for the eigenvector corresponding to the dominant eigenvalue q and referred to as the major eigenvector, and $\vec{v}_1$, the eigenvector corresponding to q_1 and referred to as the minor eigenvector. Since the eigenvalues of the matrix A in Figure 6 are complex conjugates, the eigenvectors are also complex and, hence, are not graphed. Figures 4 and 5 support the fact that the convergence of $\{\vec{x}_n\}$ is indeed asymptotic along the eigenvector $\vec{v}$. Figure 6 suggests that the trajectory of $\{\vec{x}_n\}$ for

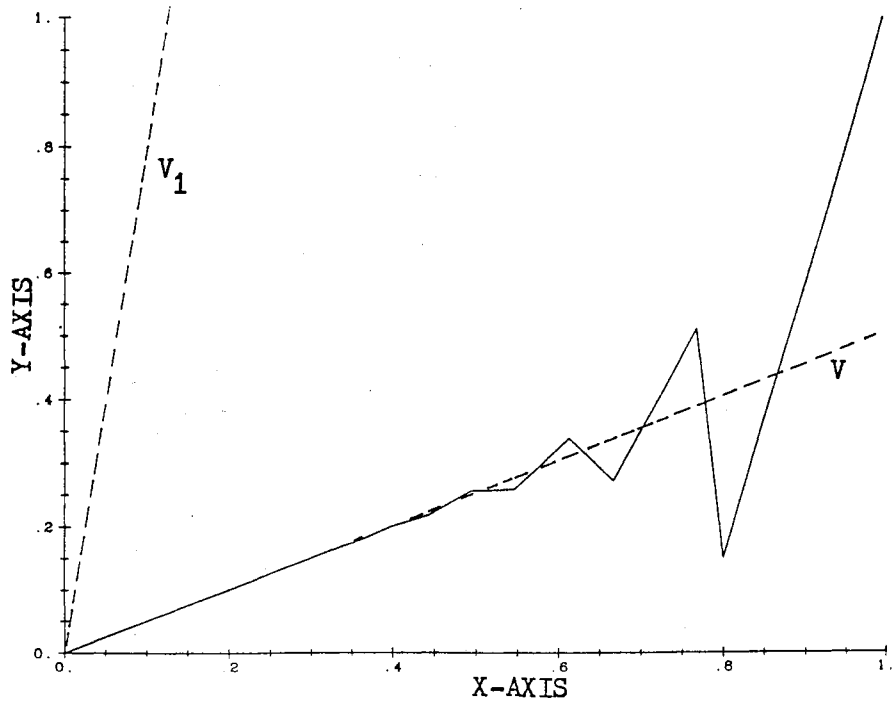

Figure 4. Trajectory for Eigenvalues
-0.9 and 0.5
$+0.9 \quad -0.5$

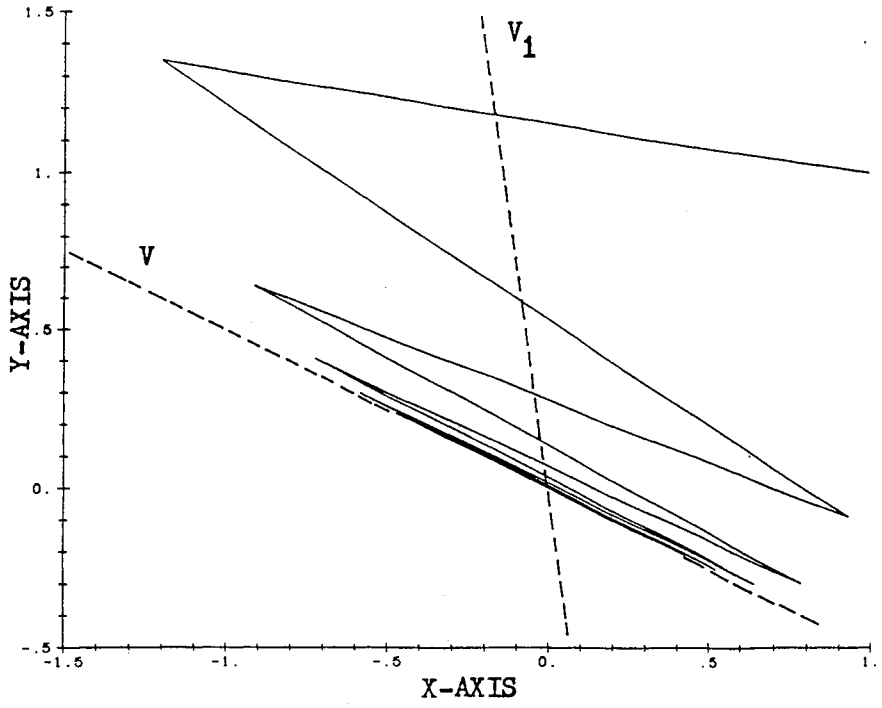

Figure 5. Trajectory for Eigenvalues
0.9 and -0.5
$-0.9 \quad +0.5$

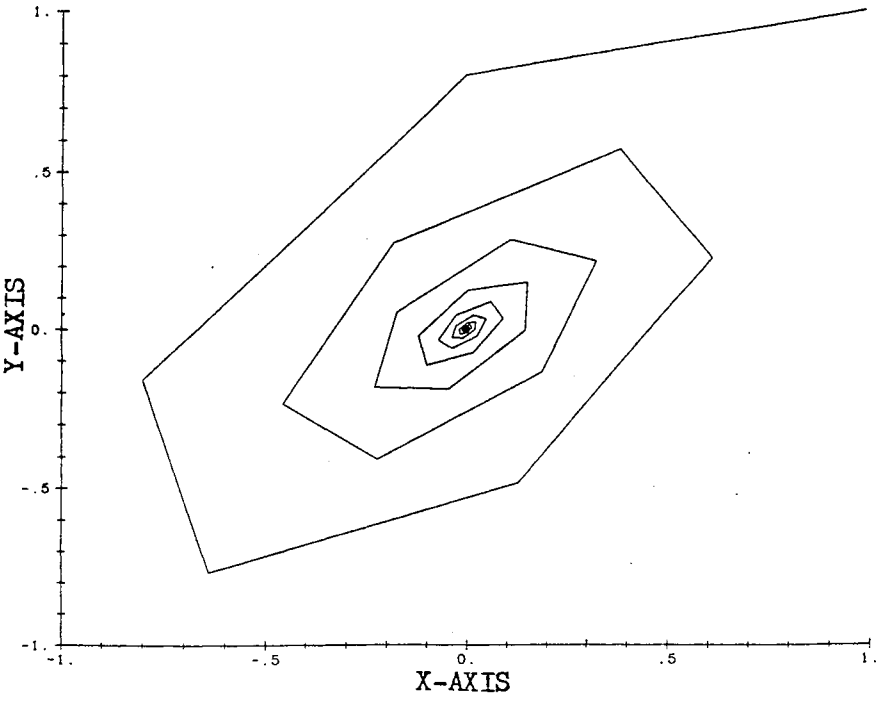

Figure 6. Trajectory for Eigenvalues
$0.4 \pm 0.8i$

problems with complex eigenvalues is some form of a spiral. The trajectory of another problem with complex conjugate eigenvalues is shown in Figure 7.a. Returning to Figure 3, it shows a smooth monotonic convergence. This is not always the case for two positive eigenvalues. However, as n continues to increase the convergence will eventually become monotonic and resemble Figure 3. Figure 7.b shows the trajectory of another problem with positive eigenvalues that initially begins to "converge" along the minor eigenvector but eventually converges monotonically along the major eigenvector. Figure 7.c shows an example of a trajectory for a problem with two negative eigenvalues.

Comparing Figures 4 and 5, we see that both trajectories zig-zag across the eigenvector associated with the negative eigenvalue. Because this is the ~~major~~ minor eigenvector in Figure 4, the zig-zag motion dampens out as the sequence approaches

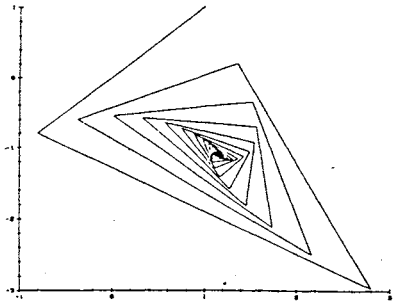

Fig. 7.a

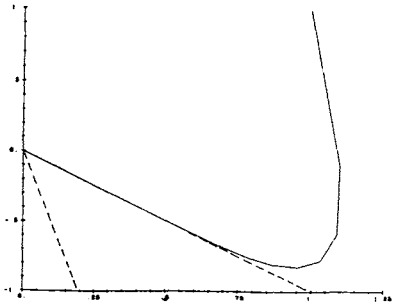

Fig. 7.b

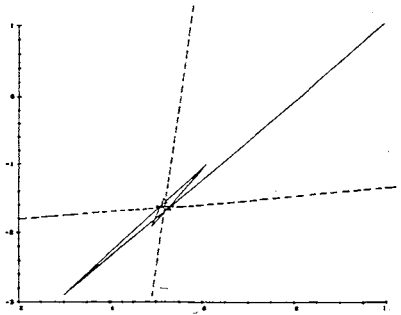

Fig. 7.c

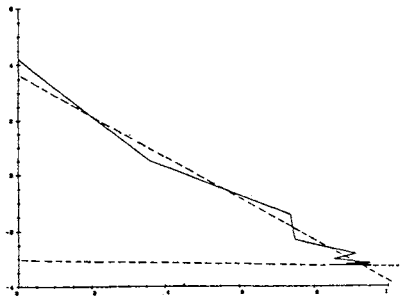

Fig. 7.d

Figure 7. Other Trajectory Examples

$\vec{s}$, and the convergence becomes almost linear along the major eigenvector. Figure 7.d shows the trajectory of a problem where the case is reversed and the graph starts in an almost linear approach and eventually ends up in the zig-zag motion. Why do the trajectories of these two problems differ?

From Equation (13), we have

$$\vec{x}_n - \vec{s} = \sum_{i=1}^{m} a_i \, \vec{v}_i \, q_i^n, \quad n = 0, 1, \ldots$$

Since the left-hand side of the equation is the difference between the n^{th} term of the sequence $\{\vec{x}_n\}$ and the solution $\vec{s}$, it will be referred to as the n^{th} error vector, denoted by $\vec{e}_n$. For $m = 2$, the error vector can be rewritten as

$$\vec{e}_n = a_1 \vec{v}_1 q_1^n + a\vec{v}q^n,$$

where q is the dominant eigenvalue and $\vec{v}$ is the eigenvector associated with q. If $n = 0$, then the initial error vector is

$$\vec{e}_0 = a_1 \vec{v}_1 + a\vec{v}.$$

If a is sufficiently smaller than a_1, then the dominant eigenvector for the first few sequence elements will be v_1. Therefore, the trajectory will start almost parallel to $\vec{v}_1$. However, as n increase, the higher powers will result in a dampening of q_1 and the dominant eigenvalue will cause a convergence in the direction of the major eigenvector. Figure 4 shows the resulting trajectory for q negative and q_1 positive. If the signs of the eigenvalues are reversed, then the trajectory starts almost linearly and terminates in a zig-zag fashion as shown in Figure 7.d.

Relaxation Factor

Up to this point, successive elements of the generated sequence $\{\vec{x}_n\}$ have been found by using the iteration Equation (3). However, there is a variation of (3) that

may be helpful in accelerating the convergence. This method uses a constant called a relaxation factor to adjust the distance the iteration moves from the previous sequence element. The new iteration equation is

$$\vec{y}_{n+1} = G_w(\vec{y}_n) = \vec{y}_n + w(G(\vec{y}_n) - \vec{y}_n), \tag{14}$$

where $n = 0, 1, \ldots,\ y_0 = x_0$ of Equation (5), and $w > 0$ is the relaxation factor. For $0 < w < 1$, (14) is called vector under-relaxation, and for $w > 1$, (14) is referred to as vector over-relaxation. With no restrictions on w, (14) is a parametric equation for the line containing the points $\vec{y}_n$ and $G(\vec{y}_n)$. For $w = 0$ and 1, $\vec{y}_{n+1}$ equals $\vec{y}_n$ and $G(\vec{y}_n)$, respectively. If $0 < w < 1$, $\vec{y}_{n+1}$ is located on the line between $\vec{y}$ and $G(\vec{y}_n)$. If $w > 1$, $\vec{y}_{n+1}$ is still on the line; however, $\vec{y}_{n+1}$ is "beyond" $G(\vec{y}_n)$, as viewed form $\vec{y}_n$.

Returning to (14) and using Equation (5),

$$\begin{aligned} \vec{y}_{n+1} &= \vec{y}_n + w([A\vec{y}_n + \vec{b}] - \vec{y}_n) \\ &= w(A\vec{y}_n + \vec{b}) + (1-w)\vec{y}_n \\ &= (wA + [1-w]I)\vec{y}_n + w\vec{b} \\ &= A_w\vec{y}_n + w\vec{b}, \end{aligned} \tag{15}$$

where $A_w = wA + (1-w)I$. Equations (14) and (15) are equivalent iteration equations.

Let q_i and p_i, $i = 1, \ldots, m$, be the eigenvalues of the matrices A and A_w, respectively. Then

$$\begin{aligned} 0 &= \det(A_w - pI) = \det(wA + [1-w]I - pI) \\ &= \det(wA + [1-w-p]I) \\ &= w^m \det(A - [1-(1-p)/w]I). \end{aligned}$$

Since $w \neq 0$, the eigenvalues of A must be $q = 1-(1-p)/w$. Hence, $p = 1+w(q-1)$.

Therefore, we have a parametric equation for a line through the point $z = 1 = (1, 0)$ and q in the complex plane. Since

$$|p - 1| = |1 + w(q - 1) - 1| = |w(q - 1)|,$$

the distance between the eigenvalues of A_w and the point z is w times the distance between the eigenvalues of the matrix A and z.

As a result, Equation (13) can be rewritten with a new set of eigenvalues. By carefully choosing w, one may be able to convert a divergent problem into a convergent one. For example, if we have a two-dimensional problem with eigenvalues 0.5 and -1.5, then iteration Equation (5) will generate a divergent sequence. However, the eigenvalues of iteration Equations (14) and (15) with a relaxation factor of $w = 0.5$ are 0.75 and -0.25. Hence, by using Equation (14), one "may" produce a convergent sequence even if some of the eigenvalues of A have moduli greater than unity. The word may is used because there are cases where no value of w will convert a divergent sequence into a convergent one; for example, a problem with an eigenvalue of 2.0. For any value of $w \neq 0$, the modulus of the converted eigenvalue will always be greater than unity. Figure 8 (page 22) shows the region in the complex plane of eigenvalues of Equation (4) for which Equation (14) with $w = 0.5$ generates a convergent sequence. The region can be described as

$$\{p : |p - a| < 4, \text{ where } a = -1 + 0\mathrm{i}\}.$$

Choosing $w > 1$ can also be useful on some occasions. Consider the two-dimensional problem where the eigenvalues are 0.6 and 0.9. Even though the iteration equation will produce a convergent sequence, because the dominant eigenvalue is close to unity, the convergence will be slow. A relaxation factor of $w = 2$ will transform the problem into an equivalent problem with eigenvalues 0.2 and 0.8. Hence, the transformation has a smaller dominant eigenvalue, resulting in faster

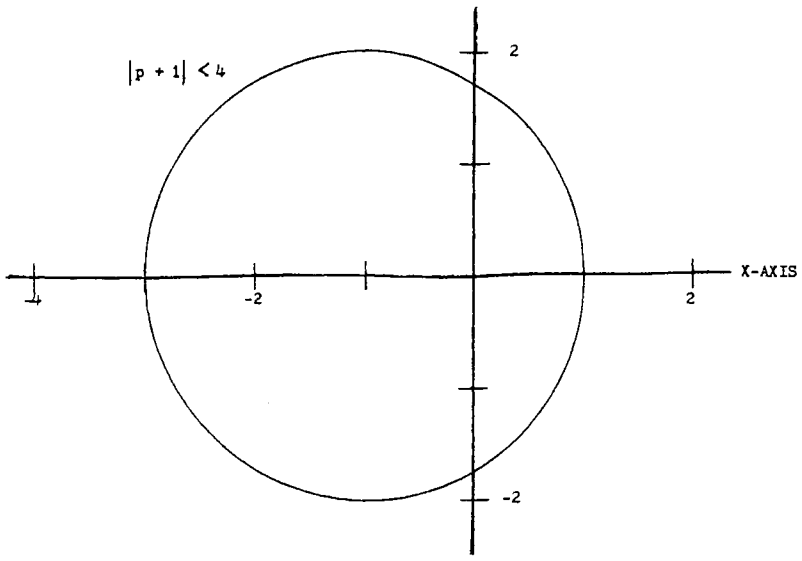

Figure 8: Eigenvalue Region of A Resulting in Convergent Sequences for A_w, $w = 0.5$

convergence.

One must show care in choosing w. There are problems where a relaxation factor less than unity can cause a convergent sequence to converge slower or a relaxation factor greater than unity can cause a convergent sequence to diverge. Consider the sequence generated by (5) with eigenvalues 0.8 and -0.2. Using Equation (14) and $w = 0.5$, the transformation problem has eigenvalues 0.9 and 0.4. The transformed problem has a dominant eigenvalue closer to unity; hence, the convergence is slower. If $w = 2.0$, the transformation problem has eigenvalues 0.6 and -1.4, resulting in a divergent problem. Unless otherwise stated, the results given in this study will be for $w = 1$.

CHAPTER III

GENERALIZED INVERSES

Before we continue with acceleration methods, there is an area that needs extended coverage beyond that which is available in a normal course of matrix theory or linear algebra. This area is the theory and applications of what is called a generalized inverse of a matrix.

Let A be a square m x m matrix with rank $R(A) = m$. Then we know that there exists a unique matrix B, called the inverse of A, such that $AB = BA = I_m$, where I_m is the identity matrix of order m. The inverse of A is normally denoted by A^{-1}. Hence, given the square m x m matrix A and the m-dimensional vector $\vec{y}$, then the solution of a set of consistent linear equations

$$\vec{y} = A\vec{x} \tag{16}$$

is $(A^{-1})\vec{y} = (A^{-1})(A\vec{x}) = I_m\vec{x} = \vec{x}$. If A has an inverse, then A is said to be nonsingular; otherwise, A is singular. If A is a rectangular matrix, then no such matrix B exists as the inverse of A and, thus, a simple expression of a solution of (16) in terms of A is more difficult.

Moore (1920) extended the normal concept of inverses to singular and rectangular matrices. However, the theoretical properties of these matrices were not fully investigated until 1955, when Penrose defined a uniquely determined inverse matrix for any matrix A which he called the generalized inverse. Moore's and Penrose's inverses are equivalent when the inner product of the two m-dimensional vectors $\vec{x} = (x_1, \ldots, x_m)^T, \vec{y} = (y_1, \ldots, y_m)^T$ is defined by

$$(\vec{x}, \vec{y}) = (\vec{y}^{\,*})\vec{x} = \sum_{i=1}^{m} \bar{y}_i \, x_i,$$

where $\vec{y}^{\,*}$ indicates the complex conjugate transpose of $\vec{y}$ and $\bar{y}_i$ is the complex conjugate of y_i. Penrose's definition of a generalized inverse is as follows:

Definition: For any matrix A, square or rectangular, real or complex, there exists a unique matrix G satisfying the conditions

$$\begin{aligned} &(1)\quad AGA = A && (2)\quad GAG = G \\ &(3)\quad (AG)^* = AG && (4)\quad (GA)^* = GA. \end{aligned} \tag{17}$$

G is called the Moore-Penrose generalized inverse of A.

With the use of generalized inverses, we can extend the concept of solving (16) where A is a m x k matrix of rank r, $r = k \leq m$, $\vec{y}$ is a m-dimensional vector, and $\vec{x}$ is a k-dimensional vector. If $r = k = m$, A is nonsingular and the problem is as before with a solution of $x = (A^{-1})y$. If $m \neq k$, then $\vec{x} = G\vec{y}$, where G is the unique solution of Equations (17). However, as Penrose pointed out, a solution of (16) does not require a matrix G which satisfies all the conditions of (17). One can find a solution of (16) which satisfies only condition (1) of (17). Given $\vec{y} = A\vec{x}$, then $AG\vec{y} = AGA\vec{x} = A\vec{x}$. Therefore, $\vec{x} = G\vec{y}$.

Some authors refer to generalized inverses by other names. Greville (1959) and Rohde (1964) prefer the use of the name "pseudo-inverses." Rao (1965) referred to them frequently as the "Moore-Penrose inverses." Albert (1972) combines the names together and calls them the "Moore-Penrose pseudo-inverse." In addition, different names are given to matrices which satisfy one or more of the conditions of (17). In this study, generalized inverses will refer to those matrices that satisfy at least condition (1) of (17).

Generalized inverses of matrices satisfying condition (1) of (17) are not unique (Pringle and Rayner, 1971). Let A be a m x k matrix. If $R(A) = m$, then a generalized inverse of A is

$$G = (VA^*)(AVA^*)^{-1},$$

where V is an arbitrary matrix such that $R(AVA^*) = R(A)$. G is called a right inverse of A in this case. If $R(A) = k$, then a generalized inverse of A is

$$G = ((A^*)VA)^{-1}(A^*)V, \tag{18}$$

where V is an arbitrary matrix such that $R((A^*)VA) = R(A)$. G is called a left inverse of A for this case.

Returning to the problem (16), when A is a m x k matrix of rank $r = k \leq m$, we can find a solution to the problem by using generalized inverses. Hence,

$$\vec{x} = I_r\vec{x} = GA\vec{x} = G\vec{y}$$

where G is of the form (18). A simple choice for V is $V = I_r$ such that

$$G = (A^*A)^{-1}A^*. \tag{19}$$

Henceforth, the definition of the generalized inverse of the m x k matrix A, $k \leq m$, is the k x m matrix G as defined in (19) with the notation $G = A^+$.

CHAPTER IV

AITKEN'S Δ^2 METHOD FOR SCALARS AND VECTORS

Theory for Scalars

Aitken's Δ^2 method was introduced in Chapter II to help solve Equation (6). Consider Equation (8) applied to a geometric series where $\{x_n\}$ is the sequence of partial sums. Then the ratio, r, is constant and, hence, (8) can be written as the sum

$$s = x_n + u_{n-1}\left(\frac{r}{1-r}\right), \quad |r| < 1.$$

Substituting $u_{n-1} = ru_{n-2} = r^{n-1}u_0$ and $u_0 = x_1 - x_0 = rx_0$ gives

$$\begin{aligned} s &= x_n + u_0 r^{n-1}\left(\frac{r}{1-r}\right) = x_n + \left(\frac{u_0}{1-r}\right) r^n \\ &= x_n + \left(\frac{x_0 r}{1-r}\right) r^n. \end{aligned} \tag{20}$$

Therefore, for $|r| < 1$, $x_n - s = cr^n$, where

$$c = \frac{-x_0 r}{1-r}.$$

So, there exist constants s and r such that as n increases by one, the distance from x_n to s is multiplied by r. It is easy to see that if $|r| < 1$, then r^n goes to zero as n increases, which implies that s is the limit of the sequence $\{x_n\}$. If $r = -1$ or $|r| > 1$, then s is the antilimit of $\{x_n\}$. This is a special case of Equation (13).

If we assume that there exists a constant $p \neq s$ such that $x_n - p = dr^n$, for some constant d, then

$$dr^n = x_n - p = s + cr^n - p.$$

Hence, $s - p = (d - c)r^n = br^n$. Since s and p are constants, br^n must remain constant as n increases; hence, $b = 0$. So $s - p = 0$, which implies $s = p$. Therefore, s is unique.

Written below is (10) in a slightly different form and two variations:

$$y_{n+2} = x_{n+2} - \frac{(u_{n+1})^2}{(v_n)} \tag{21}$$

$$= \frac{x_n x_{n+2} - x_{n+1}^2}{(x_{n+2} - x_{n+1}) - (x_{n+1} - x_n)} \tag{22}$$

$$= x_n - \frac{(u_n)^2}{(v_n)}, \tag{23}$$

where $u_n = x_{n+1} - x_n$ and $v_n = (x_{n+2} - x_{n+1}) - (x_{n+1} - x_n)$ are defined as the forward difference operators. Formula (21) is the most desirable one for a convergent sequence using floating point arithmetic since x_{n+2} is a better estimate of the limit than x_n and the computed error, $x_{n+2} - s$, is smaller than the computed error, $x_n - s$. On the other hand, if s is the antilimit, then (23) should be the choice. Aitken (1936-37) used (22), but apparently never used an automated computer and could monitor rounding errors "visually" at each step.

If the sequence $\{x_n\}$ is such that the ratios of consecutive errors converge to a nonzero constant, independent of n, then (20) still holds, where r is the limit of the ratios. Therefore, Equations (21) through (23) will also hold as an estimate for s and we can apply Aitken's method to approximate the limit of the sequence. This amounts to having what is called linear convergence if $|r| < 1$.

DEFINITION: Given the convergent sequence $\{x_n\}$ and its limit s. If

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n} = C,$$

where $e_n = x_n - s$ is the error of the nth element and $|C| < 1$, then the sequence $\{x_n\}$ converges linearly to s.

The question now arises as to what happens if the ratios of consecutive errors approach a C outside of the open interval $(-1,1)$. It can be shown that if $C = 1$, then the denominators of Equations (21) through (23) go to zero. Hence, the Δ^2 method will not work well if the ratios converge to 1. If C is -1 or outside the closed interval $[-1,1]$, we do not have convergence, but the method can still be used in hopes of finding the antilimit of the sequence.

It was shown in Chapter II how Aitken's Δ^2 method can be used in the static sense to accelerate a scalar sequence. However, a modification of this method is to compute the extrapolated value x_2' by applying Aitken's method to x_0, x_1, and x_2. Using x_2' as the initial value, generate two iterates x_3' and x_4'. Extrapolating again, we determine the value x_4'' and continue the pattern until convergence. A procedure following this type of pattern: iterating, extrapolating, and then iterating the result, is referred to as a repeated method or a semi-dynamic extrapolation model. Figure 9 (page 29) shows a diagram of a semi-dynamic procedure where $\mathrm{EXT}(x_n, x_{n+1}, x_{n+2})$ implies applying the extrapolation method to the values in parentheses.

Vector Theory

For a sequence of vectors, Aitken (1936-37) applied the extrapolation technique componentwise. In other words, if $x^{(i)}, i = 1, \ldots, m$, represents the i^{th} component of the vector $\vec{x}$, then (10) becomes

$$\vec{y}^{(i)}_{n+2} = \vec{x}^{(i)}_{n+2} - \frac{\left(\vec{x}^{(i)}_{n+2} - \vec{x}^{(i)}_{n+1}\right)^2}{\vec{x}^{(i)}_{n+2} - 2\vec{x}^{(i)}_{n+1} + \vec{x}^{(i)}_{n}} \quad n = 0, 1, \ldots$$

The major problem with this technique is that the computation of one or more component elements may involve a denominator of zero, which obviously results

$$
\begin{array}{lll}
x_0 & & \\
x_1 & & \\
x_2 & \mathrm{EXT}(x_0, x_1, x_2) \rightarrow & x_2' \\
 & & x_3' \\
 & & x_4' \quad \mathrm{EXT}(x_2', x_3', x_4') \rightarrow \quad x_4'' \quad \text{etc.}
\end{array}
$$

Figure 9. Diagram of Semi-Dynamic Model

in an invalid element in our new sequence. In addition, since the convergence of the components of a vector is usually related to one another, applying the technique componentwise loses this relationship and may result in some components accelerating the "wrong" way, in the direction opposite to the majority.

Jennings (1971) modified Aitken's vector method for the linear case to help prevent the possibility of this infinite or even erratic result. He used a vector $\vec{w}$ to help define a rate of decay between $\vec{x}_{n+1}$ and $\vec{x}_{n+2}$. His iterative method is

$$\vec{y} = \vec{x}_{n+2} + Q(\vec{x}_{n+2} - \vec{x}_{n+1}), \quad n = 0, 1, \ldots, \tag{24}$$

where

$$Q = \frac{(\vec{w}^*)(\vec{x}_{n+1} - \vec{x}_{n+2})}{(\vec{w}^*)(\vec{x}_{n+2} - 2\vec{x}_{n+1} + x_n)}, \tag{25}$$

$\vec{w}^*$ represents the complex conjugate transpose of $\vec{w}$, and $\vec{y}$ represents either $\vec{y}_{n+2}$ of Equation (10) or $\vec{x}\,'_{n+2}$ depending upon whether the model used is static or semi-dynamic.

In order to discuss possible choices of $\vec{w}^*$ Jennings' work, some definitions are needed. Because a m-dimensional vector consists of m components, it is convenient to have some method of determining its size. This measurement is provided by assigning to a vector a real-valued, nonnegative number known as a norm. However,

the assignment is not unique since there exist several norms. For purposes of this study, only two norms will be used: the infinity norm (∞-norm) and the Euclidean norm (2-norm). Given the vector $\vec{x} = (x_1, \ldots, x_m)^T$, the two norms are respectively defined as

$$\begin{aligned} \|\vec{x}\|_\infty &= \max\ |x_i|\,,\ \ i = 1, \ldots, m; \quad \text{and} \\ \|\vec{x}\|_2 &= \left(\sum_{i=1}^{m} (\bar{x}_i\ x_i)^2 \right)^{1/2} . \end{aligned}$$

Jennings gave two choices for $\vec{w}$. The first choice is when the iterative sequence is governed by a symmetric matrix A. For this case, Jennings suggested the vector $\vec{w} = \vec{x}_n - \vec{x}_{n+1}$ and referred to the formula as First Difference Modulation (FDM). Given the scalars $a_i, i = 1, \ldots, m$, such that

$$\vec{x}_0 - \vec{s} = \sum_{i=1}^{m} a_i\ \vec{v}_i,$$

where $\vec{v}_i$ is the eigenvector corresponding to the eigenvalue q_i of the iterative matrix A, he showed that

$$Q = \frac{\sum_{i=0}^{m} q_i\ z_i}{\sum_{i=0}^{m} (1 - q_i) z_i},$$

where $z_i = q_i^{2n}(1 - q_i)^2 a_i^2 \geq 0$. Since all eigenvalues of A must have moduli less than unity for convergence, Q cannot have a zero denominator except when convergence is obtained.

Jennings' second choice for $\vec{w}$, Second Difference Modulation (SDM), is when the sequence is governed by a nonsymmetric matrix. For this case he chose $\vec{w} = \vec{x}_{n+2} - 2\vec{x}_{n+1} + \vec{x}_n$. Thus the denominator is the square of the Euclidean norm of the second difference vector. Hence, the denominator cannot result in zero unless, once again, convergence is obtained. It should be added that for nonlinear cases there is no guarantee that the denominator will not be zero.

The importance of Q in Equation (24) is to specify the distance of the current extrapolation as a multiple of the last difference vector, $\vec{x}_{n+2} - \vec{x}_{n+1}$. Though

Jennings' two suggestions come naturally from Aitken's formula, values for Q can be found by other effective methods. Chandler (1987) suggests two other methods for determining Q, that based on results obtained on test problems in Chapter XI work as well, if not better in some cases, than Jennings' suggestions. First, define Q as the quotient of Euclidean norms:

$$Q = (\text{sign}) \frac{\|\vec{x}_{n+2} - \vec{x}_{n+1}\|_2}{\|\vec{x}_{n+2} - 2\vec{x}_{n+1} + \vec{x}_n\|_2}, \tag{26}$$

where sign = 1 except when the cosine of the angle between the difference vectors, $\vec{x}_{n+2} - \vec{x}_{n+1}$ and $\vec{x}_{n+1} - \vec{x}_n$, is negative or the norm of $(\vec{x}_{n+1} - \vec{x}_n)$ is less than the norm of $(\vec{x}_{n+2} - \vec{x}_{n+1})$. In these cases set sign $= -1$. His second suggestion is

$$Q = \frac{-\|\vec{x}_{n+2} - \vec{x}_{n+1}\|_2}{\|\vec{x}_{n+2} - \vec{x}_{n+1}\|_2 \pm \|\vec{x}_{n+1} - \vec{x}_n\|_2}, \tag{27}$$

where the denominator is a sum if the cosine of the difference vectors is negative and a subtraction, otherwise.

The key to these four variations is that they all extrapolate along the vector $\vec{x}_{n+2} - \vec{x}_{n+1}$. Therefore, it is essential for efficient operation of the algorithm that the cosines of the difference vectors converge to either plus or minus unity. If the cosine equals plus or minus unity, then all four methods give precisely the same results, which is also the same result as componentwise Aitken. However, there exist problems where all four suggested methods for determining Q work poorly. One example is where the dominant eigenvalues are complex, resulting in a spiraling convergence, see Figure 6. A second example is where a single real eigenvalue does not dominate, e.g., the two-dimensional problem with eigenvalues of 0.8 and -0.8 where the cosine of the angle between consecutive difference vectors is constant, approximately -0.42753, see Figure 10 (page 32). However, Jennings pointed out that this case can be handled very well by taking successive pairs of the basic iteration to form new basic iterations to accelerate, i.e., accelerate the sequence

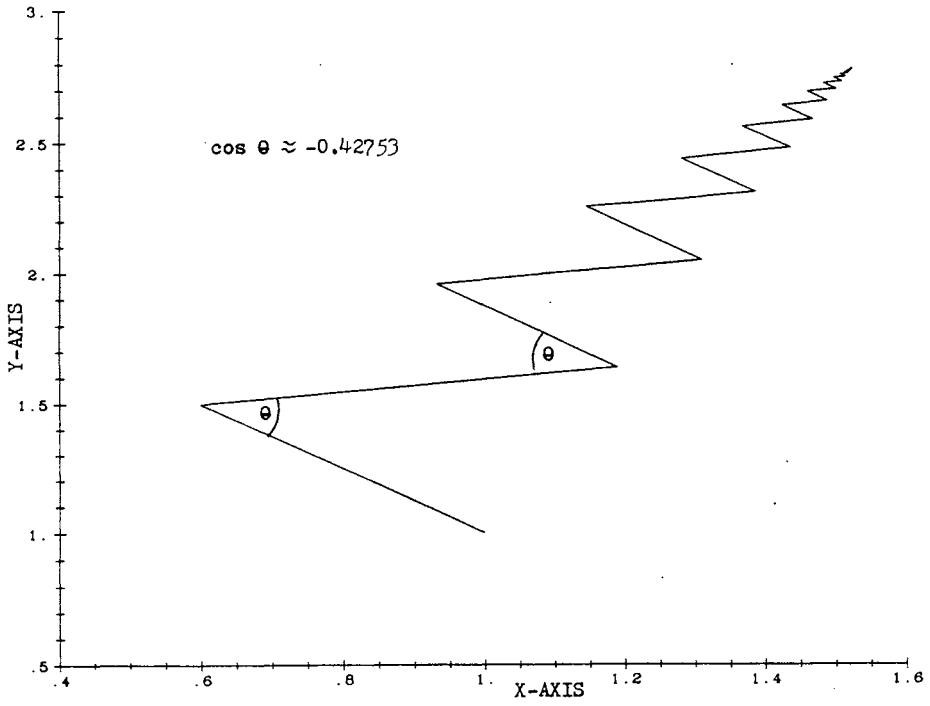

Figure 10. Trajectory for Eigenvalues 0.8, −0.8

$\{\vec{y}_n\}$ where $\vec{y}_n = \vec{x}_{2n}$ or $\vec{y}_n = \vec{x}_{2n+1}$, $n = 0, 1, \ldots$ This effectively squares all of the eigenvalues of the iteration matrix (Jennings, 1971). Fortunately, many practical iterations are dominated by a single real eigenvalue, and the vector Aitken method with the Q suggestions of Jennings and Chandler are often effective in these cases. In addition, Aitken's method will not work tremendously well on divergent sequences since the method is trying to approximate the solution along the vector $\vec{x}_{n+2} - \vec{x}_{n+1}$. Chen (1984) suggests interchanging the vectors $\vec{x}_n$ and $\vec{x}_{n+2}$ in equation (24). Hence, the extrapolation will be along the vector $\vec{x}_n - \vec{x}_{n+1}$.

Combining Equation (24) with Aitken's repeated model, a semi-dynamic algorithm for vectors results. Henceforth, when vector Aitken's method is mentioned, it is referring to this algorithm. Unless otherwise stated, Q will be Jennings' SDM.

AITKEN'S SEMI-DYNAMIC ALGORITHM 4.1:

Find a solution to $\vec{x} = G(\vec{x})$ given the initial approximation vector $\vec{x}_0$. Define the terminology "If converged" to mean "If $\|\vec{y} - \vec{x}\| < \text{Tol}$, where Tol is some predetermined tolerance value and $\vec{x}$ and $\vec{y}$ are the vectors of the current step."

Step 1. Compute $\vec{x}_1 = G(\vec{x}_0)$.

Step 2. Compute $\vec{x}_2 = G(\vec{x}_1)$. If converged, stop; otherwise, go to step 3.

Step 3. Find $\vec{y} = \vec{x}_2 - Q(\vec{x}_2 - \vec{x}_1)$, where Q is defined by Equation (25), (26), or (27).

Step 4. Compute $\vec{z} = G(\vec{y})$. If converged, stop; otherwise, set $\vec{x}_0 = \vec{y}$, $\vec{x}_1 = \vec{z}$ and go to step 2.

One may use any norm desired for the stopping criterion. There are advantages and disadvantages for all of them. The Euclidean norm will show a smoothness in the differences as they approach the tolerance value. However, the use of this norm could result in system overflow due to the squaring of the difference components unless care is taken in computing Q. The infinity norm is simply the largest component of the difference vector, $\vec{y} - \vec{x}$. It is not as smooth as the Euclidean norm since the largest component may be a different component from one iteration to the next. However, due to its simplicity, results in this study are based on the infinity norm of the difference vector.

In addition, there are other stopping criteria which may be used:

$$\|\vec{y} - \vec{x}_2\| \leq (\text{Tol})\|\vec{y}\|, \quad \vec{y} \neq \vec{0}, \quad \text{and} \quad \|F(\vec{y})\| < \text{Tol},$$

where $F(\vec{x}) = \vec{0}$. Unfortunately, difficulties can arise no matter which of the stopping criteria we use. For example, the sequence defined by $x_n = \sum_{k=1}^{n}(1/k)$ is divergent but $\lim_{n\to\infty}(x_n - x_{n-1}) = 0$. This sort of harmonic convergence is "sublinear" (Brent, 1972) and, ordinarily, is never encountered in practical fixed point

problems. For purpose of this study, the stopping criterion for all algorithms will be as stated in Algorithm 4.1.

Numerical Examples

As an example of the vector Aitken method, consider the two-dimensional problem of finding the solution to the system

$$\begin{aligned} x &= A(\sin x) + B(\cos y) \quad \text{and} \\ y &= A(\cos x) - B(\sin y), \end{aligned} \tag{28}$$

where $A = 0.7$, $B = 0.2$, and $x_0 = y_0 = 0$ (Henrici, 1964). The iteration function then becomes

$$\begin{aligned} x_{n+1} &= A(\sin x_n) + B(\cos y_n) \quad \text{and} \\ y_{n+1} &= A(\cos x_n) - B(\sin y_n). \end{aligned}$$

After thirty-one iterations, we obtain the solution vector to five decimal places, (0.52652,0.50792). If we apply vector Aitken in static form, we obtain the solution to the same degree of accuracy in 12 iterations. Table 5 (page 35) shows the Euclidean norm of the error vector, denoted by $E_n^{(i)} = \|x_n^{(i)} - s\|_2$, where the i^{th} column represents the sequence derived by applying Aitken's method i times. The zero column is the original iteration sequence.

Using Aitken's method semi-dynamically, Algorithm 4.1, the same results are obtained in 15 iterations, Table 6 (page 36). However, Anderson (1965, p. 551) says that Aitken's method "is considerably less effective if applied statically, ... than it is if applied dynamically." The static model requires 32 extrapolations to only 6 for the semi-dynamic model. Therefore, the benefit of 3 fewer iterations is loss due to the time required to perform 26 additional extrapolations. In addition, since the number of column sequences are not known beforehand for the static method, the semi-dynamic method does not require the allocation of storage space to ensure

TABLE 5

AITKEN'S STATIC METHOD APPLIED TO PROBLEM (28) WITH RESULTS AS EUCLIDEAN NORM OF ERROR VECTOR

n	$E_n^{(0)}$	$E_n^{(1)}$	$E_n^{(2)}$	$E_n^{(3)}$	$E_n^{(4)}$	$E_n^{(5)}$
0	0.53521					
1	0.37883					
2	0.23961	0.25854				
3	0.16527	0.16557				
4	0.11013	0.07612	0.01575			
5	0.73253	0.00578	0.31493			
6	0.04820	0.00328	0.00336	0.15449		
7	0.03156	0.00154	0.00182	0.00181		
8	0.02058	0.00077	0.00041	0.00062	0.00061	
9	0.01338	0.00032	0.00005	0.00015	0.00062	
10	0.00868	0.00014	0.00004	0.00003	0.00001	0.00021
11	0.00563	0.00006	0.00001	0.00001	0.00001	0.00001
12	0.00365	0.00003	0.00000	0.00000		

enough columns for convergence. Thus, the semi-dynamic model is usually more efficient than the static model.

Irons and Shrive (1987) made a modification to Aitken's method for scalars. Assume that we have the two relations $y_1 = G(y_0)$ and $y_3 = G(y_2)$, that the ratio of consecutive error terms is constant, and that s is the limit of the sequence $\{y_n\}$. Then the following is true:

$$r = \frac{y_1 - s}{y_0 - s} = \frac{y_3 - s}{y_2 - s}.$$

TABLE 6

AITKEN'S SEMI-DYNAMIC METHOD
APPLIED TO PROBLEM (28)

n	E_n	n	E_n	n	E_n
1	0.37883	6	0.00393	11	0.00005
2	0.23961	7	0.00291	12	0.00004
3	0.17611	8	0.00189	13	0.00001
4	0.11835	9	0.00016	14	0.00001
5	0.00734	10	0.00010	15	0.00000

Solving for s gives

$$s = y_3 - \frac{(y_2 - y_3)(y_1 - y_3)}{(y_0 - y_1) - (y_2 - y_3)}. \tag{29}$$

We can use (29) as a model iteration formula for estimating the limit of a sequence for which the ratios of consecutive errors converge to a constant. Given the scalars y_0, y_1, y_2, and y_3; then

$$\begin{aligned} y_{n+4} &= y_{n+3} - \frac{(y_{n+2} - y_{n+3})(y_{n+1} - y_{n+3})}{(y_n - y_{n+1}) - (y_{n+2} - y_{n+3})} \text{ and} \\ y_{n+5} &= G(y_{n+4}), \quad n = 0, 2, 4, \ldots \end{aligned} \tag{30}$$

Formula (30) gives us a third type of model, a fully dynamic method. Studying the diagram of the dynamic model in Figure 11 (page 37), we see that each extrapolation after the first one uses only one additional iterate and data obtained from previous iterations. A dynamic model does not require restarting our procedure by generating all necessary iterates from the latest extrapolation. We will see in later chapters the advantages of this model. Given the sequence $\{x_n\}$, we may apply Irons and Shrive's dynamic model after two iterations by setting $y_0 = x_0$,

$$\left.\begin{array}{l} y_0 \rightarrow y_1 \\ y_2 \rightarrow y_3 \end{array}\right\} \quad \mathrm{EXT}(y_0, y_1, y_2, y_3) \rightarrow y_4$$

$$\left.\begin{array}{l} y_2 \rightarrow y_3 \\ y_4 \rightarrow y_5 \end{array}\right\} \quad \mathrm{EXT}(y_2, y_3, y_4, y_5) \rightarrow y_6$$

$$\left.\begin{array}{l} y_4 \rightarrow y_5 \\ y_6 \rightarrow y_7 \end{array}\right\} \quad \mathrm{EXT}(y_4, y_5, y_6, y_7) \rightarrow y_8 \quad \text{etc.}$$

Figure 11. Diagram of Dynamic Model

$y_1 = y_2 = x_1$, and $y_3 = x_2$. Hence, the first extrapolation will be identical to Aitken's first extrapolation; however, the equalities stop at this point.

Consider the iterative equation

$$x_{n+1} = 2\sin(x_n), \quad n = 0, 1, \ldots, \quad \text{and} \quad x_0 = 1. \tag{31}$$

Table 7 (page 38) shows a comparison of the results obtained when Aitken's semi-dynamic method and Irons and Shrive's fully dynamic method are applied to (31). The column headings represent the current iterated value (iter) and the current extrapolated value (ext), if one was applied on that iteration.

Unsuccessful Application

In Chapter II, it was noted that wrong answers can sometimes be obtained when trying to accelerate a sequence by Aitken's method. Shanks (1955) found such a problem. Let $\{x_n\}$ be the sequence of partial sums of the function

$$\begin{aligned} f(z) &= \frac{2}{(1-z)(2-z)} \\ &= 1 + (3/2)z + (7/4)z^2 + (15/8)z^3 + \ldots \end{aligned}$$

TABLE 7

DYNAMIC VS SEMI-DYNAMIC COMPARISON
FOR PROBLEM (31)

	Semi-Dynamic		Dynamic	
n	iter	ext	iter	ext
0	1.000000		1.000000	
1	1.682942		1.682942	
2	1.987436	1.915372	1.987436	1.915372
3	1.882438		1.882438	1.892686
4	1.903662	1.895344	1.897278	1.895462
5	1.895590		1.895514	1.895494
6	1.895434	1.895495		
7	1.895494			

If $z = 4$, the series diverges. However, it was shown in Chapter II that Aitken's method can obtain the antilimit of a divergent sequence. Table 8 (page 39) shows the static results when Aitken's method is applied to this problem. Column i is the sequence obtained by the i^{th} application of the method. We see that the later columns converge to $7/27 = 0.25926...$ However, the value of $f(4)$ is $1/3$. Thus, Aitken's method did converge, but to the "wrong" value. Shanks showed that erroneous results in this problem will be obtained only for $z = 4$. Hence, one may feel confident that wrong results are few and far between in practical applications, but do exist. However, Lubkin (1952) did prove that if any two consecutive columns of the Aitken's table both converge, then they converge to the same limit. Hence, if all columns converge, then they all converge to the correct limit.

TABLE 8

CONVERGENCE TO WRONG ANSWER BY AITKEN'S STATIC METHOD APPLIED TO SHANKS' EXAMPLE

n	0	1	2	3	4
0	0				
1	1				
2	7	−0.2000			
3	35	−0.6364			
4	155	−1.5217	0.2241		
5	651	−3.2979	0.2437		
6	2667	−6.8526	0.2519	0.2589	
7	10,795	−13.9634	0.2557	0.2589	
8	43,435	−28.1854	0.2575	0.2592	0.2593

CHAPTER V

SHANKS' TRANSFORMATIONS FOR SCALARS

In this chapter the general framework for deriving the remaining acceleration techniques will be established. The motivation will be similar to Shanks' (1955) development of his e_k transformation for scalar sequences. First, consider a variety of typical scalar sequences $\{x_n\}$: convergent, divergent, monotonic and oscillatory. Plotting the sequence elements versus n and connecting them with a smooth curve give graphs similar to the samples shown in Figure 12.

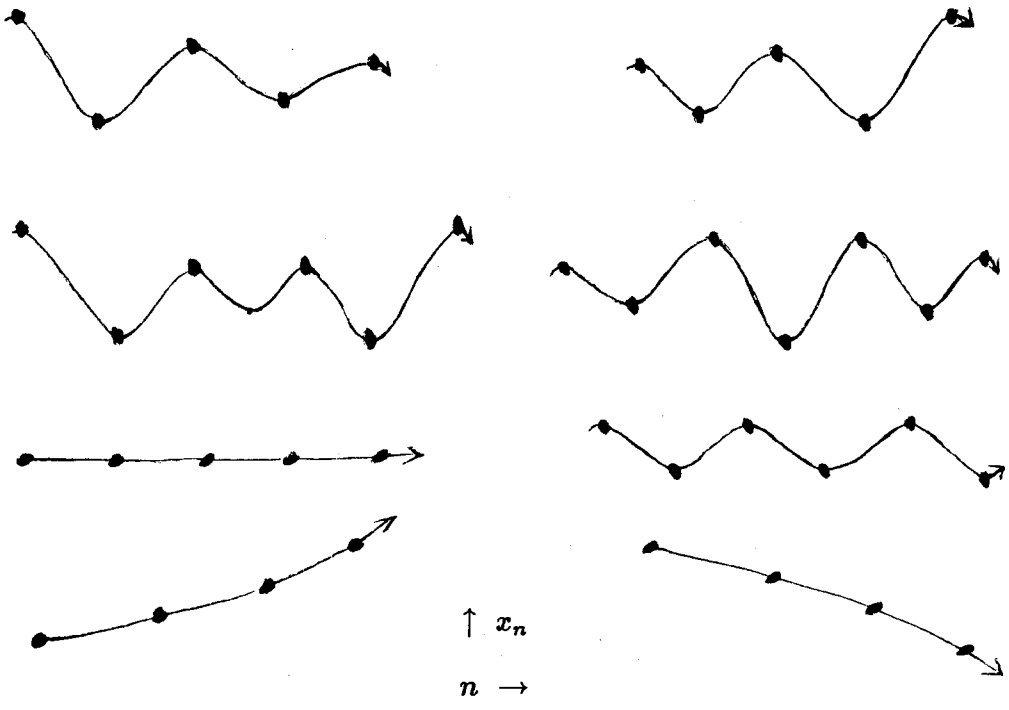

Figure 12. Plots of Typical Sequences

Shanks observed that the graphs all look like the graphs of what he called "physical transients." By this term he meant a physical quantity, p, as a function of time in the form:

$$p(t) = s + \sum_{i=1}^{k} a_i \, e^{b_i t},$$

where the b_i's are complex numbers, $b_i \neq 0$. He referred to them as "transients" so that he could apply the term in a more general sense for both convergent and divergent sequences. Since $p(t)$ is an exponential function with b_i an element of the complex numbers, the set of functions also include the trigonometric functions. Shanks represented the sequences $\{x_n\}$ as if they were "mathematical transients," a function of n in the form

$$x_n = s + \sum_{i=1}^{k} a_i \, q_i^n, \quad q_i \neq 0, 1,$$

where s, a_i, and q_i are constants independent of n and $q_i \neq q_j$ for $i \neq j$. Therefore, his "mathematical transient" equation is identical to the relationship (13) for scalars. Once again, we have the concept that s is either the limit or the antilimit of $\{x_n\}$ depending upon the moduli of the q_i's. As stated earlier, Shanks referred to a divergent sequence as "diverging from s" (1955, p. 7).

Shanks' proposed method of approximating s was to solve the $2k + 1$ system of nonlinear equations,

$$x_r = B_{k,n} + \sum_{i=1}^{k} a_i \, q_i^r, \quad n \leq r \leq n + 2k,$$

for $B_{k,n}$ with $a_i, q_i, i = 1, \ldots, k$, the rest of the unknown values. Here $B_{k,n}$ is taken to be an approximation of s. Shanks determined that the solution $B_{k,n}$ could be represented in the determinant form

$$B_{k,n} = \frac{\begin{vmatrix} x_n & x_{n+1} & \cdots & x_{n+k} \\ u_n & u_{n+1} & \cdots & u_{n+k} \\ \vdots & \vdots & & \vdots \\ u_{n+k-1} & u_{n+k} & \cdots & u_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ u_n & u_{n+1} & \cdots & u_{n+k} \\ \vdots & \vdots & & \vdots \\ u_{n+k-1} & u_{n+k} & \cdots & u_{n+2k-1} \end{vmatrix}}, \tag{32}$$

$n = 0, 1, \ldots$, and where $u_n = x_{n+1} - x_n$. Therefore, Shanks derived a new sequence $\{B_{k,n}\}$ defined by (32) where k is a nonnegative integer and for which the denominator does not vanish. If the denominator vanishes for $n = m$ and the numerator does not, then $B_{k,m}$ is assigned the value ∞. If both numerator and denominator vanished for $n = m$, then $B_{k,m} = B_{k-1,m}$. He wrote the transforms in operator form as

$$e_k(x_n) = B_{k,n}$$

Shanks called e_k "the k'th order transform of $\{A_n\}$," (Shanks, 1955, p. 2) where $\{A_n\} = \{x_n\}$. There are two transforms that need to be identified. The first one is $k = 0$ where $e_0(x_n) = x_n$. For the second one, letting $k = 1$ we have

$$B_{1,n} = \frac{\begin{vmatrix} x_n & x_{n+1} \\ u_n & u_{n+1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ u_n & u_{n+1} \end{vmatrix}} = \frac{x_n(x_{n+2} - x_{n+1}) - x_{n+1}(x_{n+1} - x_n)}{(x_{n+2} - x_{n+1}) - (x_{n+1} - x_n)}$$

$$= \frac{x_n x_{n+2} - x_{n+1}^2}{x_{n+2} - 2x_{n+1} + x_n}, \quad n = 0, 1, \ldots \tag{33}$$

Comparing (33) with (22) we see that the two right-hand expressions are equivalent. Therefore, the first extrapolated sequence of Aitken's Δ^2 method and Shanks' e_1 transformation are equivalent for scalars. However, this is not true for a second, third, or k^{th} application of the methods. In the next chapter, it will be shown how the sequences of scalars obtained by higher order applications (second, third, etc.) of Aitken's method can be obtained from the e_k transformations. However, the e_k transformation sequences will be derived by a technique different from Shanks' determinant method. As one can clearly see, the larger the value of k, the more complicated the solving for $B_{k,n}$, since two $(k+1)$ x $(k+1)$ determinants must be computed. Hence, a method of obtaining similar results without the use of determinants would be a valuable asset. In the next chapter, such a technique is presented.

CHAPTER VI

WYNN'S EPSILON AND MODIFIED EPSILON METHODS

Theory for Scalar Epsilon Method

Wynn (1956) derived the epsilon (ε) algorithm to accelerate a sequence of scalars. His simple algorithm effected Shanks' $e_k(x_n)$ transformation without the use of determinants. Wynn showed that he could calculate $e_1(x_n)$ directly from the elements of $\{x_n\}$ and $e_i(x_n)$ directly from the $e_{i-1}(x_n)$ elements and values determined from the $e_{i-2}(x_n)$ elements, $i = 2, 3, \ldots$ He used the symbol ε_k^n to represent his new values, where the k subscript refers to a column number and the n superscript refers to a diagonal number. His method will be derived for both scalars and vectors.

For the scalar case, the values ε_k^n are determined from a given sequence by setting the initial two column values as

$$\varepsilon_{-1}^n = 0, \quad \varepsilon_0^n = x_n, \quad n = 0, 1, \ldots,$$

and by the relationship

$$\varepsilon_{k+1}^n = \varepsilon_{k-1}^{n+1} + \left(\varepsilon_k^{n+1} - \varepsilon_k^n\right)^{-1}, \quad k, n = 0, 1, \ldots \tag{34}$$

The quantities ε_k^n may be arranged as shown in Figure 13 (page 45). Note that the four quantities of (34) are located at the four corners of a lozenge, as indicated for $n = 0, k = 2$ in Figure 13. Therefore, a quick way of remembering how to find the right side entry of the lozenge is

$$\text{Right} = \text{Left} + (\text{Bottom} - \text{Top})^{-1}.$$

$$
\begin{array}{ccccccc}
 & & \varepsilon_0^0 = x_0 & & & & \\
\varepsilon_{-1}^1 = 0 & & & & \varepsilon_1^0 & & \\
 & & \varepsilon_0^1 = x_1 & & & \varepsilon_2^0 & \\
\varepsilon_{-1}^2 = 0 & & & & \varepsilon_1^1 & & \varepsilon_3^0 \\
 & & \varepsilon_0^2 = x_2 & & & \varepsilon_2^1 & \\
\varepsilon_{-1}^3 = 0 & & & & \varepsilon_1^2 & & \varepsilon_3^1 \\
 & & \varepsilon_0^3 = x_3 & & & \varepsilon_2^2 & \\
\varepsilon_{-1}^4 = 0 & & & & \varepsilon_1^3 & & \\
 & & \varepsilon_0^4 = x_4 & & & &
\end{array}
$$

Figure 13. Wynn's Epsilon Arrangement

The odd and even numbered columns are quite varied in the information they give. The odd subscript columns normally diverge and give no directly useful information as to the limit or antilimit of the sequence. However, one can obviously see that they are vital as they are used to determine the next even column. The even subscript columns will often converge to the desired limit or antilimit of $\{x_n\}$ and will do so more quickly than the original sequence. However, the key sequence for convergence is the diagonal sequence whose elements are $\varepsilon_{2m}^0, m = 0, 1, \ldots$ This sequence will most often converge not only the quickest, but in some cases, will converge even when each even numbered column sequence diverges.

In order to find ε_6^0, one must first have computed ε_5^0, ε_5^1, and ε_4^1. This eventually leads to the fact that before ε_6^0 can be found, all elements in the first six "cross-diagonals" must be determined. The term cross-diagonals refers to the diagonals of Wynn's epsilon arrangement, Figure 13, which rise as one moves from left to right in the figure and where $n + k$ is constant. Wynn (1964) suggested that to prevent the use of unnecessary storage, computation of the elements can be

made by computing one cross-diagonal at a time, using only the data saved from the previous cross-diagonal.

Therefore, to use his technique as an acceleration method, elements of the arrangement diagram are computed one cross-diagonal at a time until the ε_{2m}^0 element has the desired precision of accuracy as measured by $\|G(\varepsilon_{2m}^0) - \varepsilon_{2m}^0\| < \text{Tol}$, where G is the iteration function. Wynn's theorem relating $\varepsilon_k(x_n)$ (or ε_k^n) to $e_k(x_n)$ follows.

THEOREM 6.1: If $\varepsilon_{2m}(x_n) = e_m(x_n)$ and $\varepsilon_{2m+1}(x_m) = (e_m(u_n))^{-1}$, where $u_n = x_{n+1} - x_n$; then $\varepsilon_{s+1}(x_n) = \varepsilon_{s-1}(x_{n+1}) + (\varepsilon_s(x_{n+1}) - \varepsilon_s(x_n))^{-1}$, $s = 1, 2, \ldots$

Wynn's proof was by mathematical induction (Wynn, 1956, p. 92-94). He proved the equality by showing that

$$\varepsilon_{s+1}(x_n) - \varepsilon_{s-1}(x_{n+1}) \quad \text{and} \quad (\varepsilon_s(x_{n+1}) - \varepsilon_s(x_n))^{-1}$$

are equivalent expressions in determinant form. Hence, in certain cases, the sequence ε_{2m}^0, $m = 0, 1, \ldots$, converges to the limit or antilimit of $\{x_n\}$ and the convergence is more rapid than the original sequence.

For an example of how Wynn's epsilon method works, consider Equation (7). Table 9 (page 47) shows the epsilon arrangement for five iterations. Hence, Wynn's ε method converges in the same number of iterations as Aitken's static method. In addition, this example will be used to illustrate how the scalar sequences obtained by more than one application of Aitken's Δ^2 method can be computed by the use of Wynn's epsilon method. Once column two has been computed, calculate the next two columns as if the second column were column zero. In other words, when computing column three, treat column one as if every element were zero. Repeat this process with columns four, five, and six, etc. When all is done, the even numbered columns will be the same sequences as would be derived by repeated applications

TABLE 9

WYNN'S EPSILON ARRANGEMENT FOR EQUATION (7)

n	ε_0^n	ε_1^n	ε_2^n	ε_3^n	ε_4^n
0	0.5000000				
		3.586790			
1	0.7788008		0.7044777		
		−9.867936		−1141.7298	
2	0.6774630		0.7035942		0.7034663
		28.400377		−8964.4053	
3	0.7126738		0.7034830		0.7034674
		−80.404595		−73073.1050	
4	0.7002567		0.7034693		
		228.937720			
5	0.7046047				

of Aitken's Δ^2 method. Table 10 (page 48) shows the results of this procedure for columns two, three, and four. One may check that the columns labeled $\varepsilon_0^{n'}$ and $\varepsilon_2^{n'}$ are identical to columns two and three of Table 3, which were obtained by applying Aitken's method to this same problem.

Theory for Vector Epsilon Method

Now consider $\{\vec{x}_n\}$ as a sequence of vectors. Before Wynn's algorithm can be applied to a vector sequence, the inverse of the vector $\vec{x} = (x_1, x_2, \ldots, x_m)^T$ must be defined. Wynn (1962) discusses two possible inverses:

(1) Primitive Inverse:

$$\vec{x}^{-1} = \left(x_1^{-1}, x_2^{-1}, \ldots, x_m^{-1}\right)^T, \quad \text{where} \quad x_i \neq 0 \text{ for all } i.$$

For $x_i = 0$, take $x_i^{-1} = 0$.

(2) The Samelson Inverse:

$$\vec{x}^{-1} = \sum_{r=1}^{m} \bar{x}_r \, x_r^{-1} \, (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m)^T,$$

TABLE 10

AITKEN'S Δ^2 SEQUENCES DERIVED FROM WYNN'S ARRANGEMENT

n	$\varepsilon_0^{n'}$	$\varepsilon_1^{n'}$	$\varepsilon_2^{n'}$
1	0.7044777		
		−1131.8619	
2	0.7035942		0.7034669
		−8992.8058	
3	0.7034830		0.7034674
		−72992.7007	
4	0.7034693		

where $\bar{x}_r$ is the complex conjugate of x_r and $\vec{x}$ is not the zero vector. Define the zero vector as the inverse of itself. Samelson's inverse is equivalent to the Moore-Penrose generalized inverse of $\vec{x}$ considered as a m x 1 matrix. Hence, it will always be referred to as such. The primitive inverse ignores the relationship between the scalar sequences of different components. In addition, it will frequently have major problems in that one or more of the reciprocals will be quite large numerically due to a denominator very close to zero. Further more, as far as is known, the primitive inverse seldom gives better results than the generalized inverse (Smith, Ford, Sidi; 1987, p. 223). Therefore, the generalized inverse is more useful and, hence, all work in this study involving inverses of vectors will use generalized inverses.

WYNN'S VECTOR EPSILON ALGORITHM 6.2:

Given the iteration equation $\vec{x}_{n+1} = G(\vec{x}_n)$ and the initial vector $\vec{x}_0$. Define

$$\vec{\varepsilon}_{-1}^{\,n} = \vec{0}, \quad n = 1, 2, \ldots, \quad \vec{\varepsilon}_0^{\,0} = \vec{x}_0, \quad \text{and set} \quad n = 1.$$

Step 1. $\vec{\varepsilon}_0^{\,n} = \vec{x}_n$.

Step 2. For $k = 0$ to $n-1$, find $\vec{\varepsilon}_{k+1}^{\,n-k-1} = \vec{\varepsilon}_{k-1}^{\,n-k} + \left(\vec{\varepsilon}_k^{\,n-k} - \vec{\varepsilon}_k^{\,n-k-1}\right)^{-1}$.

Step 3. If n is even and $\|\vec{\varepsilon}_n^0 - \vec{\varepsilon}_{n-2}^2\| < \text{Tol}$ or if n is odd and $\|\vec{\varepsilon}_{n-1}^1 - \vec{\varepsilon}_{n-3}^3\| < \text{Tol}$, then go to step 4; otherwise, $n = n + 1$ and go to step 1.

Step 4. Find $\vec{y} = G(\vec{\varepsilon}_n^0)$ if n is even or $\vec{y} = G(\vec{\varepsilon}_{n-1}^1)$ if n is odd. If $\|\vec{y} - \vec{\varepsilon}_n^0 \text{ (or } \vec{\varepsilon}_{n-1}^1)\| < \text{Tol}$, then stop; otherwise, n=n+1 and go to step 1.

Through the work of Cheng and Hafez (1959), the epsilon method can be modified to make a semi-dynamic model. Using only the initial vector and the first two iterates, the first extrapolated term, $\vec{\varepsilon}_2^0$, is found by Equation (34). Using this vector as a new initial vector, two new iterates are generated and another extrapolated vector determined. This pattern is continued until an iteration has the desired precision of accuracy as measured by $\|G(\vec{\varepsilon}_k^n) - \vec{\varepsilon}_k^n\| < \text{Tol}$. A diagram of the model is shown in Figure 14 (page 50). This semi-dynamic model of Wynn's epsilon method is called the Modified Epsilon Method.

MODIFIED VECTOR EPSILON ALGORITHM 6.3:

Given the iteration equation $\vec{x}_{n=1} = G(\vec{x}_n)$ and the initial vector $\vec{x}_0$. Define

$$\vec{\varepsilon}_{-1}^n = 0, \quad n = 0, 1, 2; \quad \vec{\varepsilon}_0^0 = \vec{x}_0; \quad \text{and} \quad \vec{\varepsilon}_0^1 = G(\vec{x}_0).$$

Step 1. Compute $\vec{\varepsilon}_0^2 = G(\vec{\varepsilon}_0^1)$.

Step 2. Compute $\vec{\varepsilon}_1^0$, $\vec{\varepsilon}_1^1$, and $\vec{\varepsilon}_2^0$ by equation (34).

Step 3. Compute $\vec{\varepsilon}_2^1 = G(\vec{\varepsilon}_2^0)$. If $\|\vec{\varepsilon}_2^1 - \vec{\varepsilon}_2^0\| < \text{Tol}$, then stop; otherwise, set $\vec{\varepsilon}_0^0 = \vec{\varepsilon}_2^0$, $\vec{\varepsilon}_0^1 = \vec{\varepsilon}_2^1$, and go to step 1.

The modified epsilon method as described in Algorithm 6.3 and shown in Figure 14 is just one version of several. The method shown is of order one with

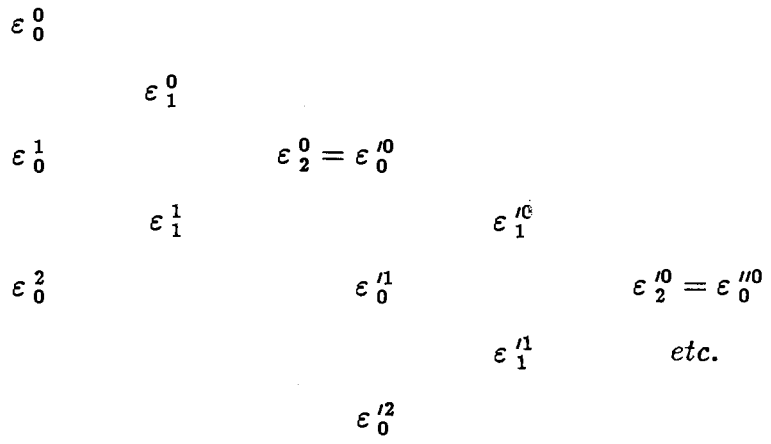

Figure 14. Diagram for Modified Epsilon Method

respect to the initial value $\vec{\varepsilon}_0^0 = \vec{x}_0$. Order two with respect to $\vec{\varepsilon}_0^0$ uses the vectors $\vec{\varepsilon}_0^0$ through $\vec{\varepsilon}_0^4$ to determine $\vec{\varepsilon}_4^0$ and then sets $\vec{\varepsilon}_0^0 = \vec{\varepsilon}_4^0$ until convergence. Higher order methods continue in a similar fashion; however, orders less than four seem to work the best, especially for divergent sequence to keep the iteration from diverging too quickly. As was the case with Aitken's method, the semi-dynamic model is more efficient than the original cross-diagonal static model due to the time and storage required to compute the triangular arrangement for problems of large dimension.

Other variations start with a different element of the original sequence as the initial value. Here, a desired number of elements, say i, of the sequence are skipped and the process begins with $\vec{\varepsilon}_0^i$. This procedure is useful if the multiplicity of the eigenvalue zero is known, even though exact zero eigenvalues are rare in practical application. Skipping the number of elements equal to the multiplicity of the eigenvalue zero will result in faster convergence. Another more practical purpose for skipping a set number of iterates is when under-relaxation is used to move various eigenvalues closer to zero.

Numerical Examples

Consider this original problem of Equation (4) where

$$A = \begin{bmatrix} 1.0 & 0.1 \\ -0.5 & 0.4 \end{bmatrix} \quad \text{and} \quad \vec{b} = (1.2, -2.0)^T. \tag{35}$$

The solution to this problem is $(10.4, -12.0)$. Using the iterative Equation (5) and the initial vector (1,1), Wynn's vector epsilon method determines the solution with only four iterations. Table 11 (page 52) shows the Euclidean norm of the error vectors for the even numbered columns. Remember that the even numbered columns in Wynn's arrangement are the only valid sequences. By checking the rate of convergence of the first column, one can easily see that Wynn's method took a very slow converging sequence and accelerated it tremendously. Table 12 (page 52) shows the results of applying the semi-dynamic (modified) model to problem (35). Results are shown for orders one, two, and three.

Wynn conjectured and McLeod (1971), Theorem 6.4, and Gekeler (1972), Theorem 6.5, proved the following theorems which were thoroughly discussed by Brezinski (1974).

THEOREM 6.4: Let the relation

$$\sum_{i=0}^{k} a_i \, \vec{x}_{n+i} = \vec{s} \sum_{i=0}^{k} a_i, \quad n = 0, 1, \ldots$$

hold for the initial values, where the coefficients a_i are real, $a_k \neq 0$, $\vec{s}$ and $\vec{x}_n$ are m-dimensional vectors over the complex numbers, and the vectors $\vec{\varepsilon}_r^{\,n}$ are determined by (34) and exist for $n + r \leq 2k$. Then

$$\vec{\varepsilon}_{2k}^{\,n} = \vec{s} \text{ for every } n \text{ if } \sum_{i=0}^{k} a_i \neq 0, \text{ and}$$

$$\vec{\varepsilon}_{2k}^{\,n} = \vec{0} \text{ for every } n \text{ otherwise.}$$

TABLE 11

EUCLIDEAN NORMS OF WYNN'S EVEN NUMBERED COLUMN ERROR VECTORS FOR PROBLEM (35)

n	E_0^n	E_2^n	E_4^n
0	16.042443		
1	12.791403	7.066314	
2	10.710378	5.960512	0.000000
3	9.245948	4.807537	
4	8.127063		

TABLE 12

EUCLIDEAN NORMS OF ERROR VECTORS FOR MODIFIED ε METHOD FOR PROBLEM (35)

n	Euclidean Norms for Order		
	1	2	3
0	16.042443	16.042443	16.042443
1	12.791403	12.791403	12.791403
2	10.710378	10.710378	10.710378
3	6.127186	9.245948	9.245948
4	5.399512	8.127063	8.127063
5	2.481779	0.000000	7.217934
6	2.078020		6.448139
7	1.188792		0.000000

THEOREM 6.5: If the vector ε-algorithm is applied to vectors produced by the linear system (4) where A is a real matrix such that $(I - A)$ is nonsingular, then

$$\vec{\varepsilon}_{2k}^{\,0} = (I - A)^{-1}\vec{b} = \vec{s}, \tag{36}$$

where k is the degree of the monic minimal polynomial of A with respect to the vector $\vec{x}_0 - \vec{s}$; that is, k is the smallest integer such that there exists the polynomial

$$p(y) = \sum_{i=0}^{k} p_i\, y^i, \quad p_k = 1$$

where

$$p(A)(\vec{x}_0 - \vec{s}) = \vec{0}.$$

Equation (36) can be generalized to

$$\vec{\varepsilon}_{2(k-q)}^{\,n+q} = \vec{s},$$

where $0 \leq q \leq r$ for r equal to the multiplicity of the eigenvalue zero of the matrix A (Brezinski, 1974).

The significance of this fact will primarily be seen in later chapters as we look at other acceleration techniques. However, the results of Problem (35) shown in Table 11 illustrate this principle. It can be shown that

$$p(y) = y^2 - 1.4y + 0.45$$

is the minimal polynomial of A with respect to $\vec{x}_0 - \vec{s}$. Hence, $k = 2$ and, therefore, $\vec{\varepsilon}_4^{\,0}$ should be equal to $\vec{s}$ if rounding errors are not considered. The minimal polynomial of the matrix A will also be considered more in later chapters.

CHAPTER VII

THE MINIMAL POLYNOMIAL EXTRAPOLATION METHOD

Theoretical Aspect

From this point on the focus of this study will center on vector sequences only. The first method to consider is a method developed by Cabay and Jackson (1976). They derived a polynomial extrapolation method for finding the limit (antilimit) of a vector sequence $\{\vec{x}\}$ governed by the linear iteration (5). They assume that $(I-A)^{-1}$ does exist so that the limit is the unique solution of equation (4).

The key item in their method is the minimal polynomial $p(y)$ of A which annihilates $\vec{u}_0 = \vec{x}_1 - \vec{x}_0$; in other words, p is that unique monic polynomial of least degree such that

$$p(A)\vec{u}_0 = \vec{0}. \tag{37}$$

Therefore, their technique is referred to as the minimal polynomial extrapolation (MPE). To derive their algorithm, let $\vec{s}$ be the solution of (4) and define

$$\vec{u} = \vec{s} - \vec{x}_0 \quad \text{and} \quad \vec{u}_n = \vec{x}_{n+1} - \vec{x}_n. \tag{38}$$

Hence,

$$\begin{aligned} &\vec{u}_{n+1} = A\vec{u}_n,\ (I-A)\vec{u} = \vec{u}_0, \quad \text{and} \\ &(I-A)(\vec{s} - \vec{x}_j) = \vec{u}_j. \end{aligned} \tag{39}$$

Let

$$S_p(y) = \sum_{i=0}^{k-1} \left(\sum_{j=i+1}^{k} c_j \right) y^i, \quad \text{where}$$

$$p(y) = \sum_{j=0}^{k} c_j\, y^j, \quad c_k = 1, \tag{40}$$

Hence,

$$\begin{aligned}(I-A)S_p(A) &= [(c_1+\ldots+c_k)I+(c_2+\ldots+c_k)A+\ldots+c_kA^{k-1}]- \\ &\quad \{(c_1+\ldots+c_k)AA^0+(c_2+\ldots+c_k)AA^1+\ldots+c_kAA^{k-1}\} \\ &= (c_1+c_2+\ldots+c_k)I-(c_1+c_2+\ldots+c_k)A^k \\ &= p(I)-p(A).\end{aligned}$$

Assuming $p(A)$ annihilates $\vec{u}_0$, then

$$(I-A)S_p(A)\vec{u}_0 = (p(I)-p(A))\vec{u}_0 = p(I)\vec{u}_0.$$

Using equations (38) and (39) and simplifying give

$$\vec{u} = \vec{s}-\vec{x}_0 = (I-A)^{-1}\vec{u}_0 = p(1)^{-1}S_p(A)\vec{u}_0. \tag{41}$$

However, it is also true that

$$\vec{u}_i = \vec{x}_{i+1}-\vec{x}_i = A\vec{x}_i - A\vec{x}_{i-1} = \ldots = A^i\vec{u}_0. \tag{42}$$

Hence, (41) becomes

$$\begin{aligned}\vec{s}-\vec{x}_0 &= p(1)^{-1}\left[\sum_{i=0}^{k-1}\left(\sum_{j=i+1}^{k}c_j\right)A^i\right]\vec{u}_0 \\ &= p(1)^{-1}\left[\sum_{i=0}^{k-1}\left(\sum_{j=i+1}^{k}c_j\right)\vec{u}_i\right].\end{aligned}$$

Therefore, solving for $\vec{s}$ gives

$$\vec{s} = \vec{x}_0 + \left[\sum_{i=0}^{k-1}\left(\sum_{j=i+1}^{k}c_j\right)\vec{u}_i\right]/p(1) \tag{43}$$

Since $(I-A)^{-1}$ exists, A has no eigenvalue at unity and thus $p(1)\neq 0$. Therefore $\vec{u}$ can be found after $k+1$ iterations, provided the annihilating polynomial exists. Cabay and Jackson made no attempt to produce the minimal polynomial $p(y)$. Instead, they found an almost-annihilating polynomial $a(y)=\sum_{i=0}^{k'}a_i\,y^i$, $a(1)\neq 0$

and $a_{k'} = 1$, such that $\sum_{i=0}^{k'} a_i \, \vec{u}_i = \delta$ for δ relatively small. Once the a_i's are found, the extrapolated vector is determined by calculating $\vec{u}$ and adding the result to $\vec{x}_0$. One method for solving the a_i's is to minimize the norm by using a least squares technique. Hence, we solve

$$\sum_{i=0}^{k'-1} a_i \, \vec{u}_i = -\vec{u}_{k'}.$$

Another approach (Sidi, 1986, and Sidi, Ford, and Smith, 1986) in developing the MPE method is to begin with the minimal polynomial, $p(y)$, of A with respect to $\vec{u}_0$, Equation (40). Using (37) and (42),

$$\vec{0} = p(A)\vec{u}_0 = \sum_{j=0}^{k} c_j \, A^j \, \vec{u}_0 = \sum_{j=0}^{k} c_j \, \vec{u}_j.$$

So the unknown coefficients of the polynomial p are $c_k = 1$ and the components of the vector $\vec{c} = (c_0, c_1, \ldots, c_{k-1})^T$ which solves the system of equations

$$U\vec{c} = -\vec{u}_k, \tag{44}$$

where U is the m x k matrix defined by

$$U = [\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_{k-1}].$$

If $k < m$, then there are more equations in the system than unknowns; however, we have shown consistency. Therefore, the unique solution can be found.

We now express any element $\vec{x}_j$ in terms of $\vec{x}_0$ by using the fact that if $\vec{s}$ is the solution of (4) then $\vec{s} = (I - A)^{-1}\vec{b}$. So

$$\begin{aligned} \vec{x}_j &= A^j\vec{x}_0 + (I + A + \ldots + A^{j-1})\vec{b} \\ &= A^j\vec{x}_0 + (I - A^j)(I - A)^{-1}\vec{b} = A^j\vec{x}_0 + (I - A^j)\vec{s} \\ &= A^j(\vec{x}_0 - \vec{s}\,) + \vec{s}. \end{aligned}$$

Hence, $A^j(\vec{x}_0 - \vec{s}\,) = \vec{x}_j - \vec{s}$.

Smith, Ford, and Sidi (1987) showed that the minimal polynomial of A with respect to the vector $\vec{x}_j - \vec{s}$ is the same polynomial as that for $\vec{u}_j$, for every j, and thus true for $j = 0$. (Hence, $p(y)$ is the same minimal polynomial that was discussed in Chapter VI when it was shown that $\vec{\varepsilon}^{\,0}_{2k} = \vec{s}$ for k the degree of $p(y)$.) So

$$\begin{aligned} \vec{0} &= \sum_{j=0}^{k} c_j A^j \vec{u}_0 = \sum_{j=0}^{k} c_j A^j (\vec{x}_0 - \vec{s}) = \sum_{j=0}^{k} c_j (\vec{x}_j - \vec{s}) \\ &= \sum_{j=0}^{k} c_j \vec{x} - \vec{s} \left(\sum_{j=0}^{k} c_j \right). \end{aligned} \tag{46}$$

Since unity is assumed not to be an eigenvalue of A, $\sum_{j=0}^{k} c_j = p(1) \neq 0$. Hence, $\vec{s}$ is computed directly from (46).

The above proof can also be shown for the starting vector $\vec{x}_n$ instead of $\vec{x}_0$. Therefore, a theorem for any $k+1$ consecutive terms of a sequence was proven by Smith, Ford, and Sidi (1987).

THEOREM 7.1: For any $k+1$ consecutive terms of the sequence $\{\vec{x}\}$, say $\vec{x}_n, \vec{x}_{n+1}, \ldots \vec{x}_{n+k}$, we have

$$\sum_{j=0}^{k} c_j \vec{x}_{n+j} = \vec{s} \left(\sum_{j=0}^{k} c_j \right). \tag{47}$$

where $c_j, j = 0, \ldots, k$, are defined by equation (40).

If (43) is rewritten in terms of the $\vec{x}_j$'s, we have

$$\vec{s} = \sum_{j=0}^{k} l_j \vec{x}_j, \quad \text{where} \quad l_j = c_j / \left(\sum_{i=0}^{k} c_i \right).$$

If r is the multiplicity of the eigenvalue zero, then r terms on each side of (47) are zero. Therefore, if r is known or suspected to be positive, there is an advantage of starting the $k+1$ consecutive terms at $\vec{x}_n, n > 0$, instead of $\vec{x}_0$. Preferably, we should start at $\vec{x}_r$.

To this point, the discussion has dealt with finding the solution of a linear system. For linear problems of large dimension, the degree of the minimal polynomial

may be difficult to determine. For nonlinear problems, the annihilating polynomial changes for each iteration and the limit cannot be obtained in a finite number of iterations. Therefore, a small value of k is chosen and Equation (43) is used as a model for approximating the solution. Repeating the process until convergence with each new initial vector set equal to the last computed extrapolated vector, a semi-dynamic method is developed for solving large linear problems and nonlinear problems. Even for small linear problems where the exact k is known, rounding errors may prevent obtaining the solution to the desired accuracy on the first extrapolation. Hence, the semi-dynamic model is used for all types of problems.

MINIMAL POLYNOMIAL EXTRAPOLATION (MPE) ALGORITHM 7.2:

Given the sequence $\vec{x}_{n+1} = G(\vec{x}_n)$, the initial value $\vec{x}_0$, and the positive integer k.

Step 1. Generate $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_{k+1}$ by the function G.

Step 2. Compute U and $\vec{u}_k$ by use of (45) and (38).

Step 3. Compute $\vec{c}$ from (44) and set $c_k = 1$.

Step 4. Compute $\vec{s}$ from (47) where $\vec{x}_{n+j} = \vec{x}_j$.

Step 5. Generate $\vec{y} = G(\vec{s})$. If $\|\vec{y} - \vec{s}\| < \text{Tol}$, then stop; otherwise, set $\vec{x}_0 = \vec{s}$ and go to step 1.

Theoretical Application to Numerical Problems

Let us consider Problem (35) again. In Chapter VI, it was shown that Wynn's vector epsilon method obtains the solution in four ($2k$) iterations since the degree of the monic minimal polynomial is $k = 2$. Since this example is a linear problem, according to theory, the MPE method should compute the solution in $k + 1 = 3$ iterations. This is the case as the solution, $(10.4, -12.0)$, is found to six decimal place accuracy after only three iterations and one extrapolation. Therefore, in theory, whereas Wynn's Vector Epsilon method requires $2k$ iterations to obtain $\vec{s}$,

the MPE method can obtain it in only $k+1$ iterations. If k is large, one can see a major advantage of the MPE method. However, the larger the value of k, the larger the dimension of our matrix U in determining the coefficient vector, $\vec{c}$. If the MPE method is applied to (35) with $k = 1$, the extrapolation procedure still converges; however, it takes 78 iterations to determine the solution to six decimal places.

For larger dimensional problems, k usually should be chosen such that $2 \leq k \leq 5$. There are two reasons for this restriction of k. The first reason is the storage space and the time required in working with large dimensional matrices. The second reason is given by Anderson (1965, p. 555), "the power of an iterative method increases slowly with degree for $M > 3$ since the "early," poor approximations are not samples of significant information content ..." (Here Anderson's M refers to k). Results found from the numerical test problems in Chapter XI show that no one k is the best choice for all problems.

Variations for Convergent/Divergent Sequences

Since Equation (47) uses only the first k iterates of the generated sequence, the most accurate approximation of the limit for a convergent sequence, $\vec{x}_{k+1}$, is not used. What would be helpful is to modify the algorithm so that the most current estimates are used. Chandler (1988) suggests the following model for the linear equation (4).

Let the finite sequence $S = \{\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_{k+1}\}$ be the $k+2$ generated vectors that are used to determine the coefficient vector $\vec{c}$ of (44). As shown in Chapter II, Equation (13), there exist constants a_i and $q_i, i = 1, \ldots, m$, where m is the order of A, such that

$$\vec{x}_n = \vec{s} + \sum_{i=1}^{m} a_i \, \vec{v}_i \, q_i^n, \quad n = 0, 1, \ldots, k+1.$$

Define $b_i = a_i q_i^{k+1}$ and $p_i = 1/q_i$, $i = 1, \ldots, m$. Then

$$\vec{x}_n = \vec{s} + \sum_{i=1}^{m} b_i \, \vec{v}_i \, p_i^{k+1-n}, \quad n = 0, 1, \ldots, k+1.$$

Then there exist a sequence $T = \{\vec{x}\,'_0, \vec{x}\,'_1, \ldots, \vec{x}\,'_{k+1}\}$ such that

$$\vec{x}\,'_r = \vec{s} + \sum_{i=0}^{m} b_i \, \vec{v}_i \, p_i^r, \quad r = 0, \ldots, k+1,$$

where $\vec{x}\,'_r = \vec{x}_{k+1-r}$. If $|q_i| < 1$ for all i, then $|p_i| > 1$. Hence, $\vec{s}$ is the limit of S and the antilimit of T. If the moduli of all q_i's are greater than unity, then $|p_i| < 1$ for all i. Therefore, $\vec{s}$ is the antilimit of S and the limit of T. Otherwise, $\vec{s}$ is the antilimit of both S and T.

If the generated sequence S diverges, then the most accurate estimate of the antilimit is $\vec{x}_0$. Therefore, the MPE method should be applied to the sequence S in that case to approximate the antilimit so that (47) will be the sum of the most accurate estimate plus a small error. If S converges, then the most accurate estimate of the limit is $\vec{x}_{k+1}$; hence, sequence T should be used in this case to approximate the limit. Even though the above theory was developed for the linear case, it can be used as a model for estimating the solution of a nonlinear problem. A comparison of the two techniques will be shown in Chapter XI.

CHAPTER VIII

THE REDUCED RANK EXTRAPOLATION METHOD

Theory for the Full Rank Extrapolation Method

Henrici (1964) set forth to extend Aitken's formula for systems of equations. His goal was to estimate the limit of a sequence of m-dimensional vectors. His formula contains two m x m matrices, of which one involves an inverse. For large problems, solving a large linear system is not exactly helpful. However, the theoretical application of his work is valid. Mešina (1977) and Eddy (1979) modified Henrici's basic formula by reducing the dimension of the linear system to a value that is reasonable for computation. Eddy referred to his method as the Reduced Rank Extrapolation (RRE) method.

Before their methods and formulas are derived, some basic definitions are needed which will be used throughout this chapter. Some of the definitions have already been used in previous chapters; however, they are mentioned again for completeness.

Let $\{\vec{x}_n\}$ be an m-dimensional vector sequence generated by the Equation (3) such that $\vec{s}$ is a solution of (2). Define the first and second difference vectors as $\vec{u}_n = \vec{x}_{n+1} - \vec{x}_n$ and $\vec{v}_n = \vec{u}_{n+1} - \vec{u}_n$, respectively. The following m x k $(1 \leq k \leq m)$ rectangular matrices are very valuable in the development of the theory. Their columns are first or second difference vectors. Define

$$U_n = [\vec{u}_n, \vec{u}_{n+1}, \ldots, \vec{u}_{n+k-1}] \quad \text{and}$$

$$V_n = [\vec{v}_n, \vec{v}_{n+1}, \ldots, \vec{v}_{n+k-1}].$$

Also, define $A = A(\vec{s})$ to be the Jacobian matrix of the function G taken at the solution $\vec{s}$. The Jacobian matrix is defined as

$$a_{ij} = \frac{\partial g_i(\vec{x})}{\partial x_j}, \quad i,j = 1,\ldots,m, \quad \text{where} \quad G(\vec{x}) = \begin{bmatrix} g_1(\vec{x}) \\ \vdots \\ g_m(\vec{x}) \end{bmatrix},$$

a_{ij} is the (i,j) component of the matrix A, and $\vec{x} = (x_1, x_2, \ldots, x_m)^T$. Henrici (1964, p. 104) showed that $\vec{e}_{n+1} = A(\vec{s})\vec{e}_n + O(\|\vec{e}_n\|)^2$ where $\vec{e}_n = \vec{x}_n - \vec{s}$ and $O(\|\vec{e}_n\|^2)$ denotes a quantity bounded by $C\|\vec{e}_n\|^2, C$ an integer. If we assume that this error formula is exact with $O(\|\vec{e}_n\|) = \vec{0}$ for finite values of n, then $\vec{x}_{n+1} - \vec{s} = A(\vec{x}_n - \vec{s})$. Then the following relationships are satisfied:

$$\begin{aligned} \vec{u}_n &= A\vec{u}_{n-1} = \cdots = A^n\vec{u}_0, \ \vec{v}_n = (A-I)\vec{u}_n, \\ V_n &= U_{n+1} - U_n = (A-I)U_n, \quad \text{and} \qquad (48) \\ U_{n+1} &= AU_n. \qquad (49) \end{aligned}$$

If $k = m$, then U_n and V_n are square matrices. Assuming that U_n is nonsingular, Equation (49) gives

$$A = U_{n+1}U_n^{-1}. \qquad (50)$$

Since A is the Jacobian matrix of G, then

$$\vec{x}_{n+1} - \vec{s} = A\vec{x}_n - A\vec{s}.$$

This implies that

$$(A-I)\vec{s} = A\vec{x}_n - \vec{x}_{n+1} + \vec{x}_n - \vec{x}_n = (A-I)\vec{x}_n - \vec{u}_n.$$

Therefore, if unity is not an eigenvalue of $(A-I)$, then $(A-I)^{-1}$ exists and

$$\vec{s} = \vec{x}_n - (A-I)^{-1}\vec{u}_n. \qquad (51)$$

Using (50), we change $(A-I)^{-1}$ in the following fashion:

$$\begin{aligned}(A-I)^{-1} &= (U_{n+1}U_n^{-1}-I)^{-1} = \left[(U_{n+1}-U_n)U_n^{-1}\right]^{-1} \\ &= \left[V_nU_n^{-1}\right]^{-1} = U_nV_n^{-1}.\end{aligned}$$

Substituting into (51), the extrapolation formula is

$$\vec{s} = \vec{x}_n - U_nV_n^{-1}\vec{u}_n. \tag{52}$$

Though the development of (52) is due to Henrici (1964), some of the notation used is due to Smith, Ford, and Sidi (1987). As was the case with the MPE method, Equation (52) should be applied to the sequence $S = \{\vec{x}_n, \vec{x}_{n+1}, \ldots, \vec{x}_{n+k+1}\}$ if the sequence S diverges. If S is a convergent sequence, then the sequence $T = \{\vec{x}_{n+k+1}, \vec{x}_{n+k}, \ldots, \vec{x}_n\}$ is used instead of S.

Eddy (1979) derived the same extrapolation formula as follows:

$$\begin{aligned}\vec{s} &= \lim_{n\to\infty}\vec{x}_n = \lim(\vec{x}_0 + (\vec{x}_1-\vec{x}_0) + (\vec{x}_2-\vec{x}_1) + \ldots) \\ &= \lim(\vec{x}_0 + \vec{u}_0 + \vec{u}_1 + \ldots) \\ &= \vec{x}_0 + \lim(I + A + A^2 + \ldots)\vec{u}_0 \\ &= \vec{x}_0 + (I-A)^{-1}\vec{u}_0. \end{aligned} \tag{53}$$

Equation (48) then yields

$$\begin{aligned}\vec{s} &= \vec{x}_0 - U_0V_0^{-1}\vec{u}_0, \quad \text{or} \\ \vec{s} &= \vec{x}_0 + U_0Z \quad \text{and} \quad 0 = \vec{u}_0 + V_0Z,\end{aligned}$$

which matches Equation (52) for $n = 0$. Therefore, for a linear system and with no rounding errors, $\vec{s}$ is computed exactly. For a nonlinear problem, the limit cannot be obtained for a finite value of k, but Equation (52) is used as a model for estimating $\vec{s}$, the solution of the problem. Since the extrapolated vector will be only an estimate of $\vec{s}$, a repeating process with a new initial vector set equal to the extrapolated

vector is used to establish a semi-dynamic procedure. This method will be referred to as the Full Rank Extrapolation (FRE) method.

The FRE method is an acceleration technique that requires $m+1$ iterations, where m is the dimension of the vector space of the problem. There is one obvious problem with this method: if m is large, then to obtain $m+1$ iterations before we can even apply the extrapolation technique defeats the purpose of accelerating. Though Henrici (1964) indicated that the technique is still valid for values much smaller than m, Eddy (1979) proved this fact.

Theory for the Reduced Rank Extrapolation Method

Assume that we choose k such that $1 \leq k < m$. Then U and V are now non-square matrices and have no inverses, so Equation (50) does not hold. Therefore, an alternate approach for establishing the basic extrapolation formula is needed. Let the exact limit, $\vec{s}$, in (53) be replaced by the extrapolated value $\vec{s}'$. Define

$$(I-A)^{-1}\vec{u}_0 = U_0 Z. \tag{54}$$

Using (48) and (54), we have

$$\vec{u}_0 = (I-A)U_0 Z = -V_0 Z$$

so that the extrapolated vector can be expressed by

$$\vec{s}' = \vec{x}_0 + U_0 Z \quad \text{and} \quad \vec{0} = \vec{u}_0 + V_0 Z. \tag{55}$$

Solving Equation (55) by the method of least squares gives

$$0 = (V_0^*)\vec{u}_0 + ((V_0^*)V_0)Z.$$

Therefore,

$$Z = -((V_0^*)V_0)^{-1}(V_0^*)\vec{u}_0 = -V_0^+ \vec{u}_0, \tag{56}$$

where V_0^+ is the generalized inverse of V_0.

Substituting (56) into (55) and generalizing give an extrapolation method for $k < m$ and any starting vector $\vec{x}_n$:

$$\vec{s} = \vec{x}_n - U_n V_n^+ \vec{u}_n. \tag{57}$$

As with the MPE and FRE methods, the sequence used in (57) may be the generated sequence, S, or S in reverse order, depending upon whether the iterated sequence diverges or converges, respectively. Eddy (1979) called this method the Reduced Rank Extrapolation (RRE) method. Once again, it is usually best to keep k less than about six for large dimensional problems. From this point on, this procedure will be referred to as the RRE method, regardless of the value of k.

REDUCED RANK EXTRAPOLATION ALGORITHM 8.1:

Given the iteration equation $\vec{x}_{i+1} = G(\vec{x}_i)$, the initial vector $\vec{x}_0$, and the positive integer k.

Step 1. For $i = 0, 1, \ldots, k$, compute $\vec{x}_{i+1} = G(\vec{x}_i)$.

Step 2. For $i = 0, 1, \ldots, k-1$, compute $\vec{u}_i = \vec{x}_{i+1} - \vec{x}_i$ and $\vec{v}_i = \vec{u}_{i+1} - \vec{u}_i$.

Step 3. Define U and V by $U = [\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_{k-1}]$ and $V = [\vec{v}_0, \vec{v}_1, \ldots, \vec{v}_{k-1}]$.

Step 4. Compute $\vec{s} = \vec{x}_0 - UH\vec{u}_0$, where $H = V^{-1}$, if $k = m$, the dimension of the problem; or $H = V^+ = ((V^*)V)^{-1}V^*$, the generalized inverse of V, if $k < m$.

Step 5. If $\|G(\vec{s}) - \vec{s}\| < \text{Tol}$, then stop; otherwise, set $\vec{x}_0 = \vec{s}$ and $\vec{x}_1 = G(\vec{s})$, generate the vectors $\vec{x}_2, \ldots, \vec{x}_{k+1}$ by the iteration equation, and go to step 2.

The computation of Step 4 of Algorithm 8.1 involves the generalized inverse of V. Eddy (1979) and Smith, Ford, and Sidi (1987) suggest that this matrix be

computed in the algorithm. Therefore, the matrix $((V^*)V)^{-1}$ must be determined, which requires costly computer time, especially for larger values of k. I suggest an alternate approach. Let $B = (V^*)V$ and $\vec{y} = (V^*)\vec{u}_0$. Then $UH\vec{u}_0$ in Step 4 can be rewritten as

$$UH\vec{u}_0 = U((V^*)V)^{-1}(V^*)\vec{u}_0 = UB^{-1}\vec{y} = U\vec{z},$$

where $B\vec{z} = \vec{y}$. The vector $\vec{z}$ is found by Gaussian elimination. Since the product $(V^*)V$ will be a symmetric matrix, the amount of computer time is reduced even more.

Numerical Examples

For an example, consider the two-dimensional problem (Henrici, 1964) of finding the solution of the system

$$\begin{aligned} x &= x^2 + y^2, \\ y &= x^2 - y^2, \end{aligned} \tag{58}$$

near the point (0.8,0.4). A quick check will show that the solution of (58) is (0.771845,0.419643) to six decimal places. However, converting (58) into its iteration equations,

$$x_{n+1} = (x_n)^2 + (y_n)^2 \quad \text{and} \quad y_{n+1} = (x_n)^2 + (y_n)^2,$$

the iterative sequence $\{z_n\}$, where $z_n = (x_n, y_n)$, diverges.

Table 13 (page 67) shows infinity norm results of applying the RRE method, $k = 2$, to this problem. The first column gives the norms of each difference vector, $\vec{u}_n$. The norms given in the second column are for the error vectors, $\vec{e}_n$, of each extrapolated vector. Since $k = 2$, there must be $k + 1 = 3$ iterations before the extrapolation technique can be applied. Hence, extrapolated results, column two, are obtained only once every three iterations.

TABLE 13

INFINITY NORMS OF DIFFERENCE AND ERROR VECTORS FOR THE RRE METHOD FOR PROBLEM (58)

n	$\vec{u}_n$	$\vec{e}_n$
0	0.080000	
1	0.070400	
2	0.180224	0.002280
3	0.003916	
4	0.004952	
5	0.009096	0.000023
6	0.000023	
7	0 000047	
8	0.000079	0.000000
9	0.000000	

Table 13 shows that the solution, to six place accuracy, is obtained upon extrapolating after the ninth iteration. However, a tenth iteration is needed to ensure that the ninth difference vector has the desired precision of accuracy for convergence. In addition, column two measures the error vector, which normally cannot be measured since the answer is not known.

Figure 15 (page 68) shows a graph of how the RRE method ($k = 2$) compares with the MPE method ($k = 2$), the modified vector epsilon method (order 2), and Aitken's semi-dynamic method (Jennings' SDM) on Problem (58). The graph plots the logarithm (base 10) of the infinity norm of the difference vector $(\vec{u}_{n-1} = \vec{x}_n - \vec{x}_{n-1})$ as a function of the number of iterations. The results show that

the RRE and MPE methods clearly converge faster then the ε method and Aitken's method. In fact, the graph suggests that the RRE and the MPE are equivalent methods since the results obtained from these two methods are identical. However, this is NOT so. In Chapter X, it will be shown that the two methods are very similar (in fact, their results are identical for some problems, as is the case for this example), but they are not equivalent methods. Probably, the most important fact shown in Figure 15 is that the MPE and RRE methods both produced an accurate fixed point solution from a divergent nonlinear iteration and obtained the results very rapidly.

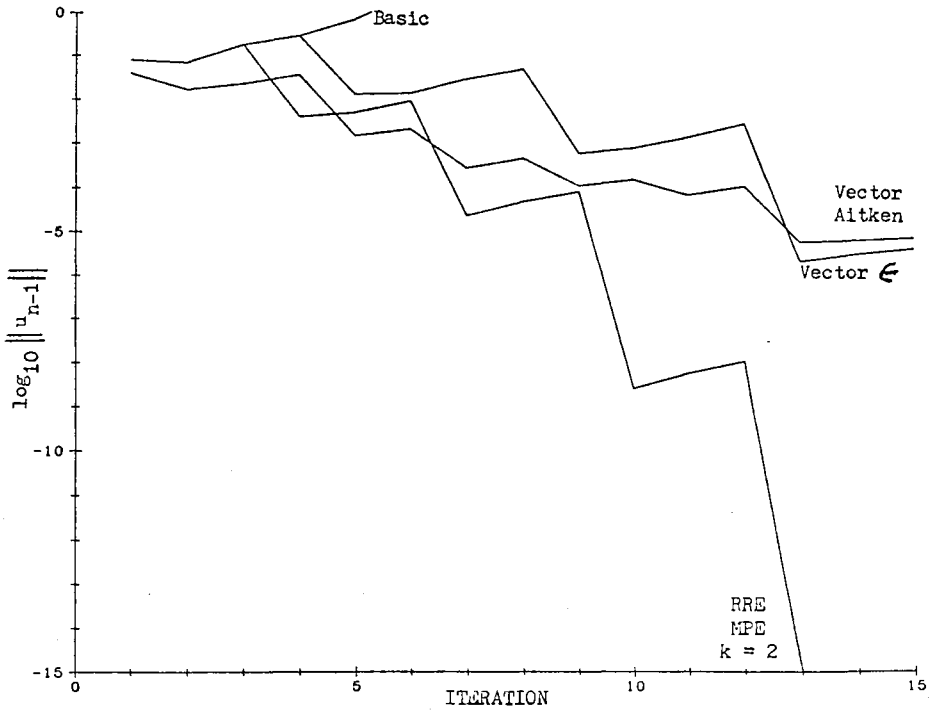

Figure 15. Graph Comparisons for Problem (58)

CHAPTER IX

ANDERSON'S GENERALIZED SECANT ALGORITHMS

Theoretical Development for Secant Methods

Anderson (1965) was motivated by the inability of Aitken's and Wynn's methods to "feed back" into the process in a fully dynamic manner iterates already obtained. He expressed his feelings by stating "The Aitken Δ^2 process, of which the ε-algorithm is a generalization, is considerably less effective if applied statically ... than if applied dynamically ..." (Anderson, 1965, p. 552). His referral to a dynamic process is what has been called a semi-dynamic method in this study. He desired to find a fully dynamic procedure which would accelerate the convergence of a vector sequence. This process is similar to the fully dynamic scalar Aitken algorithm of Irons and Shrive (1987) discussed in Chapter IV. Anderson developed one algorithm and then derived variations from it. He obtained the first algorithm by generalizing the univariate secant method geometrically; see Figure 16 (page 70). Given the equation $x = g(x)$ and the scalars x_0 and x_1, the univariate secant method is

$$x_{n+1} = x_n + B(x_{n-1} - x_n), \quad n = 1, 2, \ldots,$$

where

$$B = \frac{g(x_n) - x_n}{[g(x_n) - g(x_{n-1})] - [x_n - x_{n-1}]}.$$

Generalizing the method for m-dimensions, Wolfe (1959) saw the next element of the sequence as being the solution of a system of nonlinear equations of m secant hyperplanes through $m+1$ points. However, Anderson considered only a hyperline

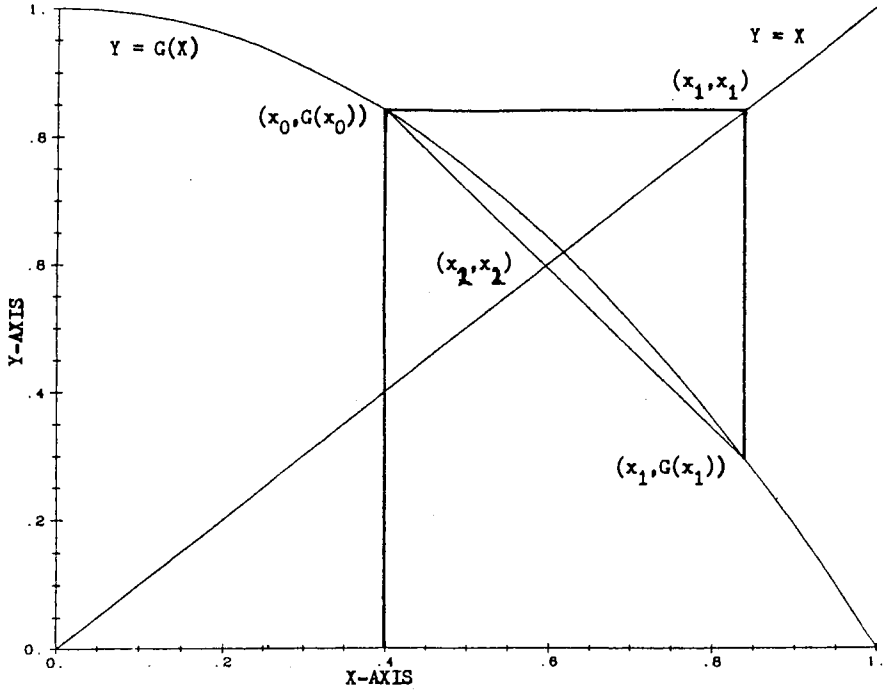

Figure 16. Graph of One Extrapolation of Secant Method

through two points. It must be noted that, in general, a hyperline does not intersect the subspace defining the solution; however, the point chosen is the point that is in some sense "closest" to this subspace.

Now for the development of Anderson's first algorithm. Given the vector sequence $\{\vec{x}_n\}$ and the basic iteration equation $\vec{x}_{n+1} = G(\vec{x}_n)$, Anderson sought two other sequences which converge to the same limit as $\{\vec{x}_n\}$, but more rapidly. He first defined a coupled pair of iterative sequences $\{\vec{y}_n\}$ and $\{\vec{z}_n\}$ by

$$\vec{z}_n = G(\vec{y}_n).$$

Also define the residual vector, $\vec{r}_n$, and the inner product, $(\vec{u}, \vec{v})$, of the two real m-dimensional vectors $\vec{u}$ and $\vec{v}$ by

$$\vec{r}_n = \vec{z}_n - \vec{y}_n \quad \text{and} \quad (\vec{u}, \vec{v}) = \sum_{i=1}^{m} u_i \, v_i \, w_i,$$

respectively, where the weights w_i are positive real numbers and the scalars u_i and v_i are the components of $\vec{u}$ and $\vec{v}$. Define $\vec{u}'_n$ and $\vec{v}'_n$ for the generalized univariate case

$$\begin{aligned} \vec{u}'_n &= \vec{y}_n + Q_n(\vec{y}_{n-1} - \vec{y}_n), \quad \text{and} \\ \vec{v}'_n &= \vec{z}_n + Q_n(\vec{z}_{n-1} - \vec{z}_n). \end{aligned} \tag{59}$$

Also define the "linearized residual" R_n by

$$R_n = 0.5(\vec{v}'_n - \vec{u}'_n, \vec{v}'_n - \vec{u}'_n).$$

If $R = (\vec{x}, \vec{x})$, then from calculus $\frac{\partial R}{\partial Q} = 2\left(\frac{\partial \vec{x}}{\partial Q}, \vec{x}\right)$. Hence, minimizing $\vec{R}_n$ with respect to the parameter Q_n yields

$$\begin{aligned} \frac{\partial R_n}{\partial Q_n} &= \left(\frac{\partial \vec{v}'_n}{\partial Q_n} - \frac{\partial \vec{u}'_n}{\partial Q_n}, \vec{v}'_n - \vec{u}'_n\right) \\ &= (\vec{r}_{n-1} - \vec{r}_n, \vec{v}'_n - \vec{u}'_n) = 0. \end{aligned}$$

Solving for Q_n, we have

$$\begin{aligned} 0 &= (\vec{r}_{n-1} - \vec{r}_n, \vec{r}_n + Q_n[\vec{r}_{n-1} - \vec{r}_n]) \\ &= (\vec{r}_{n-1} - \vec{r}_n, \vec{r}_n) + Q_n(\vec{r}_{n-1} - \vec{r}_n, \vec{r}_{n-1} - \vec{r}_n). \end{aligned}$$

Hence,

$$Q_n = (\vec{r}_n - \vec{r}_{n-1}, \vec{r}_n)/(\vec{r}_n - \vec{r}_{n-1}, \vec{r}_n - \vec{r}_{n-1}). \tag{60}$$

Define the extrapolated vector by

$$\vec{y}_{n+1} = \vec{u}'_n + B_n(\vec{v}'_n - \vec{u}'_n), \quad B_n > 0. \tag{61}$$

According to Anderson the choice of a positive B_n prevents $\vec{y}_{n+1}$ from becoming trapped in the subspace spanned by the previous $\vec{y}_n$ iterates. Usually $B_n = 1$ is most appropriate; however, one must determine the optimum value for B_n empirically. Anderson refers to this algorithm as the "extrapolation algorithm."

Before applying Anderson's extrapolation method, the first two terms of both $\{\vec{y}_n\}$ and $\{\vec{z}_n\}$ are required. Therefore, $\{\vec{y}_n\}$ and $\{\vec{z}_n\}$ usually are initiated by

setting $\vec{y}_0 = \vec{x}_0$ and computing $\vec{z}_0 = G(\vec{y}_0) = \vec{x}_1$. Also set $\vec{y}_1 = \vec{z}_0$ and compute $\vec{z}_1$ by $\vec{z}_1 = G(\vec{y}_1) = G(\vec{x}_1) = \vec{x}_2$. The extrapolation technique is now applied.

For the special case where $m = 1$, $Q_n = \vec{r}_n/(\vec{r}_n - \vec{r}_{n-1})$. Therefore, for $B_n = 0$, the next iterate is

$$\vec{y}_{n+1} = \vec{y}_n + \frac{\vec{z}_n - \vec{y}_n}{(\vec{z}_n - \vec{y}_n) - (\vec{z}_{n-1} - \vec{y}_{n-1})}(\vec{y}_{n-1} - \vec{y}_n).$$

Substituting $\vec{z}_n = G(\vec{y}_n)$ and rearranging give

$$\vec{y}_{n+1} = \vec{y}_n + \frac{G(\vec{y}_n) - \vec{y}_n}{\vec{y}_n - \vec{y}_{n-1} - G(\vec{y}_n) + G(\vec{y}_{n-1})}(\vec{y}_n - \vec{y}_{n-1}),$$

which is the univariate secant method. Hence, Anderson's method is consistent for $m = 1$.

ANDERSON'S EXTRAPOLATION ALGORITHM 9.1

Given the iteration equation $\vec{x}_{n+1} = G(\vec{x}_n)$, the initial vector $\vec{x}_0$, and the sequence $\{B_0, B_1, \ldots\}$.

Step 1. Define $\vec{y}_0 = \vec{x}_0, \vec{z}_0 = \vec{y}_1 = \vec{x}_1, \vec{r}_0 = \vec{z}_0 - \vec{y}_0$, and set $n = 1$.

Step 2. Compute $\vec{z}_n = G(\vec{y}_n)$ and $\vec{r}_n = \vec{z}_n - \vec{y}_n$.

Step 3. Find $Q_n, \vec{u}'_n$, and $\vec{v}'_n$ by (60) and (59).

Step 4. Compute $\vec{y}_{n+1} = \vec{u}'_n + B_n(\vec{v}'_n - \vec{u}'_n)$.

Step 5. If $\|\vec{y}_{n+1} - \vec{y}_n\| < \text{Tol}$, stop; otherwise, increase n by one and go to step 2.

Anderson developed an alternate algorithm he referred to as the "relaxation algorithm." The name was so given because the method defines a relaxation parameter dynamically. Define

$$\begin{aligned} \vec{u}'_n &= \vec{z}_n + Q_n\vec{r}_n, \\ \vec{v}'_n &= \vec{z}_{n-1} + Q_n\vec{r}_{n-1}, \quad \text{and} \\ R_n &= 0.5(\vec{v}'_n - \vec{u}'_n, \vec{v}'_n - \vec{u}'_n). \end{aligned}$$

Minimizing R_n with respect to Q_n yields

$$\frac{\partial R_n}{\partial Q_n} = (\vec{r}_{n-1} - \vec{r}_n, \vec{v}\,'_n - \vec{u}\,'_n) = 0.$$

Hence,

$$\begin{aligned} 0 &= (\vec{r}_{n-1} - \vec{r}_n, (\vec{z}_{n-1} - \vec{z}_n) + Q_n [\vec{r}_{n-1} - \vec{r}_n]) \\ &= (\vec{r}_{n-1} - \vec{r}_n, \vec{z}_{n-1} - \vec{z}_n + Q_n(\vec{r}_{n-1} - \vec{r}_n, \vec{r}_{n-1} - \vec{r}_n). \end{aligned}$$

So

$$Q_n = -(\vec{r}_n - \vec{r}_{n-1}, \vec{z}_n - \vec{z}_{n-1})/(\vec{r}_n - \vec{r}_{n-1}, \vec{r}_n - \vec{r}_{n-1}). \tag{62}$$

Thus define $\vec{y}_{n+1} = \vec{u}'_n$. It should be noted here that Anderson has a typographical error in his article. His equation does not have the negative sign.

ANDERSON'S RELAXATION ALGORITHM 9.2.

Given the iteration equation $\vec{x}_{n+1} = G(\vec{x}_n)$, the initial vector $\vec{x}_0$, and the sequence $\{B_0, B_1, \ldots\}$.

Step 1. Define $\vec{y}_0 = \vec{x}_0, \vec{z}_0 = \vec{y}_1 = \vec{x}_1, \vec{r}_0 = \vec{z}_0 - \vec{y}_0$, and set $n = 1$.

Step 2. Compute $\vec{z}_n = G(\vec{y}_n)$ and $\vec{r}_n = \vec{z}_n - \vec{y}_n$.

Step 3. Determine Q_n by (62).

Step 4. Set $\vec{y}_{n+1} = \vec{u}\,'_n = \vec{z}_n + Q_n\vec{r}_n$.

Step 5. If $\|\vec{y}_{n+1} - \vec{y}_n\| < \text{Tol}$, stop; otherwise, increase n by one and go to step 2.

Anderson discussed two particular variants of the first algorithm. The first is the choice of the metric of the inner product for which R_n is defined. The second variant is for higher degree methods. The higher degree methods are obtained by minimizing a linearized residual, R_n, over subspaces of higher dimensions.

Define, for a positive integer k,

$$\begin{aligned} \vec{u}\,'_n &= \vec{y}_n + \sum_{j=1}^{k} Q_n^j(\vec{y}_{n-j} - \vec{y}_n), \\ \vec{v}\,'_n &= \vec{z}_n + \sum_{j=1}^{k} Q_n^j(\vec{z}_{n-j} - \vec{z}_n), \quad \text{and} \\ R_n &= 0.5(\vec{v}\,'_n - \vec{u}\,'_n, \vec{v}\,'_n - \vec{u}\,'_n). \end{aligned} \tag{63}$$

Minimizing R_n with respect to Q_n^i yields, for $i = 1, 2, \ldots, k$,

$$\sum_{j=1}^{k} (\vec{r}_n - \vec{r}_{n-i}, \vec{r}_n - \vec{r}_{n-j}) Q_n^j = (\vec{r}_n - \vec{r}_{n-i}, \vec{r}_n). \tag{64}$$

Define $\vec{y}_{n+1}$ as in Equation (61).

This algorithm is dynamic, coupled, and can be applied after finding $k+1$ iterations. In addition, Anderson's method can "build up" the degree by being applied for $k = 1, 2$, etc., until the desired value of k is reached. Hence, for any positive integer k, Anderson's higher degree algorithm is a fully dynamic technique which can be applied after only two iterations. As previously mentioned, Anderson states that low-degree cases, limiting k to less than six, usually work best.

ANDERSON'S HIGHER-DEGREE ALGORITHM 9.3.

Given the iteration equation $\vec{x}_{n+1} = G(\vec{x}_n)$, the initial vector $\vec{x}_0$, the positive integer k, and the sequence $\{B_0, B_1, \ldots\}$.

Step 1. For $n = 0$ to k, define $\vec{y}_n = \vec{x}_n, \vec{z}_n = G(\vec{y}_n)$, and

$\vec{r}_n = \vec{z}_n - \vec{y}_n$.

Step 2. Solve the system (64) for Q_n^j.

Step 3. Determine $\vec{u}\,'_n$ and $\vec{v}\,'_n$ by (63).

Step 4. Compute $\vec{y}_{n+1} = \vec{u}\,'_n + B_n(\vec{v}\,'_n - \vec{u}\,'_n)$.

Step 5. If $\|\vec{y}_{n+1} - \vec{y}_n\| < \text{Tol}$, stop; otherwise, compute

$\vec{z}_{n+1} = G(\vec{y}_{n+1}), \vec{r}_{n+1} = \vec{z}_{n+1} - \vec{y}_{n+1}, n = n+1$, and go to step 2.

Numerical Examples

Consider the linear system (4) where A is the tridiagonal matrix whose

$$\begin{aligned} &\text{superdiagonal is } (1,0,1/3,0,0,0),\\ &\text{diagonal is } (1/2,1/2,1,-1,-1/6,1/3,1/3,1/3),\\ &\text{subdiagonal is } (0,0,-1,0,3,0), \text{and}\\ &\vec{b} \text{ is the constant vector } (-1/2,1/2,-1/3,13/6,2/3,-7/3,2/3)^T. \end{aligned} \tag{65}$$

Figure 17 compares the convergence of Anderson's method for $2 \leq k \leq 5$ and the relaxation method with the basic iteration. Convergence, the infinity norm of the difference vector less than 10^{-15}, with no acceleration is obtained in 57 iterations. The best results are obtained for Anderson's method with $k = 4$ and 5. This is not by coincidence since this problem was designed to have a matrix of order seven with a monic minimal polynomial of degree 4:

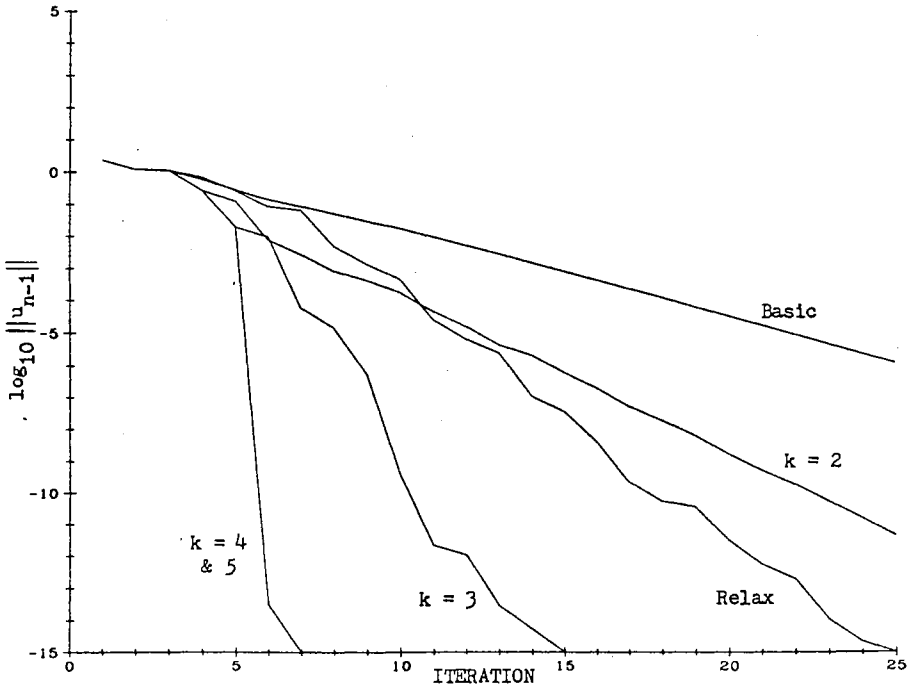

Figure 17. Graph Comparison of Anderson's Methods for Problem (65)

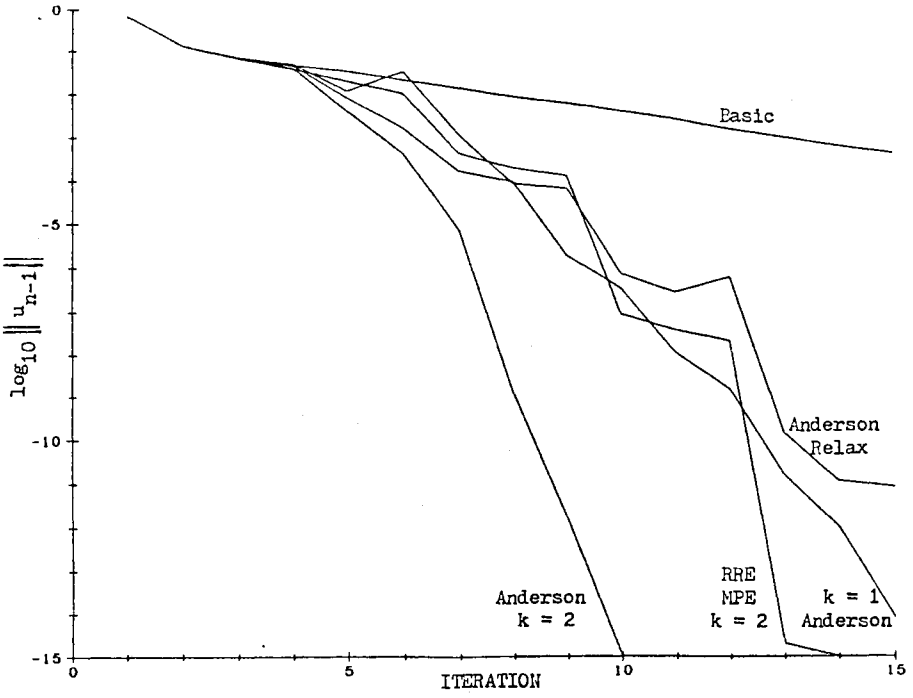

Figure 18. Comparison of Results for Anderson's Methods, the MPE Method, and the RRE Method for Problem (28)

$$p(y) = (y - 1/2)^2(y - 1/3)^2,$$

Hence, the best results are obtained when k is chosen as the degree of the minimal polynomial of A, as was the case for the MPE and RRE methods for linear problems. Another point is that if k is greater than the degree of the minimal polynomial, the sequence will still converge; however, the convergence rate will not improve. Once again, we see that an acceleration method has determined a fixed point solution much faster than the basic iteration.

Recall the familiar examples, Problems (28) and (58). In Figures 18 and 19 (Figure 19 on page 77), the results of these two problems for Anderson's three methods: $k = 1, 2$, and the relaxation method, are compared with the results obtained by the RRE and MPE methods, $k = 2$. Clearly, Anderson's method with

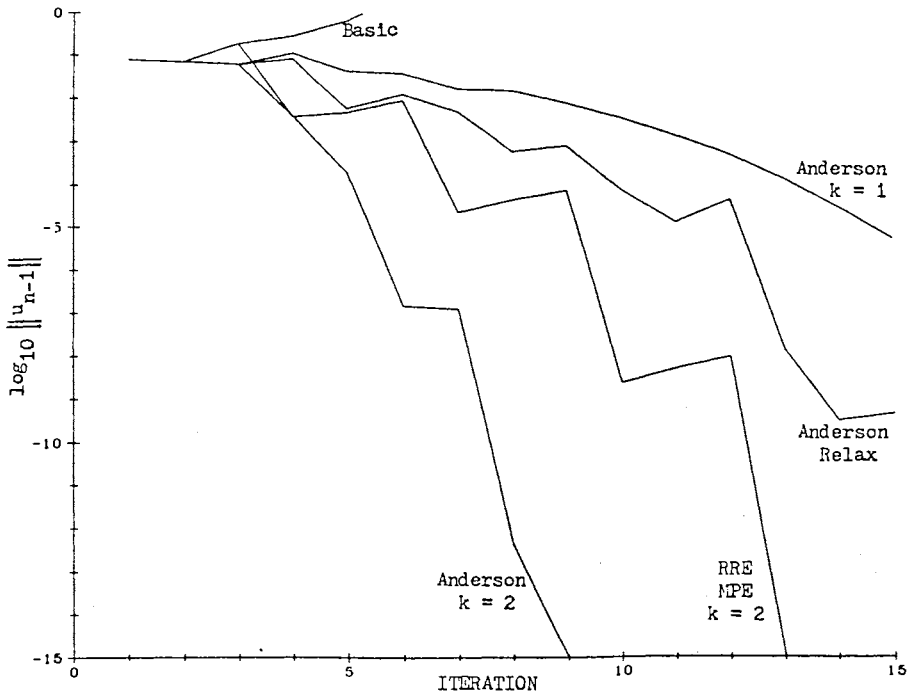

Figure 19. Comparison of Results for Anderson's Methods, the MPE Method, and the RRE Method for Problem (58)

$k = 2$ gives the best results for these examples. This leads to a few questions. Are these results consistent with results of other problems? Are there ways of comparing two or more of these techniques from the theoretical view point? The latter question leads into the next chapter where the first extrapolation method of Anderson, the MPE method, and the RRE method are compared in theory. The former question is saved for Chapter XI.

A final comment concerning Anderson's method needs to be made. Even though he wrote his article in 1965, it has been widely ignored. There were no references found to Anderson's article or to Anderson's method in the research for this thesis. The majority of the articles or papers written on the subject area concern the theoretical and/or numerical comparison of the vector Aitken, the vector ε, the RRE, and/or the MPE methods. One may speculate that the title of his article,

"Iterative Procedures for Nonlinear Integral Equations," may have something to do with the problem since it makes no reference to the subject of acceleration methods. For whatever the reason, the article and the acceleration method have received little attention.

CHAPTER X

THEORETICAL COMPARISONS

Determinant Form for the MPE Method

Now that all the acceleration techniques have been given, they can be compared theoretically. Since the MPE, the RRE, and Anderson's methods all rely on solving a system of linear equations, it seems logical to start with them. The theory presented in this chapter for the MPE and the RRE methods was derived by Sidi (1986) and Sidi, Ford, and Smith (1986). The theory for Anderson's method and the examples, to the best of our knowledge, have not been published.

Define the inner product of two sequence terms $\vec{a}_i$ and $\vec{a}_j$ to be $a_{i,j} = (\vec{a}_i, \vec{a}_j)$, $i, j \geq 0$. Also define the matrix $[t_0, \ldots, t_k]$ by

$$\begin{bmatrix} t_0 & t_1 & \cdots & t_k \\ a_{0,0} & a_{0,1} & \cdots & a_{0,k} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,k} \\ \vdots & \vdots & & \vdots \\ a_{k-1,0} & a_{k-1,1} & \cdots & a_{k-1,k} \end{bmatrix} \tag{66}$$

Denote the determinant of the matrix (66) by $D[t_0, \ldots, t_k]$, and denote by N_i the cofactor of t_i in $D[t_0, \ldots, t_k]$, $i = 0, \ldots, k$. If the elements of the first row of $D\left[\vec{t}_0, \ldots, \vec{t}_k\right]$ are vectors, then the determinant is to be interpreted as

$$D\left[\vec{t}_0, \ldots, \vec{t}_k\right] = \sum_{i=0}^{k} \vec{t}_i \, N_i. \tag{67}$$

It was shown in Chapter VII that given the m-dimensional vector sequence $\{\vec{x}_n\}$, its limit $\vec{s}$, and some positive number $k \leq m$; $\vec{s}$ can be estimated by use of the MPE method by

$$\vec{s}' = \sum_{i=0}^{k} l_i \, \vec{x}_i, \quad \text{where} \quad l_i = c_i / \left(\sum_{j=0}^{k} c_j \right), \tag{68}$$

provided $\sum_{j=0}^{k} c_j \neq 0$. The vector $\vec{c} = (c_0, c_1, \ldots, c_{k-1})^T$ solves the system of equations $U\vec{c} = -\vec{u}_k$, where U and $\vec{u}_k$ are found by (45) and (38) and $c_k = 1$. The vector $\vec{c}$ that satisfies this system satisfies the normal equations

$$\sum_{j=0}^{k-1} (\vec{u}_i, \vec{u}_j) \vec{c}_j = -(\vec{u}_i, \vec{u}_k), \quad 0 \leq i \leq k-1,$$

Consequently, the vector $\vec{l} = (l_0, \ldots, l_k)$ of Equation (68) satisfies the equations

$$\begin{aligned} &\textstyle\sum_{j=0}^{k} l_j = 1, \quad \text{and} \\ &\textstyle\sum_{j=0}^{k} (\vec{u}_i, \vec{u}_j) l_j = 0, \quad 0 \leq i \leq k-1, \end{aligned} \tag{69}$$

provided these equations have a solution. The matrix of Equations (69) is (66) where $t_i = 1$ and $a_{i,j} = (\vec{u}_i, \vec{u}_j)$. Assuming that its determinant is nonzero, then Cramer's rule can be used to write the solution of (69) as

$$l_j = \frac{N_j}{\sum_{i=0}^{k} N_i} = \frac{N_j}{D\,[1, \ldots, 1]}, \quad 0 \leq j \leq k.$$

From (67) and (68) we find

$$\begin{aligned} \vec{s}' &= \sum_{j=0}^{k} l_j \, \vec{x}_j = \sum_{j=0}^{k} \frac{N_j}{D\,[1, \ldots, 1]} \vec{x}_j \\ &= \frac{\sum_{j=0}^{k} \vec{x}_j \, N_j}{D\,[1, \ldots, 1]} = \frac{D\,[\vec{x}_0, \ldots, \vec{x}_k]}{D\,[1, \ldots, 1]}. \end{aligned} \tag{70}$$

Determinant Form for the RRE Method

Now consider the RRE method. From Equation (57) and letting $n = 0$, the first extrapolated vector found by the RRE method is

$$\vec{s}' = \vec{x}_0 - U_0 V_0^{+} \vec{u}_0, \tag{71}$$

where V^+ is the generalized inverse of V. (If $k = m$, then V^+ is the inverse of V.) Define $\vec{q} = (q_0, q_1, \ldots, q_{k-1})^T$ to be the vector which satisfies the system of equations

$$V\vec{q} = -\vec{u}_0. \tag{72}$$

Then (71) can be rewritten as

$$\vec{s}' = \vec{x}_0 + U_0\vec{q} = \vec{x}_0 + \sum_{i=0}^{k-1} q_i \vec{u}_i. \tag{73}$$

As was the case for the MPE method, the $\vec{q}$ that satisfies (72) will also satisfy the normal equations

$$\sum_{j=0}^{k-1}(\vec{v}_i, \vec{v}_j)q_i = -(\vec{v}_i, \vec{u}_0), \quad 0 \le i \le k-1. \tag{74}$$

Substituting $\vec{v}_j = \vec{u}_{j+1} - \vec{u}_j$ for the second component on the left-hand side of (74) and rearranging the equation, we have

$$\begin{aligned}
0 &= (\vec{v}_i, \vec{u}_0) + \sum_{j=0}^{k-1}(\vec{v}_i, \vec{u}_{j+1} - \vec{u}_j)q_j \\
&= (\vec{v}_i, \vec{u}_0) + (\vec{v}_i, \vec{u}_1 - \vec{u}_0)q_0 + \sum_{j=1}^{k-1}(\vec{v}_i, \vec{u}_{j+1} - \vec{u}_j)\vec{u}_j \\
&= (\vec{v}_i, \vec{u}_0) + (\vec{v}_i, \vec{u}_1)q_0 - (\vec{v}_i, \vec{u}_0)q_0 + \sum_{j=1}^{k-1}[(\vec{v}_i, \vec{u}_{j+1})q_j - (\vec{v}_i, \vec{u}_j)q_j] \\
&= (\vec{v}_i, \vec{u}_0)(1 - q_0) + (\vec{v}_i, \vec{u}_1)(q_0 - q_1) + (\vec{v}_i, \vec{u}_2)(q_1 - q_2) \\
&\quad + \ldots + (\vec{v}_i, \vec{u}_{k-1})(q_{k-2} - q_{k-1}) + (\vec{v}_i, \vec{u}_k)q_{k-1} \\
&= (\vec{v}_i, \vec{u}_0)(1 - q_0) + (\vec{v}_i, \vec{u}_k)q_{k-1} + \sum_{j=1}^{k-1}(\vec{v}_i, \vec{u}_j)(q_{j-1} - q_j), \quad 0 \le i \le k-1.
\end{aligned}$$

Define

$$\begin{aligned}
&l_0 = 1 - q_0, l_k = q_{k-1}, \quad \text{and} \\
&l_j = q_{j-1} - q_j, \quad 1 \le j \le k-1.
\end{aligned} \tag{75}$$

It is easy to see that

$$\sum_{j=0}^{k} l_j = 1. \tag{76}$$

Therefore, Equations (75) and (76) establish a one-to-one correspondence between the q_i's and l_j's. Hence, the system of linear equations (74) for the q_i's is equivalent to the linear system

$$\begin{aligned} &\textstyle\sum_{j=0}^{k} l_j = 1, \quad \text{and} \\ &\textstyle\sum_{j=0}^{k} (\vec{v}_i, \vec{u}_j) l_j = 0, \quad 0 \le i \le k-1, \end{aligned} \tag{77}$$

for the l_j's. Substituting $\vec{u}_j = \vec{x}_{j+1} - \vec{x}_j$ on the right-hand side of (73), rearranging, and applying the one-to-one correspondence (75), (73) can be written in the form of (68). The matrix of the system (77), once again, is (66), where $a_{i,j} = (\vec{v}_i, \vec{u}_j)$. Assuming that the determinant of the matrix is nonzero, Cramer's rule can be used to solve the system. The result will be (70). It is important to notice that even though the extrapolated results for both the MPE and RRE methods have the same determinant form, the results are not necessarily the same. This is because the $a_{i,j}$ elements of the matrices are different. For the MPE, they are $(\vec{u}_i, \vec{u}_j)$; whereas, for the RRE, they are $(\vec{v}_i, \vec{u}_j)$.

Determinant Form for Anderson's Methods

Lastly we consider Anderson's higher degree method. Assume that the first extrapolation is not performed until $k+1$ iterations have been obtained. Hence, the first $k+1$ iterations will generate the same sequence, $\vec{x}_0, \ldots, \vec{x}_{k+1}$, as the MPE and RRE methods do. Also assume that $B_n = 1$ for all n. Therefore, using (61) and (63) the extrapolation formula becomes

$$\vec{y}_{n+1} = \vec{v}'_n = \vec{z}_n + \sum_{j=1}^{k} Q_n^j (\vec{z}_{n-j} - \vec{z}_n). \tag{78}$$

As was shown in Chapter IX, to start Anderson's method one sets $\vec{y}_0 = \vec{x}_0, \vec{z}_0 = G(\vec{y}_0) = \vec{x}_1, \vec{y}_1 = \vec{z}_0$, etc., until the necessary number of sequence elements are obtained for extrapolation. Therefore, the two sequences $\vec{y}_0, \ldots, \vec{y}_n$ and $\vec{x}_0, \ldots, \vec{x}_n$

are the same sequences, as are the two sequences $\vec{z}_0, \ldots, \vec{z}_n$ and $\vec{x}_1, \ldots, \vec{x}_{n+1}$. This information will be very valuable later.

The Q_n^j's are found by solving the linear system of equations (64). Since $\vec{r}_n = \vec{z}_n - \vec{y}_n = \vec{x}_{n+1} - \vec{x}_n = \vec{u}_n$, then (64) is equivalent to

$$\begin{aligned}
\sum_{j=1}^{k}(\vec{u}_n - \vec{u}_{n-i}, \vec{u}_n - \vec{u}_{n-j})q_j &= (\vec{u}_n - \vec{u}_{n-i}, \vec{u}_n), \quad 1 \le i \le k, \quad \text{or} \\
\sum_{j=1}^{k}(\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_n - \vec{u}_{n-j})q_j &= (\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_n), \quad 0 \le i \le k-1, \quad \text{or} \\
\sum_{j=0}^{k-1}(\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_n - \vec{u}_{n-j-1})q_{j+1} &= (\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_n), \quad 0 \le i \le k-1,
\end{aligned}$$

where $q_j = Q_n$. The system can be rearranged as follows:

$$\begin{aligned}
0 &= -(\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_n - \vec{u}_{n-1})q_1 + (\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_n) - \\
&\quad \sum_{j=1}^{k-1}(\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_n - \vec{u}_{n-j-1})q_{j+1} \\
&= -(\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_n)q_1 + (\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_n) + (\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_{n-1})q_1 - \\
&\quad \sum_{j=1}^{k-1}(\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_n)q_{j+1} + \sum_{j=1}^{k-1}(\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_{n-j-1})q_{j+1} \\
&= (\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_n)(1 - q_1 - q_2 - \cdots - q_k) + \sum_{j=1}^{k}(\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_{n-j})q_j.
\end{aligned}$$

Set

$$l_0 = (1 - q_1 - q_2 - \cdots - q_k) \quad \text{and} \quad l_j = q_j, \quad 1 \le j \le k.$$

It is easy to verify that $\sum_{j=1}^{k} l_j = 1$. Once again, we have established a one-to-one correspondence between the q_i's, $0 \le i \le k$, and the l_j's, $0 \le j \le k$. Hence, the linear system of equations (64) for the q_i's is equivalent to the linear system

$$\begin{aligned}
&\textstyle\sum_{j=0}^{k} l_j = 1, \\
&\textstyle\sum_{j=0}^{k}(\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_{n-j})l_j = 0, \quad 0 \le i \le k-1,
\end{aligned}$$

for the l_j's. As with the other two methods, the matrix for this system of equations is (66) where

$$a_{i,j} = (\vec{r}_n - \vec{r}_{n-i-1}, \vec{r}_{n-j})$$

$$= (\vec{u}_n - \vec{u}_{n-i-1}, \vec{u}_{n-j}). \tag{79}$$

Therefore, the solution is

$$l_j = \frac{N_j}{D[1,\ldots,1]}, \quad 0 \le j \le k. \tag{80}$$

Rearranging (78) and using the one-to-one correspondence, we have

$$\begin{aligned}
\vec{y}_{n+1} &= \vec{v}'_n = \vec{z}_n + \sum_{j=1}^{k} q_j(\vec{z}_{n-j} - \vec{z}_n) \\
&= \vec{x}_{n+1} + \sum_{j=1}^{k} q_j(\vec{x}_{n+1-j} - \vec{x}_{n+1}) \\
&= (1 - q_1 - \cdots - q_k)\vec{x}_{n+1} + q_1\vec{x}_n + \cdots + q_k\vec{x}_{n+1-k} \\
&= l_0\vec{x}_{n+1} + l_1\vec{x}_n + l_2\vec{x}_{n-1} + \cdots + q_k\vec{x}_{n+1-k} \\
&= \sum_{j=0}^{k} l_j\, \vec{x}_{n+1-j}.
\end{aligned}$$

Therefore, from (67) and (80), and letting $n = k$, we have

$$\begin{aligned}
\vec{y}_{k+1} &= \sum_{j=0}^{k} l_j\, \vec{x}_{k+1-j} = \sum_{j=0}^{k} \frac{N_j \vec{x}_{k+1-j}}{D[1,\ldots,1]} \\
&= \frac{D[\vec{x}_{k+1},\ldots,\vec{x}_1]}{D[1,\ldots,1]}.
\end{aligned} \tag{81}$$

Anderson's extrapolated vector with $B_n = 1$ is quite different from the extrapolated vector for the MPE and RRE methods. First the matrix elements $a_{i,j}$ are, as before, different; but, in addition, we see that the first row of the matrix is different. Instead of the sequence elements $\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_k$; the row contains the elements $\vec{x}_{k+1}, \vec{x}_k, \ldots, \vec{x}_1$.

Consider Anderson's higher degree method with one change: let $B_n = 0$ for all n. Anderson said that this value should never be used, it is used here for theoretical purposes only. Then the extrapolated vector will be

$$\vec{y}_{n+1} = \vec{u}'_n = \vec{y}_n + \sum_{j=1}^{k} Q_n^j(\vec{y}_{n+1} - \vec{y}_n). \tag{82}$$

Since all other factors remain the same as for the case where $B_n = 1$, (82) can be rewritten as

$$\begin{aligned}
\vec{y}_{n+1} &= \vec{u}'_n = \vec{y}_n + \sum_{j=0}^{k} q_j(\vec{y}_{n-j} - \vec{y}_n) \\
&= \vec{x}_n + q_1(\vec{x}_{n-1} - \vec{x}_n) + q_2(\vec{x}_{n-2} - \vec{x}_n) + \cdots + q_n(\vec{x}_{n-k} - \vec{x}_n) \\
&= (1 - q_1 - \cdots - q_k)\vec{x}_n + q_1\vec{x}_{n-1} + q_2\vec{x}_{n-2} + \cdots + q_k\vec{x}_{n-k} \\
&= l_0\vec{x}_n + l_1\vec{x}_{n-1} + l_2\vec{x}_{n-2} + \cdots + q_k\vec{x}_{n-k} \\
&= \sum_{j=0}^{k} l_j\, \vec{x}_{n-j}. \qquad (83)
\end{aligned}$$

As was done in deriving (81), (83) is used to obtain the extrapolation formula

$$\vec{y}_{k+1} = \frac{D\,[\vec{x}_k, \ldots, \vec{x}_0]}{D\,[1, \ldots, 1]}, \qquad (84)$$

where the elements $a_{i,j}$ of Matrix (66) are defined by $(\vec{u}_k - \vec{u}_{k-i-1}, \vec{u}_{k-j})$. Hence, for Anderson's two cases, the extrapolated vectors have the same determinant form with the exception of the first row of the numerator matrices.

Anderson vs RRE Comparison

Let R be the Matrix (66) defined for the RRE case, and let A be the matrix defined for Anderson's case with $B_n = 0$. By interchanging the columns of A, A can be rewritten as $(\vec{x}_0, \ldots, \vec{x}_k)$ where the new $a_{i,j}$ elements are $(\vec{u}_k - \vec{u}_{k-i-1}, \vec{u}_j)$. Since the interchanging of columns will be identical for both the numerator and denominator matrices, the sign change, if any, will cancel out. Hence, the extrapolated vector has not changed.

Since the two matrices, R and A, have identical second components for the $a_{i,j}$ elements, let us consider only the first components which are determined by the variable i, the row variable. By interchanging rows, we can rewrite A in such a way that the j^{th} column of A will have the form

$$\vec{x}_j$$
$$(\vec{u}_k - \vec{u}_0, \vec{u}_j)$$
$$\vdots$$
$$(\vec{u}_k - \vec{u}_{k-1}, \vec{u}_j).$$

Once again, the extrapolated vector will remain the same since the sign change, if appropriate, of the numerator and the denominator will cancel out. In order, for $i = 2$ to $k-1$, set the i^{th} row equal to the i^{th} row minus the $(i+1)^{th}$ row. The resulting j^{th} column of matrix A will be

$$\vec{x}_j$$
$$(\vec{u}_1 - \vec{u}_0, \vec{u}_j)$$
$$(\vec{u}_2 - \vec{u}_1, \vec{u}_j)$$
$$\vdots$$
$$(\vec{u}_k - \vec{u}_{k-1}, \vec{u}_j).$$

This column is identical to the j^{th} column of R. Therefore, the extrapolated vector $\vec{y}_{k+1}$ is the same vector for the RRE method and this special case of Anderson's higher degree method, $B_n = 0$.

Now consider Equation (81) for Anderson's method with $B_n = 1$ applied to the linear equation (4). Let $a_j = N_j / D[1, \ldots, 1]$, $j = 0, \ldots, k$. Then

$$\begin{aligned} \vec{y}_{k+1} &= \sum_{j=0}^{k} a_j \, \vec{x}_{k+1-j} = \sum_{j=0}^{k} a_j \, A\vec{x}_{k-j} = A\left(\sum_{j=0}^{k} a_j \, \vec{x}_{k-j}\right) \\ &= A\frac{D[\vec{x}_k, \ldots, \vec{x}_0]}{D[1, \ldots, 1]}, \end{aligned}$$

which is one iteration of equation (84). Therefore, it is seen that, for the linear case, the extrapolated vector for Anderson's method with the B's equal to unity is identical to one iteration of the extrapolated vector for Anderson's method with the B's equal to zero; hence, it is equivalent also to one iteration of the extrapolated vector obtained by the RRE method. An example will follow to illustrate this fact.

Numerical Examples

Smith, Ford, and Sidi (1987) suggested using the Gauss-Seidel iteration scheme applied to the linear equation (Wynn, 1962, Eq. (14)) $F(\vec{x}) = A\vec{x} - \vec{b} = 0$, where

$$A = \begin{bmatrix} 2 & 1 & 3 & 4 \\ 1 & -3 & 1 & 5 \\ 3 & 1 & 6 & -2 \\ 4 & 5 & -2 & -1 \end{bmatrix} \quad \text{and} \quad \vec{b} = \begin{bmatrix} 10 \\ 4 \\ 8 \\ 6 \end{bmatrix}. \tag{85}$$

To define the Gauss-Seidel scheme, let a_{ij}, $i, j = 1, \ldots, m$, where m is the order of A, be the (i, j) component of A. Define the matrices L and UP by

$$L = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ a_{21} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{m,m-1} & 0 \end{bmatrix}, \tag{86}$$

$$UP = \begin{bmatrix} 0 & a_{12} & \cdots & \cdots & a_{1m} \\ 0 & 0 & \cdots & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & a_{m-1,m} \\ 0 & 0 & \cdots & \cdots & 0 \end{bmatrix}. \tag{87}$$

Also define D to be the diagonal matrix $[a_{11}, a_{22}, \ldots, a_{mm}]$. Then the Gauss-Seidel iteration is defined by

$$(D + L)\vec{x}_{n+1} = -(UP)\vec{x}_n + \vec{b}. \tag{88}$$

When this scheme is applied to Problem (85), the result is a divergent sequence; however, all three acceleration techniques obtain the solution, $(1, 1, 1, 1)$. If we let $k = 2$, then the first extrapolated vectors, to five place accuracy, for the RRE

method, $\vec{r}$, and Anderson's method, $\vec{a}$, with $B_n = 1$, are

$$\vec{r} = (-0.17247, 1.09243, 0.67697, 1.39913)^T \text{ and}$$

$$\vec{a} = (0.64007, 1.43756, 1.24008, 1.26795)^T, \text{ respectively.}$$

Applying an iteration to $\vec{r}$ shows that $\vec{a} = F(\vec{r})$, as was shown and previously discussed. This fact in itself would lead one to believe that for linear problems, Anderson's method has a major advantage over the RRE method. Now consider another example, a nonlinear one.

In Chapter VIII, Problem (58) was used to compare the RRE method with previously derived techniques. As noted in the last paragraph of Chapter VIII, the RRE method and the MPE method seemed to be equivalent procedures since they produced identical results for $k = 2$. One may wonder how this can be true since the theory proven in this chapter shows that they are not equivalent. Let us examine this problem more closely. The second and third rows of matrix (66) for the MPE, RRE, and Anderson's methods are given in Table 14 (page 89). Hence, matrix (66) is different for each method. However, when the extrapolated vector for each method is computed, we find that all four methods (MPE, RRE, and Anderson's method with $B_n = 0$ and $B_n = 1$) give identical results:

$$(0.774124, 0.419430)^T = 1.080567\vec{x}_0 + 0.286985\vec{x}_1 - 0.367552\vec{x}_2$$

for the RRE, MPE, and Anderson's ($B_n = 0$) methods and

$$(0.774124, 0.419430)^T = 1.080567\vec{x}_1 + 0.286985\vec{x}_2 - 0.367552\vec{x}_3$$

for Anderson's method ($B_n = 1$), where

$$\vec{x}_0 = (0.8, 0.4)^T, \ \vec{x}_1 = (0.8, 0.48)^T,$$

$$\vec{x}_2 = (0.8704, 0.4096)^T, \text{ and } \vec{x}_3 = (0.925368, 0.589824)^T.$$

TABLE 14

SECOND AND THIRD ROWS OF MATRIX (66) FOR THE MPE, RRE, AND ANDERSON'S METHOD FOR PROBLEM (58)

MPE			RRE		
0.00640	−0.00563	0.01442	0.00802	−0.00319	0.02108
−0.00563	0.00991	−0.00882	0.02005	−0.01873	0.04432

ANDERSON		
−0.01203	0.01554	−0.02324
0.02005	−0.01873	0.04432

Equivalent results were expected for Anderson's method with $B_n = 0$ and the RRE method; but, why did all four methods obtain the same results? Consider, for the moment, only the MPE and RRE methods, and the difference between their $a_{i,j}$ elements of matrix (66). Substituting $\vec{u}_{n+1} - \vec{u}_n$ for $\vec{v}_n$ and carrying out the determinant calculations, the resulting coefficients for $\vec{x}_0, \vec{x}_1$, and $\vec{x}_2$, respectively, in terms of inner products with notation $M0112 = (\vec{u}_0, \vec{u}_1)(\vec{u}_1, \vec{u}_2)$, are

$$M1122 - M0122 + M0112 - M1212 + M0212 - M0211,$$
$$M0212 - M0202 + M0102 - M0122 + M0022 - M0012, \quad \text{and}$$
$$M0112 - M0012 + M0011 - M0211 + M0102 - M0101$$

for the RRE method and

$$M0112 - M0211,\ M0102 - M0012, \quad \text{and} \quad M0011 - M0101$$

for the MPE method. Comparing the different coefficients, there is a difference of

$$M1122 - M0122 - M1212 + M0212,$$
$$M0212 - M0202 - M0122 + M0022, \quad \text{and} \tag{89}$$
$$M0112 - M0012 - M0211 + M0102$$

for $\vec{x}_0, \vec{x}_1$, and $\vec{x}_2$, respectively. In addition, the difference of $D[1,\ldots,1]$ for the two extrapolations is

$$\begin{aligned} &M1122 - 2M0122 + M0112 - M1212 + M0212 - M0211 + \\ &M0212 - M0202 + M0102 + M0022 - M0012. \end{aligned} \tag{90}$$

Define $w0, w1, w2$ to be the quotient of the expressions in (89) divided by (90) for $\vec{x}_0, \vec{x}_1$, and $\vec{x}_2$, respectively. The results are

$$w0 = 1.080567,\ w1 = 0.286985, \quad \text{and} \quad w2 = -0.367552.$$

These values are the same values as their corresponding coefficients.

Consider the coefficient of one term of $D[\vec{x}_0, \vec{x}_1, \vec{x}_2]$ for the MPE method. Let NM represent this value and let DM represent the value $D[1,1,1]$. In addition, define DN as the difference (89) for the chosen coefficient and DD as the difference (90). As was shown in the previous paragraph, the quotients NM/DM and DN/DD are equal. Setting this value to r, we have $DN = DD(r)$ and $NM = DM(r)$. Also,

$$NM + DN = DM(r) + DD(r) = (DM + DD)r.$$

Therefore, $(NM + DN)/(DM + DD) = r$. Since this is true for each coefficient, we see that the linear combination of $\vec{x}_i$'s is the same for both methods.

Similar results can be shown for Anderson's method with $B_n = 1$. Though the results showed equality for this case, it is easy to see that the two methods are not equivalent. To show this fact by example, consider the problem

$$\begin{aligned} x &= x^2 + y^2 - z^2, \\ y &= x_2 - y^2 + z^2, \quad \text{and} \\ z &= -x_2 + y_2 + z^2. \end{aligned}$$

One extrapolation of this problem gives

$(0.175810, -0.786335, -0.978136)$ for the MPE method,

$(0.071951, 0.020600, -0.039942)$ for Anderson's, $B_n = 1$,

$(0.236638, 0.039550, -0.063152)$ for the RRE method, and

$(0.236638, 0.039550, -0.063152)$ for Anderson's, $B_n = 0$.

Hence, they are all different with the exception of the RRE and Anderson's with $B_n = 0$ methods, as expected. The above results indicate that the MPE, the RRE, and Anderson's ($B_n = 0$) methods are very similar for one extrapolation. However, the first extrapolated vector for Anderson's method ($B_n = 1$), for the linear case, is one iteration better than the RRE method. If the fact that Anderson's method is fully dynamic is also considered, then one might assume that Anderson's method ($B_n > 0$) will accelerate a sequence to its correct limit faster than the other techniques. In the next chapter, we will check the validity of this assumption by testing Anderson's method and the other methods on several types of test problems.

CHAPTER XI

NUMERICAL TEST PROBLEMS

In this chapter we compare the acceleration methods numerically. Test problems will include linear and nonlinear problems with dimensions varying from 4 up to 8000. The problems were tested using FORTRAN coded programs. All problems except Examples 9 and 10 were computed on a Kaypro 286 PC with double precision, giving a relative precision of 1 x 10^{-19}. Examples 9 and 10 were computed on a IBM 3081K (VS FORTRAN) with double precision, giving a relative precision of 2 x 10^{-16}. Dr. John P. Chandler, Oklahoma State University, developed the main software for the semi-dynamic Vector Aitken (VA) method, Wynn's original Vector ε (Vε) method, the Modified Vector ε (MVε) method, the MPE method, and Anderson's (AND) method. I wrote the program for the RRE method and made some modifications to Chandler's Vε, MPE, and AND programs.

The results will be presented in figures which show graphs of the logarithm (base 10) of the infinity norm of the difference vector, $\vec{u}_{n-1} = \vec{x}_n - \vec{x}_{n-1}$, as a function of the number of iterations. Exceptions are Examples 9 and 10. Example 9 is the graph of the infinity norm of the error vector, $\vec{e}_n = \vec{s} - \vec{x}_n$, where $\vec{s}$ is the solution. Example 10 plots the Euclidean norm of the difference vector. The notation will be $\log_{10} \|\vec{u}_{n-1}\|$. The stopping criterion on all programs is $\|\vec{u}_n\| < 1$ x 10^{-15}. Therefore, this value, denoted by $C15$, will be considered as defining numerical convergence for all problems.

Results were obtained for the basic iteration; all four variations of the VA method; the Vε method; orders one, two, and three for the MVε method; and for k

values of two through five (or the dimension of the problem if less than five) for the RRE, MPE, and AND methods. In addition, the AND method was applied with the relaxation option. However, this method never obtained results that matched those of Anderson's higher degree methods; hence, results of the relaxation method will be shown for Example 1 only. In addition, the convergence of the VA, Vε, and MVε methods were usually much slower than that of the AND, MPE, or RRE methods. Therefore, the results of the VA, Vε, and MVε methods will not be shown for examples where their results did not compare favorably with the other methods. Because of the structure of the Vε method, the results will show the difference vector of ε_{n-1}^{0}, if n is even, or ε_{n-1}^{1}, if n is odd. The RRE, MPE, and AND methods consistently obtained the best results, faster convergence. For each of these three methods, the results for the value of k which obtained fastest convergence for that particular method will be shown. In addition, some examples will show results for varying values of k for a particular method.

Before we continue, it should be understood that throughout this study we have assumed that the fewest number of iterations implies the best results. This is not always the case. Because of the sophistication of some of the methods and the simplicity of the basic iteration scheme of some problems, more extrapolations and fewer iterations may be more time consuming than the convergence of the basic iteration. However, this type of problem is not in the majority, and fewer iterations usually means less computer time and better results.

EXAMPLE 1: The first example is a simple highly degenerate, linear problem (Anderson, 1965, Eq. (5.1)). Define $F(\vec{x})$ of Equation (1) by

$$F(\vec{x}) = A\vec{x} - d\vec{b} = \vec{0},$$

where d is a free parameter and

$$a_{ij} = \begin{cases} d, & \text{if } i = j \\ 1, & \text{if } i \neq j. \end{cases}$$

Choose $\vec{b}$ such that the component elements of the solution $\vec{s} = (s_1, \ldots, s_m)^T$ is $s_i = 2/i$, $i = 1, \ldots, m$. The matrix A can be written as $A = L + UP + D$, where L and UP are defined by (86) and (87), respectively; and D is the diagonal matrix $[a_{11}, \ldots, a_{mm}]$. Therefore, the problem can be rewritten in the Jacobian iteration form

$$\vec{x} = G(\vec{x}) = -D^{-1}(L + UP)\vec{x} + D^{-1}d\vec{b}. \tag{91}$$

For $m = 20, d = 25$, and $\vec{x}_0 = (1, \ldots, 1)^T$; the basic iteration sequence converges in fifteen iterations. All six acceleration methods also obtain convergence (results are not shown). The RRE, MPE, and AND methods converge after only four iterations ($k = 2$). The Vε and MVε methods converge in 5 iterations, while the VA method requires 18 iterations.

When d is set to 15, the problem is a little different because the basic iteration scheme produces a divergent sequence. However, all acceleration methods obtain the correct solution. Figure 20 (page 95) shows best results for all acceleration methods except the VA method. Figure 21 (page 95) shows results of the basic iteration, the VA method, the relaxation option of the AND method (relax), and the MPE method for $k = 3$ and 4. Best results are for order 2 for the MVε method and $k = 2$ for the three methods using k values. As mentioned earlier, the VA method is inappropriate for divergent problems unless the vectors $\vec{x}_n$ and $\vec{x}_{n+2}$ are interchanged in the formula. The graph of the VA method as shown in Figure 21 is without this change and is shown for comparison purposes only. The convergence rate of the other five methods are similar.

One point of interest noted in Figure 20 is that the AND method converges

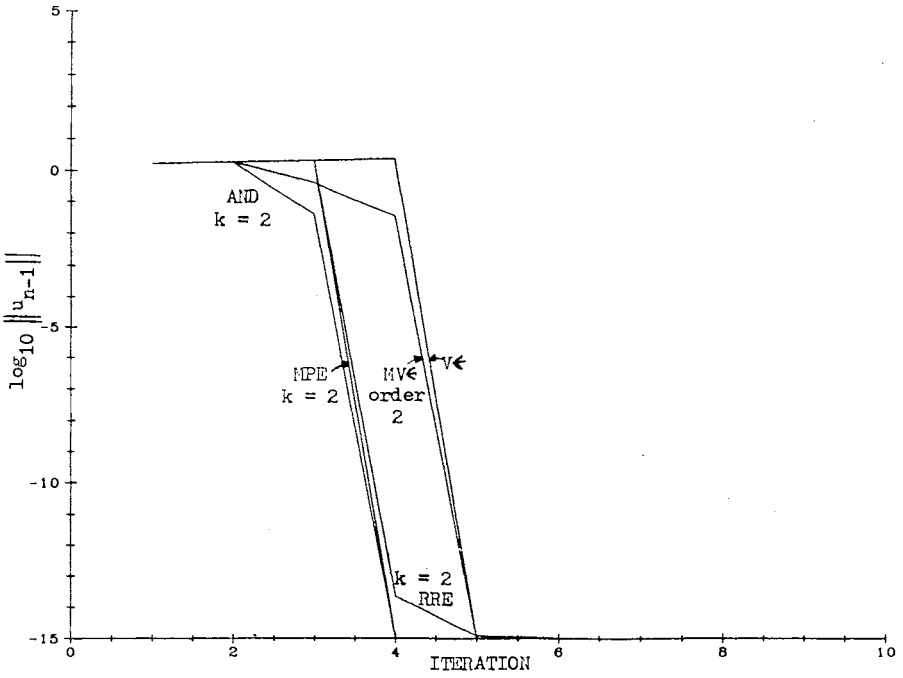

Figure 20. Results for Example 1: AND, MPE, RRE, MVε, and Vε

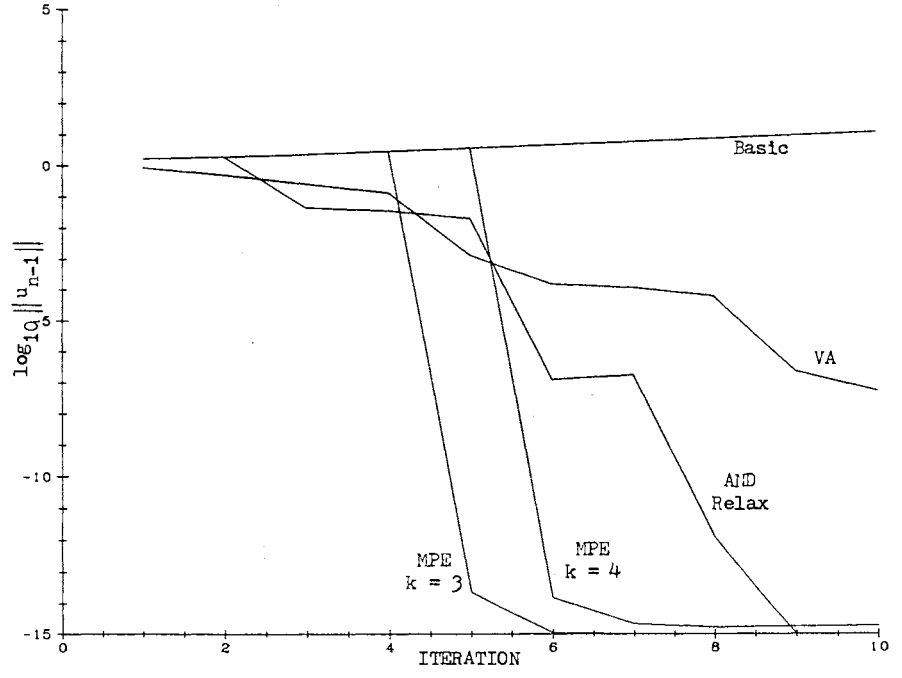

Figure 21. Results for Example 1: VA, MPE (k = 3 and 4), and Relax

for $k \geq 2$ in four iterations due to its ability to extrapolate after only two iterations even though the final value of k may be larger than 2. For the RRE and MPE methods, the number of iterations increases for $k > 2$ as shown in Figure 21 for the MPE method with $k = 3$ and 4.

EXAMPLE 2: Consider the divergent linear problem (91) (Smith, Ford, and Sidi, 1987). The iteration scheme for this problem was discussed in Chapter X, and the results are shown in Figure 22 for the RRE, MPE, AND, and MVε methods. Even though the dimension of this problem is four, the degree of the minimal polynomial is three since the matrix A has one zero eigenvalue. All three methods are able to detect the zero eigenvalue as demonstrated by the fact that the best results are obtained for $k = 3$ when the first iteration is discarded. In addition, the AND method achieves machine accuracy (the norm of the difference vector equals

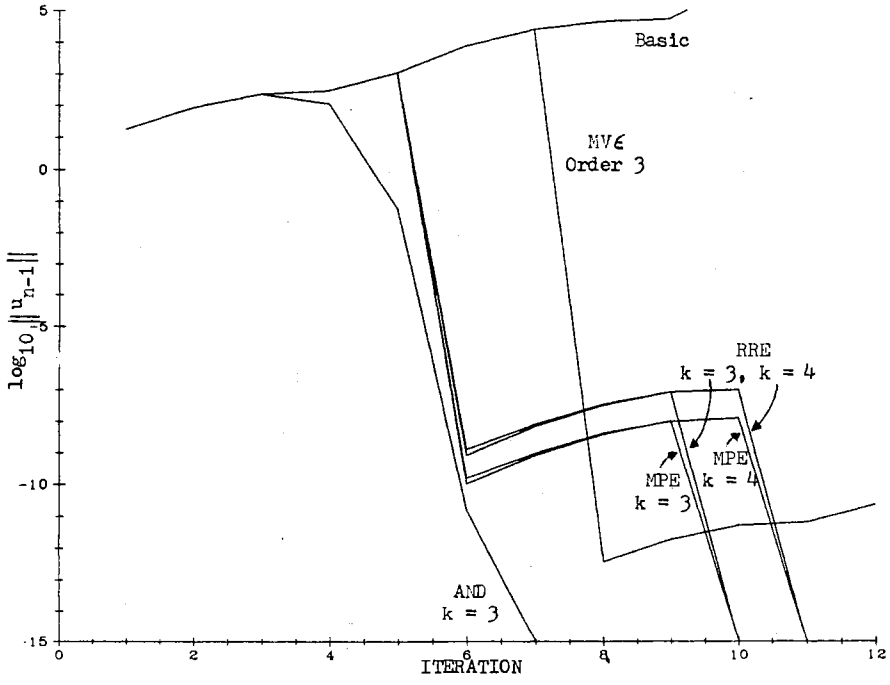

Figure 22. Results for Example 2: AND, MPE, RRE, and MVε

zero) for $k = 4$ and the first iterate discarded. However, the number of iterations remains seven and the accuracy of the sixth iterate is not as high as the sixth iterate for $k = 3$. If the first iteration is not discarded, the AND method still converges in seven iterations while the RRE and MPE methods require one additional iteration each to converge. However, for all three methods, the best results are obtained with $k = 4$. The results of the MVε method are for order 3 with the first iterate discarded.

There is another point of interest concerning this example. Since the problem is linear and the degree of the minimal polynomial is three ($k = 3$), the exact solution should be found by all three methods in $k+1 = 4$ iterations and one extrapolation. This was not done because computer computation is not exact arithmetic; hence, rounding errors result and exact convergence in four iterations is not obtained.

Smith, Ford, and Sidi (1987) obtained a "wrong" solution for this problem using the RRE method. The reason their RRE method failed to converge to the correct solution is discussed in the Appendix.

EXAMPLE 3: For a final look at a linear case, we find the solution of the system

$$\begin{bmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{bmatrix} \vec{x} = \begin{bmatrix} 23 \\ 32 \\ 33 \\ 31 \end{bmatrix},$$

by use of the Jacobian iteration, (91), with an initial vector $\vec{x}_0 = (0,0,0,0)^T$ (Smith, Ford, Sidi, 1987, Ex. 1). The solution is (1,1,1,1). One of the eigenvalues for this problem is near 0.9985, causing the matrix $I - A$ to be nearly singular. Figure 23 (page 98) gives the results for the RRE and MPE methods ($k = 4$), the AND method ($k = 4$ and 3), and the MVε (order 3). This is a linear problem whose

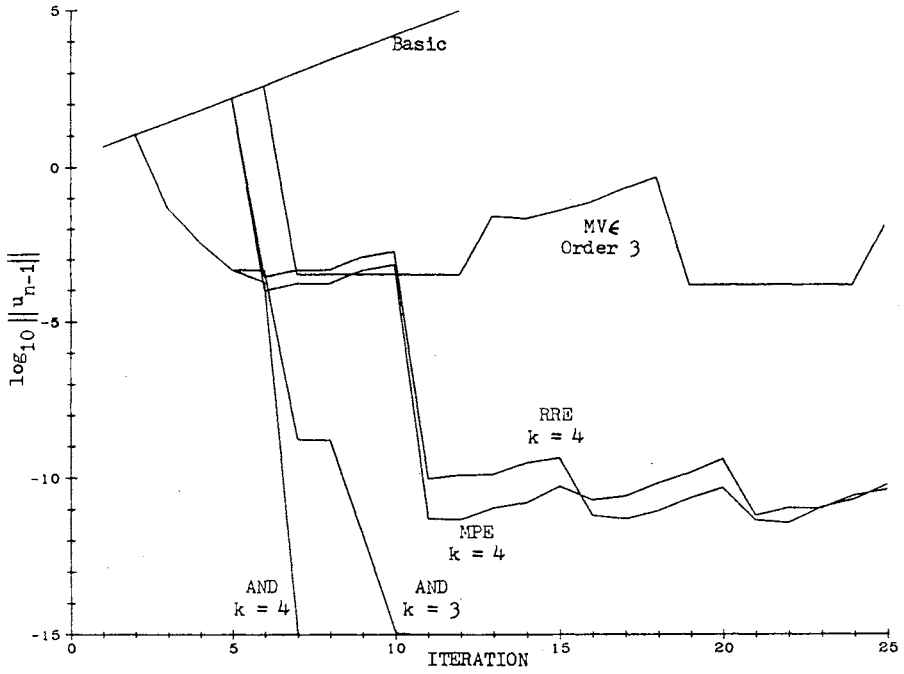

Figure 23. Results for Example 3: AND, MPE, RRE, and MVε

minimal polynomial has degree four. Therefore, ignoring roundoff, convergence should be obtained in $k+1=5$ iterations and one extrapolation. However, once again, this is not achieved due to rounding errors.

One observation is that whereas the AND methods converge in 7 and 11 iterations, the RRE and MPE methods do not converge to high accuracy in even 25 iterations. In fact, the convergence of these two methods does not improve in 250 iterations for any value of k. For both methods, the log of the difference vectors fluctuates between -9 and -13 with no set pattern. As a result, the log of the error vectors never gets smaller than -11. This inability to improve convergence beyond a certain value is referred to as "limited accuracy." The VA method and the ε methods have the same difficulty on this problem only with much lower accuracy.

There were several variations made to the basic iteration scheme to attempt

to achieve convergence. These included using combinations of different relaxation factors, discarding the first few iterates, and discarding a set number of iterates at the beginning of each extrapolation throughout the run of the program or only after the limited accuracy had been achieved. However, no improvement was made as all attempts eventually led to some value of limited accuracy.

Before investigating the cause of this limited accuracy problem, a few definitions are in order. Given the vector norm $\|\vec{x}\|$, define the norm, $\|A\|$, and the condition number, cond(A), of the matrix A by

$$\|A\| = \sup_{\vec{x}\neq\vec{0}} \frac{\|A\vec{x}\|}{\|\vec{x}\|} \quad \text{and} \quad \text{cond}(A) = \|A\|\|A^{-1}\|,$$

respectively (Conte and de Boor, 1980). For the linear system $A\vec{x} = \vec{b}$, define the relative error as $\frac{\|\vec{x}-\tilde{\vec{x}}\|}{\|\vec{x}\|}$ where $\tilde{\vec{x}}$ is the computed vector of $\vec{x}$, and the residual error by $\frac{\|\vec{b}-A\tilde{\vec{x}}\|}{\|\vec{b}\|}$. It is shown by Golub and Van Loan (1983) how the condition number and the relative errors in $A, \vec{x}$, and $\vec{b}$ relate. If cond$(A) \approx 1$, then the relative error and the residual error will be of the same order of magnitude; hence, $\tilde{\vec{x}}$ is a good approximation of $\vec{x}$. If the condition number is large, then a small change in the data MAY cause a large change in the solution. In short, the condition number "quantifies the sensitivity of the $A\vec{x} = \vec{b}$ problem" (Golub and Van Loan, 1983). Ortega and Poole (1981) give the example that a condition number of 10^6 could result in a loss of 6 decimal digits of accuracy. A matrix with a large condition number is called an "ill-conditioned" matrix.

The RRE, MPE, and AND methods all involve solving a linear system $U\vec{x} = \vec{b}$ to determine the extrapolated vector. Therefore, consider the condition number of U for each extrapolation. For the k values shown in the Figure 22, the first condition number for the RRE method is 10^{13} and for the MPE method is 10^{10}. The remaining condition numbers vary between 10^3 and 10^9. For the AND method, the first condition number is 10^4 and the condition number for the matrix when

convergence is obtained is 2.67. This is a major difference and could be a factor in allowing Anderson's method to overcome the limited accuracy problem.

Since the example is a small linear problem, it can be solved by Gaussian elimination. The condition number of the matrix A is 50 and the computed solution has a relative error of 0.1896 x 10^{-13}. The relative error of Anderson's computed solution is 0.2251 x 10^{-12}. The relative errors for the RRE and MPE methods never obtain an accuracy higher than 0.4467 x 10^{-10} and 0.1973 x 10^{-8}, respectively. The AND method, even with the low condition numbers, still cannot achieve quite the same degree of accuracy that the Gaussian elimination method achieves. However, when comparing acceleration methods, the AND method definitely shows superiority in overcoming the problem of limited accuracy. The RRE method for Smith, Ford, and Sidi (1987) failed to converge for this problem. See the Appendix for more details.

EXAMPLE 4: The first nonlinear example is a quadratic problem with solutions, (1,1,1,1) and (3,3,3,3) (Gekeler, 1972, Ex. V). Define $G(\vec{x})$ of equation (2) by

$$G(\vec{x}) = A\vec{x} + \vec{b} + Q(\vec{x}),$$

where

$$A = \begin{bmatrix} 3.9 & -3.7 & 2.4 & -0.6 \\ 2.4 & -2.0 & 2.2 & -0.6 \\ 2.4 & -3.6 & 4.1 & -0.9 \\ 2.8 & -5.2 & 4.8 & -0.4 \end{bmatrix},$$

$$\vec{b} = -0.75(1,1,1,1)^T, \quad Q(\vec{x}) = -0.25(x_1^2, x_2^2, x_3^2, x_4^2)^T, \quad \text{and}$$

$$\vec{x}_0 = 1.5(1,1,1,1)^T.$$

The basic iteration converges to the solution $(3,3,3,3)$ in 52 iterations with a con-

vergence criterion of 1 x 10^{-14}, C14; however, in 400 iterations the norm of the difference vector never gets smaller than the normal convergence criterion, C15.

This problem has some unusual properties. Figure 24 (page 102) shows the results of the basic iteration, the Vε method, the MVε method (order 4), and the MPE and RRE methods with $k = 4$. Each of these methods converges to $(3, 3, 3, 3)$. Figure 25 (page 102) shows the convergence of the VA with Q equal to formula (33), the AND method ($k = 3$), the MVε (order 3), and the MPE method ($k = 1$). These four methods converge to the solution $(1, 1, 1, 1)$. In addition, Figure 25 shows the convergence of Newton's iterative method, the "most famous iterative method for obtaining roots of equations (as well as for solving systems of nonlinear equations ...)." (Ortega and Poole, 1981, p. 128). Convergence is not obtained for the RRE method ($k < 4$) and the MPE method ($k = 2$ and 3). For these methods, the system either overflows or the limited accuracy is 10^{-1}.

A possible reason for this difference for the methods involving k values is the matrix determined for computing the first extrapolated vector. For both methods that converge to $(3, 3, 3, 3)$, the matrix is singular. As a result, the matrix of the linear system has a rank smaller than k; hence, a linear system of smaller degree is solved resulting in a first computed extrapolated vector near $(3, 3, 3, 3)$. Remaining iterates and extrapolations converge to this vector. The rank of the matrix for the other converging methods is k and the resulting extrapolated vector is near $(1, 1, 1, 1)$.

As stated in Chapter I, small nonlinear problems will in practice be solved by methods other than those discussed in this thesis. To illustrate this point, this problem is solved by Newton's method. Let $F(\vec{x}) = G(\vec{x}) - \vec{x}$, then Newton's method is the iterative formula

$$\vec{x}_{n+1} = \vec{x}_n + \vec{y}_n, \quad \text{where} \quad F'(\vec{x}_n)\vec{y}_n = -F(\vec{x}_n)$$

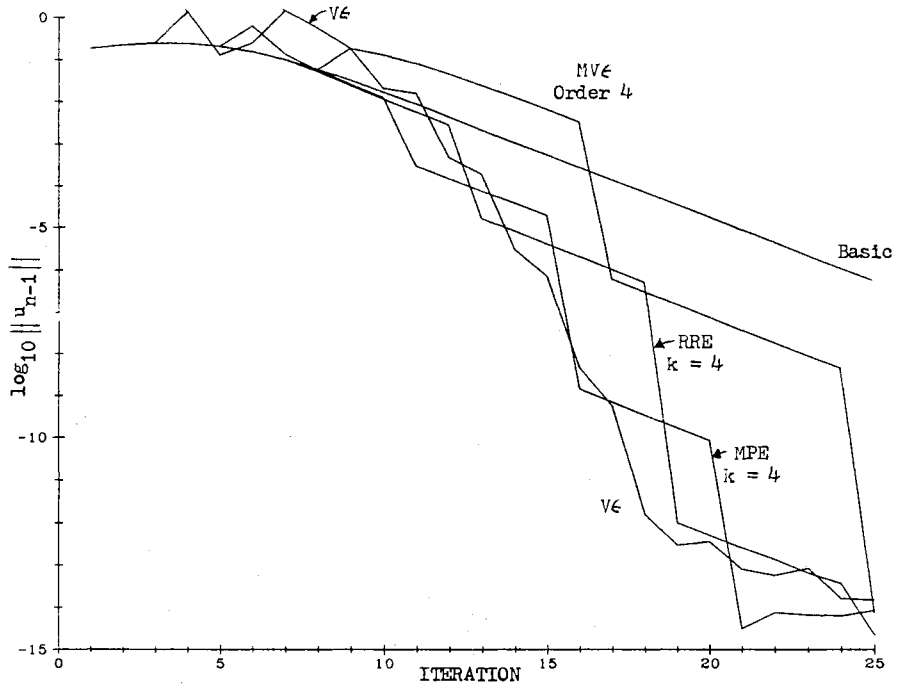

Figure 24. Results for Example 4: Methods Converging to (3,3,3,3)

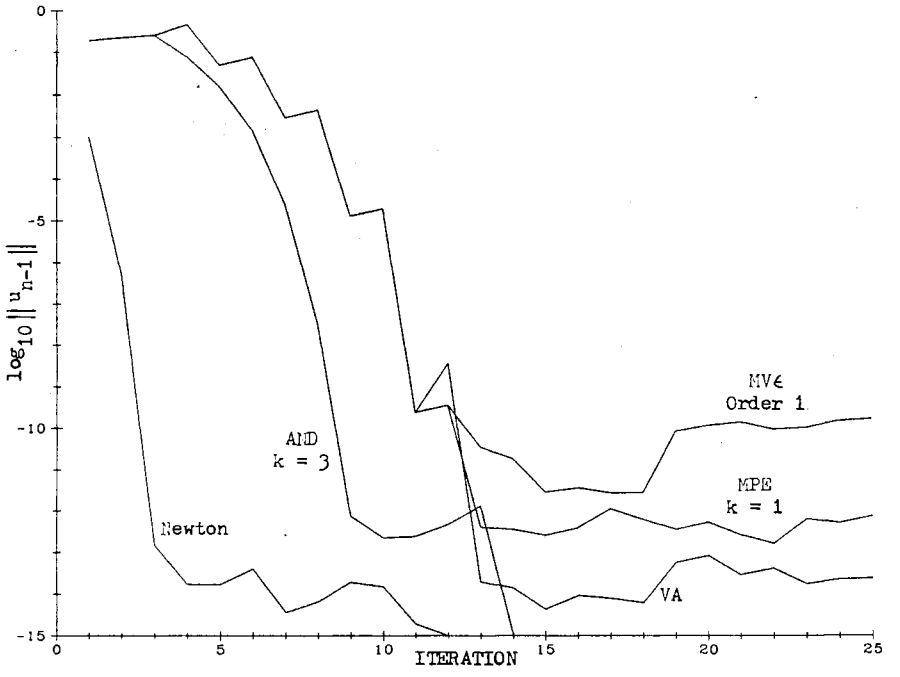

Figure 25. Results for Example 4: Methods Converging to (1,1,1,1)

for $F'(\vec{x}_n)$ the Jacobian matrix of $F(\vec{x})$ evaluated at $\vec{x}_n$. As can be seen in Figure 25, all the methods that converge to $(1,1,1,1)$ have some difficulty with limited accuracy. However, Newton's method and the AND method overcome this problem and achieve convergence fairly quickly.

Example 5: The next example comes from Wynn (1964, Eq. (1)) and is referred to as the Lichtenstein-Gershgorin integral equation. The iteration scheme for $n = 0, 1, \ldots$ is

$$\begin{aligned}\theta_{n+1}(x) &= \frac{k}{\pi}\int_0^\pi \frac{k_1\theta_n(t)}{1-k_2\cos(t+x)} - \frac{k_1\theta_n(\pi-t)}{1+k_2\cos(t+x)}\,dt \\ &\quad +2\arctan\frac{k^{-1}\sin(x)}{[1-\cos(x)]\,[k_3\cos(x)-k^{-2}]},\end{aligned}$$

where

$$k_1 = (k^2+1)^{-1},\ k_2 = k_1(k^2-1),\ k_3 = 1-k^{-2},\quad \text{and}\quad \theta_0 = \vec{0}.$$

The integrals are approximated by the trapezoid rule with end corrections:

$$\int_a^{a+mh} f(t)\,dt = h\left\{\frac{1}{2}f_0 + f_1 + \cdots + f_{m-1} + \frac{1}{2}f_m + C\right\} \tag{92}$$

for

$$C = \frac{1}{12}(\Delta f_0 - \nabla f_n) - \frac{1}{24}(\Delta^2 f_0 + \nabla^2 f_n) + \frac{19}{720}(\Delta^3 f_0 - \nabla^3 f_n) - \frac{3}{160}(\Delta^4 f_0 + \nabla^4 f_n),$$

where

$$\Delta f_0 = f_1 - f_0,\ \Delta^n f_0 = \Delta^{n+1} f_0 - \Delta^n f_0,$$
$$\nabla f_n = f_{n-1} - f_n,\quad \text{and}\quad \nabla^n f_n = \nabla^{n-1} f_n - \nabla^n f_n.$$

Choosing $m = 73$ and $x_0 = (0,\ldots,0)^T$, the basic iteration converges in 136 iterations. Figure 26 (page 104) shows the graphs for the convergence of the RRE, MPE, and AND methods with $k = 5$, the MVε method (order 3), and the VA method with Q equal to formula (33). As can be seen from the figure, this example

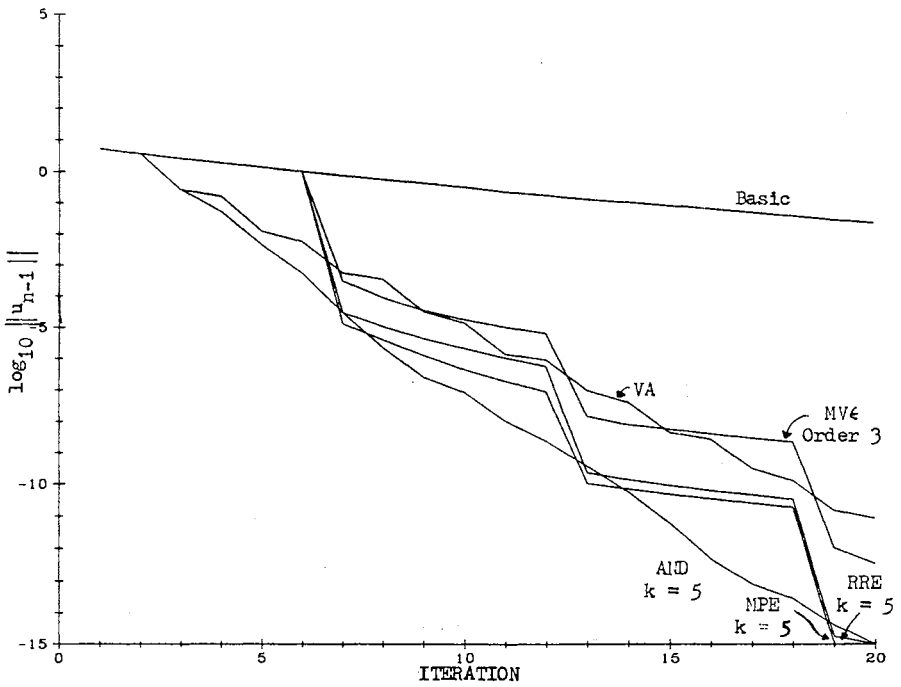

Figure 26. Results for Example 5: AND, MPE, RRE, VA, and MVε

has very similar results. This problem does illustrate the importance of an acceleration method. Due to the nature of an integral equation problem, the computer time required to compute one iterate is much more than any of the previous examples. Therefore, to decrease the number of iterations to less than one-fifth that required by the basic iteration is a significant reduction in computer time and cost. There are no other interesting points concerning the results of this example other than the fact that this is the first example in which a method other than the AND method even came close to having the best results. The AND method is still quite competitive, however.

Example 6: The next examples are the integral equation problems (Anderson, 1965, Eqs. (5.10) & (5.11)):

$$f^2(x) = \frac{2\pi\sqrt{2}}{16}\int_{-1}^{1} f(t)\left(\cos\frac{\pi|x-t|}{4}\right)^2 dt - \frac{1}{4} \quad \text{and} \tag{93}$$

$$f(x) = \frac{3\pi\sqrt{2}}{16}\int_{-1}^{1} f^2(t)\cos\frac{\pi|x-t|}{4}\,dt - \frac{1}{4}\cos\frac{\pi x}{4}, \tag{94}$$

with both solutions $f(x) = \cos(\pi x/4)$.

Letting $m = 101, f_0 = (1,\ldots,1)^T$, and integrating both problems with the iteration scheme (92); Integral (93) converges in 75 iterations while Integral (94) diverges. Results for Integral (93) are shown in Figure 27 (page 106). The AND method clearly obtains the fastest convergence. Once again, the RRE and MPE methods ($k = 2$) have similar convergence. This example will also show how using a convergent generated sequence in reverse order can affect the convergence rate. If the MPE method ($k = 2$) is applied to the sequence $S = \{\vec{x}_0,\ldots,\vec{x}_{k+1}\}$, the norm of the thirteenth difference vector is 10^{-11} as compared to 10^{-15} when the method is applied to the sequence $T = \{\vec{x}_{k+1},\ldots,\vec{x}_0\}$. In addition, it requires four more iterations to converge to 10^{-15}. Though there are exceptions to the rule, applying the acceleration method to the sequence T instead of the sequence S usually reduces the number of iterations for convergence if S is a convergent sequence. Using the most accurate estimate of the solution as the first term of the sequence will usually cause faster convergence.

For Integral (94), the basic iteration scheme produces a sequence that diverges quadratically, the logarithm of the norm of the difference vector roughly doubling as n increases by one. Unlike many linear divergences, this divergence cannot be "tamed" by using a relaxation factor in the interval $(0,1)$. The two ε methods also produce divergent sequences. Because the norm of the difference vectors increases rapidly even for the first few iterates, the RRE and MPE methods for $k \geq 2$ do not obtain convergence. For $k = 1$, both methods converge, but the convergence is slow as illustrated in Figure 28 (page 106). The graphs clearly show that the AND method gives the best results. In fact, for all values of k, the results of the AND method are almost identical.

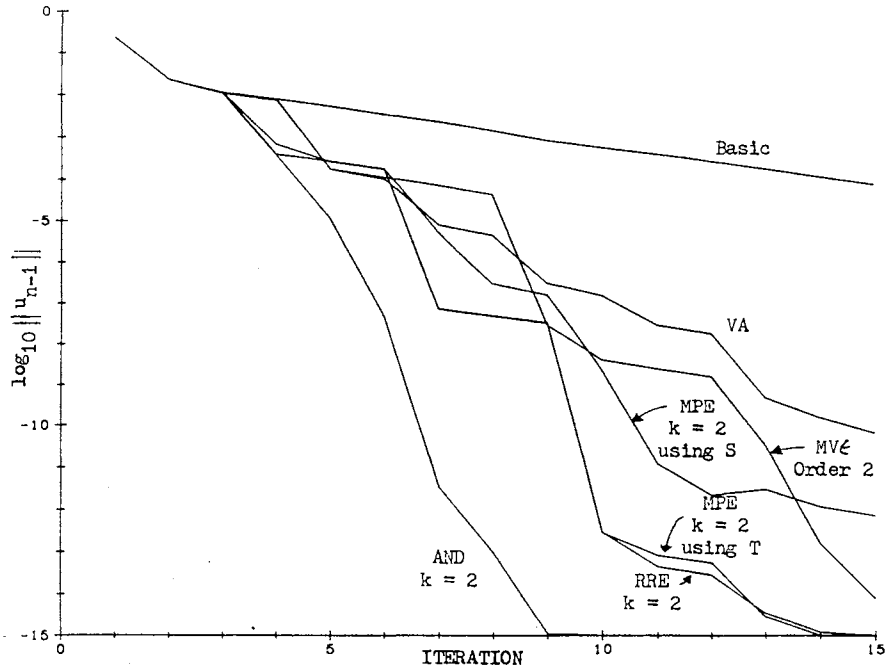

Figure 27. Results for Example 6:
Integral (93)

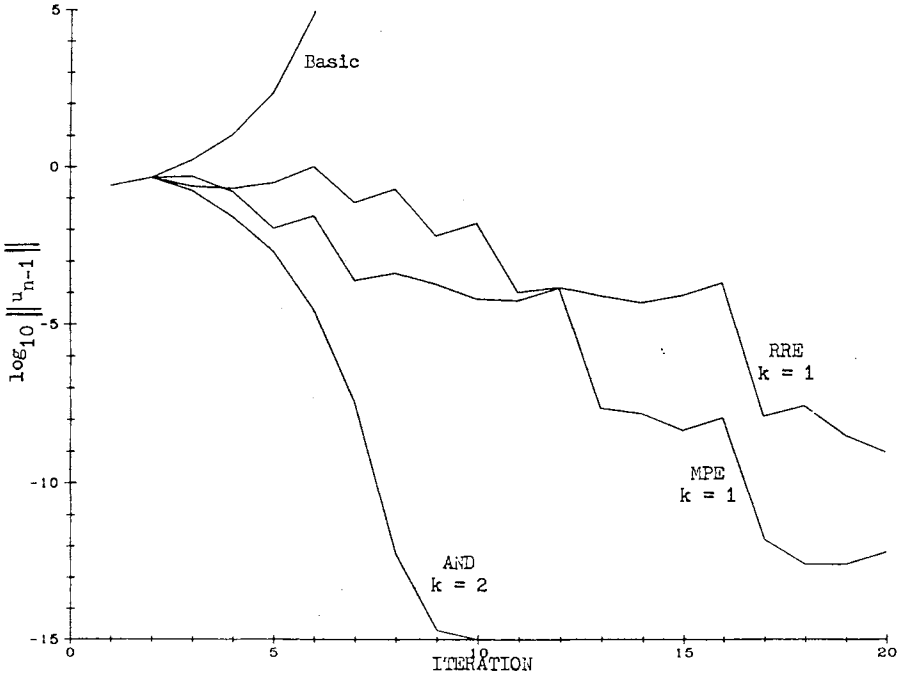

Figure 28. Results for Example 6:
Integral (94)

Example 7: This problem is the "model problem" of Varga (1962) and Young (1971). It is designed to solve the Dirichlet problem on a rectangle. Define R as the interior of a rectangle, S as the boundary of the rectangle, and $R \cup S$ as their union. Let $G(x,y)$ and $g(x,y)$ be continuous functions defined on R and S, respectively. Then the desired solution is a function $u(x,y)$ that is continuous on $R \cup S$, is twice continuously differentiable on R, and satisfies Poisson's equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = G(x,y). \tag{95}$$

In addition, $u(x,y) = g(x,y)$ on S. If $G(x,y) = 0$, then (95) reduces to Laplace's equation.

The function u is found numerically by finding approximations to the function at a finite number of interior points. These points are obtained by superimposing a rectangular mesh of horizontal and vertical lines with uniform spacing. With reference to Figure 29 (page 108), define (x_0, y_0) and (x_p, y_m), p and m integers, as the lower left point and upper right point, respectively, of the rectangle. Also define $d = x_p - x_0, w = y_m - y_0, h = d/p$, and $k = w/m$. Then the spacing of the rectangular mesh is h for the vertical lines and k for the horizontal lines. The spacing in Figure 29 is $h = d/3$ and $k = w/4$. Other points of the mesh are $(x_i, y_j) = (x_0 + hi, y_0 + kj)$. Denote the functional value $u(x_i, y_j), i = 0, \ldots, n$ and $j = 0, \ldots, m$, by $u_{ij} = u(x_i, y_j)$. Hence the solution will be the approximations u_{ij}.

Finite difference approximations to the second derivative with respect to x and y are defined by

$$\begin{aligned} \tfrac{\partial^2 u}{\partial x^2} &= \left[u(x+h,y) - 2u(x,y) + u(x-h,y)\right]/h^2 \quad \text{and} \\ \tfrac{\partial^2 u}{\partial y^2} &= \left[u(x,y+k) - 2u(x,y) + u(x,y-k)\right]/k^2, \end{aligned} \tag{96}$$

respectively (Ortega and Poole, 1981). To simplify the problem, assume the region $R \cup S$ is the unit square, $h = k, (x_0, y_0) = (0,0)$, and $G(x,y) = g(x,y) = 0$.

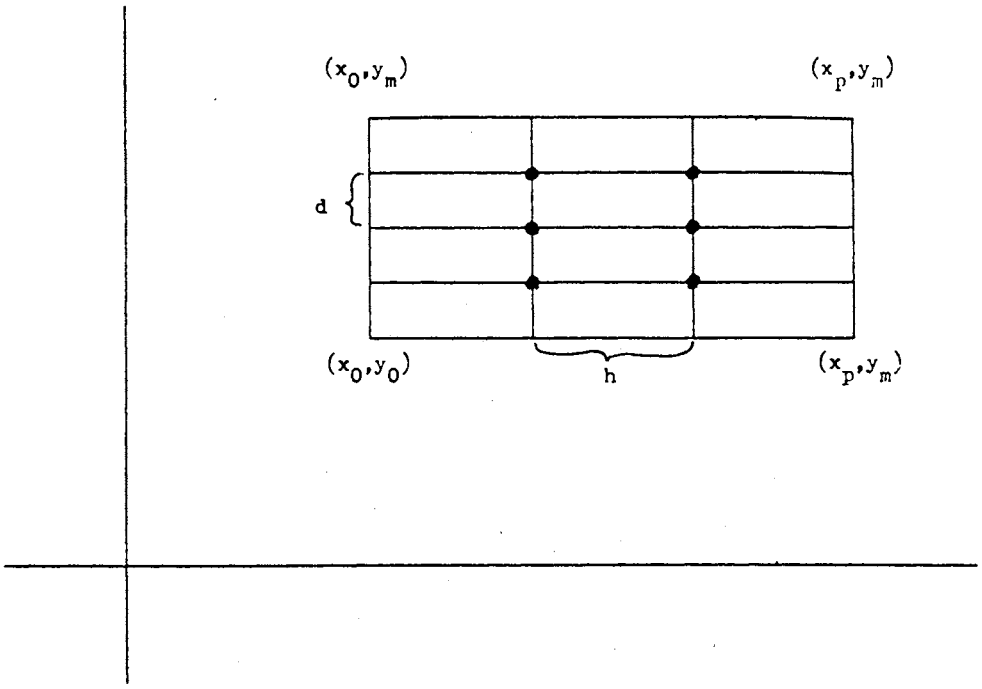

Figure 29. Rectangular Mesh with Spacing of
h = d/3 and k = w/4

Therefore, adding the two equations of (96) and using (95) result in a natural iteration formula for the problem:

$$u_{ij}^{(n+1)} = \left[u_{i+1,j}^{(n)} + u_{i-1,j}^{(n)} + u_{i,j+1}^{(n)} + u_{i,j-1}^{(n)}\right]/4, \tag{97}$$

$i, j = 1, \ldots, p-1$. The right-hand side of (97) is referred to as the five-point Jacobian operator.

The basic iteration generates a slowly convergent sequence as illustrated by the fact that the infinity norm of the 272nd difference vector is only 0.944519×10^{-7} to six place accuracy. Figure 30 (page 109) shows the first 50 iterations of the best results of the RRE, MPE, and AND methods. The other acceleration methods do not work well on this problem. As an example, all three orders of the MVε method converge slower than or similar to the basic iteration, as shown in the figure for

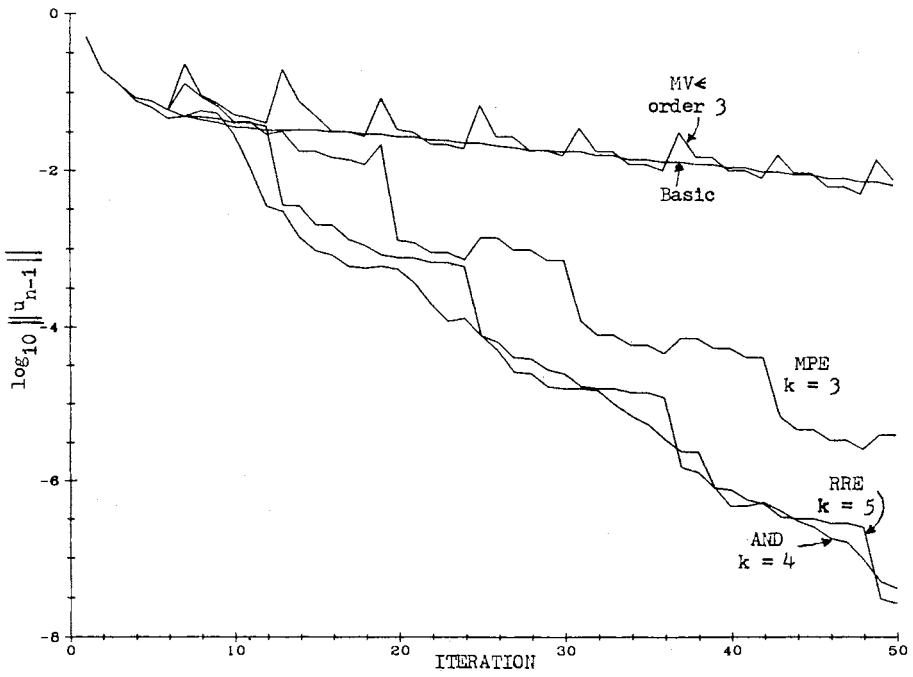

Figure 30. Results for Example 7: AND, MPE, RRE, and MVϵ

order three. This problem is a good example to illustrate how using a relaxation factor other than unity and discarding a few iterations may provide a faster convergence. Testing was done for relaxation factors between 0.5 and 1.5 and discarding one to five iterations. Though not all variations produced faster convergence, better results than those shown in Figure 30 were obtained for all three methods. Table 15 (page 110) gives some of the results obtained and the variations for the AND and RRE methods. It should be noted that this is only the second test problem in which another method matched the results of the AND method.

Example 8: This example is another nonlinear integral equation (Rall, 1969)

$$F(x) = 1 + (\pi_0/2)xF(x)\int_0^1 \frac{F(y)}{x+y}\,dy, \quad 0 \le x \le 1, \tag{98}$$

for $0 \le \pi_0 \le 1$. The background material for the equation is fairly elaborate (Chandrasekhar, 1960). Due to the natural iterative form of (98), the basic iteration

TABLE 15

ITERATIONS REQUIRED FOR CONVERGENCE OF EXAMPLE 7 WITH MODIFICATIONS TO THE ANDERSON AND RRE METHODS WITH A CONVERGENCE CRITERION OF 10^{-7}

Method	Relaxation Parameter	Number of Discarded Iterations	Number of Iterations
AND	0.5	2	37
AND	1.0	1	45
RRE	0.5	1	45
RRE	1.0	1	44

formula is

$$F_{n+1}(x) = 1 + (\pi_0/2)xF_n(x)\int_0^1 \frac{F_n(y)}{x+y}\,dy, \tag{99}$$

$n = 0, 1, \ldots$, with $F_0(x) = 1$, $0 \le x \le 1$. However, analytic difficulties do develop involving the integration portion for finding $F_2(x)$. Rall showed that for $F_0(x) = 1$, for all x an element of the interval of integration, to be a satisfactory initial approximation to the solution, w_0 is restricted by $0 \le \pi_0 \le (\sqrt{2}-1)/(\ln 2) \approx 0.59758\ldots$ Therefore, he constructed a corresponding arithmetic model by introducing a "numerical integration rule" of the form

$$\int_0^1 f(s)\,ds \to \sum_{i=1}^{m} w_i\, f(s_i),$$

where s_i, $0 \le s_i \le 1$, $i = 1, \ldots, m$, are nodes; the parameters w_i, $i = 1, \ldots, m$, are weights; and m is the order of the rule. The integral portion of (98) becomes

$$\int_0^1 \frac{F(y)}{x+y}\,dy \to \sum_{j=1}^{m} \frac{w_j}{x+y_j}F(y_j), \quad 0 \le x \le 1. \tag{100}$$

The solution $F(x)$ is approximated by determining $F(x_i)$, $0 \leq x_i \leq 1$, $i = 1, \ldots, m$. Choosing $x_i = y_i$, $i = 1, \ldots, m$ and using (100), the value at x_i is

$$F(x_i) = 1 + (\pi_0/2)F(x_i) \sum_{j=1}^{m} \frac{x_i\, w_j}{x_i + x_j} F(x_j).$$

Letting $b_{ij} = \frac{x_i\, w_j}{x_i + x_j}$, $i, j = 1, \ldots, m$, Equation (98) becomes

$$f_i = 1 + (\pi_0/2) f_i \sum_{i=1}^{m} b_{ij}\, f_j, \quad i = 1, \ldots, m,$$

where $f_i = F(x_i)$. Defining the m-dimensional vectors $\vec{x}$ and $\vec{1}$ by $\vec{x} = (f_1, \ldots, f_m)^T$ and $\vec{1} = (1, \ldots, 1)^T$, respectively, the iteration formula takes the form

$$\vec{x}_{n+1} = \vec{1} + (\pi_0/2)\vec{x}_n \bigotimes B\vec{x}_n,$$

where $B = [b_{ij}]$ and $\bigotimes$ stands for the component-by-component multiplication of the vectors: $\vec{y} \bigotimes \vec{z} = (y_1 z_1, \ldots, y_m z_m)$ for $\vec{y} = (y_1, \ldots, y_m)^T$ and $\vec{z} = (z_1, \ldots, z_m)^T$.

Table 16 gives the nodes and weights to seven places for the Gaussian integration rule of order nine (Milne, 1949). Using these values and $\pi_0 = 0.1(i)$, for $i = 1, \ldots, 10$, Rall obtained convergence to eight decimal places for all cases;

TABLE 16

NODES AND WEIGHTS FOR THE GAUSSIAN INTEGRATION RULE OF ORDER NINE

i	s_i	w_i	i	s_i	w_i
1	0.0159199	0.0406372	6	0.6621267	0.1561735
2	0.0819844	0.0903241	7	0.8066857	0.1303053
3	0.1933143	0.1303053	8	0.9180156	0.0903241
4	0.3378733	0.1561735	9	0.9840801	0.0406372
5	0.5000000	0.1651197			

TABLE 17

NUMBER OF ITERATIONS REQUIRED TO OBTAIN CONVERGENCE ON RALL'S PROBLEM FOR DIFFERENT VALUES OF π_0

π_0	Number of Iterations	π_0	Number of Iterations
0.1	7	0.6	17
0.2	8	0.7	21
0.3	10	0.8	28
0.4	12	0.9	43
0.5	14	1.0	10587

however, as π_0 increased so did the number of iterations for convergence, Table 17 (Rall, 1969). Table 17 also shows that for $\pi_0 = 1$, convergence is extremely slow, and according to Rall has limited accuracy. Because $w_0 = 1$ is by far the most difficult case, the acceleration methods are applied to this problem for this case only. Figure 31 (page 113) shows results obtained for the MPE, RRE, and AND methods for the first 50 iterations. What the figure does not show is that this problem is another example of limited accuracy for the RRE and MPE methods. Both methods converge to $C10$, but neither one converges to $C11$ in 3000 iterations even with modifications to the relaxation factor and the number of iterations discarded. It should be added that the AND method converges to $C15$ for all values of $k \geq 2$ in less than 132 iterations.

Example 9: The next example has the largest dimensional value of all the test problems. It approximates the steady-state solution of the scalar three-dimensional Burger's equation

$$u_t + u(u_x + u_y + u_z) = \varepsilon \Delta u, \tag{101}$$

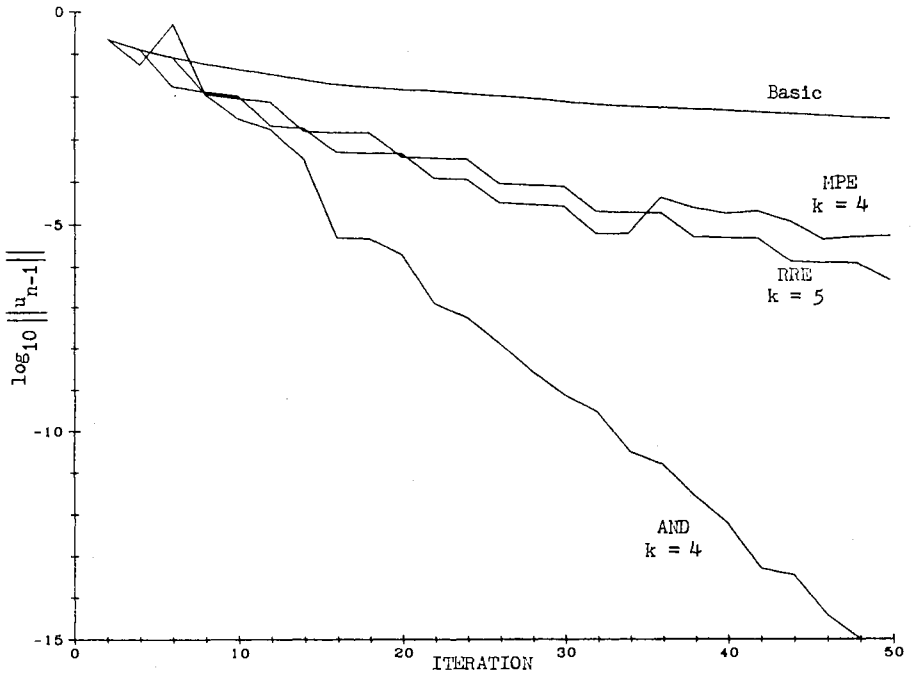

Figure 31. Results for Example 8: Rall's Problem with $\pi_0 = 1$

on the unit cube (Hyman and Manteuffel, 1984). The parameter t represents time; u_t, u_x, u_y, and u_z are the partials of u with respect to t, x, y, and z, respectively; and Δ is the Laplacian operator

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}. \tag{102}$$

Hyman and Manteuffel tested an acceleration method they developed by studying the convergence of a second-order Runge-Kutta method for this problem using $\varepsilon = 0.02$ and the Dirichlet boundary conditions

$$u(0,y,z) = u(x,0,z) = u(x,y,0) = 0 \quad \text{and}$$
$$u(1,y,z) = u(x,1,z) = u(x,y,1) = 1$$

to provide a thin boundary layer. In addition, a time step of $\Delta t = 0.5(\Delta x)$ was chosen. As in Example 7, a second order finite difference equation was used to

approximate the solution on a uniform grid of N points. They gave results for $N = 8000$ (a 20 x 20 x 20 grid of points) (Hyman and Manteuffel, 1984, p. 312).

The exact structure of their test case is not clear. First, there is a class of second order Runge-Kutta integration methods, and Hyman and Manteuffel do not mention which one of these methods they used. Gear (1971) defined the two-step calculation of a Runge-Kutta method as

$$\begin{aligned} q_1 &= y_n + ahf(y_n, t_n) \\ y_{n+1} &= y_n + bhf(y_n, t_n) + chf(q_1, t_n + dh), \end{aligned}$$

where $dy/dt = f(y,t)$; a, b, c, and d are parameters; and $h = \Delta x$. To make the expansion of y_{n+1} and the Taylor series agree as closely as possible, the relationship between the parameters must be $b = 1 - c$ and $a = d = c/2$. Therefore, the basic iteration formula for Burger's equation with $u = y$, $\Delta t = 0.5h$, and $u_t = f(y,t)$ is

$$\begin{aligned} q_1 &= u_n + a(\Delta t)(u_t)_n \\ u_{n+1} &= u_n + b(\Delta t)(u_t) + c(\Delta t)(u_t)_{n+1}, \end{aligned}$$

where $(u_t)_{n+1}$ is u_t evaluated at q_1 and $t_n + dh$. Three common second order Runge-Kutta methods are for $c = 1/2$, $3/4$, and 1 (Gear, 1971, p. 31). A second unclear area is whether the number of grid points, N, includes the boundary points. Because the software and the exact parameters for their test problem were not available, the results shown for this example are for $N = 8000$ to be the number of interior points and for $c = 1/2$. Results of the basic iteration do not exactly match those shown by Hyman and Manteuffel; however, the problem is still a good test problem due to the size of its dimension.

Results obtained for this problem are shown in Figure 32 (page 115) for the AND ($k = 4$), MPE ($k = 3$), RRE ($k = 1$), and MVε (order 3) methods for the first 50 iterations. In addition, the graphs plot the infinity norm of the error vector,

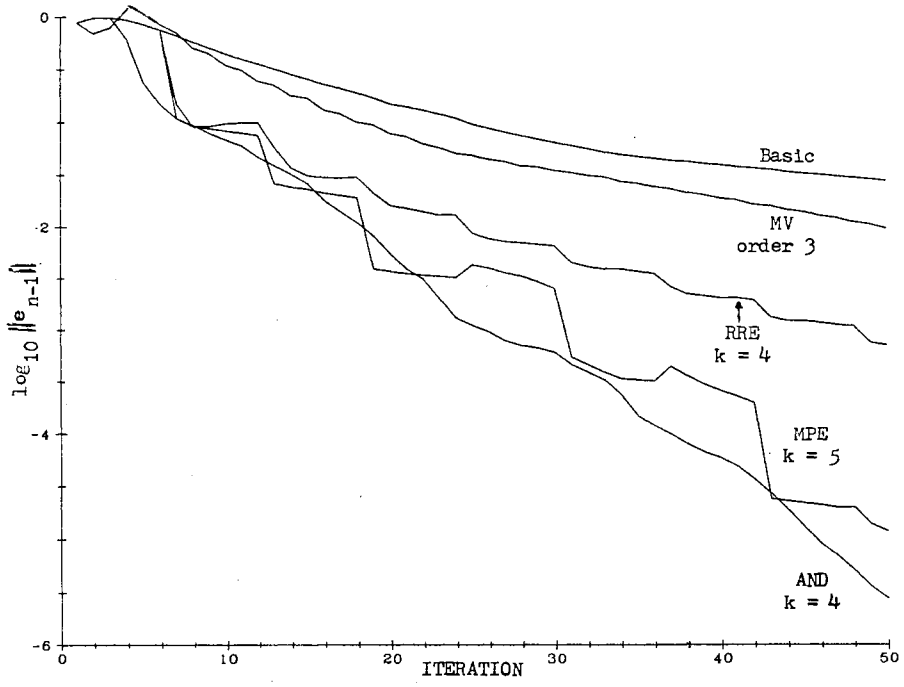

Figure 32. Results for Example 9: Hyman and Manteuffel's Test Problem

$\vec{e}_{n-1} = \vec{s} - \vec{x}_{n-1}$, instead of the difference vector. The change is to match Hyman and Manteuffel's test problem as close as possible. The only methods that consistently increase the convergence rate are the AND and MPE methods. The other methods do not work very well for this problem. In fact, the convergence of the MPE and RRE methods is not smooth as illustrated by the periodic peaks in their graphs even though the basic iteration generates a convergent sequence. In addition, the MPE method is very erratic. Hence, these two extrapolation methods have their problems in this example.

Example 10: The last example comes from Moler (1967). The problem is to solve for the eigenvalues and eigenfunctions of the Laplacian operator, Equation (102) in two variables only, on an "L" shaped region L, Figure 33 (page 116).

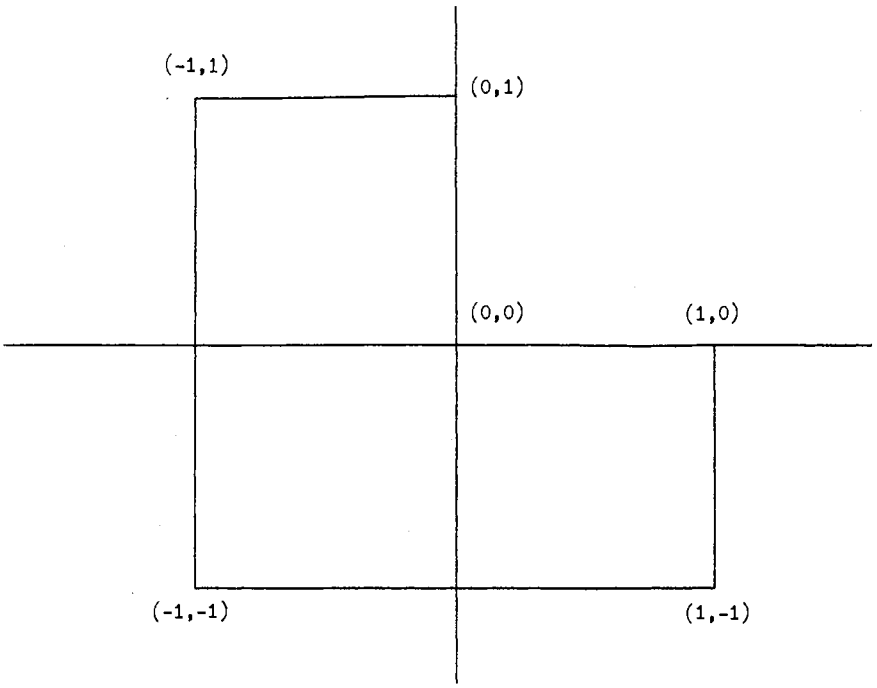

Figure 33. "L" Shaped Region for Example 10

The eigenvalues q and the functions $u(p)$, not identically zero, are to satisfy

$$\begin{aligned} &\Delta u(p) + qu(p) = 0, \quad p = (x, y) \text{ an element of } L \\ &u(p) = 0, p \text{ an element of } L. \end{aligned} \tag{103}$$

Since there are infinitely many eigenvalues, only the smallest one is considered.

Once again, the solution is approximated using finite differences over a square mesh of width $h = 1/N$, N an integer. Letting $u_{ij} = u(x_i, y_j)$, the five-point Laplacian operator is define by

$$\Delta_L u_{ij} = \left[u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{ij}\right]/h^2. \tag{104}$$

However, direct iteration of the Laplacian operator will produce the largest eigenvalue. The smallest eigenvalue can be found by use of the five-point Jacobian operator:

$$\Delta_J u_{ij} = \left[u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}\right]/4. \tag{105}$$

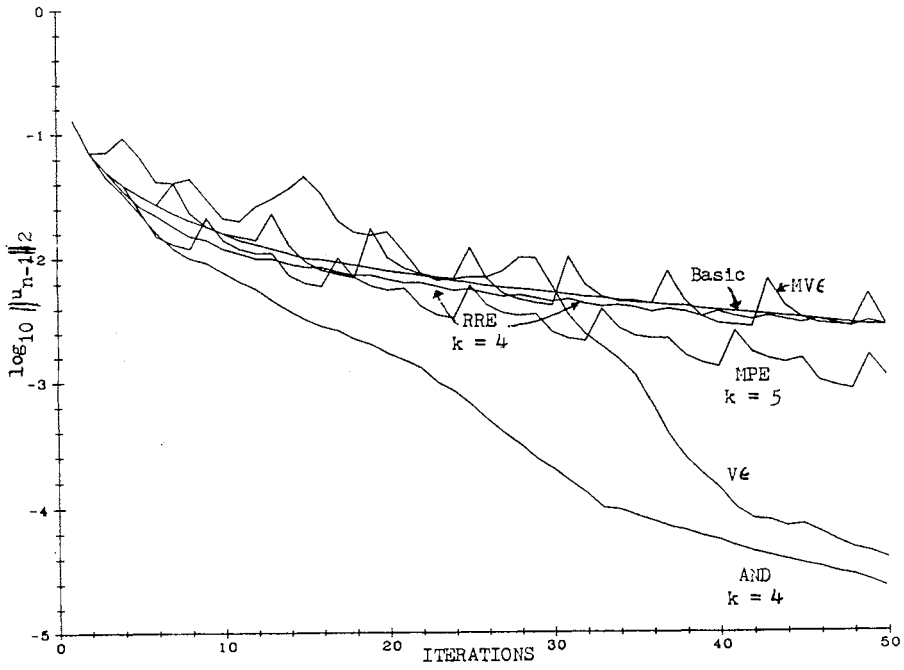

Figure 34. Results for Example 10: AND, MPE, RRE, MVε, and Vε

Denote q_L and q_J as the eigenvalues of the Laplacian and Jacobian operators, respectively. Using equations (103), (104), and (105), the following relationship holds:

$$\begin{aligned} q_L &= \left[u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{ij}\right]/h^2 u_{ij} \\ &= 4\left[(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})/4u_{ij} - 1\right]/h^2 \\ &= 4(q_J - 1)/h^2. \end{aligned} \tag{106}$$

Hence, the approach of finding q, the smallest eigenvalue, is to solve the problem by using (105) and then convert the solution from an eigenvalue of the Jacobian operator to one of the Laplacian operator by the Relation (106).

Figure 34 shows the results for the first 50 iterations for the AND, RRE, MPE, and both ε methods. For this last problem, the graphs plot the log of the Euclidean norm of the difference vector instead of the infinity norm. The RRE and

MPE methods do not work well on this problem. In fact, for values of k not shown in Figure 34, the convergence is slower than that obtained by the basic iteration. The AND and Vε methods do accelerate the convergence, but even the accelerated convergence is slow. However, the solution can be obtained in fewer iterations by applying either one of these methods. (Note that the Vε method has enormous storage and time requirements in this example.)

CHAPTER XII

THE GENERALIZED MINIMUM RESIDUAL ALGORITHM

Information on another acceleration method, the Generalized Minimum Residual (GMRES) algorithm, was received just prior to the completion of this thesis. Because of the time factor and the complexity of the software of the method, testing of the method was minimal. The GMRES algorithm was developed for linear systems by Saad and Schultz (1986). The method has had further development by Kerkhoven and Saad (1987), Brown and Saad (1987), and Burkhart and Young (1988). There now exist routines for both linear and nonlinear problems. Discussion of this method will be sketchy.

The nonlinear GMRES method resembles Newton's method for solving a system of nonlinear equations. However, GMRES reduces the effective dimension of the solution space. The method involves finding an orthonormal basis of the Krylov subspace $K_k = \text{span}\{v_1, Av_1, \ldots, A^{k-1}v_1\}$, where A is the matrix of the linear system $A\vec{x} = \vec{b}$ and $\vec{v}_1$ is the normal vector $\vec{v}_1 = \vec{r}_0/\vec{r}_0$, $\vec{r}_0 = \vec{b} - A\vec{x}_0$. The basis is found by a procedure called Arnoldi's algorithm.

Software for GMRES was obtained from Burkhart and Young (1987) of Boeing Computer Services. The test driver program that was provided for the nonlinear GMRES routine solved Laplace's equation on a square mesh, using an SSOR (symmetric successive over-relaxation) iteration (Varga, 1962) with a mesh size of $h = 1/22$ and optimizing the over-relaxation factor OMEGA automatically during the process. Nonlinear GMRES solved this problem to full double precision accuracy using a total of 129 sweeps of SSOR over the grid.

Anderson's method ($k = 5$) on this problem, with OMEGA set to 1.75, which is near the optimum value, requires only about 60 iterations to achieve full double precision accuracy. However, the GMRES software is organized in a very conservative way in order to avoid divergence. If similar search strategies were used in the Anderson routine, the number of iteration would be increased considerably. (Such search strategies should be an option of the software, used only when divergence is anticipated or detected.)

CHAPTER XIII

SUMMARY AND CONCLUSIONS

The intent of this thesis was to demonstrate the importance of acceleration methods and to compare several of these methods both theoretically and numerically. For each method, the theory and the algorithm were derived for the linear case. However, through numerically testing the algorithms on different types of problems, it was shown that the methods can be applied to both linear and nonlinear problems.

Clearly, the purpose of acceleration methods is to reduce the number of iterations required to solve numerically a mathematical problem in a vector space. All methods presented in this thesis demonstrate the capability of achieving this purpose, though the convergence rate may vary for different problems. This in itself is of great value since for the majority of practical problems reducing the number of iterations also reduces the computer time and cost. In addition, acceleration methods also have demonstrated the capability of accelerating some divergent sequences to the solution of a problem. Therefore, a greater number of problems may be solved numerically by applying an acceleration technique to the generated sequence.

There are three categories of acceleration models: the static model, the semi-dynamic model, and the fully dynamic model. If an extrapolation accelerates the convergence, then one may suspect that the fully dynamic model will provide the fastest convergence, since extrapolation is accomplished after every iterate once the first extrapolation is done. The only fully dynamic method for vectors presented,

Anderson's Generalized Secant Method, proves this intuition right. For all but a few test problems, Anderson's method is clearly superior in the number of iterations required for convergence. Even for the exceptions, Anderson's convergence rate was almost identical to the method that obtained the best results. In addition, there was not one test problem for which Anderson's method failed to converge to the solution. There were test cases, Examples 3 and 8, where the other methods had the problem of limited accuracy, the inability to achieve convergence with a precision of $C15$ even though convergence to a poorer precision is obtained. For these problems, making variations to the method by combining different relaxation parameters with different amounts of discarded iterates still did not achieve $C15$ convergence. Therefore, it is the author's conclusion that Anderson's method will consistently solve most numerical problems in fewer iterations than the other methods studied in this thesis, and that it is less susceptible to limited accuracy than are the other methods.

As stated previously, it should be emphasized that because Anderson's method does require an extrapolation every iteration after the first extrapolation, the computer time required to solve some fast iterative problems may be more than if the method is not applied. However, for most problems, especially integral equation problems and problems with a divergent generated sequence, fewer iterations is definitely desired; hence, Anderson's method will usually provide the best results.

Another area I want to stress is the reversing of the generated sequence when applying the RRE and MPE methods to a convergent generated sequence. Test results show that this procedure will produce better results (though there were a few exceptions for certain k values and a particular problem) than if the sequence is not reversed. For a divergent sequence, results prove that the original sequence produces the best result. In almost all test problems, the RRE and MPE methods gave similar results. Even though these two methods seldom equaled Anderson's method, they consistently outperformed the vector Aitken and the vector ε methods.

There are still areas of study that can be investigated. First, the GMRES method of Chapter XII can be fully tested and compared with the other methods. A second area that can receive future study is trying to convert either the RRE or MPE methods into a fully dynamic model. By using the principle introduced by Irons and Shrive (1987) in Chapter IV for the scalar case, perhaps a fully dynamic model can be derived for the RRE and/or the MPE methods.

BIBLIOGRAPHY

Aitken, A. C. "The Evaluation of the Latent Roots and Latent Vectors of a Matrix." Proceedings of the Royal Society of Edinburgh, Vol 57 (1936-37), 269-304.

Aitken, A. C. "On Bernoulli's Numerical Solution of Algebraic Equations." Proceedings of the Royal Society of Edinburgh, Vol 46 (1926), 289-305.

Albert, A. Regression and the Moore-Penrose Pseudoinverse. New York: Academic Press, 1972.

Anderson, D. G. "Iterative Procedures for Nonlinear Integral Equations." Journal ACM, Vol 12 (1965), 547-560.

Atkinson K. E. An Introduction to Numerical Analysis. New York: John Wiley & Sons, 1972.

Brezinski, C. L. "Numerical Stability of a Quadratic Method for Solving Systems of Nonlinear Equations." Computing, Vol 14 (1975), 205-211.

Brezinski, C. L. "Some Results in the Theory of the Vector ε-Algorithm." Linear Algebra and Its Applications, Vol 8 (1974), 77-86.

Brezinski, C. L. and Rieu, A. C. "The Solution of Systems of Equations Using the ε-Algorithm, and an Application to Boundary-Value Problems." Mathematics of Computation, Vol 28 (1974), 731-741.

Brent, R. P. Algorithms for Minimization Without Derivatives. Englewood Cliffs, NJ: Prentice-Hall, 1972.

Briggs, W. L. A Multigrid Tutorial. Philadelphia: Society for Industrial and Applied Mathematics, 1987.

Bromwich, T. J. I'A. An Introduction to the Theory of Infinite Series. 2nd Edition, Revised. London: MacMillan and Company, 1926.

Brown, P. N. and Saad, Y. "Hybrid Krylov Methods for Nonlinear Systems of Equations." CSRD Report No 699, Center for Supercomputing Research and Development, University of Illinois, Urbana, 1987.

Burkhart, R. H. and Young, D. P. "GMRES Acceleration and Optimization Codes." ETA-TR-88, Boeing Computer Services, 1988.

Burkhart, R. H. and Young, D. P. "Documentation for GMRES Acceleration and Optimization Codes." ETA-TR-89R1, Boeing Computer Services, 1988.

Byrne, G. D. and Hall, C. A. Numerical Solution of Systems of Nonlinear Algebraic Equations. New York: Academic Press, 1973.

Cabay, S. and Jackson, L. W. "A Polynomial Extrapolation Method for Finding Limits and Antilimits of Vector Sequences." SIAM Journal on Numerical Analysis, Vol 13 (1976), 734-752.

Chandler, J. P. "Source Code for Vector Aitken Method." Oklahoma State University, Stillwater, 1987.

Chandler, J. P. "Source Code for Minimal Polynomial Extrapolation Method." Oklahoma State University, Stillwater, 1987.

Chandrasekhar, S. Radiative Transfer. London: Oxford University Press, 1950.

Chen, M. H. "A Comparison of Some Vector Acceleration Techniques." Thesis for the Degree of Master of Science, Oklahoma State University, 1984.

Cheng, H. K. and Hafez, M. M. "Cyclic Iterative Method Applied to Transonic Flow Analyses." Pade Approximants Method and Its Applications to Mechanics. Edited by H. Cabannes. Berlin: Springer-Verlag, 1976.

Conte, S. D. and de Boor, C. Elementary Numerical Analysis: An Algorithmic Approach. New York: McGraw-Hill, 1980.

Delahaye, J. P., Sequence Transformations. Berlin: Springer-Verlag, 1988.

Eddy, R. P. "Extrapolating to the Limit of a Vector Sequence." Information Linkage Between Applied Mathematics and Industry. Edited by P. C. C. Wang. New York: Oxford University Press, 1965.

Fox, L. An Introduction to Numerical Linear Algebra. New York: Oxford University Press, 1965.

Gekeler, E. "On the solution of Sytems of Equations by the Epsilon Algorithm of Wynn." Mathematics of Computation, Vol 26 (Apr. 1972), 427-435.

Golub, G. H. "Numerical Methods for Solving Linear Least-squares Problems." Numerische Mathematik, Vol 7 (1965), 206-216.

Golub, G. H. and Van Loan C. F. Matrix Computations. Baltimore: Johns Hopkins University Press, 1983.

Grevill, T. N. E. "The Pseudoinverse of a Rectangular Matrix and its Applications to the Solution of Systems of Linear Equations." SIAM Review. Vol 1 (1959), 38-43.

Hardy, G. H. Divergent Series. Oxford: Clarendon Press, 1949.

Henrici, P. Elements of Numerical Analysis. New York: John Wiley & Sons, 1964.

Householder, A. S. The Theory of Matrices in Numerical Analysis. New York: Blaisdell Publishing Company, 1964.

Hyman, J. M. and Manteuffel, T. A. "Dynamic Acceleration of Nonlinear Iterations." Elliptic Problem Solvers II. Edited by G. Birkhoff and A. Schoenstadt. New York: Academic Press, 1984.

Irons, B. and Shrive, N. G. Numerical Methods in Engineering and Applied Science: Numbers are Fun. New York: John Wiley & Sons, 1987.

Jennings, A. "Accelerating the Convergence of Matrix Iterative Processes." Journal Inst. Maths Applics, Vol 8 (1971), 99-110.

Lubkin, S. "A Method of Summing Infinite Series." Journal of Research, National Bureau of Standards, B48 (1952), 228-254.

Kerkhoven T. and Saad, Y. "Acceleration Techniques for Decoupling Algorithms in Semiconductor Simulation." CSRD Report No 686, Center for Supercomputing Research and Development, University of Illinois, Urbana, 1987.

McLeod, J. B. "A Note on the ε-Algorithm." Computing. Vol 7 (1971), 17-24.

Mešina, M. "Convergence Acceleration for the Iterative Solution of the Equations $X = AX + f$." Computer Methods in Applied Mechanics and Engineering. Vol 10 (1977), 165-173.

Moler, C. "Extrapolation to the Limit." Numerical Analysis. University of Michigan Engineering Summer Conference, Numerical Analysis Course, 1987.

Milne, W. E. Numerical Calculus. New Jersey: Princeton University Press, 1950.

Moore, C. N. "Summable Series and Convergence Factors." American Mathematical Society. Vol XXII, 1938

Moore, E. H. "On the Reciprocal of the General Algebraic Matrix." Bulletin for American Mathematical Society. Vol 26 (1920), 394-395.

Noble, B. Applied Linear Algebra. Englewood Cliffs, NJ: Prentice-Hall, 1969.

Ortega, J. M. and Poole, W. G., Jr. An Introduction to Numerical Methods for Differential Equations. Marshfield, MA: Pitman Publishing Inc., 1981.

Ortega, J. M. and Rheinboldt, W. C. Iterative Solution of Nonlinear Equations in Several Variables. New York: Academic Press, 1970.

Penrose, R. "A Generalized Inverse for Matrices." Proceedings of the Cambridge Philosophical Society. Vol 51 (1955), 406-413. MR 16, 1082.

Pringle, R. M. and Rayner, A. A. Generalized Inverse Matrices with Applications to Statistics. New York: Hafner Publishing Company, 1971.

Rall, L. B. Computational Solution of Nonlinear Operator Equations. New York: John Wiley & Sons, 1969.

Rao, C. R. and Mitra, S. K. Generalized Inverse of Matrices and its Applications. New York: John Wiley & Sons, 1971.

Rohde, C. A. "Contributions to the Theory, Computation and Application of Generalized Inverses." Mimeo No 392, Institute of Statistics, University of North Carolina, Raleigh, 1964.

Saad, Y. and Schultz M. H. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems." SIAM Journal for Science and Statistic Computation, Vol 7 (1986), 856-869.

Schmidt, J. R. "On the Numerical Solution of Divergent and Slowly Convergent Sequences." Journal of Mathematics and Physics, Vol 34 (1955), 1-42.

Shanks, D. "Non-linear Transformations of Divergent and Slowly Convergent Sequences." Journal of Mathematics and Physics, Vol 34 (Apr. 1955), 1-42.

Sidi, A. "Convergence and Stability Properties of Minimal Polynomial and Reduced Rank Extrapolation Algorithms." SIAM Journal on Numerical Analysis, Vol 23 (1986), 197-209.

Sidi, A., Ford, W. R., and Smith, D. A. "Acceleration of Convergence of Vector Sequences." SIAM Journal on Numerical Analysis, Vol 23 (1986), 178-196.

Skelboe, S. "Computation of the Periodic Steady-State Response of Nonlinear Networks by Extrapolation Methods." IEEE Transactions on Circuits and Systems, Vol 27 (1980), 161-175.

Smith, D. A., Ford, W. F., and Sidi, A. "Extrapolation Methods for Vector Sequences." SIAM Review, Vol 29 (1987), 199-233.

Townsend, M. A. "An Introduction to the Acceleration of Scalar Sequences." Dissertation for the Degree of Doctor of Education, Oklahoma State University, 1983.

Traub, J. F. Iterative Methods for the Solution of Equations. New York: Prentice-Hall Inc., 1964.

Varga, R. S. Iterative Matrix Analysis. Englewood Cliffs, New Jersey: Prentice-Hall, 1962.

Wilkinson, J. H. The Algebraic Eigenvalue Problem. Belfast: Oxford University Press, 1965.

Wimp, J. Sequence Transformations and Their Applications. New York: Academic Press, 1981.

Wolfe, P. "The Secant Method for Simultaneous Nonlinear Equations." Comm. ACM, Vol 2 (1959), 12-13.

Wynn, P. "On a Device for Computing the $e_m(S_n)$ Transformation." Mathematical Tables and Other Aids to Computation, Vol 10 (1956), 91-96.

Wynn, P. "The Rational Approximation of Functions which are Formally Defined by a Power Series Expansion." Mathematics of Computation, Vol 14 (1960), 147-186.

Wynn, P. "Acceleration Techniques for Iterated Vector and Matrix Problems." Mathematics of Computation, Vol 16 (1962), 301-322.

Wynn, P. "General Purpose Vector Epsilon Algorithm Procedures." Numerische Mathematik, Vol 6 (1964), 22-36.

Young, D. M. Iterative Solution of Large Linear Systems. New York: Academic Press, 1971.

Zygmund, A. Trigonometric Series, Vol I. Cambridge University Press, 1959.

APPENDIX

CORRECTIONS TO ARTICLE WRITTEN BY SMITH, FORD, AND SIDI (1987)

David A. Smith, William F. Ford, and Avram Sidi wrote an article, "Extrapolation Methods for Vector Sequences," in the *SIAM Review*, Vol 29 (1987) comparing acceleration techniques. These methods included the vector epsilon method with both types of inverses, the generalized and the primitive; the MPE method; and the RRE method. Several of the comments in the paper concerning their test results are not correct. This Appendix details the errors and corrections needed, if appropriate.

In their Example 2 (Example 2 in Chapter XI also), they claim that the RRE and Vector Epsilon methods "converge" to a vector approximately equal to $(13.36, -1.940, 5.532, -5.342)$. This is not the case. Both methods converge to the unique solution $(1,1,1,1)$. When converting the problem to the Gauss-Seidel iteration scheme (88), they continued to use the original vector $\vec{b} = (10,4,8,6)^T$ instead of the converted vector $(D+L)^{-1}\vec{b} = (5, 1/3, -11/9, 163/9)^T$. As a result they determined the solution of a different fix point problem, and their method converged to the correct solution for their incorrect problem. Using the correct converted vector, the RRE and Vector Epsilon methods converge nicely to the solution $(1,1,1,1)$.

They also state that because the system has a zero eigenvalue, the system of equations is singular for $k = 4$. However, for the initial vector they used, $(0,0,0,0)$, the error vector does not lie in a subspace spanned by any three eigenvectors of the

iteration matrix, and the system is not singular for this starting point and value of k. As a result, the MPE method is exact, in the absence of rounding error, for $k = 4$ but not for $k = 3$. If the initial iterate is discarded, as discussed in Chapter XI, then $k = 3$ is appropriate.

For their Examples 1 and 8 (Examples 3 and 4 in Chapter XI), they stated that the RRE method failed to converge to the correct solution. Test results show that the RRE method does converge to the solution in both cases. For the first example, convergence is very similar to that obtained by the MPE method and is obtained for all values of k, though the convergence is hampered by the problem of limited accuracy. The second example is a problem with two solutions. The RRE method ($k = 4$) converges to the same solution as the basic iteration, $(3, 3, 3, 3)$. However, it should be noted that for $k < 4$ the RRE method caused system overflow for this problem.

VITA

Steven R. Capehart

Candidate for the Degree of

Doctor of Education

Thesis: TECHNIQUES FOR ACCELERATING ITERATIVE METHODS FOR THE SOLUTION OF MATHEMATICAL PROBLEMS

Major Field: Higher Education

Biographical:

Personal Data: Born in Fort Smith, Arkansas, October 3, 1949, the son of Eddie and Willie Capehart.

Education: Graduated from Northside High School, Fort Smith, Arkansas, in June 1967; received Bachelor of Arts degree in Mathematics from Arkansas Polytechic College at Russellville in May, 1971; received Master of Arts degree in Mathematics from University of Arkansas at Fayetteville in May 1975; Completed requirements for the Doctor of Education degree at Oklahoma State University in July, 1989.

Professional Experience: High school mathematics teacher and coach, Southside High School, Fort Smith, Arkansas, August, 1971, to May, 1977; part-time instructor, Department of Mathematics, Westark Community College, Fort Smith, Arkansas, January, 1976 to May, 1977; officer in the United States Air Force, May, 1977 to present; instructor, Department of Mathematics, United States Air Force Preparatory School, Coloardo Springs, Colorado, June, 1982 to May, 1984; instructor, Department of Mathematical Sciences, United States Air Force Academy,Colorado Springs, Colorado, June, 1984 to May, 1986.