An Investigation into the Numeric Solution of Linear Recurrence Relations

Dennis Lee Kapenga

Western Michigan University

Follow this and additional works at: https://scholarworks.wmich.edu/masters_theses

Part of the Computer Sciences Commons

Recommended Citation
Kapenga, Dennis Lee, "An Investigation into the Numeric Solution of Linear Recurrence Relations" (1986). Master's Theses. 1333.
https://scholarworks.wmich.edu/masters_theses/1333

This Masters Thesis-Open Access is brought to you for free and open access by the Graduate College at ScholarWorks at WMU. It has been accepted for inclusion in Master’s Theses by an authorized administrator of ScholarWorks at WMU. For more information, please contact maira.bundza@wmich.edu.
AN INVESTIGATION INTO THE NUMERIC SOLUTION
OF LINEAR RECURRENCE RELATIONS

by

Dennis Lee Kapenga

A Thesis
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Master of Science
Department of Computer Science

Western Michigan University
Kalamazoo, Michigan
August 1986
AN INVESTIGATION INTO THE NUMERIC SOLUTION OF LINEAR RECURRENCE RELATIONS

Dennis Lee Kapenga, M.S.
Western Michigan University, 1986

This is an investigation of the use of some techniques from numerical linear algebra in solving linear recurrence relations.

The classic methods of Oliver and Lozier are compared with several alternatives. These alternatives center on using advances in the theory of numerical linear algebra, computer software, and raw computer performance which have been made in the 15 years since the original research into linear recurrence relations was done.

A set of test examples is presented to provide a basis to compare solution methods. This set of examples was selected to allow experiments associated with the Poincare class of recurrence relations.

A flexible software system was assembled, which includes access to LINPACK routines, some with extensions, to provide an environment for quick implementation and comparison of solution methods.
ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to the many people involved in the preparation of this study. My brother John, whose theoretical advice made this study possible. Pamela Gilchrist, who provided moral support on all of the long nights during this study. Thanks also to the many individuals involved in the proofreading process.

I would also like to thank Capital University, Western Michigan University, and Delft University of Technology, the Netherlands, for the use of computing facilities.

Dennis Lee Kapenga
INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

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

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

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

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University Microfilms International
300 N. Zeeb Road
Ann Arbor, MI 48106

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Kapenga, Dennis Lee

AN INVESTIGATION INTO THE NUMERIC SOLUTION OF LINEAR RECURRENCE RELATIONS

Western Michigan University M.S. 1986

University Microfilms International 300 N. Zeeb Road, Ann Arbor, MI 48105

Copyright 1986 by Kapenga, Dennis Lee

All Rights Reserved
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS ................................ ii

CHAPTER

I. INTRODUCTION ................................... 1
   Background ...................................... 3
   Goals and Objectives ........................... 4
   Current Status ................................. 6
   Selection of Examples .......................... 7
   Summary of Chapters ............................ 9

II. GENERAL DEFINITIONS AND NOTATIONS .......... 12
   General Linear Recurrence Relations ......... 13
   Constant Coefficient Linear Recurrence
   Relations ....................................... 14
   Homogeneous Linear Recurrence Relations . 15
   Poincare Recurrence Relations .............. 16
   Recurrence Relations and Characteristic
   Polynomials ................................. 17
   Representation of Solutions
   of Recurrence Relations .................... 20
   Poincare's, Perron's and
   Birkhoff-Trjitzinsky Theorems ............ 23
   Solutions of Nonhomogeneous Recurrence
   Relations and the Concept of Dominance .. 26
   Properties of Initial and Terminal
   Conditions and Their Effect on
   Recurrence Relations ........................ 29
   The Associated Linear System of Equations
   for a Recurrence Relation .................. 31
### Table of Contents—Continued

**CHAPTER**

- Basic Matrices and Vectors .......................... 31
- The Augmented and Reduced Forms for the Associated Linear System .......... 33
- The Introduction to Computational Errors and Error Comparison Notations .......... 38

**III. GENERAL SOLUTIONS OF RECURRENCE RELATIONS .. 41**

- Simple Methods for the Solution of Recurrence Relations .................. 42
  - Forward Recurrence .................................. 42
  - Backward Recurrence .................................. 43
- Boundary Value Problems .................................. 45
- Conditions for Stability of Solutions to Recurrence Relations ................. 46
  - Stability in the Absolute Sense for Forward Recurrence .................. 46
  - Stability in the Absolute Sense for Backward Recurrence ................ 51
- Weakly Stable in Either Direction .... 54
- An Example of Contrast ....................... 61

**More Advanced Approaches to the Solution of Recurrence Relations .......... 69**

- The Miller Algorithm for a First Order Recurrence Relation ............... 69
- The Miller Algorithm for a Second Order Recurrence Relation ............ 71
- Oliver's Theorem ....................... 80
Table of Contents—Continued

CHAPTER

IV. EMPIRICAL INVESTIGATION AND COMPARISONS .... 83
   Condition Number of a Matrix ............... 84
   Augmented vs. Reduced Linear System of
   Equations ...................................... 95
   LU vs. QR Decomposition ................... 107
   Pivoting vs. Not Pivoting in the
   Context of the LU Decomposition .......... 109
   Space and Time Comparisons .............. 122
   Enhancements and Modifications .......... 126
      Weighting of Initial Conditions
      for the LU Decomposition .............. 126
      Pivoting Strategies Other Than
      Partial Pivoting and Not Pivoting .... 135

V. LEAST SQUARES .............................. 138
   Introduction to Least Squares .......... 139
   Least Squares and Recurrence Relations .. 141
   Weighting for Least Squares .......... 150

VI. SOFTWARE ................................. 166
   Existing Software .......................... 168
      LU Decomposition ......................... 170
      QR Decomposition ......................... 172
      Least Squares ........................... 173
   New Software ............................... 174
      JOBHLP ..................................... 175
      SAROUT (DAROUT) .......................... 175

v

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCPOUT (DCPOUT)</td>
<td>176</td>
</tr>
<tr>
<td>SGTVAL (DGTVAL)</td>
<td>178</td>
</tr>
<tr>
<td>SLSOUT (DLSOUT)</td>
<td>178</td>
</tr>
<tr>
<td>SLSSCL (DLSSCL)</td>
<td>180</td>
</tr>
<tr>
<td>SLUOUT (DLUOUT)</td>
<td>180</td>
</tr>
<tr>
<td>SLUSCL (DLUSCL)</td>
<td>182</td>
</tr>
<tr>
<td>SMTGEN (DMTGEN)</td>
<td>182</td>
</tr>
<tr>
<td>SQRLST (DQRLST)</td>
<td>184</td>
</tr>
<tr>
<td>SQROUT (DQROUT)</td>
<td>184</td>
</tr>
<tr>
<td>SRECUR (DRECUR)</td>
<td>186</td>
</tr>
<tr>
<td>SRIGHT (DRIGHT)</td>
<td>186</td>
</tr>
<tr>
<td>SSELCT (DSELCT)</td>
<td>187</td>
</tr>
<tr>
<td>SZCOPY (DZCOPY)</td>
<td>188</td>
</tr>
<tr>
<td>Sample Execution of Software</td>
<td>189</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>199</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>266</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

Linear recurrence relations are an integral part of many disciplines. In mathematics, physics, chemistry, biology, and others, the need to solve complex formulas for generating numeric sequences has always been a problem. Since the earliest days of modern science, it has been known that the solution of some of these complex formulas could be rewritten as much simpler recurrence relations. Although the associated recurrence relations might still be difficult to solve, it is usually the case that the recurrence relation is less complex than the original formula.

The introduction of the computer for the solution of recurrence relations significantly reduced the time to perform the computations. This reduction of computational time was offset by the introduction of computer based errors in the computations. The restriction on precision, or inability to store some floating point numbers exactly, is the major source of these errors.
This work will include both a theoretical and empirical investigation into the solution of recurrence relations. The inclusion of a brief error analysis for the different computational methods will provide a basis for the comparison of the different solution methods. The goal is to find a particular computational method that tries to minimize the errors introduced into the computations by the use of a computer. Several known solution methods will be described and compared with new methods proposed in this work.

This chapter will provide a brief background which was the motivating force for this work. Then a statement of the initial goals and objectives for this study will be presented. These goals and objectives will then be compared to the current status of the thesis as submitted. Guidelines for the choice of examples for the empirical comparisons will be discussed. The guidelines were prepared so that a set of examples could be chosen to represent typical computations, which should exclude bias. Finally a brief summary of the other chapters is provided.

Throughout this work the special symbol [ ] will be used to indicate the termination of a definition, a notation, an example, or a theorem.
Background

The motivation for this work was provided by R. Piessens (1985), who suggested using a singular value decomposition, in an algorithm, to solve recurrence relations. The suggestion provides a method for solution, at least in spirit, similar to the work of Oliver (1967, 1968a, 1968b). Oliver's work did not involve the use of the singular value decomposition, but it can be shown in many cases to involve the same computations as if an LU decomposition were used. The suggestion by Piessens leads to a large number of questions when a recurrence relation is viewed as a linear system of equations: such as, can any other numerical solution techniques be used to advantage? The availability of efficient and accurate software for the manipulation of linear systems (Dongarra et al. 1978; Moler 1980) makes this a good subject for investigation.

A review of the literature shows that the research in this area involved the use of the LU decomposition and the reduced form of the associated linear system of equations for the recurrence relation.

J. Kapenga (1985) provided several suggestions for investigation. Kapenga noted that the same ideas Piessens proposed involving the singular value decomposition could be carried out using the QR
decomposition. Furthermore, the suggestions of using least squares approximations for the solution of recurrence relations and the augmented form of the associated linear system of equations was made.

A theoretical investigation into Kapenga's suggestions showed that these new approaches might provide reasonable results. In the case of using the augmented matrix, more flexibility involving the initial conditions was observed.

At this point it became necessary to provide a reliable environment for the testing and comparison of solution methods. The software developed would provide initial results to analyze. If the initial results proved favorable, further theoretical investigation would be justified.

Goals and Objectives

The following goals and objectives were set forth after the initial investigation of the problem.

1. Produce a well documented, reliable, easy to use, portable, and easy to modify environment for the testing and comparison of solution methods. The software would be a set of FORTRAN routines. The use of the publicly available LINPACK routines (Dongarra 1978) would serve as the underlying software.
2. Collect the basic definitions and notations that would provide the needed background for an understanding of the results. This background should be sufficient enough so that justification of results can be provided. Without a sufficient background, the selection of examples could be extremely biased and therefore unable to provide meaningful results.

3. Carefully select a set of examples that could be considered typical, but still expected to differentiate between the different solution techniques. This selection process is not to find special cases, but to find general patterns that hold for a large range of recurrence relations.

4. Investigate the set of examples chosen and try to form conclusions about the different solution methods. The conclusions formed here are not to be considered definitive.

5. Based on the above conclusions suggest possible improvements and modifications to the solution techniques.

6. Incorporate the improvements and modifications into the software, design new examples for testing, test the examples, and evaluate the results of the examples in hopes that new conclusions can be formed.
Current Status

Table 1.1 will describe the current status of the major components of the thesis. The terms complete, some, and little estimate the amount of work completed compared to expectations for a thorough analysis.

The observations on the use of the augmented form of the associated linear system of equations, a new pivoting strategy for the LU decomposition, and a method for weighting the initial conditions of the LU decomposition to prevent the initial values from pivoting have come about in connection with this work. These new ideas will be described in chapter four.

Table 1.1

<table>
<thead>
<tr>
<th>Component</th>
<th>Amount of Investigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>software</td>
<td>Complete Some Little</td>
</tr>
<tr>
<td>(LU, QR, least squares)</td>
<td>XX</td>
</tr>
<tr>
<td>LU decomposition investigation</td>
<td>XX</td>
</tr>
<tr>
<td>QR decomposition investigation</td>
<td>XX</td>
</tr>
<tr>
<td>least square investigation</td>
<td>XX</td>
</tr>
<tr>
<td>examples</td>
<td>XX</td>
</tr>
<tr>
<td>new techniques</td>
<td>XX</td>
</tr>
</tbody>
</table>
Selection of Examples

The selection of examples must be done carefully so as not to introduce erroneous conclusions. The selection process should provide a typical set of examples, but still be able to differentiate between the different methods. This means that examples are chosen to show strong or weak points about a particular method without providing a rare case. By choice, Oliver's method (1967, 1968a, 1968b) of defining the number of terminal and initial conditions required for a particular solution will be used. This method is presented in chapter three.

When choosing a recurrence relation for an example, the following things need to be considered and included:

1. The number of solutions
2. The number of values to compute
3. Which solution to solve for (minimal, ..., dominate)
4. The magnitude of the desired solution
5. The asymptotic behavior of the desired solution
6. The asymptotic behavior of the desired solution compared to the other solutions
7. The ability to represent the solution exactly

All of the above considerations are independent of which method of solution is chosen. The choice of which linear system representation is used is not effected by the above considerations. In the choosing of examples to
differentiate between the different solution methods, all of the above considerations should be included in the investigation.

The following list provides some of the common methods for establishing "Bench Marks" for the comparison of solution methods:

1. Collect examples from the published literature
2. Generate special examples to show particular points
3. Generate many random examples and provide a reliable statistical summary
4. Generate a set of examples \( (E(a)) \), \( 0 \leq a \leq 1 \), where \( E(0) \) is an extremely easy problem and \( E(1) \) is very difficult; determine the value for \( a \) that will cause each solution method to break down
5. Generate a set of examples \( (E(a)) \), \( 0 \leq a \leq 1 \), where the problems are approximately of equal complexity, for \( a \) an element of \([0,1]\), and test \( (E(a_i))_{i=1}^{n} \) for some set \( \{a_i\}_{i=1}^{n} \)

The "Bench Marks" presented here, and many not explicitly given, provided the basis for comparisons and conclusions found throughout this work.

The results for this work were generated with single precision accuracy. This choice of accuracy does not affect the results presented. The interest lies in the comparison of methods, not the accuracy of results.
The machines used for the computations included a 32 bit Hewlett Packard 9000, a Digital Equipment Corp. System-10 with a word length of 36 bits, and a Control Data Corp. Cyber 815 with a word length of 60 bits.

Summary of Chapters

In chapter two, the general definitions for recurrence relations are presented. Then the subclass of Poincare type recurrence relations is defined. This is the subclass of most interest in this work. The relation of recurrence relations to polynomials, along with the solutions for recurrence relations, provides the basis for a linear algebra approach to the solution of recurrence relations. The notations for both the augmented and reduced associated linear system are provided. Finally, an introduction to the errors involved in the computer solution of recurrence relations and the notation for errors is given.

In chapter three a discussion of several known methods for the solution of recurrence relations is provided. The simple methods of forward and backward recurrence will be shown to work for particular solutions, but not in general. The Miller and Oliver methods will be shown to work somewhat better, but again not in all cases. The stability of computations for a
particular solution method will be presented and shown to be a good indicator of whether the results should be trusted. Error propagation is discussed in more detail and associated with the different methods of solution. Many examples will be included.

Chapter four contains an empirical investigation into the different solution methods and options available for computing the solution of a recurrence relation. A relationship between a condition number for a matrix and the reliability of the computations of the solution using that matrix is discussed.

Several comparisons between various solution methods to recurrence relations are presented. Conclusions about a particular comparison will be supported by examples. A discussion of each conclusion with possible reasons and insights will be given.

Some of the comparisons considered are: LU vs. QR decomposition, pivoting vs. not pivoting for the LU decomposition, augmented system vs. reduced system, and the space and time concerns of the different solution methods.

Finally, enhancements and modifications to the existing solution methods, that were discovered during this work, will be presented, along with examples in support of the new ideas.
Chapter five introduces the least squares method for the solution of recurrence relations. A discussion of the basic least squares method is followed by possible enhancements to provide better overall results. The introduction of weighting will be presented. Comparisons through examples will show the effects of the different solution strategies on the relative and absolute errors in the solution.

In chapter six will be a presentation of the software environment used for this work. A brief description of the existing software package, LINPACK, will be given. Some of the routines from LINPACK were altered to provide special results. A description of the altered routines will be included. A discussion of newly developed software, along with a sample execution will be provided. The actual FORTRAN coding for all of the altered or newly developed routines can be found in the appendix.
CHAPTER II

GENERAL DEFINITIONS AND NOTATIONS

In this chapter general definitions are provided. Then the class of linear recurrence relations is partitioned into a few subclasses, some of which this study will examine in detail. Poincare type recurrence relations form the subclass of most interest in this work.

Constant coefficient recurrence relations form a subclass of the Poincare recurrence relations. Due to the ease of representation, closed forms of solutions, and the close relationship to Poincare recurrence relations, this class is important. The constant coefficient class will provide material for many enlightening examples.

The representation of recurrence relations as linear systems will be presented. There are many properties of recurrence relations that are best expressed in terms of their associated linear system.

The definition of a basis of solutions, initial and terminal conditions, and general solutions will be given.
Finally, a brief introduction to the analysis of errors in the computer solution to recurrence relations will be presented. Minimization of the propagation of errors will be discussed in later chapters.

Most of the information and results discussed in this chapter can be found in standard treatments of recurrence relations (Wimp 1984; Milne-Thompson 1960; Miller 1968; Hildebrand 1968).

General Linear Recurrence Relations

Now general linear recurrence relations will be defined. After the definitions of general linear recurrence relations, some specific subclasses are defined for use throughout this thesis.

Definition 2.1

The general form of a linear $p$th order recurrence relation is given by:

$$y(n) + a_1(n)y(n-1) + \ldots + a_p(n)y(n-p) = f(n),$$

$n = p, p+1, \ldots$; where $a_p(n) \neq 0$ for any $n$. []

Example 2.2

The following are examples of general recurrence relations. The type and some characteristics of each will be explained further, later in this chapter. The first example is a second order recurrence relation.
(2.2.1) 

\[ y(n) - 3y(n-1) + 2y(n-2) = 0. \]

The next example has as solutions the Bessel functions \( J_n(x) \) and \( Y_n(x) \) (Abramowitz and Stegun 1964);

(2.2.2) 

\[ y(n) + \frac{2(n-1)}{x} y(n-1) - y(n-2) = 0, \]

where \( x \) is a constant for the computation of \( y(0), y(1), \ldots \).

The third example is that of a more complex third order recurrence relation,

(2.2.3) 

\[
\begin{align*}
    y(n) + (n^2+1)y(n-2) - (n^2+2)y(n-2) + (n^2+3)y(n-3) \\
    = n^2+n.
\end{align*}
\]

Constant Coefficient Linear Recurrence Relations

The class of constant coefficient recurrence relations is a special case of the general class (2.2.1) where \( a_i(n), i = 1, 2, \ldots p \) are constants instead of functions of \( n \).

**Definition 2.3**

The general form of a pth order linear constant coefficient recurrence relation is given by:
\[ y(n) + a_1 y(n-1) + a_2 y(n-2) + \ldots + a_p y(n-p) = f(n), \]
\[ n = p, p+1, \ldots ; \text{ where, } a_1, a_2, \ldots, a_p \text{ are constants, and } a_p \neq 0. \]

**Example 2.4**

The following are constant coefficient recurrence relations: example 2.2.1 above, and
\[ y(n) + 345y(n-1) + 3y(n-2) - 123y(n-3) + 1256y(n-4) = n(n+1). \]

**Homogeneous Linear Recurrence Relations**

The next definition (2.5), that of a homogeneous recurrence relation, is the first step toward the eventual goal of being able to characterize solutions to general linear recurrence relations.

**Definition 2.5**

The recurrence relation 2.1.1 is said to be homogeneous if \( f(n) = 0 \) for all \( n \); otherwise it is said to be nonhomogeneous. For equation 2.1.1 to be characterized as homogeneous, it must possess the following form:
\[ y(n) + a_1 y(n-1) + \ldots + a_p(n)y(n-p) = 0, \]
\[ n = p, p+1, \ldots ; \text{ where } a_p(n) \neq 0 \text{ for any } n. \]
The next examples (2.6) will present both homogeneous and nonhomogeneous recurrence relations.

**Example 2.6**

Examples 2.2.1 and 2.2.2 are homogeneous; whereas, examples 2.2.3 and 2.4.1 are nonhomogeneous.

The following example is also nonhomogeneous:

\[(2.6.1)\]

\[y(n) - 3y(n-1) + 2y(n-2) = (-1)^{n+1} \times 6. \]

**Poincare Recurrence Relations**

Now the definition for Poincare type recurrence relations will be given. This is the main class of interest for the remainder of this work.

**Definition 2.7**

A linear recurrence relation (2.1.1) is said to be of Poincare type if

\[(2.7.1)\]

\[\lim_{n \to \infty} a_k(n) = A_k \text{ exists, is finite for each} \]

\[k = 1, 2, \ldots, p, \text{ and } A_p \neq 0. \]

The constant coefficient class of recurrence relations (from definition 2.3) is a special case of Poincare recurrence relations, and will be investigated throughout the remainder of this work.
Example 2.8

An example of a Poincare type recurrence relation follows:

\[ y(n) - \frac{n-1}{n} y(n-1) + \frac{n-2}{n} y(n-2) = 0. \]

Note that in the above example (2.8) as \( n \) goes to infinity, the limits of the coefficients go to -3 and 1 respectively.

Recurrence Relations and Characteristic Polynomials

Now that some classes of recurrence relations have been defined, it is time to turn our attention to specific details.

The next definition (2.9) gives a correspondence between polynomials and recurrence relations.

Definition 2.9

The characteristic equation for a Poincare recurrence relation (2.7.1) is defined as the pth order polynomial

\[ x^p + A_1 x^{p-1} + A_2 x^{p-2} + \ldots + A_{p-1} x + A_p = 0, \]

where \( A_k, k = 1, 2, \ldots, p \), are defined by equation 2.7.1. []
Example 2.10

The characteristic equation for example 2.2.1 is

\[(2.10.1)\]
\[x^2 - 3x + 2 = 0.\]

The second example is the following seventh order constant coefficient recurrence relation.

\[(2.10.2)\]
\[2187y(n) - 21870y(n-1) + 90639y(n-2) - 201528y(n-3) + 259308y(n-4) - 192816y(n-5) + 76608y(n-6) - 12544y(n-7) = 0,\]
whose characteristic equation is:

\[(2.10.3)\]
\[2187x^7 - 21870x^6 + 90639x^5 - 201528x^4 + 259308x^3 - 192816x^2 + 76608x - 12544 = 0.\]

For constant coefficient recurrence relations (2.3), once the roots for the characteristic equation have been determined, they can be used to produce \(p\) solutions. It will be shown later that these \(p\) solutions are linearly independent and form a basis for the set of all solutions of the recurrence relation.

For Poincaré recurrence relations (2.7.1), the roots of the characteristic equation also give information on the behavior of the set of solutions; although, as should be expected this information is in the form of asymptotics as \(n\) goes to infinity.
The work of Branders (1974), and that of Denef and Piessens (1974) provide an understanding of the asymptotic behavior of solutions to recurrence relations. This information is important in order to generate examples and understand the conclusions obtained through an empirical investigation.

**Definition 2.11**

For a Poincare recurrence relation (2.7.1), the \( p \) roots of the characteristic equation (2.9.1) are called the characteristic values of the recurrence relation, and will be denoted as

\[
X_1, X_2, \ldots, X_p.
\]

Without a loss of generality we will assume that

\[
X_1 \leq X_2 \leq \ldots \leq X_p.
\]

The \( p \) characteristic values will also define \( L \) unique roots as follows:

\[
r_1 < r_2 < \ldots < r_L;
\]

with multiplicities \( N_1, N_2, \ldots, N_L \) such that

\[
\sum_{i=1}^{L} N_i = p.
\]
The following example (2.12) will help in the understanding of definition 2.11. Two previous examples will be examined for their characteristic values.

**Example 2.12**

The characteristic values for example 2.2.1 are

\[(2.12.1)\]

\[x_1 = 1, \ x_2 = 2.\]

The roots for equation 2.10.2 which are the characteristic values for the recurrence relation 2.10.1 are

\[(2.12.2)\]

\[x_1 = 2/3, \ x_2 = 2/3, \ x_3 = 4/3, \ x_4 = 4/3, \ x_5 = 4/3, \ x_6 = 7/4, \ x_7 = 7/4. \]}

**Representation of Solutions of Recurrence Relations**

The following definition (2.13) will provide the needed conditions for a set of functions to be linearly independent.

**Definition 2.13**

A set of functions:

\[(2.13.1)\]

\[\{ y_i : i = 1, 2, \ldots, N \} \text{ each with domain } D^+, \]

is said to be linearly independent over D when,
\[(2.13.2)\]
\[
\sum_{i=1}^{N} c_i y_i(n) = 0, \text{ for all } n \text{ in } D,
\]
implies \(c_i = 0\) for \(i = 1, 2, \ldots, N\). []

The next theorem (2.14) contains the definition of a basis for the solutions of a recurrence relation.

**Theorem 2.14**

For a homogeneous recurrence relation (2.5.1) of order \(p\), if \(p\) linearly independent solutions for the recurrence relation are \(s_1, s_2, \ldots, s_p\), then these \(p\) solutions form a basis for the set of all solutions of the recurrence relation.

That is, any solution \(s\) can be written as

\[(2.14.1)\]
\[
s = c_1 s_1 + c_2 s_2 + \ldots + c_p s_p, \text{ for some constants } c_1, c_2, \ldots, c_p.
\]
Furthermore, these constants are unique. []

Given a basis, the preceding theorem indicates that any solution to the recurrence relation must be a linear combination of the functions in the basis.

However, this does not say anything about the uniqueness of a basis for a given recurrence relation. In the next section, the Birkhoff-Trijitzinsky theorem will discuss the asymptotic uniqueness of solutions for Poincare recurrence relations.
In the next theorem (2.15), the \( p \) characteristic values of a constant coefficient recurrence relation will be used to provide a basis for the recurrence relation for all \( n \geq 0 \). Notice that the \( p \) characteristic values for a recurrence relation need not be distinct. The problem of multiple roots is addressed by the use of the multiplicities of the roots.

**Theorem 2.15**

For a constant coefficient recurrence relation (2.3.1), if the characteristic values are expressed as the unique roots

\[
\begin{align*}
& r_1 < r_2 < \ldots < r_{L'}, \\
& \text{with multiplicities } N_1, N_2, \ldots, N_{L'},
\end{align*}
\]

then \( p \) linearly independent solutions for the recurrence relation are given by:

\[
\begin{align*}
& n^j r_i^n, \quad 0 \leq j < N_i, \quad 1 \leq i \leq L. \quad [\]
\end{align*}
\]

Notice that, by theorem 2.14, equation 2.15.2 forms a basis for the recurrence relation.

**Example 2.16**

For this example two cases will be shown. The first case will be from example 2.12.1, where \( L = p \).

\[
\begin{align*}
& \text{Given, } r_1 = 1, \quad r_2 = 2, \quad N_1 = 1, \quad \text{and } N_2 = 1,
\end{align*}
\]
the linearly independent solutions for recurrence relation 2.2.1 are given by

\[ (2.16.2) \]
\[ 1^n \text{ and } 2^n. \]

For the second case, example 2.10.2 with \( L < p \) will be examined.

\[ (2.16.3) \]
Given, \( r_1 = 2/3, r_2 = 4/3, r_3 = 7/3, N_1 = 2, N_2 = 3 \)
and \( N_3 = 2, \)
the linearly independent solutions for recurrence relation 2.10.2 are given by:

\[ (2.16.4) \]
\[ (2/3)^n, n(2/3)^n, (4/3)^n, n(4/3)^n, n^2(4/3)^n, (7/4)^n, \]
and \( n(7/4)^n. \]

Poincare's, Perron's, and Birkhoff-Trjitzinsky Theorems

The following theorem (2.17), known as Poincare's theorem, shows how the characteristic values and the solutions to Poincare type recurrence relations are related.

**Theorem 2.17**

For any solution \( y(n) \) of a Poincare type homogeneous recurrence relation (2.7.1)
\[
\lim_{n \to \infty} \frac{y(n+1)}{y(n)} = r_i,
\]

where, \( r_i \) is one of the roots of the characteristic equation. [ ]

Norlund (1954) should be consulted for a proof of theorem 2.17.

To examine Poincare's theorem for the class of constant coefficient recurrence relations, we see that any solution of the recurrence relation (2.7.1) must be able to be written as a linear combination of the basis given in 2.15.2. Elementary calculus then shows that 2.17 holds.

The next theorem (2.18), known as Perron's theorem is a little more powerful than Poincare's. Perron's theorem shows a relationship of the limits of the roots to the multiplicities for those roots.

**Theorem 2.18**

There is a basis:

\[
(2.18.1)
\]

\[\{ y_i : \text{for } i = 1, 2, \ldots, p \}, \]

for any Poincare type homogeneous recurrence relation (2.7.1) with the property that for each \( i = 1, 2, \ldots, p \),
(2.18.2) \[
\lim_{n \to \infty} \frac{y_i(n + 1)}{y_i(n)} = r_j, \text{ for some } j.
\]

Furthermore, the number of solutions in the basis that have \( r_j \) as the limit of the ratio of consecutive terms is equal to \( N_j \), where \( N_j \) is the multiplicity of the root \( r_j \). \[\]

A formal proof of theorem 2.18 may be found in the work of Perron (1954).

To consider this theorem in the constant coefficient case, we note that either basis in example 2.16 satisfies the theorem.

There is a rather deep theorem about the nature of a basis for a recurrence relation. The statement of this theorem, known as the Birkhoff-Trjitzinsky theorem, is outside the scope of this work. However, to understand why we will be using constant coefficient recurrence relations as our examples, one must appreciate the connection between the constant coefficient recurrence relations and the general Poincare recurrence relations.

The Birkhoff-Trjitzinsky theorem (Birkhoff and Trjitzinsky 1932) asserts that every Poincare system has a basis and each member of this basis is asymptotic to a function of the form \( r^n q(n) \), where the behavior of \( q(n) \) is close to that of \( n^j \), in most cases only.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
differing by lower order log terms, and \( r \) is a root of the characteristic equation. (compare this to the basis in 2.15.2)

Although the initial behavior of a basis element for a Poincare system may bear no resemblance to its asymptotic behavior, it eventually will. The sooner the limits in 2.7 are reached, the sooner this limiting behavior will be observed. For most equations in applications, this limiting behavior is reached relatively soon.

Thus, although Poincare systems pose some additional computational problems, we feel justified in saying that an initial investigation centered on constant coefficient examples is appropriate.

Solutions of Nonhomogeneous Recurrence Relations and the Concept of Dominance

The following definition (2.19) and theorem (2.20) provide some insight into the solution of nonhomogeneous recurrence relations.

Definition 2.19

For the recurrence relation 2.1.1, a set of complimentary solutions is any basis of the homogeneous recurrence relation corresponding to 2.1.1, which is obtained by setting \( f(n) = 0 \), for \( n = 0, 1, \ldots \).
Theorem 2.20

If \( y_p \) is some fixed particular solution of 2.1.1, and \( s_1, s_2, \ldots, s_p \) are complimentary solutions, then any solution \( Y \) of 2.1.1 may be written as

\[
Y = c_1s_1 + c_2s_2 + \ldots + c_ps_p + y_p,
\]

for some unique constants \( c_1, c_2, \ldots, c_p \).

The concept of dominance is very important in the analysis of error propagation (Mattheij 1980). The need for knowing if one solution dominates another solution will become apparent in later chapters. For now the definition of dominance and the introduction of the minimal, intermediate, and dominant solutions for a recurrence relation will be given.

Definition 2.21

A function \( x(n) \) is said to dominate a function \( y(n) \) if

\[
\frac{y(n)}{x(n)} = o(1), \text{ as } n \text{ goes to infinity.}
\]

Definition 2.22

Suppose that \( s_1, s_2, \ldots, s_p \) form a basis for the homogeneous recurrence relation (2.5.1), if \( s_1 \) is dominated by \( s_2, s_3, \ldots, s_p \), then \( s_1 \) is called a minimal solution of the recurrence relation.
If $s_p$ dominates $s_1, s_2, \ldots, s_{p-1}$, then $s_p$ is called a dominate solution of the recurrence relation.

Any solution of a recurrence relation other than a dominate or minimal solution is called an intermediate solution of the recurrence relation. []

**Example 2.23**

For the set of solutions $s_1, s_2, \ldots, s_7$, of recurrence relation (2.14.2), $s_1$ would be a minimal solution, $s_7$ would be a dominate solution, and $s_2, s_3, \ldots, s_6$, would be intermediate solutions. []

The existence of minimal and dominate solutions for a particular recurrence relation is not guaranteed. This is due to the condition placed on dominance by $o(1)$ in definition 2.21.1. It will be shown that recurrence relations that do not possess minimal and dominate solutions will cause the most problems in the calculation of its solutions.

**Example 2.24**

For the solution set of example 2.10.1 there is no minimal or dominate solution due to the fact that as $n$ goes to infinity both solutions will approach 1 as a limit. []

Throughout the literature the definition of dominance of solution has been found to be inconsistent. Several subtle variations have been

Properties of Initial and Terminal Conditions and Their Effect on Recurrence Relations

For the solution of a recurrence relation (2.7.1), the inclusion of some additional information is important. This information can be of many types. One type is that p values for the solution are known. The definitions of the p values as initial, intermediate, or terminal conditions is given next.

Definition 2.25

For the computations of the first n+1 values of a solution Y, (y(0), y(1), y(2), ..., y(n)), of a recurrence relation 2.1.1:

(2.25.1)

\[ y(0), y(1), ..., y(i-1), \text{ are called the } i \text{ initial conditions,} \]

(2.25.2)

\[ \text{for } n-j+1 >= i, \ y(n-j+1), y(n-j+2), ..., y(n) \text{ are called the } j \text{ terminal conditions.} \]

(2.25.3)

Any values other than the initial or terminal conditions are called intermediate conditions. [ ]
The following classification of recurrence relations, with initial and terminal conditions, will be used to separate the types of solution methods presented in later chapters.

Definition 2.26

Given a pth order recurrence relation with i initial conditions and j terminal conditions (from 2.25), the following terms apply.

(2.26.1)

If $i + j = p$ then the recurrence relation is said to be determined.

(2.26.2)

If $i + j < p$ then the recurrence relation is said to be underdetermined.

(2.26.3)

If $i + j > p$ then the recurrence relation is said to be overdetermined. []

Definitions 2.25 and 2.26 provide some terms useful in talking about the uniqueness of solutions and the stability of solution methods to recurrence relations. The stability of a recurrence relations will be covered in more detail in chapter 3, but it should be noted that the selection of $i$ and $j$ (2.26) will be important.

For a determined or overdetermined system (2.26.1, 2.26.3), the uniqueness of the solution is guaranteed, but in the case of the underdetermined system (2.26.2),
(2.26.2), the uniqueness of the solution is not guaranteed; in fact the solution will have a basis of size \( p - (i+j) \).

The Associated Linear System of Equations for a Recurrence Relation

Once the type of a linear recurrence relation is known, a method for solving the recurrence relation may be considered.

Several methods are based on the use of linear systems of equations. If a linear system can be associated with a recurrence relation, numerical linear algebra techniques can be used to obtain the solution.

Basic Matrices and Vectors

There are several different linear systems that can be associated with a given recurrence relation. The following notation (2.27) will allow the construction of the basic matrices and vectors needed to form the associated linear systems.

Notation 2.27

The recurrence relation 2.1.1 for the values \( y(0), y(1), \ldots, y(N) \) can be written as a linear system.

(2.27.1)

\[
Ay = b,
\]

where, \( A \) is the \( N-p \) by \( N \) matrix below with \( k = N-p \):
(2.27.2)
\[
\begin{bmatrix}
ap(0) & a_{p-1}(0) & \cdots & a_2(0) & a_1(0) & 1 & 0 & \ldots & 0 & 0 \\
0 & a_p(1) & a_{p-1}(1) & \cdots & a_2(1) & a_1(1) & 1 & 0 & \ldots & 0 \\
\vdots & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \ldots & 0 & a_p(k-1) & a_{p-1}(k-1) & \cdots & a_2(k-1) & a_1(k-1) & 1 & 0 \\
0 & \ldots & 0 & a_p(k) & a_{p-1}(k) & \cdots & a_2(k) & a_1(k) & 1 & 0
\end{bmatrix}
\]

(2.27.3)
y is the result vector \([y(0), y(1), \ldots, y(N)]^T\),

(2.27.4)
and \(b\) is the vector \([f(0), f(1), \ldots, f(N-p)]^T\). []

The following example (2.28) will make use of the matrix notation provided in 2.27 to construct the matrix \(A\) and the vectors \(y\) and \(b\) for a recurrence relation.

Example 2.28

The first example will be a linear system representation for the recurrence relation defined by example 2.2.1, with \(N = 5\).

(2.28.1)
\[
A = \begin{bmatrix}
2 & -3 & 1 & 0 & 0 \\
0 & 2 & -3 & 1 & 0 \\
0 & 0 & 2 & -3 & 1
\end{bmatrix},
\]

(2.28.2)
\[
y = [y(0), y(1), y(2), y(3), y(4)]^T, \text{ and}
\]
(2.28.3)
\[ b = [0, 0, 0]^T. \]

The second example will be from example 2.2.2 with \( x = 2 \), and \( N = 5 \).

(2.28.4)
\[
A = \begin{bmatrix}
-1 & 1 & 1 & 0 & 0 \\
0 & -1 & 2 & 1 & 0 \\
0 & 0 & -1 & 3 & 1
\end{bmatrix},
\]

(2.28.5)
\[ y = [y(0), y(1), y(2), y(3), y(4)]^T, \]

(2.28.6)
\[ b = [0, 0, 0]^T. \]

Example 2.28 provided the basic matrices for two simple recurrence relations. Chapter four will present several more complex examples.

The Augmented and Reduced Forms for the Associated Linear System

For the solution of a recurrence relation (2.7.1), the next definition (2.29) and notation (2.30) will provide the basis for a linear algebraic approach.

Notation for the \( k \) by \( k \) identity matrix and the \( k \) by \( m \) zero matrix will be provided and used in the construction of two linear systems: an augmented system.
and a reduced system. Each of these systems provides the option for placing initial and terminal conditions on the recurrence relation.

Later chapters will show how well each of these systems works and some of the problems that may occur when numerical solutions are computed.

**Notation 2.29**

Let \( I_k \) denote the \( k \) by \( k \) identity matrix, and let \( O_{km} \) denote the \( k \) by \( m \) zero matrix. []

**Notation 2.30**

Given the \( N-p \) by \( N \) matrix \( A \) of equation 2.27.1, the \( i \) initial conditions \( y(k) = c_k, k = 0, 1, \ldots, i-1 \), and the \( j \) terminal conditions \( y(k) = d_{N-k}, k = N-j+1, N-j+2, \ldots, N \) (from 2.25), two notations for the representation of associated linear systems for a recurrence relation, are now given.

The first notation is called the augmented system.

\[(2.30.1)\]

\[
A_a y_a = b_a
\]

where,

\[(2.30.2)\]

\[
A_a = \begin{bmatrix}
I_1 & & & O_{i,N-1} \\
& \ddots & & \vdots \\
& & \ddots & O_{j,N-j} \\
& & & I_j
\end{bmatrix}
\]
(2.30.3)
\[ y_a = [y(0), y(1), \ldots, y(N-p+i+j)]^T, \text{ and} \]

(2.30.4)
\[ b_a = [c_0, c_1, \ldots, c_{i-1}, f(i), f(i+1), \ldots, f(N-p+i), d_0, d_1, \ldots, d_{j-1}]^T. \]

The second notation is called the reduced form of the associated linear system of equations.

(2.30.5)
\[ A_r y_r = b_r \]

The \( A_r \) matrix and its vectors are constructed in the following manner.

For the \( i \) initial conditions, remove the first \( i \) columns of \( A \) and adjust the values of \( f(0), f(1), \ldots, f(i-1) \) to reflect the initial values being excluded.

For the \( j \) terminal conditions, remove the last \( j \) columns of \( A \) and adjust the values of \( f(N-p-j+1), f(N-p-j+2), \ldots, f(N-p) \) to reflect the terminal values that have been excluded. The \( y_r \) and \( b_r \) vectors are given by

(2.30.6)
\[ y_r = [y(i), y(i+1), \ldots, y(N-p-i-j)]^T, \text{ and} \]

(2.30.7)
\[ b_r = [f_a(0), f_a(1), \ldots, f_a(i-1), f(i), f(i+1), \ldots, f_a(N-p-j), f_a(N-p-j+1), \ldots, f_a(N-p)]^T \]

with the \( a \) subscripts showing adjusted values. [ ]
The next examples (2.31, 2.32) will use the notations introduced in 2.30 to construct an associated linear system of equations for a recurrence relation. The examples presented here will be simple; whereas, more complex examples can be found in chapter four.

Example 2.31

Given the matrix and vectors from example 2.28.1, and the two initial conditions \( c_0 = 1 \) and \( c_1 = 2 \), then the augmented form of the matrix and vectors for recurrence relation 2.2.1 would be:

\[
\begin{align*}
A_a &= \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
2 & -3 & 1 & 0 & 0 \\
0 & 2 & -3 & 1 & 0 \\
0 & 0 & 2 & -3 & 1 \\
\end{bmatrix}, \\
y_a &= [y(0), y(1), y(2), y(3), y(4)]^T, \text{ and} \\
b_a &= [1, 2, 0, 0, 0]^T.
\end{align*}
\]

Example 2.32

Given the matrix and vectors from example 2.28.1, and the two initial conditions \( c_0 = 1 \) and \( c_1 = 2 \), then the reduced form of the matrix and vectors for recurrence relation 2.2.1 would be:
The reduced system associated with a recurrence relation is the standard form presented in the literature. The augmented system has not been considered in the solution of recurrence relations insofar as the references for this work have shown. It is easy to find examples where the calculations for the reduced system are unstable, and the calculations for the augmented system are stable.

In chapter four the comparison of the augmented and reduced form of the associated linear systems will be presented. It will be shown where the choice of a particular representation for all problems is not feasible. The software presented will allow the comparison and solution to a problem using several different methods and representations.
The Introduction to Computational Errors and Error Comparison Notations

The introduction of errors in computer computations has been a problem since the invention of the computer. The elimination of errors is impossible due to the nature of computers having fixed length representations for floating point numbers; thus, the inability to represent most numbers exactly. The best one can hope for is to minimize errors and reduce the propagation of these errors from one computation to another. The following discussion will give some of the terms and notations that will be used throughout succeeding chapters.

The inability of a computer to store a result exactly is called roundoff error. This error may be very small, but after a few computations, even using exact arithmetic, the error may become significant. The next chapters will contain several stark examples of this phenomenon.

The next definition provides a basis for comparing approximations. The absolute error is the difference of an approximation from the exact value; while, the relative error is an estimate of the number of significant digits in an approximation.
Definition 2.33

Let an exact value for the solution of a recurrence relation be denoted by \( w(n) \) and the computed value of \( w(n) \) be \( w_c(n) \) then the absolute error is given by

\[
e_a(n) = | w(n) - w_c(n) | ,
\]

and the relative error is given by

\[
e_r(n) = \left| \frac{e_a}{w(n)} \right| . \]

In the solution of recurrence relations, the result is a sequence of numbers \( W(n), n = 0, 1, \ldots, N \). The next definition will provide a notation for errors of the complete sequence \( w(0), w(1), \ldots, w(N) \).

Definition 2.34

For a recurrence relation with the exact results \( w(0), w(1), \ldots, w(N) \) and computed results \( w_c(0), w_c(1), \ldots, w_c(N) \), the errors will be denoted as either,

\[
e_a(0), e_a(1), \ldots, e_a(N),
\]

for the absolute errors, or

\[
e_r(0), e_r(1), \ldots, e_r(N),
\]

for the relative errors.
Forsythe, Malcolm, and Moler (1977) provides a pleasant introduction to roundoff errors in the solution of recurrence relations: while the classical work associated to linear systems may be found in Wilkinson (1960, 1963).
CHAPTER III

GENERAL SOLUTIONS OF RECURRENCE RELATIONS

In this chapter the standard methods for solving recurrence relations will be discussed. Only the determined class of recurrence relations will be used in this chapter (see 2.26). The overdetermined recurrence relations will be pursued in chapter five.

The general methods for solving recurrence relations include forward and backward recurrence. These methods will be shown to work in certain situations, but not in general.

The Miller and Oliver algorithms will provide a more general method for the solution of recurrence relations.

Given along with the different methods for the solution of recurrence relations will be a discussion of error propagation. Specific examples will be provided to illustrate all of the methods and typical errors involved with each method.

The concept of stability of computations for a recurrence relation will be presented. The stability of a solution method can provide meaningful information about the reliability of the computed results. The
literature reviewed for this work revealed that the definition of stability varied greatly from one author to another (Cash 1981; Gautschi 1967, 1973; Lozier 1980; Oliver 1967, 1968a, 1968b; Olver 1967a; Wimp 1984).

As a computer aid to the solution of recurrence relations, a set of FORTRAN subroutines has been implemented. These subroutines will be described in detail in chapter six. The results presented for some of the examples in this chapter utilized these routines. The examples presented here were selected to provide typical results without showing special cases. Chapter four contain more typical examples.

Simple Methods for the Solution of Recurrence Relations

The general definitions for forward and backward recurrence given next are usually effective for calculating some, but not all, solutions of a recurrence relation.

Forward Recurrence

Definition 3.1

If $y(0), y(1), \ldots, y(p-1)$ are given, then rewrite recurrence relation 2.1.1 in the following form
(3.1.1)

\[ y(n) = -a_1y(n-1) - a_2y(n-2) - \ldots - a_py(n-p) + f(n), \]

for \( n = p, p+1, \ldots \), in that order.

If equation 3.1.1 is used to calculate \( y(n) \), for \( n = p, p+1, \ldots \), in that order, then this is called forward recurrence.

For the solution of a recurrence relation of order \( p \) using forward recurrence, the inclusion of \( p \) initial conditions is required. The next example will show forward recurrence for a second order recurrence relation with two initial conditions.

Example 3.2

If example 2.2.1 is written as

(3.2.1)

\[ y(n) = 3y(n-1) - 2y(n-2), \text{ and } y(0) = 1, y(1) = 2. \]

Recurrence relation 3.2.1 will calculate \( y(n) \), for \( n = 2, 3, \ldots \), in the forward direction.

If calculated the next values would be \( y(2) = 4, y(3) = 8, \) and \( y(4) = 16. \]

Backward Recurrence

Definition 3.3

If \( y(N), y(N-1), \ldots, y(N-p+1) \) are given then rewrite recurrence relation 2.1.1 in the following form
(3.3.1)

\[ y(n) = \frac{a_{p-1}y(n+1) + a_{p-2}y(n+2) + \ldots + a_{p}y(n+p)}{a_{p}} + f(n), \]

for \( n = N-p, N-p-1, \ldots, 0 \), in that order.

If equation 3.3.1 is used to calculate \( y(n) \), for \( n = N-p, N-p-1, \ldots, 0 \), in that order, then this is called backward recurrence. []

For the solution of a recurrence relation of order \( p \) using backward recurrence, the inclusion of \( p \) terminal conditions is required. The next example will show backward recurrence for a second order recurrence relation and two terminal conditions.

Example 3.4

If equation 2.2.1 is written as

(3.4.1)

\[ y(n) = \frac{3}{2}y(n+1) - \frac{1}{2}y(n+2), \]

and \( y(5) = 32, y(6) = 64 \).

Recurrence relation 3.4.1 will calculate \( y(n) \), for \( n = 4, 3, \ldots, 0 \), in the backward direction.

If calculated the next values would be \( y(4) = 16, y(3) = 8, y(2) = 4, y(1) = 2, \) and \( y(0) = 1 \). []
The terms forward recurrence and recurrence in the forward direction will have the same meaning; likewise, the terms backward recurrence and recurrence in the backward direction will have the same meaning.

Boundary Value Problems

If the recurrence relation is determined, but the given conditions are not all initial or terminal conditions, then the recurrence relation is said to be a boundary value problem.

**Definition 3.5**

If $i$ initial conditions and $j$ terminal conditions for a recurrence relation (2.7.1) are given, $i+j = p$, and $i,j <> 0$, then this is called a boundary value problem.

Notice that for the boundary value problem no indication for direction of solution is given. Later, methods for the solutions of boundary value problems will be discussed.

In the literature when boundary problems are presented the reduced system of equations approach (2.30) is used (Oliver 1968a). The use of the augmented system of equations, which will be advocated later, is not part of the general literature.
Conditions for Stability of Solutions to Recurrence Relations

The following definitions will give some insight into the computational stability of a recurrence relation. The more stable the recurrence relation is the more dependable the results will be. Two types of stability will be presented: absolute and relative. Relative stability will be concerned with relative errors, while absolute stability will be concerned with absolute errors.

Stability in the Absolute Sense for Forward Recurrence

The next definition (3.6) provides the needed conditions for a recurrence relation to be stable in the absolute sense in the forward direction.

Definition 3.6

For a Poincare recurrence relation (2.7.1), if the norms of all of the characteristic values are $< 1$, then the recurrence relation is said to be stable in the absolute sense in the forward direction. []

In the next example (3.7) a second order recurrence relation with two roots less than 1, is shown to give good results for both roots.

Example 3.7

For this example the following second order recurrence relation will be used:
\( (3.7.1) \)

\[ y(n) = y(n-1) - \frac{2}{9}y(n-2), \]

with the two solutions \( s_1 = \frac{1}{3}^n \), \( s_2 = \frac{2}{3}^n \),

the initial conditions \( y(0) = 1, y(1) = \frac{1}{3} \) for \( s_1 \), and
\( y(0) = 1, y(1) = \frac{2}{3} \) for \( s_2 \).

If we solve equation 3.7.1 for the two solutions in
the forward direction, tables 3.7.2 and 3.7.3 are the
result. []

Definition 3.8 provides some insight into how
errors propagate through calculations. The idea that a
recurrence relation is stable depends on the behavior of
errors through repeated calculations.

The fact that the norms of all the eigenvalues of
the characteristic equation are less than 1, for an
absolutely stable recurrence relation, can be used to
easily show that the calculation of a dominate solution,
in the forward direction, will have both low absolute
and relative errors; while the calculation of a
non-dominate solution should have low absolute errors,
but may have high relative errors.

Almost every author has a different definition of
stability. These definitions are usually selected to
insure good relative accuracy, at least in theory.
One major drawback to most conditions of this type is
that they are difficult to verify in practice.
### Table 3.7.2
Calculation of Solution $s_1$ of Equation 3.7.1 Using Forward Recurrence

<table>
<thead>
<tr>
<th>$n$</th>
<th>$w(n)$</th>
<th>$w_c(n)$</th>
<th>$e_a(n)$</th>
<th>$e_e(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1000000E+01</td>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.3333333E+00</td>
<td>0.3333333E+00</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.1111111E+00</td>
<td>0.1111111E+00</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.3707036E-01</td>
<td>0.3707036E-01</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.1234567E-01</td>
<td>0.1234567E-01</td>
<td>0.1164153E-09</td>
<td>0.9429643E-08</td>
</tr>
<tr>
<td>5</td>
<td>0.4115226E-02</td>
<td>0.4115226E-02</td>
<td>0.1164153E-09</td>
<td>0.2828892E-07</td>
</tr>
<tr>
<td>6</td>
<td>0.1371742E-02</td>
<td>0.1371742E-02</td>
<td>0.1018654E-09</td>
<td>0.7425435E-07</td>
</tr>
<tr>
<td>7</td>
<td>0.4572473E-03</td>
<td>0.4572473E-03</td>
<td>0.7697555E-10</td>
<td>0.1670815E-06</td>
</tr>
<tr>
<td>8</td>
<td>0.1524157E-03</td>
<td>0.1524157E-03</td>
<td>0.5638675E-10</td>
<td>0.3599645E-06</td>
</tr>
<tr>
<td>9</td>
<td>0.5080522E-04</td>
<td>0.5080522E-04</td>
<td>0.3910271E-10</td>
<td>0.7697681E-06</td>
</tr>
<tr>
<td>10</td>
<td>0.1693509E-04</td>
<td>0.1693509E-04</td>
<td>0.2683094E-10</td>
<td>0.1534203E-05</td>
</tr>
<tr>
<td>11</td>
<td>0.5645029E-05</td>
<td>0.5645029E-05</td>
<td>0.1801936E-10</td>
<td>0.3192076E-05</td>
</tr>
<tr>
<td>12</td>
<td>0.1816763E-05</td>
<td>0.1816763E-05</td>
<td>0.1207925E-10</td>
<td>0.6419906E-05</td>
</tr>
<tr>
<td>13</td>
<td>0.6272254E-06</td>
<td>0.6272254E-06</td>
<td>0.8078709E-11</td>
<td>0.1289033E-04</td>
</tr>
<tr>
<td>14</td>
<td>0.2090751E-06</td>
<td>0.2090751E-06</td>
<td>0.5168348E-11</td>
<td>0.2652013E-04</td>
</tr>
<tr>
<td>15</td>
<td>0.6991713E-07</td>
<td>0.6991713E-07</td>
<td>0.3015635E-11</td>
<td>0.5168504E-04</td>
</tr>
</tbody>
</table>
Table 3.7.3
Calculation of Solution $s_2$ of Equation 3.7.1
Using Forward Recurrence

<table>
<thead>
<tr>
<th>$n$</th>
<th>$w(n)$</th>
<th>$w_c(n)$</th>
<th>$e_a(n)$</th>
<th>$e_c(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.66666666E+00</td>
<td>0.66666666E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.44444444E+00</td>
<td>0.44444444E+00</td>
<td>0.37259303E-08</td>
<td>0.83819033E-08</td>
</tr>
<tr>
<td>3</td>
<td>0.29629629E+00</td>
<td>0.29629629E+00</td>
<td>0.74505806E-08</td>
<td>0.25145710E-07</td>
</tr>
<tr>
<td>4</td>
<td>0.19753088E+00</td>
<td>0.19753088E+00</td>
<td>0.74505806E-08</td>
<td>0.37718565E-07</td>
</tr>
<tr>
<td>5</td>
<td>0.13168725E+00</td>
<td>0.13168725E+00</td>
<td>0.74505806E-08</td>
<td>0.56577848E-07</td>
</tr>
<tr>
<td>6</td>
<td>0.87791493E-01</td>
<td>0.87791493E-01</td>
<td>0.87979354E-08</td>
<td>0.63650758E-07</td>
</tr>
<tr>
<td>7</td>
<td>0.58527666E-01</td>
<td>0.58527666E-01</td>
<td>0.46566129E-08</td>
<td>0.79562603E-07</td>
</tr>
<tr>
<td>8</td>
<td>0.39038444E-01</td>
<td>0.39038444E-01</td>
<td>0.37259303E-08</td>
<td>0.95475120E-07</td>
</tr>
<tr>
<td>9</td>
<td>0.26012296E-01</td>
<td>0.26012296E-01</td>
<td>0.27939677E-08</td>
<td>0.10740951E-06</td>
</tr>
<tr>
<td>10</td>
<td>0.17341528E-01</td>
<td>0.17341528E-01</td>
<td>0.20954753E-08</td>
<td>0.12035760E-06</td>
</tr>
<tr>
<td>11</td>
<td>0.11561019E-01</td>
<td>0.11561019E-01</td>
<td>0.14097499E-08</td>
<td>0.14097499E-06</td>
</tr>
<tr>
<td>12</td>
<td>0.77073473E-02</td>
<td>0.77073473E-02</td>
<td>0.11641532E-08</td>
<td>0.15104463E-06</td>
</tr>
<tr>
<td>13</td>
<td>0.51382307E-02</td>
<td>0.51382307E-02</td>
<td>0.87311491E-09</td>
<td>0.16925218E-06</td>
</tr>
<tr>
<td>14</td>
<td>0.34254877E-02</td>
<td>0.34254877E-02</td>
<td>0.61113044E-09</td>
<td>0.17842147E-06</td>
</tr>
<tr>
<td>15</td>
<td>0.22836568E-02</td>
<td>0.22836568E-02</td>
<td>0.46566129E-09</td>
<td>0.20391025E-06</td>
</tr>
</tbody>
</table>
The importance of the definition of stability in the relative sense has generated a large amount of literature (Olver 1964; Oliver 1967, 1968b; Tait 1967; Wilkinson 1965a, 1968, 1971; Businger 1971; Reid 1971a; Cash 1981; Wimp 1984). The following definition (3.8) could be considered a typical definition.

**Definition 3.8**

The homogeneous recurrence relation (2.5.1) may be rewritten as the set of matrix equations:

(3.8.1)

\[ y(n) + A(n)y(n+1) = 0, \]

where \( A(n) \) is a \( p \) by \( p \) matrix and \( y(n) \) is a \( p \) vector.

Suppose we wish to calculate a solution to the recurrence relation (3.8.1) \( w(k) \), and that a fundamental matrix of size \( p \) by \( p \) is \( Y(k) \).

(3.8.2)

Let

\[ a(k,n) = \frac{|| w(k) ||}{|| w(n) ||} \cdot \frac{|| Y(n) Y^{-1}(k) ||}{|| Y(n) \ Y^{-1}(k) ||} , n > k \]

If \( \text{Sup} \ a(k,n) = c < \infty; n, k \geq 0, n > k \), then the recurrence relation is said to be stable in the relative sense in the forward direction. []
Stability in the Absolute Sense for Backward Recurrence

The next definition (3.9) provides the needed conditions for a recurrence relation to be stable in the absolute sense in the backward direction.

**Definition 3.9**

For a Poincare recurrence relation (2.7.1). If the norms of all the characteristic values are > 1, then the recurrence relation is said to be stable in the absolute sense in the backward direction. []

In the next example (3.10) a second order recurrence relation with two roots greater than 1 is shown to give good results for both roots.

**Example 3.10**

For this example the following second order recurrence relation will be used:

(3.10.1)

\[ y(n) = 5.1y(n-1) - 6.3y(n-2), \]

with the two solutions \( s_1 = 2.1^n, \) \( s_2 = 3.0^n \) the initial conditions \( y(10) = 2.1^{10}, y(9) = 2.1^9 \) for \( s_1 \) and \( y(10) = 3^{10}, y(9) = 3^9 \) for \( s_2 \).

If we solve 3.10.1 for the two solutions in the backward direction, tables 3.10.2 and 3.10.3 are the result. []
### Table 3.10.2

Calculation of Solution $s_{1}$ of Equation 3.10.1
Using Backward Recurrence

<table>
<thead>
<tr>
<th>n</th>
<th>$w(n)$</th>
<th>$w_{c}(n)$</th>
<th>$e_{a}(n)$</th>
<th>$e_{1}(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0.68122313E+05</td>
<td>0.68122313E+05</td>
<td>0.48828125E-02</td>
<td>0.71677139E-07</td>
</tr>
<tr>
<td>14</td>
<td>0.32439199E+05</td>
<td>0.32439199E+05</td>
<td>0.21972566E-02</td>
<td>0.67734898E-07</td>
</tr>
<tr>
<td>13</td>
<td>0.15447235E+05</td>
<td>0.15447235E+05</td>
<td>0.97652040E-03</td>
<td>0.63192259E-07</td>
</tr>
<tr>
<td>12</td>
<td>0.73558270E+04</td>
<td>0.73558270E+04</td>
<td>0.61035156E-03</td>
<td>0.82975247E-07</td>
</tr>
<tr>
<td>11</td>
<td>0.35027745E+04</td>
<td>0.35027745E+04</td>
<td>0.33593369E-03</td>
<td>0.95836409E-07</td>
</tr>
<tr>
<td>10</td>
<td>0.16679888E+04</td>
<td>0.16679888E+04</td>
<td>0.19836429E-03</td>
<td>0.11802427E-06</td>
</tr>
<tr>
<td>9</td>
<td>0.79428001E+03</td>
<td>0.79428001E+03</td>
<td>0.10631125E-03</td>
<td>0.13447518E-06</td>
</tr>
<tr>
<td>8</td>
<td>0.37822858E+03</td>
<td>0.37822858E+03</td>
<td>0.53405762E-04</td>
<td>0.14119970E-06</td>
</tr>
<tr>
<td>7</td>
<td>0.18010685E+03</td>
<td>0.18010685E+03</td>
<td>0.24755329E-04</td>
<td>0.13766971E-06</td>
</tr>
<tr>
<td>6</td>
<td>0.85766119E+02</td>
<td>0.85766119E+02</td>
<td>0.12397764E-04</td>
<td>0.14458319E-06</td>
</tr>
<tr>
<td>5</td>
<td>0.40841005E+02</td>
<td>0.40841005E+02</td>
<td>0.66757202E-05</td>
<td>0.16345303E-06</td>
</tr>
<tr>
<td>4</td>
<td>0.19448100E+02</td>
<td>0.19448100E+02</td>
<td>0.35762767E-05</td>
<td>0.18388834E-06</td>
</tr>
<tr>
<td>3</td>
<td>0.92699998E+01</td>
<td>0.92699998E+01</td>
<td>0.19073486E-05</td>
<td>0.20654536E-06</td>
</tr>
<tr>
<td>2</td>
<td>0.44100000E+01</td>
<td>0.44100000E+01</td>
<td>0.10728936E-05</td>
<td>0.24328427E-06</td>
</tr>
<tr>
<td>1</td>
<td>0.21000000E+01</td>
<td>0.21000000E+01</td>
<td>0.58604545E-06</td>
<td>0.28331848E-06</td>
</tr>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.31292439E-06</td>
<td>0.31292439E-06</td>
</tr>
</tbody>
</table>
Table 3.10.3
Calculation of Solution $s_2$ of Equation 3.10.1
Using Backward Recurrence

<table>
<thead>
<tr>
<th>$n$</th>
<th>$w(n)$</th>
<th>$V_n(n)$</th>
<th>$e_a(n)$</th>
<th>$e_2(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0.14348907E+08</td>
<td>0.14348907E+08</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>14</td>
<td>0.478296E+07</td>
<td>0.478296E+07</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.159432E+07</td>
<td>0.159432E+07</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>12</td>
<td>0.531410E+06</td>
<td>0.531410E+06</td>
<td>0.781250E-02</td>
<td>0.14700597E-07</td>
</tr>
<tr>
<td>11</td>
<td>0.177147E+06</td>
<td>0.177147E+06</td>
<td>0.585937E-02</td>
<td>0.33076343E-07</td>
</tr>
<tr>
<td>10</td>
<td>0.590490E+05</td>
<td>0.590490E+05</td>
<td>0.390625E-02</td>
<td>0.661562E-07</td>
</tr>
<tr>
<td>9</td>
<td>0.196300E+05</td>
<td>0.196300E+05</td>
<td>0.170353E-02</td>
<td>0.888250E-07</td>
</tr>
<tr>
<td>8</td>
<td>0.656100E+04</td>
<td>0.656100E+04</td>
<td>0.610351E-03</td>
<td>0.930272E-07</td>
</tr>
<tr>
<td>7</td>
<td>0.218700E+04</td>
<td>0.218700E+04</td>
<td>0.152587E-03</td>
<td>0.697741E-07</td>
</tr>
<tr>
<td>6</td>
<td>0.729000E+03</td>
<td>0.729999E+03</td>
<td>0.152587E-03</td>
<td>0.209311E-07</td>
</tr>
<tr>
<td>5</td>
<td>0.243000E+03</td>
<td>0.243000E+03</td>
<td>0.114449E-04</td>
<td>0.470503E-07</td>
</tr>
<tr>
<td>4</td>
<td>0.810000E+02</td>
<td>0.810000E+02</td>
<td>0.133514E-04</td>
<td>0.164832E-06</td>
</tr>
<tr>
<td>3</td>
<td>0.270000E+02</td>
<td>0.270000E+02</td>
<td>0.953674E-05</td>
<td>0.353227E-06</td>
</tr>
<tr>
<td>2</td>
<td>0.900000E+01</td>
<td>0.900000E+01</td>
<td>0.560236E-05</td>
<td>0.622537E-06</td>
</tr>
<tr>
<td>1</td>
<td>0.300000E+01</td>
<td>0.300000E+01</td>
<td>0.298023E-05</td>
<td>0.993410E-06</td>
</tr>
<tr>
<td>0</td>
<td>0.100000E+01</td>
<td>0.100000E+01</td>
<td>0.153481E-05</td>
<td>0.153481E-05</td>
</tr>
</tbody>
</table>
The fact that the norms of all the eigenvalues of the characteristic equation are greater than 1, for an absolutely stable recurrence relation, can be used to easily show that the calculation of a minimal solution, in the backward direction, will have both low absolute and relative errors; while the calculation of a non-minimal solution should have low absolute errors, but may have high relative errors.

The justification of the above statement can be formalized in the same manner as in theorem 3.8.

Weakly Stable in Either Direction

In the next definition (3.11) the conditions of definitions 3.6 and 3.9 are relaxed to include the characteristic values equal to 1. This will have an effect on the stability and reliability of the computed solutions.

Definition 3.11

If the condition for definition 3.6 was $\leq 1$, or the condition for definition 3.9 was $\geq 1$, then the recurrence relation is said to be weakly stable for the respective direction of solution. []

In a weakly stable recurrence relation the absolute errors do not tend to diminish with subsequent calculations, nor do they tend to grow too rapidly.
In the next example (3.12), two second order recurrence relations that satisfy the two cases in definition 3.11 are examined. It will be shown that good results can be achieved for both roots, but the results will not be as good as the results for an absolutely stable recurrence relation.

Example 3.12

For this example the following second order recurrence relations will be used:

\[(3.12.1)\]

\[y(n) = \frac{4}{3}y(n-1) - \frac{1}{3}y(n-2),\]

with the two solutions \(s_1 = (\frac{1}{3})^n\) and \(s_2 = 1^n\) for forward recurrence, and

\[(3.12.2)\]

\[y(n) = \frac{3.1y(n+1)}{2.1} - \frac{y(n+2)}{2.1},\]

with the two solutions \(s_1 = 1^n\) and \(s_2 = 2.1^n\) for backward recurrence.

If we solve 3.12.1 for its two solutions in the forward direction, and solve 3.12.2 for its two solutions in the backward direction, tables 3.12.3 through 3.12.6 are the result. It is assumed that the correct initial and terminal conditions were provided for each equation. \[\]
<table>
<thead>
<tr>
<th>n</th>
<th>w(n)</th>
<th>wC(n)</th>
<th>e_a(n)</th>
<th>e_L(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.33333333E+00</td>
<td>0.33333333E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.11111111E+00</td>
<td>0.11111111E+00</td>
<td>0.18526451E-08</td>
<td>0.15763607E-07</td>
</tr>
<tr>
<td>3</td>
<td>0.37037037E-01</td>
<td>0.37037037E-01</td>
<td>0.27939677E-08</td>
<td>0.75437130E-07</td>
</tr>
<tr>
<td>4</td>
<td>0.12345679E-01</td>
<td>0.12345679E-01</td>
<td>0.33704439E-08</td>
<td>0.27345603E-06</td>
</tr>
<tr>
<td>5</td>
<td>0.41522222E-02</td>
<td>0.41522222E-02</td>
<td>0.36087508E-08</td>
<td>0.87656655E-06</td>
</tr>
<tr>
<td>6</td>
<td>0.13717421E-02</td>
<td>0.13717421E-02</td>
<td>0.37384222E-08</td>
<td>0.27234512E-05</td>
</tr>
<tr>
<td>7</td>
<td>0.45724728E-03</td>
<td>0.45724728E-03</td>
<td>0.37984263E-08</td>
<td>0.82665415E-05</td>
</tr>
<tr>
<td>8</td>
<td>0.15214151E-03</td>
<td>0.15214151E-03</td>
<td>0.38062357E-08</td>
<td>0.24930414E-04</td>
</tr>
<tr>
<td>9</td>
<td>0.50802651E-04</td>
<td>0.50802651E-04</td>
<td>0.38062357E-08</td>
<td>0.74918134E-04</td>
</tr>
<tr>
<td>10</td>
<td>0.16935087E-04</td>
<td>0.16935087E-04</td>
<td>0.38062357E-08</td>
<td>0.22488866E-03</td>
</tr>
<tr>
<td>11</td>
<td>0.56450299E-05</td>
<td>0.56450299E-05</td>
<td>0.38093617E-08</td>
<td>0.67481704E-03</td>
</tr>
<tr>
<td>12</td>
<td>0.18815763E-05</td>
<td>0.18815763E-05</td>
<td>0.38093617E-08</td>
<td>0.20246173E-02</td>
</tr>
<tr>
<td>13</td>
<td>0.62722543E-06</td>
<td>0.62722543E-06</td>
<td>0.38097551E-08</td>
<td>0.60740452E-02</td>
</tr>
<tr>
<td>14</td>
<td>0.20907514E-06</td>
<td>0.20907514E-06</td>
<td>0.38097551E-08</td>
<td>0.18223468E-01</td>
</tr>
<tr>
<td>15</td>
<td>0.69631713E-07</td>
<td>0.69631713E-07</td>
<td>0.38098547E-08</td>
<td>0.54667253E-01</td>
</tr>
</tbody>
</table>
Table 3.12.4
Calculation of Solution $s_2$ of Equation 3.12.1
Using Forward Recurrence

<table>
<thead>
<tr>
<th>$n$</th>
<th>$w(n)$</th>
<th>$v(n)$</th>
<th>$e_3(n)$</th>
<th>$e_x(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>1</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>2</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>3</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>4</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>5</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>6</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>7</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>8</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>9</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>10</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>11</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>12</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>13</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>14</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
<tr>
<td>15</td>
<td>$0.100000000E+01$</td>
<td>$0.100000000E+01$</td>
<td>$0.000000000E+00$</td>
<td>$0.000000000E+00$</td>
</tr>
</tbody>
</table>
Table 3.12.5
Calculation of Solution $s_1$ of Equation 3.12.2
Using Backward Recurrence

<table>
<thead>
<tr>
<th>n</th>
<th>$w(n)$</th>
<th>$w_c(n)$</th>
<th>$e_a(n)$</th>
<th>$e_l(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>14</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>12</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>11</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>10</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>9</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>8</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
</tbody>
</table>
Table 3.12.6
Calculation of Solution $s_2$ of Equation 3.12.2
Using Backward Recurrence

<table>
<thead>
<tr>
<th>$n$</th>
<th>$w(n)$</th>
<th>$w_C(n)$</th>
<th>$e_a(n)$</th>
<th>$e_L(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0.68122313E05</td>
<td>0.68122313E05</td>
<td>0.48828125E-02</td>
<td>0.71677139E-07</td>
</tr>
<tr>
<td>14</td>
<td>0.32439197E05</td>
<td>0.32439199E05</td>
<td>0.21972656E-02</td>
<td>0.67734996E-07</td>
</tr>
<tr>
<td>13</td>
<td>0.15447237E05</td>
<td>0.15447238E05</td>
<td>0.97656250E-03</td>
<td>0.62192320E-07</td>
</tr>
<tr>
<td>12</td>
<td>0.73555276E04</td>
<td>0.73558276E04</td>
<td>0.61035156E-03</td>
<td>0.82975247E-07</td>
</tr>
<tr>
<td>11</td>
<td>0.35027748E04</td>
<td>0.35027753E04</td>
<td>0.45776379E-03</td>
<td>0.13069811E-06</td>
</tr>
<tr>
<td>10</td>
<td>0.16679880E04</td>
<td>0.16679884E04</td>
<td>0.39672852E-03</td>
<td>0.23784954E-06</td>
</tr>
<tr>
<td>9</td>
<td>0.79429001E03</td>
<td>0.79429003E03</td>
<td>0.34322756E-03</td>
<td>0.43243936E-06</td>
</tr>
<tr>
<td>8</td>
<td>0.37822859E03</td>
<td>0.37822890E03</td>
<td>0.30517578E-03</td>
<td>0.60855438E-06</td>
</tr>
<tr>
<td>7</td>
<td>0.18010885E03</td>
<td>0.18010913E03</td>
<td>0.26612268E-03</td>
<td>0.15834966E-05</td>
</tr>
<tr>
<td>6</td>
<td>0.85766119E02</td>
<td>0.85766397E02</td>
<td>0.27847261E-03</td>
<td>0.32463708E-05</td>
</tr>
<tr>
<td>5</td>
<td>0.40841001E02</td>
<td>0.40841256E02</td>
<td>0.27561188E-03</td>
<td>0.67424006E-05</td>
</tr>
<tr>
<td>4</td>
<td>0.19448100E02</td>
<td>0.19448345E02</td>
<td>0.27418138E-03</td>
<td>0.14096106E-04</td>
</tr>
<tr>
<td>3</td>
<td>0.92609999E01</td>
<td>0.92612731E01</td>
<td>0.27322769E-03</td>
<td>0.25930944E-04</td>
</tr>
<tr>
<td>2</td>
<td>0.44100000E01</td>
<td>0.44102775E01</td>
<td>0.27275085E-03</td>
<td>0.61842666E-04</td>
</tr>
<tr>
<td>1</td>
<td>0.21000000E01</td>
<td>0.21002726E01</td>
<td>0.27254224E-03</td>
<td>0.12978202E-03</td>
</tr>
<tr>
<td>0</td>
<td>0.10000000E01</td>
<td>0.10002745E01</td>
<td>0.27242303E-03</td>
<td>0.27242303E-03</td>
</tr>
</tbody>
</table>
In example 3.12, an observation comparing the two classes of stability should be made. Although the class of weakly stable recurrence relations provides good results, it does not seem to be as good as the class of absolutely stable recurrence relations.

The next definition (3.13) will cover all of the recurrence relations that do not fit into one of the previously defined classes of stability.

**Definition 3.13**

If a recurrence relation is not stable, then it is called unstable.

The classification of recurrence relations by stability only provides a guide to the expected results of repeated calculations using the recurrence relation. This does not mean that if a recurrence relation is classified as unstable, for a particular reason, that the recurrence relation will not provide reasonable results over some restricted interval; nor does it mean that a stable recurrence relation will provide accurate results for all calculations. Later in this chapter other observations will be made to support the previous statements.

The preceding definitions for stability only apply when all of the norms of the characteristic values for a particular recurrence relation are partitioned on either
side of the value 1. In general a recurrence relation will not possess this trait, but will have characteristic values on both sides of 1.

An Example of Contrast

In the next example (3.14) a third order recurrence relation with characteristic values .1, 1.1, and 10.1 will be examined. The example will compare the methods of backward and forward recurrence for all three solutions. Following the example some observations about the results will be presented.

Example 3.14

For this example the following third order recurrence relation will be used:

\[(3.14.1)\]

\[y(n) - 11.3y(n-1) + 12.23y(n-2) - 1.111y(n-3) = 0.\]

The three characteristic values for this recurrence relation are .1, 1.1, and 10.1. For this example the following three solutions will be examined:

\[(3.14.2)\]

\[s_1 = .1^n, s_2 = 1.1^n, s_3 = 10.1^n.\]

It should be noted that \(s_1\) is a minimal solution, \(s_2\) is an intermediate solution, and \(s_3\) is a dominate solution.
For each of the solutions $s_1$, $s_2$, and $s_3$, both forward and backward recurrence will be examined. For the forward recurrence method the three initial conditions will be set, and values 3 through 15 will be calculated. For the backward recurrence method the three terminal values will be set and values 12 through 0 will be calculated.

If recurrence relation 3.14.1 is solved for the six different sets of conditions outlined above, the result is tables 3.14.3 through 3.14.8. []

Several important observations can be made from tables 3.14.3 through 3.14.8:

1. By definition (3.6, 3.9) recurrence relation 3.14.1 is not stable in the absolute sense for either forward or backward recurrence.

2. Although the recurrence relation is unstable, some of the calculations were very accurate.

3. There seems to be some connection between solving a recurrence relation in the forward direction and the dominate solution; likewise, a connection between solving a recurrence relation in the backward direction and the minimal solution.

4. The solutions, other than the most accurate, seem to get progressively worse. []
Table 3.14.3
Calculation of Solution $s_1$ of Equation 3.14.1
Using Forward Recurrence

<table>
<thead>
<tr>
<th>n</th>
<th>$w(n)$</th>
<th>$w_c(n)$</th>
<th>$e_a(n)$</th>
<th>$e_r(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1000000E+01</td>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.1000000E+00</td>
<td>0.1000000E+00</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.1000000E-01</td>
<td>0.1000000E-01</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.1000000E-02</td>
<td>0.1000016E-02</td>
<td>0.1623877E-07</td>
<td>0.1623877E-04</td>
</tr>
<tr>
<td>4</td>
<td>0.1000000E-03</td>
<td>0.1007510E-03</td>
<td>0.1907510E-06</td>
<td>0.1907510E-02</td>
</tr>
<tr>
<td>5</td>
<td>0.1000000E-04</td>
<td>0.1194840E-04</td>
<td>0.1948403E-05</td>
<td>0.1948403E-02</td>
</tr>
<tr>
<td>6</td>
<td>0.1000000E-05</td>
<td>0.2070286E-04</td>
<td>0.1970286E-04</td>
<td>0.1970286E-02</td>
</tr>
<tr>
<td>7</td>
<td>0.1000000E-06</td>
<td>0.1990253E-03</td>
<td>0.1990253E-03</td>
<td>0.1990253E-04</td>
</tr>
<tr>
<td>8</td>
<td>0.1000000E-07</td>
<td>0.2010949E-02</td>
<td>0.2010949E-02</td>
<td>0.2010949E-02</td>
</tr>
<tr>
<td>9</td>
<td>0.1000000E-08</td>
<td>0.2030289E-01</td>
<td>0.2030289E-01</td>
<td>0.2030289E-02</td>
</tr>
<tr>
<td>10</td>
<td>0.1000000E-09</td>
<td>0.2050293E-00</td>
<td>0.2050293E-00</td>
<td>0.2050293E-00</td>
</tr>
<tr>
<td>11</td>
<td>0.1000000E-10</td>
<td>0.2070990E+01</td>
<td>0.2070990E+01</td>
<td>0.2070990E+12</td>
</tr>
<tr>
<td>12</td>
<td>0.1000000E-11</td>
<td>0.2091810E+02</td>
<td>0.2091810E+02</td>
<td>0.2091810E+14</td>
</tr>
<tr>
<td>13</td>
<td>0.1000000E-12</td>
<td>0.2112781E+03</td>
<td>0.2112781E+03</td>
<td>0.2112781E+16</td>
</tr>
<tr>
<td>14</td>
<td>0.1000000E-13</td>
<td>0.2133855E+04</td>
<td>0.2133855E+04</td>
<td>0.2133855E+18</td>
</tr>
<tr>
<td>15</td>
<td>0.1000000E-14</td>
<td>0.2155193E+05</td>
<td>0.2155193E+05</td>
<td>0.2155193E+20</td>
</tr>
</tbody>
</table>
Table 3.14.4
Calculation of Solution $s_2$ of Equation 3.14.1
Using Forward Recurrence

<table>
<thead>
<tr>
<th>$n$</th>
<th>$w(n)$</th>
<th>$w_c(n)$</th>
<th>$e_{a}(n)$</th>
<th>$e_{c}(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1000000E+01</td>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.1100000E+01</td>
<td>0.1100000E+01</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.1210000E+01</td>
<td>0.1210000E+01</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.1331000E+01</td>
<td>0.1330999E+01</td>
<td>0.7450506E-07</td>
<td>0.5577731E-07</td>
</tr>
<tr>
<td>4</td>
<td>0.1464100E+01</td>
<td>0.1464099E+01</td>
<td>0.9685748E-06</td>
<td>0.6615009E-06</td>
</tr>
<tr>
<td>5</td>
<td>0.1610510E+01</td>
<td>0.1610500E+01</td>
<td>0.9924734E-05</td>
<td>0.6162105E-05</td>
</tr>
<tr>
<td>6</td>
<td>0.1771561E+01</td>
<td>0.1771465E+01</td>
<td>0.1047853E-03</td>
<td>0.5671751E-04</td>
</tr>
<tr>
<td>7</td>
<td>0.1948717E+01</td>
<td>0.1947701E+01</td>
<td>0.1015312E-02</td>
<td>0.5209375E-03</td>
</tr>
<tr>
<td>8</td>
<td>0.2143588E+01</td>
<td>0.2133343E+01</td>
<td>0.1025444E-01</td>
<td>0.4737733E-02</td>
</tr>
<tr>
<td>9</td>
<td>0.2357947E+01</td>
<td>0.2254376E+01</td>
<td>0.1035702E+00</td>
<td>0.4392435E-01</td>
</tr>
<tr>
<td>10</td>
<td>0.2593742E+01</td>
<td>0.2547638E+01</td>
<td>0.1046066E+00</td>
<td>0.4030396E+00</td>
</tr>
<tr>
<td>11</td>
<td>0.2853116E+01</td>
<td>-0.7712157E+01</td>
<td>0.1056274E+02</td>
<td>0.3700643E+01</td>
</tr>
<tr>
<td>12</td>
<td>0.3138423E+01</td>
<td>-0.1035084E+03</td>
<td>0.1067287E+03</td>
<td>0.3400086E+02</td>
</tr>
<tr>
<td>13</td>
<td>0.3452271E+01</td>
<td>-0.1074311E+04</td>
<td>0.1077636E+04</td>
<td>0.3121397E+03</td>
</tr>
<tr>
<td>14</td>
<td>0.3797458E+01</td>
<td>-0.1083161E+05</td>
<td>0.1088413E+05</td>
<td>0.2966495E+04</td>
</tr>
<tr>
<td>15</td>
<td>0.4177248E+01</td>
<td>-0.1093349E+06</td>
<td>0.1099426E+06</td>
<td>0.2631942E+05</td>
</tr>
</tbody>
</table>
Table 3.14.5
Calculation of Solution $s_3$ of Equation 3.14.1
Using Forward Recurrence

<table>
<thead>
<tr>
<th>n</th>
<th>$w(n)$</th>
<th>$v(n)$</th>
<th>$e_a(n)$</th>
<th>$e_t(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1000000E+01</td>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.1010000E+02</td>
<td>0.1010000E+02</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.1020100E+03</td>
<td>0.1020100E+03</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.1030301E+04</td>
<td>0.1030301E+04</td>
<td>0.1525678E-04</td>
<td>0.1481009E-07</td>
</tr>
<tr>
<td>4</td>
<td>0.1040604E+05</td>
<td>0.1040604E+05</td>
<td>0.1207031E-03</td>
<td>0.1173071E-07</td>
</tr>
<tr>
<td>5</td>
<td>0.1051010E+06</td>
<td>0.1051010E+06</td>
<td>0.1953125E-02</td>
<td>0.1858314E-07</td>
</tr>
<tr>
<td>6</td>
<td>0.1061520E+07</td>
<td>0.1061520E+07</td>
<td>0.1562500E-01</td>
<td>0.1471945E-07</td>
</tr>
<tr>
<td>7</td>
<td>0.1072135E+08</td>
<td>0.1072135E+08</td>
<td>0.3750000E+00</td>
<td>0.3497628E-07</td>
</tr>
<tr>
<td>8</td>
<td>0.1082857E+09</td>
<td>0.1082857E+09</td>
<td>0.3000000E+01</td>
<td>0.2770496E-07</td>
</tr>
<tr>
<td>9</td>
<td>0.1093685E+10</td>
<td>0.1093685E+10</td>
<td>0.4800000E+02</td>
<td>0.4388311E-07</td>
</tr>
<tr>
<td>10</td>
<td>0.1104622E+11</td>
<td>0.1104622E+11</td>
<td>0.5120000E+03</td>
<td>0.4635031E-07</td>
</tr>
<tr>
<td>11</td>
<td>0.1115668E+12</td>
<td>0.1115668E+12</td>
<td>0.6400000E+04</td>
<td>0.5507012E-07</td>
</tr>
<tr>
<td>12</td>
<td>0.1126925E+13</td>
<td>0.1126925E+13</td>
<td>0.5536000E+05</td>
<td>0.5319672E-07</td>
</tr>
<tr>
<td>13</td>
<td>0.1138393E+14</td>
<td>0.1138393E+14</td>
<td>0.6583600E+06</td>
<td>0.5759431E-07</td>
</tr>
<tr>
<td>14</td>
<td>0.1149474E+15</td>
<td>0.1149474E+15</td>
<td>0.6291456E+07</td>
<td>0.5473337E-07</td>
</tr>
<tr>
<td>15</td>
<td>0.1160980E+16</td>
<td>0.1160980E+16</td>
<td>0.1006633E+09</td>
<td>0.8670637E-07</td>
</tr>
</tbody>
</table>
Table 3.14.6
Calculation of Solution $s_1$ of Equation 3.14.1
Using Backward Recurrence

<table>
<thead>
<tr>
<th>$n$</th>
<th>$w(n)$</th>
<th>$w_C(n)$</th>
<th>$e_a(n)$</th>
<th>$e_L(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0.1000000E-14</td>
<td>0.1000000E-14</td>
<td>0.1323489E-22</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>14</td>
<td>0.1000000E-13</td>
<td>0.1000000E-13</td>
<td>0.1097912E-21</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.1000000E-12</td>
<td>0.1000000E-12</td>
<td>0.8470329E-21</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>12</td>
<td>0.1000000E-11</td>
<td>0.1000000E-11</td>
<td>0.1355252E-19</td>
<td>0.1355252E-07</td>
</tr>
<tr>
<td>11</td>
<td>0.1000000E-10</td>
<td>0.1000000E-10</td>
<td>0.1084202E-18</td>
<td>0.1084202E-07</td>
</tr>
<tr>
<td>0</td>
<td>0.1000000E-09</td>
<td>0.1000000E-09</td>
<td>0.1374723E-17</td>
<td>0.1374723E-07</td>
</tr>
<tr>
<td>9</td>
<td>0.1000000E-08</td>
<td>0.1000000E-08</td>
<td>0.2775576E-16</td>
<td>0.2775576E-07</td>
</tr>
<tr>
<td>8</td>
<td>0.1000000E-07</td>
<td>0.1000000E-07</td>
<td>0.2220446E-15</td>
<td>0.2220446E-07</td>
</tr>
<tr>
<td>7</td>
<td>0.1000000E-06</td>
<td>0.1000000E-06</td>
<td>0.2664536E-14</td>
<td>0.2664536E-07</td>
</tr>
<tr>
<td>6</td>
<td>0.1000000E-05</td>
<td>0.1000000E-05</td>
<td>0.4265256E-13</td>
<td>0.4265256E-07</td>
</tr>
<tr>
<td>5</td>
<td>0.1000000E-04</td>
<td>0.1000000E-04</td>
<td>0.4547473E-12</td>
<td>0.4547473E-07</td>
</tr>
<tr>
<td>4</td>
<td>0.1000000E-03</td>
<td>0.1000000E-03</td>
<td>0.3637978E-11</td>
<td>0.3637978E-07</td>
</tr>
<tr>
<td>3</td>
<td>0.1000000E-02</td>
<td>0.1000000E-02</td>
<td>0.2910383E-10</td>
<td>0.2910383E-07</td>
</tr>
<tr>
<td>2</td>
<td>0.1000000E-01</td>
<td>0.1000000E-01</td>
<td>0.3492459E-09</td>
<td>0.3492459E-07</td>
</tr>
<tr>
<td>1</td>
<td>0.1000000E+00</td>
<td>0.1000000E+00</td>
<td>0.3725290E-08</td>
<td>0.3725290E-07</td>
</tr>
<tr>
<td>0</td>
<td>0.1000000E+00</td>
<td>0.1000000E+00</td>
<td>0.4470348E-07</td>
<td>0.4470348E-07</td>
</tr>
</tbody>
</table>
Table 3.14.7

Calculation of Solution $s_2$ of Equation 3.14.1
Using Backward Recurrence

<table>
<thead>
<tr>
<th>n</th>
<th>$w(n)$</th>
<th>$w_C(n)$</th>
<th>$e_a(n)$</th>
<th>$e_a(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0.4177248E+01</td>
<td>0.4177248E+01</td>
<td>0.1788139E-06</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>14</td>
<td>0.3797498E+01</td>
<td>0.3797498E+01</td>
<td>0.8940697E-07</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.3452271E+01</td>
<td>0.3452271E+01</td>
<td>0.1192052E-06</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>12</td>
<td>0.3138428E+01</td>
<td>0.3138428E+01</td>
<td>0.1192052E-06</td>
<td>0.3793754E-07</td>
</tr>
<tr>
<td>11</td>
<td>0.2851117E+01</td>
<td>0.2851117E+01</td>
<td>0.9536743E-06</td>
<td>0.3342570E-06</td>
</tr>
<tr>
<td>10</td>
<td>0.2593742E+01</td>
<td>0.2593742E+01</td>
<td>0.9236719E-05</td>
<td>0.3561926E-05</td>
</tr>
<tr>
<td>9</td>
<td>0.2357947E+01</td>
<td>0.2357947E+01</td>
<td>0.9235739E-04</td>
<td>0.3918553E-04</td>
</tr>
<tr>
<td>8</td>
<td>0.2143588E+01</td>
<td>0.2143588E+01</td>
<td>0.9236338E-03</td>
<td>0.4309375E-03</td>
</tr>
<tr>
<td>7</td>
<td>0.1948717E+01</td>
<td>0.1948717E+01</td>
<td>0.9237079E-02</td>
<td>0.4740302E-02</td>
</tr>
<tr>
<td>6</td>
<td>0.1771560E+01</td>
<td>0.1771560E+01</td>
<td>0.9237084E-01</td>
<td>0.5214030E-01</td>
</tr>
<tr>
<td>5</td>
<td>0.1610510E+01</td>
<td>0.1610510E+01</td>
<td>0.9237084E-00</td>
<td>0.5733500E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.1464100E+01</td>
<td>0.1464100E+01</td>
<td>0.9237085E+00</td>
<td>0.6300957E+01</td>
</tr>
<tr>
<td>3</td>
<td>0.1331000E+01</td>
<td>0.1331000E+01</td>
<td>0.9237085E+00</td>
<td>0.6935955E+02</td>
</tr>
<tr>
<td>2</td>
<td>0.1210000E+01</td>
<td>0.1210000E+01</td>
<td>0.9237085E+00</td>
<td>0.7633955E+03</td>
</tr>
<tr>
<td>1</td>
<td>0.1100000E+01</td>
<td>0.1100000E+01</td>
<td>0.9237085E+00</td>
<td>0.8393955E+04</td>
</tr>
<tr>
<td>0</td>
<td>0.1000000E+01</td>
<td>0.9237185E+05</td>
<td>0.9237085E+05</td>
<td>0.9237085E+05</td>
</tr>
</tbody>
</table>
Table 3.14.8
Calculation of Solution \( s_3 \) of Equation 3.14.1
Using Backward Recurrence

<table>
<thead>
<tr>
<th>( n )</th>
<th>( w(n) )</th>
<th>( w_c(n) )</th>
<th>( e_a(n) )</th>
<th>( e_p(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0.11609690E+16</td>
<td>0.11609690E+16</td>
<td>0.50331545E+03</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>14</td>
<td>0.11494742E+15</td>
<td>0.11494742E+15</td>
<td>0.20971520E+07</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.11380933E+14</td>
<td>0.11380933E+14</td>
<td>0.26214400E+06</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>12</td>
<td>0.11268251E+13</td>
<td>0.11268251E+13</td>
<td>0.34078720E+07</td>
<td>0.30431338E-05</td>
</tr>
<tr>
<td>11</td>
<td>0.11156645E+12</td>
<td>0.11156645E+12</td>
<td>0.38670360E+08</td>
<td>0.34651198E-03</td>
</tr>
<tr>
<td>10</td>
<td>0.11046221E+11</td>
<td>0.11046221E+11</td>
<td>0.39088800E+09</td>
<td>0.35831518E-01</td>
</tr>
<tr>
<td>9</td>
<td>0.10936858E+10</td>
<td>0.50071018E+10</td>
<td>0.39120249E+10</td>
<td>0.35769201E-01</td>
</tr>
<tr>
<td>8</td>
<td>0.10828567E+09</td>
<td>0.39231999E+11</td>
<td>0.39123654E+11</td>
<td>0.36130037E-03</td>
</tr>
<tr>
<td>7</td>
<td>0.10721354E+08</td>
<td>0.39126056E+12</td>
<td>0.39123960E+12</td>
<td>0.36491255E+05</td>
</tr>
<tr>
<td>6</td>
<td>0.10615202E+07</td>
<td>0.39124001E+13</td>
<td>0.39123993E+13</td>
<td>0.36895669E+07</td>
</tr>
<tr>
<td>5</td>
<td>0.10510101E+06</td>
<td>0.39123993E+14</td>
<td>0.39123994E+14</td>
<td>0.37251373E+09</td>
</tr>
<tr>
<td>4</td>
<td>0.10405040E+05</td>
<td>0.39123993E+15</td>
<td>0.39123993E+15</td>
<td>0.37597338E+11</td>
</tr>
<tr>
<td>3</td>
<td>0.10300010E+04</td>
<td>0.39123994E+16</td>
<td>0.39123994E+16</td>
<td>0.37973323E+13</td>
</tr>
<tr>
<td>2</td>
<td>0.10200000E+03</td>
<td>0.39123994E+17</td>
<td>0.39123994E+17</td>
<td>0.38353097E+15</td>
</tr>
<tr>
<td>1</td>
<td>0.10100000E+02</td>
<td>0.39123994E+18</td>
<td>0.39123994E+18</td>
<td>0.38736628E+17</td>
</tr>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>0.39123994E+19</td>
<td>0.39123994E+19</td>
<td>0.39123994E+19</td>
</tr>
</tbody>
</table>
The above observations can be formally justified (Wimp 1984), thus the observations may be considered as typical results.

More Advanced Approaches to the Solution of Recurrence Relations

Attention will now be turned to some of the more advanced methods for the solution of recurrence relations. Two popular methods for the solution of recurrence relations will be presented. The first method to be considered is Miller's algorithm, which makes use of backward recurrence. The second method will be Oliver's method, which makes use of the associated linear system of equations and the LU decomposition.

Chapter four will refer extensively to Oliver's method. It will be used for the determination of the number of initial and terminal conditions to include in examples.

The Miller Algorithm for a First Order Recurrence Relation

The Miller algorithm (Miller J.C.P. 1952; Wimp 1984) will make use of backward recurrence. Little or no information is required about the result w(n), for large n.
The Miller algorithm will be presented in steps. The first step will be in the solution of a first order recurrence relation.

**Definition 3.15**

For the first order recurrence relation

\[ y(n) + a_1(n)y(n-1) = 0, \]

Set \( w_N(N) = 0 \), and use 3.15.1 to solve in the backward direction for \( w_N(n-1), w_N(n-2), \ldots, w_N(0) \).

Then take \( w_N(n) \) as an estimate to \( y(n) \), for \( n = 1, 2, \ldots, N \). This is the Miller algorithm for a first order recurrence relation. []

The value of \( N \) is the critical point in the algorithm. The larger the value of \( N \) the more accurate the solution will tend to be. In the following example (3.16), increasing values for \( N \) will be shown to increase the accuracy of the desired solution.

**Example 3.16**

The following recurrence relation will be used to solve for the values \( y(n) \), \( n = 4, 3, \ldots, 0 \):

\[ y(n) - \frac{x}{(n+1)}y(n+1) = \frac{(-1)^ne^x - e^{-x}}{(n+1)}, \]

where \( x \) is a constant with value 1.0.
The Miller algorithm will be used for the computations with $N = 4, 8, 12, \text{ and } 16$. Table 3.16.2 is a reformation of table 3.1 from Wimp (1984).

Table 3.16.2

<table>
<thead>
<tr>
<th>$n$</th>
<th>$N$ 4</th>
<th>8</th>
<th>12</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.327386</td>
<td>2.350395</td>
<td>2.350402</td>
<td>2.350402</td>
</tr>
<tr>
<td>1</td>
<td>-0.758774</td>
<td>-0.735767</td>
<td>0.735759</td>
<td>-0.753759</td>
</tr>
<tr>
<td>2</td>
<td>0.832854</td>
<td>0.878869</td>
<td>0.878885</td>
<td>0.878885</td>
</tr>
<tr>
<td>3</td>
<td>-0.587601</td>
<td>-0.449555</td>
<td>-0.449507</td>
<td>-0.449507</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.552182</td>
<td>0.552373</td>
<td>.0.552373</td>
</tr>
</tbody>
</table>

The values in the last column of table 3.16.1 are accurate to the displayed values.

In table 3.16.2 it should be noticed that a convergence is taking place. It can be shown (Wimp 1984) that the solution $w_N(n)$ will converge to the desired solution $w(n)$ as $N$ gets larger.

The Miller Algorithm for a Second Order Recurrence Relation

The application of the Miller algorithm to a second order recurrence relation will require that a convergent normalizing series for the recurrence relation be known.
Definition 3.17

For the second order recurrence relation

\[ y(n) + a_1(n)y(n-1) + a_2(n)y(n-2) = 0, \]

assume a normalizing series \( S_N \) for the approximation \( w_N(n) \) of \( w(n) \), with \( 0 \leq n \leq N + 1 \), which has the following form:

\[ S_N = \sum_{k=0}^{N} c(k)w(k). \]

If the approximation for \( w(n) \) is given by:

\[ w_N(n) = sy_N(n) / S_N, \]

and \( y_N \) is defined as follows:

\[ y_N(n) = \begin{cases} 0 & \text{for } n = N+1, \\ 1 & \text{for } n = N, \text{ and} \\ \text{computed by recurrence relation 3.17.1} & \text{for } n = N-1, N-2, \ldots, 0, \text{ in that order.} \end{cases} \]

This is the Miller algorithm for a second order recurrence relation. []

The normalizing series for the Miller algorithm is not unique for a particular recurrence relation. The simple fact of knowing the last value that is to be computed, \( y(0) \), is possibly enough information.
The idea of setting $y(N+1) = 0$ and $y(N) = 1$ is also open for modification. If information is known about the recurrence relation, then setting $y(N+1)$ and $y(N)$ to values that more closely represent the solution may provide a better convergence to the true solution.

In the next example (3.18) some of the mentioned variations to the Miller algorithm are examined.

Example 3.18

For this example the following second order recurrence relation, which represents the Bessel function with the solution $J(n)$, will be used:

(3.18.1)

$$y(n) = \frac{2(n+1)}{x} y(n+1) + y(n+2),$$

where $x$ is a constant with the value 1.0.

There will be three sets of calculations for this example. Each set will be for a separate pair of terminal values ($y(N)$ and $y(N+1)$). For each pair of terminal values, the values for the recurrence relation will be computed and displayed.

For any given recurrence relation there is an infinite number of satisfactory normalizing series. For this example three normalizing series will be provided. The three normalizing series are now given.
(3.18.2) \[ e^x = \sum_{k=1}^{N} e_k w_k(x), \]

where \( e_k = 1 \) if \( k = 0 \), otherwise, \( e_k = 2 \), and \( w_k \) is the calculated value \( w_c(k) \).

The second normalizing series uses the fact that the value for \( w(0) \) is known and is 1.266065876. This results in the following normalizing series

(3.18.3) \[
\frac{w(0)}{w_c(0)}. \]

The third normalizing series is given by the following equation

(3.18.4) \[ 1 = \sum_{k=1}^{N} e_k(-1)^k w_{2k}(x), \]

where \( e_k = 1 \) if \( k = 0 \), otherwise, \( e_k = 2 \), and \( w_k \) is the calculated value \( w_c(k) \).

The three pairs of terminal values will now be given along with a short explanation for each.

(3.18.5)

The first values for \( y(N) \) and \( y(N+1) \) will be as defined for the Miller algorithm. The value of 1 will be assigned to \( y(9) \) and the value 0 will be assigned to \( y(10) \). The values \( w_c(8) \) through \( w_c(0) \) will then be computed.
The second pair of terminal conditions will try to make use of setting $y(10)$ and $y(9)$ so that their ratio corresponds to the asymptotic behavior of the solution. The value for $y(9)$ is still 1, but $y(10)$ will be given the value $(x/2.0)/10.0$.

The third pair of terminal conditions will try to set $y(10)$ and $y(9)$ to their correct values according to the asymptotics of the solution. The value for $y(9)$ is $(x/2.0)^9/9!$, and $y(10)$ will have the value $(x/2.0)^{10}/10!$.

Table 3.18.8 shows the calculations of the recurrence relation 3.18.1 for the three pairs of values for $y(10)$ and $y(9)$.

Table 3.18.9 shows the normalizing constants as computed from table 3.18.8 using the three normalizing series (3.18.2, 3.18.3, 3.18.4).

It should be noticed, from table 3.18.9, that all three normalizing series result in the same values for the normalizing constants. The situation of all the normalizing constants being equal is exceptional. In most situations these values are different, but usually very close as long as the formulas for the normalizing series are reasonable.
Table 3.18.8

Computations for Equation 3.18.1 Using the Miller Algorithm and Different Values for $y(N)$, $y(N+1)$

<table>
<thead>
<tr>
<th>n</th>
<th>3.18.5</th>
<th>3.18.6</th>
<th>3.18.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0000000E+00</td>
<td>0.5000000E-01</td>
<td>0.2691144E-09</td>
</tr>
<tr>
<td>9</td>
<td>0.1000000E+01</td>
<td>0.1000000E+01</td>
<td>0.5382289E-08</td>
</tr>
<tr>
<td>8</td>
<td>0.1800000E+02</td>
<td>0.1805000E+02</td>
<td>0.9715031E-07</td>
</tr>
<tr>
<td>7</td>
<td>0.2890000E+03</td>
<td>0.2898000E+03</td>
<td>0.1559787E-05</td>
</tr>
<tr>
<td>6</td>
<td>0.4064000E+04</td>
<td>0.4075250E+04</td>
<td>0.2193417E-04</td>
</tr>
<tr>
<td>5</td>
<td>0.4905700E+05</td>
<td>0.4919280E+05</td>
<td>0.2647699E-03</td>
</tr>
<tr>
<td>4</td>
<td>0.4946340E+06</td>
<td>0.4960032E+06</td>
<td>0.2669633E-02</td>
</tr>
<tr>
<td>3</td>
<td>0.4006129E+07</td>
<td>0.4017219E+07</td>
<td>0.2162183E-01</td>
</tr>
<tr>
<td>2</td>
<td>0.2453141E+08</td>
<td>0.2459932E+08</td>
<td>0.1324006E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.1021318E+09</td>
<td>0.1024145E+09</td>
<td>0.5512243E+00</td>
</tr>
<tr>
<td>0</td>
<td>0.2287949E+09</td>
<td>0.2294283E+09</td>
<td>0.1234849E+01</td>
</tr>
</tbody>
</table>
Table 3.18.9

Computations of Normalizing Constants

<table>
<thead>
<tr>
<th>Terminal Conditions</th>
<th>Normalizing Series</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.18.2</td>
</tr>
<tr>
<td>3.18.5</td>
<td>0.5533627E-08</td>
</tr>
<tr>
<td>3.18.6</td>
<td>0.5518351E-08</td>
</tr>
<tr>
<td>3.18.7</td>
<td>0.1025280E+01</td>
</tr>
</tbody>
</table>

Table 3.18.10 is the result of applying the normalizing constants (3.18.9) to the computed data in table 3.18.8. This produces the desired normalized results. Since the three sets of normalizing constants are identical for the different normalizing series, all three sets of calculations result in identical tables. For this reason, only one table of final results is presented. Table 3.18.10 contains the normalized results for example 3.18. [ ]
Table 3.18.10

Results of Applying Normalizing Constants From
Table 3.18.9 to the Data Table 3.18.8

<table>
<thead>
<tr>
<th>n</th>
<th>3.18.5</th>
<th>3.18.6</th>
<th>3.18.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0000000E+00</td>
<td>0.2759176E-09</td>
<td>0.2759176E-09</td>
</tr>
<tr>
<td>9</td>
<td>0.5533627E-08</td>
<td>0.5518351E-08</td>
<td>0.5518351E-08</td>
</tr>
<tr>
<td>8</td>
<td>0.9960529E-07</td>
<td>0.9960624E-07</td>
<td>0.9960624E-07</td>
</tr>
<tr>
<td>7</td>
<td>0.1599218E-05</td>
<td>0.1599218E-05</td>
<td>0.1599218E-05</td>
</tr>
<tr>
<td>6</td>
<td>0.2248866E-04</td>
<td>0.2248866E-04</td>
<td>0.2248866E-04</td>
</tr>
<tr>
<td>5</td>
<td>0.2714632E-03</td>
<td>0.2714632E-03</td>
<td>0.2714632E-03</td>
</tr>
<tr>
<td>4</td>
<td>0.2737120E-02</td>
<td>0.2737120E-02</td>
<td>0.2737120E-02</td>
</tr>
<tr>
<td>3</td>
<td>0.2216842E-01</td>
<td>0.2216842E-01</td>
<td>0.2216843E-01</td>
</tr>
<tr>
<td>2</td>
<td>0.1357477E+00</td>
<td>0.1357477E+00</td>
<td>0.1357477E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.5651591E+00</td>
<td>0.5651591E+00</td>
<td>0.5651591E+00</td>
</tr>
<tr>
<td>0</td>
<td>0.1266066E+01</td>
<td>0.1266066E+01</td>
<td>0.1266066E+01</td>
</tr>
</tbody>
</table>

It should be noticed, from table 3.18.10, that a slightly faster convergence by the two variant methods for setting the terminal conditions occurred. This again points out the fact that the choice of N in reference to
The value of $N$ should be chosen as large as needed to offset a possibly slow converging normalizing series or one of many other possible problems.

Many modifications of the Miller algorithm have been proposed through the literature (Olver 1968; Scraton 1972). These enhancements provide better results than the unaltered Miller algorithm, but as expected the modified versions are usually more complex and usually involve higher overhead costs.

The formal error analysis for Miller's algorithm is very complex and beyond the scope of the initial investigation presented in this work. Throughout the literature, several methods for error analysis involving Miller's algorithm were found (Mattheij and van der Sluis 1976; Olver 1964; Shintani 1965; Zahar 1977).

It should be noted that if an associated linear system approach is used to solve Miller's algorithm, and that the decomposition routine uses Gaussian elimination to decompose the matrix, this is an example of backward recurrence. This means that an error analysis for Gaussian elimination could also provide meaningful insights into the error analysis for the Miller algorithm (Wilkinson 1971; Businger 1971; Reid 1971a).
Oliver's Theorem

It is just as common to need calculations of an intermediate solution to a recurrence relation as calculations of the minimal or dominate solutions. As shown by previous examples, the use of forward or backward recurrence does not provide reliable results for the intermediate solutions. The work of Oliver (1967, 1968a, 1968b) and that of Lozier (1980) provides the fundamental methods for the calculation of intermediate solutions.

A slightly simplified version of a theorem due to Oliver, for calculating solutions to recurrence relations, will be presented next.

Theorem 3.19

Let \( y \) be a solution of a recurrence relation (2.7.1), also suppose there is a basis

(3.19.1)

\[
(y) \cup \{y_k : \text{for } k = 1, 2, ..., p, \text{ with } k \neq i + 1 \}
\]

such that \( y_k < y \) for \( i = 1, 2, ..., i \), and \( y < y_k \) for \( k = i + 2, i + 3, ..., p \).

Where \( u < v \) means

\[
\lim_{n \to \infty} \frac{u(n)}{v(n)} \leq cL^n, \text{ for } c \text{ a constant, } 0 < L < 1.
\]

Also suppose that all leading determinants of \( A_r \) (2.30) are non-zero.
Then the boundary value problem with $i$ initial conditions and $j = p-i$ terminal conditions is stable when solved using the LU factorization without pivoting on the associated reduced linear system. []

For an example of Oliver's theorem, the recurrence relation from example 3.14 will be used. Looking back at the results for the intermediate solution of recurrence relation 3.14 (tables 3.14.4 and 3.14.7), there was an extremely large relative error in the calculations. Using Oliver's theorem, we will see that the relative error is reduced to near machine precision.

Example 3.20

The third order recurrence relation 3.14.1, with the intermediate solution $l.ln$, will be investigated.

According to Oliver's theorem, for the intermediate solution $s_2$, the number of initial conditions required is 1 and the number of terminal conditions is therefore 2.

If the LU factorization without pivoting is used to decompose and solve the associated reduced linear system, table 3.20.1 is the result. []

Table 3.20.1 should be compared with tables 3.14.4 and 3.14.7, which are also for solution $s_2$ of recurrence relation 3.14.1. This comparison will provide evidence to support Oliver's theorem for the computation of intermediate solutions to recurrence relations.
Table 3.20.1

Computation of the Intermediate Solution $s_2$ of Equation 3.14.1 Using Oliver's Theorem

<table>
<thead>
<tr>
<th>n</th>
<th>$w(n)$</th>
<th>$w_c(n)$</th>
<th>$e_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1000000E+01</td>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.1100000E+01</td>
<td>0.1100000E+01</td>
<td>0.4063953E-07</td>
</tr>
<tr>
<td>2</td>
<td>0.1210000E+01</td>
<td>0.1210000E+01</td>
<td>0.3694503E-07</td>
</tr>
<tr>
<td>3</td>
<td>0.1331000E+01</td>
<td>0.1331000E+01</td>
<td>0.3358639E-07</td>
</tr>
<tr>
<td>4</td>
<td>0.1464100E+01</td>
<td>0.1464100E+01</td>
<td>0.4071077E-07</td>
</tr>
<tr>
<td>5</td>
<td>0.1610510E+01</td>
<td>0.1610510E+01</td>
<td>0.3700980E-07</td>
</tr>
<tr>
<td>6</td>
<td>0.1771561E+01</td>
<td>0.1771561E+01</td>
<td>0.3364527E-07</td>
</tr>
<tr>
<td>7</td>
<td>0.1948717E+01</td>
<td>0.1948717E+01</td>
<td>0.7646652E-08</td>
</tr>
<tr>
<td>8</td>
<td>0.2143589E+01</td>
<td>0.2143589E+01</td>
<td>0.1390300E-07</td>
</tr>
<tr>
<td>9</td>
<td>0.2357948E+01</td>
<td>0.2357948E+01</td>
<td>0.1263909E-07</td>
</tr>
<tr>
<td>10</td>
<td>0.2593742E+01</td>
<td>0.2593742E+01</td>
<td>0.2298017E-07</td>
</tr>
</tbody>
</table>

A complete error analysis for Oliver's theorem is outside the scope of this work (Oliver 1967, 1968a, 1968b). The fact that Oliver's theorem uses an LU decomposition, which uses Gaussian elimination, lends itself to an error analysis of Gaussian elimination (Wilkinson 1971; Businger 1971; Reid 1971a).
CHAPTER IV

EMPIRICAL INVESTIGATION AND COMPARISONS

This chapter will be an empirical investigation into the linear algebraic solution methods discussed throughout this work. The chapter will compare several aspects of the solution process. An introduction of condition numbers for a matrix will be given and shown to be a relative indicator to the reliability of computations.

For each of the comparisons presented, an introductory conclusion will be given. These conclusions are combinations of the theoretical and empirical considerations. Observations involving the complete example set will be included as support for each conclusion.

Suggestions for enhancements and modifications to existing methods will be presented. These enhancements and modifications will then be investigated through examples. Conclusions involving the enhancements and modifications along with explanations will be given.
Condition Number of a Matrix

A condition number of the associated linear system for a recurrence relation may provide some insight into the reliability of the computations. A condition number may be used to estimate the effects of small changes in the linear system on the solution of that system (Dongarra et al. 1978; Golub and Van Loan 1983; Stewart 1973).

If a condition number is "large," the system is said to be ill-conditioned. For a linear system Ay = b, this means that small changes, or induced errors, in A or b could cause large errors in the solution y.

If a condition number is "small," the system is said to be well-conditioned. For a linear system Ay = b, this means that small changes, or induced errors, in A or b will have less effect on the solution y then if the system were ill-conditioned.

It should be noted that a condition number is a relative measure. The terms "large" and "small" are not defined, but must be interpreted for each situation. It can easily be shown that some ill-conditioned systems provide acceptable results and some well-conditioned systems provide poor results. For this reason a condition number must only be used as a reference and not an absolute measure of result reliability.
The literature has provided many methods for the computation and interpretation of a condition number for a matrix (Dongarra et al. 1978; Fenner and Loizou 1974; O'Leary 1980; Stewart 1973).

The LINPACK subroutines (Dongarra et al. 1978) discussed in chapter six, and used throughout this work, provide condition numbers as part of their calculations. The LINPACK routines return the inverse of the condition number. This convention was chosen to prevent the possibility of overflow or division by zero in the routines.

To show that the condition number is truly a relative measure to the reliability of the calculations, the following two examples (4.1, 4.2) will be presented.

The examples will present recurrence relations that provide excellent results for the relative errors in their solutions; but, this will not be obvious from the computed condition number.

The solution to the recurrence relations will be done through the use of the associated linear system of equations. The linear system $Ay = b$ will be solved. The matrix $A$ will be a square matrix of size $n$ by $n$, the vector $b$ will show the inclusion of the initial and terminal conditions, and the vector $y$ will be the solution.
The recurrence relations in the following two examples (4.1, 4.2) will be solved for the indicated solutions. The use of the LU decomposition with no partial pivoting, along with the augmented form of the associated linear system of equations (2.30), will be utilized.

Example 4.1

The following third order recurrence relation (4.1.1) is an example where the condition number for the associated linear system of equations is relatively small. It would be expected that this recurrence relation would provide reliable results.

\[ y(n) = 1.9y(n-1) - \frac{101}{90}y(n-2) + .2y(n-3), \]

for \( n = 0, 1, \ldots, 15 \), with a basis of solutions:

\[ \{ \frac{1}{3}^n, (2/3)^n, .9^n \}. \]

Table 4.1.3 shows the augmented associated linear system of equations for recurrence relation 4.1.1. Table 4.1.3 shows the matrix \( A \) in two parts. To physically visualize the complete matrix as a whole, the bottom section must be placed to the right of the top section.

For this example the solution \( (2/3)^n \) will be computed. Two initial conditions and one terminal condition will be provided. The result of including the
initial and terminal conditions is reflected in the matrix $A$ in table 4.1.3 and the vector $b$ in table 4.1.4.

If an LU decomposition with no pivoting is used to decompose and solve the system $Ay = b$, for the above conditions, the solution is the vector $y$ in table 4.1.5. The relative error comparison of the computed solution to the true solution is provided in table 4.1.6. 

The computed condition number for the associated matrix for example 4.1 is $1.5136779E+00$. It should be noted that this value is relatively small. This would be a good indication that the computations were reliable.

The next example (4.2) will show a recurrence relation that provides excellent results despite the associated matrix having a large condition number.

Example 4.2

The following fourth order recurrence relation (4.2.1) has a relatively large condition number, and might be expected to produce unreliable results.

(4.2.1)

$$y(n) = 7y(n-1) - 14.1875y(n-2) + 9.125y(n-3) - 1.5y(n-4),$$

for $n = 0, 1, \ldots, 15$, with a basis of solutions:

(4.2.2)

$$\{ (1/4)^n, (1/2)^n, 2^n, 4^n \}.$$
Table 4.2.3 shows the augmented associated linear system of equations for recurrence relation 4.2.1. Table 4.2.3 shows the matrix $A$ in two parts. To physically visualize the complete matrix as a whole, the bottom section must be placed to the right of the top section.

For this example the solution $4^n$ will be computed. Four initial conditions and no terminal conditions will be provided. The result of including the initial and terminal conditions is reflected in the matrix $A$ in table 4.2.3 and the vector $b$, in table 4.2.4.

If an LU decomposition with no pivoting is used to decompose and solve the system $Ay = b$, for the above conditions, the solution is the vector $y$ in table 4.2.5. The relative error comparison of the computed solution to the true solution is provided in table 4.2.6. []

The computed condition number for the associated matrix of example 4.2 is $4.1882521E+09$. It should be noted that this value is relatively large. This would be a good indication that the computations were unreliable.

From examples 4.1 and 4.2, it should be observed that the range of a condition number can be large. The examples are to represent a typical pair of recurrence relations. Examples can easily be found that have a much lower condition number, but have unreliable results. Therefore, this should only be used as an estimate.
| 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 |
| 0.200000E+00 | 0.112222E+00 | 0.190000E+00 | 0.100000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 |
| 0.000000E+00 | 0.200000E+00 | 0.112222E+00 | 0.190000E+00 | 0.100000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 |
| 0.000000E+00 | 0.000000E+00 | 0.200000E+00 | 0.112222E+00 | 0.190000E+00 | 0.100000E+00 | 0.000000E+00 | 0.000000E+00 |
| 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.200000E+00 | 0.112222E+00 | 0.190000E+00 | 0.100000E+00 | 0.000000E+00 |
| 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.200000E+00 | 0.112222E+00 | 0.190000E+00 | 0.100000E+00 |
| 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.200000E+00 | 0.112222E+00 | 0.190000E+00 |
| 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.200000E+00 | 0.112222E+00 |
| 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.000000E+00 | 0.200000E+00 |
Table 4.1.4
The Vector $b_a$ of Example 4.1

<table>
<thead>
<tr>
<th>n</th>
<th>$b(n)$</th>
<th>n</th>
<th>$b(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>8</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.66666666E+00</td>
<td>9</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.00000000E+00</td>
<td>10</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.00000000E+00</td>
<td>11</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.00000000E+00</td>
<td>12</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td>13</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.00000000E+00</td>
<td>14</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.00000000E+00</td>
<td>15</td>
<td>0.22836580E-02</td>
</tr>
</tbody>
</table>

Table 4.1.5
The Vector $y_a$ of Example 4.1

<table>
<thead>
<tr>
<th>n</th>
<th>$y(n)$</th>
<th>n</th>
<th>$y(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>8</td>
<td>0.39018380E-01</td>
</tr>
<tr>
<td>1</td>
<td>0.66666666E+00</td>
<td>9</td>
<td>0.26012251E-01</td>
</tr>
<tr>
<td>2</td>
<td>0.44444437E+00</td>
<td>10</td>
<td>0.17341500E-01</td>
</tr>
<tr>
<td>3</td>
<td>0.29629617E+00</td>
<td>11</td>
<td>0.11561000E-01</td>
</tr>
<tr>
<td>4</td>
<td>0.19753073E+00</td>
<td>12</td>
<td>0.77073345E-02</td>
</tr>
<tr>
<td>5</td>
<td>0.13168712E+00</td>
<td>13</td>
<td>0.51382244E-02</td>
</tr>
<tr>
<td>6</td>
<td>0.87791390E-01</td>
<td>14</td>
<td>0.34254845E-02</td>
</tr>
<tr>
<td>7</td>
<td>0.58527581E-01</td>
<td>15</td>
<td>0.22836580E-02</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
### Table 4.1.6

The Relative Errors for Example 4.1

<table>
<thead>
<tr>
<th>n</th>
<th>LU Decomposition Without Pivoting</th>
<th>Estimate of Exact Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00000000E+00</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>1</td>
<td>0.11175871E-07</td>
<td>0.66666666E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.16763807E-06</td>
<td>0.44444444E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.40233136E-06</td>
<td>0.29629629E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.66007489E-06</td>
<td>0.19753086E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.91939003E-06</td>
<td>0.13168724E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.11669181E-05</td>
<td>0.87791492E-01</td>
</tr>
<tr>
<td>7</td>
<td>0.13684767E-05</td>
<td>0.58527661E-01</td>
</tr>
<tr>
<td>8</td>
<td>0.15395363E-05</td>
<td>0.39018441E-01</td>
</tr>
<tr>
<td>9</td>
<td>0.16469458E-05</td>
<td>0.26012294E-01</td>
</tr>
<tr>
<td>10</td>
<td>0.16916998E-05</td>
<td>0.17341529E-01</td>
</tr>
<tr>
<td>11</td>
<td>0.16514212E-05</td>
<td>0.11561019E-01</td>
</tr>
<tr>
<td>12</td>
<td>0.15104463E-05</td>
<td>0.77073461E-02</td>
</tr>
<tr>
<td>13</td>
<td>0.12234615E-05</td>
<td>0.51382307E-02</td>
</tr>
<tr>
<td>14</td>
<td>0.75616718E-06</td>
<td>0.34254871E-02</td>
</tr>
<tr>
<td>15</td>
<td>0.25488782E-07</td>
<td>0.22836580E-02</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 4.2.3
The Matrix $A_a$ of Example 4.2
(left half on top, right half on bottom)

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.100000E+01</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>0.000000E+00</td>
<td>0.100000E+01</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.100000E+01</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>0.150000E+01</td>
<td>-0.912500E+00</td>
<td>0.141875E+00</td>
<td>0.141875E+00</td>
<td>-0.700000E+00</td>
<td>0.100000E+01</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>0.000000E+00</td>
<td>0.150000E+01</td>
<td>-0.912500E+00</td>
<td>0.141875E+00</td>
<td>0.141875E+00</td>
<td>-0.700000E+00</td>
<td>0.100000E+01</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.150000E+01</td>
<td>-0.912500E+00</td>
<td>0.141875E+00</td>
<td>0.141875E+00</td>
<td>-0.700000E+00</td>
<td>0.100000E+01</td>
</tr>
<tr>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.150000E+01</td>
<td>-0.912500E+00</td>
<td>0.141875E+00</td>
<td>0.141875E+00</td>
<td>-0.700000E+00</td>
</tr>
<tr>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.150000E+01</td>
<td>-0.912500E+00</td>
<td>0.141875E+00</td>
<td>0.141875E+00</td>
</tr>
<tr>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.150000E+01</td>
<td>-0.912500E+00</td>
<td>0.141875E+00</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
### Table 4.2.4
The Vector $b_a$ of Example 4.2

<table>
<thead>
<tr>
<th>n</th>
<th>$b(n)$</th>
<th>n</th>
<th>$b(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>8</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.40000000E+01</td>
<td>9</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.16000000E+02</td>
<td>10</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.64000000E+02</td>
<td>11</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.00000000E+00</td>
<td>12</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td>13</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.00000000E+00</td>
<td>14</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.00000000E+00</td>
<td>15</td>
<td>0.00000000E+00</td>
</tr>
</tbody>
</table>

### Table 4.2.5
The Vector $y_a$ of Example 4.2

<table>
<thead>
<tr>
<th>n</th>
<th>$y(n)$</th>
<th>n</th>
<th>$y(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>8</td>
<td>0.65536000E+05</td>
</tr>
<tr>
<td>1</td>
<td>0.40000000E+01</td>
<td>9</td>
<td>0.26214400E+06</td>
</tr>
<tr>
<td>2</td>
<td>0.16000000E+02</td>
<td>10</td>
<td>0.10485760E+07</td>
</tr>
<tr>
<td>3</td>
<td>0.64000000E+02</td>
<td>11</td>
<td>0.41943040E+07</td>
</tr>
<tr>
<td>4</td>
<td>0.25600000E+03</td>
<td>12</td>
<td>0.16777216E+08</td>
</tr>
<tr>
<td>5</td>
<td>0.10240000E+04</td>
<td>13</td>
<td>0.67108864E+08</td>
</tr>
<tr>
<td>6</td>
<td>0.40960000E+04</td>
<td>14</td>
<td>0.26843546E+09</td>
</tr>
<tr>
<td>7</td>
<td>0.16384000E+05</td>
<td>15</td>
<td>0.10737418E+10</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 4.2.6

The Relative Errors for Example 4.2

<table>
<thead>
<tr>
<th>n</th>
<th>LU Decomposition Without Pivoting</th>
<th>Estimate of Exact Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00000000E+00</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>1</td>
<td>0.00000000E+00</td>
<td>0.40000000E+01</td>
</tr>
<tr>
<td>2</td>
<td>0.00000000E+00</td>
<td>0.16000000E+02</td>
</tr>
<tr>
<td>3</td>
<td>0.00000000E+00</td>
<td>0.64000000E+02</td>
</tr>
<tr>
<td>4</td>
<td>0.00000000E+00</td>
<td>0.25600000E+03</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td>0.10240000E+04</td>
</tr>
<tr>
<td>6</td>
<td>0.00000000E+00</td>
<td>0.40960000E+04</td>
</tr>
<tr>
<td>7</td>
<td>0.00000000E+00</td>
<td>0.16384000E+05</td>
</tr>
<tr>
<td>8</td>
<td>0.00000000E+00</td>
<td>0.65536000E+05</td>
</tr>
<tr>
<td>9</td>
<td>0.00000000E+00</td>
<td>0.26214400E+06</td>
</tr>
<tr>
<td>10</td>
<td>0.00000000E+00</td>
<td>0.10485760E+07</td>
</tr>
<tr>
<td>11</td>
<td>0.00000000E+00</td>
<td>0.41943030E+07</td>
</tr>
<tr>
<td>12</td>
<td>0.00000000E+00</td>
<td>0.16777216E+08</td>
</tr>
<tr>
<td>13</td>
<td>0.00000000E+00</td>
<td>0.67108864E+08</td>
</tr>
<tr>
<td>14</td>
<td>0.00000000E+00</td>
<td>0.26843546E+09</td>
</tr>
<tr>
<td>15</td>
<td>0.00000000E+00</td>
<td>0.10737418E+10</td>
</tr>
</tbody>
</table>
Augmented vs. Reduced Linear System of Equations

Through theory and experimentation, the choice of using the augmented or reduced associated linear system, showed no significant advantages.

The review of the literature indicates that only the use of the reduced system has been pursued in depth. Therefore, work presented here concerning the augmented system has yet to be investigated in detail.

In the theoretical comparison, the reduced system is functionally equivalent to the augmented system. This means that the same computations must be performed on each system to achieve the solution. It should be noted that in the reduced system some of the computations are done prior to the solution process. These prior computations could cause slight errors or cancellations in the reduced system; whereas, in the augmented system, the original unaltered matrix is provided to the solution process.

For the empirical investigation, a large number of typical examples were run. The results of these examples showed no significant difference in solution. The empirical comparison involved using several solution methods including the LU and QR decompositions. To avoid the possible inconsistencies in the solution methods, only determined systems were used.
The main advantage in the reduced system is the reduced amount of space required to store the matrix. The amount of space saved by the reduced system is approximately equal to \((i + j) \times n\), where \(i\) is the number of initial conditions, \(j\) is the number of terminal conditions, and \(n\) is the number of values to be computed. For higher order recurrence relations, or for the computation of large values of \(n\), this could be a major concern. The lack of flexibility in the reduced system, along with the computations that are performed prior to the solution process, could be termed as disadvantages.

The possible flexibility within the augmented system allows for the weighting of initial conditions. This idea will be covered later in this chapter. The lack of prior computations and the ability to easily modify the matrix, can more than offset the higher space requirements needed for storage.

To investigate this conclusion, a typical example will be presented. In this example (4.3), a third order recurrence relation will be observed in both its augmented and reduced linear system forms. Both systems will be solved to show that the results are very close.
For the comparison of the two systems, the LU decomposition and the QR decomposition will be observed. Furthermore, for the LU and QR decompositions the options of pivoting will be included. This means that four solution methods will be included in the final comparison.

Example 4.3

The following third order recurrence relation (4.3.1) will be used to compare the augmented and reduced forms of the associated linear system of equations.

(4.3.1)
\[ y(n) = 11.3y(n-1) - 12.23y(n-2) + 1.111y(n-3), \]
for \( n = 0, 1, \ldots, 15 \), with a basis of solutions:

(4.3.2)
\[ \{ .1^n, 1.1^n, 10.1^n \}. \]

Table 4.3.3 contains the matrix \( A \) before the inclusion of initial or terminal conditions. This matrix is the basic matrix from which the augmented and reduced matrices are formed (2.27.2). The \( y \) and \( b \) vectors are zero vectors and therefore need not be presented.

For this example the solution \( 1.1^n \) will be computed. Two initial conditions and one terminal condition will be provided.
Table 4.3.4 contains the augmented form of the associated matrix, \( A_a \). Table 4.3.5 contains the reduced form of the associate matrix, \( A_r \). Table 4.3.6 contains the vector \( b_a \) and table 4.3.7 contains the vector \( b_r \). Tables 4.3.4 through 4.3.7 reflect the inclusion of the initial and terminal conditions.

The solution of the recurrence relation is performed using the following methods: LU decomposition with no pivoting, LU decomposition with pivoting, QR decomposition with no pivoting, and the QR decomposition with pivoting. The computed using the augmented form are in table 4.3.8; whereas, the computed results for the reduced form are in table 4.3.9.

Finally, a comparison of the computed solutions will be given. The comparisons will show the relative errors of the computed values to the actual values. The relative errors for the augmented form can be found in table 4.3.10, while the relative errors for the reduced form can be found in table 4.3.11. 

The computed condition numbers for example 4.3 are as follows: the augmented matrix and LU decomposition 4.9150440E+02, the augmented matrix and QR decomposition 5.8305399E+01, the reduced matrix and LU decomposition 4.6429566E+01, the reduced matrix and QR decomposition 1.2817602E+01.
Table 4.3.4

The Augmented Matrix $A_a$ of Example 4.3
(left half on top, right half on bottom)

\[
\begin{array}{ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
Table 4.3.5

The Reduced Matrix $A_r$ of Example 4.3

(left half on top, right half on bottom)
### Table 4.3.6

The Vector $b_a$ of Example 4.3

<table>
<thead>
<tr>
<th>$n$</th>
<th>$b(n)$</th>
<th>$n$</th>
<th>$b(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>8</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.11000000E+01</td>
<td>9</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.00000000E+00</td>
<td>10</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.00000000E+00</td>
<td>11</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.00000000E+00</td>
<td>12</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td>13</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.00000000E+00</td>
<td>14</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.00000000E+00</td>
<td>15</td>
<td>0.41772482E+01</td>
</tr>
</tbody>
</table>

### Table 4.3.7

The Vector $b_r$ of Example 4.3

<table>
<thead>
<tr>
<th>$n$</th>
<th>$b(n)$</th>
<th>$n$</th>
<th>$b(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-0.12342000E+02</td>
<td>8</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.12221000E+01</td>
<td>9</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.00000000E+00</td>
<td>10</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.00000000E+00</td>
<td>11</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.00000000E+00</td>
<td>12</td>
<td>0.41772482E+01</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.00000000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.00000000E+00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4.3.8
The Computed Values of Example 4.3 Using the Augmented Matrix

<table>
<thead>
<tr>
<th>n</th>
<th>IU Decomposition</th>
<th>IU Decomposition</th>
<th>QR Decomposition</th>
<th>QR Decomposition</th>
<th>Estimate of Exact Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Without Pivoting</td>
<td>With Pivoting</td>
<td>Without Pivoting</td>
<td>With Pivoting</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0.10000000E+00</td>
<td>0.10000000E+00</td>
<td>0.99999973E+00</td>
<td>0.99999973E+00</td>
<td>0.10000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.11000000E+00</td>
<td>0.11000000E+00</td>
<td>0.11000000E+00</td>
<td>0.11000000E+00</td>
<td>0.11000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.12100000E+00</td>
<td>0.12100000E+00</td>
<td>0.12100000E+00</td>
<td>0.12100000E+00</td>
<td>0.12100000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.13310000E+00</td>
<td>0.13310000E+00</td>
<td>0.13310000E+00</td>
<td>0.13310000E+00</td>
<td>0.13310000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.14641000E+00</td>
<td>0.14641000E+00</td>
<td>0.14641000E+00</td>
<td>0.14641000E+00</td>
<td>0.14641000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.16105100E+00</td>
<td>0.16105100E+00</td>
<td>0.16105100E+00</td>
<td>0.16105100E+00</td>
<td>0.16105100E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.17715611E+00</td>
<td>0.17715611E+00</td>
<td>0.17715611E+00</td>
<td>0.17715611E+00</td>
<td>0.17715611E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.19457120E+00</td>
<td>0.19457120E+00</td>
<td>0.19457120E+00</td>
<td>0.19457120E+00</td>
<td>0.19457120E+00</td>
</tr>
<tr>
<td>8</td>
<td>0.21435890E+00</td>
<td>0.21435890E+00</td>
<td>0.21435890E+00</td>
<td>0.21435890E+00</td>
<td>0.21435890E+00</td>
</tr>
<tr>
<td>9</td>
<td>0.23579478E+00</td>
<td>0.23579478E+00</td>
<td>0.23579478E+00</td>
<td>0.23579478E+00</td>
<td>0.23579478E+00</td>
</tr>
<tr>
<td>10</td>
<td>0.25837427E+00</td>
<td>0.25837427E+00</td>
<td>0.25837427E+00</td>
<td>0.25837427E+00</td>
<td>0.25837427E+00</td>
</tr>
<tr>
<td>11</td>
<td>0.28531170E+00</td>
<td>0.28531170E+00</td>
<td>0.28531170E+00</td>
<td>0.28531170E+00</td>
<td>0.28531170E+00</td>
</tr>
<tr>
<td>12</td>
<td>0.31384237E+00</td>
<td>0.31384237E+00</td>
<td>0.31384237E+00</td>
<td>0.31384237E+00</td>
<td>0.31384237E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.34527156E+00</td>
<td>0.34527156E+00</td>
<td>0.34527156E+00</td>
<td>0.34527156E+00</td>
<td>0.34527156E+00</td>
</tr>
<tr>
<td>14</td>
<td>0.37974982E+00</td>
<td>0.37974982E+00</td>
<td>0.37974982E+00</td>
<td>0.37974982E+00</td>
<td>0.37974982E+00</td>
</tr>
<tr>
<td>15</td>
<td>0.41772482E+00</td>
<td>0.41772482E+00</td>
<td>0.41772482E+00</td>
<td>0.41772482E+00</td>
<td>0.41772482E+00</td>
</tr>
</tbody>
</table>
Table 4.3.9

The Computed Values of Example 4.3 Using The Reduced Matrix

<table>
<thead>
<tr>
<th>n</th>
<th>IU Decomposition Without Pivoting</th>
<th>IU Decomposition With Pivoting</th>
<th>QR Decomposition Without Pivoting</th>
<th>QR Decomposition With Pivoting</th>
<th>Estimate of Exact Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+00</td>
<td>0.10000000E+00</td>
<td>0.10000000E+00</td>
<td>0.10000000E+00</td>
<td>0.10000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.11000000E+00</td>
<td>0.11000000E+00</td>
<td>0.11000000E+00</td>
<td>0.11000000E+00</td>
<td>0.11000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.12100000E+00</td>
<td>0.12100000E+00</td>
<td>0.12100000E+00</td>
<td>0.12100000E+00</td>
<td>0.12100000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.13310000E+00</td>
<td>0.13310000E+00</td>
<td>0.13310000E+00</td>
<td>0.13310000E+00</td>
<td>0.13310000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.14610000E+00</td>
<td>0.14610000E+00</td>
<td>0.14640000E+00</td>
<td>0.14641000E+00</td>
<td>0.14641000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.16150000E+00</td>
<td>0.16150000E+00</td>
<td>0.16150000E+00</td>
<td>0.16150000E+00</td>
<td>0.16150000E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.17150000E+00</td>
<td>0.17150000E+00</td>
<td>0.17150000E+00</td>
<td>0.17150000E+00</td>
<td>0.17150000E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.19480000E+00</td>
<td>0.19480000E+00</td>
<td>0.19480000E+00</td>
<td>0.19480000E+00</td>
<td>0.19480000E+00</td>
</tr>
<tr>
<td>8</td>
<td>0.21430000E+00</td>
<td>0.21430000E+00</td>
<td>0.21430000E+00</td>
<td>0.21430000E+00</td>
<td>0.21430000E+00</td>
</tr>
<tr>
<td>9</td>
<td>0.23570000E+00</td>
<td>0.23570000E+00</td>
<td>0.23570000E+00</td>
<td>0.23570000E+00</td>
<td>0.23570000E+00</td>
</tr>
<tr>
<td>10</td>
<td>0.25930000E+00</td>
<td>0.25930000E+00</td>
<td>0.25930000E+00</td>
<td>0.25930000E+00</td>
<td>0.25930000E+00</td>
</tr>
<tr>
<td>11</td>
<td>0.28530000E+00</td>
<td>0.28530000E+00</td>
<td>0.28530000E+00</td>
<td>0.28530000E+00</td>
<td>0.28530000E+00</td>
</tr>
<tr>
<td>12</td>
<td>0.31360000E+00</td>
<td>0.31360000E+00</td>
<td>0.31360000E+00</td>
<td>0.31360000E+00</td>
<td>0.31360000E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.34520000E+00</td>
<td>0.34520000E+00</td>
<td>0.34520000E+00</td>
<td>0.34520000E+00</td>
<td>0.34520000E+00</td>
</tr>
<tr>
<td>14</td>
<td>0.37970000E+00</td>
<td>0.37970000E+00</td>
<td>0.37970000E+00</td>
<td>0.37970000E+00</td>
<td>0.37970000E+00</td>
</tr>
<tr>
<td>15</td>
<td>0.41770000E+00</td>
<td>0.41770000E+00</td>
<td>0.41770000E+00</td>
<td>0.41770000E+00</td>
<td>0.41770000E+00</td>
</tr>
</tbody>
</table>
### Table 4.3.10

The Relative Errors of Example 4.3 Using the Augmented Matrix

<table>
<thead>
<tr>
<th>n</th>
<th>IU Decomposition</th>
<th>QR Decomposition</th>
<th>Estimate of Exact Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Without Pivoting</td>
<td>Without Pivoting</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0.00000000E+00</td>
<td>0.59604645E-07</td>
<td>0.55134298E-06</td>
</tr>
<tr>
<td>1</td>
<td>0.00000000E+00</td>
<td>0.13546510E-07</td>
<td>0.3866275E-06</td>
</tr>
<tr>
<td>2</td>
<td>0.12315009E-07</td>
<td>0.24630019E-07</td>
<td>0.27093003E-06</td>
</tr>
<tr>
<td>3</td>
<td>0.3356389E-07</td>
<td>0.44781852E-07</td>
<td>0.23510473E-06</td>
</tr>
<tr>
<td>4</td>
<td>0.305300E-07</td>
<td>0.50884E-07</td>
<td>0.14641000E+01</td>
</tr>
<tr>
<td>5</td>
<td>0.3700975E-07</td>
<td>0.55514639E-07</td>
<td>0.18430143E-06</td>
</tr>
<tr>
<td>6</td>
<td>0.67290537E-07</td>
<td>0.67290537E-07</td>
<td>0.17715610E+01</td>
</tr>
<tr>
<td>7</td>
<td>0.84113172E-07</td>
<td>0.84113172E-07</td>
<td>0.1916630E-06</td>
</tr>
<tr>
<td>8</td>
<td>0.83418021E-07</td>
<td>0.83418021E-07</td>
<td>0.2054500E-06</td>
</tr>
<tr>
<td>9</td>
<td>0.1011275E-06</td>
<td>0.1011275E-06</td>
<td>0.2002251E-06</td>
</tr>
<tr>
<td>10</td>
<td>0.1149009E-06</td>
<td>0.1149009E-06</td>
<td>0.2537424E-06</td>
</tr>
<tr>
<td>11</td>
<td>0.1149009E-06</td>
<td>0.12543639E-06</td>
<td>0.2853116E+01</td>
</tr>
<tr>
<td>12</td>
<td>0.10455325E-06</td>
<td>0.12344720E-06</td>
<td>0.3139423E+01</td>
</tr>
<tr>
<td>13</td>
<td>0.10389206E-06</td>
<td>0.13812274E-06</td>
<td>0.3422711E+01</td>
</tr>
<tr>
<td>14</td>
<td>0.78478331E-07</td>
<td>0.11718525E-06</td>
<td>0.3774833E+01</td>
</tr>
<tr>
<td>15</td>
<td>0.42806638E-07</td>
<td>0.42806638E-07</td>
<td>0.4177248E+01</td>
</tr>
<tr>
<td>n</td>
<td>IU Decomposition Without Pivoting</td>
<td>IU Decomposition With Pivoting</td>
<td>QR Decomposition Without Pivoting</td>
</tr>
<tr>
<td>----</td>
<td>----------------------------------</td>
<td>--------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>0</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.123159636E+07</td>
<td>0.24630191B+07</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.338866736E-07</td>
<td>0.44781858E-07</td>
<td>0.44781858E-07</td>
</tr>
<tr>
<td>4</td>
<td>0.30833318E-07</td>
<td>0.61066162E-07</td>
<td>0.81421590E-07</td>
</tr>
<tr>
<td>5</td>
<td>0.37003796E-07</td>
<td>0.74016509E-07</td>
<td>0.64767142E-07</td>
</tr>
<tr>
<td>6</td>
<td>0.67290578E-07</td>
<td>0.84111372E-07</td>
<td>0.42056895E-07</td>
</tr>
<tr>
<td>7</td>
<td>0.84113172E-07</td>
<td>0.11469798E-06</td>
<td>0.15293034E-07</td>
</tr>
<tr>
<td>8</td>
<td>0.84113172E-07</td>
<td>0.11122403E-06</td>
<td>0.27901007E-07</td>
</tr>
<tr>
<td>9</td>
<td>0.10112758E-06</td>
<td>0.11375185E-06</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>10</td>
<td>0.11490008E-06</td>
<td>0.13783103E-06</td>
<td>0.11490008E-07</td>
</tr>
<tr>
<td>11</td>
<td>0.11490008E-06</td>
<td>0.14623745E-06</td>
<td>0.31368979E-07</td>
</tr>
<tr>
<td>12</td>
<td>0.10412532E-06</td>
<td>0.15193502E-06</td>
<td>0.94593338E-08</td>
</tr>
<tr>
<td>13</td>
<td>0.10352006E-06</td>
<td>0.16402076E-06</td>
<td>0.17253439E-07</td>
</tr>
<tr>
<td>14</td>
<td>0.78478331E-07</td>
<td>0.14126190E-06</td>
<td>0.15693769E-07</td>
</tr>
<tr>
<td>15</td>
<td>0.42806663E-07</td>
<td>0.42806663E-07</td>
<td>0.42806663E-07</td>
</tr>
</tbody>
</table>
It should be noticed that all of the condition numbers are relatively small. This would be a good indication that the results are reliable. The condition number for a particular form of associated matrix is independent of the decomposition performed. The reason for the difference in condition numbers for example 4.3 is due to the method that the LINPACK routines chose to compute the condition number.

Example 4.3 is to be considered typical. The results show that both forms of the associated linear system provide accurate results. As before, it is possible to contrive examples to show any conclusion desired, but for the large sample set used for this work, the stated conclusion was supported.

LU vs. QR Decomposition

Through theory and experimentation, the LU decomposition has been shown to produce better results for typical problems than the QR decomposition.

In the theoretical comparison, it was obvious that the QR decomposition was performing more computations to achieve the decomposition. This increase in computations could lead to the problem of roundoff error at any step. It was observed that as soon as this roundoff error was introduced, it would affect further computations.
The empirical investigation involved a large range of typical problems. The use of determined systems was chosen for all examples. The examples included the reduced and augmented forms of the associated linear systems. The choice of examples also included the option of letting the decomposition perform pivoting to achieve its results.

The empirical investigation supported the initial claim in that the LU decomposition performed better for most typical examples then did the QR decomposition. It should be noted that it was possible to contrive examples that would show that the QR decomposition performed better than the LU decomposition.

Most of the examples from the test set would have supported the conclusion. Table 4.3.9, which is the relative error comparison for example 4.3, shows that the LU decomposition method is about 1 digit more accurate. This result can be considered typical. In some of the examples this difference in accuracy was greater and in others very small. Some of the test examples displayed the QR decomposition method slightly better, but still extremely close.
Pivoting vs. Not Pivoting in the Context of the LU Decomposition

Through theory and experimentation, it has been shown that the LU decomposition without pivoting provides better results than the LU decomposition with pivoting, in most typical problems; however, there are cases where pivoting is necessary to avoid problems.

A review of the literature showed most prior investigations used the LU decomposition without pivoting. The LU decomposition without pivoting is the only method for which a complete error analysis is known (Lozier 1980; Sadowski and Lozier 1972; Oliver 1967, 1968a, 1968b; Van der Cruyssen 1979).

The theoretical investigation involved looking at the original matrix, before decomposition, and observing what effect pivoting had on the computations.

The matrix representation is a banded system. Furthermore, the band is of width \( p + 1 \), where \( p \) is the order of the recurrence relation. If pivoting is allowed to occur, and a lower row is pivoted to a higher level, it causes more computations to occur. This process of pivoting a lower row to a higher position has the effect of widening the band. The increased computation might introduce roundoff errors, or cancellations, that would not have shown up in the if pivoting were not performed.
The need for pivoting in certain situations also became apparent. If one of the coefficients of the characteristic equation was zero, then the decomposition without pivoting will fail for some choice of the number of initial and terminal conditions.

The empirical investigation into the concept of pivoting involved running many typical examples. The use of determined systems and either the reduced or augmented forms for the matrices provided the source of examples. The results from the examples supported the initial conclusion. It was possible to construct an example that would cause errors if pivoting was not performed.

To use the advantages of pivoting and the inclusion of initial conditions, an enhancement was proposed. The enhancement weights the initial conditions so that they would not pivot and possibly cause roundoff errors or cancellations in later computations. This simply involved multiplying the 1's, usually placed in the augmented matrix, and the associated initial value by a constant. This constant should be larger in magnitude than any of the coefficients in the matrix. A discussion of this enhancement will be found later in this chapter.
The literature reviewed provides many ideas of the method for choosing the pivot values. Most of the work in this area has been in the decompositions that utilize Gaussian elimination techniques (Cohen 1974; Cryer 1968; Dax and Kaniel 1977; Skeel 1981).

For a better understanding of the concept of pivoting, two examples will be presented. The first example (4.4) will show the need to perform pivoting. The solution of the recurrence relation (4.4.1) is not computable, without the option of pivoting, because the second coefficient of the characteristic equation is 0.

In the second example (4.5), a typical problem will be presented. This problem will show the LU decomposition without pivoting producing better results than the LU decompositions with pivoting.

Example 4.4

The following third order recurrence relation (4.4.1) will be used to compare the LU decomposition with pivoting and without pivoting. This example has a 0 for the second coefficient of the characteristic equation and will fail if pivoting is not performed.

(4.4.1)

\[ y(n) = 7y(n-2) - 6y(n-3), \]

for \( n = 0, 1, \ldots, 15 \), with a basis of solutions:
\begin{equation} \{ 1^n, 2^n, -3^n \}. \end{equation}

Notice there is no \( y(n-1) \) term due to the second coefficient of the characteristic equation being 0.

For this example the calculation of the intermediate solution \( z^n \) will be computed. Two initial conditions and one terminal condition will be provided.

Table 4.4.3 shows the augmented form of the matrix for recurrence relation 4.4.1 with the initial and terminal conditions included. Table 4.4.4 contains the result vector \( b \) including the initial and terminal conditions.

Notice that the matrix entry \( A(3,3) \) is 0. This is the problem for the LU decomposition without pivoting. When the LU decomposition without pivoting is allowed to decompose this matrix, it will try to divide the third column's values by the element in \( A(3,3) \), which will lead to undefined results.

Table 4.4.5 contains the computed results of the solution to the recurrence relation (4.4.1). Two methods were used for the computations in this problem: the LU decomposition without pivoting, and the LU decomposition with weighted initial conditions and pivoting. The relative error comparison for the solutions is presented in table 4.4.6.
Table 4.4.3

The Augmented Matrix $A_4$ of Example 4.4
(left half on top, right half on bottom)

\[
\begin{array}{cccccccc}
0.10000000E+01 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 \\
0.00000000E+00 & 0.10000000E+01 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 \\
0.00000000E+00 & -0.70000000E+01 & 0.00000000E+00 & 0.10000000E+01 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 \\
0.00000000E+00 & 0.60000000E+00 & 0.70000000E+01 & 0.20000000E+00 & 0.10000000E+01 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 \\
0.00000000E+00 & 0.60000000E+00 & 0.70000000E+01 & 0.20000000E+00 & 0.10000000E+01 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 \\
0.00000000E+00 & 0.60000000E+00 & 0.70000000E+01 & 0.20000000E+00 & 0.10000000E+01 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 \\
0.00000000E+00 & 0.60000000E+00 & 0.70000000E+01 & 0.20000000E+00 & 0.10000000E+01 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 \\
0.00000000E+00 & 0.60000000E+00 & 0.70000000E+01 & 0.20000000E+00 & 0.10000000E+01 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 \\
\end{array}
\]
Table 4.4.4

The Vector $b_a$ of Example 4.4

<table>
<thead>
<tr>
<th>n</th>
<th>$b(n)$</th>
<th>n</th>
<th>$b(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>8</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.20000000E+01</td>
<td>9</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.00000000E+00</td>
<td>10</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.00000000E+00</td>
<td>11</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.00000000E+00</td>
<td>12</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td>13</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.00000000E+00</td>
<td>14</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.00000000E+00</td>
<td>15</td>
<td>0.32768000E+05</td>
</tr>
</tbody>
</table>

Table 4.5.4

The Vector $b_a$ of Example 4.5

<table>
<thead>
<tr>
<th>n</th>
<th>$b(n)$</th>
<th>n</th>
<th>$b(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>8</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.21000000E+01</td>
<td>9</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.44100000E+01</td>
<td>10</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.92610000E+01</td>
<td>11</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.00000000E+00</td>
<td>12</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td>13</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.00000000E+00</td>
<td>14</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.00000000E+00</td>
<td>15</td>
<td>0.00000000E+00</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 4.4.5

The Computed Values of Example 4.4

<table>
<thead>
<tr>
<th>n</th>
<th>LU Decomposition Without Pivoting</th>
<th>LU Decomposition Weighted Initials</th>
<th>Estimate of Exact Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0.10000000E+01)</td>
<td>(0.10000000E+01)</td>
<td>(0.10000000E+01)</td>
</tr>
<tr>
<td>1</td>
<td>(0.20000000E+01)</td>
<td>(0.20000000E+01)</td>
<td>(0.20000000E+01)</td>
</tr>
<tr>
<td>2</td>
<td>(0.17014118E+39)</td>
<td>(0.39999999E+01)</td>
<td>(0.40000000E+01)</td>
</tr>
<tr>
<td>3</td>
<td>(-0.17014118E+39)</td>
<td>(0.79999997E+01)</td>
<td>(0.80000000E+01)</td>
</tr>
<tr>
<td>4</td>
<td>(0.17014118E+39)</td>
<td>(0.15999999E+02)</td>
<td>(0.16000000E+02)</td>
</tr>
<tr>
<td>5</td>
<td>(-0.17014118E+39)</td>
<td>(0.31999998E+02)</td>
<td>(0.32000000E+02)</td>
</tr>
<tr>
<td>6</td>
<td>(0.17014118E+39)</td>
<td>(0.63999996E+02)</td>
<td>(0.64000000E+02)</td>
</tr>
<tr>
<td>7</td>
<td>(-0.17014118E+39)</td>
<td>(0.12799999E+03)</td>
<td>(0.12800000E+03)</td>
</tr>
<tr>
<td>8</td>
<td>(0.17014118E+39)</td>
<td>(0.25599998E+03)</td>
<td>(0.25600000E+03)</td>
</tr>
<tr>
<td>9</td>
<td>(-0.17014118E+39)</td>
<td>(0.51199997E+03)</td>
<td>(0.51200000E+03)</td>
</tr>
<tr>
<td>10</td>
<td>(0.17014118E+39)</td>
<td>(0.10239999E+04)</td>
<td>(0.10240000E+04)</td>
</tr>
<tr>
<td>11</td>
<td>(-0.17014118E+39)</td>
<td>(0.20479999E+04)</td>
<td>(0.20480000E+04)</td>
</tr>
<tr>
<td>12</td>
<td>(0.17014118E+39)</td>
<td>(0.40959996E+04)</td>
<td>(0.40960000E+04)</td>
</tr>
<tr>
<td>13</td>
<td>(-0.17014118E+39)</td>
<td>(0.81919996E+04)</td>
<td>(0.81920000E+04)</td>
</tr>
<tr>
<td>14</td>
<td>(0.17014118E+39)</td>
<td>(0.16383998E+05)</td>
<td>(0.16384000E+05)</td>
</tr>
<tr>
<td>15</td>
<td>(0.32768000E+05)</td>
<td>(0.32768000E+05)</td>
<td>(0.32768000E+05)</td>
</tr>
<tr>
<td>n</td>
<td>LU Decomposition Without Pivoting</td>
<td>LU Decomposition Weighted Initials</td>
<td>Estimate of Exact Value</td>
</tr>
<tr>
<td>----</td>
<td>----------------------------------</td>
<td>-----------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>0</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>1</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.20000000E+01</td>
</tr>
<tr>
<td>2</td>
<td>0.42535296E+38</td>
<td>0.22351742E-07</td>
<td>0.40000000E+01</td>
</tr>
<tr>
<td>3</td>
<td>0.21267648E+38</td>
<td>0.37252903E-07</td>
<td>0.80000000E+01</td>
</tr>
<tr>
<td>4</td>
<td>0.10633824E+38</td>
<td>0.44703484E-07</td>
<td>0.16000000E+02</td>
</tr>
<tr>
<td>5</td>
<td>0.53169119E+37</td>
<td>0.59604645E-07</td>
<td>0.32000000E+02</td>
</tr>
<tr>
<td>6</td>
<td>0.26584560E+37</td>
<td>0.59604645E-07</td>
<td>0.64000000E+02</td>
</tr>
<tr>
<td>7</td>
<td>0.13292280E+37</td>
<td>0.59604645E-07</td>
<td>0.12800000E+03</td>
</tr>
<tr>
<td>8</td>
<td>0.66461399E+36</td>
<td>0.81956387E-07</td>
<td>0.25600000E+03</td>
</tr>
<tr>
<td>9</td>
<td>0.33230700E+36</td>
<td>0.59604645E-07</td>
<td>0.51200000E+03</td>
</tr>
<tr>
<td>10</td>
<td>0.16615350E+36</td>
<td>0.81956387E-07</td>
<td>0.10240000E+04</td>
</tr>
<tr>
<td>11</td>
<td>0.83076749E+35</td>
<td>0.59604645E-07</td>
<td>0.20480000E+04</td>
</tr>
<tr>
<td>12</td>
<td>0.41538375E+35</td>
<td>0.96857548E-07</td>
<td>0.40960000E+04</td>
</tr>
<tr>
<td>13</td>
<td>0.20769187E+35</td>
<td>0.52154064E-07</td>
<td>0.81920000E+04</td>
</tr>
<tr>
<td>14</td>
<td>0.10384594E+35</td>
<td>0.12665987E-06</td>
<td>0.16384000E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.32768000E+05</td>
</tr>
</tbody>
</table>
The computed condition number for the associated matrix of example 4.4 is 1.0517311E+05. This value for a condition number would be an indication for acceptable results. For this example, the condition number provides no evidence to indicate the severe problem.

The results for the LU decomposition without pivoting provided extremely bad results even though the condition number for the matrix is acceptable.

In example 4.4 it was shown that in certain situations the need to pivot is mandatory. In example 4.5, somewhat of a contrast will be presented. For most examples in the sample set the LU decomposition without pivoting provided more accurate results. Example 4.5 shows a fourth order recurrence relation that supports this claim.

Example 4.5

The following fourth order recurrence relation (4.5.1) will be used to compare pivoting and not pivoting in the LU decomposition.

(4.5.1)
\[ y(n) = 4.8y(n-1) - 8.07y(n-2) + 5.74y(n-3) - 1.47y(n-4), \]

for \( n = 0, 1, \ldots, 15 \), with a basis of solutions:

(4.5.2)
\[ \{ .7^n, 1^n, n1^n, 2.1^n \}. \]
For this example the calculation of the dominate solution $2.1^n$ will be computed. Four initial conditions and no terminal condition will be provided.

Table 4.5.3 shows the augmented form of the matrix for recurrence relation 4.5.1 with the initial conditions included. Table 4.5.4 contains the result vector $b$ including the initial conditions.

Table 4.5.5 contains the computed results of the solution to the recurrence relation (4.5.1). Three methods were used for the computations in this problem: the LU decomposition without pivoting, the LU decomposition with pivoting, and the LU decomposition with weighted initial conditions and pivoting. The relative errors of the computed results as compared to the exact values is provided in table 4.5.6.

The computed condition number for the unweighted matrix was $2.8364439E+06$; whereas, the condition number for the weighted matrix was $3.1027946E+06$. These values provide little evidence to support the results.

The LU decomposition without pivoting provided excellent results; whereas, the LU decomposition with pivoting had poor results. The LU decomposition with weighted initial conditions and pivoting fell somewhere between. The overhead of pivoting introduced errors into the solution that were propagated throughout.
Table 4.5.5
The Computed Values of Example 4.5

<table>
<thead>
<tr>
<th>n</th>
<th>IU Decomposition Without Pivoting</th>
<th>IU Decomposition With Pivoting</th>
<th>IU Decomposition Weighted initials</th>
<th>Estimate of Exact Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.100000000E+01</td>
<td>0.999999941E+00</td>
<td>0.100000000E+01</td>
<td>0.100000000E+01</td>
</tr>
<tr>
<td>1</td>
<td>0.210000000E+01</td>
<td>0.209998441E+01</td>
<td>0.210000000E+01</td>
<td>0.210000000E+01</td>
</tr>
<tr>
<td>2</td>
<td>0.441000000E+01</td>
<td>0.441002691E+01</td>
<td>0.441000000E+01</td>
<td>0.441000000E+01</td>
</tr>
<tr>
<td>3</td>
<td>0.926100000E+01</td>
<td>0.926182491E+01</td>
<td>0.926099629E+01</td>
<td>0.926099999E+01</td>
</tr>
<tr>
<td>4</td>
<td>0.194481000E+02</td>
<td>0.194514526E+02</td>
<td>0.194478476E+02</td>
<td>0.194481000E+02</td>
</tr>
<tr>
<td>5</td>
<td>0.408410000E+02</td>
<td>0.408405650E+02</td>
<td>0.408400558E+02</td>
<td>0.408410000E+02</td>
</tr>
<tr>
<td>6</td>
<td>0.857661172E+02</td>
<td>0.857634662E+02</td>
<td>0.857634462E+02</td>
<td>0.857661198E+02</td>
</tr>
<tr>
<td>7</td>
<td>0.160109841E+03</td>
<td>0.160163491E+03</td>
<td>0.160102346E+03</td>
<td>0.160108658E+03</td>
</tr>
<tr>
<td>8</td>
<td>0.378228568E+03</td>
<td>0.378397863E+03</td>
<td>0.378213793E+03</td>
<td>0.378228583E+03</td>
</tr>
<tr>
<td>9</td>
<td>0.794279959E+03</td>
<td>0.794247625E+03</td>
<td>0.794247625E+03</td>
<td>0.794280021E+03</td>
</tr>
<tr>
<td>10</td>
<td>0.166798791E+04</td>
<td>0.166798791E+04</td>
<td>0.166798791E+04</td>
<td>0.166798800E+04</td>
</tr>
<tr>
<td>11</td>
<td>0.350277458E+04</td>
<td>0.350297272E+04</td>
<td>0.350297272E+04</td>
<td>0.350277489E+04</td>
</tr>
<tr>
<td>12</td>
<td>0.735922614E+04</td>
<td>0.735889460E+04</td>
<td>0.735889460E+04</td>
<td>0.735926270E+04</td>
</tr>
<tr>
<td>13</td>
<td>0.154472378E+05</td>
<td>0.154465789E+05</td>
<td>0.154465789E+05</td>
<td>0.154472378E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.324391976E+05</td>
<td>0.324378128E+05</td>
<td>0.324378128E+05</td>
<td>0.324391976E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.681223042E+05</td>
<td>0.681194012E+05</td>
<td>0.681194012E+05</td>
<td>0.681223135E+05</td>
</tr>
</tbody>
</table>
Table 4.5.6
The Relative Errors of Example 4.5

<table>
<thead>
<tr>
<th>n</th>
<th>IU Decomposition Without Pivoting</th>
<th>IU Decomposition With Pivoting</th>
<th>IU Decomposition Weighted initials</th>
<th>Estimate of Exact Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00000000E+00</td>
<td>0.59604645E-07</td>
<td>0.00000000E+00</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>1</td>
<td>0.00000000E+00</td>
<td>0.48379103E-04</td>
<td>0.00000000E+00</td>
<td>0.21000000E+01</td>
</tr>
<tr>
<td>2</td>
<td>0.13515743E-07</td>
<td>0.59469487E-06</td>
<td>0.47849059E-05</td>
<td>0.44100000E+01</td>
</tr>
<tr>
<td>3</td>
<td>0.12872183E-07</td>
<td>0.89011458E-04</td>
<td>0.18535944E-05</td>
<td>0.92609999E+01</td>
</tr>
<tr>
<td>4</td>
<td>0.12269229E-07</td>
<td>0.17238919E-03</td>
<td>0.12982517E-04</td>
<td>0.19448100E+02</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td>0.22515524E-03</td>
<td>0.23374251E-04</td>
<td>0.40841000E+02</td>
</tr>
<tr>
<td>6</td>
<td>0.22238952E-07</td>
<td>0.27718630E-03</td>
<td>0.31167892E-04</td>
<td>0.85766195E+02</td>
</tr>
<tr>
<td>7</td>
<td>0.31769932E-07</td>
<td>0.30337105E-03</td>
<td>0.35154187E-04</td>
<td>0.18010892E+03</td>
</tr>
<tr>
<td>8</td>
<td>0.40427712E-07</td>
<td>0.32048291E-03</td>
<td>0.39102311E-04</td>
<td>0.37622832E+03</td>
</tr>
<tr>
<td>9</td>
<td>0.67237953E-07</td>
<td>0.33137056E-03</td>
<td>0.40775016E-04</td>
<td>0.79428001E+03</td>
</tr>
<tr>
<td>10</td>
<td>0.91480208E-07</td>
<td>0.33803569E-03</td>
<td>0.41708378E-04</td>
<td>0.16679803E+04</td>
</tr>
<tr>
<td>11</td>
<td>0.95386409E-07</td>
<td>0.34199635E-03</td>
<td>0.42202870E-04</td>
<td>0.35027748E+04</td>
</tr>
<tr>
<td>12</td>
<td>0.82978247E-07</td>
<td>0.34427260E-03</td>
<td>0.42433327E-04</td>
<td>0.73258270E+04</td>
</tr>
<tr>
<td>13</td>
<td>0.94628853E-07</td>
<td>0.34554053E-03</td>
<td>0.42641374E-04</td>
<td>0.15447237E+05</td>
</tr>
<tr>
<td>14</td>
<td>0.11289148E-06</td>
<td>0.34622315E-03</td>
<td>0.42710615E-04</td>
<td>0.32439197E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.14335428E-06</td>
<td>0.34655897E-03</td>
<td>0.42748246E-04</td>
<td>0.63122313E+05</td>
</tr>
</tbody>
</table>
Space and Time Comparisons

This section presents the facts and formulas for the space and timing calculations of the different decomposition and solution methods for linear systems. The LU and QR decompositions, as used throughout this work, along with observations for the LU decomposition of a banded system, will be included. The observations of the characteristics for the banded system using the LU decomposition will be provided for comparison only.

The LINPACK routines (Dongarra et al. 1978) to decompose and solve banded linear systems of equations are not used in this work, but could be an area of further investigation. The banded system routines make use of the fact that a banded system requires significantly less storage than the full matrix. The fact that the banded system usually leads to a sparse matrix, means that conventional solution techniques will not perform efficiently. The savings in time for the banded system is due to the fact that no computations need be performed on elements outside of the band.

For ease of comparison, the formulas will only compute the time and space required for determined systems. The formulas could be altered to accommodate non-determined systems. The variable n, in tables 4.6 and 4.7, represents the number of values to compute.
using the recurrence relation. In the space comparison, table 4.7, the variable \( w \) is the width of the band for the banded system. The value of \( w \) should be equal to \( p+1 \), in most applications.

For the timing comparisons, \( u \) will represent the time that a computer requires to execute the following instruction:

\[
Y(I) = Y(I) + T \times X(I).
\]

This instruction involves a floating point multiplication, a floating point add, three array index calculations, and some floating point assignments. Table 4.6 will present the timing comparisons for the LINPACK routines (Dongarra et al. 1978) used as part of this work. Also included are the timings for the routines to compute the LU decomposition of a banded system.

These values and formulas are from the LINPACK users guide (Dongarra et al. 1978). If further information is desired, this manual should be consulted.

Table 4.7 will give an estimate to the space required by each routine. These formulas are only estimates of the space required for a particular routine. The formula includes both the space needed for parameters, and local variables needed within the routine. Some of the routines contain parameters that
are not accessed in certain situations; therefore, the formulas would be considered as upper bounds on the space required.

In the literature, when the decomposition of a banded system is discussed, the LU decomposition is used (Dongarra et al. 1978; Martin and Wilkinson 1965, 1967).

Table 4.6

Timing Concerns

<table>
<thead>
<tr>
<th>Routine</th>
<th>Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>LU</td>
<td></td>
</tr>
<tr>
<td>SGECO</td>
<td>((\frac{1}{3}n^3 + 3n^2)u)</td>
</tr>
<tr>
<td>SGEFA</td>
<td>((\frac{1}{3}n^3)u)</td>
</tr>
<tr>
<td>SGESL</td>
<td>((n^2)u)</td>
</tr>
<tr>
<td>QR</td>
<td></td>
</tr>
<tr>
<td>SQRDC</td>
<td>((n^3 - n^3/3)u)</td>
</tr>
<tr>
<td>SQRSL</td>
<td>(k = \text{MIN}(n,p), (k(2n - k))u)</td>
</tr>
<tr>
<td>LU Banded</td>
<td></td>
</tr>
<tr>
<td>SGBCO</td>
<td>(((\frac{1}{6}n^2 + 3n)(p + 1))u)</td>
</tr>
<tr>
<td>SGBFA</td>
<td>((\frac{1}{6}n^2(p + 1))u)</td>
</tr>
<tr>
<td>SGBSL</td>
<td>((n(p + 1))u)</td>
</tr>
</tbody>
</table>

LINPACK does not include specific routines for the QR decomposition and solution of banded systems. This is partially due to the fact that the QR decomposition may
require more space to store the decomposition than the banded representation provides. It can be shown that if the QR decomposition without column pivoting is performed, the results could be stored and manipulated in a banded representation. Thus, if a non-pivoted QR decomposition's results would be acceptable, say for a least squares calculation, then the large savings in space and time afforded by the banded LU decomposition would be somewhat offset.

Table 4.7

Space Concerns

<table>
<thead>
<tr>
<th>Routine</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>LU</td>
<td></td>
</tr>
<tr>
<td>SGECO</td>
<td>$n^2 + 2n + 17$</td>
</tr>
<tr>
<td>SGEFA</td>
<td>$n^2 + n + 9$</td>
</tr>
<tr>
<td>SGESL</td>
<td>$n^2 + 2n + 8$</td>
</tr>
<tr>
<td>QR</td>
<td></td>
</tr>
<tr>
<td>SQRDC</td>
<td>$n^2 + 3n + 18$</td>
</tr>
<tr>
<td>SQRSL</td>
<td>$n^2 + 7n + 17$</td>
</tr>
<tr>
<td>LU Banded</td>
<td></td>
</tr>
<tr>
<td>SGBCO</td>
<td>$(w + 1)n + 2n + 26$</td>
</tr>
<tr>
<td>SGBFA</td>
<td>$(w + 1)n + n + 20$</td>
</tr>
<tr>
<td>SGBSSL</td>
<td>$(w + 1)n + 2n + 14$</td>
</tr>
</tbody>
</table>
There has been much interest in computations with sparse matrices (Duff 1977; Rose and Willoughby 1972; Willoughby 1968). This interest has been concerned with trying to conserve both space for a sparse matrix and trying to reduce the computational time for algorithms that manipulate sparse matrices.

Enhancements and Modifications

During the theoretical and empirical investigation, several interesting new ideas were proposed. This section will explain these ideas and give examples that support the results. These proposed enhancements were not part of the literature reviewed for this work.

Weighting of Initial Conditions for the LU Decomposition

During the investigation of the LU decomposition with partial pivoting, it was observed that some of the initial conditions placed on the system were being pivoted. The standard method for including initial conditions into the augmented form of the associated linear system is to place a 1 on the diagonal, in the proper position, and place the initial value in the result vector. The pivoting of initial conditions, which would occur if some of the coefficients of the
characteristic equation were greater than 1, could cause computations that involve roundoff errors and cancellations to occur.

The knowledge that the initial conditions for a recurrence relation are accurate values, led to the suggestion of weighting the initial conditions. It was observed that if the value placed in the augmented matrix was greater than any of the coefficients of the characteristic equation, no pivoting of these values could occur. It is also assumed that the initial values in the result vector are also weighted accordingly.

To be able to examine the effects of this form of weighting, the software was modified to include the option of weighting the initial augmented matrix and result vector. The modification involved adding a routine to return a weighting vector. This vector is then used to weight the augmented matrix and result vector. The values to weight the matrix must be provided prior to execution. The weighting vector would contain all 1's except in the positions where weighting occurs.

The empirical investigation into the suggestion of weighting provided surprisingly good results. Most of the examples showed improvements in accuracy through the weighting of initial conditions. Few of the examples showed a decrease in accuracy with weighting.
The following example (4.8) will show the effects of weighting the initial conditions for a typical problem.

Example 4.8

The following third order recurrence relation (4.8.1) will be used to compare the effects of weighting the initial conditions of an augmented matrix.

\[(4.8.1)\]
\[y(n) = -0.0001y(n-1) + 7.0003y(n-2) - 6.0002y(n-3),\]

for \(n = 0, 1, \ldots, 15\), with a basis of solutions:

\[(4.8.2)\]
\[
\{ 1^n, 2^n, -3.0001^n \}.
\]

The computations for the intermediate solution of \(2^n\) will be performed. Two initial conditions and one terminal condition will be included.

Table 4.8.3 shows the unweighted augmented matrix \(A\) for the recurrence relation (4.8.1), and table 4.8.4 is the unweighted result vector \(b\). Notice that the 1's are included in the correct positions for the initial and terminal conditions, and the initial and terminal values are in the result vector \(b\).

For this example, because the largest magnitude of the coefficients is about 7.0, the choice of a weighting factor of 10.0 will be used. This means that the initial conditions will be weighted by a factor of 10.0 and the
rest of the matrix will be left unaltered. Table 4.8.5 shows the weighted augmented matrix \( A \) and table 4.8.6 shows the weighted result vector \( b \).

Table 4.8.7 contains the following results for comparison: the solution using the LU decomposition without pivoting on the unweighted matrix, the solution using the LU decomposition with pivoting on the unweighted matrix, and the solution using the LU decomposition with pivoting on the weighted matrix. The comparison of the relative errors for the three solution methods is in table 4.8.8. []

The computed condition numbers for example 4.8 are as follows: The unweighted augmented associated matrix \( 5.21975188E+05 \), and the weighted augmented associated matrix \( 1.0517188E+05 \). It would be expected that the results of computations on the matrix, with the preceding condition numbers, would provide acceptable results.

The results in table 4.8.8 show a significant increase in accuracy through the use of weighting the initial conditions.

The results from example 4.8 can be considered typical in the context that the results from the LU decomposition with pivoting had somewhat inaccurate results. It is typical to achieve better results through
<table>
<thead>
<tr>
<th>Table 4.8.3</th>
<th>The Augmented Matrix $A_a$ of Example 4.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>(left half on top, right half on bottom)</td>
<td></td>
</tr>
</tbody>
</table>
### Table 4.8.4

The Vector $b_a$ of Example 4.8

<table>
<thead>
<tr>
<th>$n$</th>
<th>$b(n)$</th>
<th>$n$</th>
<th>$b(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>8</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.20000000E+01</td>
<td>9</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.00000000E+00</td>
<td>10</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.00000000E+00</td>
<td>11</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.00000000E+00</td>
<td>12</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td>13</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.00000000E+00</td>
<td>14</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.00000000E+00</td>
<td>15</td>
<td>0.32768000E+05</td>
</tr>
</tbody>
</table>

### Table 4.8.6

The Weighted Vector $b_a$ of Example 4.8

<table>
<thead>
<tr>
<th>$n$</th>
<th>$b(n)$</th>
<th>$n$</th>
<th>$b(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+02</td>
<td>8</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.20000000E+02</td>
<td>9</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.00000000E+00</td>
<td>10</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.00000000E+00</td>
<td>11</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.00000000E+00</td>
<td>12</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td>13</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.00000000E+00</td>
<td>14</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.00000000E+00</td>
<td>15</td>
<td>0.32768000E+05</td>
</tr>
</tbody>
</table>
Table 4.8.5

The Weighted Augmented Matrix $A_a$ of Example 4.8

(Left half on top, right half on bottom)
Table 4.8.7
The Computed Values of Example 4.5

<table>
<thead>
<tr>
<th>n</th>
<th>IU Decomposition Without Pivoting</th>
<th>IU Decomposition With Pivoting</th>
<th>IU Decomposition Weighted initials</th>
<th>Estimate of Exact Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.100000000E+01</td>
<td>0.099980217E+00</td>
<td>0.100000000E+01</td>
<td>0.100000000E+01</td>
</tr>
<tr>
<td>1</td>
<td>0.200000000E+01</td>
<td>0.199981491E+01</td>
<td>0.200000000E+01</td>
<td>0.200000000E+01</td>
</tr>
<tr>
<td>2</td>
<td>0.3999717E+01</td>
<td>0.400004359E+01</td>
<td>0.400000000E+01</td>
<td>0.400000000E+01</td>
</tr>
<tr>
<td>3</td>
<td>0.800000000E+01</td>
<td>0.800035625E+01</td>
<td>0.800000000E+01</td>
<td>0.800000000E+01</td>
</tr>
<tr>
<td>4</td>
<td>0.160024835E+02</td>
<td>0.160010161E+02</td>
<td>0.160000000E+02</td>
<td>0.160000000E+02</td>
</tr>
<tr>
<td>5</td>
<td>0.320070795E+02</td>
<td>0.320023714E+02</td>
<td>0.320000000E+02</td>
<td>0.320000000E+02</td>
</tr>
<tr>
<td>6</td>
<td>0.640141590E+02</td>
<td>0.640097359E+02</td>
<td>0.640000000E+02</td>
<td>0.640000000E+02</td>
</tr>
<tr>
<td>7</td>
<td>0.128033830E+03</td>
<td>0.128009565E+03</td>
<td>0.128000000E+03</td>
<td>0.128000000E+03</td>
</tr>
<tr>
<td>8</td>
<td>0.256071660E+03</td>
<td>0.256019130E+03</td>
<td>0.256000000E+03</td>
<td>0.256000000E+03</td>
</tr>
<tr>
<td>9</td>
<td>0.512163240E+03</td>
<td>0.512070785E+03</td>
<td>0.512000000E+03</td>
<td>0.512000000E+03</td>
</tr>
<tr>
<td>10</td>
<td>0.102436650E+04</td>
<td>0.102406429E+04</td>
<td>0.102400000E+04</td>
<td>0.102400000E+04</td>
</tr>
<tr>
<td>11</td>
<td>0.204873300E+04</td>
<td>0.204812858E+04</td>
<td>0.204800000E+04</td>
<td>0.204800000E+04</td>
</tr>
<tr>
<td>12</td>
<td>0.409746600E+04</td>
<td>0.409642429E+04</td>
<td>0.409600000E+04</td>
<td>0.409600000E+04</td>
</tr>
<tr>
<td>13</td>
<td>0.819493200E+04</td>
<td>0.819285353E+04</td>
<td>0.819200000E+04</td>
<td>0.819200000E+04</td>
</tr>
<tr>
<td>14</td>
<td>0.163882215E+05</td>
<td>0.163851835E+05</td>
<td>0.163840000E+05</td>
<td>0.163840000E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.327680000E+05</td>
<td>0.327650000E+05</td>
<td>0.327630000E+05</td>
<td>0.327630000E+05</td>
</tr>
<tr>
<td>n</td>
<td>IU Decomposition Without Pivoting</td>
<td>IU Decomposition With Pivoting</td>
<td>IU Decomposition Weighted initials</td>
<td>Estimate of Exact Value</td>
</tr>
<tr>
<td>----</td>
<td>-----------------------------------</td>
<td>---------------------------------</td>
<td>-----------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>0</td>
<td>0.00000000E+00</td>
<td>0.19782782E-03</td>
<td>0.00000000E+00</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>1</td>
<td>0.00000000E+00</td>
<td>0.52291720E-04</td>
<td>0.00000000E+00</td>
<td>0.20000000E+01</td>
</tr>
<tr>
<td>2</td>
<td>0.13203383E-03</td>
<td>0.10892749E-04</td>
<td>0.00000000E+00</td>
<td>0.40000000E+01</td>
</tr>
<tr>
<td>3</td>
<td>0.14901161E-07</td>
<td>0.44614077E-04</td>
<td>0.14901161E-07</td>
<td>0.80000000E+01</td>
</tr>
<tr>
<td>4</td>
<td>0.15518069E-03</td>
<td>0.63323508E-04</td>
<td>0.14901161E-07</td>
<td>0.16000000E+02</td>
</tr>
<tr>
<td>5</td>
<td>0.22120714E-03</td>
<td>0.69903472E-04</td>
<td>0.14901161E-07</td>
<td>0.32000000E+02</td>
</tr>
<tr>
<td>6</td>
<td>0.27157366E-03</td>
<td>0.77705966E-04</td>
<td>0.14901161E-07</td>
<td>0.64000000E+02</td>
</tr>
<tr>
<td>7</td>
<td>0.27067959E-03</td>
<td>0.74834620E-04</td>
<td>0.14901161E-07</td>
<td>0.12800000E+03</td>
</tr>
<tr>
<td>8</td>
<td>0.30943811E-03</td>
<td>0.83565712E-04</td>
<td>0.14901161E-07</td>
<td>0.25600000E+03</td>
</tr>
<tr>
<td>9</td>
<td>0.26999414E-03</td>
<td>0.72404742E-04</td>
<td>0.14901161E-07</td>
<td>0.51200000E+03</td>
</tr>
<tr>
<td>10</td>
<td>0.33839078E-03</td>
<td>0.92116303E-04</td>
<td>0.14901161E-07</td>
<td>0.10240000E+04</td>
</tr>
<tr>
<td>11</td>
<td>0.24044514E-03</td>
<td>0.64030290E-04</td>
<td>0.14901161E-07</td>
<td>0.20480000E+04</td>
</tr>
<tr>
<td>12</td>
<td>0.39925173E-03</td>
<td>0.10353078E-03</td>
<td>0.14901161E-07</td>
<td>0.40960000E+04</td>
</tr>
<tr>
<td>13</td>
<td>0.16702712E-03</td>
<td>0.44390522E-04</td>
<td>0.00000000E+00</td>
<td>0.81920000E+04</td>
</tr>
<tr>
<td>14</td>
<td>0.50166249E-03</td>
<td>0.13321638E-03</td>
<td>0.14901161E-07</td>
<td>0.16384000E+05</td>
</tr>
<tr>
<td>15</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.32768000E+05</td>
</tr>
</tbody>
</table>
weighting the initial conditions, but the increase in accuracy may not always be as dramatic as example 4.8 shows.

Thoughts about weighting the terminal conditions were considered, but through the theoretical analysis it was shown to have no effect on the pivoting process. This is mostly due to the fact that the augmented system is a banded system and that the LU decomposition simulates Gaussian elimination to decompose the system.

In general the correct weighting to use on a linear system before applying numerical techniques is an important topic. In the context of recurrence relations the literature reviewed for this work seemed to skirt the issue. Bauer (1963) discusses the idea of optimally scaled matrices, but does not provide any methods for achieving these in the context of recurrence relations as linear systems.

Pivoting Strategies Other Than Partial Pivoting and Not Pivoting

The possibility of using some pivoting strategy other than partial pivoting or no pivoting raises many interesting questions. A review of the literature revealed that some attention had been given to this idea for sparse systems (Bunch 1974; Cohen 1974; Cryer 1968; Dax and Kaniel 1977; Duff 1977).
The main goal in these strategies has been to keep sparse systems, sometimes banded, sparse. In the area of recurrence relations there is ample evidence that pivoting produces less accurate results (example 4.5). Therefore, the suggestion of an enhancement that could include both the concepts of pivoting and no pivoting is provided.

This strategy would perform pivoting only if the norm of the current value in the pivot position is less than C times the norm of a value below the pivot. This restriction on pivoting would prevent the pivoting of values near the same magnitude. Pivoting values near the same magnitude could increase the possibility for more computations and cancellations in later calculations.

Notice that if the value of C were 0, no pivoting would occur whereas if the value of C were 1, normal pivoting would occur.

The standard error analysis for partial pivoting (Golub 1983) can be modified to include the constant C.

The value for C is dependent on the particular problem. For initial investigation values of C ranging from .01 to .1 are being used. Further investigation into the value of C and this pivoting strategy is planned.
The empirical investigation into this enhancement involved modifying the software in order to include this new pivoting strategy. The results from the sample set did not produce significant improvements in the results. This lack of improvement is partially due to the fact that only 50 values for a particular recurrence relation could be computed. This was a restriction placed on the software by the computing facility's main memory allocation. The theoretical investigation would be supported if larger numbers of values would be computed. Examples could easily be written that would make this strategy very effective, but that is not the goal of this work. Further investigation into this area of pivoting strategies is planned.
CHAPTER V

LEAST SQUARES

This chapter is a discussion of the application of least squares solution methods to linear systems of equations. The basic strategy of a least squares solution method applied to a linear system of equations is an attempt to equally distribute the errors, either relative or absolute, over the entire solution.

The first section of the chapter will present some of the concepts involved in the least squares solution of a linear system of equations. This solution will be shown to tend to equally distribute the absolute errors over the results.

Next a presentation of how the least squares approach fits into the area of recurrence relations will be given. The use of overdetermined recurrence relations and the augmented form of the associated linear system of equations will be used.

The concept of weighting the original matrix before the application of the least squares computations will be shown to have positive effects on the results. Methods for weighting, including row and column weighting, will be presented. It will be shown how the different methods of weighting effect the results.
Examples which show typical results will be included throughout the chapter. These examples will provide results to support the use of least squares solutions and the different weighting strategies.

The area of least squares has received significant attention in the literature. The work of Lawson and Hanson (1974) serves as a good reference text in this area. Also Reid (1967) presents useful information on the use of least squares to solve a banded system of equations.

Introduction to Least Squares

As an introduction to a least squares solution of a linear system of equations, a definition to characterize a problem as being of least squares type is given.

Definition 5.1

Assume $A$ is an $n \times n$ matrix and

\begin{equation}
\text{rank}(A) = m, \text{ thus } m \leq n. \text{ In general there does not exist an exact solution to the linear system}
\end{equation}

\begin{equation}
Ay = b.
\end{equation}

Therefore, a $y$ should be chosen to make 5.1.2 as close to the solution as possible with respect to some norm, $|| \cdot ||$; that is to find a $y$ that tries to minimize
If the $L_2$ norm is chosen, that is

\begin{equation}
|| x ||_2 = \sqrt{x'x} = \sqrt{\sum_{i=1}^{n} x_i^2},
\end{equation}

then the problem defined in equation 5.1.3 is known as the least squares problem, whose solution is given by

\begin{equation}
y_{LS} = (A'A)^{-1} A'b
\end{equation}

If the equation $Ay - b = 0$ has an almost exact solution, then the errors in the residuals $Ay - b$ will be approximately equal. [ ]

For a further explanation of definition 5.1, Golub and Van Loan (1983) should be consulted.

It is well known that solving for $y$ using equation 5.1.5 leads to unstable computations (Businger 1965; Elden 1977a, 1977b; George and Heath 1980; Lawson and Hanson 1974; Peters and Wilkinson 1970; Powell and Reid 1968; van der Sluis 1975). Most of the literature that presented evidence of a problem in the computations of equation 5.1.5, also suggested options for more stable computations of $y$. Also significant progress in the software solution of the least squares problem was found (Dongarra et al. 1978; Elden 1977a; Paige and Saunders 1982a, 1982b; Moler 1980).
The connection between the general least squares problem and solving a recurrence relation using the least squares method will be presented in the next section

Least Squares and Recurrence Relations

In the case of recurrence relations, the use of overdetermined systems of equations (2.26.3) and the augmented form of the linear system of equations (2.30) seems to be an "attractive" combination. This attractiveness is due, in part, to being able to set initial and terminal conditions for the recurrence relation. These values would then be an integral part of the least squares computations. It should be noted that the number of initial and terminal conditions provided will affect the results. As expected, the more conditions provided prior to the least squares computations, the better the results.

Notice that no indication of the number of initial, intermediate, or terminal conditions is given, but only that the total number of conditions should be greater that p, the order of the recurrence relation. The next definition will advocate the use of p initial and p terminal conditions.
Definition 5.2

Suppose that in equation 5.1.2, the matrix $A$, which represents a $p$th order recurrence relation, is an augmented system of equations with $p$ initial and $p$ terminal conditions. If the least squares method of solution is used to solve 5.1.2, then take $y_{LS}$ as an estimate of the true solution $y$. []

The use of definition 5.2 provides at least two obvious advantages. The first advantage is that no prior knowledge of the relative dominance of all the solutions is required, as in Oliver's method. A second advantage is that decoupling and local variations before the coefficients reach asymptotic limits is handled in a natural manner.

The choice of using $p$ initial and $p$ terminal conditions, as in definition 5.2, is by far not the only option. The inclusion of more than $p$ initial or terminal conditions would only provide better results. The inclusion of intermediate conditions could also help in the solution process for the least squares method. It should be kept in mind that the acquisition of these values may not be a trivial process; therefore, the choice of the number and type of conditions should be determined for each situation.
To help in the understanding of the least squares method, an example will be presented. This example (5.3) shows the least squares computations for the recurrence relation 3.14.1 of example 3.14. For ease of reference, the definition of the recurrence relation and its solutions will be reintroduced.

For this example (5.3) a third order recurrence relation (5.3.1) with characteristic values .1, 1.1, and 10.1 will be examined. Also, the choice of the number of initial and terminal conditions will be equal to p, the order of the recurrence relation.

Example 5.3

For this example, the following third order recurrence relation will be used:

(5.3.1)

\[ y(n) - 11.3y(n-1) + 12.23y(n-2) - 1.111y(n-3) = 0. \]

The three characteristic values for this recurrence relation are .1, 1.1, and 10.1. For this example the following three solutions will be examined:

(5.3.1)

\[ s_1 = .1^n, \quad s_2 = 1.1^n, \quad s_3 = 10.1^n. \]

It should be noticed that \(s_1\) is a minimal solution, \(s_2\) an intermediate solution, and \(s_3\) a dominate solution.

For each of the solutions \(s_1, s_2, \) and \(s_3\), the least squares solution will be examined.
For the computations, three initial and three terminal conditions will be set. Due to the fact that all of the values in the original result vector participate in a least squares computation, the values 0 through 15 will be computed. This is slightly different than any of the previously discussed methods, where the initial and terminal values may not have been altered.

Table 5.3.3 contains the augmented matrix. The matrix is presented in two parts. To view the complete matrix, the bottom half must be placed to the right of the top half. Table 5.3.4 contains the three initial result vectors including the initial and terminal conditions. Tables 5.3.5 through 5.3.7 contain the results of the least squares computations for the three solutions. The tables will include: the accurate values, the computed values, the absolute errors, and the relative errors.

The results in tables 5.3.5 through 5.3.7 should provide support to the claim that the absolute errors will tend to be equal. These results should be compared to the tables for the same solutions in example 3.14. This comparison will reveal that the computed results may not be as accurate for all values, but the solution is more consistent over the entire solution sequence.
<table>
<thead>
<tr>
<th>Table 5.3.3 of Example 5.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Matrix $A_3$</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 5.3.4

The Initial b Vectors for Example 5.3

<table>
<thead>
<tr>
<th>n</th>
<th>Solution $s_1$</th>
<th>Solution $s_2$</th>
<th>Solution $s_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>1</td>
<td>0.10000000E-01</td>
<td>0.11000000E+01</td>
<td>0.10100000E+02</td>
</tr>
<tr>
<td>2</td>
<td>0.10000000E-02</td>
<td>0.12100000E+01</td>
<td>0.10201000E+03</td>
</tr>
<tr>
<td>3</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>8</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>9</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>10</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>11</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>12</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>14</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>15</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>16</td>
<td>0.10000000E-12</td>
<td>0.34522712E+01</td>
<td>0.11380932E+14</td>
</tr>
<tr>
<td>17</td>
<td>0.10000000E-13</td>
<td>0.37974983E+01</td>
<td>0.11494742E+15</td>
</tr>
<tr>
<td>18</td>
<td>0.10000000E-14</td>
<td>0.41772482E+01</td>
<td>0.11606960E+16</td>
</tr>
</tbody>
</table>
### Table 5.3.5
Calculation of Solution $S_1$ of Equation 5.3.1 Using Least Squares

<table>
<thead>
<tr>
<th>n</th>
<th>$w(n)$</th>
<th>$w_c(n)$</th>
<th>$e_a(n)$</th>
<th>$e_c(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1000000E+01</td>
<td>0.99999996E+00</td>
<td>0.44703484E-07</td>
<td>0.44703484E-07</td>
</tr>
<tr>
<td>1</td>
<td>0.1000000E+00</td>
<td>0.99999992E-01</td>
<td>0.74506806E-08</td>
<td>0.74506806E-08</td>
</tr>
<tr>
<td>2</td>
<td>0.1000000E+01</td>
<td>0.99999952E-02</td>
<td>0.47730282E-08</td>
<td>0.47730282E-08</td>
</tr>
<tr>
<td>3</td>
<td>0.1000000E+02</td>
<td>0.99999903E-03</td>
<td>0.49476512E-08</td>
<td>0.49476512E-08</td>
</tr>
<tr>
<td>4</td>
<td>0.1000000E+03</td>
<td>0.99994258E-04</td>
<td>0.57434598E-08</td>
<td>0.57434598E-08</td>
</tr>
<tr>
<td>5</td>
<td>0.1000000E+04</td>
<td>0.99964951E-05</td>
<td>0.35048513E-08</td>
<td>0.35048513E-08</td>
</tr>
<tr>
<td>6</td>
<td>0.1000000E+05</td>
<td>0.99638824E-06</td>
<td>0.36117598E-08</td>
<td>0.36117598E-08</td>
</tr>
<tr>
<td>7</td>
<td>0.1000000E+06</td>
<td>0.95704668E-07</td>
<td>0.40295458E-08</td>
<td>0.40295458E-08</td>
</tr>
<tr>
<td>8</td>
<td>0.1000000E+07</td>
<td>0.65081889E-08</td>
<td>0.34912813E-08</td>
<td>0.34912813E-08</td>
</tr>
<tr>
<td>9</td>
<td>0.1000000E+08</td>
<td>-0.28964392E-08</td>
<td>0.36964391E-08</td>
<td>0.36964391E-08</td>
</tr>
<tr>
<td>10</td>
<td>0.1000000E+09</td>
<td>-0.36239648E-08</td>
<td>0.37289658E-08</td>
<td>0.37289658E-08</td>
</tr>
<tr>
<td>11</td>
<td>0.1000000E+10</td>
<td>-0.37817235E-08</td>
<td>0.37917235E-08</td>
<td>0.37917235E-08</td>
</tr>
<tr>
<td>12</td>
<td>0.1000000E+11</td>
<td>-0.40685358E-08</td>
<td>0.40785348E-08</td>
<td>0.40785348E-08</td>
</tr>
<tr>
<td>13</td>
<td>0.1000000E+12</td>
<td>-0.30531712E-08</td>
<td>0.30531712E-08</td>
<td>0.30531712E-08</td>
</tr>
<tr>
<td>14</td>
<td>0.1000000E+13</td>
<td>-0.18951596E-08</td>
<td>0.18951596E-08</td>
<td>0.18951596E-08</td>
</tr>
<tr>
<td>15</td>
<td>0.1000000E+14</td>
<td>0.42225745E-10</td>
<td>0.42225745E-10</td>
<td>0.42225745E-10</td>
</tr>
</tbody>
</table>
Table 5.3.6
Calculation of Solution $s_2$ of Equation 5.3.1
Using Least Squares

<table>
<thead>
<tr>
<th>n</th>
<th>$w(n)$</th>
<th>$w_c(n)$</th>
<th>$e_a(n)$</th>
<th>$e_p(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>0.10612221E+01</td>
<td>0.61222136E-01</td>
<td>0.61222136E-01</td>
</tr>
<tr>
<td>1</td>
<td>0.11000000E+01</td>
<td>0.55457087E+00</td>
<td>0.54542913E+00</td>
<td>0.49584466E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.12100000E+01</td>
<td>0.54204147E+00</td>
<td>0.66798536E+00</td>
<td>0.55203184E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.13310000E+01</td>
<td>0.57679053E+00</td>
<td>0.75420940E+00</td>
<td>0.56664863E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.14641000E+01</td>
<td>0.62038148E+00</td>
<td>0.84730858E+00</td>
<td>0.57627953E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.16105100E+01</td>
<td>0.69988170E+00</td>
<td>0.94068238E+00</td>
<td>0.58405613E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.17715610E+01</td>
<td>0.72505678E+00</td>
<td>1.04603425E+00</td>
<td>0.59047034E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.19487171E+01</td>
<td>0.79722570E+00</td>
<td>1.16099148E+00</td>
<td>0.59577215E+00</td>
</tr>
<tr>
<td>8</td>
<td>0.21435833E+01</td>
<td>0.85707211E+00</td>
<td>1.28851678E+00</td>
<td>0.60016952E+00</td>
</tr>
<tr>
<td>9</td>
<td>0.23594769E+01</td>
<td>0.93368859E+00</td>
<td>1.42079181E+00</td>
<td>0.60394855E+00</td>
</tr>
<tr>
<td>10</td>
<td>0.25937424E+01</td>
<td>1.01570565E+00</td>
<td>1.57803689E+00</td>
<td>0.60840157E+00</td>
</tr>
<tr>
<td>11</td>
<td>0.28531166E+01</td>
<td>1.07199449E+00</td>
<td>1.73118228E+00</td>
<td>0.62429352E+00</td>
</tr>
<tr>
<td>12</td>
<td>0.31384238E+01</td>
<td>0.78970416E+00</td>
<td>2.35245796E+00</td>
<td>0.74956555E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.34522751E+01</td>
<td>0.40481815E+00</td>
<td>3.04779303E+00</td>
<td>0.82836030E+00</td>
</tr>
<tr>
<td>14</td>
<td>0.37974683E+01</td>
<td>-0.97014849E-02</td>
<td>3.00702024E+00</td>
<td>0.10025514E+01</td>
</tr>
<tr>
<td>15</td>
<td>0.41772480E+01</td>
<td>-0.30043064E-02</td>
<td>4.18025235E+00</td>
<td>0.10007192E+01</td>
</tr>
</tbody>
</table>
Table 5.3.7
Calculation of Solution $s_3$ of Equation 5.3.1
Using Least Squares

<table>
<thead>
<tr>
<th>n</th>
<th>w(r)</th>
<th>$w_c(r)$</th>
<th>$e_a(r)$</th>
<th>$e_x(r)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>-0.20674058E+13</td>
<td>0.20674058E+13</td>
<td>0.20674058E+13</td>
</tr>
<tr>
<td>1</td>
<td>0.10100000E+02</td>
<td>0.18418354E+14</td>
<td>0.18418354E+14</td>
<td>0.18418354E+13</td>
</tr>
<tr>
<td>2</td>
<td>0.10200000E+03</td>
<td>0.22659996E+14</td>
<td>0.22659996E+14</td>
<td>0.22111554E+12</td>
</tr>
<tr>
<td>3</td>
<td>0.10300000E+04</td>
<td>0.25468564E+14</td>
<td>0.25468564E+14</td>
<td>0.24719537E+11</td>
</tr>
<tr>
<td>4</td>
<td>0.10400000E+05</td>
<td>0.28491577E+14</td>
<td>0.28491577E+14</td>
<td>0.27379845E+10</td>
</tr>
<tr>
<td>5</td>
<td>0.10500000E+06</td>
<td>0.31763666E+14</td>
<td>0.31763666E+14</td>
<td>0.30222038E+09</td>
</tr>
<tr>
<td>6</td>
<td>0.10600000E+07</td>
<td>0.35323730E+14</td>
<td>0.35323730E+14</td>
<td>0.33276550E+08</td>
</tr>
<tr>
<td>7</td>
<td>0.10700000E+08</td>
<td>0.39204356E+14</td>
<td>0.39204356E+14</td>
<td>0.36567047E+07</td>
</tr>
<tr>
<td>8</td>
<td>0.10800000E+09</td>
<td>0.43442096E+14</td>
<td>0.43442096E+14</td>
<td>0.40117911E+06</td>
</tr>
<tr>
<td>9</td>
<td>0.10900000E+10</td>
<td>0.48071281E+14</td>
<td>0.48071281E+14</td>
<td>0.43962489E+05</td>
</tr>
<tr>
<td>10</td>
<td>0.11000000E+11</td>
<td>0.53107939E+14</td>
<td>0.53107939E+14</td>
<td>0.48067924E+04</td>
</tr>
<tr>
<td>11</td>
<td>0.11100000E+12</td>
<td>0.58216735E+14</td>
<td>0.58216735E+14</td>
<td>0.52181039E+03</td>
</tr>
<tr>
<td>12</td>
<td>0.11200000E+13</td>
<td>0.63225462E+14</td>
<td>0.63225462E+14</td>
<td>0.53187342E+02</td>
</tr>
<tr>
<td>13</td>
<td>0.11300000E+14</td>
<td>0.68234189E+14</td>
<td>0.68234189E+14</td>
<td>0.53657010E+01</td>
</tr>
<tr>
<td>14</td>
<td>0.11400000E+15</td>
<td>0.73242915E+14</td>
<td>0.73242915E+14</td>
<td>0.54127634E+01</td>
</tr>
<tr>
<td>15</td>
<td>0.11500000E+16</td>
<td>0.78251641E+14</td>
<td>0.78251641E+14</td>
<td>0.55534505E+00</td>
</tr>
</tbody>
</table>
Other configurations for the number of initial, intermediate, and terminal conditions were considered and examined. The typical results did not show a significant increase in accuracy, even though more consistent results were achieved. The inclusion of more than the suggested number of initial and terminal conditions leads to more computations. These added computations may offset the small increase in accuracy achieved through their inclusion.

It should be noted that the computation counts and timing formulas for a least squares solution are roughly the same as for the QR decomposition presented previously in chapter four. This is because a least squares solution involves the QR decomposition routines and an equivalent solution routine.

Weighting for Least Squares

Weighting is a very important question in the solution of linear systems of equations, as well as in least squares. A complete discussion of weighting has not been presented previously in this work, though it could have fit into previous chapters. Chapter four included a section on weighting in the context of the LU decomposition with pivoting.
As a rule of thumb for weighting, the selection of the weighting should provide that in any row of an augmented matrix the largest element is of order unity.

It is with respect to this weighting that we claim the absolute errors in the solution $y_{LS}$ will tend to be evenly distributed.

For the discussion of weighting in the context of least squares, the definition of weighting matrices will be introduced. The definition of a row weighting matrix $R$, and a column weighting matrix $C$ will be provided and used in later definitions.

**Definition 5.4**

Given a matrix $A$ of size $n$ by $m$, a matrix $R$ of size $n$ by $n$, and a matrix $C$ of size $m$ by $m$, suppose $R$ and $C$ are diagonal and have no zero diagonal elements. If $R$ is used to premultiply $A$, then $R$ will weight the rows of $A$. Likewise, if $C$ is used to postmultiply $A$, then $C$ will weight the columns of $A$. Furthermore, the $i$th diagonal element of $R$ or $C$ will weight the $i$th row or column of $A$ respectively. []

The weighting matrices may now be included in the linear system equation from 5.1.2. The inclusion of the weighting matrices will not alter the definition of the equation, but will allow for the weighting of the augmented matrix before the application of the least
squares computations. If no weighting is desired, then the weighting matrices should be set to identity matrices so as not to effect the results.

**Definition 5.5**

Given the weighting matrices R and C from 5.4, then equation 5.1.2 may identically rewritten as

\[(5.5.1) \quad RAC^{-1}Cy = Rb.\]

Then the solution of 5.5.1, in the least squares sense with weighting, may be considered. []

The choice of R and C for 5.5.1 may be made freely. The selection of R and C may be done to achieve certain results from the least squares computations. The proper selection may enhance the relative or absolute errors in the results. The following definition (5.6) will provide a general approach to the least squares solution using the weighting matrices.

**Definition 5.6**

Given the following equation

\[(5.6.1) \quad RAC^{-1}y = Rb,\]

with the least squares solution \(y_{LS}\), then the computed solution for 5.6.1 would be

\[(5.6.2) \quad y_c = C^{-1}y_{LS}.\]
The next definition (5.7) will provide one of many possible choices for R and C that will tend to evenly distribute the relative errors in the least squares solution. This choice for R and C follows when the asymptotics for the true solution y(n) are known.

Definition 5.7

Given that the asymptotic behavior for the true solution y(n) of a recurrence relation is known and can be expressed as

\[(5.7.1)\]
\[y(n) \sim n^j_1 n, \text{ for } 1 \text{ the current root.}\]

For the column weighting matrix C

\[(5.7.2)\]
\[C_{n} = \frac{1}{n^j_1 n}, \text{ for } n = 1, 2, \ldots, m;\]

which will tend to make y_{LS} a constant multiple of e, the elementary vector of all ones.

Then consider the singular value decomposition of

\[(5.7.3)\]
\[AC^{-1} = UEV, \text{ and select}\]

\[(5.7.4)\]
\[R = E^* U'.\]

Where E^* is the diagonal matrix with elements
Here is a diagonal element of the matrix $E$. 

It should be noted that for a homogeneous recurrence relation there should be no trouble in computing the right side of equation 5.6.1.

With the weighting suggested in 5.7, the relative errors of the computed solution $y_c$ should all be approximately equal. Such a statement could not be made in previous chapters without a reference to weighting. Due to the complexity of this concept, the discussion presented here has been kept to a minimum.

The next example (5.8) will provide evidence for the use of weighting in least squares. The recurrence relation from example 5.3 will be reused. This time weighting will be included. For this simple example into weighting, only row weighting will be performed. Each row of the augmented matrix will be multiplied by the reciprocal of the solution that represents that row.

The choice to use the exact solution for the weighting was only for convenience. This choice in general involves using some sequence that resembles the solution.
Example 5.8

For this example, recurrence relation 5.3.1 (3.14.1) will be used. The weighted least squares solution will be computed for solutions $s_1$, $s_2$, and $s_3$ (5.3.2). Table 5.3.3 contains the original unweighted augmented matrix $A_a$, and table 5.3.4 the unweighted result vectors for the three solutions.

Because a weighting matrix is a diagonal matrix, the representation of this matrix may be in vector notation to save space. For this example row weighting was chosen. This choice follows from definition 5.7, where the rows will be weighted by the reciprocal of the solution that represents that row. The weighting vectors, that represent the row weighting matrices, are in table 5.8.1.

The results of applying the three weighting matrices to the original augmented matrix can be found in tables 5.8.2 through 5.8.4. The results of applying the weighting matrices to the original result vectors can be found in table 5.8.5.

Tables 5.8.6 through 5.8.8 contain the results of the least squares computations. These tables contain: the accurate values, the computed values, the absolute errors, and the relative errors. []
<table>
<thead>
<tr>
<th>n</th>
<th>Solution $s_1$</th>
<th>Solution $s_2$</th>
<th>Solution $s_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>1</td>
<td>0.10000000E+02</td>
<td>0.90909091E+00</td>
<td>0.99009900E-01</td>
</tr>
<tr>
<td>2</td>
<td>0.10000000E+03</td>
<td>0.82644629E+00</td>
<td>0.98029604E-02</td>
</tr>
<tr>
<td>3</td>
<td>0.10000000E+04</td>
<td>0.75131481E+00</td>
<td>0.97059014E-03</td>
</tr>
<tr>
<td>4</td>
<td>0.10000000E+05</td>
<td>0.68301346E+00</td>
<td>0.96098034E-04</td>
</tr>
<tr>
<td>5</td>
<td>0.10000000E+06</td>
<td>0.62092133E+00</td>
<td>0.95146568E-05</td>
</tr>
<tr>
<td>6</td>
<td>0.10000000E+07</td>
<td>0.56447393E+00</td>
<td>0.94204523E-06</td>
</tr>
<tr>
<td>7</td>
<td>0.10000000E+08</td>
<td>0.51315813E+00</td>
<td>0.93271804E-07</td>
</tr>
<tr>
<td>8</td>
<td>0.10000000E+09</td>
<td>0.46650739E+00</td>
<td>0.92348321E-08</td>
</tr>
<tr>
<td>9</td>
<td>0.10000000E+10</td>
<td>0.42409763E+00</td>
<td>0.91433981E-09</td>
</tr>
<tr>
<td>10</td>
<td>0.10000000E+11</td>
<td>0.38554330E+00</td>
<td>0.90528694E-10</td>
</tr>
<tr>
<td>11</td>
<td>0.10000000E+12</td>
<td>0.35049391E+00</td>
<td>0.89632371E-11</td>
</tr>
<tr>
<td>12</td>
<td>0.10000000E+13</td>
<td>0.31863083E+00</td>
<td>0.88744921E-12</td>
</tr>
<tr>
<td>13</td>
<td>0.10000000E+14</td>
<td>0.28966439E+00</td>
<td>0.87866258E-13</td>
</tr>
<tr>
<td>14</td>
<td>0.10000000E+15</td>
<td>0.26333126E+00</td>
<td>0.86996295E-14</td>
</tr>
<tr>
<td>15</td>
<td>0.10000000E+16</td>
<td>0.23939206E+00</td>
<td>0.86134944E-15</td>
</tr>
<tr>
<td>16</td>
<td>0.10000000E+14</td>
<td>0.28966439E+00</td>
<td>0.87866258E-13</td>
</tr>
<tr>
<td>17</td>
<td>0.10000000E+15</td>
<td>0.26333126E+00</td>
<td>0.86996295E-14</td>
</tr>
<tr>
<td>18</td>
<td>0.10000000E+16</td>
<td>0.23939206E+00</td>
<td>0.86134944E-15</td>
</tr>
<tr>
<td></td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.983561E-01</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.834710E00</td>
<td>0.918838E00</td>
<td>0.948957E-01</td>
<td>0.751349E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.758987E00</td>
<td>0.835354E-01</td>
<td>0.771852E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.795843E00</td>
<td>0.759867E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.627130E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
<tr>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
<td>0.000000E00</td>
</tr>
</tbody>
</table>

Table 5.8.3

The Weighted Matrix $A_q$ for Solution $s_2$ of Example 5.8
### Table 5.8.4

The Weighted Matrix $A$ for Solution $s_3$ of Example 5.8

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
### Table 5.8.5

The Weighted $b$ Vectors for Example 5.8

<table>
<thead>
<tr>
<th>n</th>
<th>Solution $s_1$</th>
<th>Solution $s_2$</th>
<th>Solution $s_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>1</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>2</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>3</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>8</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>9</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>10</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>11</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>12</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>14</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>15</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>16</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>17</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>18</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
</tr>
</tbody>
</table>
Table 5.8.6
Calculation of Solution $s_1$ of Equation 5.3.1 for Example 5.8
Using Weighted Least Squares

<table>
<thead>
<tr>
<th>n</th>
<th>$w(n)$</th>
<th>$w_c(n)$</th>
<th>$e_a(n)$</th>
<th>$e_n(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1000000E+01</td>
<td>0.99999963E-00</td>
<td>0.31441450E-05</td>
<td>0.31441450E-05</td>
</tr>
<tr>
<td>1</td>
<td>0.1000000E+00</td>
<td>0.99999969E-01</td>
<td>0.31106174E-06</td>
<td>0.31106174E-06</td>
</tr>
<tr>
<td>2</td>
<td>0.1000000E+01</td>
<td>0.99999969E-02</td>
<td>0.31082891E-07</td>
<td>0.31082891E-07</td>
</tr>
<tr>
<td>3</td>
<td>0.1000000E+02</td>
<td>0.99999969E-03</td>
<td>0.30850060E-08</td>
<td>0.30850060E-08</td>
</tr>
<tr>
<td>4</td>
<td>0.1000000E+03</td>
<td>0.99999969E-04</td>
<td>0.30740921E-09</td>
<td>0.30740921E-09</td>
</tr>
<tr>
<td>5</td>
<td>0.1000000E+04</td>
<td>0.99999969E-05</td>
<td>0.30631759E-10</td>
<td>0.30631759E-10</td>
</tr>
<tr>
<td>6</td>
<td>0.1000000E+05</td>
<td>0.99999969E-06</td>
<td>0.30127012E-11</td>
<td>0.30127012E-11</td>
</tr>
<tr>
<td>7</td>
<td>0.1000000E+06</td>
<td>0.99999969E-07</td>
<td>0.30009088E-12</td>
<td>0.30009088E-12</td>
</tr>
<tr>
<td>8</td>
<td>0.1000000E+07</td>
<td>0.99999969E-08</td>
<td>0.29976022E-13</td>
<td>0.29976022E-13</td>
</tr>
<tr>
<td>9</td>
<td>0.1000000E+08</td>
<td>0.99999969E-09</td>
<td>0.29923980E-14</td>
<td>0.29923980E-14</td>
</tr>
<tr>
<td>10</td>
<td>0.1000000E+09</td>
<td>0.99999969E-10</td>
<td>0.29907797E-15</td>
<td>0.29907797E-15</td>
</tr>
<tr>
<td>11</td>
<td>0.1000000E+10</td>
<td>0.99999969E-11</td>
<td>0.29876022E-16</td>
<td>0.29876022E-16</td>
</tr>
<tr>
<td>12</td>
<td>0.1000000E+11</td>
<td>0.99999969E-12</td>
<td>0.29824782E-17</td>
<td>0.29824782E-17</td>
</tr>
<tr>
<td>13</td>
<td>0.1000000E+12</td>
<td>0.99999969E-13</td>
<td>0.29824782E-18</td>
<td>0.29824782E-18</td>
</tr>
<tr>
<td>14</td>
<td>0.1000000E+13</td>
<td>0.1000018E-13</td>
<td>0.29831863E-19</td>
<td>0.29831863E-19</td>
</tr>
<tr>
<td>15</td>
<td>0.1000000E+14</td>
<td>0.99999963E-15</td>
<td>0.29831863E-20</td>
<td>0.29831863E-20</td>
</tr>
</tbody>
</table>
Table 5.8.7
Calculation of Solution $s_2$ of Equation 5.3.1 for Example 5.8
Using Weighted Least Squares

<table>
<thead>
<tr>
<th>$n$</th>
<th>$w(n)$</th>
<th>$w_c(n)$</th>
<th>$e_a(n)$</th>
<th>$e_x(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$0.10000000E+01$</td>
<td>$0.99999876E+00$</td>
<td>$0.13411045E-06$</td>
<td>$0.13411045E-06$</td>
</tr>
<tr>
<td>1</td>
<td>$0.11000000E+01$</td>
<td>$0.10999999E+01$</td>
<td>$0.11920929E-06$</td>
<td>$0.10837293E-06$</td>
</tr>
<tr>
<td>2</td>
<td>$0.12100000E+01$</td>
<td>$0.12099999E+01$</td>
<td>$0.44703483E-07$</td>
<td>$0.36550208E-07$</td>
</tr>
<tr>
<td>3</td>
<td>$0.13310000E+01$</td>
<td>$0.13310000E+01$</td>
<td>$0.29802328E-07$</td>
<td>$0.22390926E-07$</td>
</tr>
<tr>
<td>4</td>
<td>$0.14641000E+01$</td>
<td>$0.14641000E+01$</td>
<td>$0.44703483E-07$</td>
<td>$0.30533003E-07$</td>
</tr>
<tr>
<td>5</td>
<td>$0.16105100E+01$</td>
<td>$0.16105100E+01$</td>
<td>$0.29802328E-07$</td>
<td>$0.18504895E-07$</td>
</tr>
<tr>
<td>6</td>
<td>$0.17715100E+01$</td>
<td>$0.17715100E+01$</td>
<td>$0.29802328E-07$</td>
<td>$0.16822634E-07$</td>
</tr>
<tr>
<td>7</td>
<td>$0.19487100E+01$</td>
<td>$0.19487100E+01$</td>
<td>$0.29802328E-07$</td>
<td>$0.15289304E-07$</td>
</tr>
<tr>
<td>8</td>
<td>$0.21435988E+01$</td>
<td>$0.21435988E+01$</td>
<td>$0.11920929E-06$</td>
<td>$0.55612048E-07$</td>
</tr>
<tr>
<td>9</td>
<td>$0.23579476E+01$</td>
<td>$0.23579476E+01$</td>
<td>$0.89406679E-07$</td>
<td>$0.37912838E-07$</td>
</tr>
<tr>
<td>10</td>
<td>$0.25937424E+01$</td>
<td>$0.25937424E+01$</td>
<td>$0.29802328E-07$</td>
<td>$0.11490096E-07$</td>
</tr>
<tr>
<td>11</td>
<td>$0.28531166E+01$</td>
<td>$0.28531166E+01$</td>
<td>$0.00000000E+00$</td>
<td>$0.00000000E+00$</td>
</tr>
<tr>
<td>12</td>
<td>$0.31384283E+01$</td>
<td>$0.31384283E+01$</td>
<td>$0.59604645E-07$</td>
<td>$0.18591772E-07$</td>
</tr>
<tr>
<td>13</td>
<td>$0.34527121E+01$</td>
<td>$0.34527121E+01$</td>
<td>$0.59604645E-07$</td>
<td>$0.17263438E-07$</td>
</tr>
<tr>
<td>14</td>
<td>$0.37974984E+01$</td>
<td>$0.37974984E+01$</td>
<td>$0.14901161E-06$</td>
<td>$0.39239416E-07$</td>
</tr>
<tr>
<td>15</td>
<td>$0.41772482E+01$</td>
<td>$0.41772482E+01$</td>
<td>$0.17881393E-06$</td>
<td>$0.42806368E-07$</td>
</tr>
</tbody>
</table>
Table 5.8.8
Calculation of Solution $s_3$ of Equation 5.3.1 for Example 5.8
Using Weighted Least Squares

<table>
<thead>
<tr>
<th>$n$</th>
<th>$w(n)$</th>
<th>$w_c(n)$</th>
<th>$e_a(n)$</th>
<th>$e_z(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>0.10100000E+02</td>
<td>0.10100000E+02</td>
<td>0.11920935E-06</td>
<td>0.11920935E-07</td>
</tr>
<tr>
<td>2</td>
<td>0.10200000E+03</td>
<td>0.10200000E+03</td>
<td>0.95367432E-06</td>
<td>0.95367432E-06</td>
</tr>
<tr>
<td>3</td>
<td>0.10300000E+04</td>
<td>0.10300000E+04</td>
<td>0.45776367E-04</td>
<td>0.45776367E-04</td>
</tr>
<tr>
<td>4</td>
<td>0.10400000E+05</td>
<td>0.10400000E+05</td>
<td>0.61351563E-03</td>
<td>0.61351563E-03</td>
</tr>
<tr>
<td>5</td>
<td>0.10500000E+06</td>
<td>0.10500000E+06</td>
<td>0.63593659E-02</td>
<td>0.63593659E-02</td>
</tr>
<tr>
<td>6</td>
<td>0.10600000E+07</td>
<td>0.10600000E+07</td>
<td>0.78125000E-01</td>
<td>0.78125000E-01</td>
</tr>
<tr>
<td>7</td>
<td>0.10700000E+08</td>
<td>0.10700000E+08</td>
<td>0.10000000E+01</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>8</td>
<td>0.10800000E+09</td>
<td>0.10800000E+09</td>
<td>0.11000000E+02</td>
<td>0.11000000E+02</td>
</tr>
<tr>
<td>9</td>
<td>0.10900000E+10</td>
<td>0.10900000E+10</td>
<td>0.12000000E+03</td>
<td>0.12000000E+03</td>
</tr>
<tr>
<td>10</td>
<td>0.11000000E+11</td>
<td>0.11000000E+11</td>
<td>0.16640000E+04</td>
<td>0.16640000E+04</td>
</tr>
<tr>
<td>11</td>
<td>0.11100000E+12</td>
<td>0.11100000E+12</td>
<td>0.21504000E+05</td>
<td>0.21504000E+05</td>
</tr>
<tr>
<td>12</td>
<td>0.11200000E+13</td>
<td>0.11200000E+13</td>
<td>0.22937600E+06</td>
<td>0.22937600E+06</td>
</tr>
<tr>
<td>13</td>
<td>0.11300000E+14</td>
<td>0.11300000E+14</td>
<td>0.24936300E+07</td>
<td>0.24936300E+07</td>
</tr>
<tr>
<td>14</td>
<td>0.11400000E+15</td>
<td>0.11400000E+15</td>
<td>0.15777216E+08</td>
<td>0.15777216E+08</td>
</tr>
<tr>
<td>15</td>
<td>0.11500000E+16</td>
<td>0.11500000E+16</td>
<td>0.10000000E+09</td>
<td>0.10000000E+09</td>
</tr>
</tbody>
</table>

163
The results in tables 5.8.6 through 5.8.8 show that the relative errors for the three solutions were approximately equal. Furthermore, the results for all three solutions were very acceptable. These tables should be compared to the tables for example 3.14, for the same solutions. For this comparison of tables, keep in mind the facts about the differences in the solution methods being compared. A true comparison of the least squares method to the other methods cannot be justified due to the different initial requirements for the different methods.

The whole area of weighting, in the least squares context, has generated many papers and opinions. There are several factors to consider when performing the weighting that may restrict the available methods. One method that has received some attention in the literature is the idea of iterative reweighting (Bjorck and Golub 1967; Bjorck 1967b, 1968; Elden 1977a, 1977b; Golub and Wilkinson 1966) This would involve the reweighting of the least squares problem during the least squares process. This reweighting could use the partial results obtained in previous computations.

The discussion and presentation of an error analysis for the least squares solution method is beyond the level of this work. The work of Jenning and Osborne
(1974) and that of van der Sluis (1975) provides both an error analysis and discussions on the concept of stability in the context of least squares solutions of recurrence relations.

In conclusion, the use of least squares techniques in the solution of recurrence relations appears promising.
CHAPTER VI

SOFTWARE

This chapter is a discussion of the software provided for the numerical solutions of recurrence relations. There are two areas of software that have been included in this work.

The first area is that of existing software. This includes the description of routines that are utilized, and that of existing routines that have been modified to provide special functions and options.

The second area is that of new software. This includes software that was designed to interact with the existing software. This software provides easier access for input of information and the formatted retrieval of desired output.

The first part of the chapter will be devoted to an explanation of the basics of the existing software. LINPACK (Dongarra et al. 1978) is the existing software package that was used in this work. Included will be a description of any modifications to the LINPACK software.

The second part will explain the newly developed software, and the third part will explain the operation and interaction of the combined software.
Most of the results for examples throughout this work were obtained as output from the software; although some of the results were reformatted to provide more suitable tables.

The LINPACK routines are designed for use with single, double or complex precision computations. The first letter of the LINPACK routine name tells the nature of the computations for that routine. If the first letter is an S, then this is a routine to perform single precision computations. If the first letter is a D, then this is a routine to perform double precision computations. If the first letter is a C, then this is a routine to perform complex computations. If an altered LINPACK routine was developed, it will be denoted by an A appended to the beginning of the original LINPACK name. This will mean that for the altered LINPACK routines the second letter will tell the type of computation to be performed. The same notation for single and double precision computations will be upheld for the newly developed special routines, with the first letter of the name indicating the type of computation. None of the complex computational routines were used for this work. Therefore, no discussion of these routines will be included.
The software descriptions will include references to the double precision versions of each routine. The double precision versions will not appear in the appendix due to the similarity to the single precision versions. The primary changes would involve: the definition of variables as double precision, the handling of input and output of double precision values, and changing of constants to their double precision equivalents. Also the obvious changes in calls to subroutines and functions would have to be performed.

Existing Software

The existing software that is the background for this work is part of a large linear algebra subroutine library. LINPACK (Dongarra et al. 1978) is a set of FORTRAN callable subroutines for the manipulation, decomposition, and solution of linear systems of equations.

MATLAB (Moler 1980), which is a system for the manipulation of linear systems, was not chosen for this work. The lack of access to the source code for MATLAB proved to be a major disadvantage. Although MATLAB was not chosen, it should be noted that MATLAB provides an interactive mode of operation, and uses calls to the
LINPACK routines for many computations. This interactive mode of MATLAB was used to check the results of some of the special routines developed for this work.

LINPACK and MATLAB both require the presence of the normal FORTRAN libraries and the Basic Linear Algebra Subroutines library (BLAS). The BLAS routines are usually part of the LINPACK subroutine library. No description of these libraries will be provided.

The LINPACK routines that provide the LU and QR decompositions and solve the associated decomposed systems are of the most interest. For a detailed explanation of the LINPACK routines, the LINPACK user guide should be consulted (Dongarra et al. 1978).

The use of the LINPACK routines is very beneficial for the solution of recurrence relations that have been set up as their associated linear systems of equations. The discussion that follows will briefly explain the function of the routines that were used.

Some of the original LINPACK routines were altered to provide special features of interest. Any of the routines that were altered will be included in the appendix.
LU Decomposition

The first type of decomposition to be considered will be the LU decomposition. LINPACK provides routines SGECO (DGECO) and SGEFA (DGEFA) for the LU decomposition. The routine SGEFA (DGEFA) actually does the decomposition, but if SGECO (DGECO) were used it would provide other information about the input matrix. This information includes the estimated condition number for the original matrix A. This condition number may be helpful in the discussions of stability of solution methods and pivoting concerns.

The LINPACK routines provide a LU decomposition of an by n input matrix A. The result is L, a lower triangular matrix, and U, an upper triangular matrix, that satisfies the equation $A = LU$. The routine uses Gaussian elimination with partial pivoting to acquire the desired matrices.

It should be noted that the LINPACK routines do not return the L and U matrices directly. If the user desires to see the L and U matrices, they must be generated from the information that the decomposition routine returns. The special routine SLUOUT performs the needed transformations to retrieve the L and U matrices if requested.
LINPACK performs all of the needed calculations for the solution using the L and U matrices in the form returned from the decomposition routine. Therefore, this generation of L and U need only be performed if the user desires to see the L and U matrices.

As an enhancement to the original routines, the addition of the option of not performing partial pivoting was included. This was accomplished through the addition of a new parameter JOB. The new routines are given the names ASGECO (ADGECO) and ASGEFA (ADGEFA) to indicate that these are altered versions of the LINPACK routines.

After the decomposition of A into L and U, the solution for the linear system Ax = b may be performed. The LINPACK routine SGESL (DGESL) will take the information from the decomposition, the vector b, and return the vector x. The vector x is the computed solution of the associated linear system for a recurrence relation. It should be noted that the LU decomposition by the LINPACK routines can only be done for determined systems. The QR decomposition should be used if the system to be decomposed and solved is not determined.
QR Decomposition

The second type of decomposition to be considered will be the QR decomposition. LINPACK provides the routine SQRDC (DQRDC) for the QR decomposition.

The LINPACK routine provides a QR decomposition of a \( n \times p \) input matrix \( A \). The result is \( Q \), an orthogonal matrix, and \( R \), an upper triangular matrix, that satisfies the equation \( A = QR \). The routine uses Householder transformations with the option of partial column pivoting to acquire the desired matrices.

It should be noted that the LINPACK routines do not return the \( Q \) and \( R \) matrices directly. If the user desires to see the \( Q \) and \( R \) matrices, they must be generated from the information that the decomposition routine returns. The special routine SQROUT performs the needed transformations to retrieve the \( Q \) and \( R \) matrices if requested.

LINPACK performs all of the needed calculations for the solution using the \( Q \) and \( R \) matrices in the form returned from the decomposition routine. Therefore, this generation of \( Q \) and \( R \) need only be performed if the user desires to see the \( Q \) and \( R \) matrices.

After the decomposition of \( A \) into \( Q \) and \( R \), the solution for the linear system \( Ax = b \) may be performed. The LINPACK routine SQRSL (DQRSL) will take
the information from the decomposition, the vector $b$, and return the vector $x$. The vector $x$ is the computed solution of the associated linear system for a recurrence relation.

**Least Squares**

The third type of solution method is the least squares method. This method utilizes the LINPACK QR decomposition routine SQRDC (DQRDC) and the QR solution routine SQRSL (DQRSL) to perform the least squares computations.

LINPACK provided a routine SQRST (DQRST) in the LINPACK users guide (Dongarra et al. 1978). This routine controls the calling of the QR decomposition and solution routines and does the needed intermediate computations.

The least squares method provides for the even distribution of errors in the computed result. These errors may be either absolute or relative depending on the weighting that was performed. The topic of weighting is covered in chapter five. Because this routine uses the QR decomposition routines, the $Q$ and $R$ matrices are available for display.

It should be noted that the LINPACK routines do not return the $Q$ and $R$ matrices directly. If the user desires to see the $Q$ and $R$ matrices, they must be
generated from the information that the decomposition routine returns. The special routine SLSOUT performs the needed transformations to retrieve the Q and R matrices if requested.

LINPACK performs all of the needed calculations for the solution using the Q and R matrices in the form returned from the decomposition routine. Therefore, this generation of Q and R need only be performed if the user desires to see the Q and R matrices.

Because this routine usually performs computations on overdetermined systems of equations, it is not possible to compare the results with the other solution methods.

New Software

The following are specially designed routines for the input and output of information of interest from the LINPACK routines. All of the FORTRAN code for the following routines will be found in the appendix. As before the first letter of the routine name will determine the precision of the computations for the routine. The inclusion of the reference to the double precision version of each routine is also included. A discussion of the conversion from single to double precision can be found earlier in this chapter.
JOBHLP

The JOBHLP routine is provided to help the user define the JOB parameter. JOBHLP is called from SMTGEN (DMTGEN) if the user requires help with the JOB parameter. JOBHLP will display choices for each value in the JOB parameter, accept the user's choice, do a simple error check on each input, and ask the user if the values are correct before returning to SMTGEN (DMTGEN). If errors are detected or the user does not agree with the values, JOBHLP will redo the parameter. For an explanation of the JOB parameter, the section describing the SSELCT routine, in this chapter, should be consulted.

SAROUT (DAROUT)

SAROUT (DAROUT) is designed to output large two dimensional arrays in a formatted manner with headings. As input the routine will accept a two dimensional array, the dimensions of the array, and the title to appear at the top of each page.

The routine will separate the 2 dimensional array into segments of 8 columns each (segments of 4 columns each for DAROUT). These segments will be printed in lineprinter format (132 column paper), with one segment per page. The title, a descriptive heading, and a page
number will be printed at the top of each page of output. Once the output is obtained, it can be torn apart and laid out to present the array as it logically appears. The routine works for one dimensional arrays as well as two dimensional arrays, with the exception that a one dimensional array is displayed vertically in all cases.

**SCPOUT (DCPOUT)**

SCPOUT (DCPOUT) writes to the output file FOR22 a comparison of the solution methods that were selected for the current execution. SCPOUT (DCPOUT) is passed the following information: the COMPAR array with all of the computed values, the CONDS array of initial and terminal conditions, the number of values computed for the comparison, and the TITLE to appear on the output file FOR22.

SCPOUT (DCPOUT) will adjust the COMPAR array if the reduced system of equations was used. This adjustment is for the inclusion of the initial and terminal conditions.

Once the initial and terminal conditions are included, the COMPAR array will be condensed to make output easier. The condensing will involve moving all of the columns to the left to fill any columns that were unused.
Once the number of active columns has been determined, the SRIGHT (DRIGHT) routine is called to provide an accurate solution to the recurrence relation for the current solution. SRIGHT (DRIGHT) will place the accurate solution in the first available column of COMPAR. This choice of column will keep the desired values for output left justified in COMPAR.

SCPOUT (DCPOUT) will output the following 4 tables of information about the requested computations.

1. The values returned from LINPACK for the condition number for each solution method. These values are the reciprocals of the estimated condition numbers.

2. A table of the computed values with column labels.

3. A table of the absolute errors as compared to the accurate values provided by SRIGHT (DRIGHT) with column labels.

4. A table of the relative errors as compared to the accurate values provided by SRIGHT (DRIGHT) with column labels.

Each of the tables will be titled and formatted so that they can be used in many ways. A strict format for the headings is not provided so that the results can be reformatted easily to fit the needs of the user. Most of the results presented throughout this work involved the reformation of the results from SCPOUT (DCPOUT).
SGTVAL (DGTVAL)

SGTVAL (DGTVAL) is called by SMTGEN (DMTGEN) and returns the values for one row of the matrix A. The routine is passed a vector C to return the values in, N the current row being computed, and the value TAU, which is a user provided value that may be used in the computations. The contents of SGTVAL (DGTVAL) are unique for each recurrence relation; therefore, this routine must be altered for each new recurrence relation.

SLSOUT (DLSOUT)

SLSOUT (DLSOUT) is the routine to provide the least squares calculations for the solution to recurrence relations. SLSOUT (DLSOUT) uses the QR decomposition in its computations.

The principle parameters passed to SLSOUT (DLSOUT) are the following: the original matrix A, the result vector b, and the JOB parameter. The JOB parameter will determine what to perform and print to the output file FOR22. JOB is a three digit number with the following form.

1. The first digit tells if the results of the decomposition are to be printed to the output file FOR22 as a Q and R matrix. If this digit is a one,
then Q and R are printed to the output file FOR22. If the first digit is zero, no output of Q and R will occur.

2. The second digit tells if the computed results are to be printed to the output file FOR22. If this digit is a one, then the results will be printed to the output file FOR22. If this digit is zero, no output of the calculated results will occur.

3. The third digit is not used and is set to 0.

SLSOUT (DLSOUT) performs weighting of the matrix A and the vector b before the least squares solution is computed. SLSSCL (DLSSCL) will be called to provide the weights. These weights will then be applied to the matrix A and the vector b. If no weighting is desired, then SLSSCL (DLSSCL) should be written to return a vector of all ones.

The JOB parameter will be separated and used to call the proper routines to perform the least squares calculations and print the requested results. On return any computed results will be returned along with the estimated condition number of the matrix A. There is no option for pivoting in this routine due to the fact that the least squares calculations using the QR decomposition always uses the option of pivoting.
SLSSCL (DLSSCL)

SLSSCL (DLSSCL) is the routine to provide the weighting constants for the SLSOUT (DLSOUT) routine. SLSSCL (DLSSCL) assigns a vector of constants to scale the matrix A and result vector b before the least squares calculations are performed. A user defined value TAU and the number of values to compute are also available for the assigning of weights. If there is to be no weighting performed, the vector returned should contain all ones. If weighting is to occur, all the values in the weighting vector should be one except for the values that represent the rows to be weighted. On return the matrix A and vector b are multiplied by the vector of constants. This routine is unique and must be changed for each different weighting situation.

SLUOUT (DLUOUT)

SLUOUT (DLUOUT) performs the required operations for the LU decomposition method. The routine will be provided the following as parameters: the matrix A to decompose and solve, the result vector b, and the JOB parameter that will tell which options to perform and what to print to the output file FOR22.

The job parameter is a three digit number consisting of the following:
1. The first digit tells if the decomposition of $A$ into $L$ and $U$ is to be printed. If the first digit is a one, then the $A$ matrix is decomposed and the $L$ and $U$ matrices are printed to the output file FOR22. If the first digit is a zero, no output of $L$ and $U$ will occur.

2. The second digit tells if the solution is to be computed and printed. If the second digit is a one, then the solution is computed and the results are printed to the output file FOR22. If the second digit is a zero, then the results of the computations will not be printed.

3. The third digit tells if the routine is to use partial pivoting. If the third digit is a one, then partial pivoting is used in the decomposition and solution of the linear system. If the third digit is a zero, then no partial pivoting will be used in the computations.

On return, SLUOUT (DLUOUT) will provide the computed results and the estimated condition number of $A$ for later comparisons. The output file FOR22 will contain all of the information requested through the JOB parameter in a neat formatted manner.
SLUSCL (DLUSCL)

SLUSCL (DLUSCL) is a routine to provide the weighting factors for the SLUOUT (DLUOUT) routine. The weighting involved in the LU decomposition is restricted to the initial conditions. A vector that contains the weighting values to apply to the matrix A and the result vector b, before the LU decomposition is performed, is assigned. The vector returned should contain ones except in the first i rows, where i is the number of initial conditions for the recurrence relation. The values in the first i rows is dependent on the desired weighting for the recurrence relation. This routine is unique and must be changed for each recurrence relation and solution.

SMTGEN (DMTGEN)

SMTGEN (DMTGEN) is a routine that generates the matrix A from supplied information. This information may come from one of two sources. Upon entry to the routine the user will be prompted to determine which source SMTGEN (DMTGEN) will use. The options are: to read an existing file FOR20 that contains the necessary information (see SMTGEN documentation for the format of this file), or to create a new data file FOR20 interactively that can be reused at a later time.
The routine creates either the reduced linear system (2.30) or the augmented linear system (2.30). The values for the matrix A are obtained from repeated calls to SGTVAL (DGTVAL).

The file FOR20 will contain the following information: a title for the output file, a value TAU that can be used for some recurrence relation calculations if needed, the number of values for the recurrence relation to calculate, the order of the recurrence relation, the number and values for the initial conditions, the number and values for the terminal conditions, and the parameter JOB. This parameter determines what the rest of the software will do with the previously supplied values, and what output will be generated. A special routine JOBHLP is available to help the user in the creation of this JOB parameter. If the option to create a new input file is selected, then all of the above information will be provided interactively by the user.

After SMTGEN (DMTGEN) obtains the needed information, an output file FOR21 will be created (see SMTGEN documentation for the format of this file). The output file FOR21 will contain the following information: the title to appear on all output, the number of rows in the matrix A, the number of columns in
the matrix A, the JOB parameter, the matrix A, the vector b, and a special array of the initial and terminal conditions.

Once the SMTGEN (DMTGEN) routine has been executed, the output file in FOR21 is ready to be used by the rest of the software for the decomposition and solution of the linear system that represents the recurrence relation.

**SORLST (DQRLST)**

**SORLST (DQRLST)** is the routine to compute the least squares solution for a linear system of equations. This routine is not part of the LINPACK subroutine library, but was adapted from the LINPACK users manual (Dongarra et al. 1978).

**SOROUT (DQROUT)**

**SOROUT (DQROUT)** performs the required operations for the QR decomposition method. The routine will be provided the following as parameters: the matrix A to decompose and solve, the result vector b, and the JOB parameter that will tell which options to perform and what to print to the output file FOR22.

The JOB parameter is a three digit number consisting of the following.
1. The first digit tells if the decomposition of $A$ into $Q$ and $R$ is to be printed. If the first digit is a one, then the $A$ matrix is decomposed and the $Q$ and $R$ matrices are printed to the output file FOR22. If the first digit is a zero, no output of $Q$ and $R$ will occur.

2. The second digit tells if the solution is to be computed and printed. If the second digit is a one, then the solution is computed and the results are printed to the output file FOR22. If the second digit is a zero, then the results of the computations will not be printed.

3. The third digit tells if the routine is to use partial pivoting. If the third digit is a one, then partial pivoting is used in the decomposition and solution of the linear system. If the third digit is a zero, then no partial pivoting will be used in the computations.

On return, SQROUT (DQRUT) will provide the computed results and the estimated condition number of $A$ for later comparisons. The output file FOR22 will contain all of the information requested through the JOB parameter in a neat formatted manner.
SRECUR (DRECUR)

SRECUR (DRECUR) is the main program that links all of the software together. This routine must be loaded with: the other routines in this chapter, the BLAS library, the LINPACK library, and the normal FORTRAN subroutine library. Once the program executes, the user will be prompted to provide the source of the input data. There are two choices at this point: call the SMTGEN (DMTGEN) routine to obtain the data, or read the data from FOR21, which was previously created using SMTGEN (DMTGEN).

Once the data has been obtained, SRECUR (DRECUR) will call SSELCT (DSELCT) to perform the needed operations. The JOB parameter is passed to SSELCT (DSELCT) to provide the information on what computations to perform and what to write to the output file FOR22.

SRECUR (DRECUR) is also responsible for calling the SCPOUT (DCPOUT) routine to output the results of the different solution methods that were selected.

SRIGHT (DRIGHT)

SRIGHT (DRIGHT) is the routine to provide accurate values to perform the error calculations in SCPOUT (DCPOUT). SRIGHT (DRIGHT) is provided with the COMPAR array and a value to tell which column of the COMPAR...
array to place the accurate values in. The computations of the accurate values makes this a unique routine for each solution to a recurrence relation; therefore, this routine must be changed for each solution.

**SSELCT (DSELCT)**

SSELCT (DSELCT) is designed to select which of the decomposition methods are to be used. The routine is passed a parameter JOB, which is a seven digit number. Each of the first six digits will correspond, in order, to each of the following decomposition methods:

1. LU decomposition without pivoting,
2. LU decomposition with pivoting,
3. QR decomposition without pivoting,
4. QR decomposition with pivoting,
5. QR decomposition and least squares, and
6. LU decomposition and weighted initial conditions.

The seventh digit will determine if the matrix A and vector b are to be printed to the output file FOR22.

The first six digits, have the following values and the associated meanings:

0. This method is ignored.
1. The decomposition of A is performed and the results of the decomposition are printed.
2. The decomposition of A is performed, the system is solved, and the solution is printed.
3. The decomposition of A is performed, the system is solved, and both the solution and the decomposition are printed.

The seventh digit of the JOB parameter has the following two options:

0. The matrix A and vector b will not be printed to the output file FOR22.

1. Both the matrix A and the vector b will be printed to the output file FOR22.

The routine, through the JOB parameter, will call the appropriate routines and on return will have printed all information that was requested and have placed any solutions in the COMPAR array for later comparisons.

SZCOPY (DZCOPY)

SZCOPY (DZCOPY) is designed to copy n by p two dimensional arrays. The routine serves two purposes. The first purpose is to restore the matrix A and the vector b to their initial conditions after each solution is obtained. The second purpose is to copy the computed results of a solution for a particular method into a designated column of the COMPAR array for a later comparison of solutions.
Sample Execution of Software

This section provides an exposure to the software. A sample execution will be presented to show the expected results for both the interactive questions, and the data files that are generated.

An explanation of the recurrence relation for the sample execution can be found in examples 2.2 and 2.28. The solution of $l^n$ will be computed using four different solution methods.

The use of underlining will distinguish the system dependent commands, system messages, and user input information, from the information being displayed by the software.

The first section of the sample execution shows the execute command and software welcome messages. This command links and executes the software found in the appendix and the LINPACK library routines. After these messages are displayed, the source for the input information will be determined. This example shows the acceptance of the initial information from the terminal. The job help option is also shown for constructing the job parameter. Once all of the information is accepted, the software creates the files FOR20, FOR21, and FOR22. At this point the end of execution message is given and control is returned to the user.
The remainder of the sample execution is concerned with looking at the data files that were created. The first file to be displayed is the file FOR20. Consult the SMTGEN documentation for an explanation of the contents of this file and FOR21. The second file displayed is FOR21. Notice the creation of the augmented matrix, and the slight reformatting of the CONDS array. Again consult SMTGEN for details on the contents of FOR21.

The third data file to be displayed is FOR22. This file contains all of the results from the software. The results presented in FOR22 are those requested through the JOB parameter. Notice the 1's in the first column before each separate result. These 1's represent a new page if this file were printed on a lineprinter.

The following sample execution is provided to show some of the expected results, format of information, and phrasing of questions. The lack of strict format for headings, in the numeric results, is intended to allow easy reformation of the results. Most of the results for examples in this work resulted from a reformation of the results in FOR22.
EXECUTE APPEN.FOR, PUB:LINPAK/LIB

LINK: Loading

LINKXCT APPEN execution

WELCOME TO THE LINEAR RECURRENCE RELATION TESTING SOFTWARE ENVIRONMENT. WRITTEN BY DENNIS KAPENGA, WESTERN MICHIGAN UNIVERSITY. FOR MORE INFORMATION CONSULT THE ASSOCIATED THESIS: AN INVESTIGATION INTO THE NUMERIC SOLUTION OF LINEAR RECURRENCE RELATIONS.

WOULD YOU LIKE TO READ AN EXISTING FILE FROM FOR21 THAT CONTAINS THE MATRIX AND INITIAL INFORMATION (Y) OR WOULD YOU LIKE TO CREATE ONE USING THE SMTGEN (MATRIX GENERATING) ROUTINE (N)? N

WOULD YOU LIKE TO READ AN EXISTING FILE FOR20 THAT CONTAINS THE INITIAL INFORMATION (Y) OR CREATE A NEW FILE FOR20 INTERACTIVELY (N)? N

PLEASE ENTER THE TITLE YOU WOULD LIKE TO USE FOR THE OUTPUT (80 CHAR MAX) SAMPLE TEST OF SOFTWARE

PLEASE ENTER THE NUMBER OF VALUES TO COMPUTE 8

PLEASE ENTER THE ORDER OF THE RECURRENCE RELATION 2

PLEASE ENTER THE TYPE OF MATRIX TO GENERATE: AUGMENTED (0), REDUCED (1) 0

PLEASE ENTER THE JOB PARAMETER OR 0 FOR HELP 0

JOB HELP

ANSWER THE FOLLOWING QUESTIONS IN ORDER TO PRODUCE THE JOB PARAMETER

FOR THE FIRST SIX OPTIONS, THE FOLLOWING VALUES WILL BE USED FOR INPUT

0 SKIPTHE PARTICULAR OPTION
1 PERFORM THE SPECIFIED DECOMPOSITION AND PRINT THE DECOMPOSED SYSTEM ONLY
2 PERFORM THE SPECIFIED DECOMPOSITION, SOLVE AND PRINT THE SOLUTION ONLY
3 PERFORM THE SPECIFIED DECOMPOSITION, SOLVE AND PRINT BOTH THE DECOMPOSED SYSTEM AND THE SOLUTION
ENTER A VALUE FOR THE LU DECOMPOSITION WITH NO PARTIAL PIVOTING (0,1,2,3) 3
ENTER A VALUE FOR THE LU DECOMPOSITION WITH PARTIAL PIVOTING (0,1,2,3) 2
ENTER A VALUE FOR THE QR DECOMPOSITION WITH NO PARTIAL PIVOTING (0,1,2,3) 3
ENTER A VALUE FOR THE QR DECOMPOSITION WITH PARTIAL PIVOTING (0,1,2,3) 2
ENTER A VALUE FOR THE QR DECOMPOSITION WITH PARTIAL PIVOTING AND LEAST SQUARES APPROXIMATIONS (0,1,2,3) 0
ENTER A VALUE FOR THE LU DECOMPOSITION WITH PARTIAL PIVOTING AND WEIGHTED INITIAL CONDITIONS (0,1,2,3) 0

WOULD YOU LIKE THE ORIGINAL MATRIX A AND VECTOR b TO APPEAR AT THE BEGINNING OF THE OUTPUT FILE (Y/N) Y

YOU HAVE PRODUCED A VALUE OF 3232001 FOR THE JOB PARAMETER. IS THIS VALUE CORRECT (Y/N)? Y

PLEASE ENTER THE NUMBER OF INITIAL CONDITIONS DESIRED 2
PLEASE ENTER THE NUMBER OF TERMINAL CONDITIONS DESIRED 0
PLEASE ENTER THE INITIAL CONDITION FOR F(0) 1.0
PLEASE ENTER THE INITIAL CONDITION FOR F(1) 1.0
PLEASE ENTER A VALUE FOR TAU 0.0 IF NOT NEEDED 0.0

End of Execution
CPU time 1.15 Elapsed time 2:46.52
EXIT
**TYPE FOR22.DAT**

1 PAGE 1

**SAMPLE TEST OF SOFTWARE**

**THIS IS THE ORIGINAL MATRIX A AND VECTOR b**

<table>
<thead>
<tr>
<th></th>
<th>b=</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
</tbody>
</table>

1 PAGE 1

**SAMPLE TEST OF SOFTWARE**

**THE L MATRIX WITH NO PARTIAL PIVOTING**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>-0.2000000E+01</td>
<td>-0.3000000E+01</td>
</tr>
<tr>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>0.2000000E+01</td>
<td>0.3000000E+01</td>
</tr>
</tbody>
</table>

1 PAGE 1

**SAMPLE TEST OF SOFTWARE**

**THE U MATRIX WITH NO PARTIAL PIVOTING**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>0.1000000E+01</td>
<td>0.0000000E+00</td>
</tr>
</tbody>
</table>

194
THE SOLUTION FOR $Ay=b$ USING LU WITH NO PIVOTING

\[
\begin{align*}
0.10000000E+01 \\
0.10000000E+01 \\
0.10000000E+01 \\
0.10000000E+01 \\
0.10000000E+01 \\
0.10000000E+01 \\
0.10000000E+01 \\
0.10000000E+01 \\
\end{align*}
\]

THE SOLUTION FOR $Ay=b$ USING LU WITH PIVOTING

\[
\begin{align*}
0.10000000E+01 \\
0.10000000E+01 \\
0.10000000E+01 \\
0.10000000E+01 \\
0.10000000E+01 \\
0.10000000E+01 \\
0.10000000E+01 \\
0.10000000E+01 \\
\end{align*}
\]

THE Q MATRIX WITH NO PARTIAL PIVOTING

\[
\begin{align*}
-0.44721359E+00 & -0.46017895E+00 & 0.23337823E+00 & 0.17910821E+00 & 0.11886877E+00 & 0.84685441E+01 & 0.14666241E+00 & 0.65311275E+00 \\
0.00000000E+00 & -0.36342536E+00 & 0.29647046E+00 & 0.17936520E+00 & 0.19387853E+00 & 0.74831205E+01 & 0.32625541E+00 & 0.75725484E+00 \\
-0.83442719E+00 & 0.23058500E+00 & -0.14668912E+00 & -0.89531048E+01 & -0.59073478E+01 & -0.42482721E+01 & -0.74831205E+01 & -1.32625541E+00 \\
0.00000000E+00 & -0.76954545E+00 & 0.40621600E+00 & -0.27919497E+00 & -0.20094689E+00 & -0.15289225E+00 & -0.29140903E+00 & -0.16386548E+00 \\
0.00000000E+00 & 0.00000000E+00 & 0.76758645E+00 & -0.37401938E+00 & -0.21717380E+00 & -0.26331253E+00 & -0.39492005E+00 & -0.77515106E+00 \\
0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.81124574E+00 & -0.30718301E+00 & -0.23397422E+00 & -0.43648516E+00 & -0.36284046E+00 \\
0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.85065744E+00 & -0.24993056E+00 & -0.47523383E+00 & -0.15941374E+00 \\
0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.87984292E+00 & -0.47523383E+00 & -0.51834397E+00 \\
\end{align*}
\]

THE R MATRIX WITH NO PARTIAL PIVOTING

\[
\begin{align*}
-0.22206176E+01 & -0.26329161E+01 & -0.38447719E+01 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 \\
0.00000000E+00 & -0.26076634E+01 & 0.25308948E+01 & -0.76664898E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 \\
0.00000000E+00 & 0.00000000E+00 & 0.26036582E+01 & -0.27051068E+01 & 0.76725495E+00 & 0.00000000E+00 & 0.00000000E+00 \\
0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.24684424E+01 & -0.25077551E+01 & 0.81124574E+00 & 0.00000000E+00 \\
0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.23511227E+01 & -0.28971542E+00 & 0.85065744E+00 \\
0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.27731939E+00 & -0.29391757E+00 \\
0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.96388928E+00 \\
0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.00000000E+00 & 0.51834397E+00 \\
\end{align*}
\]
THE SOLUTION FOR $Ay = b$ USING QR WITH NO PIVOTING

$0.99999997E+00$
$0.99999996E+00$
$0.99999992E+00$
$0.99999986E+00$
$0.99999946E+00$
$0.99999892E+00$
$0.99999785E+00$

THE SOLUTION FOR $Ay = b$ USING QR WITH PIVOTING

$0.10000001E+01$
$0.10000001E+01$
$0.10000001E+01$
$0.10000001E+01$
$0.10000002E+01$
$0.10000002E+01$
$0.10000004E+01$

THE FOLLOWING TABLE IS FOR THE CONDITION NUMBERS

$0.85721997E-03$ IS THE RECIPROCAL OF THE CONDITION NUMBER FOR THE LU DECOMPOSITION WITHOUT PIVOTING

$0.85721957E-03$ IS THE RECIPROCAL OF THE CONDITION NUMBER FOR THE LU DECOMPOSITION WITH PIVOTING

$0.23181047E-02$ IS THE RECIPROCAL OF THE CONDITION NUMBER FOR THE QR DECOMPOSITION WITHOUT PIVOTING

$0.13853310E-02$ IS THE RECIPROCAL OF THE CONDITION NUMBER FOR THE QR DECOMPOSITION WITH PIVOTING
## Sample Test of Software

The following table is for the actual computed values with the following column labels:

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Represents the LU decomposition without pivoting</td>
<td>Represents the LU decomposition with pivoting</td>
<td>Represents the QR decomposition without pivoting</td>
<td>Represents the QR decomposition with pivoting</td>
<td>Represents the estimate of exact value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.100000E+01</td>
<td>0.100000E+01</td>
<td>0.99999997E+00</td>
<td>0.100000E+01</td>
<td>0.100000E+01</td>
</tr>
<tr>
<td>1.000000E+01</td>
<td>1.000000E+01</td>
<td>1.000000E+01</td>
<td>1.000000E+01</td>
<td>1.000000E+01</td>
</tr>
<tr>
<td>2.000000E+01</td>
<td>2.000000E+01</td>
<td>2.000000E+01</td>
<td>2.000000E+01</td>
<td>2.000000E+01</td>
</tr>
<tr>
<td>3.000000E+01</td>
<td>3.000000E+01</td>
<td>3.000000E+01</td>
<td>3.000000E+01</td>
<td>3.000000E+01</td>
</tr>
<tr>
<td>4.000000E+01</td>
<td>4.000000E+01</td>
<td>4.000000E+01</td>
<td>4.000000E+01</td>
<td>4.000000E+01</td>
</tr>
<tr>
<td>5.000000E+01</td>
<td>5.000000E+01</td>
<td>5.000000E+01</td>
<td>5.000000E+01</td>
<td>5.000000E+01</td>
</tr>
<tr>
<td>6.000000E+01</td>
<td>6.000000E+01</td>
<td>6.000000E+01</td>
<td>6.000000E+01</td>
<td>6.000000E+01</td>
</tr>
<tr>
<td>7.000000E+01</td>
<td>7.000000E+01</td>
<td>7.000000E+01</td>
<td>7.000000E+01</td>
<td>7.000000E+01</td>
</tr>
</tbody>
</table>

## Sample Test of Software

The following table is for the absolute errors with the following column labels:

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Represents the LU decomposition without pivoting</td>
<td>Represents the LU decomposition with pivoting</td>
<td>Represents the QR decomposition without pivoting</td>
<td>Represents the QR decomposition with pivoting</td>
<td>Represents the estimate of exact value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000E+00</td>
<td>0.14901161E-07</td>
<td>0.28902322E-07</td>
<td>0.74505808E-07</td>
<td>0.10000000E+01</td>
</tr>
<tr>
<td>1.000000E+00</td>
<td>1.000000E+00</td>
<td>1.000000E+00</td>
<td>1.000000E+00</td>
<td>1.000000E+00</td>
</tr>
<tr>
<td>2.000000E+00</td>
<td>2.000000E+00</td>
<td>2.000000E+00</td>
<td>2.000000E+00</td>
<td>2.000000E+00</td>
</tr>
<tr>
<td>3.000000E+00</td>
<td>3.000000E+00</td>
<td>3.000000E+00</td>
<td>3.000000E+00</td>
<td>3.000000E+00</td>
</tr>
<tr>
<td>4.000000E+00</td>
<td>4.000000E+00</td>
<td>4.000000E+00</td>
<td>4.000000E+00</td>
<td>4.000000E+00</td>
</tr>
<tr>
<td>5.000000E+00</td>
<td>5.000000E+00</td>
<td>5.000000E+00</td>
<td>5.000000E+00</td>
<td>5.000000E+00</td>
</tr>
<tr>
<td>6.000000E+00</td>
<td>6.000000E+00</td>
<td>6.000000E+00</td>
<td>6.000000E+00</td>
<td>6.000000E+00</td>
</tr>
<tr>
<td>7.000000E+00</td>
<td>7.000000E+00</td>
<td>7.000000E+00</td>
<td>7.000000E+00</td>
<td>7.000000E+00</td>
</tr>
</tbody>
</table>
## SAMPLE TEST OF SOFTWARE

The following table is for the relative errors with the following column labels:

**Column 1** represents the LU decomposition without pivoting.

**Column 2** represents the LU decomposition with pivoting.

**Column 3** represents the QR decomposition without pivoting.

**Column 4** represents the QR decomposition with pivoting.

**Column 5** represents the estimate of exact value.

<table>
<thead>
<tr>
<th></th>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00000000E+00</td>
<td>0.14901161E-07</td>
<td>0.74505800E-07</td>
<td>0.10000000E+01</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.00000000E+00</td>
<td>0.14901161E-07</td>
<td>0.10430139E-07</td>
<td>0.10000000E+01</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.00000000E+00</td>
<td>0.29802322E-07</td>
<td>0.13411045E-07</td>
<td>0.10000000E+01</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.00000000E+00</td>
<td>0.59604645E-07</td>
<td>0.13411045E-06</td>
<td>0.10000000E+01</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.00000000E+00</td>
<td>0.11920529E-06</td>
<td>0.17831358E-06</td>
<td>0.10000000E+01</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.00000000E+00</td>
<td>0.23841058E-06</td>
<td>0.22517429E-06</td>
<td>0.10000000E+01</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.00000000E+00</td>
<td>0.47683170E-06</td>
<td>0.39743055E-06</td>
<td>0.10000000E+01</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.00000000E+00</td>
<td>0.95367412E-06</td>
<td>0.71528574E-06</td>
<td>0.10000000E+01</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX

The following appendix is an alphabetical list of the software routines developed as part of this study. Any existing routines from the LINPACK library that were altered, to provide special functions, are included for reference. For a brief description of the routines and a sample execution of the software, chapter six should be consulted. Most of the results presented throughout this work were a reformation of the results from these routines.
SUBROUTINE ASGECO(A,LDA,N,IPVT,RCOND,Z,JOB)
C
INTEGER LDA,N,IPVT(1),JOB
REAL A(LDA,1),Z(1),RCOND
C
ASGECO FACTORS A REAL MATRIX BY GAUSSIAN
ELIMINATION WITH OR WITHOUT PARTIAL PIVOTING
AND ESTIMATES A CONDITION NUMBER FOR THE
MATRIX.
C
THIS ROUTINE IS THE SAME AS SGECO EXCEPT FOR
THE ADDITION OF THE OPTION OF NOT PIVOTING.
C
IF RCOND IS NOT NEEDED, ASGEFA IS SLIGHTLY
FASTER.
TO SOLVE A*X = B, FOLLOW ASGECO BY SGESL.
TO COMPUTE INVERS(A)*C, FOLLOW ASGECO BY SGESL.
TO COMPUTE DET(A), FOLLOW ASGECO BY SGEDI.
TO COMPUTE INVERSE(A), FOLLOW ASGECO BY SGEDI.
C
ON ENTRY
A       REAL(LDA,N)
       THE MATRIX TO BE FACTORED.
LDA     INTEGER
       THE LEADING DIMENSION OF THE MATRIX A.
N       INTEGER
       THE ORDER OF THE MATRIX A.
JOB     INTEGER
       PARAMETER TO TELL IF PARTIAL PIVOTING
       IS TO BE DONE. IF JOB IS 0 THEN NO
       PIVOTING IS DONE, AND IF JOB IS 1 THEN
       PARTIAL PIVOTING IS TO BE USED.
C
ON RETURN
A       AN UPPER TRIANGULAR MATRIX AND THE
       MULTIPLIERS WHICH WERE USED TO OBTAIN
       IT. THE DECOMPOSITION CAN BE WRITTEN
       A = L*U WHERE L IS A PRODUCT OF THE
       PERMUTATION AND THE UNIT LOWER
       TRIANGULAR MATRICES AND U IS UPPER
       TRIANGULAR.
IPVT    INTEGER(N)
       AN INTEGER VECTOR OF PIVOT INDICES.
RCOND REAL
THE RECIPROCAL OF THE ESTIMATED
CONDITION OF A. FOR THE SYSTEM
A*X = B, RELATIVE PERTURBATIONS
IN A AND B OF SIZE EPSILON MAY
CAUSE RELATIVE PERTURBATIONS IN X OF
SIZE EPSILON/RCOND.
IF RCOND IS SO SMALL THAT THE
LOGICAL EXPRESSION
1.0 + RCOND .EQ. 1.0
IS TRUE, THEN A MAY BE SINGULAR TO
WORKING PRECISION. IN PARTICULAR,
RCOND IS ZERO IF EXACT SINGULARITY
IS DETECTED OR THE ESTIMATE UNDERFLOWS.

Z REAL(N)
A WORK VECTOR WHOSE CONTENTS ARE
USUALLY UNIMPORTANT. IF A IS CLOSE TO
A SINGULAR MATRIX, THEN Z IS AN
APPROXIMATE NULL VECTOR IN THE SENSE
THAT NORM(A*Z) = RCOND*NORM(A)*NORM(Z).

SGECO LINPACK. THIS VERSION DATED 08/14/78.
CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE
NATIONAL LAB.

ASGECO VERSION UPDATED 7/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY/

SUBRoutines AND FUNCTIONS
UPDATED LINPACK ASGEFA
BLAS: SAXPY,SDOT,SSCAL,SASUM
FORTRAN: ABS,AMAX1,SIGN

INTERNAL VARIABLES

REAL SDOT,EK,T,WK,WKM
REAL ANORM,S,SASUM,SM,YNORM
INTEGER INFO,J,K,KB,KP1,L

WHERE

MOST OF THE INTERNAL VARIABLES ARE
FUNCTIONS, TEMPS AND LOOP COUNTERS.

COMPUTE 1-NORM OF A.
ANORM = 0.0E0

DO 10 J = 1, N
   ANORM = AMAX1(ANORM, SASUM(N, A(1,J), 1))
10 CONTINUE

FACTOR THE MATRIX.

CALL ASGEFA(A, LDA, N, IPVT, INFO, JOB)

RCOND = 1/(NORM(A) * (ESTIMATE OF NORM(INVERSE(A))))
ESTIMATE = NORM(Z)/NORM(Y) WHERE A*Z = Y AND
TRANS(A)*Y = E. TRANS(A) IS THE TRANSPOSE OF A.
THE COMPONENTS OF E ARE CHOSEN TO CAUSE
MAXIMUM LOCAL GROWTH IN THE ELEMENTS OF W
WHERE
TRANS(U)*W = E. THE VECTORS ARE FREQUENTLY
RESCALED TO AVOID OVERFLOW.

SOLVE TRANS(U)*W = E.

EK = 1.0E0

DO 20 J = 1, N
   Z(J) = 0.0E0
20 CONTINUE

DO 100 K = 1, N
   IF (Z(K) .NE. 0.0E0) EK = SIGN(EK, -Z(K))
   IF (ABS(EK-Z(K)) .LE. ABS(A(K,K))) GO TO 30
      S = ABS(A(K,K))/ABS(EK-Z(K))
      CALL SSCAL(N, S, Z, 1)
      EK = S*EK
30 CONTINUE

WK = EK - Z(K)
WKM = -EK - Z(K)
S = ABS(WK)
SM = ABS(WKM)

IF (A(K,K) .EQ. 0.0E0) GO TO 40
   WK = WK/A(K,K)
   WKM = WKM/A(K,K)
GO TO 50
40 CONTINUE

WK = 1.0E0
WKM = 1.0E0

50 CONTINUE
KP1 = K + 1
IF (KP1 .GT. N) GO TO 90

DO 60 J = KP1, N
SM = SM + ABS(Z(J)+WKM*A(K,J))
Z(J) = Z(J) + WK*A(K,J)
S = S + ABS(Z(J))
60 CONTINUE

IF (S .GE. SM) GO TO 80
T = WKM - WK
WK = WKM

DO 70 J = KP1, N
Z(J) = Z(J) + T*A(K,J)
70 CONTINUE
80 CONTINUE
90 CONTINUE

Z(K) = WK
100 CONTINUE

S = 1.0E0/SASUM(N,Z,1)
call sscal(N,S,Z,1)

SOLVE TRANS(L)*Y = W.

DO 120 KB = 1, N
K = N + 1 - KB
IF (K .LT. N) Z(K) = Z(K) +
SDOT(N-K,A(K+1,K),1,Z(K+1),1)
IF (ABS(Z(K)) .LE. 1.0E0) GO TO 110
S = 1.0E0/ABS(Z(K))
call sscal(N,S,Z,1)
110 CONTINUE

L = IPVT(K)
T = Z(L)
Z(L) = Z(K)
Z(K) = T
120 CONTINUE

S = 1.0E0/SASUM(N,Z,1)
call sscal(N,S,Z,1)

YNORM = 1.0E0

SOLVE L*V = Y.
DO 140 K = 1, N
   L = IPVT(K)
   T = Z(L)
   Z(L) = Z(K)
   Z(K) = T
   IF (K .LT. N) CALL SAXPY
       (N-K,T,A(K+1,K),1,Z(K+1),1)
   IF (ABS(Z(K)) .LE. 1.0E0) GO TO 130
       S = 1.0E0/ABS(Z(K))
       CALL SSCAL(N,S,Z,1)
       YNORM = S*YNORM
130 CONTINUE
140 CONTINUE
C
   S = 1.0E0/SASUM(N,Z,1)
   CALL SSCAL(N,S,Z,1)
   YNORM = S*YNORM
C
   SOLVE U*Z = V.
C
   DO 160 KB = 1, N
      K = N + 1 - KB
      IF (ABS(Z(K)) .LE. ABS(A(K,K))) GO TO 150
          S = ABS(A(K,K))/ABS(Z(K))
          CALL SSCAL(N,S,Z,1)
          YNORM = S*YNORM
150 CONTINUE
C
      IF (A(K,K) .NE. 0.0E0) Z(K) = Z(K)/A(K,K)
      IF (A(K,K) .EQ. 0.0E0) Z(K) = 1.0E0
      T = -Z(K)
      CALL SAXPY(K-1,T,A(1,K),1,Z(1),1)
160 CONTINUE
C
   MAKE ZNORM = 1.0.
C
   S = 1.0E0/SASUM(N,Z,1)
   CALL SSCAL(N,S,Z,1)
   YNORM = S*YNORM
C
   IF (ANORM .NE. 0.0E0) RCOND = YNORM/ANORM
   IF (ANORM .EQ. 0.0E0) RCOND = 0.0E0
C
   RETURN
END
SUBROUTINE, BUT IT DOES INDICATE
THAT SGESL OR SGEDI
WILL DIVIDE BY ZERO IF CALLED.
USE RCOND IN ASGECO FOR A
RELIABLE INDICATION OF
SINGULARITY.

SGEFA LINPACK. THIS VERSION DATED 08/14/78
CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE
NATIONAL LAB.

ASGEFA VERSION UPDATED 12/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

SUBROUTINES AND FUNCTIONS

BLAS: SAXPY, SSCAL, ISAMAX

INTERNAL VARIABLES

REAL T
INTEGER ISAMAX, J, K, KPI, L, NM1

WHERE

T, J, K, L, KPI, NMI
USED AS LOOP COUNTERS AND TEMPS.

GAUSSIAN ELIMINATION WITH OPTIONAL PARTIAL
PIVOTING.

INFO = 0
NM1 = N - 1

IF (NM1 .LT. 1) GO TO 90

DO 60 K = 1, NM1
   KPI = K + 1
   FIND L = PIVOT INDEX.
   IF JOB IS 0 SKIP THE FINDING OF THE PIVOT
   AND ASSIGN PIVOTS IN NUMERIC ORDER.
   IF (JOB .EQ. 1) GO TO 10
   L = K
   GOTO 20
   10 L = ISAMAX(N-K+1, A(K,K), 1) + K - 1

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
SUBROUTINE ASGEFA(A,LDA,N,IPVT,INFO,JOB)

INTEGER LDA,N,IPVT(1),INFO,JOB
REAL A(LDA,1)

ASGEFA FACTORS A REAL MATRIX BY GAUSSIAN
ELIMINATION WITH OR WITHOUT PARTIAL PIVOTING.

THIS ROUTINE IS THE SAME AS SGEFA EXCEPT FOR
THE ADDITION OF THE OPTION OF NOT PIVOTING.

ASGEFA IS USUALLY CALLED BY ASGECO, BUT CAN BE
CALLED DIRECTLY WITH A SAVING IN TIME IF RCOND
IS NOT NEEDED.
(TIME FOR SGEFA) = (1 + 9/N)*(TIME FOR SGEFA) .

ON ENTRY

A  REAL(LDA,N)
    THE MATRIX TO BE FACTORED.
LDA  INTEGER
    THE LEADING DIMENSION OF THE MATRIX A.
N  INTEGER
    THE ORDER OF THE MATRIX A.
JOB  INTEGER
    PARAMETER TO TELL IF PARTIAL PIVOTING
    IS TO BE DONE. IF JOB IS 0 NO PIVOTING
    IS DONE, AND IF JOB IS 1 THEN PARTIAL
    PIVOTING IS TO BE USED.

ON RETURN

A  AN UPPER TRIANGULAR MATRIX AND THE
    MULTIPLIERS WHICH WERE USED TO OBTAIN
    IT. THE DECOMPOSITION CAN BE WRITTEN
    A = L*U WHERE L IS A PRODUCT OF THE
    PERMUTATION AND THE UNIT LOWER
    TRIANGULAR MATRICES AND U IS UPPER
    TRIANGULAR.
IPVT  INTEGER(N)
    AN INTEGER VECTOR OF PIVOT INDICES.
INFO  INTEGER
    = 0  NORMAL VALUE.
    = K  IF U(K,K) .EQ. 0.0 . THIS IS
    NOT AN ERROR CONDITION FOR THIS

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
If \( A(L,K) = 0 \), go to 60.

Interchange if necessary.

If \( L = K \), go to 30.

\[
T = A(L,K) \\
A(L,K) = A(K,K) \\
A(K,K) = T
\]

If \( L = K \), go to 40.

\[
T = -\frac{1}{A(K,K)} \\
\text{CALL SSCAL}(N-K,T,A(K+1,K),1)
\]

Row elimination with column indexing.

\[
\text{DO 50 } J = K+1, N \\
T = A(L,J) \\
\text{IF } (L = K) \text{ GO TO 40} \\
A(L,J) = A(K,J) \\
A(K,J) = T \\
\text{CONTINUE}
\]

\[
\text{CALL SAXPY}(N-K,T,A(K+1,K),1,A(K+1,J),1)
\]

Continue.

Go to 70.

Continue.

\[
\text{INFO} = K \\
\text{CONTINUE}
\]

Continue.

Continue.

\[
\text{IPVT}(N) = N \\
\text{IF } (A(N,N) = 0) \text{ INFO} = N
\]

Return.
SUBROUTINE JOBHLP(JOB)

INTEGER JOB

THIS ROUTINE PROVIDES THE USER WITH AN INTERACTIVE SOLUTION TO OBTAINING THE JOB PARAMETER. THIS ROUTINE IS CALLED FROM SMTGEN. THE USER WILL BE PROMPTED FOR INFORMATION THAT WILL ALLOW FOR THE CONSTRUCTION OF THE JOB PARAMETER. BRIEF ERROR CHECKING IS DONE.

ON ENTRY
 JOB INTEGER
    USED FOR RETURN ONLY.

ON RETURN
 JOB INTEGER
    THIS CONTAINS THE JOB PARAMETER FOR THE REST OF THE SOFTWARE Routines TO USE IN DETERMINING WHAT OPTIONS TO PERFORM.
 JOB IS A 7 DIGIT NUMBER TUVWXYZ,

WHERE
 T IS FOR LU DECOMPOSITION WITH NO PARTIAL PIVOTING.
 U IS FOR LU DECOMPOSITION WITH PARTIAL PIVOTING.
 V IS FOR QR DECOMPOSITION WITH NO PARTIAL PIVOTING.
 W IS FOR QR DECOMPOSITION WITH PARTIAL PIVOTING.
 X IS FOR THE LEAST SQUARES COMPUTATIONS USING THE QR DECOMPOSITION.
 Y IS FOR LU DECOMPOSITION WITH PARTIAL PIVOTING AND WEIGHTED INITIAL CONDITIONS.
 Z TELLS IF THE ORIGINAL MATRIX A AND VECTOR B ARE TO BE PRINTED AT THE TOP OF THE OUTPUT FILE.

WHERE
 T, U, V, W, X AND Y CAN ALL HAVE THE FOLLOWING VALUES:
0  SKIP THIS OPTION COMPLETELY.
1  MEANS TO DO THE PARTICULAR
    DECOMPOSITION AND PRINT THE RESULTS
    TO THE OUTPUT FILE. THE SYSTEM IS NOT
    SOLVED.
2  MEANS TO DO THE DECOMPOSITION, BUT
    NOT TO PRINT THE RESULTS OF THE
    DECOMPOSITION. THE SYSTEM IS SOLVED FOR
    X AND THIS RESULT IS PRINTED TO THE
    OUTPUT FILE.
3  MEANS TO DO THE DECOMPOSITION AND PRINT
    THE RESULTS TO THE OUTPUT FILE. THE
    SYSTEM IS SOLVED FOR X AND THIS RESULT
    IS PRINTED TO THE OUTPUT FILE.

Z  HAS THE FOLLOWING VALUES:
0  THE MATRIX A AND VECTOR B WILL NOT BE
    PRINTED TO THE OUTPUT FILE.
1  THE MATRIX A AND VECTOR B WILL BE
    PRINTED AT THE TOP OF THE OUTPUT FILE.

JOBHELP VERSION DATED 12/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

INTERNAL VARIABLES

INTEGER INPUT
CHARACTER*1 ANS

WHERE

INPUT INTEGER
ACCEPTS VALUES FROM THE USER.

ANS CHARACTER*1
ACCEPTS INTERACTIVE RESPONSES.

JOB = 0

10  WRITE(5,1001)
1001  FORMAT(/,1X,'JOB HELP',/1X,'ANSWER THE',/...
FOLLOWING QUESTIONS IN ORDER TO PRODUCE THE JOB PARAMETER

WRITE(5,1002)
1002 FORMAT(/,1X,'FOR THE FIRST SIX OPTIONS,',
+ '/,1X,'THE FOLLOWING VALUES WILL BE USED FOR INPUT',
+ '/,1X,'0 SKIP THIS PARTICULAR OPTION',
+ '/,1X,'1 PERFORM THE SPECIFIED DECOMPOSITION',
+ '/,1X,'2 PERFORM THE SPECIFIED DECOMPOSITION,",
+ '/,1X,'3 PERFORM THE SPECIFIED DECOMPOSITION,","
+ '/,1X,'4 PERFORM THE SPECIFIED DECOMPOSITION,",
+ '/,1X,'5 PERFORM THE SPECIFIED DECOMPOSITION,",
+ '/,1X,'6 PERFORM THE SPECIFIED DECOMPOSITION,"')

WRITE(5,1003)
1003 FORMAT(/,1X,'ENTER A VALUE FOR THE LU',
+ '/,1X,'DECOMPOSITION WITH NO PARTIAL PIVOTING',
+ '/,1X,'(0,1,2,3)',")

READ(5,1004)INPUT
1004 FORMAT(I1)

IF(INPUT.LT.0.OR.INPUT.GT.3)GOTO 20
JOB = JOB * 10 + INPUT

WRITE(5,1005)
1005 FORMAT(/,1X,'ENTER A VALUE FOR THE LU',
+ '/,1X,'DECOMPOSITION WITH PARTIAL PIVOTING',
+ '/,1X,'(0,1,2,3)',")

READ(5,1004)INPUT
IF(INPUT.LT.0.OR.INPUT.GT.3)GOTO 30
JOB = JOB * 10 + INPUT

WRITE(5,1006)
1006 FORMAT(/,1X,'ENTER A VALUE FOR THE QR',
+ '/,1X,'DECOMPOSITION WITH NO PARTIAL PIVOTING',
+ '/,1X,'(0,1,2,3)',")

READ(5,1004)INPUT
IF(INPUT.LT.0.OR.INPUT.GT.3)GOTO 40
JOB = JOB * 10 + INPUT

WRITE(5,1007)
1007 FORMAT(/,1X,'ENTER A VALUE FOR THE QR',
+ '/,1X,'DECOMPOSITION WITH PARTIAL PIVOTING',
+ '/,1X,'(0,1,2,3)',")
READ(5,1004) INPUT
C
IF(INPUT.LT.0.OR.INPUT.GT.3)GOTO 50
JOB = JOB * 10 + INPUT
C
60 WRITE(5,1008)
1008 FORMAT(/,1X,'ENTER A VALUE FOR THE QR',
 + ' DECOMPOSITION WITH PARTIAL PIVOTING AND',
 + '/,1X,'LEAST SQUARES APPROXIMATIONS',
 + '(0,1,2,3)','$)
C
READ(5,1004) INPUT
C
IF(INPUT.LT.0.OR.INPUT.GT.3)GOTO 60
JOB = JOB * 10 + INPUT
C
70 WRITE(5,1009)
1009 FORMAT(/,1X,'ENTER A VALUE FOR THE LU',
 + ' DECOMPOSITION WITH PARTIAL PIVOTING AND',
 + '/,1X,'WEIGHTED INITIAL CONDITIONS',
 + '(0,1,2,3)','$)
C
READ(5,1004) INPUT
C
IF(INPUT.LT.0.OR.INPUT.GT.3)GOTO 70
JOB = JOB * 10 + INPUT
C
WRITE(5,1010)
1010 FORMAT(/,1X,'WOULD YOU LIKE THE ORIGINAL',
 + ' MATRIX A AND VECTOR B','/1X,'TO APPEAR',
 + ' AT THE BEGINNING OF THE OUTPUT FILE (Y/N)','$)
C
READ(5,1011) ANS
1011 FORMAT(A1)
C
JOB = JOB * 10
IF(ANS.EQ.'Y') JOB = JOB + 1
C
DETERMINE IF THE JOB PARAMETER IS CORRECT.
C
WRITE(5,1012) JOB
1012 FORMAT(/,1X,'YOU HAVE PRODUCED A VALUE OF ',I7,
 + ' FOR THE JOB PARAMETER. ,/1X,'IS THIS VALUE',
 + ' CORRECT (Y/N)? ','$)
C
READ(5,1011) ANS
IF(ANS.NE.'Y'.AND.ANS.NE.'y')GOTO 10
C
RETURN
END

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
SUBROUTINE SAROUT(A,LDA,N,P,MESSAG,TITLE)

C INTEGER LDA,N,P
REAL A(LDA,1)
CHARACTER *80 MESSAG,TITLE

C THIS ROUTINE IS TO PRINT THE MATRICES AND
VECTORS IN A FORMATTED MANNER. THE ROUTINE WILL
BE PROVIDED WITH A MATRIX OR VECTOR THE ROUTINE
WILL PRINT OUT THE MATRIX IN THE FOLLOWING FORM

IF THE INPUT IS A VECTOR, THE VECTOR WILL BE
PRINTED VERTICALLY WITH A HEADING AT THE TOP
PROVIDED BY THE USER IN MESSAG.

IF THE INPUT IS A TWO DIMENSIONAL MATRIX, IT
WILL BE BROKEN UP INTO GROUPS OF EIGHT COLUMNS
EACH. EACH ONE OF THESE GROUPS WILL BE PRINTED
OUT VERTICALLY WITH THE SUPPLIED HEADING.
EACH SET OF EIGHT COLUMNS WILL BE GIVEN A PAGE
NUMBER. PAGE 1 BEING THE FIRST EIGHT COLUMNS.

ON ENTRY

A REAL(N,P)
THE MATRIX TO BE PRINTED OUT.

LDA INTEGER
THE LEADING DIMENSION OF A.

N INTEGER
NUMBER OF ROWS IN A.

P INTEGER
NUMBER OF COLUMNS IN A.

MESSAG CHARACTER *80
THE HEADING MESSAGE TO APPEAR AT THE
TOP OF EACH PAGE.

TITLE CHARACTER*80
THE TITLE IS THE NAME OF THE PROBLEM,
AND WILL APPEAR ON ALL OUTPUT.

SAROUT VERSION 7/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
INTERNAL VARIABLES

INTEGER I,J,L,K,PAGE

I,J,K,L INTEGER
LOOP COUNTERS AND ARRAY INDICES.

PAGE INTEGER
THE PAGE COUNTER.

INITIALIZE THE PAGE COUNTER.

PAGE = 0

PRINT THE WHOLE ARRAY IN 8 COLUMNS SEGMENTS.

DO 20 I = 1,P,8

INCREMENT PAGE AND SET THE BOUND FOR THESE EIGHT COLUMNS.

PAGE = PAGE + 1
K = I + 7

MAKE SURE NOT TO GO OUTSIDE THE ARRAY BOUNDS.
THIS IS FOR A NON DIVISIBLE BY EIGHT NUMBER OF COLUMNS. WRITE THE PAGE NUMBER AND HEADING.

IF(I+7.GT.P)K = P
WRITE(22,1001)PAGE,TITLE,MESSAG
FORMAT('1',' PAGE ',12,3X,A80,/,12X,A80/)

PRINT THE N ROWS OF THE MATRIX WITHIN THE COLUMN BOUNDS OF I AND K.

DO 10 J = 1,N
WRITE(22,1002)(A(J,L),L = I,K)
FORMAT(1X,10E16.8)

CONTINUE
CONTINUE

RETURN
END
SUBROUTINE SCPOUT(COMPAR,N,CONDS,TITLE)

C

INTEGER N
REAL COMPAR(51,7),CONDS(52)
CHARACTER *80 TITLE

C
THIS ROUTINE IS PROVIDED WITH THE COMPAR
ARRAY, WHICH IS THE RESULTS OF ALL THE
DIFFERENT SOLUTION METHODS. THIS ROUTINE WILL
PRINT TO THE OUTPUT FILE SEVERAL DIFFERENT
TABLES OF INFORMATION. THE COMPAR ARRAY, AS
PROVIDED, HAS SIX DESIGNATED COLUMNS. THE ORDER
OF THE COLUMNS IS AS FOLLOWS:
COLUMN 1 THE LU DECOMPOSITION WITHOUT PIVOTING,
COLUMN 2 THE LU DECOMPOSITION WITH PIVOTING,
COLUMN 3 THE QR DECOMPOSITION WITHOUT PIVOTING,
COLUMN 4 THE QR DECOMPOSITION WITH PIVOTING,
COLUMN 5 THE QR DECOMPOSITION AND LEAST SQUARE,
COLUMN 6 THE LU DECOMPOSITION AND WEIGHTED
INITIAL CONDITIONS.
THE COMPAR ROUTINE WILL BE CONDENSED TO MAKE
THE OUTPUT OF RESULTS EASIER. ONCE THE COMPAR
ARRAY IS CONDENSED, THE FOLLOWING INFORMATION
WITH HEADINGS IS PRINTED TO THE OUTPUT FILE:
THE CONDITION NUMBERS FOR THE REQUESTED
SOLUTION METHODS, THE ACTUAL COMPUTED RESULTS
FOR THE REQUESTED SOLUTION METHODS, THE
ABSOLUTE ERRORS FOR THE REQUESTED SOLUTION
METHODS, AND THE RELATIVE ERRORS FOR THE
REQUESTED SOLUTION METHODS.
FOR THE ERROR ESTIMATES, THE ROUTINE SRIGHT IS
CALLED TO PROVIDE ACCURATE VALUES. THIS ROUTINE
MUST BE ALTERED FOR EACH DIFFERENT SOLUTION
COMPUTED.

ON ENTRY

COMPAR REAL(51,7)
THIS ARRAY CONTAINS ALL OF THE COMPUTED
RESULTS FROM THE REQUESTED SOLUTION
METHODS. ROW 51 CONTAINS THE ESTIMATES
FOR THE CONDITION NUMBER FOR THE
INITIAL MATRIX FOR THAT PARTICULAR
SOLUTION METHOD.

CONDS REAL(52)
THIS ARRAY CONTAINS THE VALUES FOR THE
INITIAL AND TERMINAL CONDITIONS FOR THE
RECURSION RELATION. ITEM 51 TELLS THE
NUMBER OF INITIAL CONDITIONS AND ITEM
52 TELLS THE NUMBER OF TERMINAL CONDS.
IF THE ORIGINAL MATRIX USED WAS OF THE
AUGMENTED TYPE, THEN ITEM 51 WILL BE
-1.

N INTEGER
THE NUMBER OF VALUES COMPUTED.

TITLE CHARACTER *80
THE TITLE OF THE PROBLEM TO APPEAR ON
EACH PAGE OF OUTPUT.

ON RETURN

NO VALUES ARE RETURNED, BUT MANY PIECES OF
INFORMATION ARE PRINTED TO THE OUTPUT FILE.

SCPOUT VERSION DATED 12/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

SUBROUTINES SRIGHT

INTERNAL VARIABLES

INTEGER I,J,K,L,T(6)
CHARACTER *18 DECOMP,TYPE,HEADS(7)*36

WHERE

I,J,L INTEGER
LOOP COUNTERS AND ARRAY INDICES.

K INTEGER
THE NUMBER OF ACTIVE SOLUTION METHODS.

DECOMP, TYPE, HEAD CHARACTER
USED TO CONSTRUCT THE HEADINGS.

T INTEGER(6)
A TEMP ARRAY TO HELP IN THE
CONSTRUCTION OF THE HEADINGS OF THE
REQUESTED SOLUTION METHODS.

INITIALIZE THE ARRAY FOR THE HEADINGS.

DO 10 I = 1,7
   HEADS(I) =
CONDENSE THE COMPAR ARRAY BY SHIFTING COLUMNS TO THE LEFT TO FILL VACANT COLUMNS. THE RESULT IS THE COMPAR ARRAY FILLED FROM THE LEFT SIDE. K IS THE NUMBER OF ACTIVE SOLUTION METHODS.

\[ K = 0 \]

DO 30 I = 1,6
   IF(COMPAR(51,I).EQ.0.0)GOTO 30
   K = K + 1

ALSO KEEP TRACK OF THE ORIGINAL COLUMN SO THE HEADINGS CAN BE PREPARED PROPERLY.

\[ T(K) = I \]

DO 20 J = 1,N
   COMPAR(J,K) = COMPAR(J,I)
   COMPAR(51,K) = COMPAR(51,I)
20 CONTINUE

30 CONTINUE

GENERATE THE HEADINGS FOR THE ACTIVE SOLUTION METHODS.

DO 40 I = 1,K
   IF(T(I).LT.3.OR.T(I).EQ.6)DECOMP = ' LU DECOMPOSITION ' + 'QR DECOMPOSITION ' + 'WITHOUT PIVOTING ' + 'WITH PIVOTING ' + 'AND LEAST SQUARES ' + 'WEIGHTED INITIALS ' + 'ESTIMATE OF EXACT VALUE ' HEADS(I) = DECOMP // TYPE
40 CONTINUE

HEADS(K+1) = 'ESTIMATE OF EXACT VALUE'

WRITE THE CONDITION NUMBERS TO THE OUTPUT FILE.

WRITE(22,1001)TITLE
1001 FORMAT('1',11X,A80,/' THE FOLLOWING TABLE IS', ' FOR THE CONDITION NUMBERS '/)
WRITE(22,1002)COMPAR(51,I),HEADS(I)
1002 FORMAT(/1X,E16.8,' IS THE RECIPROCAL ',
     ' OF THE CONDITION NUMBER',/IX,
     'FOR THE',A36)

CONTINUE

CHECK IF THE REDUCED FORM OF THE ASSOCIATED
LINEAR SYSTEM WAS USED. IF NOT SKIP THE NEXT
SECTION. THIS SECTION OFFSETS THE RESULTS TO
INCORPORATE THE INITIAL AND TERMINAL
CONDITIONS.

IF(CONDS(51).LT.0)GOTO 120

N = N + CONDS(51) + CONDS(52)

ADD IN THE TERMINAL CONDITIONS.

DO 70 I = N,N-CONDS(52)+1,-1
   DO 60 J = 1,K
      COMPAR(I,J) = CONDS(I)
   60 CONTINUE
70 CONTINUE

MOVE THE VALUES IN THE ARRAY DOWN TO MAKE ROOM
FOR THE INITIAL VALUES.

DO 90 I = N-CONDS(51)-CONDS(52),1,-1
   J = I + CONDS(51)
   DO 80 L = 1,K
      COMPAR(J,L) = COMPAR(I,L)
  80 CONTINUE
90 CONTINUE

ADD IN THE INITIAL CONDITIONS.

DO 110 I = 1,CONDS(51)
   DO 100 J = 1,K
      COMPAR(I,J) = CONDS(I)
  100 CONTINUE
110 CONTINUE

CALL SRIGHT TO PROVIDE A COLUMN OF ACCURATE
VALUES FOR ERROR COMPARISONS.

SET K FOR THE NEXT AVAILABLE COLUMN.

K = K + 1
CALL SRIGHT(COMPAR,N,K)

WRITE THE ACTUAL VALUES TO THE OUTPUT FILE
WITH THE COLUMN HEADINGS.

WRITE(22,1003)TITLE
1003 FORMAT('1',11X,A80,/',' THE FOLLOWING TABLE IS', + ' FOR THE ACTUAL COMPUTED VALUES',/,'1X,'WITH ', + 'THE FOLLOWING COLUMN LABELS',/)

DO 130 I = 1,K
WRITE(22,1004)I,HEADS(I)
1004 FORMAT(IX,'COLUMN',I2,' REPRESENTS THE',A36)
130 CONTINUE

WRITE(22,1005)
1005 FORMAT(//)

DO 140 I = 1,N
WRITE(22,1006)I-1,(COMPAR(I,J),J=1,K)
1006 FORMAT(IX,I4,7E16.8)
140 CONTINUE

C COMPUTE THE ABSOLUTE ERRORS.

DO 160 I = 1,N
   DO 150 J = 1,K-1
      COMPAR(I,J) = ABS(COMPAR(I,J)-COMPAR(I,K))
   150 CONTINUE
160 CONTINUE

WRITE THE ABSOLUTE ERRORS TO THE OUTPUT FILE WITH COLUMN HEADINGS.

WRITE(22,1007)TITLE
1007 FORMAT('1',11X,A80,/',' THE FOLLOWING TABLE IS', + ' FOR THE ABSOLUTE ERRORS',/,'1X,'WITH THE ', + 'FOLLOWING COLUMN LABELS',/)

DO 170 I = 1,K
   WRITE(22,1004)J,HEADS(I)
170 CONTINUE

WRITE(22,1005)

DO 180 I = 1,N
   WRITE(22,1006)I-1,(COMPAR(I,J),J=1,K)
180 CONTINUE

C COMPUTE THE RELATIVE ERRORS.

DO 200 I = 1,N
   DO 190 J = 1,K-1

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
COMPAR(I,J) = ABS(COMPAR(I,J)/COMPAR(I,K))

CONTINUE

WRITE THE RELATIVE ERRORS TO THE OUTPUT FILE WITH COLUMN HEADINGS.

WRITE(22,1008)TITLE
1008 FORMAT('1',11X,A80,/, 'THE FOLLOWING TABLE IS', 'FOR THE RELATIVE ERRORS',/,'WITH THE ', 'FOLLOWING COLUMN LABELS',/)

DO 210 I = 1,7
   WRITE(22,1004)J,HEADS(I)
210 CONTINUE

WRITE(22,1005)

DO 220 I = 1,N
   WRITE(22,1006)I-1,(COMPAR(I,J),J=1,K)
220 CONTINUE

RETURN
END
SUBROUTINE SGTVAL(C,N,TAU)

INTEGER N
REAL C(50), TAU


ON ENTRY
N INTEGER
THE CURRENT ROW OF THE BEING PROCESSED.

TAU REAL
A VALUE TO USE FOR COMPUTING THE COEFFICIENTS.

ON RETURN
C REAL(50)
CONTAINS THE NTH SET OF COEFFICIENTS FOR THE MATRIX A.

SGTVAL VERSION DATED 12/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY

INTERNAL VARIABLES

INTEGER I

WHERE
I INTEGER
USED FOR A LOOP COUNTER.

THE CODE TO FILL THE ARRAY C GOES HERE.

RETURN
END
SUBROUTINE SLSOUT(X,LDX,N,P,JOB,Y,RCOND,TITLE)
C
INTEGER N,P,LDX,JOB
REAL X(LDX,1),Y(1),RCOND
CHARACTER *80 MESSAG,TITLE
C
C THIS ROUTINE IS PROVIDED WITH A MATRIX X AND IT
C WILL PRINT OUT TO A FILE FOR22 SEVERAL
C PIECES OF INFORMATION THAT ARE CALCULATED.
C THE ROUTINE USES THE QR DECOMPOSITION TO SOLVE
C THE LEAST SQUARES APPROX. ON THE MATRIX X.
C THE PARAMETER JOB WILL TELL SEVERAL OPTIONS
C THAT THE ROUTINE CAN HANDLE. THIS ROUTINE ALSO
C PROVIDES FOR THE WEIGHTING OF THE INITIAL LEAST
C SQUARES MATRIX TO PROVIDE BETTER RESULTS.
C
ON ENTRY
C
X REAL(N,P)
THE MATRIX THAT WILL BE DECOMPOSED.
C
LDX INTEGER
THE LEADING DIMENSION OF X.
C
N INTEGER
THE NUMBER OF ROWS IN X.
C
P INTEGER
THE NUMBER OF COLUMNS OF X.
C
JOB INTEGER
THIS PARAMETER WILL TELL THE ROUTINE
WHAT TO CALCULATE AND PRINT OUT TO THE
OUTPUT FILE FOR22.
JOB IS A 3 DIGIT NUMBER OF THE FORM XYZ
WHERE
C
IF X IS 1, THEN THE ROUTINE WILL WRITE
THE DECOMPOSITION TO THE OUTPUT FILE AS
A Q AND R MATRIX BOTH OF SIZE N X N.
C
IF X IS 0, NO OUTPUT OF Q AND R WILL
APPEAR.
C
IF Y IS 1, THE ROUTINE WILL WRITE THE
SOLUTION B OF THE EQUATION XB = Y TO
THE OUTPUT FILE AS A ROW VECTOR.
C
IF Y IS 0, NO SOLUTION WILL BE
C
Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
CALCULATED OR PRINTED TO THE OUTPUT FILES.

Z IS NOT USED AND WILL BE SET TO 0.

Y REAL(N)
Y IS THE INITIAL SOLUTION VECTOR FOR THE EQUATION XB = Y, WHERE WE WILL SOLVE FOR B.

TITLE CHARACTER*80
TITLE IS THE NAME OF THE PROBLEM AND APPEARS ON ALL OUTPUT GENERATED.

ON RETURN

Y REAL(N)
Y WILL CONTAIN THE SOLUTION FOR XB = Y WHERE THE INITIAL VECTOR Y WAS OVERWRITTEN BY THE SOLUTION VECTOR B.

RCOND REAL
THE ESTIMATED CONDITION NUMBER FOR THE MATRIX.

SLSOUT VERSION DATED 12/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

SUBROUTINES SQRLST, SAROUT

INTERNAL VARIABLES

INTEGER JPVT(50),INFO,IJOB,IPVT(50),I,J,K,LI,
+ LDQR
REAL WORK(50),QRDXU(50),QY(50),QTY(50),TEMP,
+ RSD(50),XB(50),Q(50,50),R(50,50),B(50),
+ E(50),C(50)

WHERE

JPVT INTEGER(50)
THIS IS THE PIVOT VECTOR USED FOR THE QR MATRIX THAT IS SENT BACK FROM LINPACK. IF NO PIVOTING WAS CHOSEN, THIS WILL CONTAIN THE NUMBERS FROM 1 TO N.

IPVT INTEGER
A TEMP TO STORE JPVT WHEN RECOVERING

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
THE MATRIX DURING PIVOTING.

**IJOB** INTEGER
A TEMP VALUE FOR JOB.

**INFO** INTEGER
USED BY LINPACK FOR ERROR CHECKING.

**I,J,K** INTEGER
LOOP COUNTERS AND ARRAY INDICES.

**L** INTEGER
USED AS A INTEGER TEMP.

**LDQR** INTEGER
LEADING DIMENSION OF Q AND R.

**QRAUX** REAL(50)
USED TO STORE THE EXTRA TERM NEEDED FOR FINDING THE L MATRIX FROM LINPACK.

**WORK** REAL(50)
USED FOR WORK SPACE BY THE LINPACK Routines.

**Q** REAL(50,50)
THE Q MATRIX IN THE N X N FORM.

**R** REAL(50,50)
THE R MATRIX IN THE N X N FORM.

**E** REAL(50)
USED FOR SETTING UP ELEMENTARY COLUMN VECTORS TO RETRIEVE THE Q MATRIX.

**QY** REAL(50)
THE SOLUTION OF THE PRODUCT Q * Y.

**QTY** REAL(50)
THE SOLUTION OF THE PRODUCT Q TRANSPOSE * Y.

**RSD** REAL(50)
THE RESIDUALS FROM THE SOLUTION OF AN EQUATION BY LINPACK.

**XB** REAL(50)
THE SOLUTION OF THE PRODUCT X * B.

**B** REAL(50)
THE RESULT OF THE EQUATION \( X \times B = Y \),
WHERE \( Y \) IS AN INITIAL VECTOR.

TEMP REAL
USED AS A REAL TEMP.

CHECK TO SEE IF WEIGHTING OF THE LEAST SQUARE
MATRIX IS NEEDED, BY CALLING SLSSCL. IF NOT
REQUESTED, SLSSCL SHOULD RETURN A VECTOR OF
ALL ONES TO USE AS WEIGHTS. PERFORM THE
WEIGHTING OF THE MATRIX \( X \) AND VECTOR \( B \).

CALL SLSSCL(C,N,TAU)

DO 20 I = 1,N
  DO 10 J = 1,P
    X(I,J) = X(I,J) \times C(I)
  CONTINUE
  Y(I) = Y(I) \times C(I)
20 CONTINUE

CALL THE LEAST SQUARES ROUTINE TO REDUCE
AND SOLVE THE SYSTEM FOR THE SOLUTION \( Y \) AND
AND SET IJOB TO 1 TO INDICATE THAT PARTIAL
PIVOTING WILL BE USED.

30 LDQR = 50
IJOB = 1

CALL SQRLST(X,LDX,N,P,Y,TOL,B,RSD,
+ K,JPVT,QRAUX,WORK)

RCOND = X(1,1)/X(K,K)

CHECK TO SEE IF THE DECOMPOSITION IS TO
BE PRINTED.

IF(JOB.LT.100)GOTO 120

RETRIEVE THE R MATRIX.

DO 50 I = 1,N
  DO 40 J = I,P
    R(I,J) = X(I,J)
  CONTINUE
50 CONTINUE
retrieve the q matrix by utilizing the solve routines option to return qy. where y will be the elementary column vectors one at a time to obtain the matrix q 1 column at a time.

do 70 i = 1,p
e(i)=1.0
call sqrs1(x,ldx,n,p,qraux,e,qy,qty,b,
srd,xb,01000,info)

    do 60 j = 1,n
        q(i,j)=qty(j)
        continue
    e(i)=0.0
70 continue

undo the pivoting.

copy jpvt into ipvt.

do 80 i = 1,p
    ipvt(i)=jpvt(i)
80 continue

use the pivot vector to interchange the columns of the matrix to obtain the correct solution.

do 110 i = 1,p
    do 100 j = 1,p
        if(ipvt(j).ne.i)goto 100
        do 90 k = 1,n
            temp=r(k,i)
            r(k,i)=r(k,j)
            r(k,j)=temp
        continue
70 continue

also interchange the pivot vector.

    l = ipvt(i)
    ipvt(i) = ipvt(j)
    ipvt(j) = l
100 continue
110 continue

write the q and r matrix to the output file along with a suitable title.

messag = 'the q matrix with partial pivoting'
call sarout(q,ldqr,n,p,messag,title)

reproduced with permission of the copyright owner. further reproduction prohibited without permission.
MESSAG = 'THE R MATRIX WITH PARTIAL PIVOTING'
CALL SAROUT(R,LDQR,N,P,MESSAG,TITLE)

C CHECK TO SEE IF THE RESULT VECTOR IS TO BE WRITTEN TO THE OUTPUT FILE.
120 IF(JOB.LT.10.OR.JOB.EQ.100.OR.JOB.EQ.101) GOTO 130

C MESSAG = 'THE SOLUTION USING LEAST SQUARES'
CALL SAROUT(B,LDB,P,1,MESSAG,TITLE)

C PLACE THE SOLUTION IN Y FOR RETURNING TO THE CALLING PROGRAM.
130 DO 140 I = 1,P
   Y(I) = B(I)
140 CONTINUE

C RETURN
END
SUBROUTINE SLSSCL(C,N,TAU)

INTEGER N
REAL C(50),TAU

THIS ROUTINE PROVIDES THE WEIGHTING FACTORS FOR
THE LEAST SQUARE APPROXIMATIONS. THE VECTOR C
IS RETURNED TO THE CALLING PROGRAM WITH THE
WEIGHTING FACTORS. THE ROUTINE FILLS IN THE
FIRST N ELEMENTS OF C. IF NO WEIGHTING IS
DESIRED THEN SET ALL OF THE VALUES IN THE
VECTOR C EQUAL TO 1.
THIS ROUTINE MUST BE CHANGED FOR EACH DIFFERENT
LEAST SQUARE PROBLEM.

ON ENTRY
N INTEGER
THE NUMBER OF WEIGHTING FACTORS.

TAU REAL
A VALUE PASSED FROM THE CALLING PROGRAM
TO AID IN THE GENERATING OF THE
WEIGHTS. THIS VALUE IS OPTIONAL TO USE.

ON RETURN
C REAL(50)
THIS IS THE VECTOR OF THE WEIGHTS TO
BE USED TO SCALE THE MATRIX FOR
THE LEAST SQUARES ROUTINE.

SLSSCL VERSION DATED 12/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

INTERNAL VARIABLES

INTEGER I

WHERE
I INTEGER
USED FOR THE LOOP COUNTER.

AS A DEFAULT ROUTINE SET ALL OF THE VALUES IN C
EQUAL TO 1.0.

DO 10 I = 1,N
   C(I) = 1
10 CONTINUE

RETURN
END
SUBROUTINE SLUOUT(A,LDA,N,JOB,B,RCOND,TITLE)

INTEGER N,LDA,JOB
REAL A(LDA,1),B(1),RCOND
CHARACTER *80 MESSAG,TITLE

THIS ROUTINE IS PROVIDED A MATRIX A AND WILL PRINT OUT TO A FILE POSSIBLY SEVERAL PIECES OF INFORMATION THAT ARE CALCULATED. THE ROUTINE USES THE LU DECOMPOSITION TO ACHIEVE A L AND U MATRIX SUCH THAT A = LU. THE PARAMETER JOB WILL TELL SEVERAL OPTIONS THAT THE ROUTINE CAN PERFORM. THE OPTION FOR THE SOLUTION OF THE LINEAR SYSTEM HAS TWO OPTIONS. THE FIRST IS TO SOLVE THE ORIGINAL SYSTEM AX = B FOR THE SOLUTION VECTOR X. THE SECOND IS TO ALLOW THE OPTION OF WEIGHTING OF THE MATRIX A AND VECTOR B BEFORE THE SOLUTION IS ATTEMPTED. THE OPTION FOR NOT PIVOTING IS ALSO AVAILABLE, BUT IS DISABLED FOR THE SECOND SOLUTION METHOD.

ON ENTRY

A REAL(N,N) THE MATRIX THAT WILL BE DECOMPOSED.

LDA INTEGER THE LEADING DIMENSION OF A.

N INTEGER THE ORDER OF A.

JOB INTEGER THIS PARAMETER WILL TELL THE ROUTINE WHAT TO CALCULATE AND PRINT OUT TO THE OUTPUT FILE. JOB IS A 3 DIGIT NUMBER OF THE FORM XYZ WHERE

IF JOB < 0, THIS INDICATES THAT THE MATRIX A AND THE VECTOR B ARE TO BE WEIGHTED. A SPECIAL ROUTINE SLUSCL IS CALLED TO PROVIDE THE WEIGHTS. ONCE THIS IS DONE THE ABSOLUTE VALUE OF JOB IS OBTAINED AND IS DEFINED AS BELOW.

IF X IS 1, THEN THE ROUTINE WILL WRITE THE DECOMPOSITION TO THE OUTPUT FILE AS A L AND U MATRIX BOTH OF SIZE N X N.
IF X IS 0, NO OUTPUT OF L AND U WILL APPEAR IN THE OUTPUT FILE.

IF Y IS 1, THE ROUTINE WILL WRITE THE SOLUTION OF THE EQUATION AX = B TO THE OUTPUT FILE AS A COLUMN VECTOR.

IF Y IS 0, NO SOLUTION WILL BE CALCULATED OR PRINTED.

IF Z IS 1, THEN THE ROUTINE WILL DO ITS CALCULATIONS USING PARTIAL PIVOTING ON THE ROWS OF THE MATRIX A.

IF Z IS 0, PIVOTING WILL NOT BE USED.

B

REAL(N)

THE INITIAL SOLUTION VECTOR FOR THE EQUATION AX = B.

ON RETURN

B

REAL(N)

THE SOLUTION TO THE SYSTEM AX = B WHERE THE INITIAL VECTOR B IS OVERWRITTEN BY THE SOLUTION X.

RCOND

REAL

AN ESTIMATE FOR THE CONDITION OF THE MATRIX A.

TITLE

CHARACTER*80

THE TITLE OF THE PROBLEM AND WILL ON ALL OUTPUT GENERATED.

SLUOUT

VERSION DATED 7/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

SUBROUTINES CALLED ASGECO, SGESL, SAROUT

INTERNAL VARIABLES

INTEGER IPVT(50), IJOB, I, J, K, LDLU
REAL Z(50), U(50, 50), L(50, 50), T, C(50), TAU

WHERE

IPVT INTEGER(50)
THE PIVOT VECTOR USED TO DECODE THE LU MATRIX THAT IS SENT BACK FROM LINPACK.

IJOB INTEGER
A TEMP VALUE FOR JOB.

Z REAL(50)
USED FOR WORK SPACE BY LINPACK.

U REAL(50,50)
THE U MATRIX IN THE N X N FORM.

L REAL(50,50)
THE L MATRIX IN THE N X N FORM.

T REAL
A TEMP FOR ACCUMULATION.

I,J,K INTEGER
LOOP COUNTERS AND ARRAY INDICES.

LDLU INTEGER
THE LEADING DIMENSION OF L AND U.

C REAL(50)
VECTOR FOR WEIGHTS.

TAU REAL
A SPECIAL VALUE TO USE FOR WEIGHTS

INITIALIZATIONS.

LDLU = 50
IJOB = 0

CHECK TO SEE IF WEIGHTING IS NEEDED. IF WEIGHTING IS NEEDED CALL THE WEIGHT ROUTINE AND APPLY THE WEIGHTS TO THE MATRIX A AND VECTOR B. THEN NEGATE THE JOB PARAMETER.

IF(JOB.GT.0)GOTO 30

CALL SLUSCL(C,N,TAU)

JOB = -1 * JOB

DO 20 I = 1,N
  DO 10 J = 1,N
    A(I,J) = A(I,J) * C(I)
  CONTINUE

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
B(I) = B(I) * C(I)
CONTINUE

SET THE IJOB PARAMETER FOR PIVOTING.

IF(JOB/2*2.NE.JOB)IJOB=1

DECOMPOSE THE MATRIX A.

CALL ASGECO(A,LDA,N,IPVT,RCOND,Z,IJOB)

CHECK IF THE ROUTINE IS TO WRITE THE
L AND U MATRIX.

IF(JOB.LT.100)GOTO 130

RETRIEVE THE L MATRIX FROM THE DECOMPOSITION.

DO 50 I = 2,N
   DO 40 J = I-1,1,-1
      L(I,J) = -1 * A(I,J)
   CONTINUE
50 CONTINUE

RETRIEVE THE U MATRIX FROM THE DECOMPOSITION.

DO 70 I = 1,N
   DO 60 J = I,N
      U(I,J) = A(I,J)
   CONTINUE
70 CONTINUE

PLACE ONES ON THE DIAGONAL OF THE L MATRIX.

DO 80 I = 1,N
   L(I,I) = 1
80 CONTINUE

UNDO THE PIVOTING OF THE L MATRIX IF NEEDED.

DO 100 I = N-1,1,-1
   K = IPVT(I)
   IF(K.EQ.I)GOTO 100
   DO 90 J = I,N
      T = L(I,J)
      L(I,J) = L(K,J)
      L(K,J) = T
   CONTINUE
90 CONTINUE
100 CONTINUE
WRITE THE L AND U MATRIX TO THE OUTPUT FILE
ALONG WITH THE TITLE.

IF(IJOB.NE.0)GOTO 110

MESSAG = 'THE L MATRIX WITH NO PARTIAL PIVOTING'
CALL SAROUT(L,LDLU,N,N,MESSAG,TITLE)

MESSAG = 'THE U MATRIX WITH NO PARTIAL PIVOTING'
CALL SAROUT(U,LDLU,N,N,MESSAG,TITLE)

GOTO 120

110 MESSAG = 'THE L MATRIX WITH PARTIAL PIVOTING'
CALL SAROUT(L,LDLU,N,N,MESSAG,TITLE)

MESSAG = 'THE U MATRIX WITH PARTIAL PIVOTING'
CALL SAROUT(U,LDLU,N,N,MESSAG,TITLE)

120 CONTINUE

CHECK TO SEE IF THE SOLUTION IS TO BE COMPUTED.
FOR THE NOW DECOMPOSED SYSTEM THE 0 PARAMETER
IN THE CALL FORCES THE SOLUTION OF EQUATION.
AX = B FOR THE INITIAL VECTOR B.

130 IF (JOB.LT.10.OR.JOB.EQ.100.OR.JOB.EQ.101)
+ GOTO 160

CALL SGESL(A,LDA,N,IPVT,B,0)

CHECK IF PIVOTING WAS USED IN THE DECOMPOSITION
OF A, AND WRITE THE PROPER HEADING ALONG WITH
THE SOLUTION VECTOR X TO THE OUTPUT FILE.

IF(IJOB.NE.0)GOTO 140

MESSAG = 'THE SOLUTION FOR Ay=b USING LU' //
+ ' WITH NO PIVOTING'
CALL SAROUT(B,LDA,N,1,MESSAG,TITLE)

GOTO 150

140 MESSAG = 'THE SOLUTION FOR Ay=b USING LU' //
+ ' WITH PIVOTING'
CALL SAROUT(B,LDA,N,1,MESSAG,TITLE)

150 CONTINUE

160 RETURN

END
SUBROUTINE SLUSCL(C,N,TAU)

INTEGER N
REAL C(50),TAU

THIS ROUTINE RETURNS THE SCALING FACTORS FOR
THE INITIAL CONDITIONS OF THE LU DECOMPOSITION.
THE VECTOR C IS FILLED TO REFLECT THE WEIGHTING
DESIRED FOR A PARTICULAR PROBLEM. THE FIRST N
VALUES IN C ARE RETURNED TO BE USED AS WEIGHTS.
IF WEIGHTS ARE NOT DESIRED, THEN ALL THE VALUES
IN THE ARRAY C SHOULD BE SET TO 1.
THIS ROUTINE MUST BE CHANGED FOR EACH NEW SET
OF WEIGHTS DESIRED.

ON ENTRY

N
THE NUMBER OF WEIGHTING FACTORS.

TAU
A VALUE PASSED FROM THE CALLING PROGRAM
TO AID IN THE GENERATING OF THE
WEIGHTS. THIS VALUE ID OPTIONAL TO USE.

ON RETURN

C
THIS IS THE VECTOR OF THE WEIGHTS TO
BE USED TO SCALE THE MATRIX FOR
THE LU DECOMPOSITION.

SLUSCL
VERSION DATED 12/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

INTERNAL VARIABLES

INTEGER I

WHERE

I
USED FOR THE LOOP COUNTER.

AS A DEFAULT ROUTINE SET ALL OF THE VALUES IN C
EQUAL TO 1.0.

DO 10 I = 1,N
    C(I) = 1
10 CONTINUE

RETURN
END
SUBROUTINE SMTGEN

THIS ROUTINE GENERATES THE ASSOCIATED LINEAR SYSTEM OF EQUATIONS FOR A SPECIFIED RECURRENCE RELATION. THIS ROUTINE IS CALLED FROM SRECUR, IF THE MATRIX IS TO BE GENERATED. THERE ARE TWO POSSIBLE SOURCES FOR THE INPUT INFORMATION NEEDED IN GENERATING THE MATRIX. THE ROUTINE WILL PROMPT THE USER FOR THE SOURCE OF INPUT. THE USER CAN CHOOSE TO USE AN EXISTING DATA FILE FOR20, OR INTERACTIVELY CREATE FOR20. THIS ALLOWS THE USER TO REUSE THE DATA AT A LATER TIME.

ONCE THE SOURCE OF DATA HAS BEEN DETERMINED AND THE DATA READ IN, THE ROUTINE WILL GENERATE EITHER THE AUGMENTED OR THE REDUCED FORM OF THE LINEAR SYSTEM OF EQUATIONS. THE INCLUSION OF INITIAL AND TERMINAL CONDITIONS IS ALLOWED. THE COEFFICIENTS FOR THE MATRIX ARE OBTAINED FROM SGTVAL, WHICH IS A SUBROUTINE TO PROVIDE 1 ROW OF COEFFICIENTS AT A TIME.

THE OUTPUT OF THE MATRIX, INITIAL AND TERMINAL CONDITIONS, AND THE PROVIDED TITLE WILL BE TO FOR21. THIS FILE IS READY TO BE USED BY SRECUR AND THE REST OF THE SOFTWARE.

ON ENTRY

THE ROUTINE PASSES NO PARAMETERS, BUT ACCEPTS INFORMATION INTERACTIVELY FROM THE USER. THIS CHOICE COULD INVOLVE READING INFORMATION FROM FOR20 OR FROM THE TERMINAL.

THE FILE FOR20 IS IN THE FOLLOWING FORMAT. EACH OF THE LINES BELOW REPRESENTS 1 LINE IN THE FILE.

THE TITLE IN A80 FORMAT
NUMBER OF VALUES TO COMPUTE IN I7 FORMAT
ORDER OF THE RECURRENCE RELATION IN I7 FORMAT
MATRIX TYPE: 0 = AUG, 1 = RED, IN I7 FORMAT
NUMBER OF INITIAL CONDITIONS IN I7 FORMAT
NUMBER OF TERMINAL CONDITIONS IN I7 FORMAT
THE JOB PARAMETER IN I7 FORMAT
VALUE OF TAU IN E16.8 FORMAT

INITIAL CONDS Y(0) ... Y(I), IN E16.8 FORMAT

TERMNAL CONDS Y(N-J+1) ... Y(N), IN E16.8 FORMAT

ON RETURN

THIS ROUTINE HAS NO PARAMETERS PASSED BACK, BUT
CREATEs A FILE FOR21 AS OUTPUT. THIS FILE
CONTAINS ALL OF THE NEEDED INFORMATION ABOUT
THE ASSOCIATED LINEAR SYSTEM.

FILE FOR21 IS IN THE FOLLOWING FORMAT.
THE TITLE IS ON THE FIRST LINE IN A80 FORMAT
NUMBER OF ROWS IN THE MATRIX IN I7 FORMAT
NUMBER OF COLUMNS IN THE MATRIX IN I7 FORMAT
THE JOB PARAMETER IN I7 FORMAT
THE MATRIX IS HERE IN ROW BY COLUMN FORM
ALL VALUES ARE IN E16.8 FORMAT
THE RESULT VECTOR B IN E16.8 FORMAT
THE CONDS VECTOR IN E16.8 FORMAT

SMTGEN VERSION DATED 12/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

SUBROUTINES SGTVAL, JOBHLP

INTERNAL VARIABLES

INTEGER NYI, NYT, N, P, ITYPE, JOB, I, J, K, L, M, R
CHARACTER*80 TITLE, ANS*1
REAL A(50,50), YI(50), YT(50), B(50), C(50),
      CONDS(52), TAU

WHERE
A REAL(50,50)
THE GENERATED MATRIX.

YI REAL(50)
VECTOR OF INITIAL CONDITIONS.

YT REAL(50)
VECTOR OF TERMINAL CONDITIONS.

B REAL(50)
THE RESULT VECTOR.

CONDS REAL(50)
THE CONDITIONS VECTOR, THIS IS USED TO RECONSTRUCT THE COMPLETE SOLUTION FOR COMPARISON.

REAL(52)
VECTOR FOR THE OBTAINING 1 ROW OF COEFFICIENTS FROM SGTVAL.

TAU
REAL
OPTIONAL VALUE THAT CAN BE USED TO GENERATE THE ROWS IN THE MATRIX.

NYI
INTEGER
THE NUMBER OF INITIAL CONDITIONS.

NYT
INTEGER
THE NUMBER OF TERMINAL CONDITIONS.

N
INTEGER
THE NUMBER OF ROWS IN THE MATRIX.

P
INTEGER
THE NUMBER OF COLUMNS IN THE MATRIX.

ITYPE
INTEGER
THE TYPE OF MATRIX TO GENERATE, 0 = AUGMENTED, 1 = REDUCED.

JOB
INTEGER
THIS IS THE JOB PARAMETER. THIS WILL DETERMINE WHICH DECOMPOSITIONS AND SOLUTION METHODS WILL BE USED.

TITLE
CHARACTER*80
THE TITLE TO BE USED ON THE OUTPUT.

ANS
CHARACTER*1
FOR ANSWERS TO INTERACTIVE QUESTIONS.

I,J,K,L,M,N,R
INTEGER
LOOP COUNTERS AND TEMPS.

INITIALIZE THE MATRIX.

DO 20 I = 1,50
  DO 10 J = 1,50
    A(I,J) = 0.0
10    CONTINUE
20    CONTINUE
PROMPT FOR AND ACCEPT THE SOURCE OF INPUT.

WRITE(5,1001)
FORMAT(/1X,'WOULD YOU LIKE TO READ AN EXISTING'
+ '/ FILE FOR THAT CONTAINS THE INITIAL '
+ '/ INFORMATION (Y) OR CREATE A NEW FILE '
+ '/ FOR INTERACTIVELY (N)? ')

READ(5,1002)ANS
FORMAT(A1)

DETERMINE THE SOURCE OF INPUT, SKIP THE
INTERACTIVE SECTION IF THE INPUT IS TO COME
FROM A FILE.

IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')GOTO 50

INTERACTIVELY ACCEPT ALL OF THE INFORMATION
NEEDED TO GENERATE THE MATRIX.

WRITE(5,1003)
FORMAT(1X,'PLEASE ENTER THE TITLE YOU WOULD',/
' LIKE TO USE FOR THE OUTPUT (80 CHARs MAX)' )
READ(5,1004)TITLE
FORMAT(A80)

WRITE(5,1005)
FORMAT(1X,'PLEASE ENTER THE NUMBER OF VALUES',
' TO COMPUTE ')
READ(5,1006)N
FORMAT(I7)

WRITE(5,1007)
FORMAT(1X,'PLEASE ENTER THE ORDER OF THE',
' RECURRENCE RELATION ')
READ(5,1006)P

WRITE(5,1008)
FORMAT(1X,'PLEASE ENTER THE TYPE OF MATRIX',
' TO GENERATE:'
+ '/ AUGMENTED (0), REDUCED (1) ')
READ(5,1006)ITYPE

WRITE(5,1009)
FORMAT(1X,'PLEASE ENTER THE JOB PARAMETER',
' OR 0 FOR HELP ')
READ(5,1006)JOB

CHECK IF HELP IS DESIRED IN SETTING UP THE
JOB PARAMETER AND CALL JOBHLP IF NEEDED.

IF(JOB.EQ.0)CALL JOBHLP(JOB)

WRITE(5,1010)
1010 FORMAT(1X,'PLEASE ENTER THE NUMBER OF INITIAL',
+ ' CONDITIONS DESIRED $)
READ(5,1006)NYI

WRITE(5,1011)
1011 FORMAT(1X,'PLEASE ENTER THE NUMBER OF ',
+ 'TERMINAL CONDITIONS DESIRED $)
READ(5,1006)NYT

PROMPT FOR AND READ EACH INITIAL CONDITION.

DO 30 I = 0,NYI-1
   WRITE(5,1012)I
   FORMAT(1X,'PLEASE ENTER THE INITIAL ',
   + 'CONDITION FOR F(',I2,') $)
READ(5,1013)YI(I+1)
30 CONTINUE

PROMPT FOR AND READ EACH TERMINAL CONDITION.

DO 40 I = N-NYT,N-1
   WRITE(5,1014)I
   FORMAT(1X,'PLEASE ENTER THE TERMINAL ',
   + 'CONDITION FOR F(',I2,') $)
READ(5,1013)YT(I-(N-NYT)+1)
40 CONTINUE

WRITE(5,1015)
1015 FORMAT(1X,'PLEASE ENTER A VALUE FOR TAU ',
+ '0.0 IF NOT NEEDED $)
READ(5,1013)TAU

WRITE ALL OF THE VALUES TO FOR20 SO THAT IT CAN BE REUSED AT A LATER TIME.

WRITE(20,1004)TITLE

WRITE(20,1016)N,P,ITYPE,NYI,NYT,JOB,TAU
1016 FORMAT(/I7//)E16.8

WRITE(20,1017)YI(I),I = 1,NYI
1017 FORMAT(/50E16.8)
WRITE(20,1017)(YT(I), I = 1, NYT)


GOTO 60

READ ALL OF THE INFORMATION FROM THE FILE FOR20 AFTER REWINDING THE FILE.

REWIND(20)

READ(20,1004) TITLE
READ(20,1016) N, P, ITYPE, NYI, NYT, JOB, TAU
READ(20,1017)(YI(I), I = 1, NYI)
READ(20,1017)(YT(I), I = 1, NYT)

DETERMINE WHICH MATRIX FORM IS TO BE GENERATED. IF THE REDUCED FORM IS REQUIRED, SKIP TO THE SECOND HALF, THIS SECTION IS FOR THE GENERATION OF THE AUGMENTED FORM OF THE ASSOCIATED LINEAR SYSTEM OF EQUATIONS.

IF(ITYPE.EQ.1) GOTO 130

SET A FLAG IN THE CONDS VECTOR TO INDICATE THAT THE AUGMENTED FORM WAS USED. THIS WILL BE IMPORTANT WHEN RECONSTRUCTING THE COMPLETE SOLUTION FOR COMPARISON.

CONDS(51) = -1

SET THE SIZE OF THE MATRIX

M = N
N = M-P+NYI+NYT

FILL IN THE TOP PART OF THE AUGMENTED MATRIX FOR THE NYI INITIAL CONDITIONS.

DO 70 I = 1, NYI
    A(I,I) = 1.0
    CONTINUE

SET THE BOUND FOR THE LOWER PART.

K = M - NYT + 1
FILL IN THE BOTTOM PART OF THE AUGMENTED MATRIX
FOR THE NYT TERMINAL CONDITIONS.

DO 80 I = N-NYT+1,N
   A(I,K) = 1.0
   K = K + 1
CONTINUE

SET THE START FOR THE INNER ROWS OF THE MATRIX.

R=1
L = P + 1

FILL IN THE INNER ROWS OF THE MATRIX.

DO 100 I = NYI+1,N-NYT

GET ONE ROW OF COEFFICIENTS.

CALL SGTVAL(C,TAU,R)

   R = R + 1
   K = 0

PUT THE VALUES IN.

   DO 90 J = L,L - P, -1
       IF(J.GT.M)GOTO 90
       K = K + 1
       A(I,J) = C(K)
   CONTINUE

   L = L + 1

CONTINUE

PUT THE INITIAL AND TERMINAL CONDITIONS INTO
THE RESULT VECTOR.

DO 110 I = 1,NYI
   B(I) = YI(I)
CONTINUE

K = 1
DO 120 I = N-NYT+1,N
   B(I) = YT(K)
   K = K + 1
CONTINUE

THIS ENDS THE AUGMENTED PORTION SKIP TO THE
OUTPUT SECTION.
GOTO 230

THIS IS THE SECTION TO GENERATE THE REDUCED FORM OF THE MATRIX.

SET THE SIZE OF THE MATRIX.

130 N = N - (NYI+NYT)
M = N + P - (NYI+NYT)

SET THE START POINT AND FILL THE MATRIX WITH THE VALUES.

R = 1
DO 160 I = 1,N
   K = P + I - NYI
   L = 1

GET THE COEFFICIENTS FROM SGTVAL.

CALL SGTVAL(C,TAU,R)
R = R + 1

PUT THE VALUES IN THE MATRIX.

DO 150 J = K,K-P,-1
   IF (J.LT.1 .OR. J.GT.M) GOTO 140
   A(I,J) = C(L)
   L = L + 1
140 L = L + 1
150 CONTINUE
160 CONTINUE

COMPUTE THE INITIAL VALUES AND PUT THEM IN THE RESULT VECTOR.

DO 180 I = 1,NYI
   L = P + 1
   DO 170 J = I,NYI
      B(I) = B(I) - C(L) * YI(J)
      L = L - 1
170 CONTINUE
180 CONTINUE

COMPUTE THE TERMINAL VALUES AND PUT THEM IN THE RESULT VECTOR.

DO 200 I = M-NYT+1,M
   L = 1
   DO 190 J = I-(M-NYT),1,-1
      B(I) = B(I) - C(L) * YI(J)
      L = L + 1
190 CONTINUE
200 CONTINUE
B(I) = B(I) - C(L) * YT(J)
L = L + 1

190 CONTINUE
200 CONTINUE

C SET FLAGS TO TELL HOW MANY INITIAL AND TERMINAL CONDITIONS ARE PRESENT. THIS WILL BE USED IN THE RECONSTRUCTION OF THE COMPLETE SOLUTION FOR COMPARISON.

CONDS(51) = NYI
CONDS(52) = NYT

C PUT THE INITIAL AND TERMINAL CONDITIONS IN THE CONDITIONS VECTOR FOR USE LATER.

C DO 210 I = 1,NYI
   CONDS(I) = YI(I)
210 CONTINUE

C DO 220 I = 1,NYT
   J = N + NYI + I
   CONDS(J) = YT(I)
220 CONTINUE

C THIS ENDS THE GENERATION OF THE REDUCED FORM.

C WRITE THE NECESSARY INFORMATION TO FOR21, TO BE USED BY THE REST OF THE SOFTWARE.

230 WRITE(21,1004)TITLE
196 FORMAT(A80)

C WRITE(21,1018)N,M,JOB
1018 FORMAT(2(I7/),I7)

C DO 240 I = 1,N
   WRITE(21,1019)(A(I,J),J = 1,M)
1019 FORMAT(52E16.8)
240 CONTINUE

C WRITE(21,1019)(B(I),I = 1,M)
C WRITE(21,1019)(COND(I),I = 1,52)
C RETURN
END
SUBROUTINE SQRLST(X,LDX,N,P,Y,TOL,B,RSD,K,JPVT,
+ QRAUX,WORK)
C
INTEGER LDX,N,P,K,JPVT(1)
REAL X(LDX,1),Y(1),TOL,B(1),RSD(1),QRAUX(1),
+ WORK (1)
C
SQRLST IS A SUBROUTINE TO COMPUTE LEAST SQUARES
SOLUTIONS TO THE SYSTEM X * B = Y,
WHICH MAY BE EITHER UNDER-DETERMINED OR OVER-
DETERMINED. THE USER MAY SUPPLY A TOLERANCE
TO LIMIT THE COLUMNS OF X USED IN COMPUTING
THE SOLUTION. IN EFFECT, A SET OF COLUMNS
WITH A CONDITION NUMBER APPROXIMATELY BOUNDED
BY 1/TOL IS USED, THE OTHER COMPONENTS OF B
BEING SET TO ZERO.
C
ON ENTRY
X REAL(LDX,P), WHERE LDX.GE.N
CONTAINS THE N X P COEFFICIENT MATRIX.
LDX INTEGER
LDX IS THE LEADING DIMENSION OF X.
N INTEGER
N IS THE NUMBER OF ROWS OF X.
P INTEGER
P IS THE NUMBER OF COLUMNS OF X.
Y REAL(N)
Y CONTAINS THE RIGHT HAND SIDE OF THE
RECURRENT RELATION MATRIX EQUATION.
TOL REAL
TOL IS THE NONNEGATIVE TOLERANCE USED
TO DETERMINE THE SUBSET OF COLUMNS OF X
INCLUDED IN THE SOLUTION. IF TOL IS
ZERO, A FULL COMPLEMENT OF MIN(N,P)
COLUMNS IS USED.
JPVT INTEGER(P)
JPVT IS AN ARRAY USED BY SQRDC
FOR PIVOT INFORMATION.
QRAUX REAL(P)
AN ARRAY USED BY SQRDC AND SQRLS
CONTAINS OUTPUT FROM THE DECOMPOSITION.
WORK REAL(P)
WORK IS A WORK ARRAY USED BY SQRDC.

ON RETURN
X REAL(N,P)
   CONTAINS THE OUTPUT ARRAY FROM SQRDC.

B REAL (P)
   B CONTAINS THE SOLUTION VECTOR.
   COMPONENTS CORRESPONDING TO COLUMNS NOT
   USED ARE SET TO ZERO.

RSD REAL(N)
   RSD CONTAINS THE RESIDUAL VECTOR Y-X*B.

K INTEGER
   K CONTAINS THE NUMBER OF COLUMNS USED
   IN THE SOLUTION.

ON RETURN THE ARRAYS X, JPVT AND QRAUX CONTAIN
   THE USUAL OUTPUT FROM SQRDC, SO THAT THE QR
   DECOMPOSITION OF X WITH PIVOTING IS FULLY
   AVAILABLE TO THE USER. IN PARTICULAR, COLUMNS
   JPVT(1), JPVT(2),...,JPVT(K) WERE USED IN THE
   SOLUTION, AND THE CONDITION NUMBER ASSOCIATED
   WITH THOSE COLUMNS IS ESTIMATED BY
   ABS(X(1,1)/X(K,K)).

SQRIST VERSION DATED 12/85
   DENNIS KAPENGA,
   WESTERN MICHIGAN UNIVERSITY.
   ADAPTED FROM ROUTINE IN LINPACK MANUAL.

SUBROUTINES SQRDC AND SQRSL.

INTERNAL VARIABLES.

   INTEGER INFO, J, KK, M
   REAL T

   THE INTERNAL VARIABLES ARE USED FOR TEMPS AND
   LOOP COUNTERS.

   INITIALIZE JPVT SO THAT ALL COLUMNS ARE FREE.

   DO 10 J=1,P
      JPVT(J) = 0
   10 CONTINUE
REDUCE $X$.

CALL SQRDC$\left(X, LDX, N, P, QRAUX, JPVT, WORK, 1\right)$

DETERMINE WHICH COLUMNS TO USE.

\begin{align*}
K & = 0 \\
M & = \text{MIN}(N, P)
\end{align*}

DO 20 \(KK=1, M\)
\begin{align*}
& \text{IF} \left(\text{ABS}(X(KK, KK)) \leq \text{TOL} \cdot \text{ABS}(X(1, 1))\right) \\
& \quad \text{GO TO 30} \\
& \quad K = KK
\end{align*}

20 CONTINUE

30 CONTINUE

SOLVE THE TRUNCATED LEAST SQUARES PROBLEM.

\begin{align*}
& \text{IF} \ (K \neq 0) \\
& \quad \text{CALL SQRSL} \left(X, LDX, N, K, QRAUX, Y, RSD, RSD, B, \\
& \quad \quad \text{RSD}, \text{RSD}, 110, \text{INFO}\right)
\end{align*}

SET THE UNUSED COMPONENTS OF $B$ TO ZERO AND INITIALIZE $JPVT$ FOR UNSCRAMBLING.

DO 40 \(J=1, P\)
\begin{align*}
& \quad JPVT(J) = -JPVT(J) \\
& \quad \text{IF} \ (J \gt K) \ B(J) = 0.
\end{align*}

40 CONTINUE

UNSCRAMBLE THE SOLUTION.

DO 70 \(J=1, P\)
\begin{align*}
& \quad \text{IF} \ (JPVT(J) \gt 0) \ \text{GO TO 70} \\
& \quad K = -JPVT(J) \\
& \quad JPVT(J) = K \\
& \quad \text{CONTINUE}
\end{align*}

50 CONTINUE
\begin{align*}
& \quad \text{IF} \ (K \quad \text{EQ.} \quad J) \ \text{GO TO 60} \\
& \quad T = B(J) \\
& \quad B(J) = B(K) \\
& \quad B(K) = T \\
& \quad JPVT(K) = -JPVT(K) \\
& \quad K = JPVT(K) \\
& \quad \text{GO TO 50}
\end{align*}

60 CONTINUE

70 CONTINUE

RETURN
END
SUBROUTINE SQROUT(X,LDX,N,P,JOB,Y,RCOND,TITLE)

INTEGER N,P,LDX,JOB
REAL X(LDX,1),Y(1),RCOND
CHARACTER *80 MESSAG,TITLE

C
C THIS ROUTINE IS PROVIDED A MATRIX X AND RESULT
C VECTOR Y. THE ROUTINE WILL PRINT TO AN OUTPUT
C FILE POSSIBLY SEVERAL PIECES OF INFORMATION
C THAT ARE CALCULATED HERE. THE ROUTINE USES THE
C QR DECOMPOSITION TO ACHIEVE A Q AND R MATRIX
C SUCH THAT X = QR. THE PARAMETER JOB WILL TELL
C WHAT INFORMATION IS TO BE PRINTED. THE OPTIONS
C FOR PRINTING THE DECOMPOSED SYSTEM ALONG WITH
C OPTION OF PERFORMING PIVOTING TO DECOMPOSE THE
C MATRIX ARE INCLUDED.

ON ENTRY
X REAL(N,P)
THE MATRIX THAT WILL BE DECOMPOSED.

LDX INTEGER
THE LEADING DIMENSION OF X.

N INTEGER
THE NUMBER OF ROWS IN X.

P INTEGER
THE NUMBER OF COLUMNS OF X.

JOB INTEGER
THIS PARAMETER WILL TELL THE ROUTINE
WHAT TO CALCULATE AND PRINT OUT TO THE
OUTPUT FILE.
JOB IS A 3 DIGIT NUMBER OF THE FORM XYZ
WHERE

IF X IS 1, THEN THE ROUTINE WILL WRITE
THE DECOMPOSITION TO THE OUTPUT FILE AS
A Q AND R MATRIX BOTH OF SIZE N X N.

IF X IS 0, NO OUTPUT OF Q AND R WILL
APPEAR.

IF Y IS 1, THE ROUTINE WILL WRITE THE
SOLUTION OF THE EQUATION XB = Y TO THE
OUTPUT FILE AS A COLUMN VECTOR.

IF Y IS 0, NO SOLUTION WILL BE
CALCULATED OR PRINTED.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
IF Z IS 1, THEN THE ROUTINE WILL DO ITS CALCULATIONS USING PARTIAL PIVOTING.

IF Z IS 0, THEN NO PARTIAL PIVOTING WILL BE USED.

Y REAL(N) THE INITIAL SOLUTION VECTOR FOR THE EQUATION XB = Y.

TITLE CHARACTER*80 NAME OF THE PROBLEM TO APPEAR ON ALL OUTPUT GENERATED.

ON RETURN Y REAL(N) THE SOLUTION FOR XB = Y, WHERE THE INITIAL VECTOR Y WAS OVERWRITTEN BY THE SOLUTION VECTOR B.

RCOND REAL THE ESTIMATED CONDITION FOR THE MATRIX.

SQROUT VERSION DATED 7/85 DENNIS KAPENGA, WESTERN MICHIGAN UNIVERSITY.

SUBROUTINES CALLED SQRDC, SQRSL, SAROUT

INTERNAL VARIABLES

INTEGER JPVT(50),INFO,IJOB,IPVT(50),I,J,K,LI, LDQR
REAL WORK(50),QRAUX(50),QY(50),QTY(50),TEMP,
+ RSD(50),XB(50),Q(50,50),R(50,50),B(50),E(50)

WHERE

JPVT INTEGER(50) THIS IS THE PIVOT VECTOR USED TO DECODE THE QR MATRIX THAT IS SENT BACK FROM LINPACK.

IJOB INTEGER A TEMP TO STORE JPVT WHEN RECOVERING THE MATRIX DURING PIVOTING.
INFO INTEGER
  USED BY LINPACK FOR ERROR CHECKING.

I,J,K INTEGER
  LOOP COUNTERS AND ARRAY INDICES.

L INTEGER
  USED AS A INTEGER TEMP.

LDQR INTEGER
  LEADING DIMENSION OF Q AND R.

QRAUX REAL(50)
  USED TO STORE THE EXTRA TERM NEEDED FOR
  FINDING THE L MATRIX FROM LINPACK.

WORK REAL(50)
  USED FOR WORK SPACE BY THE LINPACK
  ROUTINES.

Q REAL(50,50)
  THE Q MATRIX IN THE N X N FORM.

R REAL(50,50)
  THE R MATRIX IN THE N X N FORM.

E REAL(50)
  USED FOR SETTING UP ELEMENTARY COLUMN
  VECTORS TO RETRIEVE THE Q MATRIX.

QY REAL(50)
  THE SOLUTION OF THE PRODUCT Q * Y.

QTY REAL(50)
  THE SOLUTION OF THE PRODUCT Q
  TRANSPOSE * Y.

RSD REAL(50)
  THE RESIDUALS FROM THE SOLUTION OF AN
  EQUATION BY LINPACK.

XB REAL(50)
  THE SOLUTION OF THE PRODUCT X * B.

B REAL(50)
  THE RESULT OF THE EQUATION X * B = Y
  WHERE Y IS AN INITIAL VECTOR.

TEMP REAL
  USED AS A REAL TEMP.
CHECK TO SEE IF PARTIAL PIVOTING WAS REQUIRED AND SET IJOB.

LDQR = 50
IJOB = 0

IF(JOB/2*2.NE.JOB)IJOB=1

DECOMPOSE THE MATRIX X INTO Q AND R. ESTIMATE THE CONDITION OF THE MATRIX.

CALL SQRDC(X,LDX,N,P,QRAUX,JPVT,WORK,IJOB)
RCOND = X(N,N) / X(1,1)

CHECK TO SEE IF THE DECOMPOSITION IS TO BE PRINTED.

IF(JOB.LT.100)GOTO 120

RETRIEVE THE R MATRIX.

DO 20 I = 1,N
   DO 10 J = I,P
      R(I,J) = X(I,J)
   10 CONTINUE
20 CONTINUE

RETRIEVE THE Q MATRIX BY UTILIZING THE SOLVE ROUTINES OPTION TO RETURN QY. WHERE Y WILL BE THE ELEMENTARY COLUMN VECTORS ONE AT A TIME TO OBTAIN THE MATRIX Q 1 COLUMN AT A TIME.

DO 40 I = 1,P
   E(I)=1.0
   CALL SQRSL(X,LDX,N,P,QRAUX,E,QY,QTY,B,RSD,XB,01000,INFO)
   +
   DO 30 J = 1,N
      Q(I,J)=QTY(J)
30 CONTINUE
   E(I)=0.0
40 CONTINUE

UNDO THE PIVOTING IF IT WAS PERFORMED.

IF(IJOB.NE.1)GOTO 90

COPY JPVT INTO IPVT.
DO 50 I = 1, P
   IPVT(I) = JPVT(I)
50 CONTINUE

USE THE PIVOT VECTOR TO INTERCHANGE THE COLUMNS
OF THE MATRIX TO OBTAIN THE CORRECT SOLUTION.

DO 80 I = 1, P
   DO 70 J = 1, P
      IF (IPVT(J) .NE. I) GO TO 70
   C
      DO 60 K = 1, N
         TEMP = R(K, I)
         R(K, I) = R(K, J)
         R(K, J) = TEMP
50 CONTINUE

C
C ALSO INTERCHANGE THE PIVOT VECTOR.

    L = IPVT(I)
    IPVT(I) = IPVT(J)
    IPVT(J) = L
70 CONTINUE
80 CONTINUE

C
C WRITE THE Q AND R MATRIX TO THE OUTPUT FILE
ALONG WITH A TITLE.
C
90 IF (IJOB .NE. 0) GO TO 100
C
MESSAG = 'THE Q MATRIX WITH NO PARTIAL PIVOTING'
CALL SAROUT(Q, LDQR, N, P, MESSAG, TITLE)
C
MESSAG = 'THE R MATRIX WITH NO PARTIAL PIVOTING'
CALL SAROUT(R, LDQR, N, P, MESSAG, TITLE)

GOTO 110
C
100 MESSAG = 'THE Q MATRIX WITH PARTIAL PIVOTING'
   CALL SAROUT(Q, LDQR, N, P, MESSAG, TITLE)
C
MESSAG = 'THE R MATRIX WITH PARTIAL PIVOTING'
CALL SAROUT(R, LDQR, N, P, MESSAG, TITLE)

GOTO 110
C
110 CONTINUE

C
C DETERMINE IF THE SYSTEM IS TO BE SOLVED FOR B.
C
120 IF (JOB .LT. 10 .OR. JOB .EQ. 100 .OR. JOB .EQ. 101)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
SOLVE THE SYSTEM FOR \( Y \) AND OBTAIN \( B \).

CALL SQRSL(X,LDX,N,P,QRAUX,Y,QY,QTY,B,RSD,XB,30100,INFO)

IF(IJOB.NE.1)GOTO 150

UNDO THE PIVOTING IF IT WAS USED.

DO 140 I = 1,P
   DO 130 J = 1,P
      IF(JPVT(J).NE.I)GOTO 130
      TEMP=B(I)
      B(I)=B(J)
      B(J)=TEMP
      I=JPVT(I)
      JPVT(I)=JPVT(J)
      JPVT(J)=L
   130 CONTINUE
140 CONTINUE

CHECK IF PIVOTING WAS USED IN THE DECOMPOSITION
OF A AND WRITE THE PROPER HEADING ALONG WITH
THE SOLUTION VECTOR \( X \) TO THE OUTPUT FILE.

IF(IJOB.NE.0)GOTO 160

MESSAG = 'THE SOLUTION FOR \( Ay=b \) USING QR '
   + '/WITH NO PIVOTING'
CALL SAROUT(B,LDX,P,1,MESSAG,TITLE)

GOTO 170

MESSAG = 'THE SOLUTION FOR \( Ay=b \) USING QR WITH '
   + '/PIVOTING'
CALL SAROUT(B,LDX,P,1,MESSAG,TITLE)

PLACE THE SOLUTION IN \( Y \) FOR RETURNING TO THE
CALLING PROGRAM.

DO 180 I = 1,P
   Y(I) = B(I)
180 CONTINUE

RETURN

END
PROGRAM SRECUR

THIS IS THE MAIN PROGRAM FOR THE SOFTWARE DEVELOPED FOR THE EXPERIMENTATION AND COMPARISON OF DIFFERENT SOLUTION METHODS TO RECURRENCE RELATIONS. THIS MAIN PROGRAM, ALONG WITH MANY SUBROUTINES, AND THE LINPACK LIBRARY FORM THE COMPLETE SOFTWARE ENVIRONMENT. THIS SOFTWARE IS PART OF A THESIS ON THE COMPARISON AND INVESTIGATION INTO THE COMPUTER SOLUTION OF RECURRENCE RELATIONS. FOR A MORE COMPLETE EXPLANATION OF THE SOFTWARE, THE THESIS SHOULD BE CONSULTED.

THIS MAIN PROGRAM IS RESPONSIBLE FOR SETTING UP THE I/O CHANNELS TO THE DATA FILES. ONCE THE CHANNELS ARE SET UP, THE ORIGIN OF DATA IS DETERMINED. THE DATA MAY COME FROM AN EXISTING DATA FILE (FOR21) OR THE USER MAY CREATE A NEW FILE. THE CREATION OF THE NEW FILE IS DONE VIA THE SMTGEN ROUTINE.

ONCE THE DATA FILE FOR21 HAS BEEN CREATED, THE INFORMATION IS READ INTO THIS PROGRAM. THE INFORMATION IS THEN PASSED TO THE REST OF THE SOFTWARE FOR THE COMPUTATIONS AND SOME OF THE RESULTS TO BE PRINTED TO FOR22. WHEN THE RESULTS HAVE BEEN COMPUTED AND CONTROL IS RETURNED TO THE MAIN PROGRAM, A ROUTINE TO OUTPUT THE RESULTS OF THE COMPARISONS TO FOR22 IS CALLED.

SRECUR       VERSION DATED 12/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

SUBROUTINES    SSELCT, SCPOUT

OPEN STATEMENTS FOR EXTERNAL FILES.

OPEN(UNIT=20,FILE='FOR20')
OPEN(UNIT=21,FILE='FOR21')
OPEN(UNIT=22,FILE='FOR22')

INTERNAL VARIABLES

INTEGER P,N,LDA,JOB
REAL A(50,50),B(50),COMPAR(51,7),CONDS(52)
CHARACTER*1 ANS,TITLE*80
WHERE

A
REAL(50,50)
THE MATRIX TO HOLD THE ASSOCIATED
LINEAR SYSTEM OF EQUATIONS TO
DECOMPOSE AND SOLVE.

B
REAL(50)
VECTOR TO HOLD THE INITIAL VALUES FOR
THE LINEAR SYSTEM AX = B.

COMPAR
REAL(51,7)
MATRIX TO HOLD THE RESULTS FROM UP TO
SIX DIFFERENT SOLUTION METHODS. THE
COLUMNS OF COMPAR ARE DEFINED AS:
1. LU DECOMPOSITION WITH NO PIVOTING,
2. LU DECOMPOSITION WITH PIVOTING,
3. QR DECOMPOSITION WITH NO PIVOTING,
4. QR DECOMPOSITION WITH PIVOTING,
5. QR DECOMPOSITION AND LEAST SQUARES,
6. LU DECOMPOSITION AND WEIGHTED
INITIAL CONDITIONS, AND
7. ESTIMATE TO THE TRUE SOLUTION FOR
COMPARISON TO THE CALCULATIONS.

CONDS
REAL(52)
A VECTOR TO HOLD THE INITIAL AND
TERMINAL CONDITIONS PROVIDED BY THE
USER. FOR THE REDUCED FORM OF THE
SYSTEM CONDS(51) IS THE NUMBER OF
INITIAL CONDITIONS, AND CONDS(52) IS
THE NUMBER OF TERMINAL CONDS. FOR THE
AUGMENTED FORM OF THE SYSTEM, CONDS(51)
IS SET TO -1.

P
INTEGER
THE NUMBER OF COLUMNS IN A.

N
INTEGER
THE NUMBER OF ROWS IN A.

LDA
INTEGER
THE LEADING DIMENSION OF A.

JOB
INTEGER
THIS IS A PARAMETER THAT CONTROLS WHAT
IS TO BE COMPUTED AND PRINTED TO THE
OUTPUT FILE FOR22.
JOB IS A 7 DIGIT NUMBER TUVWXYZ,
WHERE

T IS FOR LU DECOMPOSITION WITH NO PARTIAL PIVOTING.

U IS FOR LU DECOMPOSITION WITH PARTIAL PIVOTING.

V IS FOR QR DECOMPOSITION WITH NO PARTIAL PIVOTING.

W IS FOR QR DECOMPOSITION WITH PARTIAL PIVOTING.

X IS FOR THE LEAST SQUARES COMPUTATIONS USING THE QR DECOMPOSITION.

Y IS FOR LU DECOMPOSITION WITH PARTIAL PIVOTING AND WEIGHTED INITIAL CONDITIONS.

Z TELLS IF THE ORIGINAL MATRIX A AND VECTOR B ARE TO BE PRINTED AT THE TOP OF THE OUTPUT FILE.

WHERE

T, U, V, W, X AND Y CAN ALL HAVE THE FOLLOWING VALUES:

0 SKIP THIS OPTION COMPLETELY.

1 MEANS TO DO THE PARTICULAR DECOMPOSITION AND PRINT THE RESULTS TO THE OUTPUT FILE. THE SYSTEM IS NOT SOLVED.

2 MEANS TO DO THE DECOMPOSITION, BUT NOT TO PRINT THE RESULTS OF THE DECOMPOSITION. THE SYSTEM IS SOLVED FOR X AND THIS RESULT IS PRINTED TO THE OUTPUT FILE.

3 MEANS TO DO THE DECOMPOSITION AND PRINT THE RESULTS TO THE OUTPUT FILE. THE SYSTEM IS SOLVED FOR X AND THIS RESULT IS PRINTED TO THE OUTPUT FILE.

Z HAS THE FOLLOWING VALUES:

0 THE MATRIX A AND VECTOR B WILL NOT BE PRINTED TO THE OUTPUT FILE.
THE MATRIX A AND VECTOR B WILL BE PRINTED AT THE TOP OF THE OUTPUT FILE.

ANS CHARACTER*1
USED FOR INTERACTIVE ANSWERS.

TITLE CHARACTER*80
THE USER PROVIDED TITLE THAT WILL APPEAR ON ALL OUTPUT.

INITIALIZE THE LEADING DIMENSION OF A.

LDA=50

PRINT HEADING

WRITE(5,1001)
1001 FORMAT(//,IX,'WELCOME TO THE LINEAR RECURRENCE'
+,'RELATION'/TESTING SOFTWARE ENVIRONMENT.'
+,' WRITTEN BY DENNIS/KAPENGA, WESTERN',
+,' MICHIGAN UNIVERSITY. FOR MORE INFORMATION'
+,' CONSULT THE ASSOCIATED THESIS.'/AN',
+,' OF LINEAR RECURRENCE RELATIONS.'//)

ASK THE USER FOR THE SOURCE OF THE INPUT DATA.

WRITE(5,1001)
1001 FORMAT(IX,'WOULD YOU LIKE TO READ AN EXISTING '
+,' FILE FROM FOR21 THAT CONTAINS THE MATRIX ',
+,' AND INITIAL INFORMATION (Y) OR WOULD YOU ',
+,' LIKE TO CREATE ONE USING THE SMTGEN ',
+,' (MATRIX GENERATING) ROUTINE (N) ? $)

READ THE ANSWER.

READ(5,1002)ANS
1002 FORMAT(A1)

DETERMINE THE SOURCE OF INPUT. IF THE USER IS GOING TO CREATE A NEW DATA FILE, THEN CALL SMTGEN. ONCE THE USER HAS CREATED THE DATA FILE FOR21, REWIND IT SO IT CAN BE READ INTO THE PROGRAM.

IF((ANS.EQ.'Y').OR.(ANS.EQ.'y'))GOTO 10

CALL SMTGEN
REWIND(21)

READ THE INFORMATION FROM FOR21.

READ(21,1003)TITLE
1003 FORMAT(A80)

READ(21,1004)N,P,JOB
1004 FORMAT(I7,2(/I7))

DO 20 I = 1,N
   READ(21,1005)(A(I,J),J=1,P)
20 CONTINUE

READ(21,1005)(B(I),I = 1,N)

READ(21,1005)(CONDS(I),I = 1,52)

CALL THE SELECTION ROUTINE TO DETERMINE AND
CALL THE PROPER ROUTINES TO PROVIDE THE
REQUESTED COMPUTATIONS IN THE JOB PARAMETER.

CALL SSELCT(A,LDA,N,P,JOB,B,COMPAR,TITLE)

CALL THE OUTPUT ROUTINE FOR THE COMPARISONS OF
THE DIFFERENT SOLUTION METHODS.

CALL SCPOUT(COMPAR,P,CONDS,TITLE)

STOP
END
SUBROUTINE SRIGHT(COMPAR,N,K)

REAL COMPAR(51,7)
INTEGER N,K

THIS ROUTINE RETURNS TO THE SCPOUT ROUTINE THE
COMPUTED VALUE FOR THE RECURRENCE RELATION. THE
VALUES RETURNED ARE USED TO ESTIMATE THE
RELATIVE ERRORS IN THE OTHER COMPUTATIONS. THE
ACCURATE VALUES ARE LOADED IN THE FIRST EMPTY
COLUMN OF THE COMPAR ARRAY AS INDICATED BY K.

THIS ROUTINE MUST BE CHANGED FOR EACH NEW
SOLUTION COMPUTED.

ON ENTRY
COMPAR REAL(51,7)
THE COMPAR ARRAY HOLDS ALL OF THE
COMPUTED RESULTS GATHERED FOR
COMPARISON.

N INTEGER
THE NUMBER OF ACCURATE VALUES TO
COMPUTE.

K INTEGER
A FLAG TO TELL WHICH COLUMN OF COMPAR
TO PLACE THE VALUES.

ON RETURN
COMPAR REAL(51,7)
THE ACCURATE VALUES ARE IN THE kth
COLUMN OF COMPAR.

SRIGHT VERSION DATED 12/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

INTERNAL VARIABLES

INTEGER I
I IS USED FOR A LOOP COUNTER.

THE CODE TO PLACE THE ACCURATE VALUES FOR THE
RECURRENCE RELATION FOR THE DESIRED SOLUTION IN
THE KTH COLUMN OF COMPAR GOES HERE.

RETURN
END
SUBROUTINE SSELCT(A,LDA,N,P,JOB,B,COMPAR,TITLE)

INTEGER JOB,LDA,N,P
REAL A(LDA,1),B(1),COMPAR(51,7)
CHARACTER *80 MESSAG,TITLE

THIS ROUTINE IS PROVIDED A MATRIX A AND A
VECTOR B. THE ROUTINE WILL SELECT MANY POSSIBLE
COMPUTATIONS ON THESE INPUTS. THE VARIABLE JOB
IS A PARAMETER THAT WILL TELL THE ROUTINE
EXACTLY WHAT TO PERFORM AND WHAT TO PRINT OUT
TO THE OUTPUT FILE. SOME OF THE OPTIONS INCLUDE
USING THE LU OR QR DECOMPOSITION FOR FACTORING,
THE LEAST SQUARES COMPUTATIONS. THE OPTION TO
USE PARTIAL PIVOTING FOR THE FACTORING
AND OPTIONS TO WRITE ANY OR ALL OF THE RESULTS
TO A SINGLE OUTPUT FILE FOR COMPARISON OF
RESULTS. ALSO RETURNED TO THE CALLING PROGRAM
IS THE ARRAY COMPAR, THIS ARRAY CONTAINS ALL OF
THE SOLUTIONS ASKED FOR BY THE JOB PARAMETER.
IN THE 51ST ENTRY FOR EACH COLUMN IS THE
ESTIMATED CONDITION NUMBER OF A, UNDER THE
DECOMPOSITION USED TO GENERATE THAT COLUMN.

ON ENTRY
A REAL(N,N)
THIS IS THE MATRIX TO BE DECOMPOSED.

LDA INTEGER
THE LEADING DIMENSION OF A.

N INTEGER
THE NUMBER OF ROWS OF A.

P INTEGER
THE NUMBER OF COLUMNS OF A.

B REAL(N)
THE INITIAL VECTOR TO BE USED IF THE
EQUATION AX = B IS TO BE COMPUTED TO
FIND X.

COMPAR REAL(51,7)
THE ARRAY TO STORE ALL OF THE RESULTS
FROM POSSIBLY 6 DIFFERENT SOLUTION
METHODS FOR COMPARISON.

TITLE CHARACTER*80
TITLE OF THE PROBLEM IT WILL APPEAR ON
ALL OUTPUT THAT IS GENERATED.
JOB INTEGER
THE JOB PARAMETER WILL DETERMINE WHAT
THIS ROUTINE IS TO CALCULATE AND WRITE.
JOB IS A 7 DIGIT NUMBER TUVWXYZ,

WHERE
T IS FOR LU DECOMPOSITION WITH NO
PARTIAL PIVOTING.
U IS FOR LU DECOMPOSITION WITH PARTIAL
PIVOTING.
V IS FOR QR DECOMPOSITION WITH NO
PARTIAL PIVOTING.
W IS FOR QR DECOMPOSITION WITH PARTIAL
PIVOTING.
X IS FOR THE LEAST SQUARES COMPUTATIONS
USING THE QR DECOMPOSITION.
Y IS FOR LU DECOMPOSITION WITH PARTIAL
PIVOTING AND WEIGHTED INITIAL
CONDITIONS.
Z TELLS IF THE ORIGINAL MATRIX A AND
VECTOR B ARE TO BE PRINTED AT THE TOP
OF THE OUTPUT FILE.

WHERE
T, U ,V ,W ,X AND Y CAN ALL HAVE THE
FOLLOWING VALUES:
0 SKIP THIS OPTION COMPLETELY.
1 MEANS TO DO THE PARTICULAR
DECOMPOSITION AND PRINT THE RESULTS
TO THE OUTPUT FILE. THE SYSTEM IS NOT
SOLVED.
2 MEANS TO DO THE DECOMPOSITION, BUT
NOT TO PRINT THE RESULTS OF THE
DECOMPOSITION. THE SYSTEM IS SOLVED FOR
X AND THIS RESULT IS PRINTED TO THE
OUTPUT FILE.
3 MEANS TO DO THE DECOMPOSITION AND PRINT
THE RESULTS TO THE OUTPUT FILE. THE
SYSTEM IS SOLVED FOR X AND THIS RESULT
IS PRINTED TO THE OUTPUT FILE.
Z HAS THE FOLLOWING VALUES:

0 THE MATRIX A AND VECTOR B WILL NOT BE PRINTED TO THE OUTPUT FILE.

1 THE MATRIX A AND VECTOR B WILL BE PRINTED AT THE TOP OF THE OUTPUT FILE.

ON RETURN
COMPAR REAL (N,6)
THE RESULTS OF THE REQUESTED SOLUTIONS WILL BE IN THIS ARRAY IN COLUMN FORM.
THE COLUMNS OF COMPAR ARE FILLED IN THE ORDER THEY ARE REQUESTED THROUGH THE JOB PARAMETER AS DEFINED ABOVE.

SELECT VERSION DATED 7/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

SUBROUTINES S2COPY, SLUOUT, SQROUT, SLSOUT, SAROUT

INTERNAL VARIABLES

INTEGER IJOB, JJOB(7), JCODE(3,6), I, J, LDT, ITMP
REAL TA(50,50), TB(50), RCOND

WHERE

IJOB INTEGER
A TEMP FOR PASSING JOB INFORMATION.

JJOB INTEGER(6)
THE PARTS OF THE JOB PARAMETER,
WHERE

JJOB(1) = T
JJOB(2) = U
JJOB(3) = V
JJOB(4) = W
JJOB(5) = X
JJOB(6) = Y
JJOB(7) = Z
AS DEFINED ABOVE.

JCODE INTEGER(3,5)
SET UP IN THE FOLLOWING DATA STATEMENT.

DATA JCODE/100, 010, 110, 101, 011, 111, 100, 010, 110,
101, 011, 111, 100, 010, 110, -101, -011, -111/
EACH ENTRY IS THE JOB CODE FOR THE
OTHER ROUTINES THAT NEED TO BE CALLED
TO PERFORM THE COMPUTATIONS.

I,J INTEGER
LOOP COUNTERS AND ARRAY INDICES.

LDT INTEGER
THE LEADING DIMENSION OF THE TEMP
ARRAYS TA AND TB.

ITEMP INTEGER
TEMP TO HELP SWAPS.

TA REAL(50,50)
A TEMP ARRAY TO SAVE THE VALUE OF THE
MATRIX A.

TB REAL(50)
A TEMP ARRAY TO SAVE THE VALUE OF THE
VECTOR B.

RCOND REAL
THE CONDITION NUMBER FOR A GIVEN THE
DECOMPOSITION.

MAKE THE COPIES OF THE ORIGINAL MATRIX A AND
THE VECTOR B. THE LEADING DIMENSIONS MUST BE
SWAPPED TO ENSURE PROPER COPIES.
SET THE DIMENSION OF THE TEMP ARRAYS.

LDT = 50
CALL SZCOPY(TA,A,LDT,N,P,TB,B,COMPAR,LDA,0)

SEPARATE THE JOB PARAMETER INTO ITS COMPONENTS.

DO 10 I = 7,1,-1
   JJOB(I) = ((FLOAT(JOB)/10.0) -
                      INT(JOB/10)) * 10 + .5
   JOB=JOB/10
10 CONTINUE

CHECK TO SEE IF THE MATRIX A AND THE VECTOR B
ARE TO BE PRINTED.

IF(JJOB(7).EQ.0)GO TO 20

WRITE OUT A HEADING AND THE MATRIX AND VECTOR.
MESSAG = 'THIS IS THE ORIGINAL MATRIX A'
    //AND VECTOR B'
CALL SAROUT(A,LDA,N,P,MESSAG,TITLE)

MESSAG = 'B='
CALL SAROUT(B,LDA,N,1,MESSAG,TITLE)

CHECK IF A LU DECOMPOSITION WITHOUT PIVOTING
COMPUTATION IS NEEDED.

20 IF(JJOB(1).EQ.0)GOTO 30

SET THE IJOB FOR THE CALCULATION AND CALL
S卢OUT.
AFTER RETURNING CALL SЗCOPY TO RESTORE A AND B.

IJOB = JCODE(JJOB(1),1)

CALL S卢OUT(A,LDA,N,IJOB,B,RCOND,TITLE)
COMПAR(51,1) = RCOND
CALL SЗCOPY(A,TA,LDA,N,P,B,TB,COMПAR,LDT,1)

CHECK IF A LU DECOMPOSITION WITH PIVOTING
COMPUTATION IS NEEDED.

30 IF(JJOB(2).EQ.0)GOTO 40

SET THE IJOB FOR THE CALCULATION AND CALL
S卢OUT.
AFTER RETURNING CALL SЗCOPY TO RESTORE A AND B.

IJOB = JCODE(JJOB(2),2)

CALL S卢OUT(A,LDA,N,IJOB,B,RCOND,TITLE)
COMПAR(51,2) = RCOND
CALL SЗCOPY(A,TA,LDA,N,P,B,TB,COMПAR,LDT,2)

CHECK IF A QR DECOMPOSITION WITHOUT PIVOTING
COMPUTATION IS NEEDED.

40 IF(JJOB(3).EQ.0)GOTO 50

SET THE IJOB FOR THE CALCULATION AND CALL
SЗROUT.
AFTER RETURNING CALL SЗCOPY TO RESTORE A AND B.

IJOB = JCODE(JJOB(3),3)

CALL SЗROUT(A,LDA,N,P,IJOB,B,RCOND,TITLE)
COMPAR(51,3) = RCOND
CALL SZCOPY(A,TA,LDA,N,P,B,TB,COMPAR,LDT,3)

CHECK IF A QR DECOMPOSITION WITH PIVOTING
   CALCULATION IS NEEDED.

IF(JJOB(4).EQ.0)GOTO 60

SET THE IJOB FOR THE CALCULATION AND CALL
   SQROUT.
   AFTER RETURNING CALL SZCOPY TO RESTORE A AND B.

IJOB = JCODE(JJOB(4),4)

CALL SQROUT(A,LDA,N,P,IJOB,B,RCOND,TITLE)
COMPAR(51,4) = RCOND
CALL SZCOPY(A,TA,LDA,N,P,B,TB,COMPAR,LDT,4)

CHECK IF A LEAST SQUARES WITH QR DECOMPOSITION
   CALCULATION IS NEEDED.

IF(JJOB(5).EQ.0)GOTO 70

SET THE IJOB FOR THE CALCULATION AND CALL
   SLSOUT.
   AFTER RETURNING CALL SZCOPY TO RESTORE A AND B.

IJOB = JCODE(JJOB(5),5)

CALL SLSOUT(A,LDA,N,P,IJOB,B,RCOND,TITLE)
COMPAR(51,5) = RCOND
CALL SZCOPY(A,TA,LDA,N,P,B,TB,COMPAR,LDT,5)

CHECK IF A LU DECOMPOSITION WITH WEIGHTED
   INITIAL CONDITIONS COMPUTATION IS NEEDED.

IF(JJOB(6).EQ.0)GOTO 80

SET THE IJOB FOR THE CALCULATION AND CALL
   SLUOUT.
   AFTER RETURNING CALL SZCOPY TO RESTORE A AND B.

IJOB = JCODE(JJOB(6),6)

CALL SLUOUT(A,LDA,N,IJOB,B,RCOND,TITLE)
COMPAR(51,6) = RCOND
CALL SZCOPY(A,TA,LDA,N,P,B,TB,COMPAR,LDT,6)

RETURN
END
SUBROUTINE SZCOPY(A, TA, LDA, N, P, B, TB, COMPARE, LDT, COL)

INTEGER N, LDA, COL, LDT, P
REAL A(LDA, 1), TA(LDT, 1), B(1), TB(1),
+ COMPARE(51, 7)

THE COPY ROUTINE SERVES 2 PURPOSES, FIRST TO
COPY THE ORIGINAL MATRIX A AND INITIAL VECTOR
B BACK FROM THE TEMP MATRIX TA AND THE TEMP
VECTOR TB. THE SECOND PURPOSE IS TO PLACE THE
RESULT VECTOR B INTO A SPECIFIED COLUMN OF THE
MATRIX COMPARE.

ON ENTRY

TA   REAL(N, N)
THE TEMP ARRAY FOR THE ORIGINAL MATRIX.

LDA   INTEGER
THE LEADING DIMENSION OF A.

N   INTEGER
THE NUMBER OF ROWS OF A, TA.

P   INTEGER
THE NUMBER OF COLUMNS OF A, TA.

TB   REAL(N)
THE TEMP ARRAY FOR THE ORIGINAL VECTOR.

COMPARE   REAL(N, 6)
THE ARRAY THAT HOLDS THE SOLUTIONS FOR
THE VARIOUS SOLUTION METHODS POSSIBLE.

LDT   INTEGER
THE LEADING DIMENSION OF TA.

COL   INTEGER
THE COLUMN OF COMPARE THAT VECTOR B'S
RESULTS WILL BE PLACED IN BEFORE B IS
RESTORED TO ITS ORIGINAL VALUES.

ON RETURN

A   REAL(N, N)
A IS RESTORED TO ITS ORIGINAL VALUE
FROM TA.

B   REAL(N)
B IS RESTORED TO ITS ORIGINAL VALUE FROM TB.

COMPAR REAL(N,6)
COMPAR WILL NOW INCLUDE THE RESULTS OF THE VECTOR B THAT WERE SENT DOWN.

SZCOPY VERSION DATED 7/85
DENNIS KAPENGA,
WESTERN MICHIGAN UNIVERSITY.

INTERNAL VARIABLES

INTEGER I,J

I,J INTEGER LOOP COUNTERS AND ARRAY INDICES.

PERFORM THE COPY OF TA TO A AND TB TO B.

DO 20 I = 1,N
   DO 10 J = 1,P
      A(I,J) = TA(I,J)
   10 CONTINUE

IF THE B IS TO BE SAVED, COPY IT INTO COMPAR IN THE COLUMN INDICATED BY COL STARTING IN THE FIRST ROW.

IF(COL.NE.0) COMPAR(I,COL) = B(I)
B(I)=TB(I)

20 CONTINUE

RETURN
END
BIBLIOGRAPHY


HACLA. See Wilkinson and Reinsch. 1971.


Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.


Piessens, R. 1985. Personal communication to J. Kapenga.


