

This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (<https://dspace.lboro.ac.uk/>) under the following Creative Commons Licence conditions.



For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

LOUGHBOROUGH
UNIVERSITY OF TECHNOLOGY
LIBRARY

AUTHOR/FILING TITLE

AUDISH, S E

ACCESSION/COPY NO.

113489/02

VOL. NO.

CLASS MARK

LOAN COPY

-1 JUL 1983

-6 JUL 1984

-1 JUL 1985

011-3489 02



THE NUMERICAL SOLUTION OF BANDED LINEAR SYSTEMS

BY GENERALIZED FACTORIZATION PROCEDURES

BY

SHAKER ELIAS AUDISH, B.Sc., M.Sc.

A Doctoral Thesis

Submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

of the Loughborough University of Technology

November, 1981.

SUPERVISOR: PROFESSOR D.J. Evans, Ph.D., D.Sc.

Department of Computer Studies

Loughborough University	
of Technology Library	
Due	Oct 82
Class	
Acc. No.	113489/02

DECLARATION

I declare that the following thesis is a record of research work carried out by me, and that the thesis is of my own composition. I also certify that neither this thesis nor the original work contained therein has been submitted to this or any other institution for a degree.

S.E. AUDISH.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Professor D.J. Evans for his continuous encouragement, capable guidance and supervision, and by helping me to alleviate the non-academic difficulties I experienced throughout the period of this research.

I am very grateful to all the staff of the Computer Centre and the Department of Computer Studies, and to friends at Loughborough, in particular Mr. M. Shuker, for their help during the difficult stages of this work.

My deep thanks go to Miss Judith M. Briers for the excellent and careful typing of the thesis.

Last, but by no means least, I am warmly indebted to my parents, sisters, brothers and other relatives for their patience, constant encouragement and financial support, especially to my brother, Abu-Ghazwan, whose personal efforts were undoubtedly instrumental in enabling me to complete this thesis.

CONTENTS

	<u>PAGE</u>
<u>Chapter 1:</u> INTRODUCTION	1
<u>Chapter 2:</u> MATHEMATICAL BACKGROUND	
2.1 Basic Concepts of Matrix Algebra.....	7
2.2 Direct and Iterative Methods for Solving Linear Systems of Equations.....	16
2.3 Contract Mapping Theorem, Newton's Method.....	32
2.4 Eigenvalue Problem.....	41
2.5 Evaluation of the Square Root of a Square Matrix.....	48
2.6 Main Properties of Continued Fractions.....	50
<u>Chapter 3:</u> NUMERICAL SOLUTION OF BOUNDARY VALUE PROBLEMS	
3.1 Different Numerical Approaches for Solving Boundary Value Problems.....	56
3.2 Finite-Difference Methods.....	61
3.3 Low-Order Discretization.....	64
3.4 High-Order Discretization.....	70
3.5 Finite Difference Methods for Partial Differential Equations.....	84
<u>Chapter 4:</u> NEW ALGORITHMIC METHODS FOR THE SOLUTION OF BANDED MATRIX EQUATIONS	
4.1 Introduction.....	97
4.2.1 Algorithm FICM1.....	98
4.2.2 Iterative Method of Solution (GITRM).....	100
4.2.3 Solution of the System $Qy=z$	108
4.2.4 A Polynomial Scheme for the Solution of the Modified Non-Linear System.....	113
4.2.5 Stability of the Method.....	115
4.2.6 Convergence of the Non-Linear System.....	119

	<u>PAGE</u>
4.2.7 Error Analysis for the Linear Systems Involved in the Algorithm FICM1.....	124
4.3.1 Algorithm FICM2.....	131
4.3.2 Derivation of the Algorithm Solution.....	136
4.3.3 Determination of the Elements of the Matrix Factors.....	148
4.3.4 Solution of Symmetric Linear Systems.....	159
4.3.5 Rounding Error Analysis.....	166
4.3.6 Convergence Analysis of the Iterative Procedure Applied in the Algorithm FICM2	168
4.4.1 Algorithm FIRM1.....	182
4.4.2 Algorithmic Solution of a Coupled System...	192
4.4.3 Determination of the Elements of the Rectangular Matrices U and L.....	206
4.5 Algorithm FICM5.....	221
4.6 Algorithm FIRM4.....	228

Chapter 5: NEW ALGORITHMIC METHODS FOR THE SOLUTION OF BLOCK
MATRIX EQUATIONS

5.1 Algorithm FICM3.....	238
5.2 Algorithm FIRM2.....	252
5.3 Algorithm FICM4.....	259
5.4 Algorithm FIRM3.....	268
5.5 Algorithm FICM6 and FIRM5.....	279

Chapter 6: APPLICATIONS TO ORDINARY DIFFERENTIAL EQUATIONS

6.1 Introduction.....	284
6.2 On the Implementation of the Procedure GITRM (Subsection 4.2.2).....	285
6.3 Non-Linear Equations Involved in FICM2 and (FIRM1).....	287

	<u>PAGE</u>
6.4 Numerical Examples of 2-Point Boundary Value Problems.....	292
6.5 Application of FIRM1 on Eigenproblems.....	302
 <u>Chapter 7:</u> APPLICATIONS TO PARTIAL DIFFERENTIAL EQUATIONS	
7.1 Introduction.....	308
7.2 On the Factorization Involved in BLOCKSOLVERS.....	309
7.3 Numerical Examples.....	314
7.4 Numerical Results and Discussion.....	323
 <u>Chapter 8:</u> CONCLUSIVE REMARKS AND FURTHER INVESTIGATIONS	
Part (A).....	331
Part (B).....	333
 REFERENCES	
 APPENDIX A	345
 APPENDIX B	350
 APPENDIX C	352

The work of this thesis mainly presents new direct computational algorithmic solvers for real linear systems of equations (of wide banded matrices) derived from the application of well-known finite-difference techniques to boundary value problems involving ordinary and partial differential equations. These algorithms are for illustrative purposes suitable for problems, not only differential equations with specific boundary conditions or two-point boundary value problems, but a wider class of differential equations can also be treated. They are applicable for partial differential equations where a banded matrix is obtained by using a high-order approximation such as a 9-point formula for the Laplace or Poisson equation. Also the application is extendable to higher order equations such as the Biharmonic equation. Whilst one type of the algorithm is suggested only for treating block linear systems, the other type is also applicable to these as well as their use in the point form applications to which they were basically proposed. The two types are respectively named in the last chapters of this thesis as BLOCKSOLVERS and BANDSOLVERS.

The two SOLVERS are categorised to suit two common kinds of problems, i) subjected to periodic boundary conditions and ii) those subjected to non-periodic or more commonly known, Dirichlet, Neumann and Robin conditions. Subsequently the factorisation procedure of the coefficient matrix takes place according to the type of the condition that the considered problem is subjected to. Precisely for a given matrix of order N with bandwidth $2r+1$, $r \geq 1$ ($N \geq 2r+1$), with type (i) the matrix is factorised into two invertible, *cyclic (or periodic)* upper and lower matrices of semi-bandwidth $r+1$, whilst with type (ii) the obtained factor matrices are *rectangular* upper and lower of size $(N \times (N+r))$ and $((N+r) \times N)$ respectively, and of semi-bandwidth $r+1$.

As an alternative approach to the conventional methods (as a LU-Decomposition), the elements of the factor matrices are obtained by adopting some iterative schemes whose convergence properties are investigated. This is applicable to the BANDSOLVERs, whilst in the BLOCKSOLVERs the factorisation procedure involves computing a matrix square root.

However, consistent with the demands of the new era of technology where high-speed computers are introduced, and the start of the revolution of micro-chips, the investigation for reliable computational methods is extensively broadening. Moreover, the emergence of parallel processing machines so far shows remarkable results on reducing the execution time for some particular numerical algorithms, although some reservations on storage demands still exist.

Numerous problems arise in the Mathematical Physics and Engineering fields which are still encountered by Numerical Analysts and other specialists for which no satisfactory solution procedures have been reached and not so for the foreseeable future.

Basically, the development of computational methods takes place in one of two directions: to obtain the solution *iteratively* or *directly*, and consequently it has become customary in literature to classify the conventional and new methods to these appropriate directions. It is known that no method has the merit of generality, but they are valued or preferred for certain problems according to many vital factors associated with the use of the computer such as the amount of storage required, computing time, levels of obtainable accuracy,... etc., and then the advantages and disadvantages of either method may accordingly be recognised or detected. The conventional types for both methods are discussed in Chapter 2. Here we present a brief indication to a few methods for both

types developed in recent years.

Iterative methods have witnessed considerable advances in the last three decades or so, in particular we refer to the contributions of Frankel, Young and others in the 50's to generalise the successive over-relaxation procedure (point form), and for the block case as given by Varga (1962) who also contributed earlier a method of normalisation of block systems so that a considerable reduction in arithmetic operations is implied (Cuthill and Varga (1959)). Other methods for sparse matrices may be found in Evans (1974). For certain cases, when the coefficient matrix of the considered linear system possess special properties some recent methods are suggested.

For example, when the matrix is symmetric and positive definite Gustafsson (1979) presents the so-called Modified Incomplete Choleski, prior to that the "Incomplete LU-Decomposition" for a symmetric M-matrix was proposed by Meijerink and van der Vorst (1977) in which both methods are based on the idea of splitting the matrix, and in the former seeking a suitable parameter to accelerate the iteration process is significant and important. Another method deals with non-negative types of matrices, as in Neumann and Plemmonns (1978) in which their work includes a study of linear stationary iterative methods with non-negative matrices for solving singular and consistent linear systems.

In direct methods too, the development in a similar period has progressed extensively, both in the theoretical and practical sides. In the former, for instance, the error analysis for the direct method contributed by Wilkinson has enabled the 'users' to predict or recognise the behaviour of the method, its stability and the bounds of the accuracy in the obtainable solution. On the other hand, fast methods have been suggested, such as in (Hockney (1965)) involving Fast Fourier transforms,

sparse factorisation by Evans (1971) and his work in the recent years. Other methods involving cyclic reduction as in Sweet (1974, 1977) or the spectral resolution methods introduced by Buzbee et al (1970). A comparison between point and block elimination schemes to solve block-tridiagonal systems and the stability for the latter scheme are given in Varah (1972); for the considered block matrix being symmetric and positive definite it is indicated in the same reference that Gene Golub has used the Choleski decomposition for this particular case. A fast numerical solution of linear systems of equations led to a block quindagonal coefficient matrix using a factoring and block elimination process as proposed by Bauer and Reiss (1972). Another type of method which deals with rather sparse matrices is suggested by Henderson and Wassyng (1978) in which the method exploits the zero elements below the diagonal of the given coefficient matrix, but the method shows superiority to Gaussian elimination only when the matrix is sparse strictly in the lower triangular part.

The presentation in this thesis is partitioned into seven chapters (excluding the current one) and may be outlined as follows.

In Chapter 2, the general mathematical background is included which involves the basic concepts, definitions and theorems; in addition to some conventional theoretical work such as, direct and iterative methods, the contract mapping theorem and Newton's method with a few of its variants. The chapter also covers some other topics which to a certain extent are directly related to the procedure of the new algorithms, such as the theory of the periodic continued fractions, the computation of a matrix square root by Newton's method, eigenvalue problem, etc.

As a matter of interest, the field of the applications for some of the algorithms, the 2-point boundary value problem concerning non-linear (or linear) ordinary differential equations is chosen. In relation to this

problem, the so-called iterative-deferred-correction technique is adopted. Thus, this technique has been covered considerably in Chapter 3. Also indicated in this chapter we extend the idea of using symmetric finite-difference formulae of high-order (or it is called in the appropriate chapter, high-order approximations) for the non-linear case, notably the work carried out by Shoosmith (1973) on the linear case is referred to. In fact, the motivation of considering such techniques is to provide us with the generality of the new algorithms indicated earlier, that is to deal with matrices of *any* bandwidth! Apart from a brief indication of the concepts involved in partial differential equations, the description of the discretisation schemes to specific continuous problems via using finite-difference approximations involve different computational molecules, is included in Chapter 3. In addition, because the chapter is devoted to the numerical solution of boundary value problems, thus an abbreviated description to some of the numerical approaches are made at the beginning, in particular, finite element methods followed by our main interest approach in this work, the finite-difference method.

The new suggested algorithms are presented in two chapters, 4 and 5. Chapter 4 includes the algorithms which are proposed for the pointwise problems (BANDSOLVERS). One of them is designed for the special case, when the coefficient matrix of the considered linear system is periodic and possesses constant elements. While the remaining BANDSOLVERS deal with the matrices of non-constant (generally, non-symmetric) elements for both periodic and non-periodic cases. The extension of these algorithms to certain skew-type matrices is also included. While Chapter 5 presents the BLOCKSOLVERS which in fact are considered as an extension to the BANDSOLVERS for special cases only.

The results of the numerical experimental work corresponding to the

algorithms of the last two chapters are given in Chapter 6 and 7 respectively. In these chapters some model problems for both ordinary and partial differential equations are introduced; in addition, a considerable discussion on the factorisation procedures applied to various common types of matrices in which some related aspects are included such as the rate of convergence of the involved iteration processes, etc. Eigenproblems are discussed in Sections 6.5 and 7.4. The tested examples as a whole may reflect to which type of matrices the new algorithms are both practical and applicable.

Finally, the main remarks in the light of this work are concluded in Chapter 8 with some recommendations for pursuing further investigations and extensions.

CHAPTER 2

MATHEMATICAL BACKGROUND

2.1 BASIC CONCEPTS OF MATRIX ALGEBRA

Numerical approaches such as finite-difference, finite element methods (see Chapter 3) are generally based on matrix algebra which by using its *concepts* the analysis of these methods or the solution process can be expressed in a suitable manner. In addition, in practice, the use of electronic computers enables matrix algebra to be an important tool in the application fields. In this presentation, we will emphasise the concepts which are (generally) associated with the subjects throughout this thesis.

The most important and well-known elementary concept is the *matrix* which is defined to be a rectangular array of ordered numbers and customarily denoted by a capital letter (our consideration is merely on *real* matrices). A matrix A is of size $(m \times n)$ if it has m rows and n columns. (Figure 2.1.1).

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,n-1} & a_{1,n} \\ a_{21} & a_{22} & \cdots & a_{2,n-1} & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n-1} & a_{m,n} \end{bmatrix}$$

FIGURE 2.1.1: A is an $m \times n$ matrix

The matrix A is said to be *square* (or *quadratic*) when $m=n$, and hence A is of *order* n (or m). When $m=1$, we have a *row vector*, and for $n=1$, a *column vector*, usually denoted by small underlined letters. The *transpose* of a matrix $A=[a_{i,j}]$ is written as A^T and obtained by interchanging the rows and columns of A , i.e. the element $a_{i,j}$ of A becomes $a_{j,i}$ of A^T . If $A=A^T$, then A is said to be *symmetric*, and anti-symmetric if $A=-A^T$ (obviously the two concepts are applicable for square matrices only), i.e. $a_{i,j}=a_{j,i}$ and $a_{i,j}=-a_{j,i}$ respectively. A square matrix (from now on any mentioned matrix is assumed square unless otherwise stated). A matrix A possesses an inverse, denoted by A^{-1} and is called a *non-singular* or *invertible* matrix (sometimes

this property is equivalent to say that A has linearly independent columns or rows), otherwise A is *singular*. On the other hand, if the determinant of A , which will be denoted by $\det(A)$, is zero then A is singular, otherwise (i.e. $\det(A) \neq 0$) A^{-1} does exist, and hence we have

$$AA^{-1} = A^{-1}A = I,$$

where I is the unit (identity) matrix.

Definition 2.1.1: (Pseudo-inverse, (Strang (1976)))

Given a rectangular ($m \times n$) matrix A which may not be invertable.

Its "inverse" which is denoted by A^+ is expressed in the form

$$A^+ = (A^T A)^{-1} A^T$$

where $A^T A$ is a square matrix of order n which can be inverted unless it is singular.

In this thesis we shall be mainly concerned with *banded* matrices.

Bandedness means that all elements beyond the bandwidth of the matrix are zero, i.e. for a banded matrix $A = [a_{i,j}]$ we can state the condition

$$a_{i,j} = 0 \text{ for } |j-i| > r$$

where $2r+1$ is the bandwidth of A .

If A has a large number of zeros, then it is said to be *sparse banded matrix*. In this chapter we may illustrate some examples of matrices such that the zero elements will be presented as a single zero notation, "0" and the non-zero elements will be denoted by "X".

Two types of bandwidth for matrix A are shown in Figure 2.1.2.

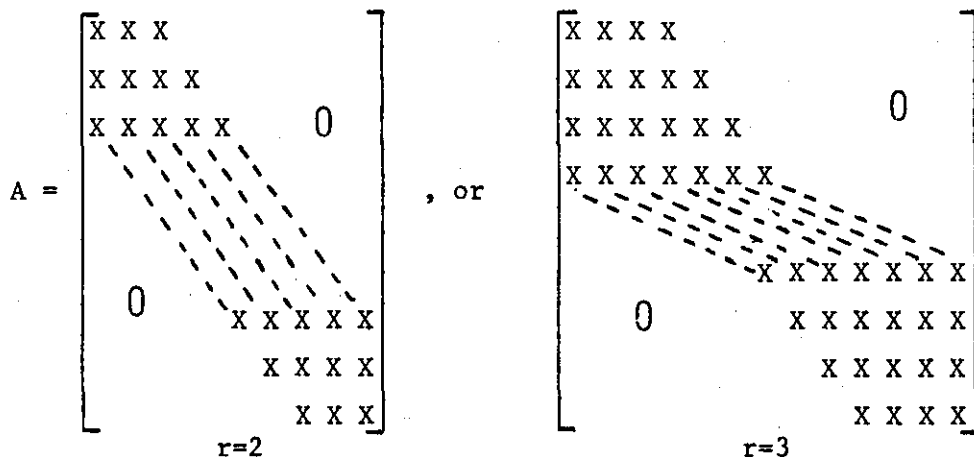


FIGURE 2.1.2: Banded matrices (pentadiagonal, septadiagonal)

If one half-bandwidth of a matrix is merely zero, then we have either an *upper* or *lower* triangular banded matrix. For example, $U=[u_{ij}]$ is upper triangular if $u_{ij}=0$ for $i>j$ and $L=[l_{ij}]$ is lower triangular, if $l_{ij}=0$ for $i<j$; also we have a *diagonal* matrix $D=[d_{ij}]$ if $d_{ij}=0$ for all $i\neq j$ and non-zero for d_{ii} (Fig.2.1.3).

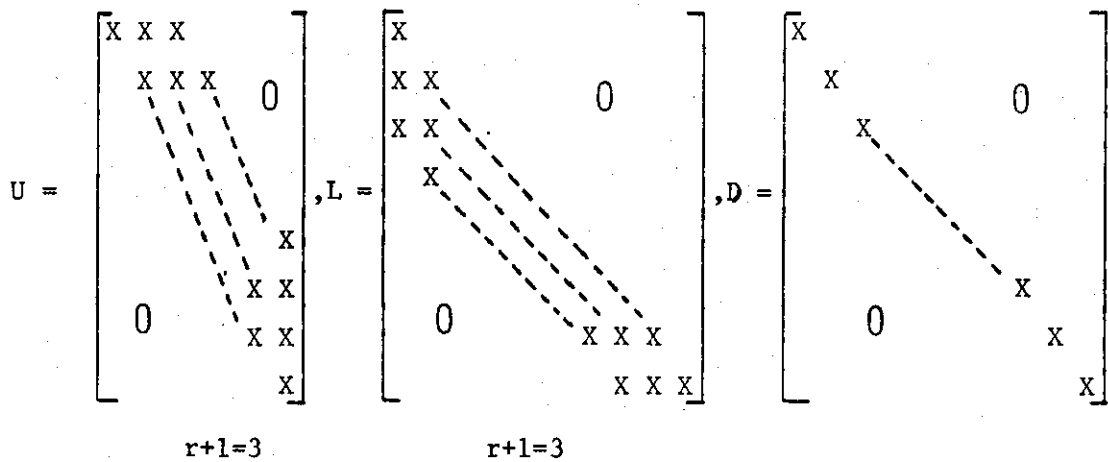


FIGURE 2.1.3: Upper, lower triangular and diagonal matrices

It may be important to indicate that we shall also consider banded matrices as presented in Fig.2.1.4 and consists of bandwidth $2r+1$ plus $\frac{r+1}{2}$ extra elements on each of the upper right hand and lower half hand corners.

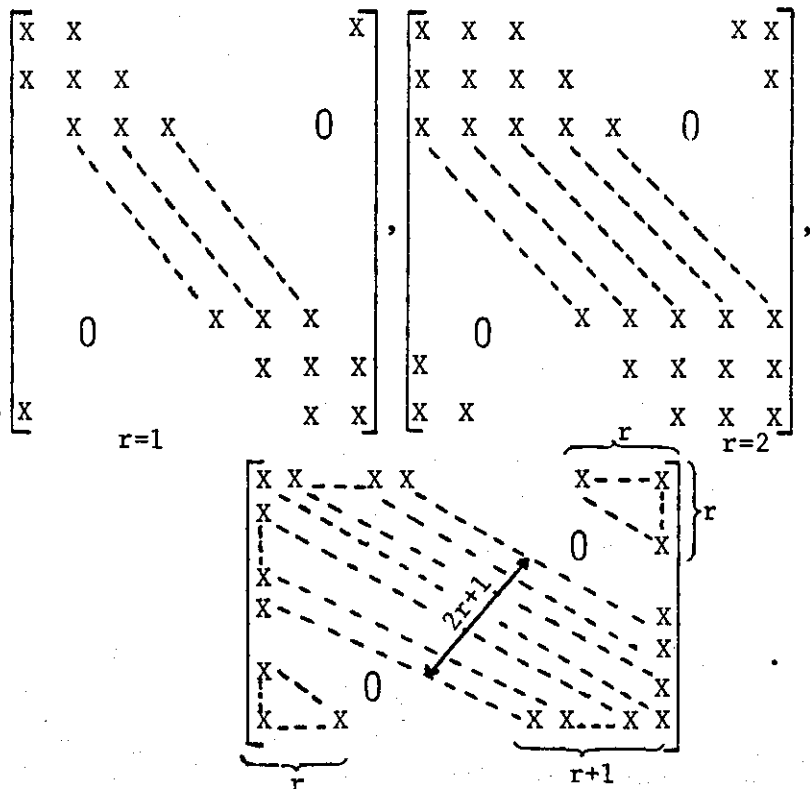


FIGURE 2.1.4: Banded matrices - Periodic type

Also we shall consider rectangular upper and lower banded matrices of bandwidth $r+1$, as in Fig.2.1.5.

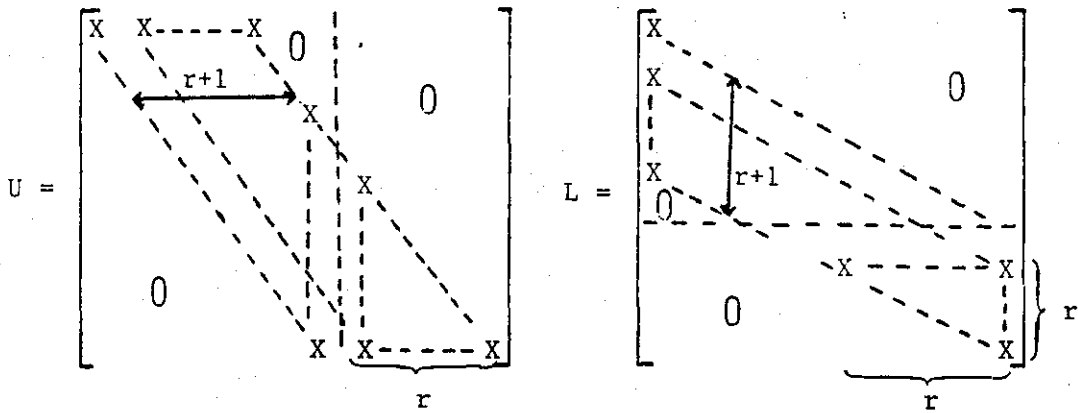


FIGURE 2.1.5: Rectangular banded matrices, U is $n \times (n+r)$ and L is $(n+r) \times n$.

We may classify the type of matrices shown in Fig.2.1.4 as of *periodic* type and in Fig.2.1.5 as non-periodic type. (see later chapter).

Definition 2.1.2: (Augmented matrix)

Given a system of linear equations $A\mathbf{x}=\mathbf{z}$, of order n , the augmented matrix is (A, \mathbf{z}) which has the form given in Fig.2.1.6.

$$\left[\begin{array}{ccc|c} a_{11} & \cdots & a_{1n} & z_1 \\ \vdots & & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} & z_n \end{array} \right]$$

FIGURE 2.1.6: Augmented matrix

Vector and Matrix Norms

In iterative solution processes using vectors and matrices, a measurement of convergence is usually required. Also, for direct solution procedures where the effect of rounding errors are considered. In this respect it is customary to measure the 'size' or magnitude of vectors and matrices by *norms*.

Definition 2.1.3:

The *norm* of an n -dimensional vector \underline{x} , written as $||\underline{x}||$, is a scalar (or number) satisfying the following three axioms:

- (1) $||\underline{x}|| \geq 0$ and $||\underline{x}|| = 0$ if and only if \underline{x} is a null vector,
- (2) $||\beta \underline{x}|| = |\beta| \cdot ||\underline{x}||$ for any scalar β (*homogeneity condition*)
- (3) $||\underline{x} + \underline{y}|| \leq ||\underline{x}|| + ||\underline{y}||$ for vectors \underline{x} and \underline{y} (*triangle inequality*).

Also
$$||\underline{x}|| - ||\underline{y}|| \leq ||\underline{x} - \underline{y}||. \quad (2.1.1')$$

Three vector norms are commonly used. These are:

Definition 2.1.4:

If $\underline{x} = [x_i]$, $i=1, 2, \dots, n$, then we have

(a) *infinite-norm* $||\underline{x}||_{\infty} = \max_i |x_i|$ (*uniform or Chebyshev norm*) (2.1.1)

(b) *one-norm* $||\underline{x}||_1 = \sum_{i=1}^n |x_i|$, (2.1.2)

(c) *two-norm* $||\underline{x}||_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{1/2}$ (*or Euclidean norm*). (2.1.3)

In fact, these norms are special cases of the general p -norm (or Hölder norm) given by, i.e.,

$$||\underline{x}||_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}, \quad p \geq 1, \quad (2.1.4)$$

where by setting p equal to ∞ , 1 and 2 in (2.1.4) yields the norms (2.1.1) to (2.1.3) respectively.

Analogous to the Definition 2.1.3, we proceed to present the definition of a matrix norm as well.

Definition 2.1.5:

A norm of a matrix A of order n , written as $||A||$, is a scalar such that the following four conditions fulfil

- (a) $||A|| > 0$ and $||A|| = 0$ if and only if $A=0$ (the null matrix),
- (b) $||\beta A|| = |\beta| \cdot ||A||$ for any scalar β (*homogeneity condition*),
- (c) $||A+B|| \leq ||A|| + ||B||$ for matrices A and B (*triangle inequality*)

and

(d) $||AB|| \leq ||A|| \cdot ||B||$ for matrices A and B (multiplicative triangular inequality).

The postulation of (d) in a matrix norm imposes the occurrence of matrix products.

Below are frequently used matrix norms:

$$||A||_{\infty} = \max_i \sum_{j=1}^n |a_{ij}| \quad (\text{the } \infty\text{-norm or maximum absolute column sum}) \quad (2.1.5)$$

$$||A||_1 = \max_j \sum_{i=1}^n |a_{ij}| \quad (\text{the 1-norm or maximum absolute row sum}) \quad (2.1.6)$$

$$||A||_2 = \{\text{maximum eigenvalue of the product } A^T A\}^{\frac{1}{2}} \quad (\text{spectral or Hilbert norm}). \quad (2.1.7)$$

Another type of norm which is used is the *Frobenius norm* which is denoted by $||A||_F$ and defined as follows:

$$||A||_F = \left(\sum_{i,j} |a_{i,j}|^2 \right)^{\frac{1}{2}}. \quad (2.1.8)$$

Further, since most applications of matrices are accompanied by vectors, therefore it is useful to apply the multiplicative triangular inequality norm (Definition 2.1.5) for the produce of a matrix and vector. Thus, for a product $A\underline{x}$ we have

$$||A\underline{x}|| \leq ||A|| \cdot ||\underline{x}||. \quad (2.1.9)$$

This inequality relation may lead to the following definition:

Definition 2.1.6:

If matrix A and vector \underline{x} have the norms $||A||$ and $||\underline{x}||$ respectively, then these two norms are said to be *compatible* provided that (2.1.9) is fulfilled.

Definition 2.1.7:

A *subordinate* or *induced* matrix norm $||A||$ is defined as follows

$$||A|| = \sup_{\underline{x} \neq 0} \frac{||A\underline{x}||}{||\underline{x}||}. \quad (2.1.10)$$

Sometimes (2.1.10) is written in an equivalent form, i.e.

$$\|A\| = \sup_{\|\underline{x}\|=1} \|\underline{Ax}\|.$$

It can be shown that matrix norms (2.1.5) to (2.1.7) are subordinate (i.e. they satisfy (2.1.10) or (2.1.9) to the corresponding vector norm (2.1.1) to (2.1.3)), whilst the Frobenius norm (2.1.8) is *not* subordinate to any vector norm (see Froberg (1974), Noble (1969)), Conte and de Boor (1972), Broyden (1975)).

Definition 2.1.8:

A vector is said to be normalised if it is multiplied by a scalar in order to produce the size of the components to numbers of value less than or equal to 1 without changing the direction of the vector.

Two common ways of normalising a vector $\underline{x}=[x_i]$, $i=1,2,\dots,n$, is by selecting a scalar β such that either:

$$(i) \beta = \left(\sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}}$$

$$\text{or } (ii) \beta = \max_i(x_i),$$

to obtain the normalized vector $\left[\frac{x_1}{\beta}, \frac{x_2}{\beta}, \dots, \frac{x_n}{\beta} \right]^T$. Notice that for (i) the relation $\underline{x}^T \underline{x} = 1$ holds.

Definition 2.1.9: (Permutation matrix)

A square matrix is called a *permutation matrix* if for any of its rows only one non-zero element is included (which is unity), for example

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

It can be shown that any permutation matrix, P (say), is orthogonal (i.e. $P^T = P^{-1}$). Also for any matrix A , the operations of pre-multiplication, i.e. PA and that of post-multiplication AP results in changing the order of rows and columns respectively.

Definition 2.1.10:

An n^{th} order matrix $A=[a_{ij}]$ is said to be:

- (i) *diagonally dominant* if $\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| < |a_{ii}|, i=1,2,\dots,n,$ *
- (ii) *strictly diagonally dominant* if $\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| < |a_{ii}|, i=1,2,\dots,n.$

Limit of a Sequence of Matrices and Its Convergence

A sequence of matrices $A^{(r)}, r=1,2,\dots$, of the same dimension converges to a ~~sole~~ limit, A (say), if the following necessary and sufficient condition is fulfilled,

$$\|A - A^{(r)}\| \rightarrow 0 \text{ as } r \rightarrow \infty,$$

or

$$\lim_{r \rightarrow \infty} \|A^{(r)}\| = A, \quad (2.1.11)$$

In fact, the result (2.1.11) does exist if Cauchy's theorem holds, i.e. for any $\epsilon > 0$ there must be an integer N such that

$$\|A^{(r+s)} - A^{(r)}\| < \epsilon \text{ for all } r > N \text{ and } s > 0. \quad (2.1.12)$$

Obviously (2.1.11) or (2.1.12) can be applied for vectors as well.

(see Demidovich and Maron (1976), Kolmogorov and Fomin (1970)).

Definition 2.1.11:

In general, if a sequence of matrices $\{A^{(s)}\}, s=1,2,\dots$ converges to a limit, then matrix $A (= A^{(1)})$ is said to be *convergent matrix*. Moreover, if $\lim_{s \rightarrow \infty} A^{(s)}$ is a zero matrix (null matrix) then A is said to be a *zero-convergent matrix* (Neumann and Plemmons (1978)).

Definition 2.1.12:

The convergence of the sequence of vectors $\{\underline{x}^{(s)}\}$ to a limit \underline{x}^* (say), is said to be of order P if

$$\lim_{s \rightarrow \infty} \frac{\|\underline{x}^{(s+1)} - \underline{x}^*\|}{\|\underline{x}^{(s)} - \underline{x}^*\|^p} = k, \text{ where } k \text{ is a non-negative constant.}$$

Thus, for $p=2$, we have *quadratic convergence*,

and for $p=1$ we have (i) *linear convergence* iff $0 < k < 1$,

(ii) *superlinear convergence* iff $k=0$.

Remark 2.1.1:

If a non-singular matrix is symmetric, antisymmetric, diagonal, upper (or unit upper) triangular, lower (or unit lower) triangular, Hermitian, positive definite, then so is its *inverse*, (Broyden (1975)), page 39).

2.2 DIRECT AND ITERATIVE METHODS FOR SOLVING LINEAR SYSTEMS OF EQUATIONS

The task of solving a linear system of equations which is usually expressed in matrix form, i.e.

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}, \quad (2.2.1)$$

or in abbreviated form,

$$\underline{Ax} = \underline{z}, \quad (2.2.2)$$

is still a major challenge in the solution of scientific problems. The derivation of the system (2.2.1) is basically from linear problems and non-linear problems as well which are usually broken down into a sequence of steps involving linear equations, and is termed sometimes a *linearization* process which forms the basis of many numerical methods (e.g., see Chapter 3, or Section 2.3). As Scarborough (1955) points out there is no single method which is best for any and all systems of equations that may arise. In other words a certain method may achieve quite a satisfactory solution for (2.2.1) if it is a *sparse* matrix (with few non-zero elements) as in problems which arise in large order differential equations but unsatisfactory if it has a *dense* matrix (with few zero elements) as in statistical problems where the dimension is small.

The available approaches for solving (2.2.1) usually lie in the following categories:

- (i) *Direct* methods (or *exact* methods)
- (ii) *Iterative* methods (or indirect methods)

Direct methods (e.g. Cramer's method, Gaussian elimination, the method of square root, etc....) are basically designed to achieve an *exact* solution for (2.2.1) after a fixed number of arithmetical steps. This is true theoretically, but unattainable in practice due to the limitation of computers (i.e. their mantissa has a limited number of digits) which eventually enables

the occurrence of rounding errors to appear in the calculation, for example the rational number $2/3$ has to be presented in a terminated form (e.g. 0.66666 for five significant digits). This is actually one of the main drawbacks of direct methods. The accumulation of rounding errors is well considered in these methods because of the alteration of the matrix A in (2.2.2) which may destroy the initial property of the matrix (i.e. sparseness) and ultimately have a considerable effect on the solution. Nevertheless, most of the computer routines for solving (2.2.2) involve direct methods since the total amount of computational labour can be determined in advance. For a given length of mantissa (i.e. number of digits) one may be able to predict the bounds of the rounding error and hence determine the range of reliability of the method. If A in (2.2.2) is dense, then the elimination methods are preferable (Jennings (1964)).

Iterative methods (such as Jacobi, Gauss-Seidel, Successive Over-relaxation method, etc...) are essentially based on generating a sequence of approximate solutions $\{\underline{x}^{(s)}\}$, $s=0,1,\dots$, for (2.2.2) and hope that this sequence approaches the solution $A^{-1}\underline{z}$ provided that the inverse exists. Generally speaking, iterative methods are considered to be *reliable* approaches provided that the existence of convergence is assured; this is because (i) there is no inherent inaccuracy, (ii) it is self-correcting, (iii) the method is applicable to systems of any number of unknowns (Scarborough (1955)) and (iv) the matrix remains unaltered. The criticism of these methods is mainly based upon: (i) there are certain systems of equations i.e. ill-conditioned one can not predict how many steps the iteration process will require to satisfy the required tolerance (ii) unless the sufficient and necessary condition is satisfied, convergence cannot be guaranteed. Thus, when using iterative methods it is advisable (i) to reduce the error each step of the iteration if it is possible or to determine an asymptotic factor of reducing the error to be less than one, and (ii) to provide an error bound to the

solution vector after a finite number of iterations (Lieberstein (1968)).

We may demonstrate some of the conventional methods of both types:

(A) Direct Methods: frequently are classified into three groups:

(Leiberstein (1968)).

- (1) Determinants: as in Cramer's method which involves unnecessarily extreme computation. For example, to solve (2.2.2) with order 10 requires some 70 million multiplications (Kunz (1957)), with order 50 the method requires 10^{64} operations. The number of operations involved in this method is of order $(n!)$ if the system is of order n (Froberg (1974)). What would be the case of a system consisting of several thousands of equations? No computer so far can provide enough storage and perform this large number of operations.
- (2) Inversion of Matrices: This strategy involves computing the inverse of the matrix A in (2.3.26) explicitly, which necessitates the solution of n systems of linear equations and hence the number of operations is proportional to (n^4) .
- (3) Systematic Eliminations: These methods are superior to the previous methods. The most widely used method is *Gaussian elimination* which involves a finite number of transformations (precisely one less than the size of the given system) that will eliminate all coefficients of the matrix below the diagonal and we end up with an upper triangular matrix. Thus, for the system (2.2.1) we have after $n-1$ transformations (Ralston (1965)):

$$\begin{bmatrix}
 a_{11} & a_{12} & \cdots & \cdots & \cdots & a_{1n} \\
 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & \cdots & a_{2n}^{(1)} \\
 & & a_{33}^{(2)} & a_{34}^{(2)} & \cdots & a_{3n}^{(2)} \\
 & & & \ddots & \ddots & \vdots \\
 & & & & a_{n-1,n-1}^{(n-2)} & a_{n-1,n}^{(n-2)} \\
 & & & & & a_{n,n}^{(n-1)}
 \end{bmatrix}
 \begin{bmatrix}
 x_1 \\
 x_2 \\
 x_3 \\
 \vdots \\
 x_{n-1} \\
 x_n
 \end{bmatrix}
 =
 \begin{bmatrix}
 z_1 \\
 z_2^{(1)} \\
 z_3^{(2)} \\
 \vdots \\
 z_n^{(n-1)}
 \end{bmatrix} \quad (2.2.3)$$

where

$$\left. \begin{aligned} a_{ij}^{(k)} &= a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)}, & k=1,2,\dots,n-1 \\ & & j=k+1,\dots,n \\ z_i^{(k)} &= z_i^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} z_k^{(k-1)}, & i=k+1,\dots,n \\ & & a_{ij}^{(0)} = a_{ij} \\ & & z_1^{(0)} = z_1 \end{aligned} \right\} \quad (2.2.4)$$

The solution for (2.2.3) is given by

$$x_n = z_n^{(n-1)} / a_{n,n}^{(n-1)}$$

$$x_i = \frac{1}{a_{ii}^{(i-1)}} \left[z_i^{(i-1)} - \sum_{j=i+1}^n a_{ij}^{(i-1)} x_j \right], \quad i=n-1, n-2, \dots, 1.$$

Alternatively the *Gauss-Jordan* process eliminates not only the lower off-diagonal elements but also the upper off-diagonal elements as well. Therefore, the final stage of the transformation produces a matrix of non-zero elements solely on the diagonal, and hence the solution is obtained straight-forwardly by dividing the components of the right hand side vector by the corresponding diagonal elements. In other words, there is no need for the *back substitution* stage as in the *Gaussian elimination*. Furthermore, Gaussian elimination can be shown to be superior to Gauss-Jordan since the number of operations are proportional to $\frac{n^3}{3}$ and $\frac{n^3}{2}$ respectively, and for large n the latter requires 50 percent more operations than the former (Ralston (1965)).

LU-Decomposition (Triangular Factorisation)

Let the $(n \times n)$ matrices M_1, M_2, \dots, M_k , $k=1, \dots, n-1$, be defined as follows (see Ralston (1965), Gault et al (1974)),

$$M_1 = \begin{bmatrix} 1 & & & \\ -m_{21} & 1 & & 0 \\ -m_{31} & 0 & 1 & \\ \vdots & \vdots & \vdots & \vdots \\ -m_{n1} & 0 & & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & & & \\ 0 & 1 & & 0 \\ & -m_{32} & 1 & \\ \vdots & \vdots & \vdots & \vdots \\ 0 & -m_{n,2} & & 1 \end{bmatrix},$$

$$M_k = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & -m_{k+1,k} & 1 \\ & & \vdots & \vdots \\ & & -m_{n,k} & & 1 \end{bmatrix}$$

kth column
↓

where for M_1 : $m_{s1} = a_{s1}/a_{11}$, $s=2,3,\dots,n$,

for M_2 : $m_{s2} = a_{s2}/a_{22}$, $s=3,4,\dots,n$,

...

for M_k : $m_{sk} = a_{sk}/a_{kk}$, $s=k+1,\dots,n$,

and the values a_{sk}, a_{kk} are obtained at the $(k-1)^{th}$ step of the transformation as illustrated in (2.2.4).

In fact, $m_{k+1,k}$'s are the multipliers of the k^{th} step of the transformation for the Gaussian elimination method. Thus, the triangular matrix form (2.2.3) is equivalent to

$$MA\underline{x} = M\underline{z}, \quad (2.2.5)$$

where $M = M_{n-1}M_{n-2}, \dots, M_1 \equiv L^{-1}, \quad (2.2.6)$

If we define U such that $U = MA, \quad (2.2.7)$

then (2.2.5) becomes $U\underline{x} = M\underline{z}. \quad (2.2.8)$

Since the inverse of a lower triangular matrix is another lower triangular matrix (see Remark 2.1.1), then we can write

$$M^{-1} = M_1^{-1}M_2^{-1}, \dots, M_{n-1}^{-1} \equiv L$$

and consequently (2.2.7) yields

$$LU = A, \quad (2.2.9)$$

and (2.2.8) gives

$$U\underline{x} = L^{-1}\underline{z},$$

or

$$LU\underline{x} = \underline{z}. \quad (2.2.10)$$

The form (2.2.5) is termed triangular or the 'LU' *decomposition*. Subsequently the solution of (2.2.2) by this algorithm follows from (2.2.10) via the introduction of an auxiliary vector, \underline{y} (say), such that the system (2.2.10) will be split into two systems, i.e.

$$L\underline{y} = \underline{z}, \quad (2.2.11a)$$

$$U\underline{x} = \underline{y}, \quad (2.2.11b)$$

where L is a *unit* lower triangular matrix and U is an upper triangular matrix. It turns out that the solution vector \underline{x} can be obtained from (2.2.11) through *forward* and *backward* substitutions (i.e. by (2.2.11a) and (2.2.11b), i.e.

$$y_1 = z_1$$

$$y_i = z_i - \sum_{k=1}^{i-1} y_k l_{ik}, \quad i=2, \dots, n,$$

and

$$x_i = (y_i - \sum_{j=i+1}^n u_{ij} x_j) / u_{ii}, \quad i=n(-1)1$$

provided $u_{ii} \neq 0$ (i.e. U is non-singular).

For sparse matrices with special form (tridiagonal, pentadaigonal, etc.) the factorisation (2.2.9) may be achieved by equating both sides so that a 'general' recurrence relation can be formulated mainly for programming purposes. Further, the intermediate vector \underline{y} in (2.2.11) does not need to be computed explicitly, for example the solution of the tridiagonal system

$$\begin{bmatrix} d_1 & a_1 & & & \\ c_2 & d_2 & a_2 & & \\ & \ddots & \ddots & \ddots & \\ 0 & & & a_{n-1} & d_n \\ & & c_n & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

can be expressed by the following recurrence relation (Varga (1962)):

$$x_n = g_n, \quad x_i = g_i - \beta_i x_{i+1}, \quad i=1,2,\dots,n-1$$

where

$$\beta_1 = \frac{a_1}{d_1}, \quad \beta_i = \frac{a_i^{i+1}}{d_i - c_i \beta_{i-1}}, \quad i=2,3,\dots,n-1 \quad (2.2.12)$$

and

$$g_1 = \frac{b_1}{d_1}, \quad g_i = \frac{b_i - c_i g_{i-1}}{d_i - c_i \beta_{i-1}}, \quad i=2,3,\dots,n.$$

In fact (2.2.12) is an equivalent (nested) form of the Gaussian elimination process.

However, the LU-decomposition may be applied when the following theorem is valid.

Theorem 2.2.1:

A non-singular matrix may be decomposed into the product LU (where L and U are lower and upper triangular matrices) if and only if every leading principal submatrix of A is non-singular.

Corollary 2.2.1:

If L is unit lower triangular then the decomposition is unique.

Proof:

Both Theorem 2.2.1 and Corollary 2.1.1 are given in Broyden (1975), see also Faddeeva (1959).

Corollary 2.2.2:

If U is a unit upper triangular matrix then the decomposition is unique.

Proof: Similar to Corollary 2.2.1.

The LU decomposition where Corollary 2.2.1 is valid is often called *Doolittle's* method, whilst if Corollary 2.2.2 is valid, it is called *Crout's* method (Goult et al (1974)).

If the matrix A in (2.2.2) is symmetric, then the decomposition (2.2.9) may have a modified variant which is an economised procedure as

far as the computational work is concerned is called the Choleski's method (or the square-root method) which can be outlined as follows:

Since A is a symmetric matrix, then U can be replaced by L^T and hence we have

$$LL^T = A. \quad (2.2.13)$$

So, if $L=[l_{ij}]$, where $l_{ij}=0$ for $i < j$, then

$$\left. \begin{aligned} l_{ii} &= [a_{jj} - \sum_{k=1}^{i-1} l_{jk}^2]^{\frac{1}{2}}, \text{ for } i=j \\ l_{ij} &= \frac{1}{l_{jj}} [a_{ji} - \sum_{k=1}^{j-1} l_{ik} l_{jk}], \text{ } j < i \leq n, \\ &\text{provided } l_{jj} \neq 0. \end{aligned} \right\} j=1,2,\dots,n \quad (2.2.14)$$

It is worthwhile to point out that if the positive square roots in (2.2.14) are chosen only, then (2.2.13) is a unique factorisation provided that the matrix A is *real symmetric* and *positive definite*. In actual fact, this latter property may place the Choleski scheme superior to other variants of the elimination methods (such as those mentioned above), in particular if double-precision arithmetic is used so that the square roots of (2.2.14) are evaluated as accurately as possible. Although, the calculation of the square roots remains one of the main disadvantages of the Choleski method, but this may be alleviated by the decomposition $LDL^T=A$, where D is a diagonal matrix (Broyden (1975)).

Practical Refinement of Gaussian Elimination Process

If any of the diagonal elements of the matrix in (2.2.1) becomes zero during the elimination process, then the final upper triangular form will be unattainable, and hence the process will break down. Nevertheless, to overcome such difficulty and to ensure the continuation of the elimination process we may apply one of two basic well known *pivoting* schemes.

Definition 2.2.1:

Any of the diagonal elements in (2.1.1), i.e. $a_{kk}^{(k-1)}$, $k=1, \dots, n$ (where $a_{11}^{(0)} = a_{11}$) is termed the k^{th} pivot. If it is zero, then it is called *zero pivot*.

The two strategies of pivoting are mainly concerned with avoiding a zero pivot which may arise during the elimination process.

(1) Partial Pivoting

This strategy is based on selecting an element with largest value in modulus from the column of the reduced matrix as a pivotal element. Eventually, the appropriate rows of the augmented matrix $(A^{(k)}, Z^{(k)})$ must be interchanged.

The following example shows that the partial pivoting scheme can be inadequate, i.e., (Williams (1973))

$$4x + 3y = 10$$

$$3x - 2y = 12$$

Any row of the above equations can be multiplied by an arbitrary constant and hence change the pivotal row. This can be overcome by normalizing the rows and thereby making them comparable in one of the two following ways: (see Def.2.1.8):

- (i) divide each row by the largest element in modulus,
- (ii) divide each row by the Euclidian norm of the row.

(2) Full (or complete) Pivoting

The pivotal element is chosen to be the element of largest magnitude amongst the elements of the reduced matrix, regardless of the position of the element in this matrix.

Both ways of pivoting can be easily illustrated in Fig.2.2.1 assuming the system (2.2.2) is of order 5.

$$\begin{bmatrix}
 X & X & X & X & X & | & X \\
 0 & X & X & X & X & | & X \\
 0 & 0 & \boxed{1} & 2 & 3 & | & \boxed{X} \\
 0 & 0 & \boxed{X} & \boxed{X} & \boxed{X} & | & \boxed{X} \\
 0 & 0 & \boxed{X} & \boxed{X} & \boxed{X} & | & \boxed{X} \\
 0 & 0 & \boxed{X} & \boxed{X} & \boxed{X} & | & \boxed{X}
 \end{bmatrix} = (A^{(2)}, \underline{Z}^{(2)})$$

FIGURE 2.2.1: The two strategies (X and \boxed{X} denote non-zero elements)

- (i) for *partial pivoting*, any of the elements in the box can be taken as the pivot. If '7' is the largest magnitude, then the 3rd and 5th rows of $(A^{(2)}, \underline{Z}^{(2)})$ have to be interchanged.
- (ii) for *full pivoting*, any of the 9 numbered elements can be taken as the pivot. If '5' is the element of largest magnitude, then the interchanging involves (1) the 3rd and 4th rows of $(A^{(2)}, \underline{Z}^{(2)})$ followed by (2) the 3rd and 4th columns and the corresponding unknowns as well.

The full pivoting is considered to be a satisfactory strategy but in practice it is time-consuming for execution. In addition, since the columns are included in the interchanging process, then it may be difficult to preserve the triangular form of the matrix to the final step. Also, searching for the pivot element may take a long time, especially for large systems of equations. Thus, the partial pivoting is, generally, preferred in practice and for most problems including the iterative improvement (or residual correction) procedure (see Gault et al (1974), Broyden (1975)).

The pivoting approach can also be applied for the LU decomposition. However it can be shown to be unnecessary for positive definite matrices.

(B) Iterative Methods

These methods may be considered as an alternative to direct methods for solving linear systems of special properties, notably when the matrix is sparse (elimination methods may fill-in the zero elements with non-zero values and/or of large dimension).

Iterative methods for solving the linear system (2.2.2) are, generally, based on generating a sequence of approximate solution vectors $\{\underline{x}^{(s)}\}$, $s=0,1,2,\dots$, such that the approximate solution $\underline{x}^{(s+1)}$ is a linear function of $\underline{x}^{(s)}$. If this sequence does converge, then the iteration process can be interrupted whenever the desired accuracy in the solution is attained or an optimal number of significant figures is reached depending on the word length of the computer. Furthermore, in contrast to direct methods, iterative methods do not suffer from the inherent inaccuracy in the calculation since they are always self-correcting procedures where the solution at the s^{th} step will not affect the solution at the next step and can be regarded as an initial solution to the $(s+1)^{\text{th}}$ iteration. On the other hand, if we are seeking a solution of N -digits accuracy, and the generated sequence of solutions is carried out retaining M -digits of accuracy ($M > N$) then for the computation to be worthwhile the loss of accuracy should not exceed $(M-N)$ digits.

Three well-known iterative procedures are presented to solve (2.2.1):

(i) Jacobi (or Simultaneous Displacements) Method

In this method, the sequence of approximate solutions can be generated successively from the formula,

$$x_i^{(s+1)} = \frac{1}{a_{ii}} \left(z_i - \sum_{\substack{j=1 \\ i \neq j}}^n a_{ij} x_j^{(s)} \right), \quad i=1,2,\dots,n \quad (2.2.15)$$

(ii) Gauss-Seidel (or Successive Displacements) Method

The iteration process described by this method has the form,

$$x_i^{(s+1)} = \frac{1}{a_{ii}} \left(z_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(s+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(s)} \right), i=1,2,\dots,n \quad (2.2.16)$$

(iii) Successive Over-Relaxation, S.O.R. (Extrapolated Gauss-Seidel Method)

This method is, basically, to accelerate the convergence of (2.2.16) by inserting an over-relaxation factor ω whose optimum value lies between 1 and 2 (sometimes the method is formed under-relaxation for $0 < \omega < 1$).

The computation form of this method (which was suggested by D.M. Young (1954)) is

$$x_i^{(s+1)} = (1-\omega)x_i^{(s)} + \frac{\omega}{a_{ii}} \left(z_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(s+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(s)} \right), i=1,2,\dots,n \quad (2.2.17)$$

However, the general matrix form for solving (2.2.2) iteratively can be given by

$$R\mathbf{x}^{(s+1)} = \mathbf{z} + T\mathbf{x}^{(s)}, \quad (2.2.18)$$

where the matrix A of (2.2.2) has been split into matrices R and T such that $A=R-T$, and R is non-singular matrix. Subsequently, if A is split into three component matrices, L,D and U, i.e. $A=-L+D-U$, where D is diagonal, L and U are lower and upper triangular matrices respectively, then on substitution of R and T in (2.2.18) as follows:

$$(a) \quad R=D, \quad T=L+U,$$

$$(b) \quad R=D-L, \quad T=U,$$

$$\text{and } (c) \quad R=\omega D^{-1}, \quad T=U+(\omega^{-1}-1)D,$$

we will obtain the equivalent matrix form of the above mentioned iterative schemes (i),(ii) and (iii) respectively, i.e.,

$$\mathbf{x}^{(s+1)} = D^{-1}(L+U)\mathbf{x}^{(s)} + D^{-1}\mathbf{z} \quad (\text{Simultaneous Displacement Method}) \quad (2.2.19)$$

$$\mathbf{x}^{(s+1)} = D^{-1}L\mathbf{x}^{(s+1)} + D^{-1}U\mathbf{x}^{(s)} + D^{-1}\mathbf{z} \quad (\text{Successive Displacement Method}) \quad (2.2.20)$$

$$\mathbf{x}^{(s+1)} = \omega D^{-1}L\mathbf{x}^{(s+1)} + [\omega D^{-1}U + (1-\omega)I]\mathbf{x}^{(s)} + \omega D^{-1}\mathbf{z} \quad (\text{S.O.R. Method}) \quad (2.2.21)$$

The three schemes can be presented by

$$\underline{x}^{(s+1)} = M\underline{x}^{(s)} + \underline{d}, \quad (2.2.22)$$

where the *iteration (or correction) matrix* $M=R^{-1}T$ and $\underline{d}=R^{-1}\underline{z}$ (where R, T and \underline{z} are defined in (2.2.18)).

In fact, the iterative scheme (2.2.22) represents the general form of a *stationary iterative process*, where the matrix M remains unchanged throughout the iteration operation, (if the relaxation factor ω in (2.2.21) depends upon s , then (2.2.21) becomes a *non-stationary iterative process*).

The iteration process (2.2.22) converges to a fixed point \underline{x} , ($\underline{x}=A^{-1}\underline{z}$), the solution of (2.2.2) for any initial solution $\underline{x}^{(0)}$ if the matrix M is *zero-convergent*. More precisely, since any matrix is zero-convergent if and only if its spectral radius is less than unity i.e. $\rho(M) < 1$ (Neumann and Plemmons (1978)), then a sufficient and necessary condition of convergence for (2.2.22) can be given by the

Theorem 2.2.2:

A necessary and sufficient condition for the iteration process (2.2.22) to converge for any initial vector $\underline{x}^{(0)}$ is that all the eigenvalues of M should be less than 1 in modulus.

Proof: (see Goult et al (1974))

Whilst a sufficient condition for convergence of (2.2.22) is merely that $\|M\| < 1$, since $|\lambda| \leq \|M\|$ (see Section 2.4), where λ refers to the largest eigenvalue of the matrix M . This means that it may happen in some cases $\|M\| > 1$ but $|\lambda| < 1$ which guarantees the convergence of the iteration process according to the above stated Theorem 2.2.2. On the otherhand as confirmed by Theorem 2.2.2 the convergence of (2.2.22) is totally *independent* of the choice of the initial vector, $\underline{x}^{(0)}$ as long as the matrix A in (2.2.2) is non-singular, whilst it is *dependent* on $\underline{x}^{(0)}$ if A is singular (Meyer and Plemmons (1977)).

The asymptotic rate of convergence of (2.2.22) is given by the value $-\log|\lambda|$ (Froberg (1974)) or the average rate of convergence for s iterations may be given by $\frac{-\ln \|M^{(s)}\|}{s}$, (Varga (1962)). So, for a given non-singular linear system we can determine the rate of convergence of the iterative algorithms (2.2.19), (2.2.20) and (2.2.21). Generally, the Gauss-Seidel scheme yields a better rate of convergence than the Jacobi. Moreover, sometimes it happens that the former might converge and the latter diverge, and vice versa, (illustrated in Fox (1964), Faddeeva (1959)). For a linear system which possesses a *diagonal dominant* matrix both schemes may converge since the *sufficient condition* (as given above) is fulfilled. Furthermore, the superiority of the Gauss-Seidel method over the Jacobi method is given by the following theorem:

Theorem 2.2.3:

If A in (2.3.26) is symmetric positive-definite, then the Gauss-Seidel method always converges since all the eigenvalues of the iteration matrix (i.e. $M=(D-L)^{-1}U$) are less than 1 in modulus.

Proof: (see Lieberstein (1968), Fox (1964))

In the former reference (see page 62) there is given a counter-example which verifies the invalidity of Theorem 2.2.3 for the Jacobi scheme, i.e. although the matrix A is symmetric and positive definite the iteration matrix $D^{-1}(L+U)$ may have eigenvalue(s) greater than 1 in modulus.

The convergence of the S.O.R. method depends upon the choice of over-relaxation factor ω so as to ensure the eigenvalues of the iteration matrix M be minimised to as small as possible and <1 in modulus. Unfortunately, there is no *general* method available to locate the optimum value of ω to satisfy this requirement. This is discussed in Varga (1962), Gault et al (1974), Froberg (1975) and Smith (1978), etc.

In general, the amount of computational work involved in any iterative method cannot always be easily determined in advance. However, it can be shown that an iterative process requires approximately $O(n^2)$ operations (multiplications and additions) per step for an $(n \times n)$ full/dense matrix. Thus, an iterative method would be superior to the conventional elimination methods if $s \leq \frac{1}{3}n$ (where s refers to the number of steps when the iterative process is interrupted). Obviously, for large sparse linear systems, the number of operations may be considerably less than n^2 . Conrad and Wallach (1979) proved that the number of operations can be reduced considerably (25% or 50% for some iterative algorithms) by a so-called *alternating technique*. This involves the combination of any two explicit iterative procedures, such as (2.2.19) to (2.2.21) in an alternating fashion, i.e. each step of (2.2.18) being replaced by two 'half' iterations of the form,

$$\begin{aligned} R_1 \underline{x}^{(s+\frac{1}{2})} &= z + T_1 \underline{x}^{(s)} \\ R_2 \underline{x}^{(s+1)} &= z_1 + T_2 \underline{x}^{(s+\frac{1}{2})}, \quad s=0,1,2,\dots \end{aligned}$$

where

$$R_1 - T_1 = R_2 - T_2 = A.$$

Finally, we outline the *residual correction* procedure which aims to improve the unacceptable solution of (2.2.2). The *residual* vector, \underline{r} (say) which is $\underline{0}$ for the exact solution can be shown to satisfy the following iteration process,

$$\underline{r}^{(i)} = \underline{b} - A \underline{x}^{(i)}, \quad i=0,1,2,\dots \quad (2.2.23)$$

where $\underline{x}^{(0)}$ is the initial solution vector, and $\underline{r}^{(i)}$ is the residual vector at the i^{th} iteration.

If the solution $\underline{x}^{(i)}$, $i \geq 0$, is not sufficiently accurate then one should proceed to compute the residual vector in *double precision* computation form (2.2.23), and consequently solve the system (using *single precision* computation),

$$A \underline{\eta}^{(i)} = \underline{r}^{(i)}, \quad (2.2.24)$$

for the correction vector $\underline{\eta}^{(i)}$ which can be added to $\underline{x}^{(i)}$ to produce the 'improved' solution $(\underline{x}^{(i)} + \underline{\eta}^{(i)})$. Further, if the factorization LU for A is computed initially and retained, then the work to carry out the iteration (2.2.23) is considerably reduced for $i=1,2,\dots$ via the process of solving $L\underline{y}^{(i)} = \underline{r}^{(i)}$ and $U\underline{u}^{(i)} = \underline{y}^{(i)}$. The iterative process can be terminated at a stage where no further improvement in the solution is obtained. Meanwhile, it is important to point out that the residual $\underline{r}^{(0)} = \underline{b} - A\underline{x}^{(0)}$ may have a 'misleading' concept, i.e. even if it has small components it does not necessarily indicate that the solution $\underline{x}^{(0)}$ is acceptable (Fox and Mayers(1977)) as for instance in ill-conditioned equations or cases where the exact solution \underline{x} is small. Thus, $\underline{r}^{(0)}$ and the remainder of the residuals, $\underline{r}^{(1)}, \underline{r}^{(2)}, \dots, \underline{r}^{(s)}$ must be calculated with double precision computation (Goult and et al, (1974)). Thus, the residual correction scheme is a reliable procedure which reduces the error in the approximate solution and in particular whenever $\underline{x}^{(0)}$ is reasonably close to $A^{-1}\underline{z}$.

2.3 CONTRACT MAPPING THEOREM, NEWTON'S METHOD

Let there be given a non-linear system of n (≥ 1) equations, i.e.

$$\left. \begin{aligned} x_1 &= \phi_1(x_1, x_2, \dots, x_n), \\ x_2 &= \phi_2(x_1, x_2, \dots, x_n), \\ &\vdots \\ x_n &= \phi_n(x_1, x_2, \dots, x_n), \end{aligned} \right\} \quad (2.3.1)$$

where the functions $\phi_1, \phi_2, \dots, \phi_n$ are defined and continuous in a given domain G , where $G \subset \mathbb{R}^n$ (the real n -dimensional space). If the values $x_1, x_2, \dots, x_n \in G$, then the function ϕ_i , $i=1, 2, \dots, n$ form the mapping of G onto itself. Moreover, rewriting (2.3.1) in the compact form,

$$\underline{x} = \underline{\phi}(\underline{x}), \quad (2.3.2)$$

where $\underline{x} = [x_1, x_2, \dots, x_n]^T$, $\underline{\phi} = [\phi_1, \phi_2, \dots, \phi_n]^T$,

we may establish the following definition.

Definition 2.3.1:

The mapping $\underline{\phi}$ in (2.3.2) is termed a *contraction mapping* in the domain G if there exists a *proper fraction* L such that for any two vectors $\underline{x}_1, \underline{x}_2 \in G$ their images $\underline{\phi}(\underline{x}_1)$ and $\underline{\phi}(\underline{x}_2)$ fulfil the following condition

$$||\underline{\phi}(\underline{x}_1) - \underline{\phi}(\underline{x}_2)|| \leq L ||\underline{x}_1 - \underline{x}_2||, \quad 0 \leq L < 1, \quad (2.3.3)$$

and L is independent of \underline{x}_1 and \underline{x}_2 and is commonly termed a Lipschitz constant. The inequality (2.3.3) is known as the *Lipschitz (contraction) condition*. It leads to an important theorem which is stated below.

Theorem 2.3.1:

Given a closed domain G , a constant $L < 1$ and a function $\underline{\phi}$ to be an contraction mapping in G satisfying the Lipschitz condition (2.3.3), then the following statements hold true:

- (i) for any irrespective choice of the initial solution $\underline{x}^{(0)} \in G$, the sequence of successive solutions $\{\underline{x}^{(r)}\}$, $r \geq 0$ and $\underline{x}^{(r)} \in G$, will converge to a limit, \underline{x}^* (say). And $\underline{x}^* \in G$ is the root of (2.3.2) i.e., $\underline{x}^* = \underline{\phi}(\underline{x}^*)$,

(ii) the non-linear vector equation (2.3.2) has a unique solution, i.e.

\underline{x}^* is a sole one,

(iii) the following relationship must hold as well,

$$||\underline{x}^* - \underline{x}^{(r)}|| \leq \frac{L^r}{1-L} ||\underline{x}^{(1)} - \underline{x}^{(0)}|| \quad (2.3.4)$$

Proof:

Let $s > r$ and writing

$$||\underline{x}^{(s)} - \underline{x}^{(r)}|| = ||(\underline{x}^{(r+1)} - \underline{x}^{(r)}) + (\underline{x}^{(r+2)} - \underline{x}^{(r+1)}) + \dots + (\underline{x}^{(s)} - \underline{x}^{(s-1)})||, \quad (2.3.5)$$

we obtain the following by applying the triangle inequality given earlier in this chapter,

$$||\underline{x}^{(s)} - \underline{x}^{(r)}|| \leq ||\underline{x}^{(r+1)} - \underline{x}^{(r)}|| + ||\underline{x}^{(r+2)} - \underline{x}^{(r+1)}|| + \dots + ||\underline{x}^{(s)} - \underline{x}^{(s-1)}||. \quad (2.3.6)$$

Now, by virtue of Lipschitz condition (2.3.3) we have

$$\begin{aligned} ||\underline{x}^{(m+1)} - \underline{x}^{(m)}|| &= ||\phi(\underline{x}^{(m)}) - \phi(\underline{x}^{(m-1)})|| \\ &\leq L ||\underline{x}^{(m)} - \underline{x}^{(m-1)}|| \\ &\leq L^2 ||\underline{x}^{(m-1)} - \underline{x}^{(m-2)}|| \\ &\vdots \\ &\leq L^m ||\underline{x}^{(1)} - \underline{x}^{(0)}|| \end{aligned} \quad (2.3.7)$$

Applying the result (2.3.7) on (2.3.6), we obtain

$$\begin{aligned} ||\underline{x}^{(s)} - \underline{x}^{(r)}|| &\leq (L^r + L^{r+1} + \dots + L^{s-1}) ||\underline{x}^{(1)} - \underline{x}^{(0)}|| \\ &= \frac{L^r - L^s}{1-L} ||\underline{x}^{(1)} - \underline{x}^{(0)}|| \quad (\text{by using the sum formula} \\ &\quad \text{of a geometric progression}). \\ &\leq \frac{L^r}{1-L} ||\underline{x}^{(1)} - \underline{x}^{(0)}|| \end{aligned} \quad (2.3.8)$$

Since $L < 1$, then $L^r \rightarrow 0$ as $r \rightarrow \infty$. Thus, for any ε the Cauchy inequality (2.1.12) can be applied on (2.3.8) and hence the sequence $\{\underline{x}^{(r)}\}$ has a limit (cf. (2.1.11)), i.e.,

$\underline{x}^* = \lim_{r \rightarrow \infty} \underline{x}^{(r)}$, and $\underline{x}^* \in G$ which completes the proof of part (i) of the theorem.

To prove part (ii) we proceed as follows.

Assume that $\underline{x}^{**} \in G$ is another solution of (2.3.2) different from \underline{x}^* , then we have,

$$\begin{aligned} ||\underline{x}^* - \underline{x}^{**}|| &= ||\underline{\phi}(\underline{x}^*) - \underline{\phi}(\underline{x}^{**})|| \\ &\leq L ||\underline{x}^* - \underline{x}^{**}|| \end{aligned}$$

or $||\underline{x}^* - \underline{x}^{**}|| (1-L) \leq 0$, (2.3.9)

since $(1-L) < 0$, then (2.3.9) cannot hold unless $\underline{x}^* = \underline{x}^{**}$.

By letting $s \rightarrow \infty$ in (2.3.8) we have $\underline{x}^* = \lim_{s \rightarrow \infty} \underline{x}^{(s)}$ and hence point (iii) of the theorem is complete.

(See Ortega and Rheinboldt (1970), Demidovich and Maron (1976), Henrici (1964)).

However, according to the Theorem 2.3.1, the Picard iteration process for (2.3.2) i.e.

$$\underline{x}^{(k+1)} = \underline{\phi}(\underline{x}^{(k)}), \quad k=0,1,\dots \quad (2.3.10)$$

converges for a unique fixed point $\underline{x}^* \in G \subset \mathbb{R}^n$ for any $\underline{x}^{(0)} \in G$. Furthermore, if $G = \mathbb{R}^n$, then we have global convergence for (2.3.10). Meanwhile, Theorem 2.3.1 in this case, may be termed as the *global convergence theorem* (Ortega and Rheinboldt (1970), page 385).

We may introduce another theorem which is associated with the preceeding theorem, concerning the convergency of the non-linear equation (2.3.10) (see Dahlquist and Bjork (1969), Demidovich and Maron (1976), Szidarovszky and Yakowitz (1978)):

Theorem 2.3.2:

Let the vector function $\underline{\phi}(\underline{x})$ be continuous together with its derivative $\underline{\phi}'(\underline{x})$ in a bounded convex closed domain G and satisfies

$$||\underline{\phi}'(\underline{x})||_1 \leq \mu < 1, \quad (2.3.11)$$

where μ is a constant and

$$||\underline{\phi}'(\underline{x})||_1 = \max_{\underline{x} \in G} \left(\max_j \sum_{i=1}^n \left| \frac{\partial \phi_i(\underline{x})}{\partial x_j} \right| \right). \quad (2.3.12)$$

If $\underline{x}^{(0)} \in G$ and all successive approximations $\underline{x}^{(1)}, \underline{x}^{(2)}, \dots$ also lie in G , then the iteration process (2.3.10) converges to a unique solution of the equation (2.3.2).

(N.B. this theorem is also valid for $||.||_\infty$ or $||.||_F$ in addition to $||.||_1$ as in (2.3.11) and (2.3.12), but not necessarily all of them at the same time).

Corollary 2.3.1:

The process of the Picard iteration (2.3.10) converges to the unique solution of equation (2.3.2), if the inequalities

$$\sum_{i=1}^n \left| \frac{\partial \phi_i(\underline{x})}{\partial x_j} \right| \leq \mu_j < 1, \quad j=1,2,\dots,n \quad (2.3.13)$$

hold true for $\underline{x} \in G$.

Newton's (or Newton-Raphson's) Method in n-Dimensions

We consider a non-linear system of equations,

$$\left. \begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \right\} \quad (2.3.14)$$

or compactly, (2.3.14) can be written in conventional vector form given

$$\text{by} \quad \underline{F}(\underline{x}) = \underline{0}, \quad (2.3.15)$$

$$\text{where} \quad \underline{F} = [f_1, f_2, \dots, f_n]^T,$$

$$\underline{x} = [x_1, x_2, \dots, x_n]^T,$$

and $\underline{0}$ is the null vector of the n-tuple.

Suppose that (2.3.15) has the exact solution $\underline{\alpha}$. By solving (2.3.15) iteratively (using the preceding iterative procedure) we may obtain an approximate solution $\underline{x}^{(s)}$ after s iterations, thus eventually we may write

$$\underline{x}^{(s)} = \underline{\alpha} + \underline{\epsilon}^{(s)}, \quad (2.3.16)$$

$$\text{where} \quad \underline{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_n]^T \quad \text{and} \quad \underline{\epsilon} = [\epsilon_1, \epsilon_2, \dots, \epsilon_n]^T,$$

which represents the error vector of the root.

Since $\underline{\alpha}$ is the exact solution, then it is trivial to write

$$\underline{F}(\underline{\alpha}) = \underline{0},$$

or from (2.3.16), we have

$$\underline{F}(\underline{x}^{(s)} - \underline{\epsilon}^{(s)}) = \underline{0} . \quad (2.3.17)$$

By Taylor's expansion, (2.3.17) yields the following result

$$\underline{0} = \underline{F}(\underline{x}^{(s)} - \underline{\epsilon}^{(s)}) \cong (\underline{x}^{(s)}) - \left[\frac{\partial \underline{F}}{\partial x_1}(\underline{x}^{(s)}), \frac{\partial \underline{F}}{\partial x_2}(\underline{x}^{(s)}), \dots, \frac{\partial \underline{F}}{\partial x_n}(\underline{x}^{(s)}) \right] \underline{\epsilon} + O(\epsilon_1, \dots, \epsilon_n) . \quad (2.3.18)$$

where $O(\epsilon_1, \dots, \epsilon_n)$ represents the high order terms of the error values $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ (> order 1). By supressing this term in (2.3.18) we obtain,

$$\underline{F}(\underline{x}^{(s)}) - \underline{J}(\underline{x}^{(s)}) \underline{\epsilon}^{(s)} = \underline{0} , \quad (2.3.19)$$

where $\underline{J}(\underline{x})$ is the Jacobian matrix which involves the derivatives of

f_1, f_2, \dots, f_n with respect to the independent variables x_1, x_2, \dots, x_n , i.e.,

$$\underline{J}(\underline{x}) = \underline{F}'(\underline{x}) \equiv \left[\frac{\partial f_i}{\partial x_j}(\underline{x}) \right] , \quad i, j=1, 2, \dots, n.$$

Assume $\underline{J}(\underline{x})$ is a non-singular matrix, thus we have from (2.3.19),

$$\underline{\epsilon}^{(s)} = [\underline{J}(\underline{x}^{(s)})]^{-1} \underline{F}(\underline{x}^{(s)}) . \quad (2.3.20)$$

Taking $\epsilon_i^{(s)} = -(x_i^{(s+1)} - x_i^{(s)})$, $i=1, 2, \dots, n$,

and substituting (2.3.20) we obtain the so-called *generalized Newton method*, i.e.

$$\underline{x}^{(s+1)} = \underline{x}^{(s)} - [\underline{J}(\underline{x}^{(s)})]^{-1} \underline{F}(\underline{x}^{(s)}), \quad s=0, 1, \dots \quad (2.3.21)$$

where $\underline{x}^{(0)}$ refers to the initial solution which is often recommended to be taken as close as possible to the desired exact solution.

It is known that (2.3.21) is impractical for implementation purposes, therefore it is usually converted to the equivalent form, i.e.

$$\underline{J}(\underline{x}^{(s)}) \Delta \underline{x}^{(s)} = -\underline{F}(\underline{x}^{(s)}) , \quad (2.3.22)$$

which can be solved for the *correction* $\Delta \underline{x}^{(s)}$ and added to $\underline{x}^{(s)}$ to produce the new approximate $\underline{x}^{(s+1)}$.

The modified form of the Newton's process is to approximate $\underline{J}(\underline{x}^{(s)})$ by $\underline{J}(\underline{x}^{(0)})$, then (2.3.21) becomes

$$\underline{x}^{(s+1)} = \underline{x}^{(s)} - [\underline{J}(\underline{x}^{(0)})]^{-1} \underline{F}(\underline{x}^{(s)}), \quad (2.3.23)$$

which sometimes is named as the *simplified Newton method*.

A geometrical comparison between the generalized and simplified forms for a single variable is given in Fig.2.3.1.

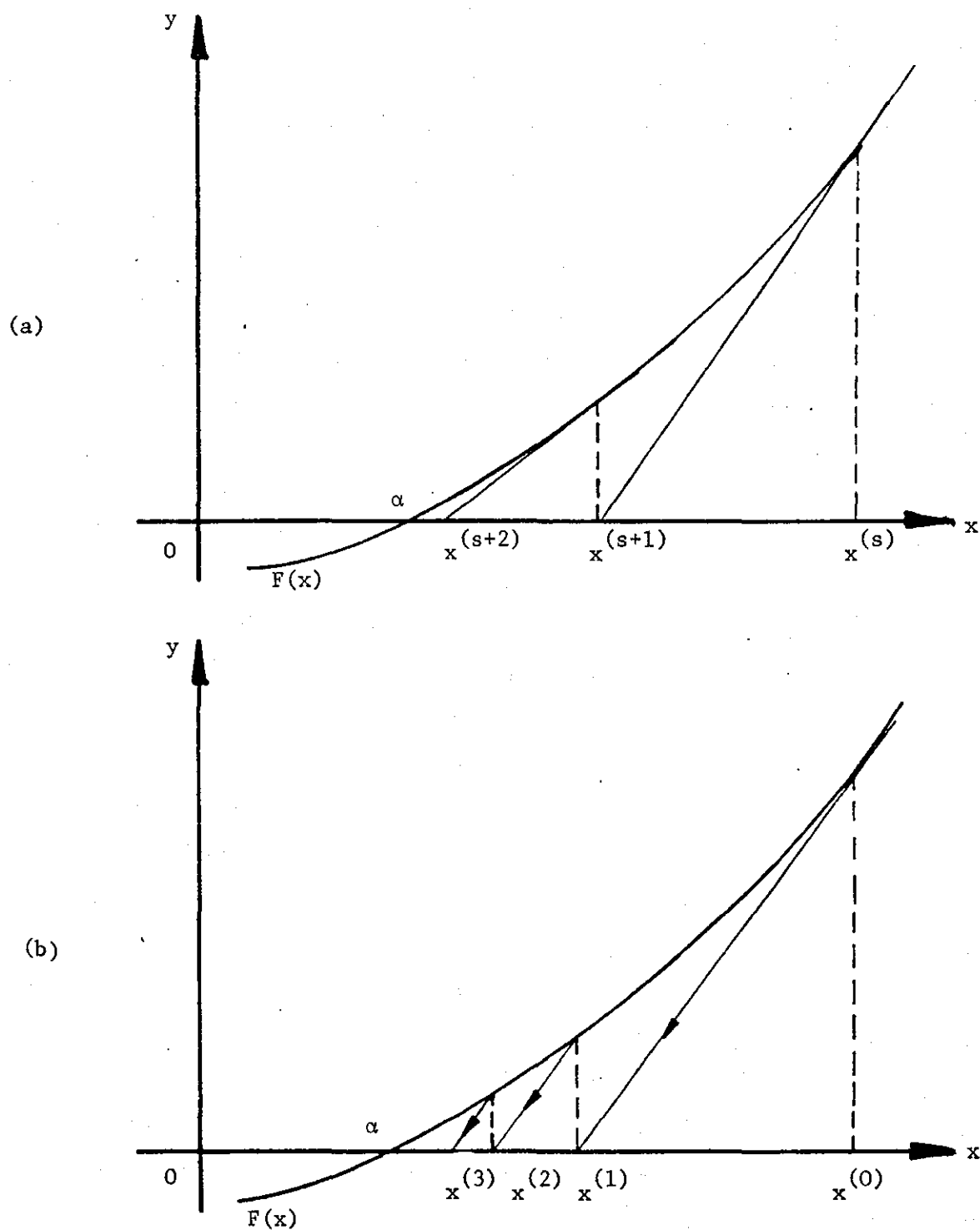


FIGURE 2.3.1: (a) Generalized Newton method (method of tangents)

(b) Modified simplified Newton method

We may derive from (2.3.21) that each step of the iteration process requires to evaluate the following:

- (i) the n components of $\underline{F}(\underline{x})$, $f_k(x_1, \dots, x_n)$, $k=1, 2, \dots, n$,
- (ii) n^2 elements of the Jacobian matrix, i.e. $\frac{\partial f_i}{\partial x_j}(x_1, \dots, x_n)$, $i, j=1, 2, \dots, n$.
- (iii) the solution of the linear system (2.3.22) by a suitable method (see previous section).

One of the procedures to economise on the amount of work is to avoid computing the Jacobian at every step and instead we either (1) use the modified Newton process (2.3.23), or (2) the Jacobian is evaluated once after several steps. Both cases however may depend upon the initial guess of the solution vector.

Generally speaking, Newton's method is still an attractive method from the theoretical viewpoint, this could be mainly due to its *quadratic convergency property*, where the error vectors in two successive steps of the iteration are associated by the relation

$$\|\underline{\varepsilon}^{(s+1)}\| \leq K \|\underline{\varepsilon}^{(s)}\|^2, \quad K \text{ is constant} \quad (2.3.24)$$

where $\underline{\varepsilon}^{(j)} = \underline{x}^{(j)} - \underline{\alpha}$ and $\underline{\alpha}$ is the exact solution.

Relation (2.3.24) is judged to be valid as long as the initial vector solution is sufficiently close to the exact solution.

Convergence of the Newton process and its sufficiency conditions have been studied and formulated by Kantorovich (see Henrici (1962), Demidovich and Maron (1976), Brown (1962)). Also it has been discussed by Ortega and Rheinboldt (1970) and Ostrowski (1966).

In practice Newton's method, unfortunately, may not be considered as an efficient and attractive computational procedure, in particular for large systems of non-linear equations where the order may exceed several thousands (as in non-linear partial differential equations). The main concern in this respect is the loss of accuracy during the solution of

the linear system (step (iii), page 38) by direct methods and loss of both practical and theoretical efficiency in solving the linear system by iteration (Lieberstein (1968)). In addition, the amount of computational effort required by step (i) and step (ii) (page 38) is too expensive and may be too difficult (unless the desired derivatives are in a simple form).

However, due to extensive investigations which have been reported in this respect so far some modifications of the Newton's process have been proposed (see Ortega and Rheinboldt (1970)). Three variants will now be introduced.

(1) Discretized Newton Iteration

In this method (2.3.21) is replaced by the iteration (by way of a simple illustration we choose a single variable example),

$$x^{(s+1)} = x^{(s)} - \left[\frac{f(x^{(s)} + \Delta x^{(s)}) - f(x^{(s)})}{\Delta x^{(s)}} \right]^{-1} f(x^{(s)}), \quad (2.3.25)$$

where $\Delta x^{(s)} = -(x^{(s)} - x^{(s-1)})$,

and the derivative $\frac{df}{dx}$ is replaced by its approximation, i.e.

$$\frac{df}{dx} \approx \frac{f(x + \Delta x) - f(x)}{\Delta x}.$$

(2) By inserting a *damping factor* ω such that the iteration process will have the form

$$\underline{x}^{(s+1)} = \underline{x}^{(s)} - \omega^{(s)} [J(\underline{x}^{(s)})]^{-1} \underline{F}(\underline{x}^{(s)}), \quad (2.3.26)$$

and to ensure the norm-reducing inequality

$$||\underline{F}(\underline{x}^{(s+1)})|| \leq ||\underline{F}(\underline{x}^{(s)})||, \quad (2.3.27)$$

to be fulfilled each step. Usually ω is less or equal to unity

(Hall and Watt (1976)).

(3) By shifting the origin of the Jacobian matrix. This method involves adding the diagonal matrix λI to the matrix J , thus (2.3.21) now becomes:

$$\underline{x}^{(s+1)} = \underline{x}^{(s)} - [J(\underline{x}^{(s)}) + \lambda^{(s)} I]^{-1} \underline{F}(\underline{x}^{(s)}), \quad (2.3.28)$$

where the factor λ can be chosen to ensure the validity of (2.3.27).

The modifications (2.3.25), (2.3.26) and (2.3.28) may have the property of superlinear convergence under certain conditions or higher-order convergence under others (Ortega and Rheinboldt (1970)).

2.4 EIGENVALUE PROBLEM

Solving a linear system of equations, such as (2.2.2) , has already been discussed in Section 2.2. The investigation of the dynamic behaviour (i.e. the stability) of such linear systems (which arise in many physical problems, e.g. in electrical or mechanical oscillations) can be based on scalar values called the *eigenvalues*. For example, for a vibration problem the eigenvalues give the natural frequencies of the system. These are especially important because, if external loads are applied at or near these frequencies, resonance will cause an amplification of motion making failure more likely.

Now consider an n^{th} order system

$$\underline{Ax} = \lambda \underline{x} \quad , \quad (2.4.1)$$

where λ is known as the *eigenvalue* (*latent root*, *characteristic number* or *proper number*) of A and \underline{x} its corresponding eigenvector etc. The n values of λ represents the roots of the polynomial which can be expanded from the determinantal equation,

$$P(\lambda) \equiv \det(A - \lambda I) = 0 \quad . \quad (2.4.2)$$

In fact the matrix A also satisfies (2.4.2) as well, i.e. $P(A)=0$.

This is given by the following theorem:

Theorem 2.4.1: (Cayley-Hamilton theorem)

Any square matrix A is a root of its characteristic equation. If

$$P(\lambda) = [\lambda^n + c_1 \lambda^{n-1} + \dots + c_n] \equiv \det(\lambda I - Ax), \text{ then}$$

$$P(A) \equiv A^n + c_1 A^{n-1} + \dots + c_n I = 0.$$

Proof: (see Faddeeva (1959), Demidovich and Maron (1976)).

The problem in (2.4.1) is called a *standard eigenproblem*, an *eigenvalue problem* if the eigenvalues only are required to be determined and an *eigenproblem* if the corresponding eigenvectors are required as well. These may be obtained from the homogeneous equation

$$(A - \lambda I)\underline{x} = \underline{0}.$$

Whenever the *characteristic equation* (2.4.2) has simple zeros, i.e. the matrix A has distinct eigenvalues, each of them possessing a unique corresponding eigenvector, and consequently those eigenvectors are linearly independent the matrix is then called non-defective, (Goult et al (1974), page 9, Ralston (1965) page 470). Otherwise, if there exists $\lambda_1 = \lambda_2 = \dots = \lambda_k \neq \lambda_j$, $1 \leq k < j \leq n$, then the number of the corresponding eigenvectors will be less than or equal to k and hence the whole set of eigenvectors of A fail to form a base of the space since their number is less than the order of the matrix (in this case a matrix is called a *defective matrix*).

Practically, (2.4.2) is not used to determine the eigenvalue(s) of a matrix unless it is of very low-order. Before referring to an alternative strategy we introduce the main definitions and theorems that might be related to this thesis.

Definition 2.4.1:

A real matrix A is said to be

- | | |
|---|---|
| $\left. \begin{array}{l} (1) \text{ Positive definite if } \underline{x}^T A \underline{x} > 0 \\ (2) \text{ Positive semi-definite if } \underline{x}^T A \underline{x} \geq 0 \end{array} \right\}$ | for all non-null, real vector \underline{x} . |
|---|---|

Remark 2.4.1:

A rectangular matrix A of order $(m \times n)$ with linearly independent columns, the product $A^T A$ is symmetric and positive definite. (Broyden (1975), page 34).

Moreover, it can be shown that a real matrix A is *positive definite*, if and only if it is symmetric and all its eigenvalues are positive, *positive semi-definite* if they are greater than or equal to zero and *indefinite* if they are negative, zero, or positive (see Noble (1969)).

Definition 2.4.2:

The n^{th} order matrices A and B are said to be similar if there is a non-singular matrix P such that $P^{-1}AP=B$. Matrix B is said to be obtained from matrix A by a *similarity transformation*, or *orthogonal transformation* if P is *orthogonal* matrix, (i.e. if $P^T=P^{-1}$).

Then both the matrices A and B have the same eigenvalues and their eigenvectors are associated with the relation $Py=\underline{x}$, where \underline{x} and \underline{y} refer to the eigenvectors of A and B respectively.

The last definition is often exploited whenever the standard eigenproblem (2.4.1) is difficult to deal with, thus by use of a similarity transformation the standard problem can be transferred to the so-called *generalised eigenproblem*, i.e.

$$(P^{-1}AP)\underline{x} = \lambda \underline{x}$$

or

$$A\underline{y} = \lambda P\underline{y}$$

where P and \underline{y} are defined as in Definition 2.4.2).

Theorem 2.4.2: (Gerschgorin or Brauer's theorem)

If $A=[a_{ij}]$ is any matrix of order n , then all the eigenvalues of A lie within the union of the circles

$$|\lambda - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i=1(1)n. \quad (2.4.3)$$

Proof: (see Varga (1962), Noble (1969), Smith (1978)).

Since the transposed matrix A^T has the same eigenvalues as A , hence the result of the above theorem for A^T yields (Froberg (1974))

$$|\lambda - a_{jj}| \leq \sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}|, \quad j=1(1)n. \quad (2.4.4)$$

Using result (2.1.1) the inequalities (2.4.3) and (2.4.4) can be written as

$$|\lambda| \leq \sum_{j=1}^n |a_{ij}|, \quad i=1(1)n,$$

and

$$|\lambda| \leq \sum_{i=1}^n |a_{ij}|, \quad j=1(1)n.$$

Hence an estimate of λ can be given by the results,

$$|\lambda| \leq \max_i \sum_{j=1}^n |a_{ij}| \equiv \|A\|_{\infty} \quad (2.4.5)$$

$$|\lambda| \leq \max_j \sum_{i=1}^n |a_{ij}| \equiv \|A\|_1. \quad (2.4.6)$$

If $\rho(A)$ is defined such that $\rho(A) = \max_i |\lambda_i|$, hence the estimate of the spectral radius of A is bounded by the ∞ -norm or the 1-norm of A . In fact, although both norms can be computed easily in practice, theoretically it can be shown that $\rho(A)$ is bounded by any norm of A , i.e.

$$\rho(A) \leq \|A\|. \quad (2.4.7)$$

This result follows from (2.4.1), i.e. $\|\lambda \underline{x}\| = |\lambda| \cdot \|\underline{x}\| = \|\underline{Ax}\| \leq \|A\| \cdot \|\underline{x}\|$ or $|\lambda| \leq \|A\|$ provided \underline{x} is non-null vector.

Determination of the Eigenvalues

In this respect two fundamental approaches are normally adopted, (i) if there exists two eigenvalues (not equal) of ratio less than unity in modulus, then this ratio may be made small if it is raised to a suitable high power. Subsequently, methods based on this approach are often used to calculate one eigenvalue of the matrix. Examples of these strategies are the Power method, inverse iteration, etc...., (ii) to perform a similarity transformation (which is often an orthogonal transformation) so that the matrix can be reduced to either *diagonal* or *tridiagonal* or triangular form where the eigenvalues appear on the principal diagonal or as a recursive Sturm sequence. Methods based on this technique give all the eigenvalues, such methods are Jacobi, Givens, Householder, QR method, etc... . However, we are interested only in method of the first type, thus we briefly demonstrate the following methods.

(a) The Power Method

Let an n^{th} order matrix A which possesses the eigenvalues λ_i , $i=1,2,\dots,n$

such that there exists one of them which has the largest value in modulus λ_1 , (say) (often termed the dominant eigenvalue), i.e.

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \dots \geq |\lambda_n|.$$

Let $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n$ be the corresponding eigenvectors of the eigenvalues λ_i such that their linear combination can be expressed as a vector \underline{y} , i.e.,

$$\underline{y} = \sum_{i=1}^n c_i \underline{x}_i, \quad (2.4.8)$$

where c_i , $i=1(1)n$, are constant coefficients.

For any eigenvalue λ_i we have from (2.4.1)

$$A\underline{x}_i = \lambda_i \underline{x}_i, \quad 1 \leq i \leq n. \quad (2.4.9)$$

Now, operating on \underline{y} in (2.4.8) by A we obtain

$$\begin{aligned} A\underline{y} &= \sum_{i=1}^n c_i A\underline{x}_i \\ &= \sum_{i=1}^n c_i \lambda_i \underline{x}_i \quad (\text{by using (2.4.9)}) \\ &= \lambda_1 \left\{ c_1 \underline{x}_1 + \sum_{i=2}^n c_i \left(\frac{\lambda_i}{\lambda_1} \right) \underline{x}_i \right\}, \end{aligned} \quad (2.4.10)$$

or the iterative form after s steps, (2.4.10) may be written as

$$\underline{y}^{(s)} \equiv A^{(s)} \underline{y} = \lambda_1^{(s)} \left\{ c_1 \underline{x}_1 + \sum_{i=2}^n c_i \left(\frac{\lambda_i}{\lambda_1} \right)^s \underline{x}_i \right\}. \quad (2.4.11)$$

Since $\left| \frac{\lambda_i}{\lambda_1} \right| < 1$, $i=2, \dots, n$ by the initial assumption, therefore the second term in the parentheses of (2.4.11) tends to zero for sufficiently large s . Subsequently, the vector $\underline{y}^{(s)}$ becomes a scalar multiple of \underline{x}_1 and the ratio between the k^{th} component, $1 \leq k \leq n$, of $\underline{y}^{(s+1)}$ and $\underline{y}^{(s)}$ tends to λ_1 , i.e.

$$\lim_{s \rightarrow \infty} \frac{y_k^{(s+1)}}{y_k^{(s)}} = \lambda_1.$$

The practical feature of the algorithm can be summarised as follows. Given a vector $\underline{x}^{(s)}$, the iteration process involves,

Step 1 $\underline{y}^{(s+1)} = A\underline{x}^{(s)}$

Step 2 Choose $\beta^{(s+1)}$ = the element of largest modulus amongst the components of $\underline{y}^{(s+1)}$

Step 3 $\underline{x}^{(s+1)} = \frac{1}{\beta^{(s+1)}} \underline{y}^{(s+1)}$ (normalisation stage)

Step 4 if $\underline{x}^{(s+1)}$ and $\underline{x}^{(s)}$ are sufficiently close, then halt the procedure, otherwise repeat from step 1.

The rate of convergence depends upon the ratio $\left| \frac{\lambda_2}{\lambda_1} \right|$ (where λ_2 is assumed to be the sub-dominant eigenvalue, i.e. the $\lambda_2 = \max_{2 \leq i \leq n} |\lambda_i|$) being very small. Obviously, the smaller the value of this ratio, the faster convergence.

(b) The Inverse Power Method

Any non-singular matrix A and its inverse A^{-1} have the same eigenvectors but reciprocal eigenvalues as can be noticed from (2.4.1) and the equation

$$A^{-1} \underline{x} = \frac{1}{\lambda} \underline{x} . \quad (2.4.12)$$

Therefore, the smallest eigenvalue of A can be determined by obtaining the largest eigenvalue of A^{-1} . Furthermore, it is unnecessary to compute A^{-1} explicitly since the iteration procedure can be carried out as follows.

At iteration s , we compute

Step 1 $\underline{y}^{(s+1)} = A^{-1} \underline{x}^{(s)}$ which can be written as

$$A \underline{y}^{(s+1)} = \underline{x}^{(s)} . \quad (2.4.13)$$

Step 2,3,4 continue as in method (a).

The system (2.4.13) can be solved by a suitable method (such as those discussed earlier or the ones proposed in this thesis). For example, if the LU decomposition process is used initially, then (2.4.13) will be solved cheaply in each successive iteration.

Further it can be shown that for any number ρ , the eigenvectors of the matrix $A - \rho I$ coincide with those of A , but its eigenvalues are $\lambda_i - \rho$, $i=1(1)n$. This is known as *shifting the origin* of the matrix A by the

amount ρ . The shifting strategy is basically introduced to speed up the convergence. For example, if the ratio $\left| \frac{\lambda_2}{\lambda_1} \right|$ is not small enough (i.e., very close to 1), then ρ can be chosen such that the ratio $\max_i \left| \frac{\rho - \lambda_i}{\rho - \lambda_1} \right| < \left| \frac{\lambda_2}{\lambda_1} \right|$ which eventually accelerates the convergence. Likewise adopting the shifting strategy for the inverse power method leads us to solve

$$(A - \rho I) \underline{y}^{(s+1)} = \underline{x}^{(s)}, \quad (2.4.14)$$

instead of (2.4.15) and hence the smallest eigenvalue of A^{-1} is given by $1/(\lambda - \rho)$.

Apart from the scheme of shifting the origin which is referred to as Wilkinson's method (1955), there are other techniques for accelerating the convergence of the Power method such as δ^2 -process, Rayleigh quotient, etc.... (see Ralston (1965), Fadeeva (1959)).

2.5 EVALUATION OF THE SQUARE ROOT OF A SQUARE MATRIX

Let a matrix A of order n possess the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$. The characteristic polynomial which is derived from $\det(A - \lambda I)$ is of order n and may be expressed in the form

$$P(\lambda) = (\lambda - \lambda_1)(\lambda - \lambda_2) \dots (\lambda - \lambda_n) .$$

By the Cayley-Hamilton theorem 2.4.1 matrix A is a root of its own characteristic equation, i.e. $P(A) = 0$, thus we have

$$P(A) = (A - \lambda_1 I)(A - \lambda_2 I) \dots (A - \lambda_n I) = 0 . \quad (2.5.1)$$

Therefore the matrices $\lambda_1 I, \lambda_2 I, \dots, \lambda_n I$ are solutions of the matrix equation $P(A) = 0$. Furthermore, the products of matrices in (2.5.1) may be zero even though no factor is zero (Hohn (1973), page 31), thus $P(A) = 0$ may also have other solutions apart from $\lambda_i I, 1 \leq i \leq n$. (See Jennings (1964), Hohn (1973)).

We should point out that in this thesis our interest is the square root, denoted by $A^{\frac{1}{2}}$, for a positive (or semi-positive) definite matrix A satisfies the following theorem.

Theorem 2.5.1:

The matrix A of order n is a definite (i.e. positive or non-negative) matrix of rank r ($r \leq n$) iff there is a definite matrix $A^{\frac{1}{2}}$ of rank r such that $(A^{\frac{1}{2}})^2 = A$.

Proof: (see Lancaster (1969), p.95).

In his paper, Laasonen (1958) recommended the use of Newton's method for computing the square root of a matrix possessing the properties as stated in the following theorem:

Theorem 2.5.2:

Let A denote a real square matrix with real, positive eigenvalues. Then, the matrix iterative algorithm

$$\left. \begin{aligned} X^{(0)} &= kI \\ X^{(i+1)} &= \frac{1}{2}X^{(i)} + \frac{1}{2}(AX^{(i)})^{-1} \end{aligned} \right\} \quad (2.5.2)$$

where k is a non-zero constant, generates a sequence of matrices which converges to the solution of

$$AX^2 - I = 0, \quad (2.5.3)$$

which has positive eigenvalues. Moreover the rate of convergence is quadratic.

Laasonen also suggested that if the matrix A is non-negative definite, then $A^{\frac{1}{2}} = X$ can be obtained from the algorithm

$$X_{(i+1)} = \frac{1}{2} X_{(i)} + \frac{1}{2} AX_{(i)}^{-1}, \quad (2.5.4)$$

where the initial matrix $X_{(0)}$ is as given in (2.5.2). Therefore, the iterative process (2.5.4) will produce an approximate solution to the equation

$$X^2 - A = 0. \quad (2.5.5)$$

According to the theorem (2.5.1), the solution of (2.5.3) and (2.5.5) by the algorithms (2.5.2) and (2.5.4) respectively preserve the property of the original matrix, i.e. the matrices $A^{-\frac{1}{2}}$ (and $A^{\frac{1}{2}}$) remain positive (and non-negative) if A is also.

Each iteration of both the processes (2.5.2) or (2.5.4) involve the solution of n^2 linear equations. It is recommended that any of the above iterative procedures should be terminated as soon as the difference between two successive solutions $X^{(i)}$ and $X^{(i+1)}$ no longer decreases, otherwise the influence of the round-off errors may be significant on the obtained solution. Laasonen pointed out that in most cases the influence of round-off errors does not become serious due to the quadratic rate of convergence of the process.

2.6 MAIN PROPERTIES OF CONTINUED FRACTIONS

We consider in this section the basic theory of continued fractions and their application which is relevant to the algorithms presented in Chapter 4. A comprehensive study of continued fractions (in particular the convergence theory) is due to H.S. Wall (1948)). Others such as Frank (1962), Blanch (1964),... etc., have contributed to develop the theory and the application of continued fractions.

Definition 2.6.1:

Consider the two variables t and ω associated by the relation

$$\left. \begin{aligned} t_0(\omega) &= b_0 + \omega, \\ t_j(\omega) &= \frac{a_j}{b_j + \omega}, \quad j=1,2,\dots \end{aligned} \right\} \quad (2.6.1)$$

where the a 's and b 's are real or (generally) complex numbers, and the linear transformation of ω into t is expressed in the form:

$$\begin{aligned} t_0 t_1(\omega) &= t_0[t_1(\omega)], \\ t_0 t_1 t_2(\omega) &= t_0 t_1[t_2(\omega)], \\ &\vdots \\ t_0 t_1 t_2 \dots t_k(\omega) &= t_0 t_1 t_2 \dots t_{k-1}[t_k(\omega)], \quad k=1,2,\dots \end{aligned}$$

or
$$T_k(\omega) = \prod_{i=1}^k t_i(\omega) = \prod_{i=1}^{n-1} t_i[t_k(\omega)].$$

Now, for $k=\infty$, we have from (2.6.1)

$$T_\infty(\omega) = b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \frac{a_4}{b_4 + \dots + \frac{a_j}{b_j + \dots}}}}} \quad ((2.6.2a))$$

which is called an *infinite continued fraction*. The abbreviated notation for (2.6.2a) will be used and is

$$T_{\infty}(\omega) = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \dots \cfrac{a_j}{b_j + \dots}}}} \quad (2.6.2b)$$

$$b_0 + \sum_{j=1}^{\infty} \frac{a_j}{b_j},$$

or
$$T_{\infty}(\omega) = [a_0; \frac{a_1}{b_1}, \frac{a_2}{b_2}, \frac{a_3}{b_3}, \dots, \frac{a_j}{b_j}, \dots] \quad (2.6.2c)$$

The fractions $b_0 = \frac{b_0}{1}$, $\frac{a_j}{b_j}$, $j=1,2,\dots$, are called the *components* or the *partial quotients* of the continued fraction (2.6.2), (N.B. the partial quotients $\frac{a_j}{b_j}$ can not be reduced), and a_j, b_j , $j=1,2,\dots$ are called the *partial numerators* and *denominators* respectively. For the case $T_n(\omega)$, $n \neq \infty$, i.e. n is a finite number then the continued fraction is said to be *finite* or *n-component*, i.e.

$$T_n(\omega) = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \dots \cfrac{a_n}{b_n}}}} \quad (2.6.3)$$

If the partial numerators are equal to 1, i.e. $a_i=1$, $i=1,2,\dots$ then (2.6.3) is said to be a *simple* or *standard continued fraction*, i.e.

$$T_{\infty}(\omega) = b_0 + \cfrac{1}{b_1 + \cfrac{1}{b_2 + \cfrac{1}{b_3 + \dots \cfrac{1}{b_j + \dots}}}} \quad (2.6.4)$$

Furthermore, the continued fraction (2.6.2) is said to *converge* if there exists a limit (or has the *value*) v such that

$$\lim_{n \rightarrow \infty} \prod_{i=1}^n t_i(\omega) \equiv \lim_{n \rightarrow \infty} T_n(\omega) = v.$$

This means that at a fixed point $\omega = \infty$ under the transformations t_i , $i=1,2,\dots$ as defined earlier the value of the continued fraction is a limit of an infinite sequence of images.

Similarly at the fixed point $\omega=0$, then $\lim_{n \rightarrow \infty} T_n(0) = \lim_{n \rightarrow \infty} \prod_{i=1}^n t_i(0)$ is defined. The quantity $T_n(0)$ is termed the n^{th} *approximant* or *convergent*. The zeroth (0^{th}) approximant is $t_0(0) = b_0$.

It is shown by mathematical induction (Wall (1948)) that

$$T_n(\omega) \equiv \prod_{i=1}^n t_i(\omega) = \frac{A_{n-1}\omega + A_n}{B_{n-1}\omega + B_n}, \quad n=0,1,2,\dots \quad (2.6.5)$$

where the quantities $A_{n-1}, A_n, B_{n-1}, B_n$ are independent of ω and can be evaluated by the following *fundamental* recurrence formulae.

$$\left. \begin{aligned} A_{j+1} &= b_{j+1}A_j + a_{j+1}A_{j-1} \\ B_{j+1} &= b_{j+1}B_j + a_{j+1}B_{j-1} \end{aligned} \right\}, j=0,1,2,\dots \quad (2.6.6a)$$

and the initial values,

$$\left. \begin{aligned} A_{-1} &= 1, & B_{-1} &= 0 \\ A_0 &= b_0, & B_0 &= 1. \end{aligned} \right\} \quad (2.6.6b)$$

Thus, the n^{th} approximant, i.e. $T_n(0)$ can be easily obtained from (2.6.5), i.e.,

$$T_n(0) = \frac{A_n}{B_n}$$

where A_n and B_n are called the n^{th} numerator and denominator respectively.

Moreover for a *simple* continued fraction, the recurrence relation (2.6.6a)

becomes

$$\left. \begin{aligned} A_{j+1} &= b_{j+1}A_j + A_{j-1} \\ B_{j+1} &= b_{j+1}B_j + B_{j-1} \end{aligned} \right\}, j=0,1,2,\dots, \quad (2.6.7)$$

with the same initial values as (2.6.6b).

Finally, the value of the continued fraction (2.6.2) does exist if the following conditions are fulfilled (Blanch (1964)):

- (i) At most a finite number of the denominators B_k vanish.
- (ii) Given a positive quantity ε , there exists an N such that,

$$\text{for } n \geq N \quad \left| \frac{A_n}{B_n} - \frac{A_{n+k}}{B_{n+k}} \right| < \varepsilon \quad \text{for all positive } k. \quad (2.6.8)$$

The implication of the validity of (2.6.8) ensures the existence of a limit quantity T such that

$$T = \lim_{n \rightarrow \infty} \frac{A_n}{B_n},$$

whereas the failure of (2.6.8) means the continued fraction is said to *diverge* or to be *divergent* and its value can not be assigned.

Periodic Continued Fraction

Definition 2.6.2:

Consider the infinite continued fraction (2.6.1) in the form

$$T(\omega) = \underbrace{\cfrac{a_1}{b_1 +} \cfrac{a_2}{b_2 +} \cdots \cfrac{a_n}{b_n +}}_{\text{1st period}} \underbrace{\cfrac{a_1}{b_1 +} \cfrac{a_2}{b_2 +} \cdots \cfrac{a_n}{b_n +}}_{\text{2nd period}} \cdots \quad (2.6.9)$$

$n=1,2,\dots$

The essential property of the continued fraction (2.6.9) is that its partial numerators and denominators are periodically repeated after n divisions, or the partial quotient $\frac{a_j}{b_j}$, $j=1,2,\dots,n$, is repeated after a period of 'length' or cycle n since its previous occurrence. Thus, equation (2.6.9) is termed an *infinite periodic continued fraction*, and its linear fractional transformation can be expressed by

$$T(\omega) = \cfrac{a_1}{b_1 +} \cfrac{a_2}{b_2 +} \cdots \cfrac{a_n}{b_n + \omega} \quad (2.6.10)$$

Consequently, as in (2.6.5), we introduce (2.6.10) in the form

$$T(\omega) = \frac{A_{n-1}\omega + A_n}{B_{n-1}\omega + B_n},$$

where A_n, B_n refer to the n^{th} numerator and denominator of the continued fraction and their values are given by the recurrence formulae (2.6.6a) with initial values

$$A_1 = 1, \quad B_{-1} = 0$$

$$A_0 = 0, \quad B_0 = 1.$$

We now define the fixed point of the continued fraction.

Definition 2.6.3:

Let the point x be such that

$$x = \frac{A_{n-1}x + A_n}{B_{n-1}x + B_n}$$

holds true. Then there are two values of x which can be obtained by solving the quadratic equation,

$$B_{n-1}x^2 + (B_n - A_{n-1})x - A_n = 0. \quad (2.6.11)$$

which for x_1, x_2 (say), are termed the *fixed points* of the transformation (2.6.10).

Some of the algorithms adopted in Chapter 4 are associated with the numerical evaluation of periodic continued fractions. This is basically formulated by the following theorem.

Theorem 2.6.1:

Let x_1 , and x_2 be the fixed points of the transformation (2.6.10) where a_i, b_i , $i=1,2,\dots,n$ are any complex numbers and $a_i \neq 0$. Let $\frac{A_m}{B_m}$ be the m^{th} approximant of the periodic continued fraction (2.6.9). Then (2.6.9) converges iff x_1 and x_2 are finite numbers satisfying one of the following two conditions:

- (i) $x_1 = x_2$
 or (ii) $\left| \frac{A_{n-1}}{B_{n-1}} - x_2 \right| > \left| \frac{A_{n-1}}{B_{n-1}} - x_1 \right|$, $\frac{A_j}{B_j} \neq x_2$, $j=0,1,2,\dots,n-1$.

If the continued fraction converges, its value is x_1 .

Proof: see Wall (1948), page 37.

Theorem 2.6.2: (Equivalence theorem)

A continued fraction is unchanged in value if some partial numerator and partial denominator, along with the immediately succeeding partial numerator, are multiplied by the same non-zero constant (see Blanch (1964)).

Such a transformation has been termed in (Wall (1948)) an *equivalence transformation*.

Now, consider the following infinite periodic continued fraction with period of 'length' n , $n \geq 1$

$$T = \frac{\alpha_1}{\beta_1} \cfrac{\alpha_2}{\beta_2} \dots \cfrac{\alpha_n}{\beta_n} \cfrac{\alpha_1}{\beta_1} \cfrac{\alpha_2}{\beta_2} \dots \cfrac{\alpha_n}{\beta_n} \dots \quad (2.6.12)$$

By virtue of Theorem 2.6.1 due to successive transformations, the periodic continued fraction (2.6.12) may be expressed in a form with unitary partial denominators, i.e.

$$\hat{T} = \frac{\gamma_1}{1-} \frac{\gamma_2}{1-} \dots \frac{\gamma_n}{1-} \frac{\gamma_1}{1-} \frac{\gamma_2}{1-} \dots \frac{\gamma_n}{1-} \dots, \quad (2.6.13)$$

where $\gamma_1 = \alpha_1/\beta_1$,

$$\gamma_i = \alpha_i/\beta_{i-1}\beta_i, \beta_i \text{ and } \beta_{i-1} \neq 0, i=2,3,\dots,n.$$

It is proved by Blanch (1964) that \hat{T} (2.6.13) will converge to a positive value less or equal to $\frac{1}{2}$ provided that any of the partial numerators is positive and does not exceed $\frac{1}{4}$, i.e.,

$$\text{If } 0 < \gamma_i \leq \frac{1}{4}, \text{ then } \hat{T} \text{ converges and } 0 < \hat{T} \leq \frac{1}{2}. \quad (2.6.14)$$

Okolie (1978) or (Gwans and Okolie(1979)) pointed out that the condition (2.6.14) for the convergence of (2.6.12) can be exploited to introduce a cyclic factorisation of a periodic tridiagonal matrix, i.e.

if α_i and β_i are given by the relations,

$$\left. \begin{aligned} \alpha_1 &= a_1 c_n, \beta_1 = b_n, \\ \alpha_i &= c_k a_{k+1} \\ \beta_i &= b_k \end{aligned} \right\} k=(n-i+1) \bmod n, i=2,3,\dots,n,$$

where $a_i, b_i, c_i, i=1(1)n$ are the coefficients of the periodic tridiagonal matrix

$$\begin{bmatrix} b_1 & c_1 & & a_1 \\ & a_2 & & \\ & & \ddots & \\ & & & c_{n-1} \\ c_n & & 0 & a_n & b_n \end{bmatrix}, \quad (2.6.15)$$

then a periodic continued fraction of the form (2.6.12) converges provided the matrix (2.6.15) is diagonally dominant, in a sense that the

inequalities $\left| \frac{c_i}{b_i} \right|, \left| \frac{a_i}{b_i} \right| \leq \frac{1}{4}, i=1,2,\dots,n$

hold true.

Likewise, we will consider the equivalence theorem and the condition (2.6.14) to introduce the method in Chapter 4 which involves the cyclic factorization of a periodic *general* matrix of bandwidth $2r+1, r \geq 1$ (see Section 2.1).

CHAPTER 3

NUMERICAL SOLUTION OF BOUNDARY VALUE PROBLEMS

3.1 DIFFERENT NUMERICAL APPROACHES FOR SOLVING BOUNDARY VALUE PROBLEMS

To deal with a suitable approach to obtain the solution of certain boundary value problems (b.v.p.) there arise many points which should be taken into account, i.e. the boundary condition(s) which the problem is subject to, the existence and uniqueness of the solution, the stability of the adopted approach, the level of accuracy in the solution which can be attained, ... etc. For example, techniques such as the factorisation of the operators and the use of projection operators are suitable for linear boundary value problems while for the non-linear boundary value problems the non-iterative schemes which are based on continuous transformation are used (Meyer (1973)).

Broadly speaking, numerical techniques have had advantages and disadvantages in practice. The Shooting (or Driving) method, for instance, is a well known approach for initial value problems, Keller (1975) in his survey indicated that this method accounts for nearly one third of the work concerned with the numerical investigation of differential equations. On the other hand the shooting method has many drawbacks due to the difficulties which are encountered in practice, such as 1) the starting solution might not be assured for the convergency of the Newton-Raphson iteration or (and) 2) the method becomes unstable due to its sensitivity to any perturbation in the initial conditions (which accounts for the growth of round-off error) although the numerical method is stable (however, the Multiple or Parallel shooting procedures are proposed to tackle such difficulties), (Hall and Watt (1976)), Keller (1968), Osborne (1969)).

Finite element methods (variational, collocation,... etc.) and finite-difference methods are used for boundary value problems. An important exposition of the recent theoretical advances have been made

on the methods for initial and b.v.p. are collected in Hall and Watt (1976).

Our sole interest is the finite-difference methods which will be discussed in the next section. Whilst for the finite element methods we briefly outline the following.

Finite Element Method

The finite element method is a recent new method which has been used widely during the last three decades. During this time the electronic digital computer has progressed to the stage where it can accomplish considerable amounts of computational work in a short time. The method is commonly used in engineering problems, in particular civil, aeronautical and mechanical engineering, especially for the analysis of stress in solid components. Furthermore, it has been applied even to three-dimensional problems, such as the time-dependent problems involving fluid flow, heat transfer, magnetic field analysis, ... etc. (Fenner (1975), Bathe and Wilson (1976), Martin and Carey (1973)).

The finite element method is based on the idea of partitioning the physical system, such as structures, solid or fluid continua into small non-overlapping *subregions* or *elements*. Each element is a basic unit which has to be considered. Within these elements an approximation function (in the form of polynomials or rational functions, ...etc.) where parameters can be adjusted to ensure the existence of the continuity of the functions in adjacent elements (Mitchell and Wait (1977)). Moreover, an approximating function, generally, can be expressed over the region under consideration containing N nodal points in the form

$$U(x_1, x_2, \dots, x_m) = \sum_{i=1}^N (p_i(x_1, \dots, x_m) u_i(x_1, \dots, x_m) + q_i(x_1, \dots, x_m) \frac{\partial U_i}{\partial x_1} (x_1, \dots, x_m) + r_i(x_1, \dots, x_m) \frac{\partial U_i}{\partial x_2} (x_1, \dots, x_m) + \dots) \quad (3.1.1)$$

where U_i refers to the value of $U(x_1, \dots, x_m)$ at the nodal point i , and p_i, q_i, r_i, \dots etc. are known as *basis functions* which, in fact, are the most important parts of the finite element method. Therefore, to construct the basis functions many techniques are suggested in the literature such as Lagrange, Hermite interpolation formulae for polygonal regions (which can be divided into triangular elements). For example, the simplest form of (3.1.1) (with the absence of the derivative terms) is when $m=2$, i.e. the two-dimensional case. The function $U(x,y)$ ($\equiv U$) can be interpolated at $\frac{1}{2}(s+1)(s+2)$ points with a polynomial of order s , i.e.

$$U(x,y) = \sum_{j=1}^{\frac{1}{2}(s+1)(s+2)} U_j p_j^{(s)}(x,y) . \quad (3.1.2)$$

If the smallest element (the basic unit) is assumed to be the triangle p_1, p_2, p_3 (Fig.3.1.1), then the polynomial (3.1.2) interpolates $U(x,y)$ at $\frac{1}{2}(s+1)(s+2)$ symmetrically placed points on the triangle p_1, p_2, p_3 . For $s>1$, the non-vertex points can be obtained geometrically by dividing each side of the triangle p_1, p_2, p_3 into s equal segments and by joining the points of subdivision by lines drawn parallel to the sides of the triangle (see Fig. 3.1.1) as an example for $s=2,3$).

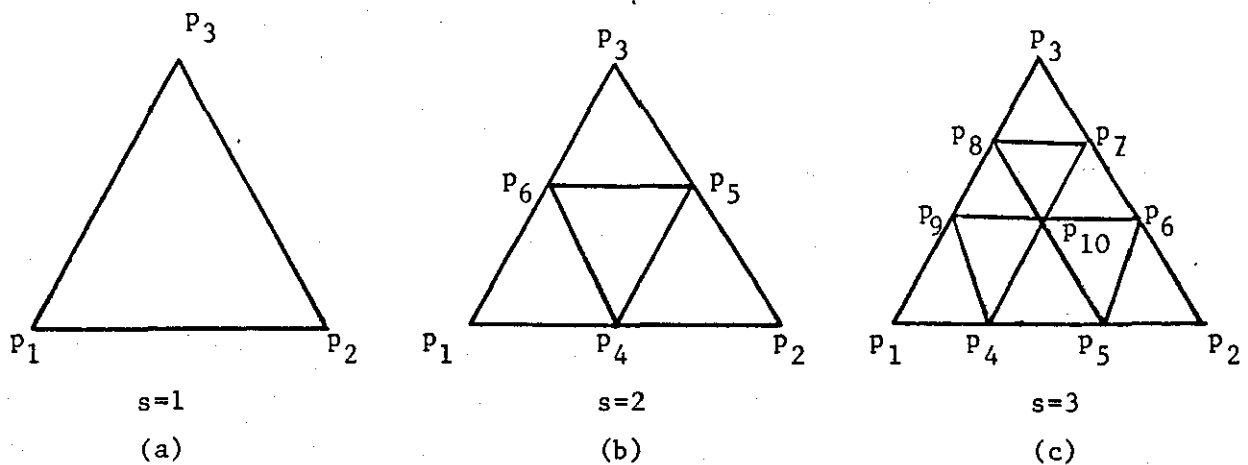


FIGURE 3.1.1

For any $s (\geq 1)$, the coordinates of p_j ($\equiv p_j(x,y)$), $j=1,2,\dots,\frac{1}{2}(s+1)(s+2)$ are determined by the following:

$$\left. \begin{aligned} x &= \frac{1}{s}(\beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3) \\ y &= \frac{1}{s}(\beta_1 y_1 + \beta_2 y_2 + \beta_3 y_3) \end{aligned} \right\} 0 \leq \beta_k \leq s, \quad k=1,2,3 \quad (3.1.3)$$

given

$$\beta_1 + \beta_2 + \beta_3 = s.$$

Now, for $s=1$ (the *linear case*), we have

$$u(x,y) = U_1 p_1(x,y) + U_2 p_2(x,y) + U_3 p_3(x,y),$$

where U_j , $j=1,2,3$ are the values of $U(x,y)$ at the vertices p_j which are now given by

$$\left. \begin{aligned} p_1 &= \det \begin{bmatrix} 1 & x & y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} / D, \\ p_2 &= \det \begin{bmatrix} 1 & x & y \\ 1 & x_1 & y_1 \\ 1 & x_3 & y_3 \end{bmatrix} / D, \\ p_3 &= \det \begin{bmatrix} 1 & x & y \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{bmatrix} / D, \end{aligned} \right\} \quad (3.1.4)$$

where

$$D = \det \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}.$$

For $s=2$ (the *quadratic case*), we have

$$U(x,y) = \sum_{i=1}^6 U_i \phi_i(x,y)$$

where $\phi_i \equiv \phi_i(x,y)$ refer to the functions at the nodes, p_i , $i=1,2,\dots,6$, in Fig.3.1.1(b) and are given by

$$\left. \begin{aligned} \phi_j &= p_j(2p_j-1), \quad j=1,2,3 \\ \phi_4 &= 4p_1 p_2 \\ \phi_5 &= 4p_2 p_3 \\ \phi_6 &= 4p_3 p_1 \end{aligned} \right\}, \quad (3.1.5)$$

and p_1, p_2, p_3 are available in (3.1.4).

For $s=3$ (the *cubic case*), we have

$$U(x,y) = \sum_{i=1}^{10} U_i \psi_i(x,y) ,$$

where $\psi_i \equiv \psi_i(x,y)$ refer to the functions at the nodals p_i , $i=1,2,\dots,10$ in Fig.3.1.1(c), and are determined as follows,

$$\psi_j = \frac{1}{2} p_j (3p_j - 1) 3p_j^{-2}, \quad j=1,2,3$$

$$\psi_4 = \frac{9}{2} p_1 p_2 (3p_1 - 1)$$

$$\psi_5 = \frac{9}{2} p_1 p_2 (3p_2 - 1)$$

Similarly, ψ_6, ψ_7 , can be expressed in terms of p_2, p_3 and ψ_8, ψ_9 in terms of p_3, p_1 .

$U_{10}(x,y)$ can be eliminated as follows

$$U_{10} = \frac{1}{4} \sum_{j=4}^9 U_j - \frac{1}{6} \sum_{j=1}^3 U_j .$$

However, the above procedures or others (such as the *patch test* for non-conforming cases) can be used with any of the finite element methods such as viz. Ritz, Galerkin (or Bubnov-Galerkin), least squares, collocation,... etc. (see Mitchell and Wait (1977)).

3.2 FINITE-DIFFERENCE METHODS

These methods are broadly used for solving boundary value problems which occur in the mathematical physics and engineering fields. The idea is based on discretizing the continuous b.v.p. in order to obtain an approximation to the numerical solution since the analytical solution is either too awkward or impossible to be obtained. Unfortunately (and this is common in the implementation of numerical algorithms) sometimes the numerical solution might be very poor or not acceptable at all; therefore the discretization procedure ought to be improved in a suitable manner (see Section 3.3) or to increase its order (see Section 3.4).

Since most of the important physical problems (such as problems of elasticity, vibration, heat flow,... etc.) are formulated by equations of order two (Gerald (1970)), therefore we will demonstrate using the finite-difference scheme on this type of b.v.p. Moreover, we restrict ourself to consider the central-difference approximations to the derivatives, it is known Fox (1962) that such methods converge rapidly; our argument is that the *forward* or *backward*^{Single sided} approximations possess the quality of poor accuracy (for the simplest form of approximation the latter are of order 1 versus 2 for the former). However, this property cannot be exploited for any system; for instance using central-differences for first order systems may cause problems of stability (Keller (1968), page 105). For this we classify two alternatives:

- (i) *Low-order* discretization which leads to a linear system of equations with *narrow* bandwidth matrix,
- (ii) *high-order* discretization which leads to a smaller linear system of equations with *broad* bandwidth matrix.

It is obvious then that the same accuracy can be achieved from a smaller number of points.

Now, consider the following real *non-linear* two-point b.v.p. for the second-order ordinary differential equation:

$$\left. \begin{aligned} N(y) \equiv y''(x) - f(x, y(x), y'(x)) &= 0, \\ \text{in the range} & \\ -\infty < a \leq x \leq b < \infty \end{aligned} \right\} \quad (3.2.1.a)$$

which has the solution $y(x)$ and with Dirichlet boundary conditions,

$$y(a) = A, \quad y(b) = B. \quad (3.2.1.b)$$

The boundary value problem (3.2.1) is usually called *linear* whenever $f(x, y(x), y'(x))$ is linear in the arguments $y(x)$ and $y'(x)$.

If the interval $[a, b]$ is partitioned into N (an integral number) subintervals, then the obtained interior points are $x_n = a + nh$, $n=0, 1, \dots, N$, and $Nh = b - a$. (Notice that sometimes it is preferred to normalize the range $[a, b]$ to the form $[0, 1]$ by setting $x = (b-a)y + a$, probably for programming purposes).

Since the purpose of the discretization process when applied to differential equations is to replace the derivatives by their corresponding finite-difference approximations, therefore we introduce the following general formulae of approximation for the first and second derivatives which occur in the boundary problem (3.2.1), (Froberg (1974)).

At any interior point x_n , we have

$$y'(x_n) = \mu h^{-1} \sum_{t=0}^{\infty} (-1)^t \frac{(t!)}{(2t+1)!} \delta^{2t+1} y(x_n), \quad (3.2.2)$$

$$y''(x_n) = 2h^{-2} \sum_{t=0}^{\infty} (-1)^t \frac{(t!)}{(2t+2)!} \delta^{2t+2} y(x_n), \quad (3.2.3)$$

where δ is the central difference operator, and μ the average operator, i.e.

$$\delta y(x) = y(x+h/2) - y(x-h/2),$$

$$\mu y(x) = \frac{1}{2} (y(x+h/2) + y(x-h/2)).$$

Finally, we see that finite-difference methods, in general, are reliable from the point of view of their convergency as long as the smoothness of the functions and small mesh sizes are provided, while they may diverge if the interval is too big or there exists discontinuities of the functions (or its derivatives).

3.3 LOW-ORDER DISCRETIZATION

The simplest common approximation to the continuous boundary value problem (3.2.1) employing (3.2.2) and (3.2.3) has the linearized discrete equations:

$$\left. \begin{aligned} N_h(y_n) &\equiv \alpha_n y_{n-1} + \beta_n y_n + \gamma_n y_{n+1} = -g(x_n), \quad 1 \leq n \leq N-1 \\ y_0 &= A, \quad y_N = B \end{aligned} \right\} \quad (3.3.1)$$

where

$$\begin{aligned} \alpha_n &= -h^{-2} - \frac{1}{2}h^{-1}p(x_n), \\ \beta_n &= 2h^{-2} + q(x_n) \\ \gamma_n &= -h^{-2} + \frac{1}{2}h^{-1}p(x_n), \end{aligned}$$

and $p(x)$ and $q(x)$ represents the Frechet derivation, i.e.,

$$p(x) = \frac{\partial}{\partial y} f(x, y(x), y'(x)) \quad , \quad (3.3.2)$$

$$q(x) = \frac{\partial}{\partial y'} f(x, y(x), y'(x)) \quad , \quad (3.3.3)$$

and $N_h(y_n)$ is the discrete nonlinear operator for the continuous $N(y)$.

It is known, that the tridiagonal system of the discrete equations (3.3.1) can be solved by many ways, e.g. by the well-known Gauss elimination in the compact form as given by Varga (1962) to yield a solution which differs from the analytical solution by an error of order h^2 ($0 \equiv (h^2)$), notably this type of error is called in the literature as a *global truncation (discretization) error* (E_n), i.e.

$$\begin{aligned} E_n &= |\text{the difference between the exact and computed solution of } X_n| \\ &= O(h^2). \end{aligned}$$

Furthermore, since each discrete equation of (3.3.1) is a truncated form of the actual derivatives, then another type of error which is associated with each equation is considered. It is called the *local truncation (discretization) error*, τ_n , in notation we may write,

$$\tau_n = N_h(y_n) - N(y_n), \quad 1 \leq n \leq N-1, \quad (3.3.4)$$

The relationship between both of the above-mentioned types of errors is as follows:

$$\max_{0 \leq n \leq N} |E_n| \leq C \max_{0 \leq n \leq N} |\tau_n|, \quad (3.3.5)$$

where C is a constant independent of h .

If the inequality (3.3.5) holds, then N_h is called *stable* and *consistent* of order p (positive integer) provided that $\max_{0 \leq n \leq N} |\tau_n| \rightarrow 0$, where $\tau_n = O(h^p)$. Thus, if N_h possesses these two properties, the convergence to the numerical solution is guaranteed. However, the stability, consistency and convergency have been investigated fully by Keller (1968), Pereyra (1973) and Henrici (1962), but here we briefly outline the procedure by which this is done for the nonlinear case only.

Let

$$K = \max(1, \frac{1}{Q^*}), \text{ where } 0 < Q^* \leq q(x), a \leq x \leq b, \quad (3.3.6)$$

$$M_r = y^r(\zeta_n), \quad x_{n-1} \leq \zeta_n \leq x_{n+1}, \quad 1 \leq n \leq N-1,$$

(where y^r denotes that the derivative is of order r),

$$P^* = \max_{0 \leq x \leq b} |p(x)|$$

and
$$h \leq \frac{2}{P^*}.$$

Therefore, for the b.v.p. (3.2.1) we have

$$\tau_n = \frac{h^2}{12} (M_4 + 2P^*M_3) + O(h^4) \equiv O(h^2), \quad (3.3.7)$$

i.e. the local discretization error is of order 2, and hence N_h is consistent of order 2. Furthermore, on the assumption that the first derivative in (3.2.1a) is absent (i.e. $p(x) \equiv 0$), then (3.3.7) can be expressed as (Pereyra (1973)):

$$\tau_n = - \sum_{k=1}^J \frac{2}{(2k+2)!} y^{2k+2}(x_n) h^{2k} + O(h^{2J+2}). \quad (3.3.8)$$

Also, N_h is stable if the result in (3.3.5) is valid, provided that the mesh size, h , is small and the existence and uniqueness of the solution are proved.

As mentioned above, the error (E_n) in the approximate solution is of $O(h^2)$ which can be improved as follows:

$$E_n = h^2 \theta(x_n) + O(h^4), \quad 0 \leq n \leq N, \quad (3.3.9)$$

where the function $\theta(x)$ is continuous and twice differentiable and satisfies the following b.v.p.:

$$\left. \begin{aligned} \theta''(x) - p(x)\theta'(x) - q(x)\theta(x) + g(x) &= + \frac{h^2}{12} (\theta'v(x) - 2p(x)\theta'''(x)) \\ \text{with } \theta(a) = \theta(b) &= 0, \end{aligned} \right\} \quad (3.3.10)$$

where $p(x), q(x)$ are given by (3.3.2) and (3.3.3) respectively.

Furthermore, if θ_{2j} , $j \geq 1$, satisfy the boundary value problem (3.3.10) which implicitly assumed the sufficient differentiability of the θ 's, then the asymptotic expression of (3.3.9) can be expressed as

$$E_n = \sum_{j=1}^K \theta_{2j}(x) h^{2j} + O(h^{2K+2}), \quad K \geq 1, \quad (3.3.11)$$

and is of order $2K$.

The result of (3.3.11) shows that the possibility of achieving high accuracy by finite-difference schemes is available. Two approaches are capable of fulfilling this task. The first, is called the *Richardson extrapolation* (deferred approach to the limit or extrapolation to zero mesh width). Starting with a coarse mesh, the procedure is based on halving the mesh at each step of the process, which consequently increases the dimension of the system to be solved at each step (see Keller (1968)). The second approach is the *difference correction* or *deferred correction* which is the only one considered in this thesis.

The Deferred Correction Method

A technique known as "difference correction" was developed by L. Fox, (1957), although from about 1962, the technique has become known interchangeably with "deferred correction". This technique has been applied by Fox and others to a range of problems involving transcendental equations.

Subsequently, this theoretical work has been applied to give solutions to problems requiring advanced calculus tools arising in natural science and engineering. For example, application to first order systems have been studied by Pereyra (1966, 1967, 1968). The implementation of this work has been considered by others, i.e., Daniel and Martin (1977) using Numerov's method with deferred corrections.

The philosophy of deferred correction is to improve the approximate solution obtainable using higher finite difference formulae by considering the asymptotic expansion for local discretization error.

A brief demonstration for the deferred correction method can be introduced as follows:

Reverting to the b.v.p. (3.2.1) and using the approximation formulae (3.2.2) and (3.2.3), we have

$$Cy_n - f(x_n, y_n, \tilde{C}y_n) = 0, \quad (3.3.12)$$

where

$$\left. \begin{aligned} C &= h^{-2}(\delta^2 - \frac{1}{12}\delta^4 + \frac{1}{90}\delta^6 - \frac{1}{560}\delta^8 + \dots), \\ \tilde{C} &= h^{-1}(\mu\delta - \frac{1}{6}\mu\delta^3 + \frac{1}{30}\mu\delta^5 - \frac{1}{140}\mu\delta^7 + \dots), \end{aligned} \right\} \quad (3.3.13)$$

To obtain a tridiagonal system (as in (3.3.1), we should consider the first terms of (3.3.13), and hence (3.3.12) becomes,

$$h^{-2}(y_{n+1} - 2y_n + y_{n-1}) + Dy_n - f(x_n, y_n, \frac{h^{-1}(y_{n+1} - y_{n-1})}{2} + \tilde{D}y_n) = 0 \quad (3.3.14)$$

where

$$\left. \begin{aligned} D &= C - h^{-2}\delta^2, \\ \tilde{D} &= \tilde{C} - h^{-1}\mu\delta \end{aligned} \right\} \quad (3.3.15)$$

It is clear that by setting $D=\tilde{D}=0$ we obtain the system of equations

of which equation (3.3.1) is its linearized form.

The iterative procedure of deferred correction is basically presented in the form:

$$\left. \begin{aligned} h^{-2}(y_{n+1}^{(s)} - 2y_n^{(s)} + y_{n-1}^{(s)}) + Dy_n^{(s-1)} - f(x_n, y_n^{(s)}) - \frac{h^{-1}(y_{n+1}^{(s)} - y_{n-1}^{(s)})}{2} + \tilde{Dy}_n^{(s-1)} &= 0 \\ \text{with } y_0^{(s)} = A, y_N^{(s)} = B, \end{aligned} \right\} \quad (3.3.16)$$

where the superscript s refers to the number of iterations, i.e. the obtainable successive approximate solutions are $y^{(1)}, y^{(2)}, y^{(3)}, \dots$, and $y^{(s)}$ is a 'good' initial starting solution to the process; also $Dy_n^{(0)} = \tilde{Dy}_n^{(0)} = 0$.

Now, assume y' does not exist in (3.2.1), i.e. $\tilde{C}=0$, therefore (3.3.16) can be rewritten in the simpler form:

$$h^{-2}(y_{n+1}^{(s)} - 2y_n^{(s)} + y_{n-1}^{(s)}) - f(x_n, y_n^{(s)}) = -D(y_n^{(s-1)}), \quad 1 \leq n \leq N-1, \quad (3.3.17)$$

It can be shown that the first solution $y^{(1)}$ has errors of $O(h^4)$, $y^{(2)}$ of $O(h^6)$, ... and $y^{(s)}$ of $O(h^{2s+2})$ (Pereyra (1973)). Obviously, during the iterative process, in order to obtain high order accuracy in the numerical solution implies the use of high order finite-difference symmetric formulae, which means the involvement of more neighbour points to any interior point of the range. Symmetric formulae of high order cannot be augmented at the grid points adjacent to the boundaries (x_0 and x_N), therefore the alternatives must be unsymmetric formulae *with* the same order.

However, to compute the coefficients due to using such formulae for the implementation purposes some automatic methods have been provided by Pereyra (see next section).

If the b.v.p. (3.2.1) is periodic, i.e. the boundary conditions are of the form (3.2.47) such difficulties do not occur because the symmetric formulae will be applicable at any grid point of the range.

The iterative deferred correction scheme involves solving a set of

non-linear algebraic equations (3.3.17) (or (3.3.16)) at each step.

Consequently, two types of iteration are required per step (Fox (1977)), i.e.

- (i) *Inner iteration* - to solve a set of non-linear equations,
- (ii) *Outer iteration* - to apply (3.3.17) (or (3.3.16)) to obtain a new approximation, $y_n^{(s)}$ say, at stage s , by computing the correction term $\Delta y_n^{(s-1)}$ (say), so that $y_n^{(s+1)} + \Delta y_n^{(s-1)} = y_n^{(s)}$, where $\Delta y_n = 0$ at the boundaries (see Section 3.4).

When the non-linear equations of the inner iteration are solved usually by Newton's method; where each step requires the solution of tridiagonal systems (of dimension $N-1$) but the Jacobian matrix would of necessity be re-computed at each step (unless the guarantee of convergence is desired). As an initial approximate solution for Newton's method the linear interpolation between the boundaries (i.e. $y(a)$ and $y(b)$) is recommended if no more than (3.2.1b) information is supplied. On the other hand, if (3.2.1) is a linear b.v.p., then the inner iteration involves one step.

Finally, it may be important to indicate that the asymptotic expansion for the global discretization error is not necessary at all for the practical implementation compared to the Richardson's extrapolation procedure, but for theoretical argument only (Pereyra (1973)). While the asymptotic expansion of the local discretization error forms the basis of the deferred correction method, which has a form in terms of powers in h similar to (3.3.11).

3.4 HIGH-ORDER DISCRETIZATION

The procedure above (Section 3.3) can be extended to prove the theoretical justification (mainly the stability). This seems to be a very difficult intractable topic which has yet to be discussed fully in the literature. In this thesis we will assume that this extension to high order discretization is possible which eventually yields a small matrix with wide bandwidth, in order to proceed to the even more interesting problem of determining new algorithms for obtaining the solution procedures.

High order discretization for linear two-point b.v.p. has been investigated by Shoosmith (1973) which will be shown briefly later in this section.

Now, reverting to the b.v.p. (3.2.1) we consider the extension of the previous approach discussed in Section 3.3 (including the deferred correction technique) and for the non-uniform spacing case.

Initially, we assume that $f(\equiv f(x, y(x), y'(x)))$ in (3.2.1) satisfies the three following conditions:

- (i) f is a continuous function of x, y and y' at least in the interior points of the considered range of integration, $[a, b]$,
- (ii) f is Lipschitzian, i.e.,

$$|f(x, y, y') - f(x, z, z')| \leq K_1 |y - z| + K_2 |y' - z'|$$

$$\text{where } K_1 = \sup_{(x, y, y')} |f_y(x, y, y')|$$

$$\text{and } K_2 = \sup_{(x, y, y')} |f_{y'}(x, y, y')|$$

provided that

- (iii) $f_y(x, y, y')$ and $f_{y'}(x, y, y')$ exists.

Consequently, a unique solution to the b.v.p. (3.2.1) must exist.

Let h_n denote the space between any two points, i.e.

$$h_n = x_{n+1} - x_n, \quad Q \leq n \leq N+Q-1, \quad (3.4.1)$$

and Q (any positive integer) is the limit of extrapolation beyond

the given interval $[a, b]$, and N is defined in Section 3.2.

Also, we define the operator 'L' as follows

$$L(\zeta_{s_1, n}, \zeta_{s_2, n}, \dots, \zeta_{s_k, n})y_n = \zeta_{s_1, n}y_{n+s_1} + \dots + \zeta_{s_k, n}y_{n+s_k}.$$

We introduce the following different order types of approximation for y'' and y' :

(1) 2nd-order discretization:

$$\left. \begin{aligned} y'' &= \alpha_{1, n}y_{n+1} + \alpha_{0, n}y_n + \alpha_{-1, n}y_{n-1} \equiv L(\alpha_{1, n}, \alpha_{0, n}, \alpha_{-1, n})y_n \\ y' &= \beta_{1, n}y_{n+1} + \beta_{0, n}y_n + \beta_{-1, n}y_{n-1} \equiv L(\beta_{1, n}, \beta_{0, n}, \beta_{-1, n})y_n \end{aligned} \right\}, \quad (3.4.2)$$

(2) 4th-order discretization:

$$\left. \begin{aligned} y'' &= L(\alpha_{2, n}, \alpha_{1, n}, \alpha_{0, n}, \alpha_{-1, n}, \alpha_{-2, n})y_n \\ y' &= L(\beta_{2, n}, \beta_{1, n}, \beta_{0, n}, \beta_{-1, n}, \beta_{-2, n})y_n \end{aligned} \right\}, \quad (3.4.3)$$

(3) 'General'-order (of order $2r$, $r \geq 1$),

$$\left. \begin{aligned} y'' &= L(\alpha_{r, n}, \alpha_{r-1, n}, \dots, \alpha_{0, n}, \dots, \alpha_{-r+1, n}, \alpha_{-r, n})y_n \\ y' &= L(\beta_{r, n}, \beta_{r-1, n}, \dots, \beta_{0, n}, \dots, \beta_{-r+1, n}, \beta_{-r, n})y_n \end{aligned} \right\}, \quad (3.4.4)$$

In fact the coefficients α 's and β 's in (3.4.4) are functions of x_n .

They have constant values whenever the equal spacing case is considered;

for example in (3.4.2):

$$\begin{aligned} \alpha_{1, n} &= \alpha_{-1, n} = 1/h^2, \quad \alpha_{0, n} = -2/h^2 \\ \text{and} \quad \beta_{1, n} &= -\beta_{-1, n} = 1/2h, \quad \beta_{0, n} = 0, \\ \text{in (3.4.3)} \quad \alpha_{2, n} &= \alpha_{-2, n} = -1/12h^2 \\ \alpha_{1, n} &= \alpha_{-1, n} = 16/12h^2, \quad \alpha_{0, n} = -30/12h^2 \\ \text{and} \quad -\beta_{2, n} &= \beta_{-2, n} = 1/12h, \\ \beta_{1, n} &= -\beta_{-1, n} = 8/12h, \quad \beta_{0, n} = 0 \end{aligned}$$

(c.f. (3.2.2) and (3.2.3)), and h is as defined in Section 3.2).

Therefore, the discretization form of (3.2.1) using (3.4.4) becomes

$$L(\alpha_{r,n}, \dots, \alpha_{0,n}, \dots, \alpha_{-r,n}) y_n - f(x_n, y_n) L(\beta_{r,n}, \dots, \beta_{0,n}, \dots, \beta_{-r,n}) y_n = 0$$

$$y(a) = A, \quad y(b) = B. \quad (3.4.5)$$

Notice that if (3.4.5) is considered for equal spacing, then the local discretization error is of $O(h^{2r})$, otherwise the order is reduced to one less.

Now, we expand y_{n+r}, \dots, y_{n+1} and y_{n-1}, \dots, y_{n-r} by a Taylor's expansion as follows:

$$\left. \begin{aligned} y_{n+1} &= y_n + h_n y'_n + \frac{h_n^2}{2!} y''_n + \dots \\ y_{n-1} &= y_n - h_{n-1} y'_n + \frac{h_{n-1}^2}{2!} y''_n - \dots \\ y_{n+2} &= y_n + (h_n + h_{n+1}) y'_n + \frac{1}{2!} (h_n + h_{n+1})^2 y''_n + \dots \\ y_{n-2} &= y_n - (h_{n-1} + h_{n-2}) y'_n + \frac{1}{2!} (h_{n-1} + h_{n-2})^2 y''_n - \dots \\ &\vdots \\ y_{n+r} &= y_n + (h_n + h_{n+1} + \dots + h_{n+r-1}) y'_n + \frac{1}{2!} (h_n + h_{n+1} + \dots + h_{n+r-1})^2 y''_n + \dots \\ y_{n-r} &= y_n - (h_{n-1} + \dots + h_{n-r}) y'_n + \frac{1}{2!} (h_{n-1} + \dots + h_{n-r})^2 y''_n - \dots \end{aligned} \right\} \quad (3.4.6)$$

Hence from (3.4.5) and employing (3.4.6) we can easily obtain the following two systems of equations in terms of the unknown α 's and β 's respectively:

$$\left. \begin{aligned} \alpha_{r,n} t_{r,n}^{\alpha_{r-1,n}} + \dots + \alpha_{0,n} t_{0,n}^{\alpha_{-r+1,n}} + \alpha_{-r,n} t_{-r,n}^{\alpha_{-r,n}} &= 0 \\ \alpha_{r,n} t_{r,n}^{\alpha_{r,n}} + \dots + \alpha_{1,n} t_{1,n}^{\alpha_{-1,n}} + \alpha_{-1,n} t_{-1,n}^{\alpha_{-1,n}} + \dots + \alpha_{-r,n} t_{-r,n}^{\alpha_{-r,n}} &= 0 \\ \alpha_{r,n} \frac{1}{2!} t_{r,n}^2 + \dots + \alpha_{1,n} \frac{1}{2!} t_{1,n}^2 + \alpha_{-1,n} \frac{1}{2!} s_{-1,n}^2 + \dots + \alpha_{-r,n} \frac{1}{2!} s_{-r,n}^2 &= 1 \\ &\vdots \\ \alpha_{r,n} \frac{1}{k!} t_{r,n}^k + \dots + \alpha_{1,n} \frac{1}{k!} t_{1,n}^k + \alpha_{-1,n} \frac{(-1)^k}{k!} s_{-1,n}^k + \dots + \alpha_{-r,n} \frac{(-1)^k}{k!} s_{-r,n}^k &= \delta \end{aligned} \right\} \quad (3.4.7)$$

and

$$\left. \begin{aligned} \beta_{r,n} + \dots + \beta_{0,n} + \dots + \beta_{-r,n} &= 0 \\ \beta_{r,n} t_{r,n}^k + \dots + \beta_{1,n} t_{1,n}^k + (-1)^k \beta_{-1,n} s_{-1,n}^k + \dots + (-1)^k \beta_{-r,n} s_{-r,n}^k &= \delta \end{aligned} \right\} \quad (3.4.8)$$

where in both (3.4.7) and (3.4.8), $k=1, \dots, 2r$ and δ is the Kronecker delta, i.e.

$$\delta = \begin{cases} 1, & k=2 \text{ in (3.4.7), } k=1 \text{ in (3.4.8)} \\ 0, & \text{otherwise} \end{cases}$$

$$\left. \begin{aligned} t_{j,n} &= h_n + \dots + h_{n+j-1} \\ s_{-j,n} &= h_{n-1} + \dots + h_{n-j} \end{aligned} \right\} \quad j=1(1)r.$$

Thus, we rewrite (3.4.7) and (3.4.8) in the compact form, i.e.

$$\underline{V}\underline{w} = \underline{z} \quad (3.4.9)$$

and

$$\underline{\widetilde{V}}\underline{\widetilde{w}} = \underline{\widetilde{z}}, \quad (3.4.10)$$

where the matrix V is of order $(2r+1)$ and has the *Vandermonde* form

$$V = \begin{bmatrix} 1 & 1 & \cdot & \cdot & \cdot & 1 & 1 \\ \gamma_0 & \gamma_1 & \cdot & \cdot & \cdot & \gamma_{2r-1} & \gamma_{2r} \\ \gamma_0^2 & \gamma_1^2 & \cdot & \cdot & \cdot & \gamma_{2r-1}^2 & \gamma_{2r}^2 \\ \cdot & \cdot & & & & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & & \cdot & \cdot \\ \gamma_0^{2r-1} & \gamma_1^{2r-1} & \cdot & \cdot & \cdot & \gamma_{2r-1}^{2r-1} & \gamma_{2r}^{2r-1} \\ \gamma_0^{2r} & \gamma_1^{2r} & \cdot & \cdot & \cdot & \gamma_{2r-1}^{2r} & \gamma_{2r}^{2r} \end{bmatrix},$$

where

$$\gamma_k = \begin{cases} t_{r-k}, & 0 \leq k \leq r-1 \\ 0, & k=r \\ s_{r-k}, & r+1 \leq k \leq 2r \end{cases} \quad k=0(1)2r,$$

or from the definitions of t and s above and h_n in (3.4.1), we define the γ 's more explicitly in terms of x_n , as follows:

$$\gamma_k = x_{n+r-k} - x_n, \quad k=0(1)2r. \quad (3.4.11)$$

The vectors \underline{w} and \underline{z} of (3.4.9), \tilde{w} and \tilde{z} of (3.4.10) have $(2r+1)$ components and have the form,

$$\underline{w} = [\alpha_{r,n}, \dots, \alpha_{0,n}, \dots, \alpha_{-r,n}]^T, \quad (3.4.12a)$$

$$\tilde{w} = [\beta_{r,n}, \dots, \beta_{0,n}, \dots, \beta_{-r,n}]^T$$

and

$$\underline{z} = [0, 0, 2, 0, \dots, 0]^T \quad (3.4.12b)$$

$$\tilde{z} = [0, 1, 0, \dots, 0]$$

The Vandermonde systems, i.e. (3.4.9) and (3.4.10) can be solved for \underline{w} and \tilde{w} respectively by using the procedure PVAND suggested by Bjorck and Pereyra (1970).

Equation (3.4.5) can be presented in a more accurate approximation form (cf. (3.3.17)), that is

$$L(\alpha_{r,n}, \dots, \alpha_{-r,n}) y_n^{(s)} - f(x_n, y_n^{(s)}, L(\beta_{r,n}, \dots, \beta_{-r,n}) y_n^{(s)}) = D(y_n^{(s-1)}) \quad (3.4.13)$$

(s is the number of iterations),

where the deferred correction term of order $2r+2$ is defined below:

$$D(y_n) = - \sum_{i=-j}^j \bar{\alpha}_{i,n} y_{n+i} + f(x_n, y_n, \sum_{i=-j}^j \bar{\beta}_{i,n} y_{n+i})$$

$$+ L(\alpha_{r,n}, \dots, \alpha_{-r,n}) y_n - f(x_n, y_n, L(\beta_{r,n}, \dots, \beta_{-r,n}) y_n)$$

where $j=r+1$, $-Q+j \leq n \leq N+Q-j$. (3.4.14)

At any grid point, x_n , the coefficients $\bar{\alpha}_{i,n}$ and $\bar{\beta}_{i,n}$ can be determined from (3.4.9) and (3.4.10) respectively; but \underline{z} and \tilde{z} both have 'zero' components and their non-zero element is as in (3.4.12b), while the elements of the matrix V , i.e. γ 's should be evaluated as follows (at any point x_n)

$$\gamma_k = x_{n-r-1+k} - x_n, \quad k=0(1)2r+2.$$

By virtue of (3.4.14) it may be desired to extrapolate values of y_n

beyond the range $[a,b]$, up to Q points from both end points (boundaries).

This can be accomplished by two alternatives:

- (A) The Newton's backward and forward interpolation formulae (see Fröberg, (1974), which can be modified into the form (Audish (1978)), (see Fig. 3.4.1),

$$\left. \begin{aligned} y_{N+i} &= \sum_{j=0}^p \phi_j y_{N-j} \\ y_{-i} &= \sum_{j=0}^p \psi_j y_j \end{aligned} \right\} \quad i=1(1)Q \text{ and } p+1 \leq N. \quad (3.4.15)$$

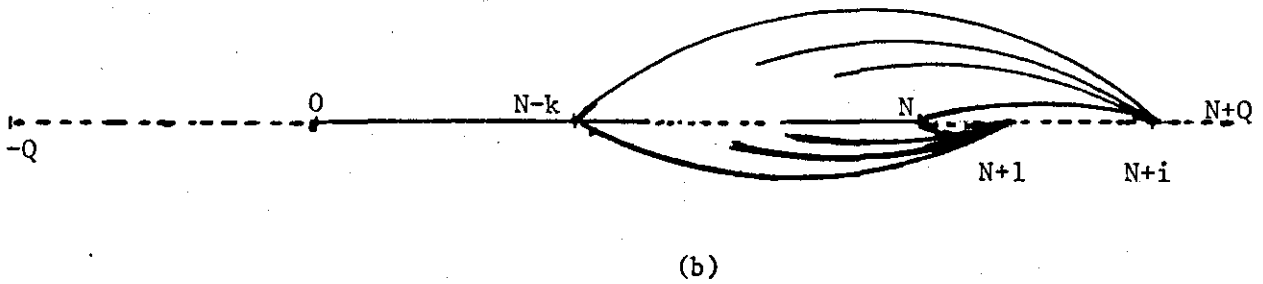
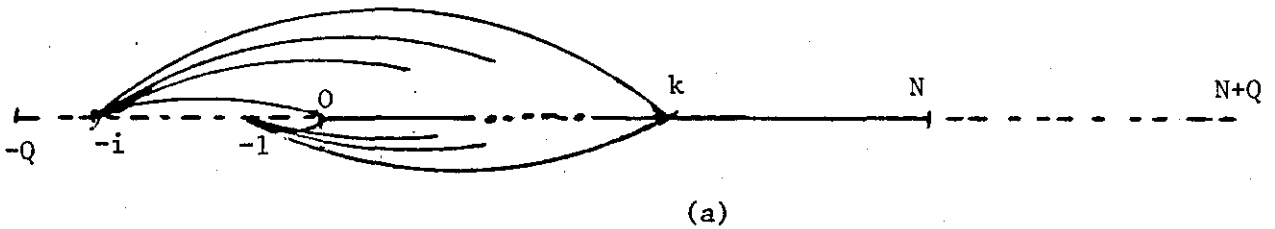


FIGURE (3.4.1a): Using the 1st of (3.4.15) to compute y_{-1}, \dots, y_{-Q}

FIGURE (3.4.1b): Using the 2nd of (3.4.15) to compute y_{N+1}, \dots, y_{N+Q}

The weights ϕ 's and ψ 's in (3.4.15) are obtained from (3.4.9) as in the same manner as for the $\bar{\alpha}$'s in (3.4.14) but the value of γ (of the matrix V) are obtained from the both sides of the range as follows, (at any point outside the range)

$$\left. \begin{array}{l} \text{for } \phi\text{'s:} \quad \gamma_j = x_{N-j} - x_{N+n} \\ \text{and for } \psi\text{'s:} \quad \gamma_j = x_j - x_{-n} \end{array} \right\} \begin{array}{l} j=0, \dots, p-1 \text{ (p is defined in (3.4.15))} \\ 1 \leq n \leq Q \end{array}$$

while the r.h.s. vector, i.e. z , for both the ϕ 's and ψ 's, will have p components, and has the form,

$$z = [1, 0, \dots, 0]^T.$$

or

(B) The differential equation scheme, i.e.

$$L(\alpha_{r,n}, \dots, \alpha_{-r,n}) y_n - f(x_n, y_n, L(\beta_{r,n}, \dots, \beta_{-r,n}) y_n) = D(y_n), \quad (3.4.16)$$

Now, since our discretization procedure is of order $2r$, hence we are willing to use a ' $2r+1$ '-point symmetric formulae at the interior points x_r, \dots, x_{N-r} . Whilst imposing the same order formulae at the remainder of the interior points (i.e. x_1, \dots, x_r , and $x_{N-r+1}, \dots, x_{N-1}$) we introduce $2r-2$ (for $r>1$) extra unknowns, i.e. $y_{-1}, \dots, y_{-(r-1)}$ and $y_{N+1}, \dots, y_{N+r-1}$, and hence we end up with $N+2r-3$ unknowns (assuming that y_0 and y_N are given) for $N-1$ equations. To overcome this difficulty one must seek for some 'practical' non-symmetric formulae (or perhaps a linear combination of non-symmetric and symmetric formulae of order $2r-2$ (see page 79).

Nevertheless, we will assume that these non-symmetric formulae are provided, thus we may write the final form at both sides of the interval, the approximation for y'' and y' as follows:

(1) for $n=1, \dots, r-1$

$$\left. \begin{array}{l} y''_n = \hat{\alpha}_{r,n} y_{n+r} + \dots + \hat{\alpha}_{0,n} y_n + \sum_{k=1}^n \hat{\alpha}_{-k,n} y_{n-k} \\ \text{or } y'_n = \hat{\beta}_{r,n} y_{n+r} + \dots + \hat{\beta}_{0,n} y_n + \sum_{k=1}^n \hat{\beta}_{-k,n} y_{n-k} \end{array} \right\}, \quad (3.4.17)$$

(2) for $n=N+1, \dots, N-r+1$

$$\left. \begin{aligned} y_n'' &= \sum_{k=n}^1 \hat{\alpha}_{k,n} y_{n+k} + \hat{\alpha}_{0,n} y_n + \dots + \hat{\alpha}_{-r,n} y_{n-r} \\ y_n' &= \sum_{k=n}^1 \hat{\beta}_{k,n} y_{n+k} + \hat{\beta}_{0,n} y_n + \dots + \hat{\beta}_{-r,n} y_{n-r} \end{aligned} \right\}, \quad (3.4.18)$$

returning to (3.4.13), as indicated in the previous section, at each step of the iterative procedure we need to solve a system which consists of the discrete equations,

$$\phi_n(\underline{y}) \equiv L(\alpha_{r,n}, \dots, \alpha_{-r,n}) y_n - f(x_n, y_n, L(\beta_{r,n}, \dots, \beta_{-r,n}) y_n) - D(y_n) = 0 \quad (3.4.19a)$$

$$\text{with } y_0 = y(a), \quad y_N = y(b), \quad (3.4.19b)$$

Assuming that f possesses the properties listed earlier in this section and considering the possibility of providing the desirable non-symmetric formulae, thus by differentiating $\phi_n(\underline{y})$ in (3.4.19b), we obtain the following,

$$\hat{A}_{j,n} = \hat{\alpha}_{j,n} - \hat{\beta}_{j,n} \frac{\partial f}{\partial y'} - C, \quad j=r, r-1, \dots, 0, -1, \dots, -n, \quad (1 \leq n \leq r-1), \quad (3.4.20a)$$

$$\hat{A}_{j,n} = \hat{\alpha}_{j,n} - \hat{\beta}_{j,n} \frac{\partial f}{\partial y'} - C, \quad j=N-n, \dots, 0, -1, \dots, -r, \quad (N-r+1 \leq n \leq N-1), \quad (3.4.20b)$$

for $r > 1$,

$$A_{j,n} = \alpha_{j,n} - \beta_{j,n} \frac{\partial f}{\partial y'} - C, \quad j=r, \dots, 1, 0, -1, \dots, r, \quad (r \leq n \leq N-r), \quad (3.4.20c)$$

where C in the above relations (3.4.20a), (3.4.20b), (3.4.20c) is defined as

$$C = \begin{cases} \frac{\partial f}{\partial y}, & \text{for } j=0 \\ 0, & \text{otherwise.} \end{cases}$$

Thus, we can proceed to solve (3.4.19) by Newton's method as follows:

$$\underline{y}^{(s)} = \underline{y}^{(s-1)} - J\phi_n(\underline{y}^{(s-1)})$$

$$\text{or } J\Delta \underline{y}^{(s-1)} = -\phi_n(\underline{y}^{(s-1)}), \quad (3.4.21)$$

where the correction term $\Delta \underline{y}^{(s-1)} = \underline{y}^{(s)} - \underline{y}^{(s-1)}$,
 $\Delta \underline{y} = 0$ at the boundaries,

investigation on the monotone property of $N_h(y)$ (the discrete operator of $N(y)$). If the discretization is of order $2r$ (r was taken up to 4 in his work), then three approaches were suggested to tackle the difficulties which arise at the interior points near the boundaries:

- (i) Non-symmetric formula of order $2r$ or $2r-2$.
- (ii) Reduction in bandwidth near the boundary, which involves using non-symmetric as in (i), a symmetric formulae of order less than $2r$ or linear combinations between such formulae.
- (iii) A matrix polynomial which is based on formulae (3.2.2) or (3.2.3), for example, on 8^{th} -order discretization ($r=4$) the matrix polynomial comes from the second derivative y'' which has the form

$$y'' = \frac{-1}{h^2} \left[\Omega + \frac{1}{12} \Omega^2 + \frac{1}{90} \Omega^3 + \frac{1}{560} \Omega^4 \right], \quad (3.4.22)$$

where

$$\Omega = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ 0 & & \ddots & -1 & 2 \end{bmatrix}$$

In his paper in (Aziz (1975)) there is given an example of using the approach (i) and (ii) above, for example, for a 6^{th} -order discretization (i.e. $r=3$), where the interior points x_1, x_2, \dots, x_{N-2} and x_{N-1} are exempted from using symmetric formulae of order 6, instead he used y''_0, y''_1 and y''_2 chosen as follows:

$$y''_0 = \frac{1}{12h^2} (45y_0 - 154y_1 + 214y_2 - 156y_3 + 61y_4 - 10y_5) + O(h^4), \quad (3.4.23)$$

$$y''_1 = \frac{1}{12h^2} (10y_0 - 15y_1 - 4y_2 + 14y_3 - 6y_4 + y_5) + O(h^4), \quad (3.4.24)$$

and

$$y''_2 = \frac{1}{12h^2} (-y_0 + 16y_1 - 30y_2 + 16y_3 - y_4) + O(h^4), \quad (3.4.25)$$

Thus,

- (i) at the interior point x_1 , the approximate y''_1 is taken as a linear

combination between (3.4.23), (3.4.24) and (3.4.25) in the form

$$y_1'' = \frac{1}{12h^2}(y_0'' + 10y_1'' + y_2'')$$

$$= \frac{-1}{h^2}(-y_0 + 2y_1 - y_2)$$

can be computed from (3.4.17) (and (3.4.18)), i.e.,

i) at x_1 , we have

$$\hat{\alpha}_{3,1} = 0, \quad \hat{\alpha}_{2,1} = -1, \quad \hat{\alpha}_{1,1} = 2, \quad \hat{\alpha}_{0,1} = -1,$$

ii) at x_2 , we have

$$\hat{\alpha}_{3,2} = \frac{1}{12}, \quad \hat{\alpha}_{2,2} = \frac{-16}{12}, \quad \hat{\alpha}_{1,2} = \frac{30}{12}, \quad \hat{\alpha}_{0,2} = \frac{-16}{12}, \quad \hat{\alpha}_{-1,2} = \frac{1}{12},$$

If the discretization of order $2r$ is used for *periodic* b.v.p.'s then a ' $2r+1$ '-point symmetric formulae will be applicable at any grid point. Therefore, if we consider the boundary problem (3.2.1) as periodic which means it is subject to the periodic boundary conditions of the form,

$$\left. \begin{array}{l} y(a) = y(b) \\ y'(a) = y'(b) \end{array} \right\}, \quad (3.4.27)$$

then as a result of this condition (3.4.20c) will be applicable at the points x_n , $n=0,1,\dots,N$, and hence we have (from (3.4.20c))

$$A_{j,k+N} \equiv A_{j,k}, \quad -r \leq j \leq r, \quad k \text{ is any integer}. \quad (3.4.28)$$

Moreover, the linear system (3.4.21) will consist of N equations and J will have the following form by virtue of the coefficients (3.4.28),

82

$$J = N \times N$$

$$(3.4.29)$$

Hence, for the linear 2-point b.v.p. under the periodic conditions (3.4.27) the matrix polynomial (3.4.22) becomes,

$$y'' = \frac{-1}{h} (\hat{\Omega}^2 + \frac{1}{12} \hat{\Omega}^4 + \frac{1}{90} \hat{\Omega}^6 + \frac{1}{560} \hat{\Omega}^8) \quad , \quad (3.4.30)$$

where

$$\hat{\Omega} = \begin{bmatrix} 2 & -1 & & -1 \\ -1 & & 0 & \\ & 0 & & -1 \\ -1 & & -1 & 2 \end{bmatrix}$$

and the matrix (3.4.26) is given as,

$$\frac{1}{180h^2} \begin{bmatrix} 490 & -270 & 27 & -2 & & & -2 & 27 & -270 \\ -270 & & & & & & -2 & 27 & \\ 27 & & & & & & & -2 & \\ -2 & & & & & & & & -2 \\ & & & & & & & & 2 \\ -2 & & & & & & & & 27 \\ 27 & -2 & & & & & & & -270 \\ -270 & 27 & -2 & & & & -2 & 27 & -270 & 490 \end{bmatrix}$$

which infact coincides with (3.4.30) up to Ω^6 .

Another example for the use of non-symmetric formulae is for the linear differential equation of *fourth* order which is associated with beam analysis (Gaw~~ai~~ⁱⁿ and Ball (1977), (1978)) who introduced the so-called 'revised' finite-difference formulae of higher order accuracy appropriate interior points adjacent to the boundaries to replace the conventional finite difference formulae of lower order accuracy.

3.5 FINITE-DIFFERENCE METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS

The previous sections were mainly devoted to using finite-difference schemes for second order differential equations of one independent variable. In this section we shall confine ourselves to differential equations of two independent variables; i.e. partial differential equations, and concentrate on equations of second order.

It is well known that most mathematical models of applied engineering problems are expressed in terms of partial differential equations which may involve more than one independent variable. On the other hand, the analytical solution for these equations, in most cases, is extremely difficult or too cumbersome to be obtained. Thus, numerical methods are found to be an attractive alternative, in particular, at the present time where the use of automatic digital computers are becoming widespread.

Furthermore, the experience of the past years have showed that finite-difference methods are still powerful techniques to obtain a reasonable solution for a wide range of applicable problems involving partial differential equations.

The general form of the second order partial differential equation of two independent variables, x and y (say), and with a dependent variable, u (say), can be expressed in the form:

$$a_1 \frac{\partial^2 u}{\partial x^2} + a_2 \frac{\partial^2 u}{\partial x \partial y} + a_3 \frac{\partial^2 u}{\partial y^2} + a_4 \frac{\partial u}{\partial x} + a_5 \frac{\partial u}{\partial y} + a_6 u + a_7 = 0, \quad (3.5.1)$$

Equation (3.5.1) is said to be:

- (i) Linear if the coefficients a_i , $i=1,2,\dots,7$, are constants or functions of one or both independent variables x and y .
- (ii) Quasi-Linear if the coefficients a_i , $i=1,2,\dots,7$ are functions of the independent variables x and y , or functions of one or both partial derivatives, $\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial y}$.
- (iii) Non-Linear if any of the coefficients, a_i , are functions of the dependent variable, u , or its derivatives.

(iv) Self-Adjoint if the equation (3.5.1) can be replaced by

$$\frac{\partial}{\partial x}(a_1(x)\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(a_3(y)\frac{\partial u}{\partial y}) + a_6 u + a_7 = 0. \quad (3.5.2)$$

(v) Elliptic, if $a_2^2 - 4a_1 a_3 < 0$,

(vi) Parabolic, if $a_2^2 - 4a_1 a_3 = 0$,

and

(vii) Hyperbolic, if $a_2^2 - 4a_1 a_3 > 0$.

Normally, equation (3.5.1) of type (v) occurs in equilibrium (or steady state) problems whilst (vi) and (vii) occur in propagation problems (diffusion and oscillating systems).

Usually, the elliptic problems are classified as of the boundary value type since the boundary conditions are accommodated or given round the (closed) region, whereas the parabolic and hyperbolic equations are classified as initial boundary problems, where the initial conditions are given or/and boundary conditions supplied on the sides of the open region; and the solution proceeds towards the open side.

Further, it is possible for an equation to be elliptic in one domain and hyperbolic in another, e.g. gas flow at high velocities, the flow can be subsonic at some places, supersonic at others (Froberg (1974)).

Common examples for the above cases are:

(1) Hyperbolic - wave equation: $\frac{\partial^2 u}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2}$, c is the propagation velocity

(the vibrating string)

initial condition $0 \leq x \leq 1$ (vibrating string stretched between $x=0$ and $x=1$)

and the boundary condition $u(x,0) = f(x)$

$$\frac{\partial u}{\partial t}(x,0) = g(x).$$

(2) Parabolic - the heat equation: $\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t}$,

initial condition $u(x,0) = f(x)$, $0 \leq x \leq 1$

and the boundary condition $u(0,t) = \phi(t)$

$$u(1,t) = \psi(t)$$

(3) Elliptic - the most common equations are:

$$(i) \quad \text{Laplace equation, } \nabla^2 u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad (3.5.3)$$

$$(ii) \quad \text{Poisson equation, } \nabla^2 u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y), \quad (3.5.4)$$

where ∇^2 is the usual harmonic operator (or Laplacian).

Our consideration will be restricted to elliptic problems, in particular Laplace and Poisson equations, since the suggested algorithms in Chapters 4 and 5 are proposed mainly for these types of problem.

Generally, the classification of elliptic problems lie in three categories according to the boundary condition accommodated at the surface ($\Gamma(R)$) of the closed domain, (three well-posed problems):

(i) Dirichlet's Problem, where the solution u is given on $\Gamma(R)$,

(ii) Neumann's Problem, where the normal derivative $\left(\frac{\partial u}{\partial n}\right)$ is given on $\Gamma(R)$, where n denotes the direction of the outward normal.

(iii) Robin's Problem, where the boundary conditions are of the type

$$\alpha u + \beta \frac{\partial u}{\partial n} \quad \text{on } \Gamma(R),$$

where α and β are given.

In the following we shall consider the application of finite-difference techniques for partial differential equations.

Consider the Dirichlet problem for Poisson equation (3.5.4) which requires to find the solution $u(\equiv u(x,y))$ satisfying (3.5.5a) inside a certain closed domain (R) and is determined on the boundary $(\Gamma(R))$ by the boundary conditions, (3.5.5b),

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y), \quad (x,y) \in R \quad (3.5.5a)$$

$$u(x,y) = g(x,y), \quad (x,y) \in \Gamma(R). \quad (3.5.5b)$$

The strategy of finite-difference methods (as indicated previously for O.D.E.s) are based on mapping the continuous problem to discrete ones and replacing the indirectional derivatives by the appropriate directional ones which they are easier for programming purposes to obtain the approximate solution.

We assume that the region under consideration R and the boundary $\Gamma(R)$ lie in the cartesian plane xOy , Fig.3.5.1, and is subdivided by two groups of straight lines parallel to Ox and Oy . The intersection of these two groups are called the nodal (mesh, net, grid, lattice, or pivotal) points and each point a discrete equation will represent an approximation to the continuous derivative at that point.

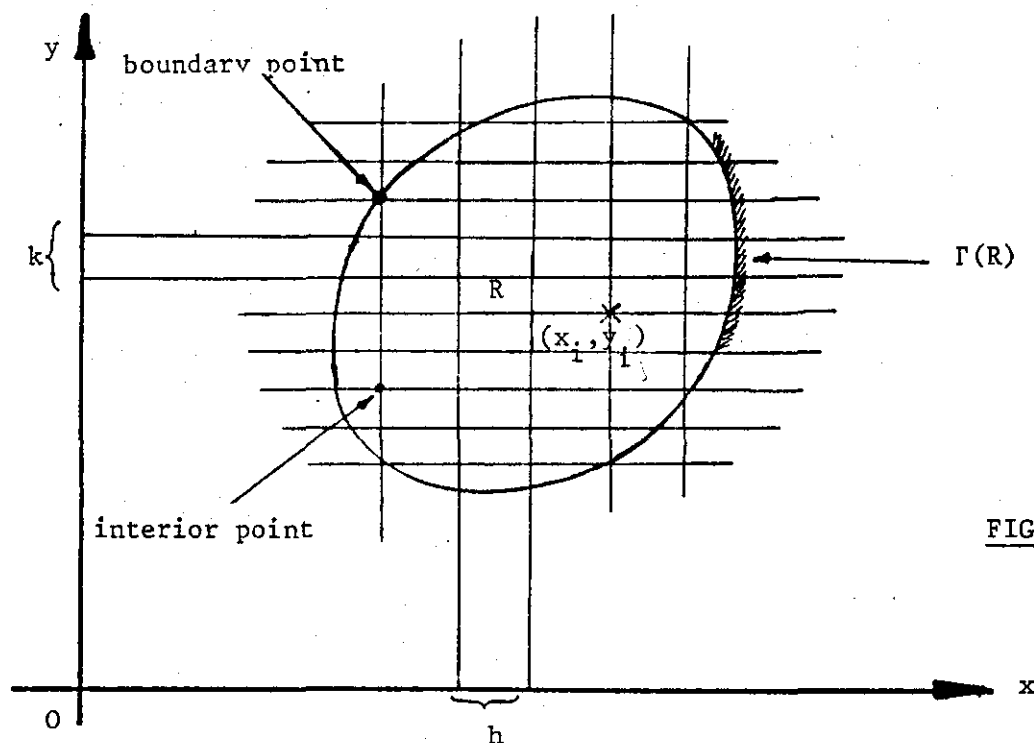


FIGURE 3.5.1

If a uniform mesh (or equally spaced) is adopted in both directions Ox and Oy , and choose h and k to be the net spacing of grid lines in the two direction (see Fig.3.5.1), then we may write

$$\begin{aligned} x_i &= x_0 + ih, & i=0, \pm 1, \pm 2, \dots \\ y_j &= y_0 + jk, & j=0, \pm 1, \pm 2, \dots \end{aligned}$$

Therefore the values of the function $u(x,y)$ are approximated by the points $(x_i, y_j) \equiv (x_0 + ih, y_0 + jk)$. Further, for sake of simplicity the region R can be considered as a square or rectangle; and the grid point (x_0, y_0) coincide with the origin. Also let the two dimensions of R be a and b length units on Ox and Oy respectively, and define the integers N and M such that $Nh=a$, $Mk=b$. Thus, a general approximated form of the derivatives in

equation (3.5.4) (cf. eq. (3.2.3)) can be expressed at the grid point

$(x_i, y_j) \equiv (ih, jk)$ as follows (Fox (1962)):

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} = \frac{1}{h^2}(\delta_x^2 - \frac{1}{12}\delta_x^4 + \frac{1}{90}\delta_x^6 - \dots)u_{i,j}, \quad (3.5.6)$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right)_{i,j} = \frac{1}{k^2}(\delta_y^2 - \frac{1}{12}\delta_y^4 + \frac{1}{90}\delta_y^6 - \dots)u_{i,j}, \quad (3.5.7)$$

where δ_x and δ_y operates on the suffices i and j respectively, i.e.

$$\delta_x u_{i,j} = u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j} \equiv u(x + \frac{h}{2}, y) - u(x - \frac{h}{2}, y)$$

and $\delta_y u_{i,j} = u_{i,j+\frac{1}{2}} - u_{i,j-\frac{1}{2}} \equiv u(x, y + \frac{k}{2}) - u(x, y - \frac{k}{2})$

which obviously, gives

$$\delta_x^2 u_{i,j} = u_{i+1,j} - 2u_{i,j} + u_{i-1,j} \quad (3.5.8)$$

$$\delta_y^2 u_{i,j} = u_{i,j+1} - 2u_{i,j} + u_{i,j-1}. \quad (3.5.9)$$

By ignoring the term involving of order δ greater than 2 in both (3.5.6) and (3.5.7) a simple, *approximated* form of (3.5.4) can be obtained, i.e.

$$\frac{1}{h^2}\delta_x^2 u_{i,j} + \frac{1}{k^2}\delta_y^2 u_{i,j} = f(x_i, y_j) + O(h^2 + k^2), \quad (3.5.10)$$

or by virtue of (3.5.8) and (3.5.9) and assuming $h=k$ (which is commonly used in practice) we have from (3.5.10) the discrete equation:

$$-u_{i+1,j} - u_{i-1,j} + 4u_{i,j} - u_{i,j+1} - u_{i,j-1} = -h^2 f_{i,j} + T_{i,j}, \quad 0 < i < N, \quad 0 < j < M \quad (3.5.11)$$

$$u_{i,j} = g_{i,j} \equiv g(ih, jh), \quad \begin{matrix} i=0, N \text{ for } j=1, \dots, N-1 \\ j=0, N \text{ for } i=1, \dots, N-1, \end{matrix} \quad (3.5.12)$$

where $T_{i,j}$ is the *local truncation error* defined on page 64. The solution $u_{i,j}$ at the point (ih, jh) can be obtained by solving the linear system (3.5.11) (where $T_{i,j}$ and further high order forms are ignored), which compactly can

be written as $A\mathbf{u} = \mathbf{z}$, (3.5.13)

where the vectors are of size $(N-1)(M-1)$,

\mathbf{u} includes the components of unknown $u_{i,j}$, $i=1, \dots, N-1$, $j=1, \dots, M-1$,

and $\mathbf{z} = \mathbf{f} + \mathbf{g}$, \mathbf{f} has the components $f_{i,j}$, $i=1, \dots, N-1$, $j=1, \dots, M-1$,

and \mathbf{g} consists of the values emanating from the boundary condition (3.5.12),

and the matrix A of order $(N-1)(M-1)$ has the tridiagonal *block* form, i.e.

$$A = \begin{bmatrix} B & -I & & & \\ -I & B & -I & & \\ & \ddots & \ddots & \ddots & \\ 0 & & -I & B & -I \\ & & & -I & B \end{bmatrix}, \quad (3.5.14a)$$

(Notice the equations of (3.5.13) are assumed to be ordered row by row from left to right or reversely, or column-wise), where

$$B = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ 0 & & -1 & 4 & -1 \\ & & & -1 & 4 \end{bmatrix}, \quad (3.5.14b)$$

and I is the unit matrix. Both matrices B and I are of order $(N-1)$.

(N.B. for the case of annular regions the derived equations may lead to a similar block matrix A (3.5.14a) but matrix B in (3.5.14b) has an extra element at the off-diagonal top right and bottom left hand corners (see King (1976)).

It is important to notice that the solution obtained from (3.5.13) will have accuracy (in relation to the exact solution) of $O(h^2)$, where the local truncation error, $T_{i,j}$ of equation (3.5.11) has the form:

$$T_{i,j} = \frac{1}{12} h^2 \left\{ \left(\frac{\partial^4 u}{\partial x^4} \right) (\theta, y_j) + \left(\frac{\partial^4 u}{\partial y^4} \right) (x_i, \lambda) \right\} = O(h^2), \quad (3.5.15)$$

where $x_i - h \leq \theta \leq x_i + h$, $y_i - h \leq \lambda \leq y_i + h$,

$$\text{or } T_{i,j} = \frac{1}{6} h^2 \max_{R \cup \Gamma(R)} \left\{ \frac{\partial^4 u}{\partial x^4}, \frac{\partial^4 u}{\partial y^4} \right\}.$$

This last result leads us to observe that whenever h tends to 0, the error term (which is proportional to h^2), $T_{i,j} \rightarrow 0$, and hence the numerical method has the consistency (compatibility) property. However, the concepts of consistency, stability and convergence have been defined in Section 3.3, but the related mathematical theory (including the existence and uniqueness of the solution) will not be discussed for p.d.e.s (relevant references: Smith(1978) Wendroff (1966), Fox (1962)).

The strategy of improving the accuracy in the solution of the discrete equations (3.5.11) can be viewed from the following points:

- (i) as always we can reduce the mesh, h , but this increases the dimension of the system (3.5.13), which becomes too large for storage in the computer memory. This approach includes Richardson (see Smith (1978)),
- (ii) including the terms δ_x^4 and δ_y^4 in (3.5.6) and (3.5.7) but this will incur the penalties of increasing the bandwidth of the matrix but decreasing the size as well as introducing extreme difficulties at the boundaries (Compare Figs. 3.5.2 and 3.5.3).

and

- (iii) the diagonal elements can be included in the approximation of the derivatives to avoid going beyond the boundaries of the region (see the molecule of Fig. 3.5.4). This will not be so accurate as scheme (ii) above but is easier. If the molecule of Fig. 3.5.4 is applied to (3.5.10) we will have a similar system as (3.5.11), but the first term of the r.h.s. will be $-\frac{1}{2}h^2 f_{i,j}$.

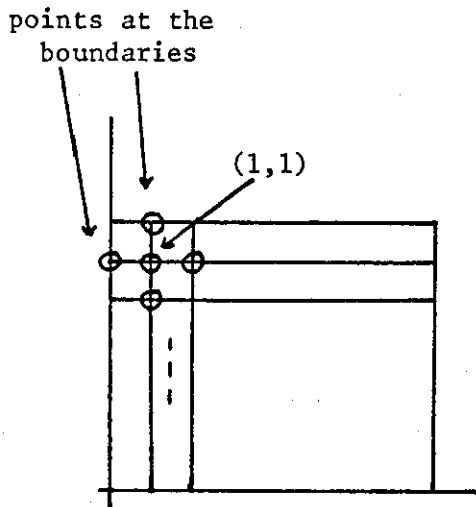


FIGURE 3.5.2: Molecule of using δ_x^2, δ_y^2 and has the form overleaf

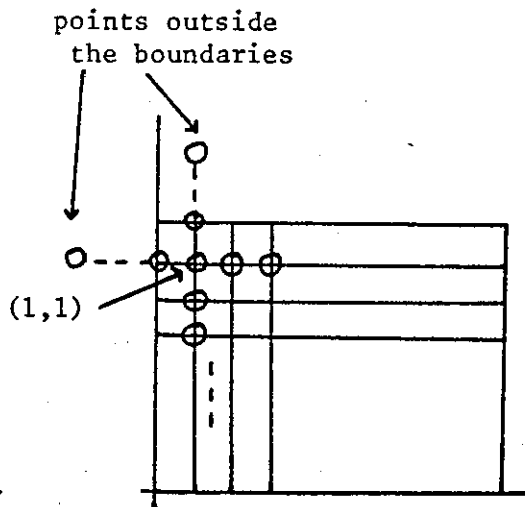


FIGURE 3.5.3: Molecule of using δ_x^4, δ_y^4 and has the form overleaf

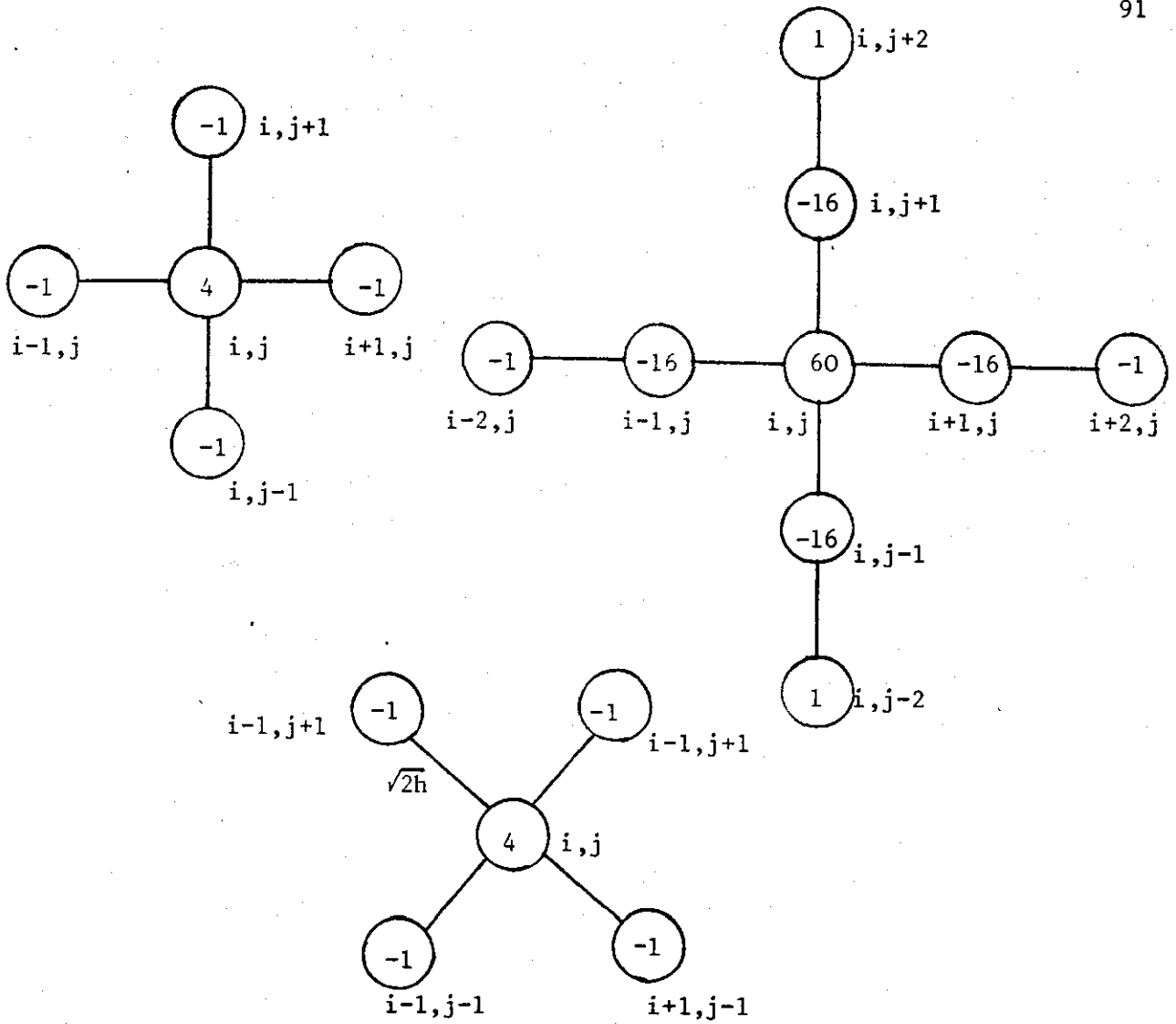


FIGURE 3.5.4: Molecule of using δ_x^2, δ_y^2 (on the diagonal grid)

The deferred correction technique (see Section 3.3) which was based on expressing the local truncation error in terms of differences can be applied for partial differential equations provided that the solution is well-behaved (i.e. the existence and uniqueness of the solution is assumed, and the appropriate functions are sufficiently differentiable). As for ordinary differential equations, we can start with an initial approximate solution, and by difference operations, evaluate the correction terms. Therefore, these corrections can now be inserted in the initial finite-difference equations and the 'new' equation solved on the same mesh for a more accurate solution (Fox (1962), Smith (1978)). For example, for the Laplace equation (3.5.3) which can be written by virtue of (3.5.6) or (3.5.7) (where $h=k$) as,

$$\frac{1}{h^2} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = C(u) \quad (3.5.16)$$

where $C = \frac{1}{h^2} \left[\frac{-1}{12} (\delta_x^4 + \delta_y^4) + \frac{1}{90} (\delta_x^6 + \delta_y^6) \dots \right]$

From (3.5.8) and (3.5.9) the discrete form of equation (3.5.16) is

$$-u_{i+1,j} - u_{i-1,j} + 4u_{i,j} - u_{i,j+1} - u_{i,j-1} + C(u)_{i,j} = 0, \quad (3.5.17)$$

Now we solve (3.5.17) for the initial approximate solution by setting $C(u)_{i,j} = 0$

The next step is to compute $C(u)_{i,j}$, only for the terms $\delta_x^4 u_{i,j}, \delta_y^4 u_{i,j}$ by differencing every line of points in each of the two directions (0x and 0y), and hence (3.5.17) yields an improved solution. The process can be repeated if necessary..

So far, we have illustrated how to solve the Poisson and Laplace equations by using the five-point formula (with its computational molecule shown in page 91), (see eqs.(3.5.11), (3.5.17)). Now, we demonstrate the procedure of deriving a more accurate formula, i.e. the *nine-point* formula (Fox (1962), Smith (1978), Salvadori and Baron (1955)) where the order of the l.t.e. is increased.

We define the following (on the assumption that $h=k$ for both the Poisson and Laplace equations as before),

$$\left. \begin{aligned} D_x &= \frac{\partial}{\partial x} \\ D_y &= \frac{\partial}{\partial y} \\ D_{xy} &= \frac{\partial^2}{\partial x \partial y} \end{aligned} \right\} \quad (3.5.18)$$

and

so that the Laplacian operator $\nabla^2 = D_x^2 + D_y^2$.

By Taylor expansion, we have for example,

$$\begin{aligned} u(x+h) &= \left(u(x) + h \frac{d}{dx} u(x) + \frac{h^2}{2!} \frac{d^2}{dx^2} u(x) + \dots \right) \\ &= \left(1 + \left(h \frac{d}{dx} \right) + \frac{1}{2!} \left(h \frac{d}{dx} \right)^2 + \frac{1}{3!} \left(h \frac{d}{dx} \right)^3 + \dots \right) u(x) \\ &= e^{h \frac{d}{dx}} u(x), \end{aligned}$$

hence, we may write on the basis of the above result

$$\left. \begin{aligned}
 u_{i\pm 1,j} &= e^{\pm h D_x} u_{i,j} \\
 u_{i,j\pm 1} &= e^{\pm h D_y} u_{i,j} \\
 u_{i\pm 1,j\pm 1} &= e^{\pm h(D_x + D_y)} u_{i,j}
 \end{aligned} \right\} \quad (3.5.19a)$$

$$\left. \begin{aligned}
 u_{i\pm 2,j} &= e^{\pm 2h D_x} u_{i,j} \\
 u_{i,j\pm 2} &= e^{\pm 2h D_y} u_{i,j}
 \end{aligned} \right\} \quad (3.5.19b)$$

D_x, D_y in (3.5.19) are defined in (3.5.18).

We define S_1, S_2 and S_3 as follows

$$\left. \begin{aligned}
 S_1 &= u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} \\
 S_2 &= u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1} \\
 S_3 &= u_{i+2,j} + u_{i-2,j} + u_{i,j+2} + u_{i,j-2}
 \end{aligned} \right\} \quad (3.5.20)$$

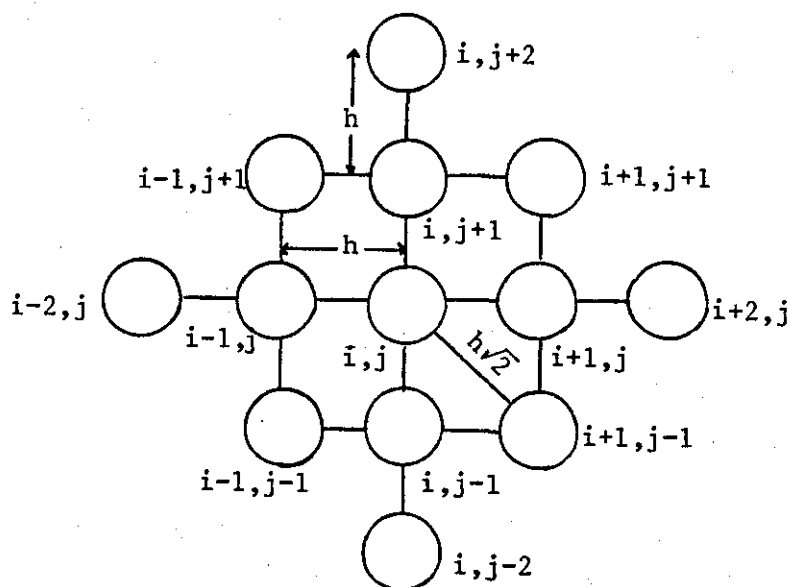


FIGURE 3.5.5

It can be easily observed from Figure 3.5.5 that the points of S_1, S_2 and S_3 in (3.5.20) have the distances $h, \sqrt{2}h$ and $2h$ from the centre point (i,j) .

Therefore, by using (3.5.19), we obtain for

$$\left. \begin{aligned} S_1 &= [4+h^2 \nabla^2 + \frac{h^4}{12}(\nabla^4 - 2D_{xy}^2) + \frac{h^6}{360}(\nabla^6 - 3\nabla^2 D_{xy}^2) + \dots] u_{i,j} , \\ S_2 &= [4+2h^2 \nabla^2 + \frac{h^4}{6}(\nabla^4 + 4D_{xy}^2) + \frac{h^6}{180}(\nabla^6 + 12\nabla^2 D_{xy}^2) + \dots] u_{i,j} , \\ \text{and} \\ S_3 &= [4+4h^2 \nabla^2 + \frac{4}{3} h^4(\nabla^4 - 2D_{xy}^2) + \frac{8}{45} h^6(\nabla^6 - 3\nabla^2 D_{xy}^2) + \dots] u_{i,j} , \end{aligned} \right\} \quad (3.5.21)$$

(where D_{xy} is defined in (3.5.18)).

By eliminating the term (D_{xy}^2) between S_1 and S_2 , we obtain the following nine-point formula:

$$\nabla^2 u_{i,j} = \frac{1}{6h^2}(4S_1 + S_2 - 20u_{i,j}) - \frac{1}{12} h^2 \nabla^4 u_{i,j} + T_p, \quad (3.5.22)$$

where $\nabla^4 u = \nabla^2(\nabla^2 u) = \nabla^2 f$ and T_p refers to the local truncation error for the Poisson equation, which is of $O(h^4)$. Further, for the Laplace equation $\nabla^2 u = 0$, the term T_p vanishes, instead we have

$$\nabla^2 u_{i,j} = (4S_1 + S_2 - 20u_{i,j}) + T_L, \quad (3.5.23)$$

where T_L is now of $O(h^6)$.

Symbolically, the Poisson equation is exhibited for:

(i) the five-point formula, in the form,

$$\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix} u = -h^2 f$$

and

(ii) the nine-point formula in the form

$$\begin{bmatrix} -1 & -4 & -1 \\ -4 & 20 & -4 \\ -1 & -4 & -1 \end{bmatrix} u = -6h^2 f - \frac{1}{2} h^4 \nabla^2 f.$$

Other combinations between (3.5.21) yield different formulae, for example, $S_3 - 16S_1$ gives for the Laplace equation as (in symbolic form):

$$\nabla^2 u = \frac{1}{12h^2} \begin{bmatrix} & & 1 & & \\ & & -16 & & \\ 1 & -16 & 60 & -16 & 1 \\ & & -16 & & \\ & & 1 & & \end{bmatrix} u_{i,j} + O(h^6).$$

For the Biharmonic equation which is a more complicated partial differential equation of elliptic type and has the form:

$$\begin{aligned} \nabla^4 u &= \nabla^2 \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) u = 0 \\ &= \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) u \end{aligned}$$

or in symbolic notation,

$$\begin{aligned} \nabla^4 u &= \frac{1}{h^4} \begin{bmatrix} & & -1 & & \\ & & 4 & & \\ -1 & 4 & -1 & & \\ & & -1 & & \end{bmatrix} \begin{bmatrix} & & -1 & & \\ & & 4 & & \\ -1 & 4 & -1 & & \\ & & -1 & & \end{bmatrix} u_{i,j} = 0 \\ &= \frac{1}{h^4} \begin{bmatrix} & & 1 & & \\ & & 2 & -8 & 2 \\ 1 & -8 & 20 & -8 & 1 \\ & & 2 & -8 & 2 \\ & & 1 & & \end{bmatrix} u_{i,j} = 0 \end{aligned}$$

Finally, we consider the Poisson's equation (3.5.5) under periodic boundary conditions on a square (or rectangle) plane such as encountered in Plasma problems (see Hockney (1965)). (Evans (1979) also considered (3.5.5) in a square region with periodic conditions). The effect of the periodic boundary conditions can be regarded as equivalent to the solution being periodically repeated in both directions Ox and Oy (or merely in one direction for some cases (Wood (1971))). By rewriting (3.5.11) (and suppressing the term T) we have

$$-u_{i+1,j} - u_{i-1,j} + 4u_{i,j} - u_{i,j+1} - u_{i,j-1} = -h^2 f_{i,j}, \quad i,j=1,2,\dots,\hat{N}, \quad (3.5.24)$$

where the region is assumed to be a unit square covered by a square grid of size $h=1/\hat{N}$. Then the periodic boundary conditions can be presented in the form,

$$\begin{aligned} u_{i+k\hat{N},j+k\hat{N}} &= u_{i,j} \\ f_{i+k\hat{N},i+k\hat{N}} &= f_{i,j} \end{aligned}, \quad (3.5.25)$$

where the k is any integer and the indices are to be interpreted modulo \hat{N} .

Consequently, equations (3.5.24) with boundary conditions (3.5.25) can be written compactly in the form

$$\underline{A}\underline{u} = \underline{d}, \quad (3.5.26)$$

where the matrix A is $(\hat{N}^2 \times \hat{N}^2)$ and has the form

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

and

$$B = \begin{bmatrix} 4 & -1 & & & -1 \\ -1 & 4 & -1 & & 0 \\ & & & & \\ & 0 & & -1 & 4 & -1 \\ -1 & & & -1 & 4 \end{bmatrix}_{(\hat{N} \times \hat{N})}$$

However, the system (3.5.26) has not got a unique solution (Berman and Plemmons (1979)) since the matrix A is singular.

CHAPTER 4

NEW ALGORITHMIC METHODS FOR THE SOLUTION

OF BANDED MATRIX EQUATIONS

4.1 INTRODUCTION

In this chapter, we shall consider several algorithmic methods for the solution of the linear system of equations which stem from the discretized mathematical physics problems via the application of finite-difference procedures. We are, as it was pointed out earlier, mainly concerned with the type of linear system where the matrix is banded (and may be sparse) and has constant elements, as in the algorithm of Section 4.2, for non-constant elements, as in the algorithm of Section 4.3.

The algorithms described here involve the factorization of the coefficient matrix into inverted semi-banded matrices, and hence the solution is obtained by forward and backward algorithmic processes.

Two kinds of factorization schemes are adopted in this thesis, i.e.

- (1) Factorization in Invertible Cyclic Matrices (FICM),
- (2) Factorization in pseudo-Invertible Rectangular Matrices (FIRM).

The FICM and FIRM algorithms are associated with periodic and non-periodic banded matrices respectively. Different variants of both methods will be included in the following sections. Convergence, stability and rounding-error analysis will be discussed for some of these methods. The extension of the FICM and FIRM algorithms for tri-diagonal and quindagonal block matrices will also be given in special cases (see Chapter 5).

The numerical applications for these algorithms will be dealt with in later chapters. This chapter will deal only with the theoretical derivation and analysis of the algorithms.

4.2.1 ALGORITHM FICM1

The matrix factorization procedure described in this algorithm is for use in the fast numerical solution of constant banded symmetric linear systems. For such a special linear system, we will show that it can be solved efficiently by the factorization of the coefficient matrix into 2 easily inverted matrices and the solution obtained by forward and backward substitution schemes.

We consider a general real linear system of the form

$$\underline{Ax} = \underline{z} \quad (4.2.1)$$

where A is a matrix of bandwidth $(2r+1)$, $r \geq 1$ of order N ($N \geq 2r+1$)

and has the (general) form:

We now follow Evans (1973) and consider the factorization of A so that (4.2.1) can be solved by simple forward and backward substitution processes in the manner, i.e.,

$$\text{and } \left. \begin{array}{l} Q\underline{y} = \underline{z} \\ Q^T \underline{x} = \underline{y} \end{array} \right\} \quad (4.2.2)$$

where $QQ^T = A$ (4.2.3a)

and Q is defined in (4.2.3b).

In this subsection we are interested only in (4.2.3a), that is, a general factorization of A into Invertible Cyclic Matrices Q and Q^T , where the solution of the systems (4.2.2) are discussed in subsection 4.2.3.

Given that the matrix Q has the general form:

$$Q = \begin{bmatrix} \alpha_0 & \cdots & \alpha_{r-1} & \alpha_r & 0 \\ & \ddots & & & \alpha_r \\ & & \ddots & & \alpha_{r-1} \\ \alpha_r & 0 & & & \\ & \ddots & & & \\ & & \alpha_1 & \cdots & \alpha_r \\ & & & & \alpha_0 \end{bmatrix}, \quad (4.2.3b)$$

then by virtue of equation (4.2.3a), if we multiply Q by Q^T , and equate corresponding elements to A, we obtain the following non-linear system of equations to solve for the unknowns $\alpha_0, \alpha_1, \dots, \alpha_r$,

$$\left. \begin{aligned} \alpha_0^2 + \alpha_1^2 + \dots + \alpha_{r-1}^2 + \alpha_r^2 &= c_0 \\ \alpha_0 \alpha_1 + \alpha_1 \alpha_2 + \dots + \alpha_{r-2} \alpha_{r-1} + \alpha_{r-1} \alpha_r &= c_1 \\ \alpha_0 \alpha_2 + \alpha_1 \alpha_3 + \dots + \alpha_{r-3} \alpha_{r-1} + \alpha_{r-2} \alpha_r &= c_2 \\ \dots & \\ \alpha_0 \alpha_r &= c_r \end{aligned} \right\}, \quad (4.2.4a)$$

or in abbreviated form (4.2.4a) can be written as

$$\sum_{i=k}^r \alpha_i \alpha_{i-k} = c_k, \quad k=0,1,\dots,r \quad (4.2.4b)$$

which has to be solved to determine the $(r+1)$ unknowns $\alpha_0, \alpha_1, \dots, \alpha_r$.

4.2.2 ITERATIVE METHOD OF SOLUTION (GITRM)

To solve the non-linear system (4.2.4) we choose a generalized iterative method, and, for reasons of algebraic simplicity, we define the quantities λ_1, λ_2 as follows:

$$\lambda_1 = c_0 + 2 \sum_{i=1}^r c_i \quad (4.2.5)$$

$$\lambda_2 = c_0 + 2 \left(\sum_{i=1}^s c_{2i} - \sum_{i=1}^t c_{2i-1} \right) \quad (4.2.6)$$

where t and s are defined as:

$$\begin{aligned} t &= \begin{cases} r/2 & - r \text{ even} \\ (r+1)/2 & - r \text{ odd} \end{cases} \\ \text{and } s &= \begin{cases} r/2 & - r \text{ even} \\ (r-1)/2 & - r \text{ odd.} \end{cases} \end{aligned} \quad (4.2.7)$$

Now, the GITRM method requires the following necessary conditions for the equations (4.2.4) to possess real roots,

$$\lambda_1 > 0 \quad (4.2.8a)$$

$$\text{and } \lambda_2 > 0. \quad (4.2.8b)$$

In fact, the conditions (4.2.8) do not require the matrix A to be diagonally-dominant, except for A being tridiagonal, i.e. the case where $r=1$ (see Evans (1973)). We clarify this point further by introducing two examples of banded matrices:

Example 1: $r=2$,

$$A = \begin{bmatrix} 7 & -4 & 1 & & & & 1 & -4 \\ -4 & 7 & -4 & 1 & & & & 1 \\ 1 & & & & & & 0 & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ 1 & & 0 & & & & & \\ -4 & 1 & & & & & & \\ & & & & & & 1 & -4 & 7 \end{bmatrix}, \quad (4.2.9a)$$

Example 2: $r=3$

$$A_2 = \begin{bmatrix} 21 & -15 & 6 & -1 & & & -1 & 6 & -15 \\ -15 & 21 & -15 & 6 & -1 & & -1 & 6 & \\ 6 & -15 & 21 & -15 & 6 & -1 & & & \\ -1 & 6 & -15 & 21 & -15 & 6 & -1 & & \\ & & & & & & & 0 & \\ & & & & & & & & -1 \\ & & & & & & & & 6 \\ & & & & & & & & -15 \\ -15 & 6 & -1 & & & & -1 & 6 & -15 & 21 \end{bmatrix}, \quad (4.2.9b)$$

where both matrices can be derived from the 2nd order periodic boundary value problem (see Chapter 3). In the examples (4.2.9), λ_1 and λ_2 can easily be obtained from (4.2.5) and (4.2.6) respectively. Thus, we have, for A_1 , $\lambda_1=1$, $\lambda_2=17$ and for A_2 , $\lambda_1=1$, $\lambda_2=65$ and hence in both examples λ_1 and λ_2 satisfy conditions (4.2.8), although the matrices A_1 and A_2 are not diagonally dominant.

However, the only restrictions we impose upon the matrix A are (i) that it must possess positive diagonal elements and (ii) that in each row the diagonal element is the largest one amongst the other elements, i.e.

$$c_0 > 0 \text{ and } c_0 > |c_i|, \quad i=1,2,\dots,r, \quad (4.2.10)$$

noting that A is symmetric and possesses constant elements.

We proceed now to solve the non-linear system (4.2.4). Our major interest in this respect is to modify the first equation of (4.2.4) to a more simpler form. This modification can be accomplished as follows.

If we multiply each equation (from the 2nd onwards) of (4.2.4) by 2 and add to the first equation, then the first equation becomes

$$\begin{aligned}
& (\alpha_0^2 + \alpha_1^2 + \dots + \alpha_r^2) + 2(\alpha_0\alpha_1 + \alpha_1\alpha_2 + \dots + \alpha_{r-1}\alpha_r) + 2(\alpha_0\alpha_2 + \alpha_1\alpha_3 + \dots + \alpha_{r-2}\alpha_r) \\
& + \dots + 2(\alpha_0\alpha_r) = c_0 + 2c_1 + 2c_2 + \dots + 2c_r .
\end{aligned} \tag{4.2.11}$$

In fact, the L.H.S. of (4.2.11) is the expansion of the expression $(\alpha_0 + \alpha_1 + \dots + \alpha_r)^2$, i.e.

$$\begin{aligned}
(\alpha_0 + \alpha_1 + \dots + \alpha_r)^2 &= (\alpha_0^2 + \alpha_1^2 + \dots + \alpha_r^2) + 2(\alpha_0\alpha_1 + \dots + \alpha_{r-1}\alpha_r) \\
&+ 2(\alpha_0\alpha_2 + \dots + \alpha_{r-2}\alpha_r) + \dots + 2(\alpha_0\alpha_r) .
\end{aligned} \tag{4.2.12}$$

Thus from (4.2.12) the equation (4.2.11) can be rewritten in the form,

$$(\alpha_0 + \alpha_1 + \alpha_2 + \dots + \alpha_r)^2 = c_0 + 2c_1 + 2c_2 + \dots + 2c_r , \tag{4.2.13a}$$

or, by the assumption (4.2.5), we can replace the R.H.S. by λ_1 , i.e.

(4.2.13a) becomes,

$$(\alpha_0 + \alpha_1 + \alpha_2 + \dots + \alpha_r)^2 = \lambda_1 . \tag{4.2.13b}$$

Since λ_1 is assumed to be a positive quantity, by condition (4.2.8a), then taking the square root of both sides of the equation (4.2.13), we have

$$\alpha_0 + \alpha_1 + \alpha_2 + \dots + \alpha_r = \pm \lambda_1^{\frac{1}{2}} .$$

At present we will consider the positive square root of λ_1 . In this case, the final form of the (modified) first equation of (4.2.4) is

$$\alpha_0 + \alpha_1 + \alpha_2 + \dots + \alpha_r = \lambda_1^{\frac{1}{2}} . \tag{4.2.14}$$

Now, taking equation (4.2.14) with the 2nd equation and subsequent equations of (4.2.4) we can formulate a system whose solution is that of (4.2.4). This system is

$$\left. \begin{aligned} \alpha_0^{\alpha_1 + \alpha_2 + \dots + \alpha_r} &= \tilde{c}_0 \\ \alpha_0^{\alpha_1 + \alpha_2 + \dots + \alpha_{r-1}} \alpha_r &= c_1 \\ \alpha_0^{\alpha_2 + \alpha_3 + \dots + \alpha_{r-2}} \alpha_{r-1} \alpha_r &= c_2 \\ \vdots \\ \alpha_0^{\alpha_r} &= c_r \end{aligned} \right\}, \quad (4.2.15a)$$

or in abbreviated form,

$$\left. \begin{aligned} \sum_{i=0}^r \alpha_i &= \tilde{c}_0 \\ \sum_{i=k}^r \alpha_i \alpha_{i-k} &= c_k, \quad k=1, 2, \dots, r \end{aligned} \right\}, \quad (4.2.15b)$$

where $\tilde{c}_0 = \lambda_1^{\frac{1}{2}}$, and hence the first modification of the system (4.2.4) is completed.

Now, the second modification involves replacing the 2nd equation of (4.2.15) by a new equation in simpler form similar to the first equation in the system (4.2.15). This can be done as follows.

If we square the first equation of (4.2.15), and multiply the 2nd equation, 4th equation, ..., (r+1)th equation, if r is odd (or the rth equation, if r is even) of the same system by (-4) and add together the new second equation of (4.2.15) will have the form,

(i) for r odd,

$$\begin{aligned} &(\alpha_0^{\alpha_1 + \alpha_2 + \dots + \alpha_r})^2 - 4(\alpha_0^{\alpha_1 + \alpha_2 + \dots + \alpha_{r-1}} \alpha_r) - 4(\alpha_0^{\alpha_3 + \alpha_4 + \dots + \alpha_{r-3}} \alpha_{r-2} \alpha_{r-1} \alpha_r) - \dots - 4(\alpha_0^{\alpha_r}) \\ &= (\tilde{c}_0)^2 - 4c_1 - 4c_3 - \dots - 4c_r \\ &\equiv \lambda_1^2 - 4c_1 - 4c_3 - \dots - 4c_r \\ &(\tilde{c}_0 \text{ is defined in (4.2.15)}) \\ &= (c_0 + 2c_1 + 2c_2 + \dots + 2c_r) - 4c_1 - 4c_3 - \dots - 4c_r \\ &(\lambda_1 \text{ given by (4.2.5)}) \end{aligned}$$

$$= c_0 - 2c_1 + 2c_2 - 2c_3 + 2c_4 - \dots + 2c_{r-1} - 2c_r \quad (4.2.16)$$

The left-hand side of (4.2.16) can be simplified further by expanding the term $(\alpha_0 + \dots + \alpha_r)^2$ which is given by (4.2.12), and then, by clearing up terms, we arrive at the result,

$$\begin{aligned} \text{L.H.S.} &\equiv (\alpha_0^2 + \alpha_1^2 + \dots + \alpha_r^2) - 2(\alpha_0\alpha_1 + \dots + \alpha_{r-1}\alpha_r) + 2(\alpha_0\alpha_2 + \dots + \alpha_{r-2}\alpha_r) \\ &\quad - 2(\alpha_0\alpha_3 + \dots + \alpha_{r-3}\alpha_r) + \dots - 2\alpha_0\alpha_r \equiv (\alpha_0 - \alpha_1 + \alpha_2 - \alpha_3 + \dots - \alpha_r)^2 \end{aligned} \quad (4.2.17)$$

Thus, replacing the L.H.S. of equation (4.2.16) by its equivalent in (4.2.17), we obtain,

$$(\alpha_0 - \alpha_1 + \alpha_2 - \alpha_3 + \dots - \alpha_r)^2 = c_0 - 2c_1 + 2c_2 - 2c_3 + \dots - 2c_r \quad (4.2.18)$$

(ii) for r even,

By following a similar procedure to the above, we can obtain the following result,

$$(\alpha_0 - \alpha_1 + \alpha_2 - \alpha_3 + \dots - \alpha_{r-1} + \alpha_r)^2 = c_0 - 2c_1 + 2c_2 - 2c_3 + \dots - 2c_{r-1} + 2c_r, \quad (4.2.19)$$

Equations (4.2.18) and (4.2.19) can be combined to be written in the form,

$$(\alpha_0 - \alpha_1 + \alpha_2 - \dots + (-1)^r \alpha_r)^2 = c_0 - 2c_1 + 2c_3 - \dots + (-1)^r 2c_r,$$

or by using the integers t and s, given by (4.2.7), this equation, can also be written in the form,

$$\begin{aligned} [(\alpha_0 + \alpha_2 + \dots + \alpha_{2s}) - (\alpha_1 + \alpha_3 + \dots + \alpha_{2t-1})]^2 &= c_0 + 2(c_2 + c_4 + \dots + c_{2s}) \\ &\quad - 2(c_1 + c_3 + \dots + c_{2t-1}). \end{aligned} \quad (4.2.20)$$

Since the right-hand side of (4.2.20) equals λ_2 , given by (4.2.6), and λ_2 is positive by the condition (4.2.8b), then by

taking the square root of both sides of the last equation (again we consider only the positive square root case), we have

$$\alpha_0^{\alpha_2+\dots+\alpha_{2s}} - (\alpha_1^{\alpha_3+\dots+\alpha_{2t-1}}) = \lambda_2^{\frac{1}{2}}. \quad (4.2.21)$$

Thus, the second modified system can be obtained from (4.2.15) by replacing its 2nd equation by the equation (4.2.21), i.e.

$$\left. \begin{aligned} \alpha_0^{\alpha_1+\alpha_2+\dots+\alpha_r} &= \tilde{c}_0 (= \lambda_1^{\frac{1}{2}}) \\ \alpha_0^{\alpha_2+\dots+\alpha_{2s}} - \alpha_1^{\alpha_3+\dots+\alpha_{2t-1}} &= \lambda_2^{\frac{1}{2}} \\ \alpha_0^{\alpha_2+\alpha_1\alpha_3+\dots+\alpha_{r-2}\alpha_r} &= c_2 \\ \dots & \\ \alpha_0^{\alpha_2} &= c_r \end{aligned} \right\} \quad (4.2.22)$$

Finally, by adding, then subtracting the first two equations, we obtain

$$\alpha_0^{\alpha_2+\dots+\alpha_{2s}} = \hat{c}_0$$

and

$$\alpha_1^{\alpha_3+\dots+\alpha_{2t-1}} = \hat{c}_1,$$

with s and t defined as in (4.2.7) and $\hat{c}_0 = \frac{1}{2}(\lambda_1^{\frac{1}{2}} + \lambda_2^{\frac{1}{2}})$, $\hat{c}_1 = \frac{1}{2}(\lambda_1^{\frac{1}{2}} - \lambda_2^{\frac{1}{2}})$.

Thus, our final form of the system, having the same solution as (4.2.4) and (4.2.15) is:

$$\left. \begin{aligned} \alpha_0^{\alpha_2+\dots+\alpha_{2s}} &= \hat{c}_0 \\ \alpha_1^{\alpha_3+\dots+\alpha_{2t-1}} &= \hat{c}_1 \\ \alpha_0^{\alpha_2+\dots+\alpha_{r-2}\alpha_r} &= c_2 \\ \dots & \\ \alpha_1^{\alpha_r} &= c_r \end{aligned} \right\} \quad (4.2.23a)$$

or in abbreviated form

$$\left. \begin{aligned} \sum_{i=0}^s \alpha_{2i} &= \hat{c}_0 \\ \sum_{i=1}^t \alpha_{2i-1} &= \hat{c}_1 \\ \sum_{i=k}^r \alpha_i \alpha_{i-k} &= c_k, \quad k=2,3,\dots,r. \end{aligned} \right\} \quad (4.2.23b)$$

Now, an iterative solution scheme for the non-linear system

(4.2.15) can be written as

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ & \alpha_0^{(n-1)} & \alpha_1^{(n-1)} & \cdots & \alpha_{r-1}^{(n-1)} \\ & & \alpha_0^{(n-1)} & \cdots & \alpha_{r-2}^{(n-1)} \\ & & & \ddots & \\ & & & & \alpha_0^{(n-1)} \\ & 0 & & & \alpha_0^{(n-1)} \end{bmatrix} \begin{bmatrix} \alpha_0^{(n)} \\ \alpha_1^{(n)} \\ \alpha_2^{(n)} \\ \vdots \\ \alpha_r^{(n)} \end{bmatrix} = \begin{bmatrix} \tilde{c}_0 \\ c_1 \\ c_2 \\ \vdots \\ c_3 \end{bmatrix} \quad (4.2.24)$$

and for the non-linear system (4.2.23), as

$$\begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & p \\ & 1 & 0 & 1 & \cdots & q \\ & & \alpha_0^{(n-1)} & \alpha_1^{(n-1)} & \cdots & \alpha_{r-2}^{(n-1)} \\ & & & \ddots & & \\ & & & & & \alpha_0^{(n-1)} \\ & 0 & & & & \alpha_0^{(n-1)} \end{bmatrix} \begin{bmatrix} \alpha_0^{(n)} \\ \alpha_1^{(n)} \\ \alpha_2^{(n)} \\ \vdots \\ \alpha_r^{(n)} \end{bmatrix} = \begin{bmatrix} \hat{c}_0 \\ \hat{c}_1 \\ c_2 \\ \vdots \\ c_r \end{bmatrix} \quad (4.2.25)$$

where n denotes the number of iterations, i.e. $n \geq 1$ and the initial vector $\underline{\alpha}^{(0)} = [\alpha_0^{(0)}, \alpha_1^{(0)}, \dots, \alpha_r^{(0)}]^T$ is an arbitrary starting solution

with $\alpha_0 \neq 0$, and

$$p = \begin{cases} 0 & - r \text{ odd} \\ 1 & - r \text{ even} \end{cases}$$

with

$$q = 1-p.$$

Since the systems (4.2.24) and (4.2.25) are upper triangular in form, then $\alpha_0, \dots, \alpha_r$ of the system (4.2.4) can be computed by a simple back substitution process for each iteration, i.e. at step n , we have:

$$\left. \begin{aligned}
 \alpha_r^{(n)} &= c_r / \alpha_0^{(n-1)} \\
 \alpha_j^{(n)} &= (c_j - \sum_{i=j+1}^r \alpha_{i-j}^{(n-1)} \alpha_i^{(n)}) / \alpha_0^{(n-1)}, \quad j=r-1, \dots, 1 \\
 \alpha_0^{(n)} &= \tilde{c}_0 - \sum_{j=1}^r \alpha_j^{(n)},
 \end{aligned} \right\} \quad (4.2.26)$$

for (4.2.24) and

$$\left. \begin{aligned}
 \alpha_r^{(n)} &= c_r / \alpha_0^{(n-1)} \\
 \alpha_j^{(n)} &= (c_j - \sum_{i=j+1}^r \alpha_{i-j}^{(n-1)} \alpha_i^{(n)}) / \alpha_0^{(n-1)}, \quad j=r-1, \dots, 2 \\
 \alpha_1^{(n)} &= \hat{c}_1 - \sum_{j=2}^t \alpha_{2j-1}^{(n)} \quad (\text{provided } t \geq 2, \text{ otherwise the } 2^{\text{nd}} \text{ term} \equiv 0) \\
 \alpha_0^{(n)} &= \hat{c}_0 - \sum_{j=1}^s \alpha_{2j}^{(n)} \quad (\text{provided } s \geq 1, \text{ otherwise the } 2^{\text{nd}} \text{ term} \equiv 0)
 \end{aligned} \right\} \quad (4.2.27)$$

for (4.2.25).

The iterative processes (4.2.26) and (4.2.27) are terminated when the solution vector has converged to the required tolerance. Hence the elements of matrix Q (and Q^T) can be determined.

We multiply the $(r+j)^{\text{th}}$ equation by $\frac{-\alpha_{k+j}}{\alpha_0}$, $j=1,2,\dots,r-k$, and add to the $(r-k)^{\text{th}}$ equation to obtain a new $(r-k)^{\text{th}}$ equation, where $k=1,2,\dots,r-1$.

The above elimination strategy can be formulated as follows:

Let

$$\hat{f}_{i,j} = \begin{cases} \alpha_{i-j} & \text{for } j \leq i \\ 0 & \text{otherwise} \end{cases} \quad \begin{matrix} i=1,2,\dots,r, \\ j=1,2,\dots,r-1 \end{matrix} \quad (4.2.29)$$

$$f_{i,N+j} = \begin{cases} -\alpha_0 & \text{for } j=i \\ 0 & \text{otherwise} \end{cases} \quad i,j=1,2,\dots,r \quad (4.2.30)$$

and

$$m_i = -\alpha_i / \alpha_0.$$

We now form the quantities,

$$\left. \begin{aligned} f_{k,j} &= \sum_{i=1}^r m_i f_{k,j+i} + \delta, \quad \delta = \begin{cases} \alpha_0, & k=j \\ 0, & \text{otherwise} \end{cases} \quad \begin{matrix} k=1(1)r \\ j=N,N-1, \\ \dots, r \end{matrix} \\ e_j &= z_j + \sum_{k=1}^r m_k e_{j+k}, \quad \text{where } e_t \equiv 0 \text{ for all } t > N \end{aligned} \right\} \quad (4.2.31a)$$

and

$$\left. \begin{aligned} f_{k,r-i} &= \sum_{j=1}^{r-i} m_{i+j} f_{k,r+j} + \hat{f}_{k,r-i}, \quad k=1(1)r \\ \text{where the } \hat{f}'\text{'s are given by (4.2.29),} & \quad i=1,2,\dots,r-1 \\ e_{r-i} &= z_{r-i} + \sum_{j=1}^{r-i} m_{i+j} e_{r+j}. \end{aligned} \right\} \quad (4.2.31b)$$

Thus, the given system (4.2.28) now has the form,

$$\begin{bmatrix} f_{11} & f_{21} & \dots & f_{r,1} \\ \vdots & f_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & f_{r,r} \\ \vdots & \vdots & \vdots & \vdots \\ f_{1,N} & f_{2,N} & \dots & f_{1,N} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} \quad (4.2.32)$$

Now, we proceed to eliminate the coefficients $f_{i,j}$ for all $i > j$, $j=1, \dots, r-1$ as follows,

for $i=1, 2, \dots, r-1$,

let $\hat{k}=r+1-i$,

$$\left. \begin{aligned} R_{i,k} &= -f_{k,k}^{\wedge} / f_{k,k}^{\wedge}, \text{ then we have} \\ f_{j,k}^{(i)} &= f_{j,k}^{(i-1)} + R_{i,k} f_{j,k}^{(i-1)}, \quad j=1, \dots, r-i \\ \text{and } e_k^{(i)} &= e_k^{(i-1)} + R_{i,k} e_k^{(i-1)} \end{aligned} \right\} k=1, 2, \dots, r-i, \quad (4.2.33)$$

where the superscript refers to the i^{th} stage of the elimination process and $f_{\ell,v}^{(0)} \equiv f_{\ell,v}$ and $e_s^{(0)} \equiv e_s$ are as given in (4.2.32),

Thus, having completed the elimination (4.2.33) the lower triangular matrix in (4.2.32) is obtained and immediately the auxiliary solution vector \underline{y} is given by,

$$\left. \begin{aligned} y_1 &= e_1 / f_{11}, \\ y_2 &= (e_2 - y_1 f_{12}) / f_{22} \\ &\vdots \\ y_r &= (e_r - \sum_{i=1}^{r-1} y_i f_{i,r}) / f_{r,r} \\ \text{and } y_j &= (e_j - \sum_{i=1}^r y_i f_{i,j}) / \alpha_0, \quad j=r+1, \dots, N \end{aligned} \right\} \quad (4.2.34)$$

where $f_{i,j}$ for $i \leq j$ and e_j , $j=1, \dots, r-1$ are given by (4.2.33) while the remaining f 's are as located in (4.2.32).

(N.B. in practice one can replace the denominator in the R.H.S. of (4.2.34) by

$$\gamma_i = \left\{ \begin{array}{ll} f_{i,j} & , \text{ for } j \leq r \\ \alpha_0 & , \text{ otherwise} \end{array} \right\} \quad j=1, 2, \dots, N.$$

Then the final relation of (4.2.21) can be used to evaluate y_1, \dots, y_r as well, provided that \underline{y} must be CLEARED in the computer store, i.e. the y_j 's set to zero).

A similar solution process is also carried out on the companion system

$$Q^T \underline{x} = \underline{y} \quad (4.2.35)$$

which will have the final form:

$$\begin{bmatrix} \alpha_0 & f_{r,N} & \cdots & f_{2,N} & f_{1,N} \\ & 0 & & & \\ & & \ddots & & \\ & & & f_{r,r} & \\ & & & & \ddots \\ & & & & & f_{2,2} \\ & & & & & & f_{1,1} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_N \end{bmatrix} \quad (4.2.36)$$

where f 's are as given in (4.2.34), and η_1, \dots, η_N are defined as follows:

$$\text{Let } \hat{\eta}_k = \eta_{N-k+1}, \quad \hat{y}_k = y_{N-k+1}, \quad k=1,2,\dots,N$$

then, $\hat{\underline{\eta}}$ can be simply evaluated as \underline{e} in (4.2.31), i.e.,

$$\hat{\eta}_j = \hat{y}_j + \sum_{k=1}^r m_k \hat{\eta}_{j+k}, \quad \text{where } \hat{\eta}_t = 0 \text{ for all } t > N, \quad j=N, N-1, \dots, r$$

$$\text{and } \hat{\eta}_{r-i} = \hat{y}_{r-i} + \sum_{j=1}^{r-i} m_{i+j} \hat{\eta}_{r+j}, \quad i=1,2,\dots,r-1.$$

Also due to the elimination procedure presented in (4.2.33) the quantities $\hat{\eta}_1, \dots, \hat{\eta}_{r-1}$ will be modified at the i^{th} step, $i=1,2,\dots,r$ exactly as e_1, \dots, e_{r-1} . Notice that we retain the notation of the modified elements to avoid further complication.

Hence, the solution is given by the back substitution process derived from (4.2.36), i.e.

$$\left. \begin{aligned} x_N &= \eta_N / f_{11} \\ x_{N-1} &= (\eta_{N-1} - x_N f_{12}) / f_{22} \\ &\vdots \\ x_{N-r+1} &= (\eta_{N-r+1} - x_N f_{1,r} - \dots - x_{N-r+2} f_{r-1,r}) / f_{r,r} \\ \text{and } x_j &= (\eta_j - \sum_{i=r}^1 x_{N+i-r} f_{r+1-i,N+1-j}) / \alpha_0, \quad j=N-r, N-r-1, \dots, 1. \end{aligned} \right\} \quad (4.2.37)$$

Again, we point out that the last relation of (4.2.37) can be used to evaluate $x_N, x_{N-1}, \dots, x_{N-r+1}$ which has been indicated previously.

Finally the computational complexity of the algorithm for the solution of (4.2.1) involves approximately $O(5Nr)$ additions and multiplications (where divisions are assumed to have roughly the same consuming time as multiplications) together with the predetermination of the α 's.

4.2.4 A POLYNOMIAL SCHEME FOR THE SOLUTION OF THE MODIFIED NON-LINEAR SYSTEM

The strategy used is to form the polynomial, $p(\alpha)$ of any one of the α 's, say α_r , and derive the smallest zero by the Newton-Raphson method.

Let us consider the following cases for the modified non-linear system (4.2.23).

- (i) For $r=1$, the system (4.2.23) will consist of two linear equations only. These equations are:

$$\left. \begin{aligned} \alpha_0 &= \hat{c}_0 \\ \alpha_1 &= \hat{c}_1 \end{aligned} \right\} \quad (4.2.38)$$

where \hat{c}_0 and \hat{c}_1 are defined earlier, and in this special case are given by

$$\left. \begin{aligned} \hat{c}_0 &= \frac{1}{2}(\lambda_1^{\frac{1}{2}} + \lambda_2^{\frac{1}{2}}) = \frac{1}{2}((c_0 + 2c_1)^{\frac{1}{2}} - (c_0 - 2c_1)^{\frac{1}{2}}) \\ \text{and } \hat{c}_1 &= \frac{1}{2}(\lambda_1^{\frac{1}{2}} - \lambda_2^{\frac{1}{2}}) = \frac{1}{2}((c_0 + 2c_1)^{\frac{1}{2}} - (c_0 - 2c_1)^{\frac{1}{2}}) \end{aligned} \right\} \quad (4.2.39)$$

The unknowns α_0, α_1 are obtained immediately by (4.2.38) and hence we conclude that, for the system (4.2.1) being periodic tridiagonal, (i.e. $r=1$) the iterative solution process is not required, i.e. the procedure (4.2.25) (or (4.2.27)) is not applicable for this special case.

- (ii) For $r=2$, the system (4.2.23) consists of the following equations:

$$\left. \begin{aligned} \alpha_0 + \alpha_2 &= \hat{c}_0 \\ \alpha_1 &= \hat{c}_1 \\ \alpha_0 \alpha_2 &= c_2 \end{aligned} \right\} \quad (4.2.40)$$

The polynomial in α_2 can be easily formulated from the first and the last equations of the system (4.2.40) and has the quadratic form

$$p(\alpha_2) \equiv \alpha_1^2 - \hat{c}_0 \alpha_2 + c_2 = 0, \quad (4.2.41)$$

from which we can easily evaluate α_2 and consequently α_0 ; with $\alpha_1 = \hat{c}_1$.

(iii) For $r=3$, the equations of the system (4.2.23) are

$$\left. \begin{aligned} \alpha_0 + \alpha_2 &= \hat{c}_0 \\ \alpha_1 + \alpha_3 &= \hat{c}_1 \\ \alpha_0 \alpha_2 + \alpha_1 \alpha_3 &= c_2 \\ \alpha_0 \alpha_3 &= c_3 \end{aligned} \right\} \quad (4.2.42)$$

The polynomial in α_3 can be derived from (4.2.42) and expressed in the form of the quartic equation,

$$p(\alpha_3) \equiv \alpha_3^4 - \hat{c}_1 \alpha_3^3 + c_2 \alpha_3^2 + \hat{c}_0 c_3 \alpha_3 + c_3^2 = 0. \quad (4.2.43)$$

Further it can be easily seen that $p(\alpha_4)$ will be a polynomial of order 8. Therefore, in general we can say that

$$P(\alpha_r) \equiv O(2^{r-1}), \quad r > 1.$$

Now, as r increases, the polynomial $p(\alpha_r)$ becomes increasingly more difficult to formulate and to solve and therefore alternative solution methods have to be relied upon.

Remark 4.1

After the completion of the work in Audish and Evans (1980) describing the iterative method (GITRM, Subsection 4.2.2), of solving the non-linear system (4.2.4), a direct method for a special case of the system (4.2.4) was presented by Berg (1981), (see Appendix A).

4.2.5 STABILITY OF THE METHOD

The method involves, as a first step, solving the non-linear set of equations derived from the factorization of the coefficient matrix, and then, as a final step, solving the two linear systems of equations to produce the solution of the given system (4.2.1). Hence the stability of the method will clearly depend upon both steps. Thus, the investigation of the stability can be categorized into two parts:

(A) The Stability of the Iterative Method (GITRM)

We have seen that equation (4.2.13b) yields two values of λ_1 , i.e., $\pm\lambda_1^{\frac{1}{2}}$, and equation (4.2.20) yields two values of λ_2 , i.e., $\pm\lambda_2^{\frac{1}{2}}$. The constant values \tilde{c}_0 of the system (4.2.15) and \hat{c}_0 and \hat{c}_1 of the system (4.2.23) are defined, \tilde{c}_0 in terms of λ_1 and \hat{c}_0 and \hat{c}_1 in terms of λ_1, λ_2 , and the positive square roots of λ_1, λ_2 were taken. Therefore, by also considering their negative square roots, we have the following possible cases tabulated below:

<u>Case 100</u>	$\tilde{c}_0 = \lambda_1^{\frac{1}{2}},$
<u>Case 101</u>	$\tilde{c}_0 = -\lambda_1^{\frac{1}{2}},$
<u>Case 110</u>	$\hat{c}_0 = \frac{1}{2}(\lambda_1^{\frac{1}{2}} + \lambda_2^{\frac{1}{2}}), \quad \hat{c}_1 = \frac{1}{2}(\lambda_1^{\frac{1}{2}} - \lambda_2^{\frac{1}{2}}),$
<u>Case 111</u>	$\hat{c}_0 = \frac{1}{2}(\lambda_1^{\frac{1}{2}} - \lambda_2^{\frac{1}{2}}), \quad \hat{c}_1 = \frac{1}{2}(\lambda_1^{\frac{1}{2}} + \lambda_2^{\frac{1}{2}}),$
<u>Case 112</u>	$\hat{c}_0 = \frac{1}{2}(-\lambda_1^{\frac{1}{2}} + \lambda_2^{\frac{1}{2}}), \quad \hat{c}_1 = -\frac{1}{2}(\lambda_1^{\frac{1}{2}} + \lambda_2^{\frac{1}{2}}),$
and <u>Case 113</u>	$\hat{c}_0 = -\frac{1}{2}(\lambda_1^{\frac{1}{2}} + \lambda_2^{\frac{1}{2}}), \quad \hat{c}_1 = \frac{1}{2}(-\lambda_1^{\frac{1}{2}} + \lambda_2^{\frac{1}{2}}).$

In fact, the cases 100 and 101 indicate that the modification procedure discussed in Subsection 4.2.2 leads to the formulation of two non-linear systems instead of one (i.e. the system (4.2.15)). Also, the cases 110 to 113 lead to the construction of the system (4.2.23)

and of three other systems related to the cases 111 to 113. The iterative method GITRM was not applicable to the systems associated with cases 111 and 112 but was applicable in the other cases. We now clarify this point further.

The cases 100 and 101 lead to values of the α 's which are equal but opposite in sign; similarly for the cases 110 and 113. Whilst, for the cases 111 and 112, the results showed that convergence was not always possible. In addition, the condition of the diagonal element (α_0) being the largest in modulus was not satisfied by cases 111 and 112, whilst for the remaining cases, the conditions,

$$|\alpha_i| < |\alpha_0|, \text{ for } i=1,2,\dots,r, \quad (4.2.44)$$

was satisfied. Noting that condition (4.2.44) is similar to condition (4.2.10) which was imposed on the elements c_0, c_1, \dots, c_r of the matrix A.

Let us now consider the convergent case 110.

- (1) We set $r=1$. Then from the equations (4.2.36) we have the ratio

$$\alpha_1 : \alpha_0 = \hat{c}_1 : \hat{c}_0.$$

Now, if we follow Evans (1973) and substitute \hat{c}_0 and \hat{c}_1 in terms of λ_1, λ_2 , and then in terms of c_0, c_1 and c_2 as they are given in (4.2.39) we have the result,

$$\frac{\alpha_1}{\alpha_0} = \frac{\lambda_1^{\frac{1}{2}} - \lambda_2^{\frac{1}{2}}}{\lambda_1^{\frac{1}{2}} + \lambda_2^{\frac{1}{2}}}$$

or
$$\frac{\alpha_1}{\alpha_0} = \frac{\lambda_1 - \lambda_2}{\lambda_1 + 2(\lambda_1 \lambda_2)^{\frac{1}{2}} + \lambda_2}$$

and by using (4.2.37),

$$= \frac{4c_1}{2c_0 + 2(c_0^2 - 4c_1^2)^{\frac{1}{2}}}.$$

By putting $\alpha_0=1$, $\alpha_1=-\alpha$, we obtain the result

$$\alpha = \frac{-2c_1}{c_0 + (c_0^2 - 4c_1^2)^{\frac{1}{2}}}$$

which coincides with the value given by Evans (1973).

- (2) Now set $r=2$. The three equations obtained from the non-linear system (4.2.23), i.e. the equations (4.2.40), are the same as those given by Evans and Hadjidimos (1978). However, they obtained α_0, α_2 (and α_1) by solving the equation (4.2.41), and 8 triples were obtained (see Evans and Hadjidimos (1978)). In fact, these triples can be easily obtained since the quadratic equation (4.2.41) yields 2 roots and the constant \hat{c}_0 of this equation has 4 possible values given by the cases 110 to 113. Two of these triples lead to the optimal solutions which coincide with cases 110 and 113 (or cases 100 and 101).
- (3) Finally, we set $r=3$. A Newton-Raphson iterative technique was used to obtain the smallest root from the quartic equation (4.2.43). The values of the α 's, i.e., the solution of the system (4.2.42) which were obtained agreed exactly with those evaluated by the iterative method (GITRM) given in subsection 4.2.2.

(B) The Stability of the Solvable Linear Systems (4.2.2)

We consider an example with the coefficient matrix A in the system (4.2.1) being periodic quindagonal, and choose A to be A_1 which is defined in (4.2.9a). Hence the matrix Q defined by (4.2.3b) has the

form,

$$Q = \begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & & 0 \\ & \alpha_2 & \alpha_0 & \alpha_1 & \\ & & \alpha_1 & \alpha_2 & \alpha_0 \\ & & & \alpha_2 & \alpha_0 & \alpha_1 \\ & & & & \alpha_1 & \alpha_2 & \alpha_0 \end{bmatrix} \quad (4.2.45)$$

The elements c_0, c_1 and c_2 of the matrix A , for this special case, are:

$$c_0 = 7, c_1 = -4, c_2 = 1$$

and hence the quantities \hat{c}_0 and \hat{c}_1 of the system (4.2.40) can be evaluated, i.e., $\hat{c}_0 = \frac{1}{2}(1+\sqrt{17})$, $\hat{c}_1 = \frac{1}{2}(1-\sqrt{17})$ since $\lambda_1=1$, $\lambda_2=17$. Therefore, the elements of the matrix Q in (4.2.45) can be determined by solving the system (4.2.40). On the other hand, if we consider one of the non-convergent cases in part (A), say case 111, then \hat{c}_0 and \hat{c}_1 become,

$$\hat{c}_0 = \frac{1}{2}(\lambda_1^{\frac{1}{2}} - \lambda_2^{\frac{1}{2}}) = \frac{1}{2}(1 - \sqrt{17})$$

and

$$\hat{c}_1 = \frac{1}{2}(\lambda_1^{\frac{1}{2}} + \lambda_2^{\frac{1}{2}}) = \frac{1}{2}(1 + \sqrt{17}) .$$

By substituting these values in the system (4.2.40) and solving for α_0, α_1 and α_2 we obtain the results:

$$\alpha_1 = \frac{1}{2}(1 + \sqrt{17})$$

$$\alpha_0 = \frac{1 - \sqrt{17}}{4}(1 \pm \sqrt{2}), \quad \alpha_2 = \frac{1 - \sqrt{17}}{4}(1 \pm \sqrt{2}).$$

Subsequently it can be easily verified that $|\alpha_0| < \alpha_1$ and hence the matrix Q of (4.2.45) does not possess a diagonal of largest magnitude (i.e. the condition (4.2.44) is not fulfilled). This is the case which we for two reasons exclude from our consideration of this method: (i) the convergence, as pointed out in part (A) was not assured and hence the determination of the elements of the factorized matrix are unattainable, and (ii) even if the polynomial or other schemes are employed to calculate the values of the α 's such that the inequalities (4.2.44) are not satisfied, then the elimination process of solving the systems (4.2.2), described in subsection (4.2.3), cannot guarantee to obtain the solution to the desired accuracy due to the influence of the growth of rounding errors.

However, in the solution of the two systems of (4.2.2), we have used an elimination *without* pivoting on the assumption that the conditions (4.2.44) are fulfilled. Wilkinson (1961) has shown that the Gaussian elimination is stable against the growth of rounding error if the diagonal element is the largest element (in modulus) in each row. Since, in our case, we stipulate the conditions (4.2.44) then the algorithm is STABLE.

4.2.6 CONVERGENCE OF THE NON-LINEAR SYSTEM

The general form of the non-linear systems (4.2.15) or (4.2.23) can be written compactly as

$$G\alpha = \underline{d} \quad (4.2.46)$$

where matrix G of order $(r+1)$ and vector \underline{d} of $(r+1)$ components (r as defined earlier) are defined as follows:

either

$$G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ & \alpha_0 & \alpha_1 & \cdots & \alpha_{r-1} \\ & & \alpha_0 & \alpha_1 & \cdots & \alpha_{r-2} \\ & & & \ddots & \ddots & \ddots \\ & 0 & & & \alpha_1 & \alpha_0 \end{bmatrix}, \quad d = \begin{bmatrix} \tilde{c}_0 \\ c_1 \\ c_2 \\ \vdots \\ c_r \end{bmatrix} \quad (4.2.47)$$

for the case of the system (4.2.15),

or

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & p \\ & 1 & 0 & 1 & \cdots & q \\ & & \alpha_0 & \alpha_1 & \cdots & \alpha_{r-2} \\ & & & \alpha_0 & \cdots & \vdots \\ & & & & \ddots & \alpha_1 \\ & 0 & & & & 0 \end{bmatrix}, \quad d = \begin{bmatrix} \hat{c}_0 \\ \hat{c}_1 \\ c_2 \\ \vdots \\ c_r \end{bmatrix} \quad (4.2.48)$$

for the case of the system (4.2.23) (p and q as defined in (4.2.25)).

To investigate the convergence of the iterative schemes (4.2.24) or (4.2.25) let us consider a splitting of the matrix G , such that

$$G = D + U \quad (4.2.49)$$

where D is the diagonal matrix comprising the diagonal elements of G ,

and U is a strictly upper triangular matrix containing the elements of G .

Then, when (4.2.49) is substituted into (4.2.46), we have

$$(D+U)\underline{\alpha} = \underline{d},$$

which, after premultiplication by D^{-1} in both sides and re-arrangement, becomes,

$$\underline{\alpha} = D^{-1}\underline{d} + (-D^{-1}U)\underline{\alpha}. \quad (4.2.50)$$

Now, we introduce the vector function

$$\underline{\phi}(\underline{\alpha}) = [\phi_0(\underline{\alpha}), \phi_1(\underline{\alpha}), \dots, \phi_r(\underline{\alpha})]^T$$

and suppose that

$$\underline{\alpha} = \underline{\phi}(\underline{\alpha}). \quad (4.2.51)$$

Therefore, from (4.2.50) and (4.2.51), we obtain

$$\underline{\phi}(\underline{\alpha}) = D^{-1}\underline{d} + (-D^{-1}U)\underline{\alpha}$$

which involves $(r+1)$ non-linear equations, of the form

(i) for the case (4.2.47) being considered

$$\left. \begin{aligned} \phi_r &= \frac{1}{\alpha_0} [c_r] \\ \phi_j &= \frac{1}{\alpha_0} [c_j - \sum_{i=j+1}^r \alpha_{i-j} \alpha_i], \quad j=r-1, \dots, 1 \\ \phi_0 &= \tilde{c}_0 - \sum_{i=1}^r \alpha_i \end{aligned} \right\} \quad (4.2.52)$$

(ii) for the case (4.2.48) being considered

$$\left. \begin{aligned} \phi_r &= \frac{1}{\alpha_0} [c_r] \\ \phi_j &= \frac{1}{\alpha_0} [c_j - \sum_{i=j+1}^r \alpha_{i-j} \alpha_i], \quad j=r-1, \dots, 2 \\ \phi_1 &= \hat{c}_1 - \sum_{j=2}^t \phi_{2j-1} \quad (\text{provided } t \geq 2, \text{ otherwise the 2}^{\text{nd}} \text{ term} \equiv 0) \\ \phi_0 &= \hat{c}_0 - \sum_{j=1}^s \phi_{2j} \end{aligned} \right\} \quad (4.2.53)$$

Now, the iterative form for (4.2.51) can be written

(a) for the scheme (4.2.24), as,

$$\left. \begin{aligned} \alpha_r^{(n)} &= \phi_r(\alpha_0^{(n-1)}) \\ \alpha_k^{(n)} &= \phi_k(\alpha_r^{(n)}, \alpha_{r-1}^{(n)}, \dots, \alpha_{k+1}^{(n)}, \alpha_{r-k}^{(n-1)}, \dots, \alpha_0^{(n-1)}), \\ &\quad k=r-1, \dots, 1 \\ \alpha_0^{(n)} &= \phi_0(\alpha_r^{(n)}, \alpha_{r-1}^{(n)}, \dots, \alpha_1^{(n)}) \end{aligned} \right\} \quad (4.2.54)$$

and

(b) for the scheme (4.2.25) as,

$$\left. \begin{aligned} \alpha_r^{(n)} &= \phi_r(\alpha_0^{(n-1)}) \\ \alpha_k^{(n)} &= \phi_k(\alpha_r^{(n)}, \alpha_{r-1}^{(n)}, \dots, \alpha_{k+1}^{(n)}, \alpha_{r-k}^{(n-1)}, \dots, \alpha_0^{(n-1)}), \\ &\quad k=r-1, \dots, 2 \\ \alpha_1^{(n)} &= \phi_1(\alpha_t^{(n)}, \dots, \alpha_5^{(n)}, \alpha_3^{(n)}) \\ \alpha_0^{(n)} &= \phi_0(\alpha_s^{(n)}, \dots, \alpha_4^{(n)}, \alpha_2^{(n)}) \end{aligned} \right\} \quad (4.2.55)$$

It can be readily seen that (4.2.54) and (4.2.55) are Seidel-type iteration methods (Szidarovszky and Yakowitz (1978)). The convergence criteria to these two non-linear systems of equations can be shown below. The following analysis is based on the Theorem 2.3.2 and Corollary 2.3.1 of Chapter 2.

From the system (4.2.52) and (4.2.53) it can be shown that the partial derivatives $\frac{\partial \phi_i(\alpha)}{\partial \alpha_j}$, $0 \leq i, j \leq r$ exist. Moreover, if we consider the system (4.2.52), for instance, and differentiate w.r.t. α_0 , then we obtain the result,

$$\left. \begin{aligned}
 \frac{\partial \phi_r}{\partial \alpha_0} &= \frac{-1}{\alpha_0^2} [c_r] \\
 \frac{\partial \phi_j}{\partial \alpha_0} &= \frac{-1}{\alpha_0^2} [c_j - \sum_{i=j+1}^r \alpha_{i-j} \alpha_i], \quad j=r-1, \dots, 1 \\
 \frac{\partial \phi_0}{\partial \alpha_0} &= \sum_{i=1}^r \frac{\partial \phi_i}{\partial \alpha_0}
 \end{aligned} \right\} \quad (4.2.56)$$

and

By substituting c_r, \dots, c_1 from (4.2.15) then the system (4.2.56) becomes,

$$\begin{aligned}
 \frac{\partial \phi_r}{\partial \alpha_0} &= \frac{-1}{\alpha_0} [\alpha_r] \\
 \frac{\partial \phi_j}{\partial \alpha_0} &= \frac{-1}{\alpha_0} [\alpha_j], \quad j=r-1, \dots, 1 \\
 \text{and} \quad \frac{\partial \phi_0}{\partial \alpha_0} &= \frac{1}{\alpha_0} \sum_{i=1}^r \alpha_i.
 \end{aligned}$$

Hence, we obtain

$$\sum_{i=1}^r \left| \frac{\partial \phi_i(\alpha)}{\partial \alpha_0} \right| = 2 \sum_{i=1}^r \left| \frac{\alpha_i}{\alpha_0} \right| \equiv \mu_0. \quad (4.2.57a)$$

Similarly, we obtain μ_1, \dots, μ_r such that

$$\sum_{i=1}^r \left| \frac{\partial \phi_i(\alpha)}{\partial \alpha_k} \right| = 2 \sum_{\substack{i=2 \\ i \neq k}}^r \left| \frac{\alpha_i}{\alpha_0} \right| \equiv \mu_k, \quad k=1, 2, \dots, r \quad (4.2.57b)$$

By applying a similar differentiation procedure to the system (4.2.53), we obtain

$$\hat{\mu}_0 = 2 \sum_{i=2}^r \left| \frac{\alpha_i}{\alpha_0} \right|, \quad (4.2.58a)$$

$$\left. \begin{aligned}
 \hat{\mu}_1 &= 2 \sum_{i=3}^r \left| \frac{\alpha_i}{\alpha_0} \right|, \\
 &\vdots \\
 \mu_k &= 2 \sum_{\substack{i=1 \\ i \neq k-1, k+1}}^r \left| \frac{\alpha_i}{\alpha_0} \right|, \quad k=2, 3, \dots, r-1, \\
 &\vdots \\
 \mu_r &= 2 \sum_{i=1}^{r-2} \left| \frac{\alpha_i}{\alpha_0} \right|
 \end{aligned} \right\} \quad (4.2.58b)$$

and

It can be observed that, from (4.2.58a) and (4.2.57a) we have the result

$$\mu_0 = \hat{\mu}_0 + 2 \left| \frac{\alpha_1}{\alpha_0} \right|$$

or $\hat{\mu} < \mu_0$. (4.2.59)

Now, the sufficient condition for the convergence of the systems (4.2.54) or (4.2.55) (cf. (4.2.26) or (4.2.27)) can be obtained by applying Theorem 2.3.2 or Corollary 2.3.1 to give the result,

$$\mu_k < 1 \quad (4.2.60)$$

and $\hat{\mu}_k < 1$. (4.2.61)

More precisely, from (4.2.57) and (4.2.58) it can be easily shown that $\mu_0 = \max_i(\mu_i)$ or $\hat{\mu}_0 = \max_i(\hat{\mu}_i)$. Therefore, the final form for the condition required by the appropriate theorem is

$$0 < (\hat{\mu}_0) < \mu_0 < 1, \quad (4.2.62)$$

where the bracketed term is placed by virtue of the relation (4.2.59).

This condition for convergence was tested numerically on the results presented in Chapter 6 (Sec. 6.2). The values obtained for μ_0 are tabulated for various values of r (i.e. matrices of bandwidth $2r+1$), and presented in Table 4.1. For the initial case $r=1$, it can be seen that only weak convergence can be proved.

Bandwidth (2r+1)	Value of $\frac{1}{2}\mu_0$ at convergence (obtained from the relation (4.2.57a))
3	5.00×10^{-1}
5	4.17×10^{-1}
7	3.96×10^{-1}
9	3.80×10^{-1}
11	3.79×10^{-1}

TABLE 4.1

4.2.7 ERROR ANALYSIS FOR THE LINEAR SYSTEMS INVOLVED IN THE ALGORITHM FICM1

It is known that, because computers cannot perform exact arithmetic, any numerical process involving a matrix system generally produces an approximate solution. Thus, to ensure the stability of the solution obtained by such a process, it is worthwhile to assess the bounds of the rounding errors which grow during the course of its implementation.

We also need to point out that the algorithm FICM1 involves the factorization of a given matrix into two matrices, Q and its transpose Q^T , whose elements are determined iteratively. Therefore, the rounding error analysis will apply equally to either of the two systems in (4.2.2), i.e., the bounds of rounding errors which affect the solution will be formulated in terms of the elements of Q (or Q^T).

Initially, we shall assume that in binary floating point computer arithmetic, each number, say x , is represented internally in the form,

$$x = a \cdot 2^t, \quad \frac{1}{2} \leq |a| \leq 1,$$

where a is the mantissa, and t the exponent which is bounded by the binary word lengths of the given machine registers. Following Wilkinson (1963), the computed result of multiplying together two floating point numbers x and y will be designated by $fl(x,y)$. Then, in general the conventional exact mathematical relationships can be expressed as follows,

$$\left. \begin{aligned} fl(x \pm y) &= (x \pm y)(1 + \epsilon_1) , \\ fl(xy) &= (xy)(1 + \epsilon_2) , \\ \text{and } fl(x/y) &= x/y(1 + \epsilon_3) . \end{aligned} \right\} \quad (4.2.63)$$

Each ϵ_i in (4.2.63) refers to the rounding error associated with the respective arithmetic operation and is some value of ϵ such that $|\epsilon| = 2^{-t}$, where t is the number of binary digits allocated to the mantissa of the floating point number in the computer.

Now, we consider one of the two systems in (4.2.2) (since either matrix

$$B^{(k)} \underline{x} = \underline{y}^{(k)}, \quad (4.2.67)$$

where $k=1, \dots, N$; ($k=1$ refers to the original system).

Then, after $(N-1)$ transformations, i.e. at $k=N$, the system (4.2.67) yields an upper triangular matrix (see Chapter 2) of the form

$$U \equiv B^{(N)} = \begin{bmatrix} b_{1,1}^{(1)} & b_{1,N-r+1}^{(1)} & \cdots & b_{1,N}^{(1)} \\ & b_{2,N-r+1}^{(2)} & \cdots & b_{2,N}^{(2)} \\ & & \ddots & \\ & & & b_{N-r,N-r}^{(N-r)} & \cdots & b_{N-r,N}^{(N-r)} \\ & & & b_{N-r+1,N-r+1}^{(N-r+1)} & \cdots & \\ & & & & \ddots & \\ & & & & & b_{N,N}^{(N)} \end{bmatrix} \quad (4.2.68a)$$

$\longleftarrow r \longrightarrow$

Let a lower triangular matrix, say L , be defined as follows:

$$L = \begin{bmatrix} 1 & & & & \\ m_{2,1} & 1 & & & \\ m_{3,1} & m_{3,2} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ m_{r+1,1} & & & & \\ m_{r+2,2} & & & & \\ \vdots & & & & \\ 0 & & & & \\ & & & m_{N,N-r} & \cdots & m_{N,N-1} & 1 \end{bmatrix} \quad (4.2.68b)$$

where $m_{i,k} = b_{i,k}^{(k)} / b_{k,k}^{(k)}$, $k=1, \dots, N-1$, and b 's are given by (4.2.66a).

Thus it is correct to write (see Chapter 2),

$$LU = B^{(1)} + \delta B, \quad (4.2.69)$$

where the matrices L, U are given in (4.2.68), and δB is the perturbation in B whose bounds we are interested in assessing.

It has been shown in Chapter 2 that at the k^{th} step, $k > 1$, i.e. when $B^{(k)}$ of the system (4.2.64) is obtained, the element b_{kj} , $j = k, \dots, N$ is modified and $b_{i,k-1}$, $i = k, \dots, N$ is eliminated. Details of the formulation of the appropriate equations of the whole elimination procedure are given in Wilkinson (1963). Here we merely introduce the final (general) form of the modified and eliminated elements which can be expressed as follows.

- (1) For $i \leq j$, $i = 2, \dots, N$ when $B^{(i)}$ is obtained, we have the modified elements,

$$b_{i,j}^{(i)} = b_{i,j}^{(1)} - m_{i,1} b_{1,j}^{(1)} - m_{i,2} b_{2,j}^{(2)} - m_{i,3} b_{3,j}^{(3)} - \dots - m_{i,i-1} b_{i-1,j}^{(i-1)} + e_{i,j}, \quad (4.2.70)$$

where the form $e_{i,j}$ refers to the summation of the rounding errors which occur during the elimination process from the 2nd step up to the i^{th} step, i.e. if we let $\epsilon_{i,j}^{(k)}$ be the error of the k^{th} step, then we may express $e_{i,j}$ in the form

$$e_{i,j} = \epsilon_{i,j}^{(2)} + \epsilon_{i,j}^{(3)} + \dots + \epsilon_{i,j}^{(i)}. \quad (4.2.71)$$

- (2) For $i > j$, $j = 1, \dots, N-1$, when $B^{(j+1)}$ is obtained, we have

$$0 \equiv b_{i,j}^{(1)} - m_{i,1} b_{1,j}^{(1)} - m_{i,2} b_{2,j}^{(2)} - m_{i,3} b_{3,j}^{(3)} - \dots - m_{i,j} b_{j,j}^{(j)} + e_{i,j}, \quad (4.2.72)$$

Here, the error term is expressed as

$$e_{i,j} = \epsilon_{i,j}^{(2)} + \epsilon_{i,j}^{(3)} + \dots + \epsilon_{i,j}^{(j+1)}. \quad (4.2.73)$$

Notice that in both relations (4.2.70) and (4.2.72) the values of the multipliers (m 's) correspond to those given by (4.2.68b).

Furthermore, since B possesses zero elements (given by (4.2.66b)) which remain zeros, except in the case of those above the diagonal in the columns, $N, N-1, \dots, N-r+1$, then the corresponding error terms $e_{i,j}$ equal zero and $e_{i,i} = 0$ as well, i.e.,

for the relationship (4.2.71), (where $i \leq j$)

$$e_{i,j} = 0, \quad j=i, i+1, \dots, N-r, \quad i=1, 2, \dots, N-r, \quad (4.2.74a)$$

and for the relationship (4.2.73), (where $i > j$)

$$e_{i,j} = 0, \quad i=j+r-1, \dots, N, \quad j=1, 2, \dots, N-r-1. \quad (4.2.74b)$$

Now, the computed element $b_{i,j}^{(k)}$ is defined in floating point using (4.2.63) as follows,

$$\begin{aligned} b_{i,j}^{(k)} &= fl(b_{i,j}^{(k-1)} - m_{i,k-1} b_{k-1,j}^{(k-1)}) \\ &= [b_{i,j}^{(k-1)} - m_{i,k-1} b_{k-1,j}^{(k-1)} (1 + \epsilon_1)] (1 + \epsilon_2) \\ &= (b_{i,j}^{(k-1)} - m_{i,k-1} b_{k-1,j}^{(k-1)} \epsilon_1) (1 + \epsilon_2). \end{aligned} \quad (4.2.75)$$

Therefore, each $\epsilon_{i,j}$ in (4.2.61) and (4.2.73) can be expressed as

$$\begin{aligned} \epsilon_{i,j}^{(k)} &= b_{i,j}^{(k)} - (b_{i,j}^{(k-1)} - m_{i,k-1} b_{k-1,j}^{(k-1)}) \\ &= b_{i,j}^{(k)} - \left(\frac{b_{i,j}^{(k)}}{1 + \epsilon_2} + m_{i,k-1} b_{k-1,j}^{(k-1)} \epsilon_1 \right), \quad (\text{by (4.2.75)}) \\ &= \frac{\epsilon_2}{1 + \epsilon_2} b_{i,j}^{(k)} - m_{i,k-1} b_{k-1,j}^{(k-1)} \epsilon_1. \end{aligned}$$

It can be easily verified that the modulus of $\epsilon_{i,j}$ satisfies the inequality given below, i.e.,

$$\begin{aligned} |\epsilon_{i,j}^{(k)}| &\leq \frac{|\epsilon_2|}{1 - |\epsilon_2|} |b_{i,j}^{(k)}| + |m_{i,k-1}| |b_{k-1,j}^{(k-1)}| |\epsilon_1| \\ &\leq \frac{1}{1 - |\epsilon|} (|b_{i,j}^{(k)}| + |m_{i,k-1}| |b_{k-1,j}^{(k-1)}|) |\epsilon|, \end{aligned} \quad (4.2.76)$$

where $|\epsilon| = \max_i |\epsilon_i|$, and $|\epsilon| = 2^{-t}$ as defined earlier.

By virtue of the condition (4.2.44) matrix B possesses a diagonal element with the largest magnitude. This implies that no pivoting is required throughout the elimination process, and hence the multipliers have values (in modulus) less than unity, i.e., $|m_{i,k}| \leq 1$. Moreover, if we assume the maximum element (in modulus) in any $B^{(k)}$ is designated by g,

$$|\epsilon_{i,j}^{(k)}| \leq \frac{1}{1-2^{-t}}(g+g)2^{-t}$$

$$< (2.01)g 2^{-t} \text{ (say) .} \quad (4.2.77)$$

Wilkinson (1963) shows this result is applicable to all $\epsilon_{i,j}^{(k)}$ and $\epsilon_{i,j}^{(j+1)}$ ($i > j$) as well.

Subsequently, by applying (4.2.77) on (4.2.71), we have ($i \leq j$)

$$|e_{i,j}| \leq |\epsilon_{i,j}^{(2)}| + |\epsilon_{i,j}^{(3)}| + \dots + |\epsilon_{i,j}^{(i)}|$$

$$< 2.01g 2^{-t} + 2.01g 2^{-t} + \dots + 2.01g 2^{-t}$$

$$= (2.01)(i-1)g 2^{-t} , \quad (4.2.78a)$$

and from the relation (4.2.73), we have ($i > j$)

$$|e_{i,j}| \leq |\epsilon_{i,j}^{(2)}| + |\epsilon_{i,j}^{(3)}| + \dots + |\epsilon_{i,j}^{(j+1)}|$$

$$< 2.01g 2^{-t} + 2.01g 2^{-t} + \dots + 2.01g 2^{-t}$$

$$= 2.01jg 2^{-t} . \quad (4.2.78b)$$

Hence, by combining (4.2.74) and (4.2.78), we have deduced that the error matrix, denoted by δB earlier, is bounded by

$$|\delta B| \leq (2.01)g 2^{-t} E , \quad (4.2.79a)$$

where E has the form

$$E = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 1 & \cdots & \cdots & \cdots & 1 & 1 & \cdots & \cdots & 1 & 1 \\ 1 & 2 & \cdots & \cdots & 2 & 2 & \cdots & \cdots & 2 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (r+1)^{\text{th}} & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \text{row} \rightarrow & 1 & 2 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ E = & 0 & 2 & 3 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 0 & 3 & 4 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \quad (4.2.79b)$$

$\xleftarrow{\quad r \quad} \xrightarrow{\quad r \quad}$
 $\uparrow \qquad \qquad \uparrow$
 $(N-2r+1)^{\text{th}} \text{ col.} \quad (N-r)^{\text{th}} \text{ col.}$

(N×N)

Also, it can be shown that the augmented matrix $(\delta B, \delta \underline{y})$ where \underline{y} is the perturbation in the vector \underline{y} , is bounded as follows,

$$|(\delta B, \delta \underline{y})| \leq (2.01)g \ 2^{-t}(E, \delta \underline{E}) \ ,$$

where E is given by (4.2.79b), and the vector $\delta \underline{E}$ consists of the elements of the N^{th} column of E (see Noble (1969, page 272)).

Finally, the bounds of $|\delta B|$ in terms of 1-norm can be given below, from the inequality (4.2.79a),

$$\begin{aligned} ||\delta B||_1 &\leq 2.01g \ 2^{-t} ||E||_1 \\ &= \frac{N(N-1)}{2} (2.01)g \ 2^{-t} \text{ since } ||E||_1 = \sum_{i=1}^{N-1} i \\ &< 1.01N(N-1)g \ 2^{-t} \ . \end{aligned}$$

4.3.1 ALGORITHM FICM2

This algorithm is basically devoted to obtaining a solution for a linear system of equations, where the matrix is periodic and possesses non-constant elements (see Chapter 3, Section 3.4).

The FICM2 algorithm, as in the preceding algorithm, involves two major steps; firstly, factorizing the matrix of the given system into two cyclic matrices, and hence formulating two linear systems of equations; secondly, solving each of these systems via a triangularization procedure followed by backward and forward schemes. In addition, the elements of the two cyclic factorized matrices will be computed iteratively, which involves the use of the periodic continued fraction theory as discussed in Chapter 2.

Let the given linear system of order N be of the form

$$A\underline{x} = \underline{z}, \quad (4.3.1)$$

where the matrix A is a circulant of bandwidth $2r+1$, such that $N \geq 2r+1$, (r is positive integer), and has the form:

$$A = \begin{bmatrix} a_{0,1} & a_{1,1} & \cdots & a_{r-1,1} & a_{r,1} & & a_{-r,1} & a_{-r+1,1} & \cdots & a_{-1,1} \\ a_{-1,2} & a_{0,2} & a_{1,2} & & & & & & & \\ \vdots & \vdots & \vdots & \ddots & \vdots & & & & & \\ a_{-r+1,r} & & & & & & & & & \\ a_{-r,r+1} & & & & & & & & & \\ \vdots & \vdots & \vdots & \ddots & \vdots & & & & & \\ a_{r,N-r+1} & & & & & & & & & \\ \vdots & \vdots & \vdots & \ddots & \vdots & & & & & \\ a_{2,N-1} & & & & & & & & & \\ a_{1,N} & \cdots & a_{r-1,N} & a_{r,N} & & & a_{-r,N} & a_{-r+1,N} & \cdots & a_{-1,N} & a_{0,N} \end{bmatrix}$$

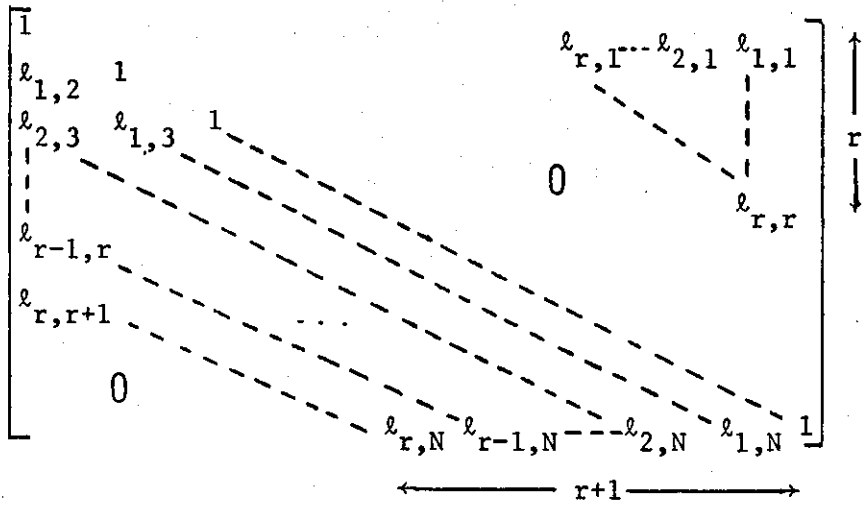
$\xleftarrow{\quad r+1 \quad} \xrightarrow{\quad}$

$\uparrow r \downarrow$

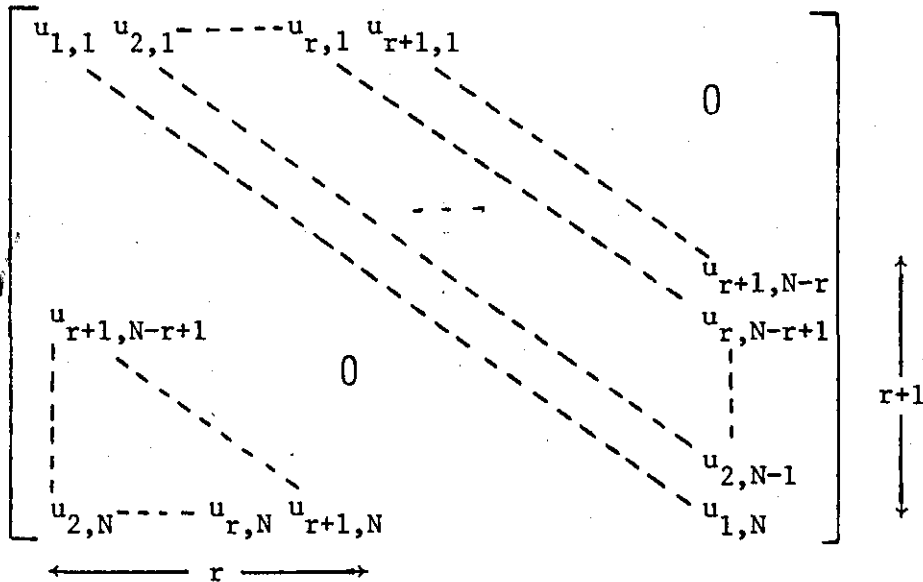
The factorization of the matrix A proposed by the present algorithm consists of evaluating the elements of a lower cyclic triangular matrix, L (say) and an upper cyclic triangular matrix, U (say), such that

$$LU = A, \quad (4.3.2)$$

where the matrices L and U are of the semi-banded form (of order N each) illustrated below.

$L =$ 

and

$U =$ 

The implication of this type of factorization is that, instead of solving the system (4.3.1), we replace this system by the alternative form,

$$LU\underline{x} = \underline{z}. \quad (4.3.3)$$

Consequently, the new factorization strategy has the merit of preserving the form (and the sparsity, if it is the case) of the original periodic matrix, which can be considered as an advantage from the storage strategy viewpoint.

However, the product of the two matrices L and U having been computed are equated with the corresponding elements of A. This procedure eventually yields a set of $(r+1)$ equations which formulates the interlocking relations between the elements of L and U. These relations are,

$$\left. \begin{aligned}
 & \ell_{r,i}^u \ell_{1,i-r}^u &= a_{-r,i} \\
 & \ell_{r-1,i}^u \ell_{1,i-r+1}^u + \ell_{r,i}^u \ell_{2,i-r}^u &= a_{-r+1,i} \\
 & \ell_{r-2,i}^u \ell_{1,i-r+2}^u + \ell_{r-1,i}^u \ell_{2,i-r+1}^u + \ell_{r,i}^u \ell_{3,i-r}^u &= a_{-r+2,i} \\
 & \text{-----} \\
 & \ell_{1,i}^u \ell_{1,i-1}^u + \ell_{2,i}^u \ell_{2,i-2}^u + \dots + \ell_{r,i}^u \ell_{r,i-r}^u &= a_{-1,i} \\
 & u_{1,i}^u + \ell_{1,i}^u \ell_{2,i-1}^u + \ell_{3,i}^u \ell_{3,i-2}^u + \dots + \ell_{r,i}^u \ell_{r+1,i-r}^u &= a_{0,i} \\
 & u_{2,i}^u + \ell_{1,i}^u \ell_{3,i-1}^u + \dots + \ell_{r-1,i}^u \ell_{r+1,i-r+1}^u &= a_{1,i} \\
 & \text{-----} \\
 & u_{r,i}^u + \ell_{1,i}^u \ell_{r+1,i-1}^u &= a_{r-1,i} \\
 & u_{r+1,i}^u &= a_{r,i}
 \end{aligned} \right\}, (4.3.4)$$

where $i=1,2,\dots,N$ for each individual equation, and the second suffix of the u's is interpreted as the modulo of N, i.e., for $u_{k,v}$, $k=1,2,\dots,r+1$ the suffix v is defined by

$$v \equiv v \text{ modulo } N.$$

The system (4.3.4) of $(r+1)$ equations can be reduced to an abbreviated form consisting of five equations (provided $r>1$) which will be considered later in the further analysis of the method. These equations are:

$$\ell_{r,i} u_{1,i-r} = a_{-r,i} \quad (4.3.5a)$$

$$\sum_{j=1}^{r-k+1} \ell_{k+j-1,i} u_{j,i-(k+j-1)} = a_{-k,i}, \quad k=r-1, r-2, \dots, 1 \quad (4.3.5b)$$

$$\sum_{j=r}^1 \ell_{j,i} u_{j+1,i-j} + u_{1,i} = a_{0,i} \quad (4.3.5c)$$

$$\sum_{j=1}^{r-k+1} \ell_{j,i} u_{k+j,i-j} + u_{k,i} = a_{k-1,i}, \quad k=r, r-1, \dots, 2 \quad (4.3.5d)$$

$$u_{r+1,i} = a_{r,i} \quad (4.3.5d)$$

where $i=1,2,\dots,N$ (and as indicated in the equivalent system (4.3.4)).

For the two well-known cases in the numerical problems quoted in Chapter 3 where the coefficient matrix A is tridiagonal or quindagonal, i.e. $r=1,2$ respectively, the equation (4.3.4) becomes,

(a) for $r=1$ (A is periodic tridiagonal),

$$\left. \begin{aligned} \ell_{1,i} u_{1,i-1} &= a_{-1,i} \\ u_{1,i} + \ell_{1,i} u_{2,i-1} &= a_{0,i} \\ u_{2,i} &= a_{1,i} \end{aligned} \right\} i=1,2,\dots,N, \quad (4.3.6)$$

$$u_{1,0} \equiv u_{1,N}, \quad u_{2,0} \equiv u_{2,N} = a_{1,N}$$

(b) for $r=2$ (A is periodic quindagonal),

$$\left. \begin{aligned} \ell_{2,i} u_{1,i-2} &= a_{-2,i} \\ \ell_{1,i} u_{1,i-1} + \ell_{3,i} u_{2,i-2} &= a_{-1,i} \\ u_{1,i} + \ell_{1,i} u_{2,i-1} + \ell_{2,i} u_{3,i-2} &= a_{0,i} \\ u_{2,i} + \ell_{1,i} u_{3,i-1} &= a_{1,i} \\ u_{3,i} &= a_{2,i} \end{aligned} \right\} i=1,2,\dots,N, \quad (4.3.7)$$

where $u_{1,-1} \equiv u_{1,N-1}, \quad u_{1,0} \equiv u_{1,N},$

$u_{2,-1} \equiv u_{2,N-1}, \quad u_{2,0} \equiv u_{2,N},$

$u_{3,-1} \equiv u_{3,N-1},$

and $u_{3,0} \equiv u_{3,N} = a_{2,N}.$

Notice that for the case of the system (4.3.1) being periodic tri-diagonal, the elements of the upper circulant off-diagonal (represented by $u_{2,i}$, $i=1,2,\dots,N$) of the matrix U , in (4.3.2), are known by virtue of the last equation of the system (4.3.6). In the quindagonal case also, the elements $u_{3,i}$, $i=1,2,\dots,N$ are determined already from the last equation of the system (4.3.7). Thus, for the general case (i.e. $r \geq 1$), the elements of the furthest circulant off-diagonal (denoted by $u_{r+1,i}$, $i=1,2,\dots,N$) of the factorized matrix U , in (4.3.2), are equal to the corresponding elements of the original matrix A . This is confirmed by equation (4.3.5c). Hence, each of the factorized matrices L, U includes r unknown circulant diagonals which have to be determined. These circulant diagonals, i.e., $l_{k,i}, u_{k,i}$, $k=1,2,\dots,r$, $i=1,2,\dots,N$, will be computed by an iterative procedure and will be discussed later in this section. Therefore, we proceed now to solve the modified system (4.3.3).

[illegible]

$$\Gamma_r = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & \vdots & \ddots & & \\ & & & 1 & \\ & & & & 0 \\ & -\ell_{r-1,r} & -\ell_{r-2,r} & \dots & \ell_{1,r} & 1 \\ & & & & & \vdots \\ & & & & & & 1 \end{bmatrix},$$

$$\Gamma_k = \begin{bmatrix} 1 & & & & \\ & \downarrow & & & \\ & & 1 & & \\ & & \vdots & \ddots & \\ & & & 1 & \\ & -\ell_{r,k} & \dots & -\ell_{1,k} & 1 \\ & & & & \vdots \\ & & & & & 1 \end{bmatrix} \leftarrow (k+1)^{\text{th}} \text{ row}$$

where $k=r+1, \dots, N-r+1,$

$$\Gamma_{N-r+2} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ & \ell_{r,N-r+2} & \dots & -\ell_{2,N-r+2} & 0 & 1 \\ & & & & & \vdots \\ & & & & & & 1 \end{bmatrix}, \Gamma_{N-r+3} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ & \ell_{r,N-r+3} & \dots & -\ell_{3,N-r+3} & 0 & 0 & 1 \\ & & & & & & \vdots \\ & & & & & & & 1 \end{bmatrix}, \dots$$

up to Γ_N , which has the form,

$$\Gamma_N = \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & \ddots & & & \\ & & & & 1 & & \\ & 0 & & & & \ddots & \\ & & & & & & 1 \\ & & & & & & & -\ell_{r,N} & \cdots & 0 & 1 \end{bmatrix}$$

\uparrow
 $(r+1)^{\text{th}} \text{ col.}$

Now, by premultiplying both sides of the system (4.3.9a) by the matrices $\Gamma_2, \Gamma_3, \dots, \Gamma_r$ in succession we obtain the following system (in compact form),

$$\Gamma_r \Gamma_{r-1} \dots \Gamma_2 \underline{L} \underline{y} = \Gamma_r \Gamma_{r-1} \dots \Gamma_2 \underline{z} . \quad (4.3.10a)$$

The purpose of this strategy is the premultiplication of the system (4.3.9a) by Γ_2 eliminates the element $\ell_{1,2}$ by Γ_3 , eliminates the elements $\ell_{2,3}, \ell_{1,3}$, and so on up to Γ_r eliminating the elements $\ell_{r-1,r}, \ell_{r-2,r}, \dots, \ell_{1,r}$. We proceed now to eliminate r elements, i.e. $\ell_{r,k}, \ell_{r-1,k}, \dots, \ell_{1,k}$ on the k^{th} row, $k=r+1, \dots, N-r+1$, of the system (4.3.9a) by successive premultiplication of both sides of the system (4.3.10a) by $\Gamma_{r+1}, \Gamma_{r+2}, \dots, \Gamma_{N-r+1}$, i.e.

$$\Gamma_{N-r+1} \Gamma_{N-r} \dots \Gamma_3 \Gamma_2 \underline{L} \underline{y} = \Gamma_{N-r+1} \dots \Gamma_3 \Gamma_2 \underline{z} . \quad (4.3.10b)$$

Then, we premultiply both sides of (4.3.10b) by the matrix Γ_{N-r+2} to eliminate $r-1$ elements (i.e. $\ell_{r,N-r+2}, \dots, \ell_{2,N-r+2}$), by the matrix Γ_{N-r+3} to eliminate $r-2$ elements (i.e. $\ell_{r,N-r+3}, \dots, \ell_{3,N-r+3}$), and so on up to Γ_N to eliminate one element (i.e. $\ell_{r,N}$). Hence, the system (4.3.10b) becomes

$$\Gamma_N \Gamma_{N-1} \dots \Gamma_3 \Gamma_2 \underline{L} \underline{y} = \Gamma_N \Gamma_{N-1} \dots \Gamma_3 \Gamma_2 \underline{z} , \quad (4.3.10c)$$

or $\Gamma \underline{y} = \Gamma \underline{z}$, (4.3.10d)

where $\Gamma = \Gamma_N \Gamma_{N-1}, \dots, \Gamma_2$.

In fact the system (4.3.10d) assumes an 'incomplete' triangularized form, and the form of matrix (ΓL) is illustrated in Figure 4.3.1.

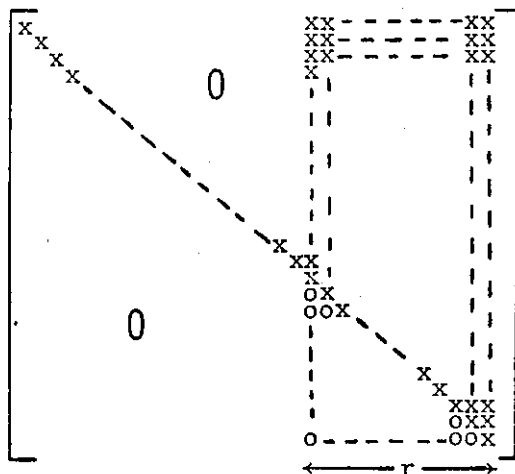


FIGURE 4.3.1: The diagram of the matrix (ΓL) of the system (4.3.10d), where 'x' and 'o' denote non-zero elements. The elements denoted by o will be eliminated when the triangularization process is completed.

The $(N-1)$ transformations to the system (4.3.9a) which are replaced by the equation (4.3.10c) can be algebraically formulated as follows.

Let
$$\hat{f}_{i,N-j+1} = \begin{cases} 1 & , j=i \\ l_{i-j,N-j+1} & , j < i \\ 0 & , \text{otherwise} \end{cases} \begin{cases} i=1,2,\dots,r, \\ j=1,2,\dots,r-1, \end{cases} \quad (4.3.11a)$$

$$f_{k,-j+1} = \begin{cases} -1, & j=k \\ 0 & , \text{otherwise} \end{cases} \quad j,k=1,2,\dots,r, \quad (4.3.11b)$$

$$f_{k,i} = \sum_{j=r}^1 (-l_{j,i}) f_{k,i-j} + \delta, \quad \delta = \begin{cases} 1 & \text{if } i+k=N+1 \\ 0 & \text{otherwise} \end{cases} \quad k=1,2,\dots,r, \quad \left. \begin{array}{l} \text{where } f_{k,t} \text{ for } t \leq 0 \text{ are given by (4.3.11b)} \\ e_i = z_i + \sum_{j=r}^1 (-l_{j,i}) e_{i-j}, \text{ where } e_t \equiv 0, \text{ for all } t < 1 \\ i=1,2,\dots,N-r+1 \end{array} \right\} \quad (4.3.11c)$$

$$\Lambda_{N-2} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}, \dots, \Lambda_{N-r+1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

$$\Lambda_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}, \quad k=N-r, N-r-1, \dots, r,$$

$\uparrow \qquad \qquad \qquad \uparrow$
 $(k+1)^{\text{th}} \text{ col.} \quad (k+r)^{\text{th}} \text{ col.}$

whilst the matrices $\Lambda_{r-1}, \Lambda_{r-2}, \dots, \Lambda_1$ are defined below,

$$\Lambda_{r-1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix},$$

$$\begin{array}{c}
 \begin{array}{c}
 (r+1)^{\text{th}} \text{ col.} \quad (2r-2)^{\text{th}} \text{ col.} \\
 \downarrow \qquad \qquad \downarrow \\
 \Lambda_{r-2} = \left[\begin{array}{ccccccc}
 1 & & & & & & \\
 & 0 & & & & & \\
 & & \downarrow & & & & \\
 & & & 1 & 0 & 0 & \frac{u_{r,r-2}}{u_{1,r+1}} \dots - \frac{u_{r+1,r-2}}{u_{1,2r-1}} \\
 & & & & 1 & & \\
 & & & & & 1 & 1 \\
 & & & & & & \ddots \\
 & 0 & & & & & 1 & 0 \\
 & & & & & & & \ddots \\
 & & & & & & & 1
 \end{array} \right] \dots
 \end{array} \\
 \begin{array}{c}
 (r+1)^{\text{th}} \text{ col.} \\
 \downarrow \\
 \Lambda_1 = \left[\begin{array}{cccc}
 1 & 0 & \dots & 0 - \frac{u_{r+1,1}}{u_{1,r+1}} \\
 & \ddots & & \\
 & & 1 & 0 \\
 & & & \ddots \\
 & 0 & & 1
 \end{array} \right]
 \end{array}
 \end{array}$$

The $(N-1)$ successive premultiplications of the system (4.3.9b) (or its equivalent, system (4.3.8b) can be abbreviated in the form,

$$\Lambda_1 \Lambda_2 \dots \Lambda_{N-2} \Lambda_{N-1} \underline{Ux} = \Lambda_1 \Lambda_2 \dots \Lambda_{N-2} \Lambda_{N-1} \underline{y} \quad (4.3.13a)$$

or

$$\Lambda \underline{Ux} = \Lambda \underline{y}, \quad (4.3.13b)$$

where the matrix $\Lambda = \prod_{i=1}^{N-1} \Lambda_i$.

Furthermore, the algebraic formulation of the transformations of the system (4.3.9b) which are represented by the equation (4.3.13a) can be introduced below.

We define the elements g 's and η 's as follows:

$$\hat{g}_{i,j} = \begin{cases} u_{i-j+1,j} & , j \leq i \\ 0 & , \text{otherwise} \end{cases} \quad \begin{cases} i=1,2,\dots,r \\ j=1,2,\dots,r-1 \end{cases} \quad (4.3.14a)$$

$$g_{i,N+j} = \begin{cases} -u_{1,i} & , \text{ for } j=i \\ 0 & , \text{ otherwise} \end{cases} \quad i, j=1, 2, \dots, r, \quad (4.3.14b)$$

$$g_{k,j} = \sum_{i=1}^r (-u_{i+1,j}/u_{1,i+j}) g_{k,j+i} + \delta, \quad \delta = \begin{cases} u_{1,r} & , \text{ for } k=j \\ 0 & , \text{ otherwise} \end{cases} \quad k=1, 2, \dots, r,$$

where $g_{k,s}$ for $s \geq N$ are defined in (4.3.14b),

$$\eta_j = y_j + \sum_{i=1}^r (-u_{i+1,j}/u_{1,i+j}) \eta_{j+i}, \quad \text{where } \eta_t = 0 \text{ for all } t > N$$

$$j=N, N-1, \dots, r \quad (4.3.14c)$$

and

$$g_{k,r-i} = \sum_{j=1}^{r-i} (-u_{i+j+1,r-i}/u_{1,r+j}) g_{k,r+j} + \hat{g}_{k,r-i}, \quad k=1, 2, \dots, r,$$

where \hat{g} 's are defined in (4.3.14a)

$$\eta_{r-i} = y_{r-i} + \sum_{j=1}^{r-i} (-u_{i+j+1,r-i}/u_{1,r+j}) \eta_{r+j},$$

$$i=1, 2, \dots, r-1, \quad (4.3.14d)$$

Note that the u 's and y 's appearing in the relations of (4.3.14) are those given by the system (4.3.9b) (or the matrices $\Lambda_{N-1}, \dots, \Lambda_1$ for the values of the u 's).

Then the system (4.3.13) can be written in terms of the values g 's and η 's which are defined in (4.3.14), by,

$$\begin{bmatrix} g_{1,1} & g_{2,1} & \dots & g_{r,1} \\ g_{1,2} & g_{2,2} & \dots & g_{r,2} \\ \vdots & \vdots & \ddots & \vdots \\ g_{1,r-1} & g_{2,r-1} & \dots & g_{r,r-1} \\ g_{1,r} & g_{2,r} & \dots & g_{r,r} \\ \vdots & \vdots & \ddots & \vdots \\ g_{1,N} & g_{2,N} & \dots & g_{r,N} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_r \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_r \\ \vdots \\ \eta_N \end{bmatrix}, \quad (4.3.15)$$

0

0

$u_{r,r+1}$

$u_{1,N}$

where the elements $\eta_1, \eta_2, \dots, \eta_N$ and the $g_{k,i}$, $k=1,2,\dots,r$, $i=1,2,\dots,N$ are given by the relations (4.3.14c) and (4.3.14d).

Now, to complete the triangularization of the systems (4.3.12) and (4.3.15), we eliminate the lower and the upper off-diagonal elements in the two systems respectively. Therefore, the next step is to eliminate the elements $f_{k,N-j}$, $k=j+2, j+3, \dots, r$, $j=r-2, r-3, \dots, 0$, of the former system and the elements $g_{i,j}$, $j=1,2,\dots,r-1$, $i=j+1, \dots, r$ of the latter system. This can be performed by the following elimination procedure:

(i) For the system (4.3.12),

the elimination of the elements $f_{k,N-j}$, $k=j+2, \dots, r$, $j=r-2, \dots, 0$ requires $(r-1)$ steps, i.e.,

for $i=1,2,\dots,r-1$

let $\hat{k}=r+1-i$,

$$R_{i,k} = -f_{\hat{k},N-k+1}^{\wedge} / f_{\hat{k},N-\hat{k}+1}^{\wedge},$$

then we have,

$$f_{j,N-k+1}^{(i)} = f_{i,N-k+1}^{(i-1)} + R_{i,k} f_{j,N-\hat{k}+1}^{(i-1)}, \quad j=1,2,\dots,r-i, \quad k=1,2,\dots,r-i$$

$$\text{and} \quad e_{N-k+1}^{(i)} = e_{N-k+1}^{(i-1)} + R_{i,k} e_{N-\hat{k}+1}^{(i-1)},$$

(4.3.16)

where the superscript refers to the i^{th} stage of the elimination process and $f_{\ell,v}^{(0)} \equiv f_{\ell,v}$, $e_s^{(0)} \equiv e_s$ are as given in (4.3.12).

(ii) For the system (4.3.15),

the elimination of the elements $g_{i,j}$, $j=1,2,\dots,r-1$, $i=j+1, \dots, r$, requires $(r-1)$ steps also, i.e.,

for $i=1,2,\dots,r-1$

let $\hat{k}=r+1-i$

$$R_{i,k} = -g_{\hat{k},k}^{\wedge} / g_{\hat{k},\hat{k}}^{\wedge},$$

then we have,

$$g_{j,k}^{(i)} = g_{j,k}^{(i-1)} + R_{i,k} g_{j,\hat{k}}^{(i-1)}, \quad j=1,2,\dots,r-i, \quad k=1,2,\dots,r-i$$

$$\text{and} \quad \eta_k^{(i)} = \eta_k^{(i-1)} + R_{i,k} \eta_{\hat{k}}^{(i-1)}.$$

(4.3.17)

Finally, the backward and forward procedures for the systems (4.3.18) yield the auxiliary vector \underline{y} of the system (4.3.8a) and the solution vector \underline{x} of the system (4.3.8b), i.e., from the system (4.3.18a), we have

$$\left. \begin{aligned} y_N &= e_N / f_{1,N} \\ y_{N-1} &= (e_{N-1} - y_N f_{1,N-1}) / f_{2,N-1} \\ &\vdots \\ y_{N-r+1} &= (e_{N-r+1} - y_{N-r+2} f_{r-1,N-r+2} - \dots - y_N f_{r,N-r+2}) / f_{r,N-r+1} \end{aligned} \right\}$$

and

$$y_j = e_j - \sum_{k=1}^r y_{N-k+1} f_{k,j}, \quad j = N-r, N-r-1, \dots, 1 \quad (4.3.19)$$

and for the system (4.3.18b), we have

$$\left. \begin{aligned} x_1 &= \eta_1 / g_{1,1} \\ x_2 &= (\eta_2 - x_1 g_{1,2}) / g_{2,2} \\ &\vdots \\ x_r &= (\eta_r - \sum_{i=1}^{r-1} x_i g_{i,r}) / g_{r,r} \end{aligned} \right\} \quad (4.3.20)$$

and

$$x_j = (\eta_j - \sum_{i=1}^r x_i g_{i,j}) / u_{1,j}, \quad j = r+1, \dots, N$$

where $\eta_1, \eta_2, \dots, \eta_N$ are computed in terms of the y 's which are obtained from (4.3.19), by the relations (4.3.14) and (4.3.17).

The computational complexity of the algorithm for the solution of (4.3.1) involves approximately $O((6r-2)N)$ additions and $O((6r+1)N)$ multiplications with the predetermination of the elements of matrix factors L and U ; (the case $r=1$, the number can be reduced to $5N$ multiplications (with $4N$ additions) as given by Evans and Okolie (1979)).

4.3.3 DETERMINATION OF THE ELEMENTS OF THE MATRIX FACTORS

Initially, we shall consider the case where the system (4.3.1) is of periodic tridiagonal form, i.e. $r=1$, which has been investigated by Evans and Okolie (1979). In this case, the factorization of the coefficient matrix represented by (4.3.2) obviously yields the relations (4.3.6) which may be rewritten in the form

$$\left. \begin{aligned} \ell_{1,i} &= a_{-1,i}/u_{1,i-1} \\ u_{1,i} &= a_{0,i} - \ell_{1,i} a_{1,i-1} \end{aligned} \right\} \quad i=1,2,\dots,N, \quad (4.3.21)$$

and $u_{1,0} \equiv u_{1,N}, \quad a_{1,0} \equiv a_{1,N};$

bearing in mind that $a_{1,k}, k=1,2,\dots,N$ has been replaced by $u_{2,k}$ according to the last relation of the system (4.3.6).

If the element $\ell_{1,1}$ is assumed as given or determined by an efficient procedure described below, then the elements $\ell_{1,i}$ and $u_{1,i}$ can be easily evaluated by the recurrence relations (4.3.21), and hence the elements of the factorised matrices L and U , are determined for this special case of $r=1$.

However, the element $\ell_{1,1}$ can be computed in a suitable manner based on the theory of the periodic continued fraction. The method was suggested by Evans and Okolie (1979) and can be outlined as follows.

From the relations (4.3.21), we set the index i equal to $1, N, N-1, \dots, 2$ so that we obtain the relations,

$$\ell_{1,1} = a_{-1,1}/u_{1,N},$$

$$u_{1,N} = a_{0,N} - \ell_{1,N} a_{1,N-1}, \text{ i.e., } \ell_{1,1} = a_{-1,1}/(a_{0,N} - \ell_{1,N} a_{1,N-1})$$

$$\ell_{1,N} = a_{-1,N}/u_{1,N-1}, \quad \text{i.e., } \ell_{1,1} = \frac{a_{-1,1}}{a_{0,N} - \frac{a_{-1,N} a_{1,N-1}}{u_{1,N-1}}}$$

$$u_{1,N-1} = a_{0,N-1} - \ell_{1,N-1} a_{1,N-2}, \text{ i.e., } \ell_{1,1} = \frac{a_{-1,1}}{a_{0,N} - \frac{a_{-1,N} a_{1,N-1}}{a_{0,N-1} - \ell_{1,N-1} a_{1,N-2}}}$$

...

and so on. Therefore, the $\ell_{1,1}$ element can be expressed in terms of the elements $a_{-1,i}, a_{0,i}, a_{1,i}$, $i=1,2,\dots,N$, and hence we can formulate an infinite periodic continued fraction whose partial numerators consist of the off-diagonal elements of the matrix A (i.e. $a_{-1,i}, a_{1,i-1}$) and partial denominators consist of the diagonal elements (i.e. $a_{0,i}$), i.e.,

$$a_{1,N} \ell_{1,1} = \frac{a_{-1,1} a_{1,N}}{a_{0,N}} \frac{a_{-1,N} a_{1,N-1}}{a_{0,N-1}} \dots \frac{a_{-1,2} a_{1,1}}{a_{0,1}} \frac{a_{-1,1} a_{1,N}}{a_{0,N}} \\ \frac{a_{-1,N} a_{1,N-1}}{a_{0,N-1}} \dots \frac{a_{-1,2} a_{1,1}}{a_{0,1}} \frac{a_{-1,1} a_{1,N}}{a_{0,N}} \dots \quad (4.3.22a)$$

or

$$a_{1,N} \ell_{1,1} = \frac{\alpha_1}{\beta_1} \frac{\alpha_2}{\beta_2} \dots \frac{\alpha_N}{\beta_N} \frac{\alpha_1}{\beta_1} \frac{\alpha_2}{\beta_2} \dots \frac{\alpha_N}{\beta_N} \frac{\alpha_1}{\beta_1} \dots \quad (4.3.22b)$$

where $\alpha_i = a_{-1,N-i+2} a_{1,N-i+1}$ } $i=1,2,\dots,N$
 $\beta_i = a_{0,N-i+1}$

and $a_{t,k}$, $t=-1,0,1$ is defined such that $k \equiv k$ modulo N .

The linear fractional transformation of (4.3.22) can be expressed by (cf. (2.6.10), Chapter 2),

$$T(\omega) = \frac{\alpha_1}{\beta_1} \frac{\alpha_2}{\beta_2} \dots \frac{\alpha_N}{\beta_N} \omega, \quad (4.3.23)$$

whose fixed points (see definition 2.6.3, Chapter 2) ω_1, ω_2 (say) are the zeros of the quadratic equation

$$\omega = \frac{E_{n-1} \omega + E_n}{F_{n-1} \omega + F_n}, \quad (4.3.24)$$

where E_n, F_n (as indicated in Section 2.6) are given by the following recurrence formulae,

$$\left. \begin{aligned} E_0 &= 0, & F_0 &= 1 \\ E_1 &= \alpha_1, & F_1 &= \beta_1 \\ E_n &= \beta_j E_{n-1} - \alpha_j E_{n-2} \\ F_n &= \beta_j F_{n-1} - \alpha_j F_{n-2} \end{aligned} \right\} \begin{aligned} & j=j \text{ modulo } N, \\ & j=n=2,3,4,\dots \end{aligned} \quad (4.3.25)$$

By virtue of Theorem 2.6.1, if $\omega_1 \neq \omega_2$, then the continued fraction (4.2.23) converges to the value, $\hat{\omega}$ (say), such that $\hat{\omega} = \max(\omega_1, \omega_2)$, where ω_1 and ω_2 are the roots of the equation (4.2.24). Hence,

$$a_{1,N} \ell_{1,1} = \hat{\omega} \quad (4.3.26)$$

or $\ell_{1,1} = \hat{\omega} / a_{1,N}$.

The convergence of the sequence $\left\{ \frac{E_n}{F_n} \right\}$ occurs after a sufficient number of steps, when the magnitude of the difference between two successive approximants of the continued fraction is sufficiently small or satisfies the following relation, i.e.,

for the i^{th} and $(i-1)^{\text{th}}$ approximants, we need the inequality

$$\left| \frac{E_i}{F_i} - \frac{E_{i-1}}{F_{i-1}} \right| < \epsilon \quad (\epsilon \text{ is small}) \quad (4.3.27)$$

to be satisfied. The sufficient condition for the convergence of the periodic continued fraction (4.3.22) has already been given in Section 2.6.

We now revert to the equations (4.3.5) and consider the general case, where $r > 1$, i.e. the case where the circulant matrix A of (4.3.1) is quindagonal, septadiagonal, ... etc. To determine the unknown coefficient of the L and U matrices from the non-linear equations (4.3.5) we adopt an iterative procedure as discussed below.

We assume that the initial guess values

$$\left. \begin{array}{l} \ell_{1,i}^{(0)} \\ \ell_{2,i}^{(0)} \\ \vdots \\ \ell_{r-1,i}^{(0)} \end{array} \right\} \quad i=1,2,\dots,N \quad (4.3.28)$$

and

are given. The superscripts in (4.3.28) (and in the discussion which follows) refer to the iterative step.

Now, whilst the $u_{r+1,i}$, $i=1,2,\dots,N$ is given by the equation (4.3.5e)

as pointed out earlier in this section, the $u_{r,i}^{(1)}, u_{r-1,i}^{(2)}, \dots, u_{2,i}^{(1)}$, $i=1,2,\dots,N$ can be determined by the equation (4.3.5d), i.e.,

$$u_{k,i}^{(1)} = a_{k-1,i} - \sum_{j=1}^{r-k+1} \ell_{j,i}^{(0)} u_{k+j,i-j}^{(1)}, \quad k=r, r-1, \dots, 2, \quad i=1,2,\dots,N \quad (4.3.29)$$

where the $\ell_{j,i}^{(0)}$ are given in (4.3.28).

Furthermore, we assume that the $\ell_{r,k}^{(1)}$, $k=1,2,\dots,r$ are evaluated by considering the use of a periodic continued fraction procedure which will be shown later (similar to the manner of evaluating the $\ell_{1,1}$ in (4.3.6) or (4.3.21)).

Therefore, the $\ell_{r,k}$, $k=r+1,\dots,N$ and $u_{1,i}$, $i=1,2,\dots,N$ can be obtained from the following recursive relations which are derived from (4.3.5a) and (4.3.5c), i.e.,

$$\left. \begin{aligned} \ell_{r,i}^{(1)} &= a_{-r,i} / u_{1,i-r}^{(1)}, \quad i \neq 1,2,\dots,r \\ \text{and } u_{1,i}^{(1)} &= a_{0,i} - \ell_{r,i}^{(1)} u_{r+1,i-r}^{(1)} - \sum_{j=r-1}^1 \ell_{j,i}^{(0)} u_{j+1,i-j}^{(1)} \end{aligned} \right\} \quad i=1,2,\dots,N. \quad (4.3.30)$$

To calculate the first step of the iteration process we are required to calculate the $\ell_{r-1,i}^{(1)}, \ell_{r-2,i}^{(1)}, \dots, \ell_{1,i}^{(1)}$, $i=1,2,\dots,N$. This may be achieved by rewriting the equations (4.3.5b) with an additional term $\epsilon_{k,i}$, $k=r-1,\dots,1$, $i=1,2,\dots,N$, i.e.,

$$\left. \begin{aligned} \ell_{r-1,i}^{(0)} u_{1,i-(r-1)}^{(1)} + \ell_{r,i}^{(1)} u_{1,i-r}^{(1)} + \epsilon_{r-1,i}^{(0)} &= a_{-r+1,i} \\ \ell_{r-2,i}^{(0)} u_{1,i-(r-2)}^{(1)} + \ell_{r-1,i}^{(1)} u_{2,i-(r-1)}^{(1)} + \ell_{r,i}^{(1)} u_{3,i-r}^{(1)} + \epsilon_{r-2,i}^{(0)} &= a_{-r+2,i} \\ \vdots & \\ \ell_{1,i}^{(0)} u_{1,i-1}^{(1)} + \ell_{2,i}^{(1)} u_{2,i-2}^{(1)} + \dots + \ell_{r-1,i}^{(1)} u_{r-1,i-(r-1)}^{(1)} + \ell_{r,i}^{(1)} u_{r,i-r}^{(1)} + \epsilon_{1,i}^{(0)} &= a_{-1,i} \end{aligned} \right\} \quad i=1,2,\dots,N, \quad (4.3.31)$$

where $\epsilon_{k,i}^{(0)}$, $k=r-1, r-2, \dots, 1$, $i=1,2,\dots,N$ refers to the error term due to the 'incorrect' value of the corresponding $\ell_{k,i}^{(0)}$.

If we assume that $\ell_{k,i}^{(1)}$ (and the $u^{(1)}$'s) are 'accurate' values, then the relations (4.3.31) may be written as,

$$\left. \begin{aligned} \ell_{r-1,i}^{(1)} u_{1,i-(r-1)}^{(1)} + \ell_{r,i}^{(1)} u_{1,i-r}^{(1)} &= a_{-r+1,i} \\ \ell_{r-2,i}^{(1)} u_{1,i-(r-2)}^{(1)} + \ell_{r-1,i}^{(1)} u_{2,i-(r-1)}^{(1)} + \ell_{r,i}^{(1)} u_{3,i-r}^{(1)} &= a_{-r+2,i} \\ \vdots & \\ \ell_{1,i}^{(1)} u_{1,i-1}^{(1)} + \ell_{2,i}^{(1)} u_{2,i-2}^{(1)} + \dots + \ell_{r-1,i}^{(1)} u_{r-1,i-(r-1)}^{(1)} + \ell_{r,i}^{(1)} u_{r,i-r}^{(1)} &= a_{-1,i} \end{aligned} \right\} \quad i=1,2,\dots,N. \quad (4.3.32)$$

By subtracting the first equation, the second equation, up to the last equation of (4.3.32) from the corresponding equation of (4.3.31), we obtain the result,

$$\left. \begin{aligned} (\ell_{r-1,i}^{(0)} - \ell_{r-1,i}^{(1)}) u_{1,r-(r-1)}^{(1)} + \epsilon_{r-1,i}^{(0)} &= 0 \\ (\ell_{r-2,i}^{(0)} - \ell_{r-2,i}^{(1)}) u_{1,i-(r-2)}^{(1)} + \epsilon_{r-2,i}^{(0)} &= 0 \\ \vdots & \\ (\ell_{1,i}^{(0)} - \ell_{1,i}^{(1)}) u_{1,i-1}^{(1)} + \epsilon_{1,i}^{(0)} &= 0 \end{aligned} \right\}$$

which can be rearranged to give,

$$\left. \begin{aligned} \ell_{r-1,i}^{(1)} &= \ell_{r-1,i}^{(0)} + \epsilon_{r-1,i}^{(0)} / u_{1,i-(r-1)}^{(1)} \\ \ell_{r-2,i}^{(1)} &= \ell_{r-2,i}^{(0)} + \epsilon_{r-2,i}^{(0)} / u_{1,i-(r-2)}^{(1)} \\ \vdots & \\ \text{and } \ell_{1,i}^{(1)} &= \ell_{1,i}^{(0)} + \epsilon_{1,i}^{(0)} / u_{1,i-1}^{(1)} \end{aligned} \right\} \quad i=1,2,\dots,N. \quad (4.3.33)$$

In fact, the equations of (4.3.31) and (4.3.33) are associated in an alternate manner. This is because we compute $\epsilon_{r-1,i}^{(0)}$ from the first equation of (4.3.31), followed by $\ell_{r-1,i}^{(1)}$ from the first equation of (4.3.33); then back to the second equation of (4.3.31) to compute $\epsilon_{r-2,i}^{(1)}$ followed by $\ell_{r-2,i}^{(1)}$ from the second equation of (4.3.33); and so on.

However, the s^{th} stage of the iterative process for determining the ℓ 's and u 's of the equations (4.3.5) can be summarized as follows:

Step 1 Initialize $\ell_{1,i}^{(0)}, \ell_{2,i}^{(0)}, \dots, \ell_{r-1,i}^{(0)}$, $i=1,2,\dots,N$.

Step 2 Obtain $u_{r,i}^{(s)}, u_{r-1,i}^{(s)}, \dots, u_{2,i}^{(s)}$ successively from the relation

$$u_{k,i}^{(s)} = a_{k-1,i} - \sum_{j=1}^{r-k+1} \ell_{j,i}^{(s-1)} u_{k+j,i-j}^{(s)}, \quad k=r,r-1,\dots,2, \quad i=1,2,\dots,N,$$

and $u_{r+1,v}^{(s)} = a_{r,v}$, $v=1,2,\dots,N$.

Step 3 (a) Determine $\ell_{r,1}^{(s)}, \ell_{r,2}^{(s)}, \dots, \ell_{r,r}^{(s)}$ by the continued fraction scheme (discussed later),

and

(b) Determine $\ell_{r,r+1}^{(s)}, \dots, \ell_{r,N}^{(s)}$ and $u_{1,1}^{(s)}, \dots, u_{1,N}^{(s)}$ from the recurrence relations,

$$\left. \begin{aligned} \ell_{r,i}^{(s)} &= a_{-r,i} / u_{1,i-r}^{(s)}, \quad i=1,2,\dots,r \\ \text{and} \\ u_{1,i}^{(s)} &= a_{0,i} - \ell_{r,i}^{(s)} u_{r+1,i-r}^{(s)} - \sum_{j=r-1}^1 \ell_{j,i}^{(s-1)} u_{j+1,i-j}^{(s)} \end{aligned} \right\} \quad i=1,2,\dots,N$$

Step 4 Evaluate $\varepsilon_{k,i}^{(s-1)}$ and $\ell_{k,i}^{(s)}$, $k=r-1,\dots,1$, $i=1,2,\dots,N$ as follows:

from (4.3.31) and (4.3.33) alternately, we have

$$(a_1) \quad \varepsilon_{r-1,i}^{(s-1)} = a_{-r+1,i} - \ell_{r-1,i}^{(s-1)} u_{1,i-(r-1)}^{(s)} - \ell_{r,i}^{(s)} u_{2,i-r}^{(s)}$$

$$(b_1) \quad \ell_{r-1,i}^{(s)} = \ell_{r-1,i}^{(s-1)} + \varepsilon_{r-1,i}^{(s-1)} / u_{1,i-(r-1)}^{(s)}$$

$$(a_2) \quad \varepsilon_{r-2,i}^{(s-1)} = a_{-r+2,i} - \ell_{r-2,i}^{(s-1)} u_{1,i-(r-2)}^{(s)} - \ell_{r-1,i}^{(s)} u_{2,i-(r-1)}^{(s)} - \ell_{r,i}^{(s)} u_{3,i-r}^{(s)}$$

$$(b_2) \quad \ell_{r-2,i}^{(s)} = \ell_{r-2,i}^{(s-1)} + \varepsilon_{r-2,i}^{(s-1)} / u_{1,i-(r-2)}^{(s)}$$

\vdots

$$(a_{r-1}) \quad \epsilon_{1,i}^{(s-1)} = a_{-1,i}^{-l_{1,i}^{(s-1)}} u_{1,i-1}^{(s)} l_{2,i}^{(s)} u_{2,i-2}^{(s)} \dots l_{r,i}^{(s)} u_{r,i-r}^{(s)}$$

$$(b_{r-1}) \quad l_{1,i}^{(s)} = l_{1,i}^{(s-1)} + \epsilon_{1,i}^{(s-1)} / u_{1,i-1}^{(s)}$$

where i runs from 1 to N in all the above relations.

Step 5

We define ϵ such that

$$|\epsilon| = \max_k (\max_i |\epsilon_{k,i}|) .$$

Thus, if $|\epsilon| \leq \text{TOL}$ (the desired accuracy), then the iterative process is halted, otherwise we repeat from step 2.

Determination of $l_{r,1}, l_{r,2}, \dots, l_{r,r}$ at each step of the iteration

For simplicity, we consider N to be a multiple of r , i.e.,

$N = tr$, where t is any positive integer, such

$$\text{that } N \geq 2r+1 \quad (4.3.34)$$

By rewriting the equations (4.3.5a) and (4.3.5c) in the form,

$$\begin{aligned} l_{r,i} &= a_{-r,i} / u_{1,i-r} \\ u_{1,i} &= a_{0,i} - \sum_{j=r}^1 l_{j,i} u_{j+1,i-j} \end{aligned} \quad (4.3.35)$$

we may be able to construct r continued fractions to express the terms

$l_{r,1}, l_{r,2}, \dots, l_{r,r}$. For example in the quindagonal case, i.e. $r=2$, from the relations (4.3.35) (or the first and the third equations of (4.3.7))

$l_{2,1}$ and $l_{2,2}$ can be expressed in terms of periodic continued fractions as follows,

$$\begin{aligned} a_{-2,1} a_{2,N-1} \\ a_{2,N-1} l_{2,1} &= (a_{0,N-1} - l_{1,N-1} u_{2,N-2}) - \frac{a_{-2,N-1} a_{2,N-3}}{(a_{0,N-3} - l_{1,N-3} u_{2,N-4}) - \frac{a_{-2,N-3} a_{2,N-5}}{(a_{0,N-5} - \dots) - \dots}} \\ &\quad \vdots \\ &\quad \frac{a_{-2,3} a_{2,1}}{(a_{0,1} - \dots) - \frac{a_{-2,1} a_{2,N-1}}{(a_{0,N-1} - \dots) - \dots}} \\ &\quad \vdots \end{aligned}$$

and

$$a_{2,N}^{l_{2,2}} = \frac{a_{-2,2}^{a_{2,N}}}{(a_{0,N}^{-l_{1,N}})^{u_{2,N-1}}} - \frac{a_{-2,N}^{a_{2,N-2}}}{(a_{0,N-2}^{-l_{1,N-2}})^{u_{2,N-3}}} - \frac{a_{-2,N-2}^{a_{2,N-4}}}{(a_{0,N-4}^{-\dots})} - \dots$$

$$\frac{a_{-2,4}^{a_{2,2}}}{(a_{0,2}^{-\dots})} - \frac{a_{-2,2}^{a_{2,N}}}{(a_{0,N}^{-\dots})} - \dots$$

In a similar way by considering the two recurrence relations of (4.3.35) the $l_{r,1}, l_{r,2}, \dots, l_{r,r}$ can be expressed in the form of continued fractions where abbreviated forms are:

$$a_{r,N-r+1}^{l_{r,1}} = \left\{ \frac{\alpha_{1,1}}{\beta_{1,1}} - \frac{\alpha_{1,2}}{\beta_{1,2}} - \frac{\alpha_{1,3}}{\beta_{1,3}} - \dots - \frac{\alpha_{1,t}}{\beta_{1,t}} - \frac{\alpha_{1,1}}{\beta_{1,1}} - \dots - \frac{\alpha_{1,t}}{\beta_{1,t}} - \frac{\alpha_{1,1}}{\beta_{1,1}} - \dots \right\}$$

$$a_{r,N-r+2}^{l_{r,2}} = \left\{ \frac{\alpha_{2,1}}{\beta_{2,1}} - \frac{\alpha_{2,2}}{\beta_{2,2}} - \frac{\alpha_{2,3}}{\beta_{2,3}} - \dots - \frac{\alpha_{2,t}}{\beta_{2,t}} - \frac{\alpha_{2,1}}{\beta_{2,1}} - \dots - \frac{\alpha_{2,t}}{\beta_{2,t}} - \frac{\alpha_{2,1}}{\beta_{2,1}} - \dots \right\}$$

$$\vdots$$

and

$$a_{r,N}^{l_{r,r}} = \left\{ \frac{\alpha_{r,1}}{\beta_{r,1}} - \frac{\alpha_{r,2}}{\beta_{r,2}} - \frac{\alpha_{r,3}}{\beta_{r,3}} - \dots - \frac{\alpha_{r,t}}{\beta_{r,t}} - \frac{\alpha_{r,1}}{\beta_{r,1}} - \dots - \frac{\alpha_{r,t}}{\beta_{r,t}} - \frac{\alpha_{r,1}}{\beta_{r,1}} - \dots \right\}$$

(4.3.36a)

where

$$\left. \begin{aligned} \alpha_{k,i} &= a_{-r,s+r}^{a_{r,s}} \\ \beta_{k,i} &= a_{0,s} - \sum_{j=r-1}^1 l_{j,s} u_{j+1,s-j} \end{aligned} \right\} \begin{aligned} s &= N-ir+k, \\ k &= 1, 2, \dots, r \\ i &= 1, 2, \dots, t \end{aligned} \quad (4.3.36b)$$

and the second suffices of the a's, l's and u's are interpreted as modulo of N.

The linear fractional transformation of each fraction in (4.3.36a) will have a form similar to (4.3.23), i.e.,

$$T(\omega_k) = \frac{\alpha_{k,1}}{\beta_{k,1}} \frac{\alpha_{k,2}}{\beta_{k,2}} \dots \frac{\alpha_{k,t}}{\beta_{k,t}} \omega_k, \quad k=1,2,\dots,r \quad (4.3.37)$$

whose fixed points, $\omega_{k,1}, \omega_{k,2}$ (say), are the roots of the equation

$$\omega_k = \frac{E_{n-1}\omega_k - E_n}{F_{n-1}\omega_k - F_n}, \quad (4.3.38)$$

where E_n, F_n (cf. (4.3.75)) are given by

$$\begin{aligned} E_0 &= 0, \quad F_0 = 1, \\ E_1 &= \alpha_{k,1}, \quad F_1 = \beta_{k,1}, \\ \left. \begin{aligned} E_n &= \beta_{k,j} E_{n-1} - \alpha_{k,j} E_{n-2} \\ F_n &= \beta_{k,j} F_{n-1} - \alpha_{k,j} F_{n-2} \end{aligned} \right\} \begin{aligned} &j=j \text{ modulo } (t), \\ &j=n=2,3,\dots, \end{aligned} \end{aligned} \quad (4.3.39)$$

Now, if the roots of the quadratic equations (4.3.38) are unequal,

i.e. $\omega_{k,1} \neq \omega_{k,2}$, then according to Theorem 26.1 the fraction (4.3.37)

converges to the value, $\hat{\omega}_k$ (say), where $\hat{\omega}_k = \max(\omega_{k,1}, \omega_{k,2})$. Consequently,

by assuming $\hat{\omega}_k$ as the limit of the k^{th} ($k=1,2,\dots,r$) fraction of (4.3.36a),

we write

$$a_{r,N-r+k} \ell_{r,k} = \hat{\omega}_k$$

$$\text{or} \quad \ell_{r,k} = \hat{\omega}_k / a_{r,N-r+k}, \quad k=1,2,\dots,r. \quad (4.3.39)$$

The convergence of the sequence $\left\{ \frac{E_n}{F_n} \right\}$ has been discussed earlier,

and the sufficient condition for the convergence of the k^{th} fraction

(4.3.36a) is (see Section 2.6):

$$0 < \gamma_{k,i} \leq \frac{1}{4}, \quad (4.3.40)$$

where $k=1,2,\dots,r$, $i=1,2,\dots,t$,

and

$$\gamma_{k,1} = \alpha_{k,1} / \beta_{k,1},$$

$$\gamma_{k,i} = \alpha_{k,i} / \beta_{k,i-1} \beta_{k,i}, \quad \beta_{k,i-1} \text{ and } \beta_{k,i} \neq 0.$$

From the experimental results we notice that condition (4.3.40)

was satisfied at each iteration step.

However, if $N(\geq 2r+1)$ is relaxed from the restriction (4.3.35) and an integer, J (say), is introduced such that

$$J = N - tr, \quad (4.3.41)$$

where $t = \left\lfloor \frac{N}{r} \right\rfloor$, then we may conclude the following points:

- (i) for $J=0$, then we have the case followed by (4.3.34), which implied that the lengths of the cycles of the continued fractions (4.3.36a) were equal, i.e. for the k^{th} fraction of (4.3.36), $\alpha_{k,i}$ and $\beta_{k,i}$ were such that $i=1,2,\dots,t$, whilst,
- (ii) for $J \neq 0$, the only fractions of (4.3.36) related to $l_{r,1}, \dots, l_{r,r-J}$ will have cycles of length t , while the remaining fractions, $l_{r,r-J+1}, \dots, l_{r,r}$, will have cycles of length $t+1$, i.e.,

$$\left. \begin{array}{l} \text{for } l_{r,s}, s=1,2,\dots,r-J, \text{ we have } \alpha_{k,i} \text{ and } \beta_{k,i}, \text{ such that} \\ \qquad \qquad \qquad i=1,2,\dots,t, \\ \text{for } l_{r,s}, s=r-J+1,\dots,r, \text{ we have } \alpha_{k,i} \text{ and } \beta_{k,i}, \text{ such that} \\ \qquad \qquad \qquad i=1,2,\dots,t+1, \end{array} \right\}$$

$$\text{where } t \text{ is as given in (4.3.41).} \quad (4.3.42)$$

Finally, it is important to point out that the use of the periodic continued fraction procedure described above was adopted on the basis of the extension of the method suggested by Okolie (1978) or Evans and Okolie (1979) for the tridiagonal case which has been outlined earlier. Later, from the experimental results it was noticed without considering the use of continued fractions, that the iterative procedure, summarized by step 1, ..., step 5 earlier, does converge.

In this case, the steps of the iterative procedure to evaluate the l 's and u 's coefficients of the matrices L and U respectively, can be re-written as follows:

Step 1' Initialize $l_{1,i}^{(0)}, l_{2,i}^{(0)}, \dots, l_{r,i}^{(0)}, i=1,2,\dots,N$.

Step 2' Obtain $u_{r,i}^{(s)}, \dots, u_{2,i}^{(s)}, u_{1,i}^{(s)}$ in succession from the relation

$$u_{k,i}^{(s)} = a_{k-1,i} - \sum_{j=1}^{r-k+1} \ell_{j,i}^{(s-1)} u_{k+j,i-j}^{(s)}, \quad k=r, \dots, 2, 1, \quad i=1, 2, \dots, N,$$

$$(\text{and } u_{r+1,v}^{(s)} = a_{r,v}, \quad v=1, 2, \dots, N,)$$

Step 3'

Evaluate $\epsilon_{k,i}^{(s-1)}$ and $\ell_{k,i}^{(s)}$, $i=1, 2, \dots, N$, $k=r, r-1, \dots, 1$ alternately, from the relations (c.f. (4.3.5b))

$$\left. \begin{aligned} \epsilon_{k,i}^{(s-1)} &= a_{-k,i} - \ell_{k,i}^{(s-1)} u_{1,i-k}^{(s)} - \sum_{j=2}^{r-k+1} \ell_{k+j-1,i}^{(s)} u_{j,i-(k+j-1)}^{(s)} \\ \text{and (c.f. (4.3.33))} \\ \ell_{k,i}^{(s)} &= \ell_{k,i}^{(s-1)} + \epsilon_{k,i}^{(s-1)} / u_{1,i-k}^{(s)}, \end{aligned} \right\} (4.3.43)$$

$i=1, 2, \dots, N, \quad k=r, r-1, \dots, 1$

where the process of computing the two quantities operates such that, after obtaining $\epsilon_{r,i}^{(s-1)}$, we have to compute $\ell_{r,i}^{(s)}$, then $\epsilon_{r-1,i}^{(s-1)}, \ell_{r-1,i}^{(s)}, \dots$, etc. as in step 4 which was given earlier.

Step 4'

As in step 5 of the previous procedure.

4.3.4 SOLUTION OF SYMMETRIC LINEAR SYSTEMS

We re-consider the system (4.3.1) with the assumption that its coefficient matrix A is symmetric and possesses non-constant elements, i.e.,

$$A\underline{x} = \underline{z}, \quad (4.3.44a)$$

where A is a $(N \times N)$ matrix and of the form,

$$A = \begin{bmatrix} a_{0,1} & a_{1,1} & a_{2,1} & \cdots & a_{r,1} & & a_{r,N-r+1} & \cdots & a_{2,N-1} & a_{1,N} \\ a_{1,1} & a_{0,2} & a_{1,2} & a_{2,2} & a_{r,2} & & & & & a_{2,N} \\ a_{2,1} & a_{1,2} & a_{0,3} & & & & & & & a_{3,N} \\ \vdots & a_{2,2} & & & & & & & & \vdots \\ a_{r,1} & a_{r,2} & & & & & & & & a_{r,N} \\ & & & & & & & & & \vdots \\ & & & & & & & & & a_{r,N-r} \\ & & & & & & & & & \vdots \\ & & & & & & & & & a_{2,N-2} \\ & & & & & & & & & a_{0,N-1} \\ & & & & & & & & & a_{1,N-1} \\ & & & & & & & & & a_{0,N} \end{bmatrix} \quad (4.3.44b)$$

In this case, the factorization (4.3.2) may be written as

$$LDL^T = A, \quad (4.3.45a)$$

where A is symmetric given in (4.3.44b), D is a diagonal matrix of the form,

$$\begin{bmatrix} d_1 & & & & \\ & d_2 & & & \\ & & d_3 & & \\ & & & & 0 \\ & & & & & \ddots \\ & & & & & & d_N \end{bmatrix} \quad (4.3.45b)$$

and L is a unit lower triangular matrix given in (4.3.2).

Thus, the system (4.3.44a) whose coefficient matrix given in (4.3.44b) or (4.3.45a) can be replaced by an alternative form, that is,

$$LDL^T \underline{x} = \underline{z} . \quad (4.3.46)$$

As before, we split this system into two systems with the insertion of an auxiliary vector, \underline{y} (say) such that

$$L\underline{y} = \underline{z} \quad (4.3.47a)$$

and

$$DL^T \underline{x} = \underline{y}$$

or

$$L^T \underline{x} = D^{-1} \underline{y} \equiv \underline{\eta} , \quad (4.3.47b)$$

where $\underline{y} \equiv [y_1, \dots, y_N]^T$, $\underline{\eta} \equiv [\eta_1, \dots, \eta_N]^T$ such that

$$\eta_i = y_i / d_i, \quad i=1,2,\dots,N \text{ and } d_i \text{ are the elements of } D \text{ in}$$

(4.3.45b).

Since the system (4.3.47a) is exactly the same as the system (4.3.8a), hence the elimination procedure discussed in subsection 4.3.2 is applicable to the former; subsequently we can write the final result of the elimination procedure given by (4.3.18a), i.e. (by considering (4.3.11) and (4.3.16) the system (4.3.47a) becomes),

$$\begin{bmatrix} 1 & & & & & & & & & & \\ & 1 & & & & & & & & & \\ & & 0 & & & & & & & & \\ & & & 1 & & & & & & & \\ & & & & 1 & & & & & & \\ & & & & & 1 & & & & & \\ & & & & & & 1 & & & & \\ & & & & & & & 1 & & & \\ & & & & & & & & 1 & & \\ & & & & & & & & & 1 & \\ & & & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} f_{r,1} & \dots & -f_{2,1} & f_{1,1} \\ f_{r,2} & \dots & -f_{2,2} & f_{1,2} \\ \vdots & & \vdots & \vdots \\ f_{r,N-r+1} & \dots & -f_{2,N-r+1} & \\ \vdots & & \vdots & \vdots \\ f_{2,N-1} & f_{1,N-1} \\ f_{1,N} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_{N-r+1} \\ \vdots \\ e_N \end{bmatrix} \quad (4.3.48)$$

where e_i , $i=1,2,\dots,N$ are given by (4.3.11) and (4.3.16).

Also, the elimination procedure for the comparison system (4.3.47b) takes place in the same way as for the system (4.3.8b) (or (4.3.9b)).

Thus, the algebraic formulation of the elimination given by (4.3.14) and (4.3.17) can be applied to the system (4.3.47b), i.e. by taking,

$u_{k,i}$, $k=1, \dots, r+1$, $i=1, 2, \dots, N$ in (4.3.8b) as follows

$$\text{with } \left. \begin{array}{l} u_{k+1,i} = \ell_{k,i}, \quad k=1, 2, \dots, r \\ u_{1,i} = 1 \end{array} \right\} i=1, 2, \dots, N, \quad (4.3.49)$$

where $\ell_{k,i}$ are the elements of the coefficient matrix of the system (4.3.47b) (i.e., L^T). [N.B. the relation (4.3.49) can easily be derived by equating the corresponding elements of U in (4.3.8b) and of L^T in (4.3.47b)].

Hence, the relations of (4.3.14) after the substitution for the expressions of the u 's given in (4.3.4a), can be rewritten as follows,

$$\hat{g}_{i,j} = \left\{ \begin{array}{ll} 1 & , \text{ for } i=j \\ \ell_{j-1,i} & , \text{ for } j < i \\ 0 & , \text{ otherwise} \end{array} \right\} \quad \begin{array}{l} i=1, 2, \dots, r \\ j=1, 2, \dots, r-1, \end{array} \quad (4.3.50a)$$

$$g_{i,N+j} = \left\{ \begin{array}{ll} -1 & , \text{ for } j=i \\ 0 & , \text{ otherwise} \end{array} \right\} \quad i, j=1, 2, \dots, r, \quad (4.3.50b)$$

$$g_{k,j} = \sum_{i=1}^r (-\ell_{i,j+i}) g_{k,j+i} + \delta, \quad \delta = \left\{ \begin{array}{ll} 1 & , \text{ for } k=j \\ 0 & , \text{ otherwise} \end{array} \right\} \quad k=1, 2, \dots, r,$$

where the $g_{k,s}$ for $s > N$ are given by (4.3.50b);

$$\hat{\eta}_j = \eta_j + \sum_{i=1}^r (-\ell_{i,j+i}) \hat{\eta}_{j+i}, \quad \text{where the } \hat{\eta}_t \equiv 0 \text{ for all } t > N$$

$$j=N-N-1, \dots, r \quad (4.3.50c)$$

and

$$\left. \begin{array}{l} g_{k,r-i} = \sum_{j=1}^{r-i} (-\ell_{i+j,j+r}) g_{k,r+j} + \hat{g}_{k,r-i}, \quad k=1, 2, \dots, r \\ \text{where the } g\text{'s are given in (4.3.50a)} \end{array} \right\} i=1, 2, \dots, r-1,$$

$$\hat{\eta}_j = \eta_{r-i} + \sum_{j=1}^{r-i} (-\ell_{i+j,j+r}) \quad (4.3.50d)$$

where the $\eta_1, \eta_2, \dots, \eta_N$ are the components of the right-hand side vector of the system (4.3.47b).

Subsequently, by virtue of (4.3.50) the system (4.3.47b) becomes,

$$\begin{bmatrix} g_{1,1} & g_{2,1} & \dots & g_{r,1} & & \\ g_{1,2} & g_{2,2} & \dots & g_{r,2} & & \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ g_{1,r} & g_{2,r} & \dots & g_{r,r} & & \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ g_{1,N} & g_{2,N} & \dots & g_{r,N} & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \hat{\eta}_1 \\ \hat{\eta}_2 \\ \vdots \\ \hat{\eta}_r \\ \vdots \\ \hat{\eta}_N \end{bmatrix}, \quad (4.3.51)$$

0
1
0
1

where the $\hat{\eta}_1, \hat{\eta}_2, \dots, \hat{\eta}_N$ and $g_{k,i}$, $k=1,2,\dots,r$, $i=1,2,\dots,N$ are given by the relations (4.3.50c) and (4.3.50d).

To complete the solution procedure we should eliminate the elements $g_{i,j}$, $j=1,2,\dots,r-1$, $i=j+1,\dots,r$, of the system (4.3.51) as follows (see (4.3.17)):

For $i=1,2,\dots,r-1$,

let $\hat{k}=r+1-i$,

$$R_{i,k} = -g_{k,k}^{(i-1)} / g_{k,k}^{(i-1)}$$

then we have

$$g_{j,k}^{(i)} = g_{j,k}^{(i-1)} + R_{i,k} g_{j,\hat{k}}^{(i-1)}, \quad j=1,2,\dots,r-i$$

$$\hat{\eta}_k^{(i)} = \hat{\eta}_k^{(i-1)} + R_{i,k} \hat{\eta}_{\hat{k}}^{(i-1)},$$

(4.3.52)

where the superscript refers to the elimination step, and

$$g_{\lambda,\nu}^{(0)} \equiv g_{\lambda,\nu}, \quad \eta_s^{(0)} \equiv \eta_s \text{ are as given in (4.3.51).}$$

The triangularization of the system (4.3.51) is now complete and has the form,

$$\begin{bmatrix}
 g_{1,1} & & & & & \\
 g_{1,2} & g_{2,2} & & & & \\
 \vdots & & \ddots & & & \\
 g_{1,r} & & & g_{r,r} & & \\
 \vdots & & & & \ddots & \\
 g_{1,N-1} & & & & & g_{r-1,N} \\
 g_{1,N} & g_{2,N} & \dots & \dots & g_{r,N} & 0
 \end{bmatrix}
 \begin{bmatrix}
 x_1 \\
 x_2 \\
 \vdots \\
 x_r \\
 \vdots \\
 x_N
 \end{bmatrix}
 =
 \begin{bmatrix}
 \hat{\eta}_1 \\
 \hat{\eta}_2 \\
 \vdots \\
 \hat{\eta}_r \\
 \vdots \\
 \hat{\eta}_N
 \end{bmatrix}
 \quad (4.3.53)$$

where the elements $\hat{\eta}_i$ and $g_{k,i}$, $k=1,2,\dots,r$, $i=r,\dots,N$ are as located in the system (4.3.51), whilst the elements $\hat{\eta}_j$, $j=1,2,\dots,r-1$ and $g_{k,j}$, $k=1,2,\dots,r-1$, $j=1,2,\dots,k$ are given by (4.3.52).

Finally, the solution of the system (4.3.47a) is obtained by the back substitution process from (4.3.48) which in fact yields exactly the relations (4.3.19). Whilst the solution vector \underline{x} (of the system (4.3.47b) is obtained by the forward substitution process from (4.3.53), i.e.,

$$\left. \begin{aligned}
 x_1 &= \hat{\eta}_1 / g_{1,1} \\
 x_2 &= (\hat{\eta}_2 - x_1 g_{1,2}) / g_{2,2} \\
 &\vdots \\
 x_r &= (\hat{\eta}_r - \sum_{i=1}^{r-1} x_i g_{i,r}) / g_{r,r} \\
 x_j &= \hat{\eta}_j - \sum_{i=1}^r x_i g_{i,j}, \quad j=r+1,\dots,N
 \end{aligned} \right\} \quad (4.3.54)$$

We now describe the determination of the elements of the matrices L and D given in (4.3.5a), this has been given in section (4.3.3).

Since the coefficient matrix A considered here is symmetric and is factorized by (4.3.45a), the equations (4.3.4) can be reduced to (r+1) equations only as follows.

A comparison between the two factorizations of A given in (4.3.2) and (4.3.45a) yields,

$$U = DL^T,$$

which by equating the corresponding elements implies

$$u_{k+1,i} = d_i \ell_{k,i+k}, \quad i=1,2,\dots,N, \quad k=1,2,\dots,r \quad (4.3.54)$$

(and the second suffix of ℓ is interpreted as modulo of N).

Hence, the equations (4.3.4) by considering (4.3.54) can be reduced to the following form,

$$\left. \begin{aligned} \ell_{r,i} d_{i-r} &= a_{r,i} \\ \ell_{r-1,i} d_{i-r+1} + \ell_{r,i} d_{i-r} \ell_{1,i-r+1} &= a_{r-1,i} \\ \hline \ell_{1,i} d_{i-1} + \ell_{2,i} d_{i-2} \ell_{1,i-1} + \dots + \ell_{r,i} d_{i-r} \ell_{r-1,i-1} &= a_{1,i} \\ d_i + \ell_{1,i} d_{i-1} \ell_{1,i+1} + \ell_{2,i} d_{i-2} \ell_{2,i+2} + \dots + \ell_{r,i} d_{i-r} \ell_{r,i+r} &= a_{0,i} \end{aligned} \right\} \quad (4.3.55)$$

where i assumes the values 1 to N in each of these equations.

However, the iterative procedure discussed in subsection (4.3.3) can be applied to the equations (4.3.55); here we shall consider the procedure which does not involve the use of periodic continued fractions, i.e. the step 1', ..., step 4' given at the end of the previous subsection. These steps become,

Step 1'' Initialize $\ell_{1,i}^{(0)}, \ell_{2,i}^{(0)}, \dots, \ell_{r,i}^{(0)}$, $i=1,2,\dots,N$.

Step 2'' Obtain $d_i^{(s)}$, $i=1,2,\dots,N$ as follows:-

$$d_i^{(s)} = a_{0,i} / \left[1 + \sum_{k=1}^r \ell_{k,i}^{(s-1)} \ell_{k,i+k}^{(s-1)} \right], \quad i=1,2,\dots,N.$$

Step 3'' Evaluate $\epsilon_{k,i}^{(s-1)}$ and $\ell_{k,i}^{(s)}$, $i=1,2,\dots,N$, $k=r,r-1,\dots,1$

alternately, (we shall rewrite the relations (4.3.43) after the substitution for the u 's given by (4.3.54)),

$$\left. \begin{aligned} \epsilon_{k,i}^{(s-1)} &= a_{k,i} - \ell_{k,i}^{(s-1)} d_{i-k}^{(s)} - \sum_{j=2}^{r-k+1} \ell_{k+j-1,i}^{(s)} d_{i-(k+j-1)}^{(s)} \\ \text{and } \ell_{k,i}^{(s)} &= \ell_{k,i}^{(s-1)} + \epsilon_{k,i}^{(s-1)} / d_{i-k}^{(s)} \end{aligned} \right\} \begin{matrix} i=1,2,\dots,N, \\ k=r,r-1,\dots,1. \end{matrix} \quad (4.3.56)$$

the process of computing the two quantities is as given for (4.3.45).

Step 4''

Finally, see step 5 (or step 4') of the previous procedures.

4.3.5 ROUNDING ERROR ANALYSIS

The error analysis for the algorithm solution can be discussed briefly as shown below.

We shall consider the systems (4.3.8a) and (4.3.8b) (or their equivalent (4.3.9a) and (4.3.9b) respectively), i.e.,

$$L\underline{y} = \underline{z} \quad (4.3.57a)$$

and
$$U\underline{x} = \underline{y} . \quad (4.3.57b)$$

In fact, the system (4.3.57a) is similar to the system (4.2.62) from the structure of the coefficient matrix (or the locations of the non-zero elements) viewpoint. On these grounds, the error analysis of (4.3.57a) takes place in an analogous manner to the system (4.2.62) and ends up with similar result, i.e.,

By assuming δL to be the perturbation in L of (4.3.57a), then we should obtain the following bounds for the modulus of δL , that is

$$|\delta L| \leq (2.01) \tilde{g} 2^{-t} E , \quad (4.3.58)$$

where \tilde{g} is taken as the modulo of the maximum element during the elimination procedure of (4.3.57a), and the matrix E is given by (4.2.77b).

A similar analysis of subsection 4.2.7 can be applied to the system (4.3.57b) to derive the bounds of the perturbation in U , δU (say), and finally gives the result,

$$|\delta U| \leq (2.01) \hat{g} 2^{-t} \hat{E} ,$$

where \hat{g} is taken again as the modulo of the maximum element appearing throughout the elimination process of (4.3.57b), and matrix \hat{E} is of the following form (c.f. E in (4.2.77b)),

$(r+1)^{\text{th}}$
 col.
 \downarrow

r^{th} row \rightarrow

$E =$

$$\begin{bmatrix}
 (N-1) & (N-1) & \cdots & (N-r+1) & (N-r) & 0 & \cdots & 0 \\
 \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
 (N-r) & (N-r) & \cdots & (N-r) & (N-r) & (N-2r+1) & 0 & \cdots & 0 \\
 \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\
 (N-r-1) & (N-r-1) & \cdots & (N-r-1) & 0 & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\
 \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\
 3 & 3 & \cdots & 3 & \vdots & \vdots & \vdots & \vdots & \vdots \\
 2 & 2 & \cdots & 2 & \vdots & \vdots & \vdots & \vdots & \vdots \\
 1 & 1 & \cdots & 1 & \vdots & \vdots & \vdots & \vdots & \vdots \\
 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0
 \end{bmatrix}$$

$\leftarrow r \rightarrow$

$\leftarrow (N-r)^{\text{th}}$
 row

$(N \times N)$

4.3.6 CONVERGENCE ANALYSIS OF THE ITERATIVE PROCEDURE APPLIED IN THE ALGORITHM FICM2

We shall consider the theoretical analysis of the convergence of the iterative procedure summarized by step 1', up to step 4' in the preceding subsection (which also applies to the other iterative procedures given earlier). Previously it was shown that the iterative procedure is used to determine the unknown elements of the matrix factors L and U of A, or, alternatively, to solve the non-linear equations (4.3.4) (or (4.3.5)).

After some rearrangement of (4.3.4) we rewrite the system as follows (where $i=1,2,\dots,N$ for each equation)

$$\begin{array}{rcl}
 u_{r+1,i} & & = a_{r,i} \\
 u_{r,i} + l_{1,i}^u a_{r,i-1} & & = a_{r-1,i} \quad (4.3.59a) \\
 u_{r-1,i} + l_{1,i}^u u_{r,i-1} + l_{2,i}^u a_{r,i-2} & & = a_{r-2,i} \\
 \hline
 u_{2,i} + l_{1,i}^u u_{3,i-1} + l_{2,i}^u u_{4,i-2} + \dots + l_{r-2,i}^u u_{r,i-r+2} + l_{r-1,i}^u a_{r,i-r+1} & & = a_{1,i} \\
 u_{1,i} + l_{1,i}^u u_{2,i-1} + l_{2,i}^u u_{3,i-2} + \dots + l_{r-1,i}^u u_{r,i-r+1} + l_{r,i}^u a_{r,i-r} & & = a_{0,i} \\
 l_{r,i}^u u_{1,i-r} & & = a_{-r,i} \\
 l_{r-1,i}^u u_{1,i-r+1} + l_{r,i}^u u_{2,i-r} & & = a_{-r+1,i} \\
 l_{r-2,i}^u u_{1,i-r+2} + l_{r-1,i}^u u_{2,i-r+1} + l_{r,i}^u u_{3,i-r} & & = a_{-r+2,i} \\
 \hline
 l_{1,i}^u u_{1,i-1} + l_{2,i}^u u_{2,i-2} + l_{3,i}^u u_{3,i-3} + \dots + l_{r,i}^u u_{r,i-r} & & = a_{-1,i}
 \end{array}$$

(4.3.59b)

where the second suffix of the u's is illustrated in (4.3.4). We also notice that the $u_{r+1,i}$ is replaced by $a_{r,i}$ by virtue of (4.3.59a) in the appropriate equations of (4.3.59b).

It can easily be observed from (4.3.59b), where the suffix i runs from 1 to N for each equation, that the system (4.3.59b) consists of

2rN non-linear equations. These equations can be written compactly (in a vector-form of 2rN components) as

$$\underline{F}(\underline{x}) = \underline{0}, \quad (4.3.60a)$$

$$\left. \begin{array}{l} \text{where} \quad \underline{x} = [x_1, x_2, \dots, x_{2rN}]^T, \\ \text{and} \quad \underline{F} = [f_1, f_2, \dots, f_{2rN}]^T. \end{array} \right\} \quad (4.3.60b)$$

The vectors in (4.3.60b) may be partitioned into 2r sub-vectors of N components, i.e.,

$$\begin{aligned} \underline{x} &= [\underline{x}^1, \underline{x}^2, \dots, \underline{x}^{2r}] \\ \text{and} \quad \underline{F} &= [\underline{f}^1, \underline{f}^2, \dots, \underline{f}^{2r}]; \end{aligned}$$

this means that the N components of the k^{th} sub-vector, $\underline{x}^k = [x_1^k, \dots, x_N^k]$, $\underline{f}^k = [f_1^k, \dots, f_N^k]$ are expressed in terms of the corresponding components of the vectors \underline{x} and \underline{F} in (4.3.60b) respectively:

$$\left. \begin{aligned} x_i^k &= x_{(k-1)N+i} \\ f_i^k &= f_{(k-1)N+i} \end{aligned} \right\} \quad i=1, 2, \dots, N, \quad k=1, 2, \dots, 2r. \quad (4.3.61)$$

Furthermore, we now define x_i^k in (4.3.61) in terms of the unknowns of the equations (4.3.59b) as follows, (for $i=1, 2, \dots, N$)

$$\left. \begin{aligned} x_i^1 &= u_{r,i} \\ x_i^2 &= u_{r-1,i} \\ \hline x_i^r &= u_{1,i} \\ x_i^{r+1} &= l_{r,i} \\ x_i^{r+2} &= l_{r-1,i} \\ \hline \text{and} \quad x_i^{2r} &= l_{1,i} \end{aligned} \right\} \quad (4.3.62)$$

(or $x_i^k = u_{r-k+1,i}$ and $x_i^{r+k} = l_{r-k+1,i}$, $k=1, 2, \dots, r$).

Thus, equations (4.3.59b) can be rewritten in terms of x_i^k using the

definitions (4.3.62) as follows (after some rearrangement which includes defining the f_i^k in (4.3.61)),

$$\begin{aligned}
 f_i^1 &= -(x_i^1 + x_i^{2r} a_{r,i-1}^{-a_{r-1,i}}) = 0 \\
 f_i^2 &= -(x_i^2 + x_i^{2r} x_{i-1}^1 + x_i^{2r-1} a_{r,i-2}^{-a_{r-2,i}}) = 0 \\
 \hline
 f_i^{r-1} &= -(x_i^{r-1} + x_i^{2r} x_{i-1}^{r-2} + x_i^{2r-1} x_{i-2}^{r-3} + \dots + x_i^{r+3} x_{i-r+2}^1 + x_i^{r+2} a_{r,i-r+1}^{-a_{1,i}}) = 0 \\
 f_i^r &= -(x_i^r + x_i^{2r} x_{i-1}^{r-1} + x_i^{2r-1} x_{i-2}^{r-2} + \dots + x_i^{r+2} x_{i-r+1}^1 + x_i^{r+1} a_{r,i-r}^{-a_{0,i}}) = 0 \\
 f_i^{r+1} &= -(x_i^{r+1} x_{i-r}^r - a_{-r,i}) = 0 \\
 f_i^{r+2} &= -(x_i^{r+2} x_{i-r+1}^r + x_i^{r+1} x_{i-r}^{r-1} - a_{-r+1,i}) = 0 \\
 f_i^{r+3} &= -(x_i^{r+3} x_{i-r+2}^r + x_i^{r+2} x_{i-r+1}^{r-1} + x_i^{r+1} x_{i-r}^{r-2} - a_{-r+2,i}) = 0 \\
 \hline
 f_i^{2r} &= -(x_i^{2r} x_{i-1}^r + x_i^{2r-1} x_{i-2}^{r-1} + x_i^{2r-2} x_{i-3}^{r-2} + \dots + x_i^{r+1} x_{i-r}^1 - a_{-1,i}) = 0
 \end{aligned}
 \tag{4.3.63}$$

In fact, it is necessary to emphasize that (4.3.63) is the explicit form of (4.3.60), whilst the implicit form may be given by,

$$\begin{aligned}
 f_i^1((x_1^1, x_2^1, \dots, x_N^1), (x_1^2, x_2^2, \dots, x_N^2), \dots, (x_1^{2r}, x_2^{2r}, \dots, x_N^{2r})) &= 0 \\
 f_i^2((x_1^1, x_2^1, \dots, x_N^1), (x_1^2, x_2^2, \dots, x_N^2), \dots, (x_1^{2r}, x_2^{2r}, \dots, x_N^{2r})) &= 0 \\
 \hline
 f_i^{2r}((x_1^1, x_2^1, \dots, x_N^1), (x_1^2, x_2^2, \dots, x_N^2), \dots, (x_1^{2r}, x_2^{2r}, \dots, x_N^{2r})) &= 0
 \end{aligned}
 \tag{4.3.64}$$

where each of f_i^k , $k=1,2,\dots,2r$ is (generally) taken to consist of $2rN$ arguments.

The $2rN$ non-linear equations in (4.3.63) can be solved by a so-called Seidel-type (or Gauss-Seidel (Ortega and Rheinboldt (1970)) iterative procedure, whose principle is that the most recent information obtained is immediately exploited in the following steps. The s^{th}

available step of the equations (4.3.63) is as follows:

$$\left. \begin{aligned}
 & f_i^1((x_1^1)^{(s)}, \dots, (x_{i-1}^1)^{(s)}, (x_i^1)^{(s-1)}, \dots, (x_N^1)^{(s-1)}, ((x_1^2)^{(s-1)}, \dots, \\
 & \quad (x_N^2)^{(s-1)}, \dots, ((x_1^{2r})^{(s-1)}, \dots, (x_N^{2r})^{(s-1)})) = 0 \equiv f_i^1((y_i^1)^{(s-1)}) \\
 & f_i^2((x_1^1)^{(s)}, \dots, (x_N^1)^{(s)}, ((x_1^2)^{(s)}, \dots, (x_{i-1}^2)^{(s)}, (x_i^2)^{(s-1)}, \dots, (x_N^2)^{(s-1)}, \\
 & \quad \dots, ((x_1^{2r})^{(s-1)}, \dots, (x_N^{2r})^{(s-1)})) = 0 \equiv f_i^2((y_i^2)^{(s-1)}) \\
 & \hline
 & f_i^{2r}((x_1^1)^{(s)}, \dots, (x_N^1)^{(s)}, \dots, ((x_1^{2r-1})^{(s)}, \dots, (x_N^{2r-1})^{(s)}, ((x_1^{2r})^{(s)}, \dots, \\
 & \quad (x_{i-1}^{2r})^{(s)}, (x_i^{2r})^{(s-1)}, \dots, (x_N^{2r})^{(s-1)})) = 0 \equiv f_i^{2r}((y_i^{2r})^{(s-1)})
 \end{aligned} \right\} \quad (4.3.65)$$

On the other hand the one-step *Gauss-Seidel-Newton* method (Ortega and Rheinboldt (1970), page 220, see also Ames (1969), page 250 and Wendroff (1966), page 162)), would take the following form:

$$\left. \begin{aligned}
 (x_i^1)^{(s)} &= (x_i^1)^{(s-1)} - \left(\frac{\partial f_i^1}{\partial x_i^1} ((y_i^1)^{(s-1)}) \right)^{-1} \cdot f_i^1((y_i^1)^{(s-1)}) \\
 (x_i^2)^{(s)} &= (x_i^2)^{(s-1)} - \left(\frac{\partial f_i^2}{\partial x_i^2} ((y_i^2)^{(s-1)}) \right)^{-1} f_i^2((y_i^2)^{(s-1)}) \\
 &\hline
 (x_i^{2r})^{(s)} &= (x_i^{2r})^{(s-1)} - \left(\frac{\partial f_i^{2r}}{\partial x_i^{2r}} ((y_i^{2r})^{(s-1)}) \right)^{-1} f_i^{2r}((y_i^{2r})^{(s-1)})
 \end{aligned} \right\} \quad (4.3.66)$$

where y_i^k , $k=1,2,\dots,2r$ are defined in (4.3.65). It is clear that the relations (4.3.66) involve the evaluation of $2rN$ partial derivatives (which are required for the Gauss-Seidel Newton process, (Ortega and Rheinboldt (1970), page 223)).

However, the partial derivative terms of (4.3.66) can be derived from (4.3.63) to obtain,

$$\begin{aligned}
 \frac{\partial f_i^1}{\partial x_i^1} &= -1, \quad i=1,2,\dots,N, \\
 \frac{\partial f_i^2}{\partial x_i^2} &= -1, \quad i=1,2,\dots,N, \\
 \hline
 \frac{\partial f_i^r}{\partial x_i^r} &= -1, \quad i=1,2,\dots,N, \\
 \frac{\partial f_i^{r+1}}{\partial x_i^{r+1}} &= -x_{i-r}^r, \quad i=1,2,\dots,N, \\
 \frac{\partial f_i^{r+2}}{\partial x_i^{r+2}} &= -x_{i-r+1}^r, \quad i=1,2,\dots,N, \\
 \hline
 \frac{\partial f_i^{2r}}{\partial x_i^{2r}} &= -x_{i-1}^r, \quad i=1,2,\dots,N, \\
 \text{(or)} \quad \frac{\partial f_i^k}{\partial x_i^k} &= -1, \quad \frac{\partial f_i^{r+k}}{\partial x_i^{r+k}} = -x_{i-r+k-1}^r, \quad i=1,2,\dots,N, \quad k=1,2,\dots,r)
 \end{aligned} \tag{4.3.67}$$

The substitution of these terms in (4.3.66) gives the result,

$$\begin{aligned}
 (x_i^1)(s) &= (x_i^1)(s-1) + f_i^1((y_i^1)(s-1)), \\
 (x_i^2)(s) &= (x_i^2)(s-1) + f_i^2((y_i^2)(s-1)), \\
 \hline
 \end{aligned} \tag{4.3.68a}$$

$$\begin{aligned}
 (x_i^r)(s) &= (x_i^r)(s-1) + f_i^r((y_i^r)(s-1)), \\
 (x_i^{r+1})(s) &= (x_i^{r+1})(s-1) + f_i^{r+1}((y_i^{r+1})(s-1)) / (x_{i-r}^r)(s), \\
 (x_i^{r+2})(s) &= (x_i^{r+2})(s-1) + f_i^{r+2}((y_i^{r+2})(s-1)) / (x_{i-r+1}^r)(s), \\
 \hline
 \end{aligned} \tag{4.3.68b}$$

$$\text{and} \quad (x_i^{2r})(s) = (x_i^{2r})(s-1) + f_i^{2r}((y_i^{2r})(s-1)) / (x_{i-1}^r)(s).$$

Moreover, by substituting in (4.3.68a) for $f_i^k(y_i^k)$, $k=1,2,\dots,2r$ as

defined exactly in (4.3.63), the relations in (4.3.68) now take the form,

$$\begin{aligned}
 (x_i^1)(s) &= (x_i^1)(s-1) - ((x_i^1)(s-1) + (x_i^{2r})(s-1) a_{r,i-1}^{-a_{r-1,i}}) \\
 &= a_{r-1,i}^{- (x_i^{2r})(s-1)} a_{r,i-1} \\
 (x_i^2)(s) &= (x_i^2)(s-1) - ((x_i^2)(s-1) + (x_i^{2r})(s-1) (x_{i-1}^1)(s) + (x_i^{2r-1})(s-1) a_{r,i-2} \\
 &\quad - a_{r-2,i}) = a_{r-2,i}^{- (x_i^{2r})(s-1)} (x_{i-1}^1)(s) - (x_i^{2r-1})(s-1) a_{r,i-2} \\
 \hline
 (x_i^{r-1})(s) &= (x_i^{r-1})(s-1) - ((x_i^{r-1})(s-1) + (x_i^{2r})(s-1) (x_{i-1}^{r-2})(s) + (x_i^{2r-1})(s-1) (x_{i-2}^{r-3})(s) \\
 &\quad + \dots + (x_i^{r+3})(s-1) (x_{i-r+2}^1)(s) + (x_i^{r+2})(s-1) a_{r,i-r+1}^{-a_{1,i}}) \\
 &= a_{1,i}^{- (x_i^{2r})(s-1)} (x_{i-1}^{r-2})(s) - (x_i^{2r-1})(s-1) (x_{i-2}^{r-3})(s) - \dots - \\
 &\quad - (x_i^{r+3})(s-1) (x_{i-r+2}^1)(s) - (x_i^{r+2})(s-1) a_{r,i-r+1} \\
 (x_i^r)(s) &= (x_i^r)(s-1) - ((x_i^r)(s-1) + (x_i^{2r})(s-1) (x_{i-1}^{r-1})(s) + (x_i^{2r-1})(s-1) (x_{i-2}^{r-2})(s) \\
 &\quad + \dots + (x_i^{r+2})(s-1) (x_{i-r+1}^1)(s) + (x_i^{r+1})(s-1) a_{r,i-r}^{-a_{0,i}}) \\
 &= a_{0,i}^{- (x_i^{2r})(s-1)} (x_{i-1}^{r-1})(s) - (x_i^{2r-1})(s-1) (x_{i-2}^{r-2})(s) - \dots - \\
 &\quad - (x_i^{r+2})(s-1) (x_{i-r+1}^1)(s) - (x_i^{r+1})(s-1) a_{r,i-r}
 \end{aligned}
 \tag{4.3.69}$$

If we define the quantities $\epsilon_{k,i}$, $k=r, r-1, \dots, 1$, $i=1, 2, \dots, N$ as

$$\epsilon_{r-k+1,i} = f_i^{r+k}, \quad i=1, 2, \dots, N, \quad k=1, 2, \dots, r, \tag{4.3.70}$$

then by substituting in (4.3.68) for $\epsilon_{k,i}$ as in (4.3.70), and for

x_i^j , $j=1, 2, \dots, 2r$ in terms of the u 's and the ℓ 's as defined in (4.3.62)

we have from (4.3.68) the following result (noting that (4.3.68a) is

replaced by its equivalent form (4.3.69)),

$$\begin{aligned}
 u_{r,i}^{(s)} &= a_{r-1,i}^{-\ell_{1,i}^{(s-1)}} a_{r,i-1}^{(s-1)} \\
 u_{r-1,i}^{(s)} &= a_{r-2,i}^{-\ell_{1,i}^{(s-1)}} u_{r,i-1}^{(s-1)} a_{r,i-2}^{(s-1)} \\
 \hline
 u_{2,i}^{(s)} &= a_{1,i}^{-\ell_{1,i}^{(s-1)}} u_{3,i-1}^{(s-1)} a_{2,i}^{-\ell_{2,i}^{(s-1)}} u_{4,i-2}^{(s-1)} \dots a_{r-2,i}^{-\ell_{r-2,i}^{(s-1)}} u_{r,i-r+2}^{(s-1)} \\
 &\quad a_{r-1,i}^{-\ell_{r-1,i}^{(s-1)}} a_{r,i-r+1}^{(s-1)} \\
 u_{1,i}^{(s)} &= a_{0,i}^{-\ell_{1,i}^{(s-1)}} u_{2,i-1}^{(s-1)} a_{2,i}^{-\ell_{2,i}^{(s-1)}} u_{3,i-2}^{(s-1)} \dots a_{r-1,i}^{-\ell_{r-1,i}^{(s-1)}} u_{r,i-r+1}^{(s-1)} \\
 &\quad a_{r,i}^{-\ell_{r,i}^{(s-1)}} a_{r,i-r}^{(s-1)}
 \end{aligned}
 \tag{4.3.71a}$$

$$\begin{aligned}
 \ell_{r,i}^{(s)} &= \ell_{r,i}^{(s-1)} + \epsilon_{r,i}^{(s-1)} / u_{1,i-r}^{(s)} \\
 \ell_{r-1,i}^{(s)} &= \ell_{r-1,i}^{(s-1)} + \epsilon_{r-1,i}^{(s-1)} / u_{1,i-r+1}^{(s)} \\
 \hline
 \ell_{1,i}^{(s)} &= \ell_{1,i}^{(s-1)} + \epsilon_{1,i}^{(s-1)} / u_{1,i-1}^{(s)}
 \end{aligned}
 \tag{4.3.71b}$$

Hence, it can easily be noticed that the relations in (4.3.71a) and (4.3.71b) coincide with the corresponding ones in step 2' and step 3' respectively of the iterative procedure given in subsection 4.3.3.

Convergence Criteria

The investigation of the convergence of the above non-stationary process is similar to the stationary processes discussed in Section 2.2, (Chapter 2), where in both cases the formulation of the equation (2.2.22) is required, certainly the iteration matrix of this equation (unlike the stationary case) varies at each step of the non-stationary process. Moreover, as in the linear case (Section 2.2) the coefficient matrix is split into three matrices, the Jacobian matrix, J , of the non-linear

equations (4.3.63) is also split as follows, (at $(s-1)^{\text{th}}$ step of the iteration),

$$J^{(s-1)} = D^{(s-1)} - L^{(s-1)} - U^{(s-1)}, \quad s=0,1,\dots, \quad (4.3.72)$$

where D , L and U are non-singular diagonal, strictly lower and upper triangular matrices respectively.

The Jacobian J is of order $2rN$, and its partial derivatives can be derived from the equation (4.3.63). It can be introduced in the following block form,

$$J = \begin{bmatrix} J_{1,1} & J_{1,2} & J_{1,3} & \cdots & J_{1,2r} \\ J_{2,1} & J_{2,2} & J_{2,3} & \cdots & J_{2,2r} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ J_{2r,1} & J_{2r,2} & J_{2r,3} & \cdots & J_{2r,2r} \end{bmatrix}_{(2rN \times 2rN)}, \quad (4.3.73a)$$

where the matrices $J_{k,v}$, $k,v=1,2,\dots,2r$ are of order N each, and their coefficients are determined from (4.3.63) as follows,

$$J_{k,v} = \left[\frac{\partial f_i^k}{\partial x_{(v-1)N+j}} \right], \quad i,j=1,2,\dots,N. \quad (4.3.73b)$$

Furthermore, since the derivatives (4.3.73b) were derived from (4.3.63) it can be noticed that some of the submatrices of J in (4.3.73a) are zero (i.e. null matrices); these are,

$$\left. \begin{aligned} J_{i,j} &= 0, \quad j=i+1, i+2, \dots, 2r-i, \quad i=1, 2, \dots, r-1, \\ J_{r+k,j} &= 0 \\ J_{r+k,r+1+j} &= 0 \end{aligned} \right\} \quad j=1, 2, \dots, r-k, \quad k=1, 2, \dots, r-1. \quad (4.3.74)$$

We note that this only applies for $r > 1$.

Therefore, by virtue of (4.3.74), the matrix J in (4.3.73a) becomes,

$$J = \begin{bmatrix} J_{1,1} & & & & & J_{1,2r} \\ J_{2,1} & J_{2,2} & & 0 & & J_{2,2r-1} & J_{2,2r} \\ \vdots & \vdots & \ddots & & & \vdots & \vdots \\ J_{r,1} & J_{r,2} & \cdots & J_{r,r} & J_{r,r+1} & \cdots & J_{r,2r} \\ & 0 & & J_{r+1,r} & J_{r+1,r+1} & & 0 \\ J_{2r,1} & \cdots & J_{2r,r} & J_{2r,r+1} & \cdots & J_{2r,2r} \end{bmatrix}, \quad (4.3.75)$$

noting that the upper submatrices (including the diagonal) are diagonal submatrices and $-J_{i,i}$, $i=1,2,\dots,r$ are unit matrices, whilst the lower off-diagonal are sparse submatrices with N elements each.

However, the *iteration matrix* M of the equation (2.2.22) takes the following form,

$$M^{(s-1)} = (D^{(s-1)} - L^{(s-1)})^{-1} U^{(s-1)}, \quad (4.3.76)$$

where D, L and U are defined in (4.3.72).

Thus, the scheme (4.3.66) converges to a solution of (4.3.63) provided that the iteration matrix $M^{(s-1)}$ of (4.3.76) has the property required for the linear case (Section 2.2) (see Ames (1969), that the spectral radius, $\rho(M)$, is less than 1.

In this respect, we may formulate a convergent condition for the special case, when $r=1$. For this particular case, the Jacobian matrix $J_{(r=1)}$ has the form,

$$J_{(r=1)} = \begin{bmatrix} J_{1,1} & J_{1,2} \\ J_{2,1} & J_{2,2} \end{bmatrix}, \quad (4.3.77)$$

where $J_{i,j}$, $i,j=1,2$, can be expressed from (4.3.73b) as follows:

$$\left. \begin{aligned} J_{1,1} &= \frac{\partial f_i^1}{\partial x_j} \\ J_{1,2} &= \frac{\partial f_i^1}{\partial x_{N+j}} \\ J_{2,1} &= \frac{\partial f_i^2}{\partial x_j} \\ J_{2,2} &= \frac{\partial f_i^2}{\partial x_{N+j}} \end{aligned} \right\} \quad i, j=1, 2, \dots, N \quad (4.3.78)$$

where x_{N+j} is equivalent to x_j^2 by the assumption (4.3.61).

The derivatives (4.3.78) can be derived from (4.3.63) in the following manner,

$$\begin{aligned} J_{1,1} &= \begin{bmatrix} -1 & & & \\ & -1 & & \\ & & \ddots & \\ & 0 & & -1 \end{bmatrix}_{N \times N}, \quad J_{1,2} = \begin{bmatrix} -a_{-1,N} & & & \\ & a_{-1,1} & & \\ & & \ddots & \\ & 0 & & -a_{-1,N-1} \end{bmatrix}_{N \times N} \\ & \hspace{15em} (4.3.79) \\ J_{2,1} &= \begin{bmatrix} -x_2^2 & & & -x_1^2 \\ & -x_3^2 & & \\ & & \ddots & \\ & 0 & & -x_N^2 \end{bmatrix}_{N \times N}, \quad \text{and } J_{2,2} = \begin{bmatrix} -x_N^1 & & & \\ & -x_1^1 & & \\ & & \ddots & \\ & 0 & & -x_{N-1}^1 \end{bmatrix}_{N \times N} \\ & \hspace{15em} (4.3.79) \end{aligned}$$

By expressing $J_{(r=1)}$ in terms of its coefficient matrices given in (4.3.79) and the splitting procedure as in (4.3.72) then the matrices D-L and U can be obtained in the form,

$$D-L = \begin{bmatrix} -1 & & & & \\ & -1 & & & \\ & & \ddots & & \\ & & & -1 & \\ & & & & 0 \\ x_1^2 & -x_N^1 & & & \\ & & -x_1^1 & & \\ & & & \ddots & \\ & & & & -x_{N-1}^1 \\ 0 & & & & & 0 \\ & & x_2^2 & & & \\ & & & \ddots & & \\ & & & & x_N^2 & \\ 0 & & & & & & -x_{N-1}^1 \end{bmatrix} \quad \leftarrow (N+1)^{\text{th}} \text{ row} \quad (4.3.80a)$$

and

$$U = \begin{bmatrix} & & & & (N+1)^{\text{th}} \text{ col} \\ & & & & \downarrow \\ & -a_{1,N} & & & \\ & & -a_{1,1} & & 0 \\ & & & -a_{1,2} & \\ & & & & \ddots \\ & 0 & & & -a_{1,N-1} \end{bmatrix} \quad (2N \times 2N) \quad (4.3.80b)$$

It is easy to invert the matrix (4.3.80a) which gives,

[illegible]

$$= (-1)^{N,N} \left[(-1)^{N,N} + (-1)^{N+1} \frac{x_1^2 a_{1,N-1}}{x_N^1} \times \frac{x_2^2 a_{1,N}}{x_1^1} \times \frac{x_3^2 a_{1,1}}{x_2^1} \times \dots \times \frac{x_N^2 a_{1,N-2}}{x_{N-1}^1} \right] \quad (4.3.82)$$

By substituting for x_i^2 , $i=1,2,\dots,N$ from the appropriate equation of (4.3.63) (i.e. the equation with f_i^{r+1}) in (4.3.82), then we obtain,

$$\lambda^N + (-1)^{N+1} \frac{a_{-1,1} a_{1,N-1}}{(x_N^1)^2} \times \frac{a_{-1,2} a_{1,N}}{(x_1^1)^2} \times \frac{a_{-1,3} a_{1,1}}{(x_2^1)^2} \times \dots \times \frac{a_{-1,N} a_{1,N-2}}{(x_{N-1}^1)^2} = 0$$

or

$$\lambda^N = \prod_{i=1}^N \frac{a_{-1,i} a_{1,i}}{(x_i^1)^2} \quad (4.3.83)$$

Thus, to ensure that M in (4.3.76) possesses a spectral radius less than 1 in modulus for the special case (i.e. $r=1$), we should have the condition,

$$1 > |\lambda| = \left(\prod_{i=1}^N \frac{a_{-1,i} a_{1,i}}{(x_i^1)^2} \right)^{1/N} \equiv \left(\prod_{i=1}^N \left| \frac{a_{-1,i}}{x_i^1} \right| \left| \frac{a_{1,i}}{x_i^1} \right| \right)^{1/N}, \quad (4.3.84)$$

to be satisfied at each step of the iteration.

On the other hand, if we now assume that $(x_i^2)^{(s-1)} = 0$ at $s=1$ (i.e. the initial solution), then,

$$x_i^1 = a_{0,i}, \quad \text{for } i=1,2,\dots,N$$

and on substitution in (4.3.84) we obtain

$$|\lambda| = \left(\prod_{i=1}^N \left| \frac{a_{-1,i}}{a_{0,i}} \right| \left| \frac{a_{1,i}}{a_{0,i}} \right| \right)^{1/N} < 1. \quad (4.3.85)$$

Moreover, if the coefficient matrix is symmetric then (4.3.85) becomes,

$$|\lambda| = \left(\prod_{i=1}^N \left| \frac{a_{1,i}}{a_{0,i}} \right|^2 \right)^{1/N} < 1, \quad (4.3.86)$$

and if the matrix is constant at the same time, these relations take the simpler form, i.e.,

$$|\lambda| = \left| \frac{a_{1,i}}{a_{0,i}} \right|^2 < 1 \text{ for any } i. \quad (4.3.87)$$

which can be readily identified as the condition for diagonal dominance.

4.4.1 ALGORITHM FIRM1

The type of linear system of equations considered in this section is when the coefficient matrix is *non-periodic* and possesses real non-constant elements. The matrix of the system (3.4.21) which is derived from the two-point boundary-value problem is an example of the type of matrix that is under consideration with the present algorithm.

The FIRML algorithm differs from the algorithms discussed in the previous sections for the periodic-type matrices in that it involves, (i) factorizing the coefficient matrix into two pseudo-inverse rectangular upper and lower triangular matrices and (ii) formulating a coupled system consisting of *underdetermined* and *overdetermined* systems to solve. These two systems will be solved in a related manner. The determination of the elements of the two factor matrices is completed by an iterative procedure.

The linear system related to the present method is assumed to be of order N and has the form,

[illegible]

$$L = \begin{bmatrix} 1 & & & & & \\ \ell_{1,1} & 1 & & & & \\ \ell_{2,1} & \ell_{1,2} & 1 & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ \ell_{r-1,1} & \ell_{r-2,2} & & & & \\ \ell_{r,1} & \ell_{r,2} & & & & \\ & & & & & 0 \\ & & & & & \vdots \\ & & & & & \ell_{r,N-r} & \ell_{2,N-2} & \ell_{1,N-1} & 1 \\ & & & & & \ell_{r,N-r+1} & \ell_{3,N-2} & \ell_{2,N-1} & \ell_{1,N} \\ & & & & & & & & \vdots \\ & & & & & & & & \ell_{r-1,N} \\ & & & & & & & & \ell_{r,N} \end{bmatrix} \equiv \begin{bmatrix} L_1 \\ \hline K_1 \end{bmatrix} \quad (4.4.3b)$$

((N+r) × N)

For the definition of $U_1, V_1, L_1 + K_1$ see subsection 4.4.3.

From the factorization of the matrix A defined by (4.4.2), it follows that the system (4.4.1) is replaced by the alternative system

$$ULx = z, \quad (4.4.4)$$

whose method of solution will be discussed in the next subsection.

Now having completed the product of the two rectangular matrices U and L given in (4.4.3), we can equate the obtained elements of the matrix A (by virtue of the relation (4.4.2)) and so derive the following non-linear equations for determining the elements of L and U ,

$$u_{1,i} \ell_{r,i-r} = a_{-r,i}, \quad i=r+1, \dots, N, \quad (4.4.5a)$$

$$\left. \begin{aligned} u_{1,i} \ell_{r-1,i-r+1} + u_{2,i} \ell_{r,i-r+1} &= a_{-r+1,i}, \quad i=r, \dots, N \\ u_{1,i} \ell_{r,i-1} + u_{2,i} \ell_{r-1,i-1} + \dots + u_{r,i} \ell_{r,i-1} &= a_{-1,i}, \quad i=2, 3, \dots, N \end{aligned} \right\} \quad (4.4.5b)$$

$$u_{1,i} + u_{2,i} \ell_{1,i} + \dots + u_{r,i} \ell_{r-1,i} + u_{r+1,i} \ell_{r,i} = a_{0,i}, \quad i=1, 2, \dots, N, \quad (4.4.5c)$$

$$\left. \begin{aligned}
 u_{2,i} + u_{3,i} \ell_{1,i+1} + \dots + u_{r+1,i} \ell_{r-1,i+1} &= a_{1,i}, i=1,2,\dots,N-1, \\
 u_{r,i} + u_{r+1,i} \ell_{1,i+r-1} &= a_{r-1,i}, i=1,2,\dots,N-r+1, \\
 u_{r+1,i} &= a_{r,i}, i=1,2,\dots,N-r.
 \end{aligned} \right\}$$

(4.4.5d)

For the special cases where the matrix A of the system (4.4.1) is tridiagonal or quindigonal, the matrices of (4.4.3) are of the form,

(i) for $r=1$,

$$U =_{(r=1)} \begin{bmatrix} u_{1,1} & u_{2,1} & & & 0 \\ & u_{1,2} & & & 0 \\ & & \ddots & & \\ 0 & & & u_{2,N-1} & \\ & & & u_{1,N} & u_{2,N} \end{bmatrix} \quad \text{and } L =_{(r=1)} \begin{bmatrix} 1 & & & & 0 \\ \ell_{1,1} & 1 & & & \\ & \ell_{1,2} & 1 & & \\ 0 & & & \ell_{1,N-1} & 1 \\ - & - & - & - & - \\ & 0 & & & \ell_{1,N} \end{bmatrix}$$

(4.4.6a)

and

(ii) $r=2$,

$$U =_{(r=2)} \begin{bmatrix} u_{1,1} & u_{2,1} & u_{3,1} & & 0 \\ & \ddots & \ddots & & 0 \\ & & u_{3,N-2} & & \\ 0 & & & u_{3,N-1} & u_{3,N} \\ & & & u_{1,N} & u_{2,N} \end{bmatrix}$$

and

$$L =_{(r=2)} \begin{bmatrix} 1 & & & & \\ \ell_{1,1} & 1 & & & \\ & \ell_{2,1} & \ell_{1,2} & 1 & \\ & & \ell_{2,2} & & \\ 0 & & & \ell_{2,N-2} & \ell_{1,N-1} & 1 \\ - & - & - & - & - \\ & 0 & & & \ell_{2,N-1} & \ell_{1,N} \\ & & & & & \ell_{2,N} \end{bmatrix}$$

(4.4.6b)

Consequently, the equations in (4.4.5) become, for these two special cases,

in case (i),

$$\left. \begin{aligned} u_{1,i}^{\ell} u_{1,i-1}^{\ell} &= a_{-1,i}, \quad i=2, \dots, N, \\ u_{1,i}^{\ell} + u_{2,i}^{\ell} u_{1,i}^{\ell} &= a_{0,i}, \quad i=1, 2, \dots, N, \\ u_{2,i} &= a_{1,i}, \quad i=1, 2, \dots, N-1, \end{aligned} \right\} (4.4.7a)$$

and in case (ii),

$$\left. \begin{aligned} u_{1,i}^{\ell} u_{2,i-2}^{\ell} &= a_{-2,i}, \quad i=3, \dots, N, \\ u_{1,i}^{\ell} u_{1,i-1}^{\ell} + u_{2,i}^{\ell} u_{2,i-1}^{\ell} &= a_{-1,i}, \quad i=2, \dots, N, \\ u_{1,i}^{\ell} + u_{2,i}^{\ell} u_{1,i}^{\ell} + u_{3,i}^{\ell} u_{2,i}^{\ell} &= a_{0,i}, \quad i=1, 2, \dots, N, \\ u_{2,i}^{\ell} + u_{3,i}^{\ell} u_{1,i+1}^{\ell} &= a_{1,i}, \quad i=1, 2, \dots, N-1, \\ u_{3,i} &= a_{2,i}, \quad i=1, 2, \dots, N-2. \end{aligned} \right\} (4.4.7b)$$

Furthermore, the solution of the system (4.4.1) with A symmetric has, for the cases of the tridiagonal and quindagonal matrices with constant elements been the subject of investigation by Evans (1972) and Okolie (1978) respectively. The matrix equation of the two cases can be written as follows:

For the tridiagonal case,

$$\begin{bmatrix} a_0 & a_1 & & & 0 \\ a_1 & a_0 & a_1 & & \\ & a_1 & a_0 & a_1 & \\ & & a_1 & a_0 & a_1 \\ 0 & & & a_1 & a_0 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix} \quad (4.4.8a)$$

or

$$A_1 \hat{x} = \hat{z}, \quad (4.4.8b)$$

and for the quindagonal case,

$$\begin{bmatrix}
 a_0 & a_1 & a_2 & & & \\
 & a_1 & a_0 & a_1 & a_2 & \\
 & & a_2 & a_1 & a_0 & a_1 & a_2 \\
 & & & \ddots & \ddots & \ddots & \ddots \\
 & & & & a_2 & a_1 & a_0 \\
 & & & & & a_2 & a_1 & a_0 \\
 & & & & & & a_2 & a_1 & a_0
 \end{bmatrix}
 \begin{bmatrix}
 x_1 \\
 \vdots \\
 x_N
 \end{bmatrix}
 =
 \begin{bmatrix}
 z_1 \\
 \vdots \\
 z_N
 \end{bmatrix}
 \quad (4.4.9a)$$

or $A_2 \tilde{x} = \tilde{z}$. (4.4.9b)

Evans, (1972) suggested that the factorization of the matrix A_1 in (4.4.8) can be of the form,

$$A_1 = D_1 Q Q^T, \quad (4.4.10)$$

where

$$Q = \begin{bmatrix}
 1 & -\alpha & & & \\
 & 1 & 0 & & \\
 & & \ddots & \ddots & \\
 & & & 1 & -\alpha \\
 & & & & 1
 \end{bmatrix}_{N \times (N+1)}, \quad D_1 = \begin{bmatrix}
 a_0(1+\alpha^2)^{-1} & & & \\
 & 0 & & \\
 & & \ddots & \\
 & & & a_0(1+\alpha^2)^{-1}
 \end{bmatrix}_{N \times N}$$

and $\alpha = -2a_1 / [a_0 + (a_0^2 - 4a_1^2)^{1/2}]$.

Hence, from (4.4.10) the system (4.4.8) becomes,

$$D_1 Q Q^T \hat{x} = \hat{z}$$

or $Q Q^T \hat{x} = D_1^{-1} \hat{z} = \underline{\eta}$, (4.4.11)

where $\underline{\eta} = [\eta_1, \dots, \eta_N]^T$ and $\eta_i = (1+\alpha^2) \hat{z}_i / a_0$, $i=1, 2, \dots, N$.

Okolie (1978) extended the idea of the factorization (4.4.10) to the quindagonal matrix; and suggested that the matrix A_2 of (4.4.7b) can be factorized as

$$A_2 = D_2 P P^T, \quad (4.4.12)$$

where

$$P = \begin{bmatrix} 1 & -\alpha & -\beta & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & & -\alpha & -\beta \\ & & & & & 1 \\ & 0 & & & & & -\alpha & -\beta \\ & & & & & & & 1 \end{bmatrix}_{N \times (N+2)}, \quad D_2 = \begin{bmatrix} a_0(1+\alpha^2+\beta^2)^{-1} & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & a_0(1+\alpha^2+\beta^2)^{-1} \end{bmatrix}_{N \times N}$$

$$\text{and} \quad \left. \begin{aligned} \alpha &= \beta a_1 / (1 - \beta) a_2, \\ \beta &= \alpha a_2 / (a_1 + a_2 \alpha) \end{aligned} \right\} \quad (4.4.13)$$

Hence from (4.4.12) the system (4.4.7b) becomes

$$D_2 P P^T \tilde{\underline{x}} = \tilde{\underline{z}}$$

$$\text{or} \quad P P^T \tilde{\underline{x}} = D_2^{-1} \tilde{\underline{z}} = \underline{\xi}, \quad (4.4.14)$$

where $\underline{\xi} = [\xi_1, \dots, \xi_N]^T$ and $\xi_i = (1 + \alpha^2 + \beta^2) \tilde{z}_i / a_0$, $i=1, \dots, N$.

Finally, by inserting auxiliary vectors into (4.4.11) and (4.4.14) say \underline{y} and $\tilde{\underline{y}}$ respectively, such that these vectors are defined as

$$\begin{aligned} \underline{y} &= [y_1, \dots, y_N, y_{N+1}]^T, \\ \tilde{\underline{y}} &= [y_1, \dots, y_N, y_{N+1}, y_{N+2}]^T, \end{aligned}$$

then, we may write the system (4.4.11) as,

$$Q \underline{y} = \underline{\eta} \quad (4.4.15a)$$

$$\text{and} \quad Q^T \underline{x} = \underline{y} \quad (4.4.15b)$$

and the system (4.4.14) as

$$P \tilde{\underline{y}} = \underline{\xi} \quad (4.4.16a)$$

$$\text{and} \quad P^T \tilde{\underline{x}} = \tilde{\underline{y}}. \quad (4.4.16b)$$

The treatment of the system (4.4.15a) was considered by Evans (1972) via an elimination process, ending up with relations in which each of the components y_1, y_2, \dots, y_N of the vector \underline{y} is expressed in terms of its component y_{N+1} , i.e.

$$y_k - \alpha^{N-k+1} y_{N+1} = \eta_k + \alpha \eta_{k+1} + \alpha^2 \eta_{k+2} + \dots + \alpha^{N-k} \eta_N, \quad (4.4.17)$$

$$k=1, 2, \dots, N-1,$$

whilst the system (4.4.15b) yields the result,

$$x_i = y_i + \alpha y_{i-1} + \dots + \alpha^{i-1} y_1, \quad i=1,2,\dots,N-1 \quad (4.4.18a)$$

$$x_N = y_N + \alpha y_{N-1} + \dots + \alpha^{N-1} y_1 \quad (4.4.18b)$$

$$-\alpha x_N = y_{N+1} \quad (4.4.18c)$$

Now, substitutions for y_N, y_{N-1}, \dots, y_1 in (4.4.18b) in terms of y_{N+1} , given by (4.4.17), together with (4.4.18c), will give us the result,

$$x_N = [\eta_N + \alpha \eta_{N-1} + \dots + \alpha^i \eta_{N-i} + \dots + \alpha^{N-1} \eta_1 - \alpha^{N+1} \eta_1 - \alpha^{N+2} \eta_2 - \dots - \alpha^{2N} \eta_N] / (1 - \alpha^{2N+2}) \quad (4.4.19)$$

Finally, the value of x_N having been determined, y_{N+1} is obtained from (4.4.18c), and then y_N, \dots, y_1 are computed by the back substitution process, i.e., we have

$$y_i = \eta_i + \alpha y_{i+1}, \quad \text{for } i=N, N-1, \dots, 2, 1,$$

and the solution vector \underline{x} is given by,

$$x_i = y_i + \alpha x_{i-1}, \quad \text{for } i=2, 3, \dots, N-1,$$

where, from (4.4.18a) $x_1 = y_1$.

The amount of work involved in this special method is $4N$ additions and $5N$ multiplications.

A similar elimination procedure is extended to the systems (4.4.16) after the determination of the elements of the matrix P , i.e. α and β . This may be done by solving a quartic equation for α (or β) which can be derived from the two equations of (4.4.13), Okolie (1978) and Newton's method to solve the quartic equation.

However, in analogy to (4.4.17), we can express the components \tilde{y}_1, \tilde{y}_N of the vector $\tilde{\underline{y}}$ in (4.4.16a), in terms of its last two components \tilde{y}_{N+1} and \tilde{y}_{N+2} , i.e.,

$$\tilde{y}_{N+1-j} = c_j + e_j \tilde{y}_{N+1} + e_j \beta \tilde{y}_{N+2}, \quad (4.4.20)$$

where $c_1 = \xi_N$, $c_2 = \xi_{N-1} + \alpha c_1$, $c_j = \eta_{N-j+1} + \alpha c_{j-1} + \beta c_{j-2}$, $j=3, 4, \dots, N$

and $e_1 = 1$, $e_2 = \alpha$, $e_j = \alpha e_{j-1} + \beta e_{j-2}$, $j=3, 4, \dots, N+1$.

From the system (4.4.16b), by a successive forward substitution process we have,

$$\tilde{x}_i = \sum_{j=1}^i e_j \tilde{y}_{i-j+1}, \quad i=1,2,\dots,N. \quad (4.4.21)$$

After considerable algebraic work (see Okolie (1978)) we end up with two equations in the two unknowns, \tilde{y}_{N+1} and \tilde{y}_{N+2} . These equations are,

$$\left. \begin{aligned} \tilde{y}_{N+2} R_1 &= -(R_2 \tilde{y}_{N+1} + R_3) \\ \tilde{y}_{N+2} S_1 &= -(S_2 \tilde{y}_{N+1} + S_3) \end{aligned} \right\} \quad (4.4.22)$$

where

$$R_1 = 1 + \beta^2 \sum_{j=1}^N c_j^2, \quad R_2 = \beta \sum_{j=1}^N e_j e_{j+1},$$

$$R_3 = \beta \sum_{j=1}^N e_j c_j, \quad S_1 = \beta^3 \sum_{j=1}^{N-1} e_j e_{j+1}^{-\alpha},$$

$$S_2 = \beta + \beta^2 \sum_{j=1}^{N-1} e_j e_{j+2}, \quad S_3 = \beta^2 \sum_{j=1}^{N-1} e_j c_{j+1},$$

and the e's and c's are given in (4.4.20).

After \tilde{y}_{N+1} and \tilde{y}_{N+2} have been determined from (4.4.22), then from the system (4.4.1a) we have,

$$\tilde{y}_{N-i+1} = \xi_{N-i+1} + \alpha \tilde{y}_{N-i+2} + \beta \tilde{y}_{N-i+3}, \quad i=1,2,\dots,N$$

and the vector solution \tilde{x} is given by (4.4.21) or from (4.4.16b)

$$\tilde{x}_1 = \tilde{y}_1, \quad \tilde{x}_2 = \tilde{y}_2 + \alpha \tilde{y}_1,$$

and

$$\tilde{x}_i = \tilde{y}_i + \alpha \tilde{x}_{i-1} + \beta \tilde{x}_{i-2}, \quad i=3,4,\dots,N.$$

The total amount of work required for solving (4.4.9) by the above approach, excluding the procedure of solving the equation (4.4.13), is of the order $13N$ multiplications and $11N$ additions. Okolie (1978) points out that this strategy is an unattractive method for the quindagonal matrix, as in (4.4.9), when the coefficient matrix is symmetric and has non-constant elements. This conclusion was based on the fact that the latter

case leads to the solution of N quartic equations for the determination of the elements of the factor matrices α_i, β_i .

However, our alternative strategy for handling the non-constant and non-symmetric matrix (of wide bandwidth) as in (4.4.1) has been discussed earlier for the factorization of the coefficient matrix, whilst the determination of the elements of the factor matrices and the solution of the coupled system will be considered next.

4.4.2 ALGORITHMIC SOLUTION OF A COUPLED SYSTEM

The solution of the original system (4.4.1) can be investigated by considering its alternative (4.4.4) which can be split into *underdetermined* and *overdetermined* linear systems by inserting an auxiliary vector \underline{y} i.e.

$$U\underline{y} = \underline{z} \quad (4.4.23a)$$

and
$$L\underline{x} = \underline{y}, \quad (4.4.23b)$$

where the rectangular matrices U and L are given by (4.4.3), the vector \underline{y} consists of $N+r$ components, i.e., $\underline{y} = [y_1, \dots, y_N, y_{N+1}, \dots, y_{N+r}]^T$, and the vectors \underline{z} and \underline{x} are both of N components as given in (4.4.1).

This strategy of splitting the given linear system into two systems differs from the strategies adopted in the previous method since, (i) the systems (4.4.23a) and (4.4.23b) are underdetermined and overdetermined by r respectively, and (ii) these two systems were solved separately in the previous algorithms whilst here they are treated in a coupled manner, so that the 'redundant' components of \underline{y} , i.e. y_{N+1}, \dots, y_{N+r} are determined first, then the remaining components y_1, \dots, y_N , and finally the components x_1, \dots, x_N of the solution vector are obtained.

We shall first consider the general matrix analysis of the solution of the coupled system (4.4.23). Following Evans and Hadjidimos (1979) we consider the partitioned forms of the matrices U and L which are given on the right-hand sides of (4.4.3) respectively and hence we may rewrite (4.4.23) in the form,

$$\begin{bmatrix} U_1 & V_1 \end{bmatrix} \begin{bmatrix} \hat{\underline{y}} \\ \tilde{\underline{y}} \end{bmatrix} = \underline{z} \quad (4.4.24a)$$

and

$$\begin{bmatrix} L_1 \\ K_1 \end{bmatrix} \underline{x} = \begin{bmatrix} \hat{\underline{y}} \\ \tilde{\underline{y}} \end{bmatrix} \quad (4.4.24b)$$

where the matrix U_1 is of size $(N \times N)$, V is $(N \times r)$, L_1 is $(N \times N)$ and K_1 is $(r \times N)$,

respectively, while the vectors $\hat{\underline{y}}$ and $\tilde{\underline{y}}$ are defined as,

$$\hat{\underline{y}}_1 = [y_1, \dots, y_N]^T, \quad \tilde{\underline{y}}_1 = [y_{N+1}, \dots, y_{N+r}]^T.$$

The two systems of (4.4.24) can be easily converted to the form, i.e.,

$$U_1 \hat{\underline{y}} + V_1 \tilde{\underline{y}} = \underline{z}, \quad (4.4.25a)$$

$$L_1 \underline{x} = \hat{\underline{y}}, \quad (4.4.25b)$$

$$\text{and} \quad K_1 \underline{x} = \tilde{\underline{y}}. \quad (4.4.25c)$$

If we substitute $\tilde{\underline{y}}$ from (4.4.25c) into (4.4.25a) then we have

$$U_1 \hat{\underline{y}} + V_1 K_1 \underline{x} = \underline{z}$$

$$\text{or} \quad \hat{\underline{y}} = U_1^{-1} [\underline{z} - V_1 K_1 \underline{x}]. \quad (4.4.26)$$

Therefore, after substituting $\hat{\underline{y}}_1$ from (4.4.26), we write (4.4.25b) as

$$L_1 \underline{x} = U_1^{-1} [\underline{z} - V_1 K_1 \underline{x}]$$

$$\text{or} \quad \underline{x} = (U_1 L_1)^{-1} [\underline{z} - V_1 K_1 \underline{x}]$$

and finally, by rearranging this result the solution vector \underline{x} may be expressed as,

$$\underline{x} = [I + (U_1 L_1)^{-1} V_1 K_1]^{-1} (U_1 L_1)^{-1} \underline{z}, \quad (4.4.27)$$

where I is the unit matrix of order N ; noting that the relation (4.4.27) is valid if the appropriate matrices are invertible.

However, our approach to determine the solution vector \underline{x} unlike (4.4.27) does not involve computation of inverse matrices, but rather involves partially solving a linear system of order r to evaluate the unknowns y_{N+1}, \dots, y_{N+r} , followed by the back and forward substitution process. This can be accomplished as illustrated in detail below.

The equivalent form of the systems (4.4.18a) and (4.4.18b) can be written respectively as,

where $\gamma_{k,i} = u_{k+1,i}$, for $k=0,1,\dots,r$, $i=1,2,\dots,N$, (4.4.29a)

and $\alpha_{0,i} = 1$,
 $\alpha_{k,i} = l_{k,i}$, $k=1,2,\dots,r$ } $i=1,2,\dots,N$. (4.4.29b)

An elimination process now disposes of the elements $\gamma_{k,j}$, $k=1,2,\dots,r$, $j=1,2,\dots,N-r$, and $\gamma_{k,N-i}$, $i=r-1,r-2,\dots,1$, $k=1,2,\dots,r$ of the system (4.4.28a) and the elements $\alpha_{k,j}$, $k=1,2,\dots,r$, $j=1,2,\dots,N$ of the system (4.4.28b); viz.

(I) For the system (4.4.28a),

we leave the N^{th} equation unchanged, then multiply the N^{th} equation by $\frac{-\gamma_{1,N-1}}{\gamma_{0,N}}$ and add to the $(N-1)^{\text{th}}$ equation to obtain the new $(N-1)^{\text{th}}$ equation and thus eliminate the elements $\gamma_{1,N-1}$. We now multiply the new $(N-1)^{\text{th}}$ equation by $\frac{-\gamma_{1,N-2}}{\gamma_{0,N-1}}$ and the N^{th} equation by $\frac{-\gamma_{2,N-2}}{\gamma_{0,N}}$ and add to the $(N-2)^{\text{th}}$ equation to obtain the new $(N-2)^{\text{th}}$ equation and thus eliminate the elements $\gamma_{1,N-2}$ and $\gamma_{2,N-2}$, so, generally, to obtain the new $(N-k)^{\text{th}}$ equation, $k=1,2,\dots,r-1$, we multiply the $(N-k)^{\text{th}}$ equation, $k=1,2,\dots,r-1$, we multiply the $(N-j+1)^{\text{th}}$ equation by $\frac{-\gamma_{j,N-k}}{\gamma_{0,N-k+j}}$, $j=1,2,\dots,k$ and add these (k) equations to the $(N-k)^{\text{th}}$ equation, thus eliminating the elements $\gamma_{1,N-k}, \gamma_{2,N-k}, \dots, \gamma_{k,N-k}$. We proceed now to eliminate r elements each time, so obtaining a new j^{th} equation, where j runs from $N-r$ to 1. We multiply the $(j+k)^{\text{th}}$ equation, $k=1,2,\dots,r$ by $\frac{-\gamma_{k,j}}{\gamma_{0,j+k}}$ and add to the j^{th} equation to obtain the new j^{th} equation, and thus eliminate the elements $\gamma_{1,j}, \gamma_{2,j}, \dots, \gamma_{r,j}$.

The algebraic formulation of the above elimination procedure can be arranged as follows:

Let
$$g_{t,N+i} = \begin{cases} -1, & \text{for } i=t \\ 0, & \text{otherwise} \end{cases} \quad t,i=1,2,\dots,r, \quad (4.4.30a)$$

and

$$g_{k,j} = \sum_{s=1}^r (-\gamma_{s,j}/\gamma_{0,j+s}) g_{k,j+s}, \quad k=1,2,\dots,r, \quad j=N,N-1,\dots,1, \quad (4.4.30b)$$

where $g_{k,t}$ for $t > N$ are given by (4.4.30a), and

$$\gamma_{0,i} \equiv 1 \text{ for all } i > N,$$

$$\tilde{z}_j = \sum_{s=1}^r (-\gamma_{s,j}/\gamma_{0,j+s}) \tilde{z}_{j+s} + z_j, \quad j=N,N-1,\dots,1, \quad (4.4.30c)$$

where $\tilde{z}_t \equiv 0$ for all $t > N$.

Thus, from the described elimination procedure which is formulated by (4.4.30), the system (4.4.28a) becomes

$$\begin{bmatrix} \gamma_{0,1} & & & & & \\ & \gamma_{0,2} & & & & \\ & & 0 & & & \\ & & & \ddots & & \\ & 0 & & & \gamma_{0,N-1} & \\ & & & & & \gamma_{0,N} \end{bmatrix} \begin{bmatrix} g_{1,1} & g_{2,1} & \dots & g_{r-1,1} & g_{r,1} \\ g_{1,2} & g_{2,2} & \dots & g_{r-1,2} & g_{r,2} \\ \vdots & \vdots & & \vdots & \vdots \\ g_{1,N-1} & g_{2,N-1} & \dots & g_{r-1,N-1} & g_{r,N-1} \\ g_{1,N} & g_{2,N} & \dots & g_{r-1,N} & g_{r,N} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ y_{N+1} \\ \vdots \\ y_{N+r} \end{bmatrix} = \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_N \end{bmatrix} \quad (4.4.31)$$

Then, the system (4.4.31) immediately yields the result,

$$\left. \begin{aligned} y_1 &= \frac{1}{\gamma_{0,1}} (\tilde{z}_1 - y_{N+1} g_{1,1} - y_{N+2} g_{2,1} - \dots - y_{N+r-1} g_{r-1,1} - y_{N+r} g_{r,1}), \\ y_2 &= \frac{1}{\gamma_{0,2}} (\tilde{z}_2 - y_{N+1} g_{1,2} - y_{N+2} g_{2,2} - \dots - y_{N+r-1} g_{r-1,2} - y_{N+r} g_{r,2}), \\ &\dots \\ y_N &= \frac{1}{\gamma_{0,N}} (\tilde{z}_N - y_{N+1} g_{1,N} - y_{N+2} g_{2,N} - \dots - y_{N+r-1} g_{r-1,N} - y_{N+r} g_{r,N}), \end{aligned} \right\} \quad (4.4.32a)$$

equation by $\frac{-\alpha_{1,1}}{\alpha_{0,1}}$ and add to the 2nd equation to obtain the new 2nd equation and thus eliminate the element $\alpha_{1,1}$. We now multiply the new 2nd equation by $\frac{-\alpha_{1,2}}{\alpha_{0,2}}$ and the first equation by $\frac{-\alpha_{2,1}}{\alpha_{0,1}}$ and add to the 3rd equation, and thus eliminate the elements $\alpha_{1,2}$ and $\alpha_{2,1}$, so, generally, to obtain a new j^{th} equation, $j=2, \dots, r$ we multiply the k^{th} equation by $\frac{-\alpha_{k,j-k}}{\alpha_{0,j-k}}$, $k=1, 2, \dots, j-1$ and add these $(j-1)$ equations to the j^{th} equation, thus eliminate the elements $\alpha_{k,j-k}$, $k=1, 2, \dots, j-1$, for $j=2, \dots, r$.

We now continue the elimination procedure so that in the following i steps, where i runs from 1 up to N , we can eliminate r elements per i^{th} step; to obtain a new i^{th} equation we multiply the $(i-k)^{\text{th}}$ equation by $\frac{-\alpha_{k,i-k}}{\alpha_{0,i-k}}$, $k=1, 2, \dots, r$ and add these (r) equations to the i^{th} equation, thus eliminating the elements $\alpha_{1,i-1}, \alpha_{2,i-2}, \dots, \alpha_{r,i-r}$.

Finally, the elimination procedure carries over to the remaining equations, so that we multiply the $(N+j-k)^{\text{th}}$ equation by $\frac{-\alpha_{k,N+j-k}}{\alpha_{0,N+j-k}}$, $k=j, j+1, \dots, r$ and $j=1, 2, \dots, r$ and add these $(k-j+1)$ equations to the $(N+j)^{\text{th}}$ equation to obtain a new $(N+j)^{\text{th}}$ equation, and thus eliminate the elements $\alpha_{k,N+j-k}$, $j=j, \dots, r$, $j=1, 2, \dots, r$; consequently the elimination process for the system (4.4.28b) is complete.

The implications of the above elimination procedure on the system (4.4.28b) are that (i) the $(N+r) \times N$ rectangular coefficient matrix is left with the diagonal elements $\alpha_{0,1}, \alpha_{0,2}, \dots, \alpha_{0,N}$, and (ii) its $(N+r)$ -component vector on the right-hand side is modified; we shall denote the new components of this vector by $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N, \dots, \tilde{y}_{N+r}$, i.e.,

$$\begin{bmatrix} \alpha_{0,1} & & & & & \\ & \alpha_{0,2} & & & & \\ & & \ddots & & & \\ & & & \alpha_{0,N-1} & & \\ & & & & \alpha_{0,N} & \\ \hline & & & & & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_N \\ \tilde{y}_{N+1} \\ \vdots \\ \tilde{y}_{N+r} \end{bmatrix} \quad (4.4.36)$$

The components $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_{N+r}$ of the vector on the right-hand side of the system (4.4.36) can be expressed as follows, according to the elimination procedure described earlier:

$$\left. \begin{aligned} \tilde{y}_1 &= y_1, \\ \tilde{y}_j &= y_j + \sum_{k=1}^{j-1} \left(\frac{-\alpha_{k,j-k}}{\alpha_{0,j-k}} \right) \tilde{y}_{j-k}, \quad j=2, \dots, r, \\ \tilde{y}_i &= y_i + \sum_{k=1}^r \left(\frac{-\alpha_{k,i-k}}{\alpha_{0,i-k}} \right) \tilde{y}_{i-k}, \quad i=r+1, \dots, N, \\ \text{and} \\ \tilde{y}_{N+j} &= y_{N+j} + \sum_{k=j}^r \left(\frac{-\alpha_{k,N+j-1}}{\alpha_{0,N+j-k}} \right) \tilde{y}_{N+j-k}, \quad j=1, 2, \dots, r, \end{aligned} \right\} \quad (4.4.37)$$

where y_1, y_2, \dots, y_{N+r} are as given in the system (4.4.28b).

Further, in order to express \tilde{y}_j , $j=1, 2, \dots, N$ in terms of y_k , $k=1, 2, \dots, j$ and \tilde{y}_{N+k} , $k=1, 2, \dots, r$ in terms of y_1, \dots, y_N , a large amount of tedious algebraic work is necessary. Alternatively it may be easier to re-formulate (4.4.37) by introducing some extra notations which we now introduce.

We define the quantities $T_{j,k}$, $j=1, 2, \dots, N+r$ for $k=0, 1, \dots, r$ and $c_{j,i}$, $j=1, 2, \dots, N+r$, for $i=0, 1, \dots, N$ as follows,

$$\left. \begin{aligned} T_{j,0} &= 1, \\ \text{and } T_{j,k} &= -\frac{\alpha_{k,j-k}}{\alpha_{0,j-k}} \text{ for } 1 \leq j-k \leq N, k=1,2,\dots,r \end{aligned} \right\} j=1,2,\dots,N+r, \quad (4.4.38a)$$

$$c_{j,0} = 1, \text{ for } j=1,2,\dots,N+r,$$

$$c_{j,i} = \left\{ \begin{aligned} &1, \text{ for } i=0 \\ &\sum_{k=1}^s T_{j,k} c_{j-k,i-k}, \text{ for } s = \begin{cases} i, & \text{for } i < r \\ r, & \text{otherwise} \end{cases} \text{ provided } 2 \leq j \leq N+1 \\ &\sum_{k=1}^s T_{j,r-d+k} c_{N+1-k,i-k}, \text{ for } s = \begin{cases} i, & \text{for } i < d \\ d, & \text{otherwise} \end{cases} \text{ where } d=N+r+1-j \\ &\text{for } i=1,2,\dots,N \text{ provided } j \geq N+2 \end{aligned} \right\} \\ j=1,2,\dots,N+r. \quad (4.4.38b)$$

where the quantities $T_{j,k}$ are given by (4.4.38a).

It can be seen that the values of the quantities $c_{j,i}$ in (4.4.38b) are computed recursively in Figure 4.4.1, where we have set up a computational scheme to illustrate this relation.

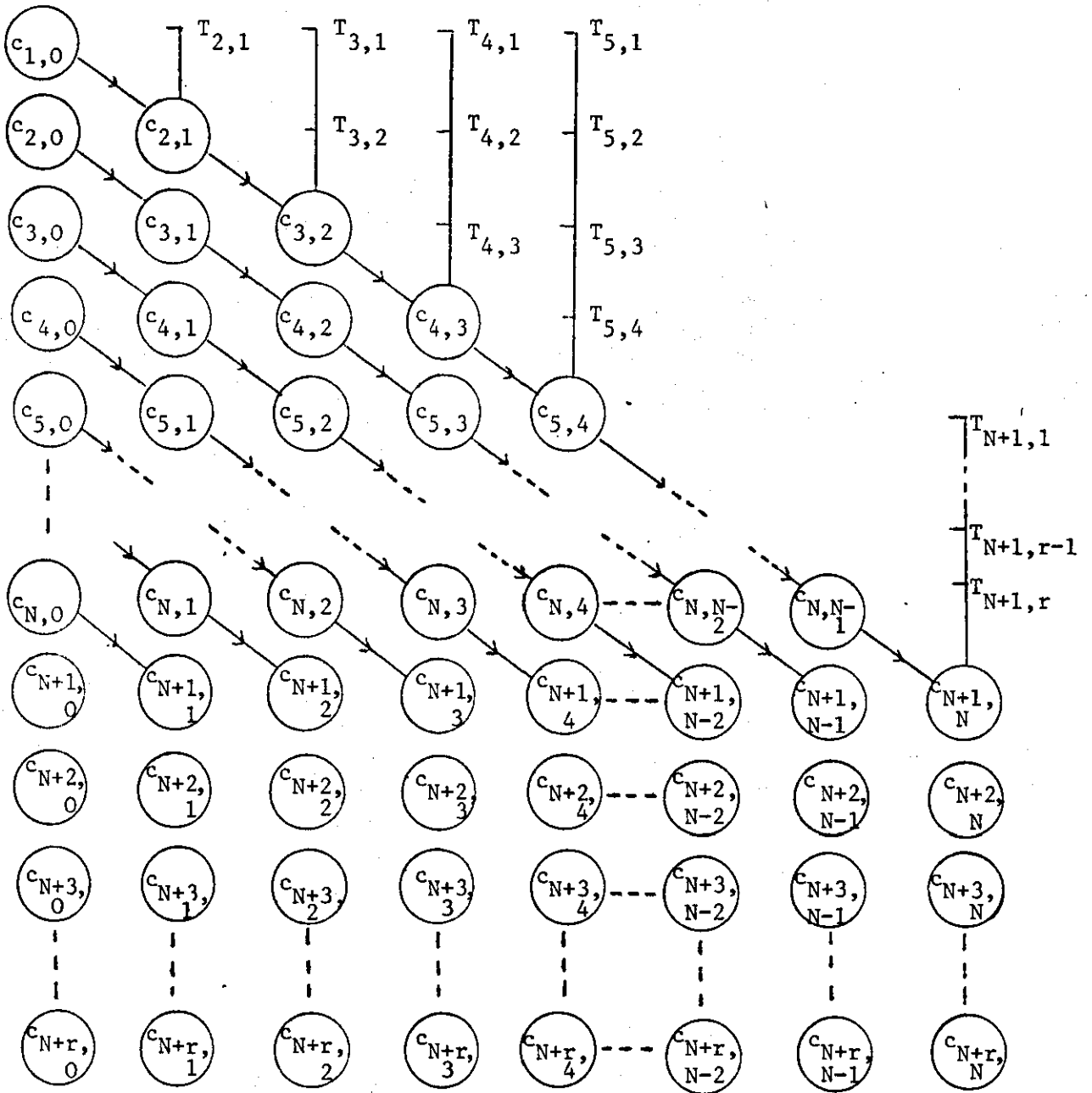


FIGURE 4.4.1: Computation of the $c_{j,i}$ of the relation (4.4.38)

For example to obtain the element $c_{5,4}$ (r is assumed to be ≥ 5), we add together the products of the elements on the same diagonal ($j < 5$, $i < 4$), multiply by $T_{j,i}$ on the column i ($i=4$, in this case), i.e.,

$$c_{5,4} = c_{4,3}T_{5,4} + c_{3,2}T_{5,3} + c_{2,1}T_{5,2} + c_{1,0}T_{5,1};$$

and $c_{j,i}$, where $r+1 \leq j \leq N+1$, is derived from the r preceding elements on the same diagonal, whilst the elements $c_{j,i}$ on each of the rows $N+2, N+3, \dots, N+r$ are associated with appropriate elements, given by the relation of (4.4.38b), on the rows $N, N-1, \dots, N-r+1$.

Therefore, taking the assumption (4.4.38) we can rewrite the relations (4.4.37) in the form,

$$\left. \begin{aligned} \tilde{y}_1 &= c_{1,0} y_1, \\ \tilde{y}_2 &= c_{2,0} y_2 + c_{2,1} y_1, \\ \tilde{y}_3 &= c_{3,0} y_3 + c_{3,1} y_2 + c_{3,2} y_1, \\ \tilde{y}_4 &= c_{4,0} y_4 + c_{4,1} y_3 + c_{4,2} y_2 + c_{4,3} y_1, \\ &\dots \\ \tilde{y}_N &= c_{N,0} y_N + c_{N,1} y_{N-1} + c_{N,2} y_{N-2} + \dots + c_{N,N-1} y_1 \end{aligned} \right\} \quad (4.4.39a)$$

and

$$\left. \begin{aligned} \tilde{y}_{N+1} &= c_{N+1,0} y_{N+1} + c_{N+1,1} y_N + c_{N+1,2} y_{N-1} + \dots + c_{N+1,N} y_1, \\ \tilde{y}_{N+2} &= c_{N+2,0} y_{N+2} + c_{N+2,1} y_N + c_{N+2,2} y_{N-1} + \dots + c_{N+2,N} y_1, \\ &\dots \\ \tilde{y}_{N+r} &= c_{N+r,0} y_{N+r} + c_{N+r,1} y_N + c_{N+r,2} y_{N-1} + \dots + c_{N+r,N} y_1. \end{aligned} \right\} \quad (4.4.39b)$$

Moreover, we can write the system (4.4.39) in matrix form (noting that we substitute $c_{j,0}$ for 1, $j=1,2,\dots,N$ as defined in (4.4.38b)), i.e. from the relations (4.4.39b) we have,

$$\begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{y}_3 \\ \vdots \\ \tilde{y}_N \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ c_{2,1} & 1 & & & \\ & c_{3,2} & c_{3,1} & 1 & \\ & c_{4,3} & & & \ddots \\ & \vdots & & & c_{N-1,N-2} \\ c_{N,N-1} & & & & c_{N,3} & c_{N,2} & c_{N,1} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix} \quad (4.4.40a)$$

or in abbreviated form,

$$\tilde{\underline{y}} = G \hat{\underline{y}}, \quad (4.4.40b)$$

where the vector $\tilde{\underline{y}} = [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N]^T$,

$\hat{\underline{y}}$ is as defined in (4.4.35b),

and

G is an $(N \times N)$ matrix.

Now, recalling the system (4.4.36) we immediately obtain the following result,

$$\left. \begin{aligned} x_1 &= \frac{1}{\alpha_{0,1}} \tilde{y}_1, \\ x_2 &= \frac{1}{\alpha_{0,2}} \tilde{y}_2, \\ &\text{-----} \\ x_N &= \frac{1}{\alpha_{0,N}} \tilde{y}_N \end{aligned} \right\} \quad (4.4.41a)$$

and $\tilde{y}_{N+k} = 0$, for $k=1,2,\dots,r$. (4.4.41b)

By substituting the values of $\tilde{y}_{N+1}, \dots, \tilde{y}_{N+r}$ into the left-hand side of the relations (4.4.39b) we have

$$\left. \begin{aligned} 0 &= c_{N+1,0} y_{N+1} + c_{N+1,1} y_N + c_{N+1,2} y_{N-1} + \dots + c_{N+1,N} y_1, \\ 0 &= c_{N+2,0} y_{N+2} + c_{N+2,1} y_N + c_{N+2,2} y_{N-1} + \dots + c_{N+2,N} y_1, \\ &\text{-----} \\ 0 &= c_{N+r,0} y_{N+2} + c_{N+r,1} y_N + c_{N+r,2} y_{N-1} + \dots + c_{N+r,N} y_1. \end{aligned} \right\} \quad (4.4.42)$$

By setting $c_{N+k,0}=1$, for $k=1,2,\dots,r$ as defined in (4.4.38b) and re-arranging the above equations, then (4.4.42) can be written in the matrix form,

$$- \begin{bmatrix} y_{N+1} \\ y_{N+2} \\ y_{N+3} \\ \vdots \\ y_{N+r} \end{bmatrix} = \begin{bmatrix} c_{N+1,N} & \text{---} & \text{---} & \text{---} & c_{N+1,3} & c_{N+1,2} & c_{N+1,1} \\ c_{N+2,N} & \text{---} & \text{---} & \text{---} & c_{N+2,3} & c_{N+2,2} & c_{N+2,1} \\ c_{N+3,N} & \text{---} & \text{---} & \text{---} & c_{N+3,3} & c_{N+3,2} & c_{N+3,1} \\ \vdots & & & & \vdots & \vdots & \vdots \\ c_{N+r,N} & \text{---} & \text{---} & \text{---} & c_{N+r,3} & c_{N+r,2} & c_{N+r,1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_r \\ \vdots \\ y_N \end{bmatrix} \quad (4.4.43a)$$

or in a compact form,

$$-\underline{\tilde{\eta}} = H \underline{\hat{\eta}} \quad (4.4.43b)$$

where the vectors $\underline{\tilde{\eta}}$ and $\underline{\hat{\eta}}$ are as defined in (4.4.35b) and the rectangular matrix H is of size $(r \times N)$.

Now, the substitution of $\hat{\underline{n}}$ from (4.4.35b) in the equation (4.4.53b) yields the result,

$$\hat{\underline{n}} = -H[\underline{\phi} + \psi \tilde{\underline{n}}] \quad (4.4.44)$$

The sizes of the matrices H, ϕ, ψ and $\tilde{\underline{n}}$ as defined earlier are $(r \times N), (N \times 1), (N \times r)$ and $(r \times 1)$ respectively, and so the multiplication of the appropriate matrices in (4.4.44) is valid; hence with a slight rearrangement (4.4.44) can be written as,

$$(I + H\psi) \tilde{\underline{n}} = -H\underline{\phi}, \quad (4.4.45a)$$

(where I is the unit matrix of size $(r \times r)$),

$$\text{or} \quad B \tilde{\underline{n}} = -\underline{d}, \quad (\text{a linear system of order } r) \quad (4.4.45b)$$

$$\text{where} \quad B = I + H\psi \quad (4.4.46a)$$

$$\text{and} \quad \underline{d} = H\underline{\phi} \quad (4.4.46b)$$

The elements of matrix $B = [b_{i,j}]$, $i, j = 1, 2, \dots, r$ and the elements of the vector $\underline{d} = [d_i]$, $i = 1, 2, \dots, r$ can be determined from (4.4.46) and given by

$$b_{i,j} = \delta + \sum_{k=1}^N c_{N+i, N+1-k} \psi_{k,j}, \quad \delta = \begin{cases} 1 & \text{for } i=j \\ 0 & \text{otherwise} \end{cases} \quad (4.4.47a)$$

$$\text{and} \quad d_i = \sum_{k=1}^N c_{N+i, N+1-k} \phi_k, \quad i = 1, 2, \dots, r. \quad (4.4.47b)$$

Therefore, having determined the elements by d_i from (4.4.47), the linear system (4.4.45b) can be solved to yield the vector $\tilde{\underline{n}}$ (or the values y_{N+1}, \dots, y_{N+r}) thus enabling the computation of the values y_1, y_2, \dots, y_N from the relation (4.4.35). We then apply a forward substitution process to the system (4.4.40) to evaluate $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N$. Finally, the solution vector components x_1, x_2, \dots, x_N can be determined from the relations (4.4.41a).

It might be unnecessary to compute the values y_1, \dots, y_N and $\tilde{y}_1, \dots, \tilde{y}_N$ explicitly. Instead we may write from the relations (4.4.35b) and (4.4.40b),

$$\tilde{\underline{\eta}} = \underline{\hat{G}} \underline{\eta} = G[\underline{\phi} + \underline{\psi} \tilde{\underline{\eta}}], \quad (4.4.48)$$

and from (4.4.41a) we have

$$\underline{x} = D^{-1} \tilde{\underline{\eta}}, \quad D \equiv \text{diag}(\alpha_{0,1}, \dots, \alpha_{0,N}) . \quad (4.4.49)$$

Hence, by substituting $\tilde{\underline{\eta}}$ from (4.4.48) into (4.4.49), the solution vector is given by,

$$\underline{x} = D^{-1} G[\underline{\phi} + \underline{\psi} \tilde{\underline{\eta}}] . \quad (4.4.50)$$

Moreover, because $\alpha_{0,i}=1$, $i=1,2,\dots,N$ by the assumption (4.4.29b) the elements of the diagonal matrix D are 1's, i.e. D is a unit matrix and hence the final expression of the solution vector \underline{x} given by (4.4.50) becomes,

$$\underline{x} = G(\underline{\phi} + \underline{\psi} \tilde{\underline{\eta}}) , \quad (4.4.51)$$

where the matrices G ($N \times N$) and $\underline{\psi}$ ($N \times r$) are defined in (4.4.40) and (4.4.35) respectively; the vector $\underline{\phi}$ (of N components) is given in (4.4.35) and $\tilde{\underline{\eta}}$ is the solution of the linear system (4.4.45).

Finally, the computational complexity of the algorithm for the solution of (4.4.1) is approximately of order: (provided that the system (4.4.28a) is normalized first) $O[\frac{r}{2}(N-r-1)(N-1) + (r(r+5)+1)N]$ multiplications and $O[\frac{r}{2}(N-r-1)(N-1) + (r(r+5)-1)N]$ additions with the predetermination of the elements of the matrices U and L . This result does not apply for the case $r=1$ (i.e., the tridiagonal case), where the order can be drastically reduced to $7N$ multiplications and $4N$ additions (see Appendix B). Also it does not apply to the case $r=2$ and when the coefficient matrix is symmetric and constant (see the system (4.4.9a)).

$$h_1 = \hat{r}, \quad (4.4.55^*)$$

$$h_{s+1} = \hat{r}-s, \quad s=1,2,\dots,\hat{r}-1 \text{ (for } \hat{r} \geq 2) \quad (4.4.55^{**})$$

$$\text{and} \quad h_{\hat{r}+s} = r-s+1, \quad s=1,2,\dots,r-m \text{ (for } \hat{r}+s \leq r), \quad (4.4.55^{***})$$

where \hat{r} and m are defined in (4.4.54).

Hence the elements of V_1 in (4.4.52b) can be chosen in terms of the elements of U_1 in (4.4.52a) as follows,

$$u_{k,N-j+1} = u_{k,h_j}, \quad j=1,2,\dots,k-1, k=r+1, r, \dots, 2, \quad (4.4.56a)$$

and the elements of K_1 in (4.4.53b) can be chosen in terms of the elements of L_1 in (4.4.53a) as follows:

$$l_{k,N-j+1} = l_{k,h_j}, \quad j=1,2,\dots,k, k=r, r-1, \dots, 1, \quad (4.4.56b)$$

where the h_j are given in (4.4.55).

To clarify the strategy of selecting the elements of V_1 and K_1 we consider the following examples (for $r=1,2,3$, or 4):

(1) For $r=1$:

From (4.4.55*) we have

$$h_1 = 1,$$

then, it follows from (4.4.56) that

$$\left. \begin{aligned} u_{2,N} &= u_{2,1} \\ \text{and} \quad l_{1,N} &= l_{1,1} \end{aligned} \right\} \quad (4.4.57)$$

(2) For $r=2$:

In this case, m is either zero or one. Hence, from (4.4.54) \hat{r} is 2 or 1 respectively:

(i) For $\hat{r}=2$ ($m=0$):

from (4.4.55*) and (4.4.55**) we have

$$h_1 = 2, \quad h_2 = 1,$$

then, it follows from (4.4.56) that,

$$\begin{array}{l}
 u_{3,N} = u_{3,2} \\
 u_{3,N-1} = u_{3,1}, \quad u_{2,N} = u_{2,2}, \\
 \text{and} \quad l_{2,N} = l_{2,2} \\
 l_{2,N-1} = l_{2,1}, \quad l_{1,N} = l_{1,2}
 \end{array}
 \quad \left. \vphantom{\begin{array}{l} u_{3,N} = u_{3,2} \\ u_{3,N-1} = u_{3,1}, \quad u_{2,N} = u_{2,2}, \\ l_{2,N} = l_{2,2} \\ l_{2,N-1} = l_{2,1}, \quad l_{1,N} = l_{1,2} \end{array}} \right\} (4.4.58a)$$

(ii) For $\hat{r}=1$ ($m=1$),

from (4.4.55*) and (4.4.55**) we have

$$h_1 = 1, \quad h_2 = 2,$$

then, it follows from (4.4.56), that

$$\begin{array}{l}
 u_{3,N} = u_{3,1}, \\
 u_{3,N-1} = u_{3,2}, \quad u_{2,N} = u_{2,1}, \\
 \text{and} \quad l_{2,N} = l_{2,1}, \\
 l_{2,N-1} = l_{2,2}, \quad l_{1,N} = l_{1,1}.
 \end{array}
 \quad \left. \vphantom{\begin{array}{l} u_{3,N} = u_{3,1}, \\ u_{3,N-1} = u_{3,2}, \quad u_{2,N} = u_{2,1}, \\ l_{2,N} = l_{2,1}, \\ l_{2,N-1} = l_{2,2}, \quad l_{1,N} = l_{1,1}. \end{array}} \right\} (4.4.58b)$$

(3) For $r=3$:

In this case m is either 0,1 or 2 consequently from (4.4.54) the corresponding values of \hat{r} are 3,1,2:

(i) For $r=3$ ($m=0$):

from (4.4.55*) and (4.4.55**) we have,

$$h_1 = 3, \quad h_2 = 2, \quad h_3 = 1,$$

then, it follows from (4.4.56) that,

$$\begin{array}{l}
 u_{4,N} = u_{4,3}, \quad u_{4,N-1} = u_{4,2}, \quad u_{4,N-2} = u_{4,1}, \\
 u_{3,N} = u_{3,3}, \quad u_{3,N-1} = u_{3,2}, \\
 \text{and,} \quad u_{2,N} = u_{2,3}, \\
 l_{3,N} = l_{3,3}, \quad l_{3,N-1} = l_{3,2}, \quad l_{3,N-2} = l_{3,1}, \\
 l_{2,N} = l_{2,3}, \quad l_{2,N-1} = l_{2,2}, \\
 l_{1,N} = l_{1,3}.
 \end{array}
 \quad \left. \vphantom{\begin{array}{l} u_{4,N} = u_{4,3}, \quad u_{4,N-1} = u_{4,2}, \quad u_{4,N-2} = u_{4,1}, \\ u_{3,N} = u_{3,3}, \quad u_{3,N-1} = u_{3,2}, \\ u_{2,N} = u_{2,3}, \\ l_{3,N} = l_{3,3}, \quad l_{3,N-1} = l_{3,2}, \quad l_{3,N-2} = l_{3,1}, \\ l_{2,N} = l_{2,3}, \quad l_{2,N-1} = l_{2,2}, \\ l_{1,N} = l_{1,3}. \end{array}} \right\} (4.4.59a)$$

(ii) For $\hat{r}=1$ ($m=1$),

from (4.4.55*) and (4.4.55***) we have

$$h_1 = 1, \quad h_2 = 3, \quad h_3 = 2,$$

then, it follows from (4.4.56), that

$$\left. \begin{aligned} u_{4,N} &= u_{4,1}, \quad u_{4,N-1} = u_{4,3}, \quad u_{4,N-2} = u_{4,2}, \\ u_{3,N} &= u_{3,1}, \quad u_{3,N-1} = u_{3,3}, \\ u_{2,N} &= u_{2,1}, \\ \text{and} \\ l_{3,N} &= l_{3,1}, \quad l_{3,N-1} = l_{3,3}, \quad l_{3,N-2} = l_{3,2}, \\ l_{2,N} &= l_{2,1}, \quad l_{2,N-1} = l_{2,3}, \\ l_{1,N} &= l_{1,1}. \end{aligned} \right\} (4.4.59b)$$

(iii) For $\hat{r}=2$ ($m=2$),

from (4.4.55) we have,

$$h_1 = 2, \quad h_2 = 1, \quad h_3 = 3,$$

then, it follows from (4.4.56), that,

$$\left. \begin{aligned} u_{4,N} &= u_{4,2}, \quad u_{4,N-1} = u_{4,1}, \quad u_{4,N-2} = u_{4,3}, \\ u_{3,N} &= u_{3,2}, \quad u_{3,N-1} = u_{3,1}, \\ u_{2,N} &= u_{2,2}, \\ \text{and} \\ l_{3,N} &= l_{3,2}, \quad l_{3,N-1} = l_{3,1}, \quad l_{3,N-2} = l_{3,3}, \\ l_{2,N} &= l_{2,2}, \quad l_{2,N-1} = l_{2,1}, \\ l_{1,N} &= l_{1,2}. \end{aligned} \right\} (4.4.59c)$$

Subsequently we can rewrite the non-zero elements of the matrices V_1 and K_1 , say for $r=3$ by virtue of (4.4.59) as follows:

(a) the non-zero elements of V_1 as they are located in (4.4.52b), are:

$$\begin{array}{ccc} u_{4,1} & u_{4,2} & u_{4,3} \\ u_{3,2} \quad u_{4,2} & u_{3,3} \quad u_{4,3} & u_{3,1} \quad u_{4,1} \\ u_{2,3} \quad u_{3,3} \quad u_{4,3} & u_{2,1} \quad u_{3,1} \quad u_{4,1} & u_{2,2} \quad u_{3,2} \quad u_{4,2} \\ (m=0, \hat{r}=3) & (m=1, \hat{r}=1) & (m=2, \hat{r}=2) \end{array}$$

In fact, the purpose of adopting this strategy of choosing the elements of the matrices V_1 and K_1 in terms of the elements of U_1 and L_1 respectively is to enable us to express $l_{r,1}, l_{r,2}, \dots, l_{r,r}$ of the system (4.4.5) in the form of infinite periodic continued fractions as was done previously in the periodic matrix case (algorithm FICM2, subsection 4.3.3).

For example, if $r=1$ (i.e., the case where (4.4.1) is a tridiagonal system), then from the equations (4.4.7a) and using (4.4.57) we are able to express $l_{1,1}$ as an infinite periodic fraction similar to the periodic case in FICM2 (cf. the continued fraction (4.3.22)), i.e.,

$$a_{1,1} l_{1,1} = \frac{a_{-1,2} a_{1,1}}{a_{0,2}} \frac{a_{-1,3} a_{1,2}}{a_{0,3}} \frac{a_{-1,4} a_{1,3}}{a_{0,4}} \dots \frac{a_{-1,N} a_{1,N-1}}{a_{0,N} a_{2,1} l_{1,1}} \quad (4.4.61a)$$

or

$$a_{1,1} l_{1,1} = \frac{\alpha_1}{\beta_1} \frac{\alpha_2}{\beta_2} \dots \frac{\alpha_{N-1}}{\beta_{N-1}} \frac{\alpha_1}{\beta_1} \frac{\alpha_2}{\beta_2} \dots \frac{\alpha_{N-1}}{\beta_{N-1}} \dots, \quad (4.4.61b)$$

where

$$\left. \begin{aligned} \alpha_i &= a_{-1,i+1} a_{1,i} \\ \text{and } \beta_i &= a_{0,i+1} \end{aligned} \right\} \quad i=1,2,\dots,N-1,$$

Also, for $r=2$ (i.e. the case where (4.4.1) is a quindigonal system), from equations (4.4.7b) and using either (4.4.58a) or (4.4.58b) we are able to express $l_{2,1}, l_{2,2}$ as an infinite continued fraction (cf. p.154), i.e.,

(i) if $m=0$ in (4.4.54), i.e. N is even, then from the equations (4.4.7b) and (4.4.58a), $l_{2,1}$ and $l_{2,2}$ are expressed as,

$$a_{2,1} l_{2,1} = \frac{a_{-2,3} a_{2,1}}{(a_{0,3} l_{1,3} u_{2,3})} \frac{a_{-2,5} a_{2,3}}{(a_{0,5} l_{1,5} u_{2,5})} \frac{a_{-2,7} a_{2,5}}{(a_{0,7} l_{1,7} u_{2,7})} \dots \frac{a_{-2,2n+1} a_{2,2n-1}}{a_{0,2n-1} l_{1,2n+1} u_{2,2n+1} a_{2,1} l_{2,1}} \quad (4.4.62a)$$

where $n = \frac{N}{2} - 1$

and $u_{3,N-1}$ has been replaced by $u_{3,1} = a_{2,1}$ and

$l_{2,N-1}$ by $l_{2,1}$ by virtue of (4.4.58a),

and

$$a_{2,2} l_{2,2} = \frac{a_{-2,4} a_{2,2}}{(a_{0,4}^{-l_{1,4}} u_{2,4})^{-\frac{a_{-2,6} a_{2,4}}{(a_{0,6}^{-l_{1,6}} u_{2,6})^{-\frac{a_{-2,8} a_{2,6}}{(a_{0,8}^{-\dots})^{-\dots}}}}}} \dots \frac{a_{-2,2n+2} a_{2,2n}}{(a_{0,2n+2}^{-l_{1,2n+2}} u_{2,2n+2})^{-a_{2,2} l_{2,2}}} \quad (4.4.62b)$$

where $l_{1,2n+2} l_{1,N}$ and $u_{2,2n+2} \equiv u_{2,N} = u_{2,2}$ by virtue of (4.4.58a), and

$u_{3,N}$ has been replaced by $u_{3,2} = a_{2,2}$ and $l_{2,N}$ by $l_{2,2}$.

(ii) if $m=1$ in (4.4.54), i.e. N is *odd*, then from the equations (4.4.7b)

and (4.4.58b), $l_{2,1}$ and $l_{2,2}$ are expressed as,

$$a_{2,1} l_{2,1} = \frac{a_{-2,3} a_{2,1}}{(a_{0,3}^{-l_{1,3}} u_{2,3})^{-\frac{a_{-2,5} a_{2,3}}{(a_{0,5}^{-l_{1,5}} u_{2,5})^{-\frac{a_{-2,7} a_{2,5}}{(a_{0,7}^{-\dots})^{-\dots}}}}}} \dots \frac{a_{-3,2n+1} a_{2,2n-1}}{(a_{0,2n+1}^{-l_{1,2n+1}} u_{2,2n+1})^{-a_{2,1} l_{2,1}}} \quad (4.4.62c)$$

where n in (4.4.62c) is defined as $n = \frac{N-1}{2}$; hence,

$$l_{1,2n+1} \equiv l_{1,N} = l_{1,1} \text{ and } u_{2,2n+1} \equiv u_{2,N} = u_{2,1}$$

by virtue of (4.4.58b), and $u_{3,N}$ has been replaced by $u_{3,1} = u_{2,1}$ and

$l_{2,N}$ by $l_{2,1}$.

and

$$a_{2,2}^{l_{2,2}} = \frac{a_{-2,4} a_{2,2}}{(a_{0,4}^{-l_{1,4}} u_{2,4}) - \frac{a_{-2,6} a_{2,4}}{(a_{0,6}^{-l_{1,6}} u_{2,6}) - \frac{a_{-2,8} a_{2,6}}{(a_{0,8}^{-l_{1,8}} \dots) - \dots}} \dots \frac{a_{-2t+2} a_{2,2t}}{(a_{0,2t+2}^{-l_{1,2t+2}} u_{2,2t+2}) - a_{2,2}^{l_{2,2}}}$$

where t in (4.4.62d) is defined as $t = \frac{N-1}{2} - 1$, hence, $u_{3,N-1}$ has been replaced by $u_{3,2} = a_{2,2}$ and $l_{2,N-1}$ by $l_{2,2}$ by virtue of (4.4.58b).

Similarly, for $r=3$ (i.e. where (4.4.1) is a septadiagonal system) we can express $l_{3,1}, l_{3,2}$ and $l_{3,3}$ as periodic fractions, again N (the order of the coefficient matrix) is considered and hence three cases arise due to the relations (4.4.59a), (4.4.59b) and (4.4.59c). Thus, in general, for $r \geq 1$, from the equation (4.4.5) (precisely equation (4.4.5a) and (4.4.5c)) we express the elements $l_{r,1}, l_{r,2}, \dots, l_{r,r}$ as infinite periodic fractions as in the algorithm FICM2 (where the coefficient matrix is periodic). But, in contrast to the periodic case (FICM2), for the present (non-periodic) case, N must be taken into account.

Now by using the same notation and the abbreviated form for the continued fraction as in (4.4.36), we can express $l_{r,1}, l_{r,2}, \dots, l_{r,r}$ of (4.4.5a) as follows,

$$\left. \begin{aligned} a_{r,1}^{l_{r,1}} &= \frac{\alpha_{1,1}}{\beta_{1,1}} - \frac{\alpha_{1,2}}{\beta_{1,2}} - \frac{\alpha_{1,3}}{\beta_{1,3}} - \dots - \frac{\alpha_{1,t}}{\beta_{1,t}} - \frac{\alpha_{1,1}}{\beta_{1,1}} - \frac{\alpha_{1,2}}{\beta_{1,2}} - \dots - \frac{\alpha_{1,t}}{\beta_{1,t}} - \frac{\alpha_{1,1}}{\beta_{1,1}} - \dots \\ a_{r,2}^{l_{r,2}} &= \frac{\alpha_{2,1}}{\beta_{2,1}} - \frac{\alpha_{2,2}}{\beta_{2,2}} - \frac{\alpha_{2,3}}{\beta_{2,3}} - \dots - \frac{\alpha_{2,t}}{\beta_{2,t}} - \frac{\alpha_{2,1}}{\beta_{2,1}} - \frac{\alpha_{2,2}}{\beta_{2,2}} - \dots - \frac{\alpha_{2,t}}{\beta_{2,t}} - \frac{\alpha_{2,1}}{\beta_{2,1}} - \dots \\ &\vdots \\ a_{r,r}^{l_{r,r}} &= \frac{\alpha_{r,1}}{\beta_{r,1}} - \frac{\alpha_{r,2}}{\beta_{r,2}} - \frac{\alpha_{r,3}}{\beta_{r,3}} - \dots - \frac{\alpha_{r,t}}{\beta_{r,t}} - \frac{\alpha_{r,1}}{\beta_{r,1}} - \frac{\alpha_{r,2}}{\beta_{r,2}} - \dots - \frac{\alpha_{r,t}}{\beta_{r,t}} - \frac{\alpha_{r,1}}{\beta_{r,1}} - \dots \end{aligned} \right\} \quad (4.4.63)$$

where

$$\alpha_{k,i} = a_{-r,s} a_{r,s-r}$$

$$\text{and } \beta_{k,i} = a_{0,s} - \sum_{j=r-1}^1 \ell_{j,s} u_{j+1,s}$$

$$\left. \begin{array}{l} s=ir+k \leq N, \\ k=1,2,\dots,r, \\ \text{and } i=1,2,\dots,t, \end{array} \right\}$$

such that t is defined as follows,

$$t = \begin{cases} \lfloor \frac{N}{r} \rfloor, & \text{for calculating } \ell_{r,1}, \dots, \ell_{r,m} \\ \lfloor \frac{N}{r} \rfloor - 1, & \text{for calculating } \ell_{r,m+1}, \dots, \ell_{r,r} \end{cases}$$

and m is defined in (4.4.54).

However, the analysis of the convergence of the fractions (4.4.36a) applies to the fractions (4.4.36). Hence, if $\hat{\omega}_k$, $k=1,2,\dots,r$ is assumed to be the limit of the k^{th} fraction in (4.4.36), we have,

$$\ell_{r,k} = \hat{\omega}_k / a_{r,k}, \quad k=1,2,\dots,r. \quad (4.4.64)$$

We now proceed to determine the elements of the matrices U_1 in (4.4.52a) and L_1 in (4.4.53a). An iterative procedure similar to the one applied to the equations (4.3.4) (algorithm FICM2) is adopted to compute these elements from the equations (4.4.5). (In both cases, the continued fractions are used at each step in an iterative process).

The equations (4.4.5d) can be written in an iterative form so that the $u_{k,i}$, $k=r+1,r,\dots,2$, $i=1,2,\dots,N$ at the s^{th} step ($s \geq 1$) are given by,

$$u_{r+1,i} = a_{r,i}, \quad i=1,2,\dots,N-r, \quad (r)$$

$$\text{and } u_{r+1,N-j+1} = u_{r+1,h_j}, \quad j=1,2,\dots,r, \quad (r^*)$$

$$u_{r,i}^{(s)} + u_{r+1,i} \ell_{1,i+r-1}^{(s-1)} = a_{r-1,i}, \quad i=1,2,\dots,N-r+1, \quad (r-1)$$

$$\text{and } u_{r,N-j+1}^{(s)} = u_{r,h_j}^{(s)}, \quad j=1,2,\dots,r-1, \quad (r-1^*)$$

$$u_{r-1,i}^{(s)} + u_{r,i} \ell_{1,i+r-2}^{(s-1)} + u_{r+1,i} \ell_{2,i+r-2}^{(s-1)} = a_{r-2,i}, \quad i=1,2,\dots,N-r+2, \quad (r-2)$$

$$\text{and } u_{r-1,i}^{(s)} = u_{r-1,h_j}^{(s)}, \quad j=1,2,\dots,r-2, \quad (r-2^*)$$

$$u_{3,i}^{(s)} + u_{4,i} \ell_{1,i+2}^{(s-1)} + u_{5,i} \ell_{2,i+2}^{(s-1)} + \dots + u_{r+1,i} \ell_{r-2,i+2}^{(s-1)} = a_{2,i}, \quad i=1,2,\dots,N-2, \quad (2)$$

$$\text{and } u_{3,N-j+1}^{(s)} = u_{3,h_j}^{(s)}, \quad j=1,2, \quad (2^*)$$

$$u_{2,i}^{(s)} + u_{3,i} \ell_{1,i+1}^{(s-1)} + u_{4,i} \ell_{2,i+1}^{(s-1)} + \dots + u_{r+1,i} \ell_{r-1,i+1}^{(s-1)} = a_{1,i}, \quad i=1,2,\dots,N-1, \quad (1)$$

$$\text{and } u_{2,N}^{(s)} = u_{2,h_1}^{(s)}, \quad (1^*)$$

(4.4.65)

INTER-LIBRARY

LOAN REQUEST

Applicant:

JOHN H. LIGHT

22/9

University
Address

Geog.

Periodical Title or

Book Author & Title

or Report Details

PLANNING OUTLOOK

Author or Title of
Periodical Article~~WILLIAMS, R~~

Kalamazoo

BUSINESS SYSTEMS
732537-5x71

Year

1980

Vol.

23

Part/Edition

1

Pages

(Whole volume)
~~13-15~~

ISBN BNB No.

Publisher

Place of Publication

SMITH, P. A.

6-1-86.

ATKINSON / PASCAL PROGRAMS

BOWLES / BEGINNER'S GUIDE FOR THE UCSD
PASCAL SYSTEM

BARUARD / PASCAL PROGRAMMING

where the equations $(r), (r-1), \dots, (1)$ are those of (4.4.5d), whilst those denoted by $(r^*), (r-1^*), \dots, (2^*)$ and (1^*) are derived from the relation (4.4.56a); the superscript s is dropped from the relation (r) since no iteration is involved in this relation because it includes none of the other unknowns, $\ell_{k,i}$ or $u_{k,i}$. The values $\ell_{k,i}^{(s-1)}$, $k=1, 2, \dots, r-1$, $i=1, \dots, N-k$, are computed from the previous iteration step (except at $s=1$, $\ell_{k,i}^{(0)}$ are taken as initial guesses) and

$$\ell_{k,N-j+1}^{(s-1)} = \ell_{k,h_j}^{(s-1)}, \quad j=1, 2, \dots, k, \quad k=1, 2, \dots, r-1, \quad (4.4.66a)$$

are obtained from (4.4.56b).

The values $\ell_{r,1}, \ell_{r,2}, \dots, \ell_{r,r}$ can be obtained from the periodic continued fractions (4.4.63), hence we can rewrite (4.4.64) as

$$\ell_{r,k}^{(s)} = \hat{w}_k^{(s)} / a_{r,k}, \quad k=1, 2, \dots, r, \quad (4.4.66b)$$

where $\hat{w}_k^{(s)}$ is the limit of the k^{th} fraction in (4.4.63) at the s^{th} step of the iteration procedure;

Subsequently, the $\ell_{r,i}$, $i=r+1, \dots, N-r$, $i=1, \dots, N$ can be determined from the following recursive relations which are derived from (4.4.5a) and (4.4.5c), i.e.

$$\left. \begin{aligned} \ell_{r,i}^{(s)} &= a_{-r,i+r} / u_{1,i+r}^{(s)}, \quad r+1 \leq i \leq N-r \\ \text{and } \ell_{r,N-j+1}^{(s)} &= \ell_{r,h_j}^{(s)}, \quad j=1, 2, \dots, r \text{ (obtained from (4.4.56b))}, \\ u_{1,i}^{(s)} &= a_{0,i} - (u_{2,i}^{(s)} \ell_{1,i}^{(s-1)} + u_{3,i}^{(s)} \ell_{2,i}^{(s-1)} + \dots + u_{r,i}^{(s)} \ell_{r-1,i}^{(s-1)} + u_{r+1,i}^{(s)} \ell_{r,i}^{(s)}) \end{aligned} \right\} i=1, 2, \dots, N. \quad (4.4.66c)$$

Also, the equations (4.4.56) can be written in iterative form with additional terms $\epsilon_{k,i}$ for $k=1, \dots, r-1$, $i=1, 2, \dots, N-k$ (c.f. (4.3.31)),

$$\left. \begin{aligned} u_{1,i+r-1}^{(s)} \ell_{r-1,i}^{(s-1)} + u_{2,i+r-1}^{(s)} \ell_{r,i}^{(s-1)} + \epsilon_{r-1,i}^{(s)} &= a_{-r+1,i+r-1}, \quad i=1, 2, \dots, N-r+1, \\ u_{1,i+r-2}^{(s)} \ell_{r-2,i}^{(s-1)} + u_{2,i+r-2}^{(s)} \ell_{r-1,i}^{(s-1)} + u_{3,i+r-2}^{(s)} \ell_{r,i}^{(s-1)} + \epsilon_{r-2,i}^{(s)} &= a_{-r+2,i+r-2}, \\ &\vdots \\ u_{1,i+1}^{(s)} \ell_{1,i}^{(s-1)} + u_{2,i+1}^{(s)} \ell_{2,i}^{(s-1)} + \dots + u_{r,i+1}^{(s)} \ell_{r,i}^{(s-1)} + \epsilon_{1,i}^{(s)} &= a_{-1,i+1}, \quad i=1, 2, \dots, N-1 \end{aligned} \right\} i=1, 2, \dots, N-r+2, \quad (4.4.67)$$

where $\epsilon_{k,i}^{(s)}$, $k=r-1, \dots, 1$, $i=1, 2, \dots, N-k$ refers to the error term due to the 'inaccurate' value of the corresponding $\ell_{k,i}^{(s-1)}$. On the other hand, if the $\ell_{k,i}^{(s-1)}$ (and the u 's) are assumed to be 'accurate', then (4.4.67) may be written as,

$$\left. \begin{aligned} u_{1,i+r-1}^{(s)} \ell_{r-1,i}^{(s)} + u_{2,i+r-1}^{(s)} \ell_{r,i}^{(s)} &= a_{-r+1,i+r-1}, \quad i=1, 2, \dots, N-r+1 \\ u_{1,r+r-2}^{(s)} \ell_{r-2,i}^{(s)} + u_{2,i+r-2}^{(s)} \ell_{r-1,i}^{(s)} + u_{3,i+r-2}^{(s)} \ell_{r,i}^{(s)} &= a_{-r+2,i+r-2}, \\ &\quad i=1, 2, \dots, N-r+2 \\ \hline u_{1,i+1}^{(s)} \ell_{1,i}^{(s)} + u_{2,i+1}^{(s)} \ell_{2,i}^{(s)} + \dots + u_{r,i+1}^{(s)} \ell_{r,i}^{(s)} &= a_{-1,i+1}, \quad i=1, 2, \dots, N-1 \end{aligned} \right\} \quad (4.4.68)$$

Now the subtraction of the first equation, the second equation, up to the last equation of (4.4.67) from the corresponding equation of (4.4.68) with rearrangement yields the result (c.f. (4.3.33)),

$$\left. \begin{aligned} \ell_{r-1,i}^{(s)} &= \ell_{r-1,i}^{(s-1)} + \epsilon_{r-1,i}^{(s-1)} / u_{1,i+r-1}^{(s)}, \quad i=1, 2, \dots, N-r+1 \\ \ell_{r-2,i}^{(s)} &= \ell_{r-2,i}^{(s-1)} + \epsilon_{r-2,i}^{(s-1)} / u_{1,i+r-2}^{(s)}, \quad i=1, 2, \dots, N-r+2 \\ \hline \text{and } \ell_{1,i}^{(s)} &= \ell_{1,i}^{(s-1)} + \epsilon_{1,i}^{(s-1)} / u_{1,i+1}^{(s)}, \quad i=1, 2, \dots, N-1. \end{aligned} \right\} \quad (4.4.69)$$

The equations of (4.4.67) and (4.4.69) are associated in an alternate manner analogous to the relations between the equations of (4.3.31) and (4.3.33).

The summary of the above iterative procedure can now be outlined by the following steps.

Step 1 Initialize $\ell_{1,i}^{(0)}, \ell_{2,i}^{(0)}, \dots, \ell_{r-1,i}^{(0)}$, $i=1, 2, \dots, N$.

Step 2 (i) Obtain $\ell_{k,N-j+1}^{(s-1)}$ from (4.4.56b), i.e.

$$\ell_{k,N-j+1}^{(s-1)} = \ell_{k,h_j}^{(s-1)}, \quad j=1, 2, \dots, k, \quad k=1, 2, \dots, r-1,$$

(ii) Determine $u_{r,i}^{(s)}, u_{r-1,i}^{(s)}, \dots, u_{2,i}^{(s)}$ successively from the following relations,

$$u_{r+1,i}^{(s)} = a_{r,i}, \quad i=1,2,\dots,N-r$$

and $u_{r+1,N-j+1}^{(s)} = u_{r+1,h_j}^{(s)}, \quad j=1,2,\dots,r, \text{ (obtained from (4.4.56a))},$

$$\left. \begin{aligned} u_{k,i}^{(s)} &= a_{k-1,i} - \sum_{j=1}^{r-k+1} u_{k+j,i}^{(s)} \ell_{j,i+k-1}^{(s-1)}, \quad i=1,2,\dots,N-k+1 \\ u_{k,N-j+1}^{(s)} &= u_{k,h_j}^{(s)}, \quad j=1,2,\dots,k-1 \text{ (obtained from (4.4.56a))} \end{aligned} \right\} \quad k=r,r-1,\dots,2.$$

Step 3 (i) Determine $\ell_{r,1}^{(s)}, \ell_{r,2}^{(s)}, \dots, \ell_{r,r}^{(s)}$ by the continued fraction (4.4.63), and

(ii) Determine $\ell_{r,r+1}^{(s)}, \ell_{r,r+2}^{(s)}, \dots, \ell_{r,N-r}^{(s)}$ and $u_{1,1}^{(s)}, \dots, u_{1,N}^{(s)}$ from the recurrence relations,

$$\ell_{r,i}^{(s)} = a_{-r,i+r} / u_{1,i+r}^{(s)}, \quad r+1 \leq i \leq N-r$$

and

$$\begin{aligned} \ell_{r,N-j+1}^{(s)} &= \ell_{r,h_j}^{(s)}, \quad j=1,2,\dots,r \text{ (obtained from (4.4.56b))} \\ \ell_{1,i}^{(s)} &= a_{0,i} - \sum_{k=1}^{r-1} u_{k+1,i}^{(s)} \ell_{k,i}^{(s-1)} - u_{r+1,i}^{(s)} \ell_{r,i}^{(s)}. \end{aligned}$$

Step 4 Evaluate $\epsilon_{k,i}^{(s-1)}$ and $\ell_{k,i}^{(s)}$, $k=1,2,\dots,r-1$, $i=1,2,\dots,N-k$ as follows from (4.4.67) and (4.4.69) alternately, we have,

$$(a_1) \quad \epsilon_{r-1,i}^{(s-1)} = a_{-r+1,i+r-1} - u_{1,i+r-1}^{(s)} \ell_{r-1,i}^{(s-1)} - u_{2,i+r-3}^{(s)} \ell_{r,i}^{(s)}, \quad i=1,2,\dots,N-r+1$$

$$(b_1) \quad \ell_{r-1,i}^{(s)} = \ell_{r-1,i}^{(s-1)} + \epsilon_{r-1,i}^{(s-1)} / u_{1,i+r-1}^{(s)}, \quad i=1,2,\dots,N-r+1,$$

$$(a_2) \quad \epsilon_{r-2,i}^{(s-1)} = a_{-r+2,i+r-2} - u_{1,i+r-2}^{(s)} \ell_{r-2,i}^{(s-1)} - u_{2,i+r-2}^{(s)} \ell_{r-1,i}^{(s-1)} - u_{3,i+r-2}^{(s)} \ell_{r,i}^{(s)}, \quad i=1,2,\dots,N-r+2,$$

$$(b_2) \quad \ell_{r-2,i}^{(s)} = \ell_{r-2,i}^{(s-1)} + \epsilon_{r-2,i}^{(s-1)} / u_{1,i+r-2}^{(s)}, \quad i=1,2,\dots,N-r+2$$

\vdots

$$(a_{r-1}) \quad \epsilon_{1,i}^{(s-1)} = a_{-1,i+1} - u_{1,i+1}^{(s)} \ell_{1,i}^{(s-1)} - u_{2,i+1}^{(s)} \ell_{2,i}^{(s-1)} - \dots - u_{r,i+1}^{(s)} \ell_{r,i}^{(s-1)}, \quad i=1,2,\dots,N-1,$$

$$(b_{r-1}) \quad \ell_{1,i}^{(s)} = \ell_{1,i}^{(s-1)} + \epsilon_{1,i}^{(s-1)} / u_{1,i+1}^{(s)}, \quad i=1,2,\dots,N-1.$$

Step 5 We define ϵ such that

$$|\epsilon| = \max_k (\max_i |\epsilon_{k,i}|).$$

Thus, if $|\epsilon| \leq \text{TOL}$ (the desired accuracy), then the iterative

process is halted, otherwise we repeat the process from Step 2.

Finally, as indicated in algorithm FICM2, the above iterative process does converge without considering the use of the continued fractions, on the other hand, it was observed, even in this case, that the choice of elements of the matrices V_1 and K_1 in terms of the elements of the matrices U_1 and L_1 respectively, in the manner discussed earlier in this subsection was satisfactory. Thus, the iterative procedure outlined above may be written without the use of continued fractions as:

Step 1' Initialize $\ell_{1,i}^{(0)}, \ell_{2,i}^{(0)}, \dots, \ell_{r,i}^{(0)}$, $i=1,2,\dots,N$.

Step 2' (i) Obtain $\ell_{k,N-j+1}^{(s-1)}$ from (4.4.56b), i.e.,

$$\ell_{k,N-j+1}^{(s-1)} = \ell_{k,h_j}^{(s-1)}, \quad j=1,2,\dots,k, \quad k=1,2,\dots,r,$$

(ii) Determine $u_{r,i}^{(s)}, u_{r-1,i}^{(s)}, \dots, u_{2,i}^{(s)}, u_{1,i}^{(s)}$ successively from the following relations,

$$u_{r+1,i}^{(s)} = a_{r,i}, \quad i=1,2,\dots,N-r,$$

and

$$u_{r+1,N-j+1}^{(s)} = u_{r+1,h_j}^{(s)}, \quad j=1,2,\dots,r \text{ (obtained from (4.4.56a))},$$

$$\left. \begin{aligned} u_{k,i}^{(s)} &= a_{k-1,i} - \sum_{j=1}^{r-k+1} u_{k+j,i}^{(s)} \ell_{j,i+k-1}^{(s-1)}, \quad i=1,2,\dots,N-k+1 \\ \text{and} \\ u_{k,N-j+1}^{(s)} &= u_{k,h_j}^{(s)} \quad (\text{for } k>1 \text{ only}), \quad j=1,2,\dots,k-1, \\ &\quad k=r,r-1,\dots,1. \end{aligned} \right\}$$

Step 3' Evaluate $\epsilon_{k,i}^{(s-1)}$ and $\ell_{k,i}^{(s)}$, $k=1,2,\dots,r$, $i=1,2,\dots,N-k$ as follows,

$$(a_1) \quad \epsilon_{r,i}^{(s-1)} = a_{-r,i+r} - u_{1,i+r}^{(s)} \ell_{r,i}^{(s-1)}, \quad i=1,2,\dots,N-r$$

$$(b_1) \quad \ell_{r,i}^{(s)} = \ell_{r,i}^{(s-1)} + \epsilon_{r,i}^{(s-1)} / u_{1,i+r}^{(s)}, \quad i=1,2,\dots,N-r$$

$$(a_2) \quad \epsilon_{r-1,i}^{(s-1)} = a_{-r,i+r-1}^{-u_{1,i+r-1}^{(s)}} \ell_{r-1,i}^{(s-1)} - u_{2,i+r-2}^{(s)} \ell_{r,i}^{(s)}, \\ i=1,2,\dots,N-r+1$$

$$(b_2) \quad \ell_{r-1,i}^{(s)} = \ell_{r-1,i}^{(s-1)} + \epsilon_{r-1,i}^{(s-1)} / u_{1,i+r-1}^{(s)}, \quad i=1,2,\dots,N-r+1$$

$$\vdots$$

$$(a_r) \quad \epsilon_{1,i}^{(s-1)} = a_{-1,i+1}^{-u_{1,i+1}^{(s)}} \ell_{1,i}^{(s-1)} - u_{2,i+1}^{(s)} \ell_{3,i}^{(s)} - \dots - u_{r,i+1}^{(s)} \ell_{r,i}^{(s)},$$

$$(b_r) \quad \ell_{1,i}^{(s)} = \ell_{1,i}^{(s-1)} + \epsilon_{1,i}^{(s-1)} / u_{1,i+1}^{(s)},$$

Step 4' As in step 5 of the previous procedure.

The convergence proof is similar to that discussed in subsection 4.3.6.

4.5 ALGORITHM FICM5

The current algorithm deals with real linear systems, where the coefficient matrix is of special form, that is: constant, periodic and skew-symmetric. These type of matrices may arise in solving the transport equation by finite difference techniques (Evans, (1980)), also in the solution of partial differential equations with periodic equations, as in Korteweg de Vries equation (Buckley, 1977)). The general form of the real linear system considered in this algorithm is,

$$\underline{Ax} = \underline{z}, \quad (4.5.1)$$

where A is a constant periodic and skew-symmetric matrix of bandwidth $2r+1$ ($r \geq 1$), and of order N (and $N \geq 2r+1$) and has the following form,

Diagram (4.5.2a) illustrates a grid of points labeled $a_0, a_1, a_2, \dots, a_{r-1}, a_r, a_{-r}, a_{-r+1}, a_{-r+2}, \dots, a_{-1}, a_0$. The points are arranged in a grid with dashed lines connecting them. The grid is labeled $A=$ on the left and $(4.5.2a)$ on the right. The origin is marked with a 0.

with $a_k = -a_{-k}$, $k=1,2,\dots,r$ (4.5.2b)

Evans (1980) suggests that for a certain Toeplitz tridiagonal case of (4.5.1), i.e. for $r=1$, the matrix A in (4.5.2a) can be factorized as follows,

$$A_{(r=1)} = PQ, \quad (4.5.3a)$$

where

$$P = \begin{bmatrix} \gamma_0 & & & & -\gamma_1 \\ & -\gamma_1 & \gamma_0 & & 0 \\ & & & & \\ & & & & \\ 0 & & & & -\gamma_1 & \gamma_0 \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} \gamma_0 & \gamma_1 & & & 0 \\ & \gamma_0 & & & \\ & & & & \\ & & 0 & & \gamma_1 \\ \gamma_1 & & & & \gamma_0 \end{bmatrix} \quad (4.5.3b)$$

Then, by equating the elements of $A_{(r=1)}$ and the product PQ by virtue of (4.5.3a) leads to a quartic equation in terms of γ_0 (or γ_1). To proceed to the elimination process the suitable values of γ_0 and γ_1 (as proposed by Evans) are given by,

$$\left. \begin{aligned} \gamma_0 &= [0.5\{a_0 + (a_0^2 + 4a_1^2)^{\frac{1}{2}}\}]^{\frac{1}{2}} \\ \text{and} \quad \gamma_1 &= [0.5\{-a_0 + (a_0^2 + 4a_1^2)^{\frac{1}{2}}\}]^{\frac{1}{2}} \end{aligned} \right\} \quad (4.5.4)$$

where a_0 and a_1 are the elements of A in (4.5.2a).

However, the efforts to extend the factorization (4.5.3) to a quin-diagonal, septadiagonal or even more for the general case as in (4.5.2a) were not satisfactory, thus the alternative is briefly illustrated below which involves a modification of both the coefficient matrix and the right-hand side vector except the vector solution of (4.5.1).

The premultiplication of the matrix equation (4.5.1) by A^T yields the system

$$B\underline{x} = \underline{v}, \quad (4.5.5)$$

where

$$B = A^T A, \quad (4.5.6)$$

and

$$\underline{v} = A^T \underline{z}, \quad (4.5.7)$$

with A given in (4.5.2a).

From (4.5.6) it can be easily verified that B is *symmetric* and preserves the remaining properties of A , i.e. periodic and constant with wider bandwidth; of bandwidth $2\hat{r}+1$ (where $\hat{r}=2r$). This implies that the system (4.5.5) is exactly similar to (4.2.1) and hence algorithm FICM1

(section 4.2) is applicable to the system (4.5.5). Noting that the elements of B and \underline{y} are obtained from (4.5.6) and (4.5.7) respectively as follows.

Let the diagonal elements of B be b_0 with $b_1, b_2, \dots, b_{\hat{r}} (\equiv b_{2r})$ the off-diagonal elements. Then, by equating the corresponding elements on both sides of (4.5.6) we can obtain the elements of B , i.e. $b_{\hat{r}}, b_{\hat{r}-1}, \dots, b_1, b_0$. These elements may be expressed in terms of the elements of A in matrix form as follows:

$$\begin{bmatrix} b_{\hat{r}} \\ b_{\hat{r}-1} \\ \vdots \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} -a_r & & & & & \\ & -a_{r-1} & -a_r & & & \\ & & \ddots & & & \\ & & & 0 & & \\ & -a_1 & -a_{r-1} & -a_r & & \\ & a_0 & & & & \\ & a_1 & & & & \\ & a_2 & & & & \\ & & & & & \\ a_r & a_2 & a_1 & a_0 & -a_1 & -a_{r-1} & -a_r \end{bmatrix} \begin{bmatrix} a_r \\ a_{r-1} \\ \vdots \\ a_1 \\ a_0 \\ -a_1 \\ -a_2 \\ \vdots \\ -a_r \end{bmatrix} \quad (4.5.8)$$

where $\hat{r}=2r$, $r \geq 1$; noting that we have substituted for a_{-k} by $-a_k$, $k=1,2,\dots,r$ due to (4.5.2b). The latter also applies to the relation (4.5.7) from which we determine the components v_j , $j=1,2,\dots,N$, of the vector \underline{v} given by

$$v_j = a_0 z_j + \sum_{k=1}^r a_k (z_{j-k} - z_{j+k}), \quad j=1,2,\dots,N, \quad (4.5.9)$$

with

$$\left. \begin{aligned} z_{-k+1} &= z_{N-k+1} \\ z_{N+k} &= z_k \end{aligned} \right\} \quad k=1,2,\dots,r.$$

and

It may be worthwhile to consider an example to clarify the above strategy of solving the system (4.5.1). We choose the simplest case when $r=1$, that is the case when A is skew-symmetric, tridiagonal and its transpose A^T has the form,

$$A_{(r=1)}^T = \begin{bmatrix} a_0 & -a_1 & & & a_1 \\ a_1 & & 0 & & \\ & & & & \\ & 0 & & & -a_1 \\ -a_1 & & & a_1 & a_0 \end{bmatrix} \quad (4.5.10)$$

Whilst the product $A^T A$ matrix has the form,

$$A^T A = \begin{bmatrix} a_0^2 + 2a_1^2 & 0 & -a_1^2 & & -a_1^2 & 0 \\ 0 & & & & 0 & -a_1^2 \\ -a_1^2 & & & & & -a_1^2 \\ & & 0 & & & 0 \\ -a_1^2 & & & & -a_1^2 & 0 \\ 0 & -a_1^2 & & & -a_1^2 & a_0^2 + 2a_1^2 \end{bmatrix} \quad (4.5.11)$$

In fact, the constant symmetric periodic quindagonal matrix in (4.5.11) is equivalent to B by virtue of (4.5.6).

If we now recall the factorization procedure of algorithm FICM1 (section 4.2), then B can be factorized as,

$$B (= A^T A) = Q Q^T, \quad (4.5.12)$$

where,

$$Q = \begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & & & \\ & & & & 0 & \\ & & & & & \alpha_2 \\ & 0 & & & & \alpha_1 \\ \alpha_2 & & & & & \alpha_0 \\ \alpha_1 & \alpha_2 & & & & \end{bmatrix} \quad (4.5.13)$$

The three equations derived from equating the corresponding elements on both sides of (4.5.12) (c.f. the system (4.2.4)), can be modified to the form (4.2.23) which immediately yields the values of $\alpha_0, \alpha_1, \alpha_2$, i.e.,

$$\left. \begin{aligned} \alpha_1 &= 0, \\ \alpha_0 &= \max\left(\frac{1}{2}[a_0^2 + (a_0^2 + 4a_1^2)^{\frac{1}{2}}]\right) \\ \alpha_2 &= \min\left(\frac{1}{2}[a_0^2 + (a_0^2 + 4a_1^2)^{\frac{1}{2}}]\right) \end{aligned} \right\} \quad (4.5.14)$$

and

By recalling the system (4.5.5) we substitute for B as in (4.5.12) then the two alternative systems for (4.5.5) may be written as

$$Q\underline{y} = \underline{v} \quad (4.5.15a)$$

$$\text{and} \quad Q^T \underline{x} = \underline{y}, \quad (4.5.15b)$$

where Q is defined in (4.5.13) and vector \underline{y} is an auxiliary vector of N components.

Though the systems in (4.5.15) are similar to (4.2.2) hence their solution proceeds as in subsection 4.2.3. Here we consider just one system say (4.5.15a) since the matrix Q is of special structure, i.e. its element α_1 is zero (as given in (4.5.14)).

The system (4.5.15a) can be rewritten in the form,

$$\begin{bmatrix} \alpha_0 & 0 & \alpha_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 0 & & & \\ & & & \ddots & & \\ & & & & 0 & \\ & & & & & \ddots \\ \alpha_2 & & & & & & \alpha_2 \\ & 0 & \alpha_2 & & & & 0 \\ & & & & & & \alpha_0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \quad (4.5.16)$$

By applying the elimination procedure discussed in subsection 4.2.3 to the system (4.5.17), we arrive at the result where (4.5.16) takes the form,

(i) for *N even*,

$$\begin{bmatrix} f_1 + \alpha_0 & & & & \\ & 0 & f_1 + \alpha_0 & & \\ f_2 & & 0 & \alpha_0 & \\ & 0 & f_2 & & 0 \\ \vdots & & \vdots & & \vdots \\ f_n & & 0 & & \\ 0 & f_n & & & \alpha_0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} \tilde{v}_1 \\ \tilde{v}_2 \\ \vdots \\ \tilde{v}_N \end{bmatrix} \quad (4.5.17a)$$

where

$$n = N/2,$$

$$f_j = (-\alpha_2/\alpha_0)^{n-j} \alpha_2, \quad j=n, n-1, \dots, 1,$$

and

$$\tilde{v}_j = v_j + (-\alpha_2/\alpha_0) \tilde{v}_{j+2}, \quad j=N-2, N-3, \dots, 1,$$

with

$$\tilde{v}_N = v_N, \quad \tilde{v}_{N-1} = v_{N-1},$$

(ii) and for *N odd*,

$$\begin{bmatrix} \alpha_0 & f_1 & & & \\ f_2 & \alpha_0 & & & \\ 0 & f_2 & \alpha_0 & & \\ f_3 & 0 & & & 0 \\ 0 & f_3 & & & \\ \vdots & \vdots & & & \vdots \\ f_n & 0 & & & \\ 0 & f_n & & & \alpha_0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} \tilde{v}_1 \\ \tilde{v}_2 \\ \vdots \\ \tilde{v}_N \end{bmatrix} \quad (4.5.17b)$$

where

$$\tilde{v}_j, \quad j=1, 2, \dots, N, \text{ and } f_k, \quad k=1, 2, \dots, n \text{ as defined in (4.5.17a)}$$

with

$$n = (N+1)/2.$$

Hence, the y_j , $j=1, 2, \dots, N$ for *N even* can be determined from (4.5.17a)

as follows,

$$\begin{aligned}
 y_1 &= \tilde{v}_1 / (f_1 + \alpha_0) , \quad y_2 = \tilde{v}_2 / (f_1 + \alpha_0) , \\
 y_{2j-1} &= (\tilde{v}_{2j-1} - f_j y_1) / \alpha_0 , \\
 \text{and } y_{2j} &= (\tilde{v}_{2j} - f_j y_2) / \alpha_0 , \quad \left. \vphantom{\begin{aligned} y_{2j-1} &= (\tilde{v}_{2j-1} - f_j y_1) / \alpha_0 , \\ y_{2j} &= (\tilde{v}_{2j} - f_j y_2) / \alpha_0 , \end{aligned}} \right\} j=2,3,\dots,n \ (\equiv N/2),
 \end{aligned}$$

whilst for N odd from (4.5.17b) we have,

$$\begin{aligned}
 y_1 &= (\tilde{v}_1 - \frac{f_1}{\alpha_0} \tilde{v}_2) / (\alpha_0 - \frac{f_1}{\alpha_0} f_2) , \\
 y_2 &= (\tilde{v}_2 - f_2 y_1) / \alpha_0 , \\
 y_{2j-1} &= (\tilde{v}_{2j-1} - f_j y_2) / \alpha_0 , \\
 \text{and } y_{2j} &= (\tilde{v}_{2j} - f_{j+1} y_1) / \alpha_0 , \quad j \neq n \quad \left. \vphantom{\begin{aligned} y_{2j-1} &= (\tilde{v}_{2j-1} - f_j y_2) / \alpha_0 , \\ y_{2j} &= (\tilde{v}_{2j} - f_{j+1} y_1) / \alpha_0 , \end{aligned}} \right\} j=2,3,\dots,n \ (\equiv (N+1)/2)
 \end{aligned}$$

Finally, the number of operations involved in solving the skew-symmetric system (4.5.1) via the modified form (4.5.5) is of $O(5\hat{r}N + (r+1)N)$, where the first term is given in subsection 4.2.3 and the second term due to the relation (4.5.9), (noting that the operations involved in the multiplication of $A^T A$ in (4.5.6) are ignored since the elements of B are obtained from (4.5.8) which require less than $(2r+1)(r+1)$ operations taking into account that some cancellations may occur). Since $\hat{r}=2r$, the order may be written in the form $O((11r+1)N)$. For a specific case this order (of the general form) may be reduced considerably, as for example when $r=1$, the solution of (4.5.15) is of $O(6\frac{1}{2}N)$ which is slightly higher than the scheme (4.5.3) of Evans (1980), but the latter requires 4 square roots as given in (4.6.5) while FICM5 for this particular case requires only 1 square root as given in (4.5.14). (Another important advantage of our scheme over Evans' scheme is that the matrix (4.5.11) is *strictly* diagonally dominant when $a_0=0$ which guarantees the stability of the elimination process for the system (4.5.15)).

(4.6.11) is suggested by Evans and Hadjidimos (1979) and they propose that β_1, β_2 and γ_1, γ_2 in (4.6.12a) and (4.6.12c) respectively may be chosen as follows:

$$\beta_1 = \beta_2 = [a_0^2 + a_1^2 - \alpha_1^2 - \alpha_2^2]^{\frac{1}{2}}$$

and $\gamma_1 = \gamma_2 = [a_0^2 + a_1^2 - \alpha_0^2 - \alpha_1^2]^{\frac{1}{2}} ;$

or by substituting for $\alpha_0, \alpha_1, \alpha_2$ in (4.6.13), we have,

$$\beta_1 = \beta_2 = \left[\frac{1}{2}a_0^2 + \frac{1}{2}a_1^2 (a_0^2 + 4a_1^2)^{\frac{1}{2}} \right]^{\frac{1}{2}} \quad \left. \vphantom{\beta_1 = \beta_2} \right\} \quad (4.6.14)$$

and $\gamma_1 = \gamma_2 = \left[\frac{1}{2}a_0^2 - \frac{1}{2}a_1^2 (a_0^2 + 4a_1^2)^{\frac{1}{2}} \right]^{\frac{1}{2}} .$

where a_0 and a_1 in both (4.6.13) and (4.6.14) are the elements of matrix B in (4.6.9).

Furthermore, the system (4.6.3) for the particular case where B is a quindagonal matrix as given in (4.6.9), can be replaced by two coupled underdetermined and overdetermined by 2 and have the form respectively (by considering the factorization (4.6.11)),

$$\begin{bmatrix} \beta_1 & 0 & \alpha_2 & & & \\ & \alpha_0 & 0 & \alpha_2 & & \\ & & & & 0 & \\ & & & & & \alpha_2 \\ & & & & & & \alpha_0 \\ & 0 & & & & & & \alpha_0 \\ & & & & & & & 0 & \gamma_1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ y_{N+1} \\ y_{N+2} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \quad (4.6.15a)$$

and

$$\begin{bmatrix} \beta_2 & & & & & \\ & 0 & \alpha_0 & & & \\ & \alpha_2 & 0 & & & \\ & & \alpha_2 & & & \\ & & & 0 & & \\ & & & & \alpha_2 & 0 \\ & & & & & \alpha_0 \\ & & & & & & 0 \\ & & & & & & & \alpha_2 & 0 \\ & & & & & & & & \gamma_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ y_{N+1} \\ y_{N+2} \end{bmatrix} \quad (4.6.15b)$$

Generally speaking, the solution of the coupled system (4.6.15) can be obtained following the procedure of subsection 4.4.2 (also see the relevant procedure of subsection 4.5.4), but because of the special structure of the coefficient matrices in (4.6.15a) and (4.6.15b) it may be of interest to describe the elimination procedure briefly and for one case, i.e. N being even.

$$\left. \begin{aligned} \text{Let} \quad n &= \frac{N}{2}, \\ f_i &= (-\alpha_2/\alpha_0)^{n-i} \equiv m^{n-i}, \quad i=1,2,\dots,n \end{aligned} \right\} \quad (4.6.18)$$

(N.B. for N odd $n=(N-1)/2$, and $f_i = m^{n-i+1}$, $i=1,2,\dots,n+1$).

Then the system (4.6.17a) can be modified to the form (c.f. (4.4.31)),

$$\begin{bmatrix} \beta_1 & & & & & \\ & 0 & \alpha_0 & & & \\ & & & 0 & & \\ & & & & \alpha_0 & \\ & & & & & 0 \\ & & & & & & \alpha_2 & 0 \\ & & & & & & & f_n \alpha_2 \\ & & & & & & & 0 \end{bmatrix} \begin{bmatrix} f_1 \alpha_2 \\ 0 \\ \vdots \\ 0 \\ f_n \alpha_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ f_1 \gamma_1 \\ \vdots \\ f_{n-1} \gamma_1 \\ 0 \\ f_n \gamma_1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ y_{N+1} \\ y_{N+2} \end{bmatrix} = \begin{bmatrix} \tilde{v}_1 \\ \tilde{v}_2 \\ \vdots \\ \tilde{v}_N \end{bmatrix} \quad (4.6.19)^{(*)}$$

where f_i , $i=1,2,\dots,n$ are given in (4.6.18) and \tilde{v}_j , $j=1,2,\dots,N$ are

(*) for N odd, the $(N+1)^{th}$ and $(N+2)^{th}$ column should have their first elements $0, f_1 \gamma_1$ respectively.

$$\left. \begin{aligned} \tilde{y}_{N+1} &= y_{N+1} + m\tilde{y}_{N-1} \equiv y_{N+1} + m y_{N-1} + m^2 y_{N-3} + \dots + m^{n-1} y_3 + m^{n-1} \theta y_1, \\ \text{(or } y_{2j-1} &= y_{2j-1} + m y_{2j-3} + \dots + m^{j-2} y_3 + m^{j-2} y_1, j=2,3,\dots,n+1) \end{aligned} \right\} \quad (4.6.22a)$$

$$\left. \begin{aligned} \tilde{y}_4 &= y_4 + m\tilde{y}_2 \equiv y_4 + m y_2, \\ \tilde{y}_6 &= y_6 + m\tilde{y}_4 \equiv y_6 + m y_4 + m^2 y_2, \\ \hline \tilde{y}_N &= y_N + m\tilde{y}_{N-2} \equiv y_N + m y_{N-2} + \dots + m^{n-1} y_2, \\ \text{(or } \tilde{y}_{2j} &= y_{2j} + m y_{2j-2} + \dots + m^{j-2} y_4 + m^{j-1} y_2, j=2,3,\dots,n), \\ \tilde{y}_{N+2} &= y_{N+2} + \epsilon \tilde{y}_N = y_{N+2} + \epsilon y_N + \epsilon m y_{N-2} + \dots + \epsilon m^{n-1} y_2, \\ &\quad (\epsilon = -\gamma_2/\alpha_0) \end{aligned} \right\} \quad (4.6.22b)$$

From (4.6.21), it is clear that both \tilde{y}_{N+1} and \tilde{y}_{N+2} are equal to zero, thus by substituting these values into the last equations of (4.6.22a) and (4.6.22b) respectively, we obtain after a slight rearrangement,

$$\left. \begin{aligned} -y_{N+1} &= m y_{N-1} + m^2 y_{N-3} + \dots + m^{n-1} y_3 + m^{n-1} \theta y_1, \\ -y_{N+2} &= \epsilon y_N + \epsilon m y_{N-2} + \dots + \epsilon m^{n-2} y_N + \epsilon m^{n-1} y_2, \end{aligned} \right\} \quad (4.6.23)$$

Returning now to (4.6.20a) we multiply its first equation by m^{n-1} , the second by m^{n-1} , the third by m^{n-2} , and so on up to the last one by m ; whilst for (4.6.20b) we multiply the first equation by ϵm^{n-1} , the second by ϵm^{n-2} , and so on up to the penultimate one by ϵm and the last by ϵ , then we substitute the two results in the first and the second equations of (4.6.23) to obtain the result,

$$\begin{aligned} -y_{N+1} &= [m^{n-1} \theta \tilde{v}_1 / \beta_1 + \frac{1}{\alpha_0} \sum_{j=1}^{n-1} m^{n-j} \tilde{v}_{2j+1}] - \left(\frac{\alpha_2}{\alpha_0} \sum_{j=1}^n m^{n-j+1} f_j + m^{n-1} \theta / \beta_1 \right) y_{N+1}, \\ -y_{N+2} &= \frac{\epsilon}{\alpha_0} \sum_{j=1}^n m^{n-j} \tilde{v}_{2j} - \left(\frac{\epsilon \gamma_1}{\alpha_0} \sum_{j=1}^n m^{n-j} f_j \right) y_{N+2}, \end{aligned}$$

$$\left. \begin{aligned} \text{or } y_{N+1} &= k_1 / (\ell_1 - 1) \quad (\text{provided } \ell_1 \neq 1), \\ \text{and } y_{N+2} &= k_2 / (\ell_2 - 1) \quad (\text{provided } \ell_2 \neq 1) \end{aligned} \right\} \quad (4.6.24)$$

where $k_1 = m^{n-1} \theta \tilde{v}_1 / \beta_1 + \frac{1}{\alpha_0} \sum_{j=1}^{n-1} m^{n-j} \tilde{v}_{2j+1}$,

$$\ell_1 = \frac{\alpha_2}{\alpha_0} \sum_{j=1}^n m^{n-j+1} f_j + m^{n-1} \theta / \beta_1 \equiv -m \sum_{j=1}^n m^{2(n-j)+1} + m^{n-1} \theta / \beta_1 ,$$

$$k_2 = \frac{\varepsilon}{\alpha_0} \sum_{j=1}^n m^{n-j} \tilde{v}_{2j} ,$$

and $\ell_2 = \frac{\varepsilon v_1}{\alpha_0} \sum_{j=1}^n m^{n-j} f_j \equiv \frac{\varepsilon \gamma_1}{\alpha_0} \sum_{j=1}^n m^{2(n-j)} ,$

where we have substituted for f_j in terms of m as defined in (4.6.18).

Therefore, y_{N+1} and y_{N+2} are obtained from (4.6.24) and then returning to (4.6.20) to determine y_j , $j=1,2,\dots,N$ followed by (4.6.22) to determine \tilde{y}_j , $j=1,2,\dots,N$, and finally the solution x_j , $j=1,2,\dots,n$ obtained immediately from (4.6.21), i.e.,

$$x_1 = \tilde{y}_1 / \beta_1 ,$$

$$x_j = \tilde{y}_j / \alpha_0 , j=2,3,\dots,N.$$

The number of operations of the above procedure is of $O(8N)$.

CHAPTER 5

NEW ALGORITHMIC METHODS FOR THE SOLUTION

OF BLOCK MATRIX EQUATIONS

5.1 ALGORITHM FICM3

This algorithm is proposed as a numerical solver for periodic block-tridiagonal linear systems which are derived from the finite-difference approximations to certain elliptic partial differential equations subject to periodic boundary conditions (see section 3.5, Chapter 3). In particular the type of real linear system considered in this algorithm is of the form,

$$\begin{bmatrix} B & C & & & \\ C & B & C & & \\ & C & B & C & \\ & & C & B & C \\ C & & & C & B \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_N \end{bmatrix} = \begin{bmatrix} \underline{z}_1 \\ \underline{z}_2 \\ \vdots \\ \underline{z}_N \end{bmatrix} \quad (5.1.1a)$$

or more compactly as,

$$A \underline{x} = \underline{z}, \quad (5.1.1b)$$

where each block B, C are $m \times m$ (real submatrices and each subvector \underline{x}_i and \underline{z}_i partitioned corresponding to the block subvectors are of length m , i.e. $\underline{x}_i \equiv [x_{i,1}, x_{i,2}, \dots, x_{i,m}]^T$, $\underline{z}_i \equiv [z_{i,1}, z_{i,2}, \dots, z_{i,m}]^T$, $i=1,2,\dots,N$, and A is a constant and symmetric circulant block-tridiagonal matrix of order mN .

We shall consider solving the system (5.1.1) by two coupled block linear systems after the factorization of the coefficient matrix A into two circulant block matrices which are the transpose of each other. For this, we assume that the submatrix B is non-singular, and A is *block* strictly diagonally dominant with respect to the matrix norm $||.||$, i.e., (see Varah (1972))

$$2||B^{-1}|| ||C|| < 1, \text{ provided } B \text{ is non-singular.} \quad (5.1.2)$$

The Block Factorization Procedure

The factorization strategy which is applied to the coefficient matrix in (5.1.1) is similar to the point case when $r=1$ in the algorithm FICM1,

(Subsection 4.2.1). Thus, the factorization of the matrix in (5.1.1) takes the form,

$$\begin{bmatrix} B & C & & 0 \\ & C & & \\ & & 0 & \\ C & & & C \\ & & 0 & \\ & & & C \\ C & & & B \end{bmatrix} = \begin{bmatrix} Q_0 & Q_1 & & 0 \\ & Q_1 & & \\ & & 0 & \\ Q_1 & & & Q_0 \end{bmatrix} \begin{bmatrix} Q_0 & & & Q_1 \\ Q_1 & & 0 & \\ & & Q_1 & \\ & 0 & & Q_0 \end{bmatrix} \quad (5.1.3)$$

$$\equiv \begin{bmatrix} Q_0^2 + Q_1^2 & Q_1 Q_0 & & Q_0 Q_1 \\ Q_0 Q_1 & & & \\ & & 0 & \\ Q_1 Q_0 & & & Q_0^2 + Q_1^2 \end{bmatrix}$$

where the blocks Q_0 and Q_1 are $m \times m$ submatrices and the three matrices on the right-hand side of (5.1.3) are of the same order as A .

We now equate the corresponding elements of both sides in (5.1.3) to obtain the following two matrix equations, i.e.,

$$\left. \begin{aligned} Q_0^2 + Q_1^2 &= B \\ Q_0 Q_1 &= C \quad (\text{or } Q_1 Q_0 = C) \end{aligned} \right\} \quad (5.1.4)$$

We note that the second equation and the bracketed one imply that Q_0 and Q_1 are commutative matrices. This latter property may be exploited so that the following expression holds true,

$$(Q_0 + Q_1)^2 = Q_0^2 + 2Q_0 Q_1 + Q_1^2 \quad (5.1.5)$$

If we now multiply the second equation in (5.1.4) by 2 and add or subtract to the first equation, then by virtue of (5.1.5) we obtain the relations,

$$\left. \begin{aligned} (Q_0 + Q_1)^2 &= (B + 2C) \equiv \tilde{B} \\ (Q_0 - Q_1)^2 &= (B - 2C) \equiv \tilde{C} \end{aligned} \right\} \quad (5.1.6)$$

and

Under the validity of the condition (5.1.2), we can define $\tilde{B}^{\frac{1}{2}}$ and $\tilde{C}^{\frac{1}{2}}$ as the square roots of matrices \tilde{B} and \tilde{C} respectively. Hence, from (5.1.6) we define the sum and difference of Q_0 and Q_1 as follows:

$$\left. \begin{aligned} Q_0 + Q_1 &= \tilde{B}^{\frac{1}{2}} \equiv (B+2C)^{\frac{1}{2}} \\ Q_0 - Q_1 &= \tilde{C}^{\frac{1}{2}} \equiv (B-2C)^{\frac{1}{2}} \end{aligned} \right\} . \quad (5.1.7)$$

The addition and subtraction of the two equations in (5.1.7) enables us to express Q_0 and Q_1 in the form,

$$\left. \begin{aligned} Q_0 &= 0.5[\tilde{B}^{\frac{1}{2}} + \tilde{C}^{\frac{1}{2}}] \equiv 0.5[(B+2C)^{\frac{1}{2}} + (B-2C)^{\frac{1}{2}}] \\ \text{and } Q_1 &= 0.5[\tilde{B}^{\frac{1}{2}} - \tilde{C}^{\frac{1}{2}}] \equiv 0.5[(B+2C)^{\frac{1}{2}} - (B-2C)^{\frac{1}{2}}] \end{aligned} \right\} \quad (5.1.8)$$

The computation of $\tilde{B}^{\frac{1}{2}}$ and $\tilde{C}^{\frac{1}{2}}$ is recommended to be accomplished in an efficient manner, for example by adopting the iterative procedure described in Section 2.5, provided that \tilde{B} and \tilde{C} satisfy the required property of this procedure, i.e. they must be positive definite. For other references which deal with the square root of a matrix see Spath (1967), Scofield (1973), etc. It follows immediately from (5.1.8) that since $(B+2C)$ and $(B-2C)$ and their square roots are positive definite then Q_0 is positive definite.

However, having determined the matrices Q_0 and Q_1 we now proceed to solve the system (5.1.1).

The Block Elimination Procedure

When the coefficient matrix in (5.1.1) is replaced by the two factors given in (5.1.3), then the system can be split into block linear systems after the insertion of an auxiliary vector \underline{y} of length mN and partitioned into N sub-vectors of length m each, i.e., $\underline{y} = [\underline{y}_1, \underline{y}_2, \dots, \underline{y}_N]^T$ (thus from (5.1.1) and (5.1.3) we write,

$$\begin{bmatrix} Q_0 & Q_1 & & 0 \\ & \ddots & \ddots & \\ & & Q_1 & Q_0 \\ 0 & & & \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} \quad (5.1.9a)$$

and

$$\begin{bmatrix} Q_0 & Q_1 & & 0 \\ & \ddots & \ddots & \\ & & Q_1 & Q_0 \\ 0 & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad (5.1.9b)$$

Prior to presenting the elimination process of the system (5.1.9) it is essential to point out that the stability of the system is guaranteed if the norm-relation of Q_0 and Q_1 is satisfied, i.e.,

$$||Q_1|| < ||Q_0|| \quad (5.1.10)$$

(the equality relation is excluded since it does not occur unless matrix $C=0$). Since Q_0 is positive definite which implies that its inverse Q_0^{-1} does exist, the normalization of both systems in (5.1.9) is possible and may be constructed as follows,

and

$$\begin{bmatrix} I & \tilde{Q} & & 0 \\ & I & \tilde{Q} & \\ & & \ddots & \ddots \\ \tilde{Q} & & & I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_N \end{bmatrix} \quad (5.1.10a)$$

$$\begin{bmatrix} I & \tilde{Q} & & 0 \\ & I & \tilde{Q} & \\ & & \ddots & \ddots \\ \tilde{Q} & & & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_N \end{bmatrix} \quad (5.1.10b)$$

for (5.1.9a) and (5.1.9b) respectively, (I is unit matrix of order m),

where

$$\left. \begin{aligned} \tilde{Q} &= Q_0^{-1} Q_1 \\ Q_0 \tilde{Q} &= Q_1 \end{aligned} \right\} \quad (5.1.11)$$

or

$$\left. \begin{aligned} \tilde{z}_j &= Q_0^{-1} z_j \\ Q_0 \tilde{z}_j &= z_j, \quad j=1,2,\dots,N \end{aligned} \right\} \quad (5.1.12)$$

or

$$\left. \begin{aligned} \tilde{y}_j &= Q_0^{-1} y_j \\ Q_0 \tilde{y}_j &= y_j, \quad j=1,2,\dots,N \end{aligned} \right\} \quad (5.1.13)$$

As can be noticed, the determination of \tilde{Q} , \tilde{z}_j and \tilde{y}_j (obviously after the determination of y_j from (5.1.10a)) follows from the last three equations respectively.

We now define the submatrices, F_1, F_2, \dots, F_N of order m as follows,

$$\left. \begin{aligned} F_j &= (-1)^{j+1} \tilde{Q}^{N-j+1}, \quad (\text{for } N \text{ odd}) \\ F_j &= (-1)^j \tilde{Q}^{N-j+1}, \quad (\text{for } N \text{ even}) \end{aligned} \right\} \quad j=N, N-1, \dots, 1. \quad (5.1.14)$$

Then, the elimination process can be applied to the block-systems (5.1.10a) and (5.1.10b) in an analogous way to the point-case discussed previously, (Chapter 4), obviously the process commences from the N^{th} equation backwards for the former system, and from the first equation forwards for the latter. After the elimination procedure has been completed for both systems in (5.1.10), taking into consideration the assumption (5.1.14), the systems (5.1.9a) and (5.1.9b) take the following forms, respectively,

$$\begin{bmatrix} F_1 + I & & & \\ & I & & \\ & & I & 0 \\ & & & \ddots & \ddots \\ & & 0 & & I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_N \end{bmatrix} \quad (5.1.15a)$$

and

$$\begin{bmatrix} I & & & & & & \\ & \ddots & & & & & \\ & & 0 & & & & \\ & & & I & & & \\ & 0 & & & I & & \\ & & & & & I & \\ & & & & & & I+F_1 \end{bmatrix} \begin{bmatrix} F_N \\ \vdots \\ F_3 \\ F_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \quad (5.1.15b)$$

where F_j , $j=1,2,\dots,N$ are defined in (5.1.14), the vectors \tilde{z}_j and \tilde{y}_j , $j=1,2,\dots,N$ are defined as follows,

$$\left. \begin{aligned} \tilde{z}_N &= \tilde{z}_N \\ \tilde{z}_j &= \tilde{z}_j - \tilde{Q} \tilde{z}_{j+1}, \quad j=N-1,\dots,1 \end{aligned} \right\} \quad (5.1.16a)$$

and

$$\left. \begin{aligned} \tilde{y}_1 &= \tilde{y}_1 \\ \tilde{y}_j &= \tilde{y}_j - \tilde{Q} \tilde{y}_{j-1}, \quad j=1,2,\dots,N \end{aligned} \right\}, \quad (5.1.16b)$$

where \tilde{z}_j, \tilde{y}_j are as given in (5.1.10a) and (5.1.10b) (or (5.1.12) and (5.1.13)) respectively, and Q as in (5.1.11).

The solution vector can now be obtained from the forward and backward substitution schemes for (5.1.15a) and (5.1.15b) respectively, i.e.,

$$\left. \begin{aligned} y_1 &= (F_1 + I)^{-1} \tilde{z}_1, \text{ or } (F_1 + I)y_1 = \tilde{z}_1 \\ y_j &= \tilde{z}_j - F_j y_1 \end{aligned} \right\}_{j=2,3,\dots,N}, \quad (5.1.17a)$$

$$\left. \begin{aligned} x_N &= (F_1 + I)^{-1} \tilde{y}_N, \text{ or } (F_1 + I)x_N = \tilde{y}_N \\ x_j &= \tilde{y}_j - F_j x_N \end{aligned} \right\}_{j=N-1,N-2,\dots,1}, \quad (5.1.17b)$$

provided that $(F_1 + I)^{-1}$ exists.

The solution procedure can be summarized in the following steps:

Step 1 Compute the submatrices Q_0 and Q_1 from (5.1.8) which involves the computation of the square root of a matrix by Newton's method.

Step 2 Obtain the submatrix \tilde{Q} from

$$Q_0 \tilde{Q}_1 = Q_1.$$

Step 3 Obtain F_j from

$$F_j = s Q^{N-j+1}, \quad s = \begin{cases} (-1)^{j+1}, & \text{for } N \text{ odd} \\ (-1)^j, & \text{for } N \text{ even.} \end{cases}$$

Step 4 Compute $\tilde{z}_j, \tilde{\tilde{z}}_j$ and y_j , $j=1,2,\dots,N$, as follows,

$$Q_0 \tilde{z}_j = z_j, \\ \tilde{\tilde{z}}_N = \tilde{z}_N, \quad \tilde{\tilde{z}}_j = \tilde{z}_j - \tilde{Q} \tilde{\tilde{z}}_{j+1}, \quad j=N-1,\dots,1$$

and

$$(F_1 + I) y_1 = \tilde{\tilde{z}}_1, \\ y_j = \tilde{\tilde{z}}_j - F_j y_1,$$

Step 5 Compute $\tilde{y}_j, \tilde{\tilde{y}}_j$ and x_j , $j=1,2,\dots,N$, as follows

$$Q_0 \tilde{y}_j = y_j, \\ \tilde{\tilde{y}}_1 = \tilde{y}_1, \quad \tilde{\tilde{y}}_j = \tilde{y}_j - \tilde{Q} \tilde{\tilde{y}}_{j-1}, \quad j=2,3,\dots,N,$$

and

$$(F_1 + I) x_N = \tilde{\tilde{y}}_N, \\ x_j = \tilde{\tilde{y}}_j - F_j x_N, \quad j=N-1, N-2, \dots, 1.$$

In general, the number of operations involved in the above procedure (excluding step 1) is of order $O(N(\frac{5}{3}m^3 + 4m^2))$. This may be reduced if we consider systems whose coefficient matrix Q_0 in steps 2 and 5 are such that Q_0 (assumed non-singular) can be decomposed into LU (see Chapter 2). Consequently the forward and backward substitution process (of $O(m^2)$) are required for these systems and hence the number of operations reduces to $O(Nm^3)$.

It is possible to reduce this order further if some efficient techniques are used for the matrix-vector multiplications, such as the Fast Fourier Transform (see Cooley and Tukey (1965), Brigham (1974), McConaghten and Hoare (1977)).

If the submatrix B in equation (5.1.1a) is periodic tridiagonal, and C is a diagonal matrix (normally, I or $-I$) then the above procedure may be reduced to the form considered by Okolie (1978) (which in fact is an extension of the tridiagonal point-case suggested by Evans (1973)).

For this special case, the matrix A can be factorized as follows:-
(c.f. Algorithm FICM2 with $r=1$),

$$\begin{bmatrix} B & C & & C \\ C & & 0 & \\ & & & C \\ C & 0 & C & B \end{bmatrix} = \begin{bmatrix} I & & & L \\ L & & 0 & \\ & 0 & & L \\ & & & I \end{bmatrix} \begin{bmatrix} U & C & & 0 \\ & & & C \\ & 0 & & \\ C & & & U \end{bmatrix} \quad (5.1.18)$$

where L, U and I (unitary) are $m \times m$ matrices.

If we set $C=I$, then (4.5.18) yields the result,

$$\left. \begin{array}{l} LU = I \\ \text{and } L+U = B \end{array} \right\} . \quad (5.1.19)$$

Okolie (1978) defines L according to (5.1.19) as follows,

$$L = 0.5(B - (B^2 - 4I)^{\frac{1}{2}}), \text{ provided } ||B|| > 2. \quad (5.1.20a)$$

Moreover, if L is assumed to be non-singular, then from the first equation of (5.1.19) U can be taken as,

$$U = L^{-1} . \quad (5.1.20b)$$

Subsequently, the systems (5.1.1) can be split into two coupled systems by virtue of (5.1.18) and (5.1.20), i.e.,

$$\begin{bmatrix} I & & & L \\ L & I & & \\ & & 0 & \\ & & & L \\ & 0 & & I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} \quad (5.1.21a)$$

and

and

$$\begin{bmatrix} L^{-1} & I & & & \\ & L^{-1} & I & & \\ & & \ddots & \ddots & 0 \\ & & & I & \\ I & & & & L^{-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad (5.1.21b)$$

As before, when the elimination procedure is applied to (5.1.21a) it yields the result, (bearing in mind that the condition (5.1.10) should be satisfied, i.e. have it is required that $||L|| < 1$ for stability).

$$\left. \begin{aligned} y_N &= (I + s_{L^N}^{N,N})^{-1} \tilde{z}_N \\ \text{or } (I + s_{L^N}^{N,N}) y_N &= \tilde{z}_N \\ \text{and } y_j &= \tilde{z}_j - s_{L^j}^{j,j} y_N, \quad j=N-1, N-2, \dots, 1 \end{aligned} \right\} s^k = \begin{cases} 1, & \text{for } k \text{ odd} \\ -1, & \text{for } k \text{ even,} \end{cases} \quad (5.1.22)$$

where \tilde{z}_j is given by

$$\left. \begin{aligned} \tilde{z}_1 &= z_1 \\ \text{and } \tilde{z}_j &= z_j - L \tilde{z}_{j-1}, \quad j=2, 3, \dots, N \end{aligned} \right\} \quad (5.1.23)$$

Similarly, for the system (4.5.21b), if we define \tilde{y}_j

$$\left. \begin{aligned} \tilde{y}_N &= L y_N \\ \text{and } \tilde{y}_j &= L(y_j - \tilde{y}_{j+1}), \quad j=N-1, N-2, \dots, 1, \end{aligned} \right\} \quad (5.1.24)$$

then the solution vector x_j can be obtained from the relations,

$$\left. \begin{aligned} x_1 &= (I + s_{L^N}^{N,N})^{-1} y_1 \\ \text{or } (I + s_{L^N}^{N,N}) x_1 &= \tilde{y}_1 \\ \text{and } x_j &= \tilde{y}_j - s_{L^j}^{j,j} x_1, \quad j=2, 3, \dots, N, \end{aligned} \right\} \quad (5.1.25)$$

and s is defined in (5.1.22).

Okolie (1978) applied the *spectral resolution method* to the latter procedure which reduced the order of operations to $O(4m^2N)$. Thus, in the following we shall apply this method to the procedure given earlier in this section.

Spectral Resolution Method

We first describe this method in connection with the procedure above (from (5.1.18) to (5.1.25)) which has been studied by Okolie (1978, Chapter 5) based on the work of Buzbee et al (1970).

We assume that the eigenvalues of the submatrix B are $\lambda_1, \lambda_2, \dots, \lambda_m$ and we define a diagonal submatrix Λ_B (of order m) such that

$$\Lambda_B = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m). \quad (5.1.26)$$

Also we assume the submatrix Q (of order m) to be orthogonal (i.e. $Q^T = Q^{-1}$) and consists of the eigenvectors of B , then the orthogonal transformation (see Definition 2.4.2)

$$B = Q \Lambda_B Q^T \quad (5.1.27)$$

exists.

From (5.1.27) the following results can be obtained (Okolie (1978)), for any integer k ,

$$(i) \quad B^k = Q \Lambda_B^k Q^T, \quad (5.1.28)$$

and

$$(ii) \quad \left. \begin{aligned} P(B) &= P(\Lambda_B) Q^T \\ \text{and } P(B^k) Q P(\Lambda_B^k) Q^T & \end{aligned} \right\} \quad (5.1.29)$$

where $P(B)$ and $P(\Lambda_B)$ are polynomials of degree k in the matrix B and Λ_B respectively (see Noble (1969)).

Furthermore, if we denote the eigenvalues of the submatrix L in (5.1.20a) by μ_j , $j=1,2,\dots,m$ and define a diagonal matrix Λ_L such that $\Lambda_L = \text{diag}(\mu_1, \mu_2, \dots, \mu_m)$, by virtue of (5.1.27)-(5.1.29) then L can be expressed as follows (Okolie (1978)),

$$\left. \begin{aligned} L &= Q \Lambda_L Q^T \\ L^k &= Q \Lambda_L^k Q^T, \quad k \text{ is any integer } > 0 \end{aligned} \right\} \quad (5.1.30)$$

and

where

$$\left. \begin{aligned} \Lambda_L &= P(\Lambda_B) = \text{diag}(\mu_1, \mu_2, \dots, \mu_m) \\ \Lambda_L^k &= P(\Lambda_B^k) = \text{diag}(\mu_1^k, \mu_2^k, \dots, \mu_m^k), \quad k \text{ is any integer,} \end{aligned} \right\} \quad (5.1.31a)$$

and

where μ_j in terms of λ_j are given by

$$\mu_j = 0.5(\lambda_j - (\lambda_j^2 - 4)^{\frac{1}{2}}) . \quad (5.1.31b)$$

Similarly, the matrix $(I + s^N L^N)$ in (5.1.22) (or (5.1.25)) can be expressed as follows,

$$I + s^N L^N = Q \Lambda_{I+L} Q^T , \quad (5.1.32)$$

where $\Lambda_{I+L} \equiv \text{diag}(1 + s^N \mu_1, 1 + s^N \mu_2, \dots, 1 + s^N \mu_m)$, ($\equiv I + s^N \Lambda_L$).

Reverting now to the solution procedure given at the beginning of this subsection and assuming that the submatrices B and C are commutative (i.e. $BC=CB$) which implies that B and C have a common set of m independent eigenvectors (Noble (1969), page 342). In this case the orthogonal matrix Q consists of columns which are the set of eigenvectors of B and C (Okolie (1978)), then we have,

$$\left. \begin{aligned} Q^T B Q &= \Lambda_B \equiv \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m) \\ \text{and } Q^T C Q &= \Lambda_C \equiv \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m) \end{aligned} \right\} . \quad (5.1.32)$$

Let the matrices P_1 and P_2 (of order m) be defined as

$$\left. \begin{aligned} P_1 &= 0.5(B + 2C)^{\frac{1}{2}} \\ P_2 &= 0.5(B - 2C)^{\frac{1}{2}} \end{aligned} \right\} , \quad (5.1.33)$$

then from (5.1.8), Q_0 and Q_1 in terms of P_1 and P_2 are given as follows,

$$\left. \begin{aligned} Q_0 &= P_1 + P_2 \\ Q_1 &= P_1 - P_2 \end{aligned} \right\} . \quad (5.1.34)$$

Furthermore, we set Λ_{P_1} and Λ_{P_2} as diagonal matrices possessing the same eigenvalues as P_1 and P_2 respectively such that

$$\left. \begin{aligned} Q^T P_1 Q &= \Lambda_{P_1} \\ Q^T P_2 Q &= \Lambda_{P_2} \end{aligned} \right\} . \quad (5.1.35)$$

Then, from (5.1.34) and (5.1.35), we have

$$\left. \begin{aligned} Q^T Q_0 Q &= Q^T (P_1 + P_2) Q \equiv Q^T f(P_1, P_2) Q = f(\Lambda_{P_1}, \Lambda_{P_2}) \\ Q^T Q_1 Q &= Q^T (P_1 - P_2) Q \equiv Q^T g(P_1, P_2) Q = g(\Lambda_{P_1}, \Lambda_{P_2}) \end{aligned} \right\} , \quad (5.1.36)$$

where

$$\left. \begin{aligned} f(\Lambda_{P_1}, \Lambda_{P_2}) &= \Lambda_{P_1} + \Lambda_{P_2} \\ g(\Lambda_{P_1}, \Lambda_{P_2}) &= \Lambda_{P_1} + \Lambda_{P_2} \end{aligned} \right\} \quad (5.1.37)$$

and f, g are polynomials in P_1 and P_2 as defined in (5.1.36).

Thus, Q_0 and Q_1 can be expressed in the form (for any integer k)

$$\left. \begin{aligned} Q_0 &= Q \Lambda_{Q_0} Q^T, \\ Q_1 &= Q \Lambda_{Q_1} Q^T, \\ Q_0^k &= Q \Lambda_{Q_0}^k Q^T, \\ Q_1^k &= Q \Lambda_{Q_1}^k Q^T, \end{aligned} \right\} \quad (5.1.38)$$

and

where

$$\Lambda_{Q_0}^k = f(\Lambda_{P_1}^k, \Lambda_{P_2}^k) \equiv \text{diag} (\sigma_1^k, \sigma_2^k, \dots, \sigma_m^k),$$

and

$$\Lambda_{Q_1}^k = g(\Lambda_{P_1}^k, \Lambda_{P_2}^k) \equiv \text{diag} (\xi_1^k, \xi_2^k, \dots, \xi_m^k),$$

with f and g as defined in (5.1.36),

and σ_j and ξ_j are the eigenvalues of Q_0 and Q_1 respectively.

In addition σ_j and ξ_j are expressed in terms of λ_j (the eigenvalues of B) and $\bar{\lambda}_j$ (the eigenvalues of C) and are given by (see (5.1.8))

$$\left. \begin{aligned} \sigma_j &= 0.5[(\lambda_j + 2\bar{\lambda}_j)^{\frac{1}{2}} + (\lambda_j - 2\bar{\lambda}_j)^{\frac{1}{2}}] \\ \xi_j &= 0.5[(\lambda_j + 2\bar{\lambda}_j)^{\frac{1}{2}} - (\lambda_j - 2\bar{\lambda}_j)^{\frac{1}{2}}] \end{aligned} \right\} \quad j=1, 2, \dots, m \quad (5.1.39)$$

which is analogous to the form (4.5.31b).

Similarly, if the diagonal matrix $\Lambda_{\tilde{Q}}$ is taken such that it comprises the eigenvalues of \tilde{Q} in (5.1.11), i.e.,

$$\tilde{Q} = Q \Lambda_{\tilde{Q}} Q^T, \quad (5.1.40)$$

thus from (5.1.38), (5.1.11) and (5.1.40) we have

$$\begin{aligned} \Lambda_{Q_0} \Lambda_{\tilde{Q}} &= \Lambda_{Q_1} \\ \Lambda_{\tilde{Q}} &= (\Lambda_{Q_0})^{-1} \Lambda_{Q_1}. \end{aligned} \quad (5.1.41)$$

or

Moreover, if ω_j are the elements of $\Lambda_{\tilde{Q}}$ (i.e. the eigenvalues of \tilde{Q}) then from (5.1.41) and (5.1.39), we can express ω_j in terms of σ_j and ξ_j as follows:-

$$\begin{aligned}\omega_j &= [(\lambda_j + 2\bar{\lambda}_j)^{\frac{1}{2}} + (\lambda_j - 2\bar{\lambda}_j)^{\frac{1}{2}}][(\lambda_j + 2\bar{\lambda}_j)^{\frac{1}{2}} - (\lambda_j - 2\bar{\lambda}_j)^{\frac{1}{2}}] \\ &= \bar{\lambda}_j / [\lambda_j + (\lambda_j^2 - 4\bar{\lambda}_j^2)^{\frac{1}{2}}], \quad j=1,2,\dots,m,\end{aligned}\quad (5.1.42)$$

where ω_j , λ_j and $\bar{\lambda}_j$ are the corresponding eigenvalues of \tilde{Q}_1, Q_0 and Q_1 respectively.

In the light of the above analysis, the 5 steps of the procedure given earlier may be replaced by the following:

Step 1' Determine $\Lambda_{Q_0} = \text{diag}(\sigma_1, \dots, \sigma_m)$,

$$\Lambda_{Q_1} = \text{diag}(\xi_1, \dots, \xi_m),$$

$$\text{and } \Lambda_{\tilde{Q}} = \text{diag}(\omega_1, \dots, \omega_m),$$

from the appropriate relation in (5.1.39) and (5.1.42).

Step 2' (i) Obtain \tilde{z}_j , $j=1,2,\dots,N$, from

$$\tilde{z}_j = Q_0^{-1} z_j = Q(\Lambda_{Q_0})^{-1} Q^T z_j,$$

(ii) \tilde{z}_j , $j=1,2,\dots,N$ from

$$\tilde{z}_N = \tilde{z}_N,$$

$$\begin{aligned}\tilde{z}_j &= \tilde{z}_j - \tilde{Q} \tilde{z}_{j+1} \\ &= \tilde{z}_j - Q \Lambda_{\tilde{Q}} Q^T \tilde{z}_{j+1}, \quad j=N-1, N-2, \dots, 1\end{aligned}$$

and

(iii) y_j , $j=1,2,\dots,N$,

$$(F_1 + I)y = \tilde{z}_1.$$

Again, the matrix $F_1 + I$ can be diagonalized as before such that

$$Q^T (F_1 + I) Q = \bar{\Lambda} \equiv \text{diag}(\mu_1, \dots, \mu_m),$$

where $\mu_j = 1 + s\omega_j^N$, $j=1,2,\dots,m$, thus

$$\begin{aligned}
 \underline{z}_1 &= (F_1 + I)^{-1} \tilde{\underline{z}}_1 \\
 &= Q^T (\bar{\Lambda})^{-1} Q \tilde{\underline{z}}_1 . \\
 \underline{z}_j &= \tilde{\underline{z}}_j - s Q \Lambda_Q^{N-j+1} Q^T \underline{z}_1, \quad j=2, \dots, N,
 \end{aligned}$$

Step 3' (i) Obtain $\tilde{\underline{y}}_j$, $j=1, 2, \dots, N$ from

$$\begin{aligned}
 Q_0 \tilde{\underline{y}}_j &= \underline{y}_j \\
 \text{or } \tilde{\underline{y}}_j &= Q_0^{-1} \underline{y}_j = Q(\Lambda_{Q_0})^{-1} Q^T \underline{y}_j,
 \end{aligned}$$

(ii) $\tilde{\underline{y}}_j$, $j=1, 2, \dots, N$ from

$$\begin{aligned}
 \tilde{\underline{y}}_1 &= \tilde{\underline{y}}_1, \\
 \tilde{\underline{y}}_j &= \tilde{\underline{y}}_j - \tilde{Q} \tilde{\underline{y}}_{j-1} \\
 &= \tilde{\underline{y}}_j - Q \Lambda_Q^{j-1} Q^T \tilde{\underline{y}}_{j-1}, \quad j=2, 3, \dots, N,
 \end{aligned}$$

(iii) \underline{x}_j , $j=1, 2, \dots, N$ from

$$\begin{aligned}
 \underline{x}_N &= (F_1 + I)^{-1} \tilde{\underline{y}}_N = Q^T (\bar{\Lambda}) Q \tilde{\underline{y}}_N \quad (\text{see (iii) in step 2' above}) \\
 \underline{x}_j &= \tilde{\underline{y}}_j - s Q \Lambda_Q^{N-j+1} Q^T \underline{x}_N.
 \end{aligned}$$

It is necessary to point out that the above steps involve the matrix vector multiplication, $Q^T \underline{z}_1$ in (iii) step 2' and $Q^T \underline{x}_N$ in (iii) step 3' and are computed only once; consequently the above procedure is estimated to be of computational complexity of $O(10m^2N)$ (excluding step 1').

5.2 ALGORITHM FIRM2

The system considered here is similar to the one given in the previous algorithm, but is non-periodic and has the form,

$$\begin{bmatrix} B & C & & 0 \\ & C & & \\ & & \ddots & \\ 0 & & & C & B \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} \quad (5.2.1)$$

where the submatrices B, C and the vectors $\underline{x}_j, \underline{z}_j, j=1,2,\dots,N$ are as defined in (5.1.1).

The factorization of the coefficient matrix differs from (5.1.3) and here involves two block rectangular matrices of size $mN \times (N+1)m$ and $(N+1)m \times mN$ respectively takes the following form,

$$\begin{bmatrix} B & C & & 0 \\ & C & & \\ & & \ddots & \\ 0 & & & C & B \end{bmatrix} = \begin{bmatrix} Q_0 & Q_1 & & 0 \\ & Q_1 & & \\ & & \ddots & \\ 0 & & & Q_1 & Q_0 \end{bmatrix}_{mN \times (N+1)m} \begin{bmatrix} Q_0 & & & 0 \\ Q_1 & & & \\ & 0 & & Q_1 & Q_0 \\ & & 0 & & Q_1 & Q_0 \end{bmatrix}_{(N+1)m \times mN} \quad (5.2.2)$$

$$= \begin{bmatrix} Q_0^2 + Q_1^2 & Q_1 Q_0 & & 0 \\ Q_0 Q_1 & & & \\ & & \ddots & \\ 0 & & & Q_0 Q_1 & Q_1^2 Q_0^2 + Q_1^2 \end{bmatrix}_{mN \times mN}$$

By equating the corresponding elements of the tridiagonal matrices on both sides of (5.2.2) we obtain the matrix equations, i.e.,

$$\left. \begin{aligned} Q_0^2 + Q_1^2 &= B \\ Q_0 Q_1 &= C \quad (\text{or } Q_1 Q_0 = C) \end{aligned} \right\} \quad (5.2.3)$$

Since (5.2.2) is exactly (5.1.4), thus the values of Q_0 and Q_1 are taken as given in (5.1.8).

However, the elimination process in the present algorithm is different to the one given in algorithm FICM3, since here we have two block under-determined and overdetermined systems (c.f. algorithm FIRMI). These two systems are:

$$\begin{bmatrix} Q_0 & Q_1 & & & & \\ & \ddots & \ddots & & & \\ & & 0 & & & \\ & & & \ddots & & \\ & 0 & & & Q_1 & Q_0 \\ & & & & & Q_1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ y_{N+1} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \\ z_{N+1} \end{bmatrix}, \quad (5.2.4a)$$

and

$$\begin{bmatrix} Q_0 & & & & & \\ Q_1 & & & & & \\ & \ddots & \ddots & & & \\ & & 0 & & & \\ & & & \ddots & & \\ & 0 & & & Q_1 & Q_0 \\ & & & & & Q_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \\ x_{N+1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ y_{N+1} \end{bmatrix}, \quad (5.2.4b)$$

where y_j , $j=1,2,\dots,N+1$, are the sub-vectors of length m as well as z_j and x_j .

If we now assume that the condition (5.1.10) is valid here and Q_0 is defined as in Section 5.1, (i.e. Q_0^{-1} exists) then the two systems in (5.2.4) may be modified into the following forms, respectively.

$$\begin{bmatrix} I & \tilde{Q} & & & & \\ & \ddots & \ddots & & & \\ & & 0 & & & \\ & & & \ddots & & \\ & 0 & & & \tilde{Q} & I \\ & & & & & \tilde{Q} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ y_{N+1} \end{bmatrix} = \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_N \\ \tilde{z}_{N+1} \end{bmatrix} \quad (5.2.5a)$$

and

$$\begin{bmatrix} \bar{I} & & & & \\ & \tilde{Q} & & & \\ & & 0 & & \\ & & & \ddots & \\ & & & & \tilde{Q} \\ & 0 & & & & \bar{I} \\ & & & & & & \tilde{Q} \\ & & & & & & & 0 \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_N \end{bmatrix} = \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_{N+1} \end{bmatrix} \quad (5.2.5b)$$

where

$$\left. \begin{aligned} \tilde{Q} &= Q_0^{-1} Q_1 \\ Q_0 \tilde{Q} &= Q_1 \end{aligned} \right\} \quad (5.2.6)$$

or

$$\left. \begin{aligned} \tilde{z}_j &= Q_0^{-1} z_j, \\ Q_0 \tilde{z}_j &= z_j, \quad j=1,2,\dots,N \end{aligned} \right\} \quad (5.2.7)$$

or

$$\left. \begin{aligned} \tilde{y}_j &= Q_0^{-1} y_j \\ Q_0 \tilde{y}_j &= y_j, \quad j=1,2,\dots,N+1 \end{aligned} \right\} \quad (5.2.8)$$

and

or

Furthermore, by considering the definition of the submatrices F_j in (5.1.14) the system (5.2.5a) can be taken a step further where the off-"diagonal" elements (i.e. Q) are eliminated and then to end up with the following form,

$$\begin{bmatrix} \bar{I} & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & 0 & & & \bar{I} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_{N-1} \\ F_N \end{bmatrix} \begin{bmatrix} \underline{y}_1 \\ \underline{y}_2 \\ \vdots \\ \underline{y}_{N+1} \end{bmatrix} = \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_N \end{bmatrix}, \quad (5.2.9)$$

where F_j , $j=1,2,\dots,N$ are given by (5.1.14) and \tilde{z}_j are given by

$$\left. \begin{aligned} \tilde{z}_N &= \tilde{z}_N \\ \tilde{z}_j &= \tilde{z}_j - \tilde{Q} \tilde{z}_{j+1}, \quad j=N-1,\dots,1 \end{aligned} \right\} \quad (5.2.10)$$

and

\tilde{z}_j given by (5.2.7).

Hence, from (5.2.9), \underline{y}_j , $j=1,2,\dots,N$ can be expressed in terms of \underline{y}_{N+1} as follows,

$$\left. \begin{aligned} y_1 &= \tilde{z}_1^{-F} y_{N+1} \\ y_2 &= \tilde{z}_2^{-F} y_{N+1} \\ &\vdots \\ y_N &= \tilde{z}_N^{-F} y_{N+1} \end{aligned} \right\} \quad (5.2.11)$$

On the other hand, the elimination process on the system (5.2.5b) takes place such that the elements Q are eliminated leaving the unit matrix; in this respect it can be noticed the right-hand side vector is changed and the final form for (5.2.5b) becomes,

$$\begin{bmatrix} I & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & \ddots & \\ & & & & I \\ & & 0 & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_{N+1} \end{bmatrix} \quad (5.2.12)$$

where

$$\left. \begin{aligned} \tilde{\tilde{y}}_1 &= \tilde{y}_1 \\ \tilde{\tilde{y}}_j &= \tilde{y}_j - \tilde{Q}_j \tilde{\tilde{y}}_{j-1}, \quad j=1,2,\dots,N, \end{aligned} \right\} \quad (5.2.13a)$$

$$\tilde{\tilde{y}}_{N+1} = \tilde{y}_{N+1} - \tilde{Q}_N \tilde{\tilde{y}}_N, \quad (5.2.13b)$$

and $\tilde{y}_j, j=1,2,\dots,N+1$ are given in (5.2.8).

Thus, it follows from (5.2.12) that the solution sub-vectors x_1, \dots, x_N , are equal to the corresponding sub-vectors $\tilde{\tilde{y}}_1, \dots, \tilde{\tilde{y}}_N$ and \tilde{y}_{N+1} is the null sub-vector, i.e.,

$$\left. \begin{aligned} x_1 &= \tilde{\tilde{y}}_1, \\ x_2 &= \tilde{\tilde{y}}_2, \\ &\vdots \\ x_N &= \tilde{\tilde{y}}_N, \end{aligned} \right\} \quad (5.2.14a)$$

$$\text{and} \quad 0 = \tilde{\tilde{y}}_{N+1}. \quad (5.2.14b)$$

In fact, in the relations (5.2.13) we can easily express each of the $\tilde{\tilde{y}}_j$ in terms of $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_j$, i.e.,

$$\left. \begin{aligned}
 \tilde{y}_1 &= \tilde{y}_1, \\
 \tilde{y}_2 &= \tilde{y}_2 + \tilde{Q}\tilde{y}_1 = \tilde{y}_2 + \tilde{Q}\tilde{y}_1, \\
 \tilde{y}_3 &= \tilde{y}_3 + \tilde{Q}\tilde{y}_2 = \tilde{y}_3 + \tilde{Q}\tilde{y}_2 + \tilde{Q}^2\tilde{y}_1, \\
 \tilde{y}_4 &= \tilde{y}_4 + \tilde{Q}\tilde{y}_3 = \tilde{y}_4 + \tilde{Q}\tilde{y}_3 + \tilde{Q}^2\tilde{y}_2 + \tilde{Q}^3\tilde{y}_1, \\
 \hline
 \tilde{y}_N &= \tilde{y}_N + \tilde{Q}\tilde{y}_{N-1} + \tilde{Q}^2\tilde{y}_{N-2} + \dots + \tilde{Q}^{N-1}\tilde{y}_1,
 \end{aligned} \right\} \quad (5.2.15a)$$

and

$$\tilde{y}_{N+1} = \tilde{y}_{N+1} + \tilde{Q}\tilde{y}_N + \tilde{Q}^2\tilde{y}_{N-1} + \dots + \tilde{Q}^N\tilde{y}_1. \quad (5.2.15b)$$

But $\tilde{y}_{N+1} = 0$ by virtue (5.2.14b) thus from (5.2.15a) we have

$$-\tilde{y}_{N+1} = \tilde{Q}\tilde{y}_N + \tilde{Q}^2\tilde{y}_{N-1} + \dots + \tilde{Q}^N\tilde{y}_1. \quad (5.2.16)$$

We now substitute for \tilde{y}_j , $j=1,2,\dots,N$ given by (5.2.8) so that

(5.2.16) will take the form,

$$-Q_0^{-1}y_{N+1} = \tilde{Q}Q_0^{-1}y_N + \tilde{Q}^2Q_0^{-1}y_{N-1} + \dots + \tilde{Q}^NQ_0^{-1}y_1,$$

or

$$\left. \begin{aligned}
 -y_{N+1} &= R_1y_N + R_2y_{N-1} + \dots + R_Ny_1 \\
 &= \sum_{j=1}^N R_j y_{N-j+1},
 \end{aligned} \right\} \quad (5.2.17a)$$

where $R_j = Q_0 \tilde{Q}^j Q_0^{-1}$. (5.2.17b)

If we now return to the equations in (5.2.11), and multiply the first equation, the second equation, up to the last equation by R_N, R_{N-1}, \dots, R_1 respectively (or say multiply the k^{th} equation by R_{N-k+1}), and add together we arrive at the following result after some rearrangement,

$$\sum_{j=1}^N R_{N-j+1} y_j = \sum_{j=1}^N R_{N-j+1} \tilde{z}_j - \left(\sum_{j=1}^N R_{N-j+1} F_j \right) y_{N+1}. \quad (5.2.18)$$

But the right-hand side in the last relation is equal to $-y_{N+1}$ by virtue of (5.2.17a) thus (5.2.18) becomes,

$$-y_{N+1} = \sum_{j=1}^N R_{N-j+1} \tilde{z}_j - \left(\sum_{j=1}^N R_{N-j+1} F_j \right) y_{N+1},$$

or with some rearrangement it may be written in the form,

$$(I - \sum_{j=1}^N R_{N-j+1} F_j) y_{N+1} = - \sum_{j=1}^N R_{N-j+1} \tilde{z}_j. \quad (5.2.19)$$

Further, if we substitute for R_j and F_j as given in (5.2.17b) and (5.1.14) respectively, then (5.2.19) becomes,

$$(I - \sum_{j=1}^N Q_0 \tilde{Q}^{N-j+1} Q_0^{-1} \tilde{Q}^{N-j+1}) y_{N+1} = - \sum_{j=1}^N \tilde{Q}^{N-j+1} \tilde{z}_j, \quad (5.2.20)$$

where \tilde{z}_j is given in (5.2.20), and s for any j of the summation operator is defined as

$$s = \begin{cases} (-1)^{j+1}, & \text{for } N \text{ odd} \\ (-1)^j, & \text{for } N \text{ even.} \end{cases}$$

It may be possible to do further simplification in the second term of the left-hand side of (5.2.20) by taking into account the fact that Q_0 and Q_1 are commutative due to (5.2.3) which consequently implies the equality

$$(\tilde{Q}) \quad Q_0^{-1} Q_1 = Q_1 Q_0^{-1}. \quad (5.2.21)$$

Therefore, the quantity $Q_0 \tilde{Q}^k Q_0^{-1} \tilde{Q}^k$, $k=1,2,\dots,N$ may be modified as follows,

$$\begin{aligned} Q_0 \tilde{Q}^k Q_0^{-1} \tilde{Q}^k &= Q_0 \underbrace{(Q_0^{-1} Q_1 \times Q_0^{-1} Q_1 \times \dots \times Q_0^{-1} Q_1)}_{k \text{ times}} Q_0^{-1} \tilde{Q}^k \\ &= \underbrace{(Q_1 Q_0^{-1} \times Q_1 Q_0^{-1} \times \dots \times Q_1 Q_0^{-1})}_{k \text{ times}} \tilde{Q}^k \\ &= (Q_0^{-1} Q_1 \times Q_0^{-1} Q_1 \times \dots \times Q_0^{-1} Q_1) \tilde{Q}^k \quad (\text{by (5.2.21)}) \\ &= \tilde{Q}^k \tilde{Q}^k = \tilde{Q}^{2k}. \end{aligned} \quad (5.2.22)$$

Thus, by virtue of (5.2.22) the relation (5.2.20) takes the form,

$$(I - \sum_{j=1}^N s \tilde{Q}^{2(N-j+1)}) y_{N+1} = - \sum_{j=1}^N \tilde{Q}^{N-j+1} \tilde{z}_j, \quad (5.2.23)$$

and s is as defined in (5.2.20).

However, with the system (5.2.23) (of order m) solved for y_{N+1} , it enables us to proceed for the computation of y_j , $j=1,2,\dots,N$ from (5.2.11). This is followed by considering the equations (5.2.8) to determine \tilde{y}_j ,

$j=1,2,\dots,N$ and then to (5.2.13b) to determine \tilde{y}_j , $j=1,2,\dots,N$ which are equal to the solution x_j by virtue of (5.2.14a).

The outline of the above procedure may be briefly represented by the following:

Step 1, Step 2 and Step 3 (see algorithm FICM3 in the previous subsections).

Step 4 Compute y_{N+1} by solving the linear system (5.2.23).

Step 5 Compute \tilde{z}_j , \tilde{z}_j and y_j , $j=1,2,\dots,N$ as follows,

$$Q_0 \tilde{z}_j = z_j, \quad j=1,2,\dots,N,$$

$$\tilde{z}_N = \tilde{z}_N, \quad \tilde{z}_j = \tilde{z}_j - Q \tilde{z}_{j+1}, \quad j=N-1,\dots,1$$

$$\text{and} \quad y_j = \tilde{z}_j^{-F} y_{N+1}, \quad j=1,2,\dots,N.$$

Step 6 Compute \tilde{y}_j and \tilde{y}_j ($=x_j$) as follows:

$$Q_0 \tilde{y}_j = y_j, \quad j=1,2,\dots,N$$

$$(x_j) \tilde{y}_j = \tilde{y}_j - Q \tilde{y}_{j-1}, \quad j=2,3,\dots,N,$$

$$\text{where} \quad (x_1) = \tilde{y}_1 = \tilde{y}_1.$$

The order of operations involved in the above procedure (excluding step 1) is approximately of $O(N(\frac{8}{3}m^3 + 4m^2))$ which is an improvement over that given by Isaacson & Keller (1966).

Finally, we point out that the spectral resolution method discussed in Section 5.1, can be applied to the system (5.2.1), provided that the conditions required by this method are fulfilled. Obviously, we can apply this method on the system (5.2.23), for example, and using the same notation, we obtain,

$$(I - \sum_{j=1}^N s Q \Lambda_Q^{2(N-j+1)} Q^T) y_{N+1} = - \sum_{j=1}^N Q \Lambda_Q^{N-j+1} Q^T \tilde{z}_j,$$

where the orthogonal matrix Q and the diagonal matrix Λ_Q are as defined in the previous section. Thus, by adopting the spectral resolution method the previous procedure may reduce the order of operations to $O(9m^2N)$ approximately.

5.3 ALGORITHM FICM4

As an extension to the system considered in algorithm FICM3 (Section 5.1), we shall consider here a periodic block-quindiagonal linear system of the form,

$$\begin{bmatrix} B & C & D \\ C & & \\ D & & \end{bmatrix} \begin{bmatrix} D & C \\ & D \\ & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}, \quad (5.3.1)$$

$$= \begin{bmatrix} Q_0^2 + Q_1^2 + Q_2^2 & Q_1 Q_0 + Q_2 Q_1 & Q_2 Q_0 & Q_0 Q_2 & Q_0 Q_1 + Q_1 Q_2 \\ Q_0 Q_1 + Q_1 Q_2 & 0 & & & Q_0 Q_2 \\ Q_0 Q_2 & & & & Q_2 Q_0 \\ Q_2 Q_0 & 0 & & & Q_1 Q_0 + Q_2 Q_1 \\ Q_1 Q_0 + Q_2 Q_1 & Q_0 Q_2 & Q_0 Q_1 + Q_1 Q_2 & Q_0^2 + Q_1^2 + Q_2^2 & \end{bmatrix}, \quad (5.3.3)$$

where the submatrices Q_0, Q_1 and Q_2 are each of order m .

By equating the two block-quindiagonal matrices in (5.3.3), and comparing corresponding elements yields the following relations,

$$\left. \begin{aligned} Q_0^2 + Q_1^2 + Q_2^2 &= B \\ Q_1 Q_0 + Q_2 Q_0 &= C \text{ (or } Q_0 Q_1 + Q_1 Q_2 = C) \\ Q_2 Q_0 &= D \text{ (or } Q_0 Q_2 = D) \end{aligned} \right\}. \quad (5.3.4)$$

and

Although, the system (5.3.4) consists of *matrix* equations, it can be reduced to a simpler form similar to the quindiagonal point case (i.e. when $r=2$) in the algorithm FICM1 (see subsection 4.2.1 or subsection 4.2.4 equation (4.2.38)), this is due to the fact that the expression,

$$(Q_0^2 + Q_1^2 + Q_2^2) = Q_0^2 + Q_2^2 + (Q_1 Q_0 + Q_2 Q_1) + (Q_0 Q_1 + Q_1 Q_2) + Q_2 Q_0 + Q_0 Q_2, \quad (5.3.5)$$

holds true by virtue of the commutative property of the matrices

$(Q_1 Q_0 + Q_2 Q_1)$ and $(Q_0 Q_1 + Q_1 Q_2)$, Q_2 and Q_0 which are confirmed by the second and third equations respectively of (5.3.4).

Thus, the equations (5.3.4) under the validity of (5.3.5) can be replaced by the following equation (see subsection 4.2.1),

$$\left. \begin{aligned} Q_0 + Q_2 &= 0.5 \{ [B + 2(D+C)]^{\frac{1}{2}} + [B + 2(D-C)]^{\frac{1}{2}} \} \equiv G \\ Q_1 &= 0.5 \{ [B + 2(D+C)]^{\frac{1}{2}} - [B + 2(D-C)]^{\frac{1}{2}} \} \\ Q_2 Q_0 &= D \text{ (or } Q_0 Q_2 = D) \end{aligned} \right\}. \quad (5.3.6)$$

in order to determine the submatrices Q_0, Q_1 and Q_2 , provided that the matrices $[B + 2(D+C)]$ and $[B + 2(D-C)]$ are positive definite and their roots

are defined, notably by the strictly inequality (5.3.2).

It is clear from (5.3.6) that the submatrix Q_1 is determined already, whilst Q_0 and Q_2 are defined as follows:

$$\left. \begin{aligned} Q_0 &= 0.5(G + [G^2 - 4D]^{\frac{1}{2}}) \\ \text{and} \quad Q_2 &= 0.5(G - [G^2 - 4D]^{\frac{1}{2}}) \end{aligned} \right\} , \text{ provided } ||G^2|| > 4 ||D||, \quad (5.3.7)$$

where G is a constant matrix defined in the first of (5.3.6).

In fact, the choice of Q_0 (or Q_2) in the form defined in (5.3.7) does satisfy the quadratic matrix equation in Q_0 (or Q_2) derived from the first and the last equations of (5.3.6), i.e.,

$$Q_0^2 - GQ_0 + D = 0 \quad (\text{or } Q_0^2 - Q_0G + D = 0), \quad (5.3.8)$$

taking into account the commutative property of the matrices Q_0 and Q_2 .

It is assumed the same procedure of Section 2.5 is applicable here to evaluate the appropriate square root matrix in (5.3.6) and (5.3.7) as long as the relevant matrix is positive definite. This implies eventually that Q_0 is positive definite.

Two Alternative Block Systems for (5.3.1)

When in the system (5.3.1) the coefficient matrix is replaced by its two factors given in (5.3.3), we can formulate the following two systems where coefficient matrices are of upper and lower circulant block type respectively, i.e.,

$$\begin{bmatrix} Q_0 & Q_1 & Q_2 & & 0 \\ & Q_2 & Q_1 & Q_0 & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}, \quad (5.3.9a)$$

\tilde{z}_j and \tilde{y}_j , $j=1,2,\dots,N$ are defined as in (5.1.12) and (5.1.13) respectively.

As shown in the previous section, the implication of the elimination procedure is to get rid of the off-diagonal elements, precisely the submatrices \tilde{Q}_1 and \tilde{Q}_2 in both systems of (5.3.10). We confine ourselves to the algebraic formulation of this procedure. This may require the introduction of the submatrices F_i and G_i (each of order m), $i=1,2,\dots,N$, and the sub-vectors \hat{z}_i (each of length m), $i=1,2,\dots,N$ as follows,

$$\left. \begin{aligned} F_N &= \tilde{Q}_1, \quad G_N = \tilde{Q}_2, \quad \hat{z}_N = \tilde{z}_N, \\ F_{N-1} &= (-\tilde{Q}_1)F_N + \tilde{Q}_2, \quad G_{N-1} = (-\tilde{Q}_1)G_N, \quad \hat{z}_{N-1} = \tilde{z}_{N-1} + (-\tilde{Q}_1)\hat{z}_N, \\ F_j &= (-\tilde{Q}_1)F_{j+1} + (-\tilde{Q}_2)F_{j+2} + H, \quad \text{where } H = \begin{cases} I \text{ (unit matrix) for } j=1 \\ 0 \text{ (null matrix) otherwise} \end{cases} \\ G_j &= (-\tilde{Q}_1)G_{j+1} + (-\tilde{Q}_2)G_{j+2} + K, \quad \text{where } K = \begin{cases} I, \text{ for } j=2 \\ \tilde{Q}_1, \text{ for } j=1 \\ 0 \text{ (null) otherwise} \end{cases} \\ \text{and } \hat{z}_j &= \tilde{z}_j + (-\tilde{Q}_1)\tilde{z}_{j+1} + (-\tilde{Q}_2)\tilde{z}_{j+2} \end{aligned} \right\} \quad j=N-2, N-3, \dots, 1. \quad (5.3.12)$$

where \tilde{z}_j , $j=1,2,\dots,N$ are the components of the right-hand side vector in (5.3.10a) and \tilde{Q}_1 and \tilde{Q}_2 are given by (5.3.11).

Hence the system (5.3.10a) takes the form,

$$\begin{bmatrix} F_1 & G_1 & & & \\ F_2 & G_2 & & & \\ F_3 & G_3 & & & \\ \vdots & \vdots & & & \\ F_N & G_N & & & \end{bmatrix} \begin{bmatrix} \hat{z}_1 \\ \hat{z}_2 \\ \vdots \\ \hat{z}_N \end{bmatrix} = \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_N \end{bmatrix}, \quad (5.3.13)$$

where F_j, G_j and \hat{z}_j , $j=1,2,\dots,N$ are given in (5.3.12).

Further, we eliminate G_1 so that (5.3.13) would result in a triangular form, that is,

$$\begin{bmatrix} \hat{F}_1 \\ F_2 & G_2 \\ F_3 & G_3 & I & 0 \\ \vdots & \vdots & \ddots & \ddots \\ F_N & F_N & 0 & I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} \hat{z}_1 \\ \hat{z}_2 \\ \hat{z}_3 \\ \vdots \\ \hat{z}_N \end{bmatrix}, \quad (5.3.14a)$$

$$\left. \begin{aligned} \text{where } \hat{F}_1 &= F_1 - G_1 G_2^{-1} F_2, \\ \text{and } \hat{z}_1 &= z_1 - G_1 G_2^{-1} z_2, \end{aligned} \right\} \quad (5.3.14b)$$

provided G_2 is non-singular.

The components of the auxiliary vector, y_1, \dots, y_N can be obtained from (5.3.14a) by the forward substitution scheme, i.e.,

$$\left. \begin{aligned} \hat{F}_1 y_1 &= \hat{z}_1, \text{ which is solved for } y_1 \\ \text{and } G_2 y_2 &= \hat{z}_2 - F_2 y_1, \text{ which is solved for } y_2 \\ \text{whilst } y_k, k=3,4,\dots,N &\text{ are obtained from,} \\ y_k &= \hat{z}_k - (F_k y_1 + G_k y_2). \end{aligned} \right\} \quad (5.3.15)$$

Similarly, if we define the sub-vectors $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N, \hat{y}_1, \hat{y}_2, \dots, \hat{y}_N$ and \hat{y}_N as follows:-

$$\left. \begin{aligned} \hat{y}_1 &= \tilde{y}_1 \\ \hat{y}_2 &= \tilde{y}_2 + (-\tilde{Q}_1) \hat{y}_1 \\ \hat{y}_j &= \tilde{y}_j + (-\tilde{Q}_1) y_{j-1} + (-\tilde{Q}_2) y_{j-2}, \quad j=3,4,\dots,N \\ \hat{y}_N &= \hat{y}_N - G_1 G_2^{-1} \hat{y}_{N-1}, \end{aligned} \right\} \quad (5.3.16a)$$

where $\tilde{y}_j, j=1,2,\dots,N$ are defined in (5.3.16a)

then the system (5.3.14b) can be reduced to its final form where the coefficient matrix now has an upper triangular form, i.e.,

The number of operations involved in the above procedure (excluding step 1) is $O(\frac{38}{3} m^3 N)$. Again, if the first system in step 4 and step 5 (which solves for \tilde{z}_j and \tilde{y}_j respectively) are solved efficiently such that the coefficient matrix Q_0 is factorized just once, then the complexity order may be brought down to $O((12m^3 + 2m^2)N)$.

It may be possible to extend the spectral resolution method discussed in Section 5.1, (for the periodic block-tridiagonal system) to the periodic block-quindiagonal system represented by (5.3.1). This extension may occur under some restrictions imposed on the relationship between the submatrices B, C and D in (5.3.1). For example, it is required that (i) B and C commute, and their product to commute with D, (ii) B and D commute, and their product to commute with C, or (iii) C and D to commute, and their product to commute with B. This is equivalent to the statement any two of three matrices B, C, D to commute, and their product also to commute with the third matrix. Subsequently, the desired condition may be formulated as

$$\left. \begin{aligned} \text{(i)} \quad BC &= CB \quad \text{and} \quad DBC = BCD, \\ \text{(ii)} \quad BD &= DB \quad \text{and} \quad CBD = BDC, \\ \text{(iii)} \quad DC &= CD \quad \text{and} \quad BDC = DCB. \end{aligned} \right\} \quad (5.3.18)$$

If any of the conditions of (5.3.18) are fulfilled, then we conclude that the three submatrices have a common set of m independent eigenvectors, and thus it is possible to construct an orthogonal matrix Q (see section 5.1), whose columns comprise of the set of eigenvectors of B, C and D. (N.B. if D is a unit matrix, then Q is restricted to B and C, similarly for B and C also), such that,

$$\left. \begin{aligned} Q^T B Q &= \Lambda_B \equiv \text{diag}(\lambda_1, \dots, \lambda_m) \\ Q^T C Q &= \bar{\Lambda}_C \equiv \text{diag}(\bar{\lambda}_1, \dots, \bar{\lambda}_m) \\ Q^T D Q &= \bar{\bar{\Lambda}}_D \equiv \text{diag}(\bar{\bar{\lambda}}_1, \dots, \bar{\bar{\lambda}}_m) \end{aligned} \right\}, \quad (5.3.19)$$

where $\lambda_j, \bar{\lambda}_j, \bar{\bar{\lambda}}_j$, $j=1, 2, \dots, m$ are the eigenvalues of B, C and D respectively.

Hence, by following a similar procedure as given in Section 5.1, we can express the eigenvalues $\sigma_i, \bar{\sigma}_i$ and $\bar{\bar{\sigma}}_i$ of the submatrices Q_0, Q_1 and Q_2 respectively, in terms of the eigenvalues of B, C and D as given in (5.3.19). This is given as follows, taking into account that Q_1 is given by (5.3.6), Q_0 and Q_2 by (5.3.7).

$$\left. \begin{aligned} \bar{\sigma}_i &= 0.5\{[\lambda_i + 2(\bar{\bar{\lambda}}_i + \bar{\lambda}_i)]^{\frac{1}{2}} - [\lambda_i + 2(\bar{\bar{\lambda}}_i - \bar{\lambda}_i)]^{\frac{1}{2}}\} \\ \sigma_i \text{ (or } \bar{\bar{\sigma}}_i) &= 0.25\{[\lambda_i + 2(\bar{\bar{\lambda}}_i + \bar{\lambda}_i)]^{\frac{1}{2}} + [\lambda_i + 2(\bar{\bar{\lambda}}_i - \bar{\lambda}_i)]^{\frac{1}{2}}\} + \text{(or -)} \\ &\quad 0.5\{[0.5\lambda_i + 2(\bar{\bar{\lambda}}_i + \bar{\lambda}_i)]^{\frac{1}{2}} + 0.5[\lambda_i + 2(\lambda_i - \bar{\lambda}_i)]^{\frac{1}{2}}\}^2 - 4\bar{\bar{\lambda}}_i\}^{\frac{1}{2}}. \end{aligned} \right\} \quad (5.3.20)$$

In fact, the determination of these three eigenvalues, could replace step 1 of the procedure in the present subsection, and the continuation of the remaining steps can proceed in an analogous way to the appropriate steps for the periodic block-tridiagonal case (see Section 5.1).

5.4 ALGORITHM FIRM3

The type of linear systems considered in this algorithm are non-periodic block-quindiagonal, which in fact, are similar to (5.3.1), except that the coefficient matrix is non-circulant, i.e.,

$$\begin{bmatrix} B & C & D & & 0 \\ & C & D & & \\ & D & & & \\ & & 0 & & \\ & & & D & C & B \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_N \end{bmatrix} = \begin{bmatrix} \underline{z}_1 \\ \underline{z}_2 \\ \vdots \\ \underline{z}_N \end{bmatrix}, \quad (5.4.1)$$

where the coefficient matrix is a constant and symmetric block-quindiagonal matrix of order mN , with B, C, D real block-submatrices of order m and each sub-vector \underline{x}_i and \underline{z}_i has m components as in (5.3.1).

It is assumed that the matrix in (5.4.1) is a block-diagonally dominant which can be ensured as long as the inequality (5.3.2) is true. By taking this property into account, we proceed to factorize the block-quindiagonal matrix in (5.4.1). The factorization of this matrix takes place in an analogous way to the procedure adopted in algorithm FIRM2 (Section 5.2) to obtain two rectangular block matrices of size $mN \times (mN+2m)$ and $(mN+2m) \times mN$ respectively, or precisely the coefficient matrix of (5.4.1) can be factorized as follows:

$$\begin{bmatrix} B & C & D & & 0 \\ & C & D & & \\ & D & & & \\ & & 0 & & \\ & & & D & C & B \end{bmatrix} = \begin{bmatrix} Q_0 & Q_1 & Q_2 & & 0 \\ & Q_1 & Q_2 & & \\ & & Q_2 & & \\ & & & Q_1 & Q_2 \\ & 0 & & & Q_0 & Q_1 & Q_2 \end{bmatrix}_{mN \times (mN+2m)}$$

$$= \begin{bmatrix} Q_0^2 + Q_1^2 + Q_2^2 & Q_1 Q_0 + Q_2 Q_1 & Q_2 Q_0 & 0 \\ Q_0 Q_1 + Q_1 Q_2 & & & \\ Q_0 Q_2 & & & \\ 0 & & & \end{bmatrix} \quad (5.4.2)$$

where the submatrices Q_0, Q_1 and Q_2 are each of order m .

It can be readily noticed that by equating the corresponding elements of the two block-quindiagonal matrices on both sides of (5.4.2) leads to three matrix equations to be satisfied. These equations are exactly as given in (5.3.4), thus Q_0, Q_1 and Q_2 can be determined from (5.3.6) (for Q_1) and (5.3.7) (for Q_0 and Q_2). Also we shall assume as in Section 5.3, that Q_0 is non-singular and the value of its norm exceeds the value of the norm of both Q_1 and Q_2 so that the stability of the following elimination process is guaranteed.

First of all, we replace the given block system (5.4.1) by two alternative systems whose coefficient matrices are, respectively, the 'upper' and 'lower' block triangular factors given in (5.4.2). These two systems which are overdetermined and underdetermined by $2m$ have the following form respectively,

$$\begin{bmatrix} Q_0 & Q_1 & Q_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 0 & & & \\ & & & Q_2 & & \\ & & & Q_1 & & \\ & 0 & & Q_0 & Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ y_{N+1} \\ y_{N+2} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}, \quad (5.4.3a)$$

and

$$\begin{bmatrix} Q_0 & & & & & \\ & Q_1 & & & & \\ & & Q_2 & & & \\ & & & 0 & & \\ & & & & Q_1 & Q_0 \\ & 0 & & & Q_2 & Q_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ y_{N+1} \\ y_{N+2} \end{bmatrix}, \quad (5.4.3b)$$

where the sub-vector components of the auxiliary vector, y_1, \dots, y_{N+2} are each of length m .

Since Q_0 is assumed to be non-singular, the systems (5.4.3a) and (5.4.3b) may be modified to have the following form respectively (c.f. the systems in (5.2.5b)),

$$\begin{bmatrix} I & \tilde{Q}_1 & \tilde{Q}_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 0 & & & \\ & & & \tilde{Q}_2 & & \\ & & & \tilde{Q}_1 & & \\ & 0 & & I & \tilde{Q}_1 & \tilde{Q}_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ y_{N+1} \\ y_{N+2} \end{bmatrix} = \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_N \end{bmatrix}, \quad (5.4.4a)$$

and

$$\begin{bmatrix} \tilde{\bar{x}}_1 \\ \tilde{\bar{x}}_2 \\ \vdots \\ \tilde{\bar{x}}_N \end{bmatrix} = \begin{bmatrix} \tilde{\bar{y}}_1 \\ \tilde{\bar{y}}_2 \\ \vdots \\ \tilde{\bar{y}}_N \\ \tilde{\bar{y}}_{N+1} \\ \tilde{\bar{y}}_{N+2} \end{bmatrix}, \quad (5.4.4b)$$

where

$$\left. \begin{aligned} \tilde{Q}_1 &= Q_0^{-1} Q_1, \quad \tilde{Q}_2 = Q_0^{-1} Q_2, \\ Q_0 \tilde{Q}_1 &= Q_1, \quad Q_0 \tilde{Q}_2 = Q_2. \end{aligned} \right\} \quad (5.4.5)$$

or

Also, we have

$$\left. \begin{aligned} \tilde{z}_j &= Q_0^{-1} z_j, \\ Q_0 \tilde{z}_j &= z_j, \quad j=1, 2, \dots, N \end{aligned} \right\} \quad (5.4.6)$$

or

and

$$\left. \begin{aligned} \tilde{\underline{y}}_j &= Q^{-1} \underline{y}_j, \\ Q_0 \tilde{\underline{y}}_j &= \underline{y}_j, \quad j=1, 2, \dots, N+2. \end{aligned} \right\} \quad (5.4.7)$$

Further, we reconsider the sub-matrices F_j, G_j and the sub-vectors $\hat{z}_j, j=1,2,\dots,N$ as defined in (5.3.12) (but the third term H in F_j and K in G_j are discarded), then the off-"diagonal" elements (i.e. \tilde{Q}_1 and \tilde{Q}_2) in the system (5.4.4a) can be eliminated such that this system would assume the form.

$$\begin{bmatrix} \mathbf{I} & & & & & \\ & \mathbf{0} & & & & \\ & & \vdots & & & \\ & & & \mathbf{F}_1 & & \mathbf{G}_1 \\ & & & \mathbf{F}_2 & & \mathbf{G}_2 \\ & & & & \ddots & \\ & & & & & \mathbf{F}_{N-1} & \mathbf{G}_{N-1} \\ & & & & & \mathbf{F}_N & \mathbf{G}_N \\ & & & & & & \\ & \mathbf{0} & & & & & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \\ \mathbf{y}_{N+1} \\ \mathbf{y}_{N+2} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{z}}_1 \\ \hat{\mathbf{z}}_2 \\ \vdots \\ \hat{\mathbf{z}}_N \end{bmatrix} \quad (5.4.8)$$

To express y_j , $j=1,2,\dots,N$ in terms of y_{N+1} and y_{N+2} , i.e. from (5.4.8) we have,

$$\left. \begin{aligned} \underline{y}_1 &= \hat{\underline{z}}_1 - (F_1 \underline{y}_{N+1} + G_1 \underline{y}_{N+2}) \\ \underline{y}_2 &= \hat{\underline{z}}_2 - (F_2 \underline{y}_{N+1} + G_2 \underline{y}_{N+2}) \\ \underline{y}_N &= \hat{\underline{z}}_N - (F_N \underline{y}_{N+1} + G_N \underline{y}_{N+2}) \end{aligned} \right\} \quad (5.4.9)$$

Then in system (5.4.4b) after eliminating the elements \tilde{Q}_1 and \tilde{Q}_0 of the coefficient matrix, the final form of the system becomes

$$\begin{bmatrix} \text{I} & & & & & \\ & 0 & & & & \\ & & \ddots & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & \text{I} \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_N \end{bmatrix} = \begin{bmatrix} \hat{\underline{y}}_1 \\ \hat{\underline{y}}_2 \\ \vdots \\ \hat{\underline{y}}_N \\ \hat{\underline{y}}_{N+1} \\ \hat{\underline{y}}_{N+2} \end{bmatrix}, \quad (5.4.10)$$

$$\text{where} \quad \left. \begin{aligned} \hat{\underline{y}}_1 &= \tilde{\underline{y}}_1, \\ \hat{\underline{y}}_2 &= \tilde{\underline{y}}_2 + (-\tilde{Q}_1) \hat{\underline{y}}_1, \end{aligned} \right\} \quad (5.4.11a)$$

$$\left. \begin{aligned} \hat{\underline{y}}_j &= \tilde{\underline{y}}_j + (-\tilde{Q}_1) \hat{\underline{y}}_{j-1} + (-\tilde{Q}_2) \hat{\underline{y}}_{j-2}, \quad j=3,4,\dots,N, \\ \hat{\underline{y}}_{N+1} &= \tilde{\underline{y}}_{N+1} + (-\tilde{Q}_1) \hat{\underline{y}}_N + (-\tilde{Q}_2) \hat{\underline{y}}_{N-1} \end{aligned} \right\} \quad (5.4.11b)$$

$$\text{and} \quad \hat{\underline{y}}_{N+2} = \tilde{\underline{y}}_{N+2} + (-\tilde{Q}_2) \hat{\underline{y}}_N,$$

where \tilde{Q}_1, \tilde{Q}_2 and $\tilde{\underline{y}}_j$, $j=1,2,\dots,N+2$ are as defined in (5.4.4b).

Thus, it follows from (5.4.10) that the solution sub-vectors $\underline{x}_1, \dots, \underline{x}_N$ are equal to the corresponding components, $\hat{\underline{y}}_1, \dots, \hat{\underline{y}}_N$ and the redundant sub-vectors $\hat{\underline{y}}_{N+1}$ and $\hat{\underline{y}}_{N+2}$ are null (i.e. of zero components), i.e.,

$$\left. \begin{aligned} \underline{x}_1 &= \hat{\underline{y}}_1, \\ \underline{x}_2 &= \hat{\underline{y}}_2, \\ \underline{x}_N &= \hat{\underline{y}}_N, \end{aligned} \right\} \quad (5.4.12a)$$

$$\left. \begin{aligned} \underline{0} &= \hat{\underline{y}}_{N+1}, \\ \underline{0} &= \hat{\underline{y}}_{N+2}. \end{aligned} \right\} \quad (5.4.12b)$$

On the other hand, in order to proceed to further the analysis we introduce the matrices R_j (of size $m \times m$), $j=0,1,2,\dots,N$ to be defined by

$$\left. \begin{aligned} R_0 &= I \text{ (unit)}, R_1 = -\tilde{Q}_1, \\ \text{and } R_j &= (-\tilde{Q}_1)R_{j-1} + (-\tilde{Q}_2)R_{j-2}, j=2,3,\dots,N \end{aligned} \right\} \quad (5.4.13)$$

where \tilde{Q}_1 and \tilde{Q}_2 are given by (5.4.5).

Thus, from (5.4.11a) we can express each of the \hat{y}_j in terms of $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_j$, i.e.,

$$\left. \begin{aligned} \hat{y}_1 &= \tilde{y}_1, \\ \hat{y}_2 &= \tilde{y}_2 + R_1 \tilde{y}_1, \\ \hat{y}_3 &= \tilde{y}_3 + R_1 \tilde{y}_2 + R_2 \tilde{y}_1, \\ \hat{y}_4 &= \tilde{y}_4 + R_1 \tilde{y}_3 + R_2 \tilde{y}_2 + R_3 \tilde{y}_1, \\ &\dots \\ \hat{y}_N &= \tilde{y}_N + R_1 \tilde{y}_{N-1} + R_2 \tilde{y}_{N-2} + \dots + R_{N-1} \tilde{y}_1, \end{aligned} \right\} \quad (5.4.14a)$$

and from (5.4.14b) \hat{y}_{N+1} and \hat{y}_{N+2} can also be expressed in the form,

$$\left. \begin{aligned} \hat{y}_{N+1} &= \tilde{y}_{N+1} + R_1 \tilde{y}_N + R_2 \tilde{y}_{N-1} + \dots + R_N \tilde{y}_1, \\ \hat{y}_{N+2} &= \tilde{y}_{N+2} + \hat{R}_1 \tilde{y}_N + \hat{R}_2 \tilde{y}_{N-1} + \dots + \hat{R}_N \tilde{y}_1, \end{aligned} \right\} \quad (5.4.14b)$$

where R_j , $j=1,2,\dots,N$ are as given in (5.4.13), and the \hat{R}_j are defined by

$$\hat{R}_j = (-\tilde{Q}_2)R_{j-1}, j=1,2,\dots,N. \quad (5.4.15)$$

But $\hat{y}_{N+1} = \hat{y}_{N+2} = 0$ by virtue of (5.4.12b), thus by substituting for these values in (5.4.14b), with some rearrangement we obtain,

$$\left. \begin{aligned} -\tilde{y}_{N+1} &= R_1 \tilde{y}_N + R_2 \tilde{y}_{N-1} + \dots + R_N \tilde{y}_1, \\ -\tilde{y}_{N+2} &= \hat{R}_1 \tilde{y}_N + \hat{R}_2 \tilde{y}_{N-1} + \dots + \hat{R}_N \tilde{y}_1, \end{aligned} \right\} \quad (5.4.16)$$

where R_j and \hat{R}_j , $j=1,2,\dots,N$ are given in (5.4.13) and (5.4.15) respectively.

Moreover, we substitute for \tilde{y}_j as given in (4.5.93) in terms of y_j in both relations of (5.4.16), to yield the result,

$$\left. \begin{aligned} -Q_0^{-1} y_{N+1} &= R_1 Q_0^{-1} y_N + R_2 Q_0^{-1} y_{N-1} + \dots + R_N Q_0^{-1} y_1 \\ -Q_0^{-1} y_{N+2} &= \hat{R}_1 Q_0^{-1} y_N + \hat{R}_2 Q_0^{-1} y_{N-1} + \dots + \hat{R}_N Q_0^{-1} y_1 \end{aligned} \right\} \quad (5.4.17)$$

If we now premultiply both sides of each equation in (5.4.17) by Q_0 and introduce the matrices $T_j, \hat{T}_j, j=1,2,\dots,N$ such that

$$\left. \begin{aligned} T_j &= Q_0 R_j Q_0^{-1} \\ \hat{T}_j &= Q_0 \hat{R}_j Q_0^{-1} \end{aligned} \right\} j=1,2,\dots,N, \quad (5.4.18a)$$

where R_j and \hat{R}_j are as given earlier, or T_j may be computed recursively, i.e.,

$$\left. \begin{aligned} T_j &= (-Q_1 Q_0^{-1}) T_{j-1} + (-Q_2 Q_0^{-1}) T_{j-2}, \quad j=2,\dots,N \\ \text{with } T_0 &= I, \quad T_1 = -Q_1 Q_0^{-1} \\ \text{and } \hat{T}_j &\text{ from the relations,} \\ \hat{T}_j &= (-Q_2 Q_0^{-1}) T_{j-1} (= (-Q_0^{-1} Q_2) T_j), \quad j=1,2,\dots,N, \end{aligned} \right\} \quad (5.4.19)$$

then the two equations of (5.4.17) can be written in the form,

$$\left. \begin{aligned} -y_{N+1} &= T_1 y_N + T_2 y_{N-1} + \dots + T_N y_1 \\ -y_{N+2} &= \hat{T}_1 y_N + \hat{T}_2 y_{N-1} + \dots + \hat{T}_N y_1 \end{aligned} \right\} \quad (5.4.20)$$

Further, we premultiply the first equation by T_N , the second by T_{N-1} and so on up to the N^{th} equation by T_1 (or the k^{th} equation by T_{N-k+1} , $k=1,2,\dots,N$) in (5.4.9) and a similar multiplication takes place with \hat{T}_{N-k+1} , followed by substituting for the terms $T_{N-k+1} y_k$ and $\hat{T}_{N-k+1} y_k$, $k=1,2,\dots,N$ in the first and second relation of (5.4.20) respectively, then these two relations after some algebraic simplifications may be written in the form

$$\begin{aligned} -y_{N+1} &= \sum_{j=1}^N T_j \hat{z}_{N-j+1} - \left(\sum_{j=1}^N T_j F_{N-j+1} \right) y_{N+1} - \left(\sum_{j=1}^N T_j G_{N-j+1} \right) y_{N+2} \\ -y_{N+2} &= \sum_{j=1}^N \hat{T}_j \hat{z}_{N-j+1} - \left(\sum_{j=1}^N \hat{T}_j F_{N-j+1} \right) y_{N+1} - \left(\sum_{j=1}^N \hat{T}_j G_{N-j+1} \right) y_{N+2} \end{aligned}$$

Or in a simpler form, the latter equations can be written as

$$\left. \begin{aligned} (E_1 - I) y_{N+1} + s_1 y_{N+2} &= v_1 \\ E_2 y_{N+1} + (s_2 - I) y_{N+2} &= v_2 \end{aligned} \right\} \quad (5.4.21a)$$

where the matrices E_1, E_2, S_1, S_2 (each of size $m \times m$) and the vectors \underline{v}_1 and \underline{v}_2 (each of length m) are given by

$$\left. \begin{aligned} E_1 &= \sum_{j=1}^N T_j F_{N-j+1} & E_2 &= \sum_{j=1}^N \hat{T}_j F_{N-j+1} \\ S_1 &= \sum_{j=1}^N T_j G_{N-j+1} & S_2 &= \sum_{j=1}^N \hat{T}_j G_{N-j+1} \\ \underline{v}_1 &= \sum_{j=1}^N T_j \hat{z}_{N-j+1} \\ \text{and } \underline{v}_2 &= \sum_{j=1}^N \hat{T}_j \hat{z}_{N-j+1} \end{aligned} \right\} \quad (5.4.21b)$$

In matrix notation, the system (5.4.21a) takes the form

$$\begin{bmatrix} E_1 - I & S_1 \\ E_2 & S_2 - I \end{bmatrix} \begin{bmatrix} \underline{y}_{N+1} \\ \underline{y}_{N+2} \end{bmatrix} = \begin{bmatrix} \underline{v}_1 \\ \underline{v}_2 \end{bmatrix}, \quad (5.4.22)$$

which is of order $2m$ and can be solved for \underline{y}_{N+1} and \underline{y}_{N+2} provided that the coefficient matrix is non-singular.

However, having determined \underline{y}_{N+1} and \underline{y}_{N+2} from (4.5.106), the \underline{y}_j , $j=1,2,\dots,N$ can be determined from (5.4.9), followed by using (5.4.7) to determine $\tilde{\underline{y}}_j$, $j=1,2,\dots,N$. Finally, the $\hat{\underline{y}}_j$, $j=1,2,\dots,N$ can be obtained from (5.4.11a) which in fact are equivalent to the solution $\underline{x}_1, \dots, \underline{x}_N$ by virtue of (5.4.12a).

The summary of the above procedure can be abbreviated in the following steps:

Step 1 and Step 2 (see the corresponding steps in Section 5.3).

Step 3 Compute F_j and G_j , $j=1,2,\dots,N$ from (5.4.2) (noting that the terms H and K must be ignored).

Step 4 Compute $\tilde{\underline{z}}_j$, $\hat{\underline{z}}_j$, $j=1,2,\dots,N$ as follows,

$$Q_0 \tilde{\underline{z}}_j = \underline{z}_j$$

and $\hat{\underline{z}}_j$ from (5.3.12).

- Step 5 Obtain T_j and \hat{T}_j , $j=1,2,\dots,N$ from (5.4.19).
- Step 6 Solve the linear system (5.4.22) whose elements can be computed from (5.4.21b) and hence $y_{N+1} = y_{N+2}$ can be determined.
- Step 7 Obtain y_j , \tilde{y}_j and \hat{y}_j (or x_j), $j=1,2,\dots,N$ as follows,
- y_j from (5.4.9),
- \tilde{y}_j from (5.4.7),
- \hat{y}_j from (5.4.11a),
- and $x_j = \hat{y}_j$ by virtue of (5.4.12a).

The number of operations involved in the above procedure (excluding Step 1) is of order $O((\frac{40}{3}m^3 + 8m^2)N)$. As pointed out in the previous algorithm, if \tilde{z}_j (in Step 4) and \tilde{y}_j (in Step 7) are evaluated such that the coefficient matrix (i.e. Q_j) is factorised once, the order of the operation may be reduced to $O(13m^3 + 10m^2)N$.

If the submatrices B, C and D of the given system (5.4.1) satisfy the appropriate conditions incicated in Section 5.3 (in particular, the conditions (5.4.7), then it may be possible to adopt the spectral resolution method to the system (5.4.1). In this case the above procedure can be converted analogous to the ones discussed in the previous subsections (see Section 5.2), for example Step 1 would involve the computation of the eigenvalues of Q_0, Q_1 and Q_2 which in fact are given by (5.3.20).

Finally, we briefly outline the case where the linear system (5.4.1) possessed a slightly different coefficient matrix such that the first and last diagonal submatrices are different, i.e.,

$$\begin{bmatrix} B_1 & C & D \\ C & B & C & D \\ D & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} \quad (5.4.23)$$

In fact, the system (5.4.25b) is treated exactly as (5.3.4) whilst the four sub-matrices β, δ, γ and ϵ may be determined as follows:

either by choosing $\beta=Q_0$, $\gamma=Q_2$ and then from the relations (5.4.25a) and (5.4.25c) we have,

$$Q_0 \delta = B_1 - (Q_1^2 + Q_2^2) \quad (5.4.26)$$

$$\text{and} \quad Q_2 \epsilon = B_2 - (Q_0^2 + Q_1^2) ,$$

which have to be solved for δ and ϵ respectively, provided that Q_0 and Q_2 are non-singular matrices,

or by setting $\beta=\delta$, $\gamma=\epsilon$ and then from (5.4.25a) and (5.4.25c) we have

$$\left. \begin{aligned} \beta^{\frac{1}{2}} &= \delta^{\frac{1}{2}} = (B_1 - Q_1^2 - Q_2^2)^{\frac{1}{2}} \\ \gamma^{\frac{1}{2}} &= \epsilon^{\frac{1}{2}} = (B_2 - Q_0^2 - Q_1^2)^{\frac{1}{2}} \end{aligned} \right\} \quad (5.4.27)$$

provided that the square roots of the appropriate matrices are defined.

Thus, the factorization procedure for the system (5.4.23) involves 5 matrix square roots if the scheme (5.4.27) is considered.

The elimination process now can continue such that the coefficient matrices of the systems (5.4.3a) and (5.4.3b) must be replaced by the relevant ones from (5.4.24). Then, the remaining steps of the elimination process are carried out as before provided that the submatrices β, δ, γ and ϵ must be taken into account which in fact does not require major modification. Note that the FIRM3 algorithm has been programmed on the basis of system (5.4.23) rather than (5.4.1) for more general application purposes.

5.5 ALGORITHM FICM6 AND ALGORITHM FIRM5

The two algorithms in Section 4.5 and 4.6 may be extended to the block case and for the special case when the coefficient matrix is constant and skew-symmetric, tridiagonal (periodic or non-periodic respectively). In fact, the block systems under consideration are of the form,

$$\begin{bmatrix} B & C & & & -C \\ -C & & 0 & & \\ & 0 & & C & \\ & & & -C & B \\ C & & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}, \quad (5.5.1a)$$

or $\underline{Ax} = \underline{z}$ (5.5.1b)

and

$$\begin{bmatrix} B & C & & & \\ -C & & 0 & & \\ & 0 & & C & \\ & & & -C & B \\ & & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}, \quad (5.5.2a)$$

or $\tilde{\underline{A}}\underline{x} = \underline{z}$, (5.5.2b)

where the submatrices and subvectors in both systems are each of size $(m \times m)$ and $(m \times 1)$ respectively (see Section 5.1 or 5.2),

The first algorithm (i.e. FICM6) deals with the periodic case which is represented by the system (5.5.1), whilst the other algorithm (i.e. FIRM5) deals with the non-periodic case which is represented by the system (5.5.2).

The modification strategy of the given system adopted in both algorithm FICM5 and FIRM4 (Sections 4.5 and 4.6) can be applied to the systems (5.5.1) and (5.5.2) respectively, that is we premultiply both

sides of each system by the transpose of its coefficient matrix to yield the result,

$$(i) \text{ for (5.5.1), we have } A^T A \underline{x} = A^T \underline{z}, \quad (5.5.3a)$$

$$\text{or } G \underline{x} = \underline{v}, \quad (5.5.3b)$$

$$\text{where } G = A^T A, \quad (5.5.3c)$$

$$\text{and } \underline{v} = A^T \underline{z} \equiv [\underline{v}_1, \underline{v}_2, \dots, \underline{v}_N]^T, \quad (5.5.3d)$$

and

$$(ii) \text{ for (5.5.2), we have } \tilde{A}^T \tilde{A} \underline{x} = \tilde{A}^T \underline{z}, \quad (5.5.4a)$$

$$\text{or } H \underline{x} = \underline{u}, \quad (5.5.4b)$$

$$\text{where } H = \tilde{A}^T \tilde{A}, \quad (5.5.4c)$$

$$\text{and } \underline{u} = \tilde{A}^T \underline{z} \equiv [\underline{u}_1, \underline{u}_2, \dots, \underline{u}_N]^T \quad (5.5.4d)$$

Moreover, it can be easily verified that G in (5.5.3c) and H in (5.5.4c) are *symmetric* block matrices, quindagonal periodic and non-periodic respectively (c.f. (5.3.1) and (5.4.1)), then the (5.5.3b) and (5.5.4b) may be written explicitly in the form respectively,

$$\begin{bmatrix} B^2+2C^2 & 0 & -C^2 & & -C^2 & 0 \\ 0 & B^2+2C^2 & 0 & -C^2 & & 0 \\ -C^2 & 0 & B^2+2C^2 & 0 & -C^2 & \\ & -C^2 & 0 & B^2+2C^2 & 0 & -C^2 \\ -C^2 & 0 & & -C^2 & 0 & B^2+2C^2 \\ 0 & -C^2 & & & -C^2 & 0 \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_N \end{bmatrix} = \begin{bmatrix} \underline{v}_1 \\ \underline{v}_2 \\ \vdots \\ \underline{v}_N \end{bmatrix}, \quad (5.5.5)$$

and

$$\begin{bmatrix} B^2+C^2 & 0 & -C^2 & & & \\ 0 & B^2+2C^2 & 0 & -C^2 & & \\ -C^2 & 0 & B^2+2C^2 & 0 & -C^2 & \\ & -C^2 & 0 & B^2+2C^2 & 0 & -C^2 \\ & & 0 & -C^2 & 0 & B^2+2C^2 \\ & & & -C^2 & 0 & B^2+C^2 \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_N \end{bmatrix} = \begin{bmatrix} \underline{u}_1 \\ \underline{u}_2 \\ \vdots \\ \underline{u}_N \end{bmatrix}. \quad (5.5.6)$$

$$\begin{bmatrix}
 B^2+C^2 & -C^2 & & & & \\
 -C^2 & B^2+2C^2 & -C^2 & & & \\
 & \ddots & \ddots & \ddots & & \\
 & & 0 & & & \\
 & & & \ddots & \ddots & \\
 & & & & -C^2 & B^2+2C^2 & -C^2 \\
 & & & & -C^2 & B^2+C^2 &
 \end{bmatrix}
 \begin{bmatrix}
 x_2 \\
 x_4 \\
 x_6 \\
 \vdots \\
 x_N
 \end{bmatrix}
 =
 \begin{bmatrix}
 u_2 \\
 u_4 \\
 u_6 \\
 \vdots \\
 u_N
 \end{bmatrix}
 \quad (5.5.10b)$$

where the four systems are of order $mN/2$ each (whilst for N odd the systems (5.5.9a) and (5.5.10a) are of order $m(N+1)/2$ each and the systems (5.5.9b) and (5.5.10b) are of order $mN=m(N-1)/2$ each).

In fact, the two systems in (5.5.9) which possess the same coefficient matrix can be solved by the procedure of algorithm FICM3 (Section 5.1). In this case the submatrices of the factor matrices in (5.1.3) would take the form (c.f. (5.1.8)),

$$\text{and } \left. \begin{aligned} Q_0 &= 0.5[B+(B^2-4C^2)^{\frac{1}{2}}] \\ Q_1 &= 0.5[B-(B^2-4C^2)^{\frac{1}{2}}] \end{aligned} \right\} \text{ provided that } ||B|| > 2||C||. \quad (5.5.11)$$

Similarly for the systems in (5.5.10) which are solved by adopting the procedure of FIRM2 (Section 5.2), except that the factorisation of the coefficient here is slightly different, i.e. from (5.5.10a) (or (5.5.10b)) we may factorise the matrix as follows,

$$\begin{bmatrix}
 B^2+C^2 & -C^2 & & & & \\
 -C^2 & B^2+2C^2 & -C^2 & & & \\
 & \ddots & \ddots & \ddots & & \\
 & & 0 & & & \\
 & & & \ddots & \ddots & \\
 & & & & -C^2 & B^2+2C^2 & -C^2 \\
 & & & & -C^2 & B^2+C^2 &
 \end{bmatrix}_{mM \times mM}
 =
 \begin{bmatrix}
 P & Q_1 & & & & \\
 & Q_0 & Q_1 & & & \\
 & & \ddots & \ddots & & \\
 & & & 0 & & \\
 & & & & \ddots & \\
 & & & & & Q_1 & Q_0 \\
 & & & & & & R
 \end{bmatrix}_{m'f \times m(M+1)}$$

$$\left[\begin{array}{ccc} P & & \\ Q_1 & Q_0 & \\ & Q_1 & Q_0 \\ & & 0 \end{array} \right]_{(M+1)m \times mM} \equiv \left[\begin{array}{ccc} P^2 + Q_1^2 & Q_1 Q_0 & \\ Q_0 Q_1 & Q_0^2 + Q_1^2 & Q_1 Q_0 \\ & Q_0 Q_1 & Q_0^2 + Q_1^2 \\ & & Q_1 Q_0 & 0 \end{array} \right]_{mM \times mM} \quad (5.5.12)$$

CHAPTER 6

APPLICATIONS TO ORDINARY DIFFERENTIAL EQUATIONS

6.1 INTRODUCTION

The application of the algorithms presented in Chapter 4 which are proposed for the point form, will be discussed here, and we may refer to these algorithms as BANDSOLVERS. As the factorisation processes involved in these algorithms yield a set of non-linear equations, the criteria for convergence has to be studied from the practical application point of view. In this respect we shall denote the maximum tolerance at which the appropriate iterative process is terminated by ϵ_f (this applies to step 4' in subsection 4.3.3 and 4.4.3). In addition, some numerical examples for both periodic and non-periodic 2-point boundary value problems are considered which may reflect to a certain extent how far the application of the new algorithms can be considered to be worthwhile.

6.2 ON THE IMPLEMENTATION OF THE PROCEDURE GITRM (Subsection 4.2.2)

The GITRM procedure has been presented as two schemes represented by the systems (4.2.24) (or (4.2.26)) and (4.2.25) (or (4.2.27)). These two schemes were tested for values of $r=1,2,\dots,20$, i.e. thus yielding matrices of bandwidth $3,5,7,\dots,41$, with matrix elements c_0, c_1, \dots, c_r , as defined in the system (4.2.1) (given by the expression of even powers of the central operator δ). For $c_k = (-1)^{k+2} \binom{2r}{k}$, $k=r-1, \dots, 0$, $r \geq 1$ we choose the diagonal element c_0 as,

$$c_0 = k \left(2 \sum_{i=1}^r c_i \right), \quad (k > 1), \quad (6.2.1)$$

(for r odd, the elements c_k , $k=0,1,\dots,r$, were multiplied by -1).

From the results it was found that for scheme (4.2.24) the maximum modulus error of the $\alpha_0, \alpha_1, \dots, \alpha_r$ (the elements of the matrix Q in (4.2.3b)), between two consecutive iterations, is improved in general by one decimal for every 2 iterations. Whilst for scheme (4.2.25) apart from the first few iterations, one iteration is sufficient to give one further decimal place each time.

Moreover, for small r ($r > 1$) the number of iterations using (4.2.25) does not exceed a half of those required by the scheme (4.2.24) and this is supported theoretically by the result given in the relation (4.2.59).

It was noticed also by choosing c_0 in (6.2.1) with small k the convergence becomes slower, in other words the closer k is taken to 1 the rate of convergence (if any) decreases. For example for $r=2$ and $r=3$ with $k=7/6$ and $21/20$ respectively (which is the case for the matrices (4.2.9a) and (4.2.9b)) converge in 19 and 39 iterations respectively with a maximum error of $O(10^{-11})$.

Other tests were made on the cases where the magnitude of the ratios between c_1 and c_0 appears to exceed $\cos(\pi/(r+2))$ where no convergence

occurred. The failure of the convergence (to a real solution) in such cases is supported by the conclusions suggested by Berg (1981). As a result, the conditions given in Appendix A may be considered as *necessary* conditions for the convergence of the iterative procedure GITRM.

Finally, an interesting application also occurs in the field of digital communication. To optimise the number of levels to be used in AM (Amplitude Modulation) for the most noise-tolerant operation, a particular detection method has to be selected when dealing with the samples of the received waveform and a digital filter is used to perform the required correlation (Passas (1979)).

The output of the filter is related to the input signal via the following matrix system involving the non-linear equations:

$$(a_0, a_1, \dots, a_r) \begin{bmatrix} a_0 & & & & \\ & a_1 & & & \\ & & a_2 & & \\ & & & \ddots & \\ & & & & a_r \end{bmatrix} = (\phi_0, \phi_1, \dots, \phi_r), \quad (6.2.2)$$

where $\underline{\phi}^T$ refers to the correlation sequence.

and \underline{a}^T are the sample values of the filter input.

The equations (6.2.2) can be rewritten compactly in the form,

$$\underline{a}^T A = \underline{\phi}^T. \quad (6.2.3)$$

Then, taking the transpose of (6.2.3), we obtain the equations

$$A^T \underline{a} = \underline{\phi}, \quad (6.2.4)$$

which has to be solved for \underline{a} , the filter characteristics. In fact,

(6.2.4) is exactly similar to the system (4.2.4).

6.3 NON-LINEAR EQUATIONS INVOLVED IN FICM2 (AND FIRMI)

The procedure of solving the non-linear equations (4.3.4) (and (4.4.5)) iteratively which has been discussed in subsection 4.3.3 (and subsection 4.4.3) was tested on various types of banded and block matrices (the latter type applies to FIRMI only, see Chapter 7). The vital point in this respect is to decide whether the new factorisation strategy for the given matrix exists. This in fact is associated with the occurrence of the convergence of the iterative process^{used here}. Many examples have been studied, a few are presented below.

Since the following examples consist of symmetric matrices with constant elements it may be convenient to refer to the periodic matrix A of (4.3.1) and its factor matrices L and U in (4.3.2) in the abbreviated forms:-

$$\begin{aligned} A_r(0, a_r, \dots, a_1, a_0, a_1, \dots, a_r, 0) & \quad (\text{of bandwidth } 2r+1), & (6.3.1) \\ L_r(0, \ell_r, \dots, \ell_1, 1, 0) & \quad (\text{of semibandwidth } r+1), \\ \text{and } U_r(0, u_1, \dots, u_{r+1}, 0) & \quad (\text{of semibandwidth } r+1) \end{aligned}$$

respectively.

Example 6.3.1: For $r=2$, A_2 is taken as,

$$A_2(0, 1, -16, k, -16, 1, 0), \text{ with } k=45, 35, 34, 33, 32, 31$$

which leads to L_2 and U_2 to be given as,

$$L_2(0, \ell_2, \ell_1, 1, 0), \quad U_2(0, u_1, u_2, u_3, 0),$$

where ℓ_1, ℓ_2 and u_1 are given in Table 6.3.1 and $u_2 = \ell_1 u_1$, $u_3 = 1$ (see subsection 4.3.4).

k	ℓ_1	ℓ_2	u_1
45	$-4.0316417989 \times 10^{-1}$	$2.5849100710 \times 10^{-2}$	3.8686065377×10
35	$-5.9816996880 \times 10^{-1}$	$3.8837590572 \times 10^{-2}$	2.5748750220×10
34	$-6.3479727020 \times 10^{-1}$	$4.1313953443 \times 10^{-2}$	2.4204897298×10
33	$-6.7956808383 \times 10^{-1}$	$4.4356979461 \times 10^{-2}$	2.2544366460×10
32	$-7.3745480935 \times 10^{-1}$	$4.8317944364 \times 10^{-2}$	2.0696244702×10
31	$-8.2148142904 \times 10^{-1}$	$5.4121317910 \times 10^{-2}$	1.8477007556×10

TABLE 6.3.1

The number of iterations to achieve convergence yields a maximum error, ϵ_f , of the given order are tabulated in the following table.

k	45	35	34	33	32	31
No. of iterations	15	24	27	29	36	51
$O(\epsilon_f)$	10^{-10}	10^{-10}	10^{-9}	10^{-9}	10^{-9}	10^{-9}

TABLE 6.3.2

Example 6.3.2: For $r=3$, A_3 is taken,

$$A_3(0, -2, 27, -300, k, -300, 27, -2, 0), \text{ with } k=1200, 900, 600, 570, 560, 551,$$

which leads L_3 and U_3 to be

$$L_3(0, \ell_3, \ell_2, \ell_1, 1, 0), \quad U_3(0, u_1, u_2, u_3, u_4, 0),$$

where ℓ_1, ℓ_2, ℓ_3 and u_1 are given in Table 6.3.3 and $u_2 = \ell_1 u_1$, $u_3 = \ell_2 u_1$ and $u_4 = -2$.

k	ℓ_1	ℓ_2	ℓ_3	u_1
1200	$-2.6097047668 \times 10^{-1}$	$2.3580141069 \times 10^{-2}$	$-1.7811079753 \times 10^{-3}$	1.1228965497×10^3
900	$-3.6618180301 \times 10^{-1}$	$3.3132047171 \times 10^{-2}$	$-2.5226515759 \times 10^{-3}$	7.9281658200×10^2
600	$-7.0529687886 \times 10^{-1}$	$6.4040471205 \times 10^{-2}$	$-5.0052330722 \times 10^{-3}$	3.9958179193×10^2
570	$-8.2579955744 \times 10^{-1}$	$7.5048859332 \times 10^{-2}$	$-5.9213829335 \times 10^{-3}$	3.3775893618×10^2
560	$-8.9525011488 \times 10^{-1}$	$8.1396970187 \times 10^{-2}$	$-6.4576426521 \times 10^{-3}$	3.0971054221×10^2
551	-1.0246225541	$9.3218778579 \times 10^{-2}$	$-7.4722213562 \times 10^{-3}$	2.6765802395×10^2

TABLE 6.3.3

The number of iterations where convergence occurs and yields an error ϵ_f of the given order are given in Table 6.3.4.

k	1200	900	600	570	560	551
No. of iterations	11	13	28	46	61	68 104
$O(\epsilon_f)$	10^{-9}	10^{-10}	10^{-9}	10^{-10}	10^{-9}	10^{-4} 10^{-6}

TABLE 6.3.4

Example 6.3.3: For $r=4$, A_4 is taken as

$A_4(0,1,-8,28,-56,k,-56,28,-8,1,0)$, with $k=140,75,74,73,72,71$, which for $k=140$ implies that the factor matrices L_4 and U_4 are

$$L_4(0,\ell_4,\ell_3,\ell_2,\ell_1,1,0), \quad U_4(0,u_1,u_2,u_3,u_4,u_5,0),$$

where $\ell_1=-3.8267015251 \times 10^{-1}$, $\ell_2=2.1246247731 \times 10^{-1}$,

$$\ell_3=-6.5067421195 \times 10^{-2}, \ell_4=8.5420248957 \times 10^{-3}, u_1=1.1706826100 \times 10^2,$$

$$u_2=\ell_1 u_1, \quad u_3=\ell_2 u_1, \quad u_4=\ell_3 u_1 \quad \text{and} \quad u_5=1.$$

The number of iterations with the corresponding $O(\epsilon_f)$ involved in the evaluation of the elements of L_4 and U_4 are tabulated in Table 6.3.5.

k	140	75	74	73	72	71
No. of iterations	20	37	29	26	29	no.convg.
$O(\epsilon_f)$	10^{-9}	10^{-7*}	10^{-5*}	10^{-4*}	10^{-3*}	-

(*no improvement was obtainable in further iterations)

TABLE 6.3.5

In Example 6.3.1 it can be noticed that when k decreases the modulus of ℓ_1, ℓ_2 increases whilst u_1 decreases (see Table 6.3.1). Similar remarks apply to Example 6.3.2 (Table 6.3.3). Further, for $k=551$ the modulus of ℓ_1 gets greater than the diagonal elements of L_3 which are unity. One of the consequences of the latter case may imply no guarantee for the

stability of elimination process involved in the solution procedure.

Again, this case arises in Example 6.3.3 except for $k=140$.

On the other hand, the largest value of k apart from the first few iterations one step (as in Table 6.3.2 and 6.3.4) or two steps (as in Table 6.3.5) are sufficient to yield a one decimal place improvement. Also, it is observed that the rate of convergence decreases as k does, for example with $k=551$ (Table 6.3.4) to attain an ϵ_f of order 10^{-4} and 10^{-6} requires 68 and 104 iterations respectively, or even divergence may occur as in Table 6.3.5 (with $k=71$).

Thus, in general by having a small difference between the diagonal element a_0 and the summation of the off-diagonal elements of A_r in (6.3.1) may imply a 'poor' (or inaccurate) factorisation or non-existent (i.e. the case where no convergence is attainable); otherwise, when $(a_0 - 2 \sum_{i=1}^r a_i)$ (c.f. (6.2.1)) is rather large the factorisation of A_r is possible; bearing in mind that it is *not* necessary for A_r to be diagonally dominant (although it is a convenient case). This remark may be generalised to the non-constant case, i.e. when the matrix A is as defined in (4.3.1) and we conclude that its factorisation may occur in the form defined as given in (4.3.2) if each of its rows possesses a *diagonal element* ($a_{0,i}$ say) greater than *the summation of the off-diagonal elements* (s_i say); this also may depend upon the bandwidth of the matrix (i.e. the size of r), for example $(a_{0,i} - s_i)$ being 1 for the quindagonal case ($r=2$) may be a reasonable limit (see for example Table 6.3.1, $k=31$) whilst for the case $r=4$ may not be sufficient (see for example Table 6.3.5, $k=71$), and with $r=3$ it might be sufficient but a 'poor' factorisation may be expected (see for example Table 6.3.4, $k=551$).

The above discussion may be extended to the case where A_2, A_3 and A_4 in the previous examples are non-periodic matrices, provided that the

diagonal elements are large enough to satisfy the condition indicated above. Whilst for the case where the diagonal elements are at 'critical' values (i.e. the cases where slow (or no) convergence may occur in the periodic case) the results show a better and more satisfactory convergence rate for the non-periodic case. For example, Table 6.3.6 presents the results of A_4 (in Example 6.3.3) as being non-periodic, and by comparing them to the corresponding periodic case in Table 6.3.5 it is clear that for all values of k the convergence exists with maximum attainable order of error (and faster), though the value of $|\lambda_1|$ exceeds 1 here as well with all values of k , excluding $k=140$. This is related to the property that the spectral radius of the iteration matrix of the factorisation process may approach and exceed unity for various values of k . This is clarified further by considering the quindagonal matrices $A_2(0,1,-4,k,-4,1)$ and $\tilde{A}_2(0,1,-4,k,-4,1,0)$ which refer to periodic and non-periodic forms, respectively. While convergence was possible to yield an ϵ_f of $O(10^{-4})$ for A_2 with $k=6.2$ an ϵ_f of $O(10^{-9})$ for \tilde{A}_2 with $k=6.05$ and an ϵ_f of $O(10^{-6})$ for $k=6.01$. Whilst in A_2 with the last two values of k , the convergence was not obtainable and it is clear that A_2 is 'close' to becoming singular. This fact is supported by increasing the size of the matrix where slower convergence would be expected.

k	140	75	74	73	72	71
No. of iterations	16	41	45	50	54	75
$O(\epsilon_f)$	10^{-10}	10^{-10}	10^{-10}	10^{-10}	10^{-10}	10^{-10}

TABLE 6.3.6 (A_4 non-periodic)

The above discussion is not suggested only for the particular aforementioned examples, but also for the matrices involved in the numerical examples included in the following section.

6.4 NUMERICAL EXAMPLES OF 2-POINT BOUNDARY VALUE PROBLEMS

The examples considered are for the 2-point boundary value problem concerning the second-order ordinary differential equation (linear and non-linear) involving periodic and non-periodic conditions (details of the definition of the problem are given in Chapter 3).

Example 6.4.1

$$y'' - 4\pi^2 y = \sin 2\pi x (1 + \sin 2\pi x) e^{\sin 2\pi x},$$

$$y(0) = y(1), y'(0) = y'(1),$$

which has the true solution: $y = e^{\sin 2\pi x}.$

Example 6.4.2

$$y'' - 4y = -4\sin 2x - 5\sin x,$$

$$y(a) = y(b), y'(a) = y'(b), a = -\pi/2, b = 3\pi/2,$$

with the true solution: $y = 0.5(\sin 2x + \sin x).$

Example 6.4.3

$$y'' - y - y^3 - e^{\sin 2\pi x} [4\pi^2 (\cos^2 2\pi x - \sin 2\pi x) - e^{2\sin 2\pi x} - 1] = 0,$$

$$y(0) = y(1), y'(0) = y'(1),$$

with the true solution: $y = e^{\sin 2\pi x}.$

(Pereyra (1973), Daniel and Martin (1977)).

Example 6.4.4

$$y'' - (1 - y^2)y' - 4y = -5\sin x - \cos^3 x,$$

$$y(0) = y(2\pi), y'(0) = y'(2\pi),$$

with the true solution: $y = \sin x.$

(Pereyra (1968)).

Example 6.4.5

$$y'' = y^3 - \sin x (1 + \sin^2 x),$$

$$y(0) = y(2\pi) , y'(0) = y'(2\pi) ,$$

with the true solution: $y = \sin x$.

(Pereyra (1968),(1973)).

Example 6.4.6

$$y'' - \cos\left(\frac{3\pi}{2}x\right)y' - y - (\gamma^2 - \gamma \cos\left(\frac{3\pi}{2}x\right) - 1)e^{\gamma x} = 0 ,$$

$$y(0) = 1, y(1) = 0.1,$$

with the true solution: $y = e^{\gamma x}$, $\gamma = \ln 0.1$.

(Shoosmith in Aziz (1975)).

Example 6.4.7

$$y'' - \frac{1}{2}(y+x+1)^3 = 0 ,$$

$$y(0) = y(1) = 0 ,$$

with the true solution: $y = \frac{2}{2-x} - x - 1$.

(Pereyra (1973)).

Prior to the discussion we refer to the notation list below in Table 6.4.0 which are related to the appropriate tables of this section, and may be used in the following discussion.

N=number of mesh points

BNDM=bandwidth of the matrix being used
($=2r+1$, $r \geq 1$)

* refers to the terms which are of order 10^{-10} or less

$$||\epsilon|| = \frac{1}{N+1} \left[\sum_{i=0}^N |y_i^{(s)} - y_i^{(s-1)}|^2 \right]^{\frac{1}{2}}, \text{ the Euclidean norm}$$

TABLE 6.4.0: Notations used in Tables 6.4.1-6.4.7

The aforementioned examples were solved by the iterative deferred correction procedure discussed in Chapter 3, using both low and high-order discretisation (LOD and HOD) schemes (Sections 3.3 and 3.4 respectively).

The linear systems involved in this procedure are represented by the matrix equation (3.4.21) whose coefficient matrix (i.e. the Jacobian) is referred to as the *non-periodic* case whilst the Jacobian for the *periodic* case is defined in (3.4.29). Subsequently, these systems were treated by the appropriate algorithm according to the related case, i.e., the algorithm FICM1 was used in Examples 6.4.1 and 6.4.2, FICM2 in Examples 6.4.3-6.4.5 and FIRMI in Examples 6.4.6 and 6.4.7.

From the programming and computational work viewpoint, the periodic case (Examples 6.4.1-6.4.5) seem to be easier and more economical than the non-periodic case (Examples 6.4.6 and 6.4.7). This may be due to:

- i) algorithm FIRMI requiring more operations and larger storage space than FICM1 or FICM2, and
- ii) the periodic problem (which in our case the solution is required over the whole range)^(*), does not involve any extrapolation procedure or difficulties at the grid points near the boundaries as in the non-periodic case.

What is meant by the difficulties is mainly when the HOD scheme is adopted, the high-order finite difference formulae cannot be applied at the grid points near the boundaries. Thus the use of suitable *non-symmetric* formulae (see Section 3.4) is required, otherwise the aim of implementing HOD may not be achieved. As the search for such formulae seem to be a difficult task, it was imprudent to proceed to consider the non-periodic Examples 6.4.6 and 6.4.7 with the HOD scheme. In addition, the extrapolation procedure raised another difficulty with employing HOD since

^(*) It is possible to work on half of the integrated range as in Example 6.4.5, which reduces the amount of computational work by one half. (Pereyra(1968)). This may not be the case if high-order discretisation is used!

extrapolation by the differential equation (3.4.16) was impractical and using the alternative scheme represented by (4.3.15), inspite of its practicality the question of choosing an 'optimal' order was faced in order to achieve a better tolerance, in particular when the value of BNDM is greater than 5. However, with BNDM=5 by keeping the order of extrapolation unchanged in the successive solutions seems to be a reasonable approach (Audish, (1978)).

The above stressed points, i.e. the extrapolation and forming a 'suitable' Jacobian (near the boundaries) have their consequences on the convergence of Newton's Method and on solving (3.4.21) by the FIRML algorithm. For example, in Example 6.4.7 (Table 6.4.7) for the three cases $N=10, 20$ and 40 although the first solution, $y^{(1)}$, of BNDM=5 is an improvement to BNDM=3, however for the successive solutions ($y^{(2)}, y^{(3)}, y^{(4)}$) it is not so. On the other hand, the factorisation involved in FIRML was not accurate enough; for instance in BNDM=3 with $N=20, 40$ the obtained ϵ_f after 61 iterations was of $O(10^{-4})$ and $O(10^{-3})$ respectively, and it changes to $O(10^{-3})$ and $O(10^{-2})$ respectively for BNDM=5. The effect of an inaccurate (or poor) factorisation by FIRML on Newton's convergence can be clearly noticed in the *linear*-boundary value problem (Example 6.4.6, Table 6.4.7, see the brackted numbers). For $N=20$, the factorisation was achieved up to ϵ_f of $O(10^{-4})$, whilst this order comes down to $O(10^{-1})$ with $N=40$, and the method fails with BNDM=7. In the light of the above non-periodic example it seems that unsatisfactory results were obtained, but it would be inadequate to conclude that the HOD scheme (and FIRML implicitly) is completely unbeneficial unless other factors (as indicated earlier) involved are alleviated or overcome. Whilst in the periodic case, the HOD scheme (and the related algorithms FICM1 and FICM2) showed quite reasonable results.

On applying algorithm FICM2 to the periodic non-linear Examples 6.4.3 6.4.5 we may draw the following conclusions.

One of the best of the three examples that the factorisation procedure involved in FICM2 worked perfectly is Example 6.4.4, whose results are shown in Table 6.4.4. From the theoretical point of view, for a given r , the successive solutions with deferred corrections, i.e. $\underline{y}^{(2)}, \underline{y}^{(3)}, \dots, \underline{y}^{(k)}$, $k \geq 2$, should coincide (or at least possess error tolerances of the same order) with the first solution $\underline{y}^{(1)}$ (i.e. no deferred correction) of the HOD scheme for values of BNDM equal to $2(r+1)+1, 2(r+2)+1, \dots, 2(r+k-1)+1$, respectively. This is justified in Table 6.4.4; for example with $r=1$, $N=20, 40$ or 80 the solutions in the first row and first column have the same order of accuracy $||\epsilon||$. Also there is no significant difference between the corresponding ratios in the following two types, (i) the ratios of in $\underline{y}^{(k)}$, $k \geq 2$, for $N=20$ and 40 (or $N=40$ and 80) for $\text{BNDM}=3$ and (ii) the ratios of $||\epsilon||$ in $\underline{y}^{(1)}$ for the same BNDM in $N=20$ and 40 (or $N=40$ and 80). Precisely, from Table 6.4.4, with $\text{BNDM}=3$ the ratios of $||\epsilon||$ in $\underline{y}^{(1)}, \underline{y}^{(2)}, \underline{y}^{(3)}$ with $N=20$ to the corresponding ones with $N=40$ are respectively $4.01, 15.86$ and $62.92^{(*)}$, whilst the ratios of $||\epsilon||$ in $\underline{y}^{(1)}$ (with $\text{BANDM}=j$, $j=5, 7$) for $N=20$ and $N=40$ are respectively (according to j) 15.99 and 63.76 .

It was noticed (for $N=20, 40, 80$) that Newton's method takes 4 iterations for the first solution, $\underline{y}^{(1)}$ for all cases of HOD (i.e. $\text{BNDM}=3, 5, 7, 9, 11$) whilst the successive solutions takes 1 or 2 iterations. Now, in this respect the question may arise as to what is gained by using the HOD scheme? For this particular example (in which the *first* derivative is inclusive) it can be observed from Table 6.4.4 that with $N=40$, $\text{BANDM}=3$, for instance, accuracy $||\epsilon||$ of $O(10^{-10})$ is achieved in the 4th correction in which it

^(*) According to the theory, the ratios must be $4, 16, 64$ (Pereyra (1968)).

involves a total of 11 Newton's iterations *plus* the implementation of the iterative deferred correction procedure, whilst the same accuracy is obtained in the first solution with $\text{BNDM}=9$ and involves 4 Newtons iterations only, without involving the deferred correction procedure. Similarly with $N=20$, $\text{BNDM}=3$, where the 5th correction produced an accuracy $||\epsilon||$ of $O(10^{-9})$, whilst for $\text{BNDM}=11$ leads to an accuracy $||\epsilon||$ of $O(10^{-10})$ in the first solution. This remark is, generally, applicable to the other non-linear periodic Examples 6.4.3 and 6.4.5 whose results are tabulated in Table 6.4.3 and 6.4.5 respectively, although the factorisation of FICM2 was not as accurate as in Example 6.4.4.

Also, the application of the HOD scheme to the linear periodic Examples 6.4.1 and 6.4.2 using algorithm FICM1 led to satisfactory results. It should be mentioned that in these two examples the procedures of higher order matrix polynomials represented by the relation (3.4.30) was applied up to $\hat{\Omega}^{10}$ (i.e. up to $\text{BNDM}=11$). The results of both examples are listed in Table 6.4.1 and 6.4.2 respectively, from which it verifies the practicability of HOD and FICM1 as well. For the latter, in the iterative procedure GITRM (see subsection 4.2.2) the relevant convergence occurred in 10 iterations for $\text{BNDM}=5$, 14 for $\text{BNDM}=11$ ($\text{BNDM}=3$ is excluded since (2.2.27) scheme was used), both with $N=20$, whilst with $N=40$ or 80 just one or two extra iterations for the corresponding BNDM were required. Besides that the ratio of α_1/α_0 (the elements of matrix (4.2.3b)) was less than 1 in modulus, except in Example 6.4.1 with $N=80$ and $\text{BNDM} \geq 7$, for which it turned out to be slightly greater than 1. Whilst the Newton's method (as expected) required 1 or 2 iterations in all cases.

N	BNDM	ϵ = Norm of Error in the Successive Solutions				
		$\underline{y}^{(1)}$	$\underline{y}^{(2)}$	$\underline{y}^{(3)}$	$\underline{y}^{(4)}$	$\underline{y}^{(5)}$
20	3	6.43×10^{-3}	1.80×10^{-4}	3.29×10^{-5}	4.97×10^{-6}	1.45×10^{-6}
	5	3.55×10^{-4}	3.80×10^{-5}	6.86×10^{-6}	1.71×10^{-6}	5.38×10^{-7}
	7	3.97×10^{-5}	7.17×10^{-6}	1.79×10^{-6}	5.63×10^{-7}	2.14×10^{-7}
	9	7.25×10^{-6}	1.81×10^{-6}	5.70×10^{-7}	2.17×10^{-7}	9.58×10^{-8}
	11	1.81×10^{-6}	5.73×10^{-7}	2.18×10^{-7}	9.64×10^{-8}	4.80×10^{-8}
40	3	1.61×10^{-3}	1.19×10^{-5}	5.7×10^{-7}	2.41×10^{-8}	2.04×10^{-9}
	5	2.32×10^{-5}	6.98×10^{-7}	3.58×10^{-8}	2.67×10^{-9}	3.35×10^{-10}
	7	7.09×10^{-7}	3.64×10^{-8}	2.77×10^{-9}	8.73×10^{-10}	*
	9	3.65×10^{-8}	2.73×10^{-9}	3.65×10^{-10}	*	*
	11	2.74×10^{-9}	3.27×10^{-10}	*	*	*
80	3	4.00×10^{-4}	6.98×10^{-7}	9.19×10^{-9}	2.08×10^{-9}	7.13×10^{-10}
	5	1.47×10^{-6}	1.14×10^{-8}	1.36×10^{-9}	6.34×10^{-10}	*
	7	1.15×10^{-8}	1.32×10^{-9}	4.25×10^{-10}	*	*
	9	4.12×10^{-10}	*	*	*	*
	11	*	*	*	*	*

TABLE 6.4.1

N	BNDM	ϵ = Norm of Error in the Successive Solutions				
		$\underline{y}^{(1)}$	$\underline{y}^{(2)}$	$\underline{y}^{(3)}$	$\underline{y}^{(4)}$	$\underline{y}^{(5)}$
20	3	5.82×10^{-3}	2.05×10^{-4}	1.48×10^{-5}	9.93×10^{-7}	7.34×10^{-8}
	5	2.89×10^{-4}	1.76×10^{-5}	1.20×10^{-6}	8.73×10^{-8}	6.61×10^{-9}
	7	1.78×10^{-5}	1.22×10^{-6}	8.82×10^{-8}	6.67×10^{-9}	5.23×10^{-10}
	9	1.22×10^{-6}	8.83×10^{-8}	6.67×10^{-9}	5.34×10^{-10}	6.01×10^{-11}
	11	8.83×10^{-8}	6.68×10^{-9}	5.28×10^{-10}	5.66×10^{-11}	4.11×10^{-11}
40	3	1.47×10^{-3}	1.30×10^{-5}	2.43×10^{-7}	4.14×10^{-9}	1.38×10^{-10}
	5	1.88×10^{-5}	2.94×10^{-7}	5.14×10^{-9}	1.55×10^{-10}	*
	7	2.95×10^{-7}	5.13×10^{-9}	1.41×10^{-10}	*	*
	9	5.12×10^{-9}	1.30×10^{-10}	*	*	*
	11	1.09×10^{-10}	*	*	*	*
80	3	3.69×10^{-4}	8.18×10^{-7}	3.83×10^{-9}	4.65×10^{-10}	*
	5	1.19×10^{-6}	4.66×10^{-9}	4.29×10^{-10}	*	*
	7	4.72×10^{-9}	4.11×10^{-10}	*	*	*
	9	5.16×10^{-10}	*	*	*	*
	11	*	*	*	*	*

TABLE 6.4.2

N	BNDM	ε ≡Norm of Error in the Successive Solutions				
		γ ⁽¹⁾	γ ⁽²⁾	γ ⁽³⁾	γ ⁽⁴⁾	γ ⁽⁵⁾
20	3	1.14×10 ⁻²	2.00×10 ⁻⁴	3.58×10 ⁻⁵	5.16×10 ⁻⁶	1.48×10 ⁻⁶
	5	4.44×10 ⁻⁴	4.16×10 ⁻⁵	7.19×10 ⁻⁶	1.77×10 ⁻⁶	5.52×10 ⁻⁷
	7	4.36×10 ⁻⁵	7.53×10 ⁻⁶	1.85×10 ⁻⁶	5.78×10 ⁻⁷	2.18×10 ⁻⁷
	9	7.61×10 ⁻⁶	1.87×10 ⁻⁶	5.86×10 ⁻⁷	2.22×10 ⁻⁷	9.77×10 ⁻⁸
	11	1.88×10 ⁻⁶	5.89×10 ⁻⁷	2.22×10 ⁻⁷	9.83×10 ⁻⁸	4.89×10 ⁻⁸
40	3	2.79×10 ⁻³	1.23×10 ⁻⁵	6.17×10 ⁻⁷	2.50×10 ⁻⁸	2.09×10 ⁻⁹
	5	2.86×10 ⁻⁵	7.60×10 ⁻⁷	3.73×10 ⁻⁸	3.21×10 ⁻⁹	1.41×10 ⁻⁹
	7	7.71×10 ⁻⁷	3.81×10 ⁻⁸	2.94×10 ⁻⁹	3.10×10 ⁻¹⁰	*
	9	3.83×10 ⁻⁸	2.86×10 ⁻⁹	5.81×10 ⁻¹⁰	*	*
	11	3.13×10 ⁻⁹	8.18×10 ⁻¹⁰	*	*	*
80	3	3.93×10 ⁻⁴	7.68×10 ⁻⁷	1.47×10 ⁻⁸	7.38×10 ⁻⁹	2.03×10 ⁻⁹
	5	1.81×10 ⁻⁶	1.26×10 ⁻⁸	8.91×10 ⁻⁹	2.42×10 ⁻⁹	*
	7	1.23×10 ⁻⁸	5.70×10 ⁻⁹	3.84×10 ⁻⁹	*	*
	9	*	*	*	*	*
	11	*	*	*	*	*

TABLE 6.4.3

N	BNDM	ε ≡Norm of Error in the Successive Solutions				
		γ ⁽¹⁾	γ ⁽²⁾	γ ⁽³⁾	γ ⁽⁴⁾	γ ⁽⁵⁾
20	3	2.21×10 ⁻³	3.60×10 ⁻⁵	8.62×10 ⁻⁷	3.83×10 ⁻⁸	5.10×10 ⁻⁹
	5	3.95×10 ⁻⁵	7.90×10 ⁻⁷	1.68×10 ⁻⁸	3.60×10 ⁻¹⁰	*
	7	7.97×10 ⁻⁷	1.70×10 ⁻⁸	3.88×10 ⁻¹⁰	*	*
	9	1.70×10 ⁻⁸	3.81×10 ⁻¹⁰	*	*	*
	11	3.73×10 ⁻¹⁰	*	*	*	*
40	3	5.51×10 ⁻⁴	2.27×10 ⁻⁶	1.37×10 ⁻⁸	1.54×10 ⁻¹⁰	*
	5	2.47×10 ⁻⁸	1.25×10 ⁻⁸	8.53×10 ⁻¹¹	*	*
	7	1.25×10 ⁻⁸	7.73×10 ⁻¹¹	*	*	*
	9	1.11×10 ⁻¹⁰	*	*	*	*
	11	*	*	*	*	*
80	3	1.37×10 ⁻⁴	1.42×10 ⁻⁷	2.59×10 ⁻¹⁰	*	*
	5	1.54×10 ⁻⁷	2.29×10 ⁻¹⁰	*	*	*
	7	2.72×10 ⁻¹⁰	*	*	*	*
	9	*	*	*	*	*
	11	*	*	*	*	*

TABLE 6.4.4

N	BNDM	ϵ \equiv Norm of Error in the Successive Solutions				
		$y^{(1)}$	$y^{(2)}$	$y^{(3)}$	$y^{(4)}$	$y^{(5)}$
20	3	1.78×10^{-3}	1.82×10^{-5}	6.67×10^{-7}	5.79×10^{-8}	1.15×10^{-8}
	5	2.32×10^{-5}	3.62×10^{-7}	6.32×10^{-9}	6.84×10^{-11}	*
	7	3.66×10^{-7}	6.35×10^{-9}	1.05×10^{-10}	*	*
	9	6.34×10^{-9}	9.74×10^{-11}	*	*	*
	11	9.11×10^{-11}	*	*	*	*
40	3	4.50×10^{-4}	1.15×10^{-6}	1.07×10^{-8}	3.38×10^{-10}	*
	5	1.48×10^{-6}	5.83×10^{-9}	1.82×10^{-10}	*	*
	7	5.80×10^{-9}	2.73×10^{-10}	*	*	*
	9	2.26×10^{-10}	*	*	*	*
	11	*	*	*	*	*
80	3	1.13×10^{-4}	7.25×10^{-8}	6.03×10^{-10}	*	*
	5	9.29×10^{-8}	1.71×10^{-10}	*	*	*
	7	*	*	*	*	*
	9	*	*	*	*	*
	11	*	*	*	*	*

TABLE 6.4.5

N	BNDM	ϵ \equiv Norm of Error in the Successive Solutions		
		$y^{(1)}$	$y^{(2)}$	$y^{(3)}$
20	3	1.40×10^{-4} (4)	3.29×10^{-7} (2)	2.08×10^{-9} (2)
	5	7.92×10^{-6} (4)	1.81×10^{-6} (2)	7.04×10^{-7} (2)
	7	7.73×10^{-7} (4)	1.68×10^{-7} (2)	5.89×10^{-8} (2)
40	3	3.56×10^{-5} (6)	2.02×10^{-8} (4)	8.16×10^{-11} (2)
	5	5.33×10^{-7} (6)	1.20×10^{-7} (2)	4.68×10^{-8} (2)
	7	No convergence		

TABLE 6.4.6

N	BNDM	ε ≡Norm of Error in the Successive Solutions			
		$\underline{y}^{(1)}$	$\underline{y}^{(2)}$	$\underline{y}^{(3)}$	$\underline{y}^{(4)}$
10	3	5.03×10^{-4}	2.7×10^{-6}	2.01×10^{-6}	6.62×10^{-7}
	5	1.11×10^{-4}	2.94×10^{-5}	1.11×10^{-5}	4.29×10^{-6}
20	3	1.29×10^{-4}	2.23×10^{-7}	1.24×10^{-8}	2.87×10^{-9}
	5	9.31×10^{-6}	2.31×10^{-6}	8.71×10^{-7}	3.09×10^{-7}
40	3	3.29×10^{-5}	1.39×10^{-8}	4.54×10^{-10}	*
	5	7.02×10^{-7}	1.60×10^{-7}	6.31×10^{-7}	2.22×10^{-8}

TABLE 6.4.7

6.5 APPLICATION OF FIRML ON EIGENPROBLEMS

Various types of matrices are considered in this section, for which the difference between the diagonal element and the summation of the off-diagonal elements in each row may be negative or non-positive.

Furthermore, we shall confine ourselves to apply the Inverse Power Method (IPM) as discussed in Section 2.4 to deal with the eigenproblems under consideration. This method basically involves the determination of the dominant eigenvalue, λ^{-1} (say), of the inverse of a non-singular matrix, A , via the use of equation (2.4.12) which yields λ^{-1} when an iterative process is applied as follows:

$$\left. \begin{array}{ll} \text{Step 1:} & A\mathbf{y}^{(s+1)} = \mathbf{x}^{(s)} \\ \text{Step 2:} & \beta^{(s+1)} = \max_i |y_i^{(s+1)}| \\ \text{Step 3:} & \mathbf{x}^{(s+1)} = \frac{1}{\beta^{(s+1)}} \mathbf{y}^{(s+1)} \\ \text{and Step 4:} & ||\mathbf{x}^{(s+1)} - \mathbf{x}^{(s)}|| < \epsilon \text{ (given tolerance) ,} \end{array} \right\} \quad (6.5.1)$$

where vector \mathbf{x} is the corresponding eigenvector of λ , $s=0,1,2,\dots$, $\mathbf{x}^{(0)}$ is chosen as a *unit* vector. Then the smallest eigenvalue in modulus of A is obtained immediately.

Algorithm FIRML is used to solve the linear system of Step 1, notably the factorisation of $A=UL$ (see subsection 4.4.1) takes place once only at $s=0$. An illustrative example of applying IPM is given in Example 6.5.1 below.

Example 6.5.1

Let matrix A be a septadiagonal matrix (i.e. $r=3$) of size (11×11) as given in (Gregory and Karney (1969)),

adopted to accelerate the rate of convergence. A reverse result may be expected if $(A+\rho I)$ shift is considered; we shall refer to the latter strategy as *positive shifting* as a distinguishable concept to the former. The sole reason behind considering this unusual strategy is to make algorithm FIRML applicable to certain forms of matrices. The following example may clarify this point further.

Example 6.5.2

Let matrix A be

$$A = \begin{bmatrix} 5 & -4 & 1 & & & & & & & \\ & -4 & 6 & & & & & & & \\ & 1 & & & & & & & & \\ & & 0 & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \end{bmatrix} \quad (10 \times 10) \quad (6.5.2a)$$

whose eigenvalues are given by,

$$\lambda_k = 16 \sin^4\left(\frac{k\pi}{2(N+1)}\right), \quad k=1,2,\dots,N, \quad (\text{Gregory \& Karney (1969)}) \quad (6.5.2b)$$

It was noticed that the factorisation of matrix A in its present form cannot be achieved by FIRML (c.f. A_2 at the end of Section 6.3), but it can be obtained if the positive shifting strategy is adopted. Thus, A in (6.5.2) is shifted to

$$A+\rho I, \quad \text{with } \rho=0.4 \text{ or } 0.5, \quad (6.5.3)$$

Then the procedure (6.5.1) is used on the matrix in (6.5.3), bearing in mind that the s^{th} step of the iteration λ_{\min} is defined as

$$\lambda_{\min}^{(s)} = \frac{1}{\beta^{(s)}} - \rho, \quad (6.5.4)$$

where β is as defined in (6.5.1).

Subsequently, it was observed, as expected, that the number of iterations involved in factorising the matrix $(A+\rho I)$ is decreased as ρ

is increased (81 and 40 iterations for $\rho=0.4$ and $\rho=0.5$ respectively, with $\epsilon_f=0(10^{-9})$), and conversely for the number of iterations of the procedure (6.5.1), i.e. $\lambda_{\min}=0.006563274$ to an accuracy of $0(10^{-9})$ is obtained in 23 and 27 iterations for $\rho=0.4$ and 0.5 respectively. This value of λ_{\min} coincides with the first 8 significant figures of the value obtained by λ_1 in (6.5.2b).

On the other hand, another alternative strategy may be applied to the matrix A in (6.5.2a), where no shifting is considered, viz.

Permutation Matrix Strategy

If a permutation matrix P (see Definition 2.1.9) of order 10 is chosen as follows:

$$P = [\underline{e}_1 \ \underline{e}_4 \ \underline{e}_2 \ \underline{e}_5 \ \underline{e}_3 \ \underline{e}_6 \ \underline{e}_7 \ \underline{e}_{10} \ \underline{e}_8 \ \underline{e}_9] , \quad (6.5.5)$$

where \underline{e}_k are column-vectors, each of 9 zero components and the k^{th} component is 1.

Since the matrix P is orthogonal (see Definition 2.1.9), the orthogonal transformation (see Definition 2.4.2) implies that $P^T A P$ and A share similar eigenvalues whilst any eigenvector of the former matrix, \underline{v} say, is defined as $P\underline{x}=\underline{v}$. Subsequently, the procedure (6.5.1) may be replaced by:

$$\left. \begin{array}{ll} \text{Step 1':} & P^T A \underline{y}^{(s+1)} = \underline{v}^{(s)} , \\ \text{Step 2':} & \text{same as in (6.5.1)} , \\ \text{Step 3':} & \underline{v}^{(s+1)} = \frac{1}{\beta^{(s+1)}} \underline{y}^{(s+1)} , \\ \text{Step 4':} & \text{same as in (6.5.1)} . \end{array} \right\} \quad (6.5.6)$$

Clearly, at the s^{th} step of the above procedure the eigenvector of A is computed from,

$$\underline{x}^{(s)} = P^{-1} \underline{v}^{(s)} = P^T \underline{v}^{(s)} . \quad (6.5.7)$$

Now, the numerical results showed that $P^T A$ is factorisable, and 75 iterations are sufficient to produce $\epsilon_f=0(10^{-8})$. Whilst procedure (6.5.6)

at $s=5$ was sufficient to give $\lambda_{\min} = 0.006563283$ which may be improved if ϵ_f is decreased further. It should be noticed that matrix $P^T A$ is treated as $r=4$ (i.e. of bandwidth 9), since the fourth off-diagonal (above and below) includes non-zero elements.

Fortunately, matrix A in (6.5.2a) with small size was not sufficiently awkward to use the above strategy, otherwise in the light of some attempts in this respect lead one to believe that the generalisation of this strategy for any size may be possible if further investigation is pursued.

On the other hand, another interesting application of the latter strategy (or a combination of both of the above strategies) on singular matrices such as the one given in Example 6.5.3 below.

Example 6.5.3

Let a tridiagonal matrix A be singular and defined as

$$A = \begin{bmatrix} 2 & -2 & & & \\ -1 & 2 & -1 & & 0 \\ & -1 & 2 & -1 & \\ & 0 & -1 & 2 & -1 \\ & & & -2 & 2 \end{bmatrix}_{(5 \times 5)} \quad (6.5.8)$$

where $\lambda_{\min} = 0$, and choose a permutation matrix P as,

$$P = [\underline{e}_1 \ \underline{e}_3 \ \underline{e}_2 \ \underline{e}_4 \ \underline{e}_5] \quad (6.5.9)$$

where \underline{e}_k , $k=1, \dots, 5$, vectors are defined likewise in (6.5.5).

Therefore, having matrix $P^T A$ formulated and treated as a quindagonal matrix (i.e. $r=2$) for similar reasons indicated in Example 6.5.2, the procedure (6.5.6) can now be applied. However, the obtained factorisation of $P^T A$ to $\epsilon_f = 0(10^{-8})$ is achieved in 32 iterations whilst (6.5.6) should be taken one step only, thus at $s=0$ $\lambda_{\min} = 4.58 \times 10^{-9}$ (and $\beta = 2.18 \times 10^8$) and the components of its eigenvector are of $O(10^{-10})$; if s is increased, the rounding errors rapidly dominate the value of λ . But s may be increased if a shifting strategy is adopted to $P^T A$ in which the number of iterations

required for the factorisation of $P^T A + \rho I$ is decreased; also this number may be reduced to less than half for values of ρ in the range $1 < \rho < 1.8$. No optimal value of ρ was obtainable due to the special structure of the matrix (for example, with $\rho = 0.4$ or 0.6 the factorisation fails). A further point which may be considered as an advantage of the shifting strategy for this particular matrix A in (6.5.8), the coefficient matrix of the system (4.4.45a) involved in the applied algorithm is nearly singular only with $\rho = 0$.

CHAPTER 7

APPLICATIONS TO PARTIAL DIFFERENTIAL EQUATIONS

7.1 INTRODUCTION

To distinguish the algorithms presented in Chapter 5 from those in Chapter 4 (i.e. BANDSOLVERs) will be generally referred to as BLOCKSOLVERs. A block factorisation for some well-known block matrices (tridiagonal and quindagonal types) will be considered. Also, applications of BLOCKSOLVERs to numerical examples, such as the Laplace, 2nd order Elliptic and Biharmonic equations are included. For some of these examples a comparison between the appropriate BLOCKSOLVER and BANDSOLVER (FIRM1) is made.

7.2 ON THE FACTORISATION INVOLVED IN BLOCKSOLVERS

The block tridiagonal and quindagonal matrices considered here are respectively of the form,

$$A_1 = \begin{bmatrix} B & C & & & 0 \\ C & B & C & & \\ & C & B & C & \\ & & C & B & C \\ 0 & & & C & B \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} B_1 & C & D & & & 0 \\ C & B & C & D & & \\ D & C & B & C & D & \\ & 0 & & D & C & B & C \\ & & & D & C & B_2 \end{bmatrix}$$

where the sub-matrices in both A_1 and A_2 are of size $(m \times m)$. Factorisation of A_1 and A_2 is assumed to take place as given in Chapter 5, in which the factor matrices of the former include the sub-matrices Q_0, Q_1 and the latter include the submatrices $\beta, \delta, Q_0, Q_1, Q_2, \gamma, \epsilon$. Each of these sub-matrices is of order m . Furthermore, each sub-matrix in both A_1 and A_2 will be assumed as of non-periodic type (unless it is stated) and abbreviated as in Section 6.2.

We now consider the following two examples related to A_1 and A_2 respectively.

Example 7.2.1

Consider A_1 such that its sub-matrices are defined as follows:

- (i) $B(0, -1, 4, -1, 0)$ and $C = -I$ (I unit matrix),
- (ii) $B(0, -1, k, -1, 0)$ and it is periodic, with $k \geq 4$, $C = -I$,
- (iii) $B = 4I$, $C(0, -1, 0, -1, 0)$

and

- (iv) $B(0, -4, 20, -4, 0)$, $C(0, -1, -4, -1, 0)$

(both (iii) and (iv) taken from Gregory and Karney (1969)).

N.B. The eigenvalues of A_1 for the case (i) and (iv) are given respectively by the formulae

$$\lambda_{ij} = 4 - 2\left(\cos \frac{i\pi}{p} + \cos \frac{j\pi}{q}\right), \quad i=1, 2, \dots, p-1, \quad j=1, 2, \dots, q-1. \quad (7.2.1)$$

where $p-1$ is the order of the sub-matrices B and C and $(p-1)(q-1)$ is the

order A_1 , (Smith (1978)).

$$\lambda_{ij} = 20 - 8\cos k\theta - 8\cos j\theta - 4\cos k\theta \cos j\theta \quad (7.2.2)$$

where $\theta = \frac{\pi}{m+1}$, $k, j = 1, 2, \dots, m$, and m is the order of B and C while A_1 is of order m^2 (Gregory and Karney (1969)).

Example 7.2.2

Consider A_2 such that its sub-matrices are defined as follows:

(i)
$$\begin{bmatrix} 59 & -16 & 1 & & & & \\ -16 & 60 & & & & 0 & \\ 1 & & & & & & \\ & & & & & & \\ & & & & & & \\ & 0 & & 1 & -16 & 60 & -16 \\ & & & & 1 & -16 & 59 \end{bmatrix}, \quad C = -16I, \quad D = I,$$

$$B_1 = B_2 = B - I$$

and

(ii) as given in the system (7.3.30) (next section).

It should be noticed that the aforementioned examples have been selected such that the matrices A_1 and A_2 are block-diagonally dominant (see Section 5.2 and 5.4) in which the required condition for the appropriate algorithm is fulfilled.

As far as the factorisation is concerned two essential points will be stressed, the computation of the matrix square root (MTXSQRT) and the properties of the submatrices Q_0, Q_1 in Example 7.2.1 (or the relevant ones in Example 7.2.2).

In Example 7.2.1, the two MTXSQRTs involved in the iterative process to determine Q_0 and Q_1 (see (5.1.8)) require on average 5 or 6 iterations for $m=5$ (the order of Q_0 and Q_1) and increase by 1 or 2 iterations for $m=10$ or 20. The highest number of iterations is related to the (positive

definite) matrix $B+2C$ which possesses weaker diagonal elements in comparison with the matrix $B-2C$. For this type of matrix, obtaining its square root (using the method of Section 2.5 and single precision) becomes critical as m increases. For instance, for $m=20$, when MTXSQRT was computed, it was multiplied by itself and then compared to the original matrix. The zero elements of the original matrix appeared to be of $O(10^{-6})-O(10^{-8})$ in the product. In such cases, either double precision has to be used or an alternative method of computing MTXSQRT is recommended.

In the cases (i)-(iv) of the current example some common properties of both submatrices Q_0 and Q_1 exist. Q_0 is strictly-diagonally dominant. Q_0 and Q_1 are symmetric matrices. $||Q_0|| > ||Q_1||$, this is a vital property since it is associated with the stability of the elimination process involved in the solution (see Section 5.1 or 5.2). The two norms $||Q_0||$ and $||Q_1||$ do decrease and increase respectively as m increases (even if the appropriate MTXSQRTs are evaluated accurately). This is related to the changes in the elements of Q_0 and Q_1 . The diagonal and off-diagonal elements of Q_0 decrease and increase (in modulus) respectively as m increases; whilst for Q_1 all its elements increase (in modulus). These changes may become significant for large m in which the stability of the elimination process will be affected seriously.

In Example 7.2.1 (ii), where the sub-matrix B is periodic, both Q_0 and Q_1 are circulant matrices with *constant* elements. For instance, for $k=6$, Q_0 and Q_1 have the following form (for $m=10$),

$$Q_0 = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_4 & a_3 & a_2 & a_1 \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \end{bmatrix}$$

symmetric

and

$$Q_1 = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_4 & b_3 & b_2 & b_1 \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \end{bmatrix}$$

symmetric

where

$$\begin{aligned} a_0 &= 2.3919486932, & b_0 &= -0.4252634457, \\ a_1 &= -0.2171539811, & b_1 &= -0.0393190914, \\ a_2 &= -0.0113295412, & b_2 &= -0.0056939231, \\ a_3 &= -0.0013152862, & b_3 &= -0.0009577176, \\ a_4 &= -0.0002109688, & b_4 &= -0.0001823602, \\ a_5 &= -0.0000734897, & b_5 &= -0.0000684475. \end{aligned}$$

The matrices Q_0, Q_1 have a similar property for $k=5$, whilst for $k=4$, the matrix $B+2C$ becomes singular and its square root could not be attained.

In Example 7.2.1(iii), the elements of both Q_0 and Q_1 are zero on the following off-diagonals; the 1st, 3rd, 5th, ..., etc. for Q_0 , 2nd, 4th, 6th, ..., etc. for Q_1 as well as its diagonal.

We now consider Example 7.2.2, where the factor matrices include the

main sub-matrices, Q_0, Q_1 and Q_2 ; in addition to the other four sub-matrices β, δ, γ and ϵ (see the factor matrices in (5.4.24)). As indicated in Section 5.3 the determination of Q_0, Q_1 and Q_2 involves 3 MTXSQRTs (see (5.3.6) and (5.3.7)). The remarks relating to the MTXSQRT indicated in Example 7.2.1 also apply here; the matrix of weak diagonal elements is $B+2C+2D$ to which the highest number of iteration is related (7 to 8 iterations). Also, the general properties of the submatrices in the previous example are applied to Q_0, Q_1 and Q_2 , in particular the norm-relation, $||Q_0|| > ||Q_1|| + ||Q_2||$ is valid. As Q_0, Q_1 and Q_2 are the solution of the matrix non-linear system (5.4.25b) in which each of its three equations should be satisfied by the solution matrices. The results showed that this is true for Example 7.2.2(i), whilst for Example 7.2.2(ii) where the sub-matrix C (of the original matrix A_2) is tridiagonal, the solution matrices Q_0, Q_1 and Q_2 do satisfy the first and the last equation of the system (5.4.25b) and its second equation is satisfied as follows. The two matrices $(Q_0 Q_1 + Q_1 Q_2)$ and $(Q_1 Q_0 + Q_2 Q_1)$ have some common equal elements but opposite in sign, thus by adding the two matrices yields $2C$.

Finally, the computation of the sub-matrices β, δ, γ and ϵ by the scheme (5.4.26) was preferred to (5.4.27). This is because the latter required 2 MTXSQRTs which makes the factorisation procedure uneconomical and not accurate enough in comparison with the former; in addition the two relevant matrices (see (5.4.27)) may not be positive-definite matrices. However, in general, as long as the matrices Q_0 and Q_2 in (5.4.26) are non-singular (as in our case) the first scheme is recommended.

or its equivalent block form

$$\begin{bmatrix} B & C & 0 \\ C & B & C \\ 0 & C & B \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \quad (7.3.4b)$$

where B and C are 7×7 matrices, each sub-vector is of length 7, whose components can easily be recognised from (7.3.4a).

Example 7.3.1(ii)

The Laplace equation (7.3.1) on a *unit* square R (say), and subject to the Dirichlet's boundary condition,

$$\left. \begin{aligned} u(x,0) &= f(x) = x(1-x), \quad 0 \leq x \leq 1 \\ u(x,1) &= 0, \\ u(0,y) &= u(1,y) = 0, \quad 0 \leq y \leq 1 \end{aligned} \right\} \quad (7.3.5)$$

The *analytical* solution of (7.3.1) under the boundary conditions (7.3.5) is given by,

$$u(x,y) = \sum_{k=1}^{\infty} \left[\beta_k \frac{\sinh(k\pi(1-y))}{\sinh(k\pi)} \sin(k\pi x) \right], \quad (7.3.6)$$

where

$$\beta_k = \begin{cases} 8/(k\pi)^3, & \text{for } k \text{ odd,} \\ 0 & \text{for } k \text{ even,} \end{cases}$$

(Burak et al (1964)).

By partitioning the region R into m^2 equal squares, each of length h (i.e. $h=1/(m+1)$), we can replace the Laplace equation (7.3.1) by two types of finite-difference equations, that is by,

(i) using the 5-point formula, we have

$$\frac{1}{h^2} (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}) = 0, \quad i,j=1,2,\dots,m, \quad (7.3.7)$$

and

(ii) using the 9-point formula (see Section 3.4), we have

$$\frac{1}{6h^2} (4u_{i+1,j} + 4u_{i-1,j} + 4u_{i,j+1} + 4u_{i,j-1} + u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1}) = 0, \quad i,j=1,2,\dots,m. \quad (7.3.8)$$

Any of these two equations under the boundary conditions (7.3.5) lead to a block tridiagonal system of the form,

$$\begin{bmatrix} B & C & & 0 \\ C & B & C & \\ & 0 & B & C \\ & & C & B \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix} \quad (7.3.9)$$

where the sub-matrices and the sub-vectors are of size $(m \times m)$ and $(m \times 1)$ respectively. Moreover B, C and z_i are defined as follows,

(a) for the finite-difference equation (7.3.7), we have

$$B = \begin{bmatrix} 4 & -1 & & 0 \\ -1 & 4 & & \\ & -1 & 4 & \\ & & -1 & 4 \end{bmatrix}_{(m \times m)}, \quad C = -I \text{ (I unit matrix)} \quad (7.3.9a)$$

$$\text{and } z_{-1} = 0, \text{ for } i=1,2,\dots,m-1, \quad z_m \equiv [z_{m,1}, z_{m,2}, \dots, z_{m,m}]^T \quad \left. \begin{array}{l} \text{where } z_{m,j} = x_j(1-x_j) = jh(1-jh), \quad j=1,2,\dots,m \\ (7.3.9b) \end{array} \right\}$$

and

(b) for the finite-difference equation (7.3.8) we have

$$B = \begin{bmatrix} 20 & -4 & & 0 \\ -4 & 20 & & \\ & -4 & 20 & \\ & & -4 & 20 \end{bmatrix}_{(m \times m)}, \quad C = \begin{bmatrix} -4 & -1 & & 0 \\ -1 & -4 & & \\ & -1 & -4 & \\ & & -1 & -4 \end{bmatrix}_{(m \times m)} \quad (7.3.10a)$$

$$\text{and } z_{-1} = 0, \text{ for } i=1,2,\dots,m-1, \quad z_m \equiv [z_{m,1}, z_{m,2}, \dots, z_{m,m}]^T, \quad \left. \begin{array}{l} \text{where } z_{m,j} = x_{j-1}(1-x_{j-1}) + 4x_j(1-x_j) + x_{j+1}(1-x_{j+1}), \quad x_0 = x_{m+1} = 0, \\ = (j-1)h(1-(j-1)h) + 4jh(1-jh) + (j+1)h(1-(j+1)h), \\ j=1,2,\dots,m, \\ \text{(last term}=0 \text{ for } j=m). \end{array} \right\} \quad (7.3.10b)$$

Example 7.3.2(i)

The linear 2nd order *Elliptic Equation* expressed by the form,

$$4\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \alpha u = f(x,y) \quad , \quad (7.3.11)$$

on a *unit* square, \mathbb{R} (say), subject to the boundary conditions

$$u = 0 \quad \text{on the boundaries of } \mathbb{R} \quad (7.3.12)$$

and the *exact* solution is given by

$$u = 2(x^2 - x)(\cos 2\pi y - 1), \quad 0 \leq x, y \leq 1, \quad (7.3.13)$$

(Rice et al (1980)).

By virtue of (7.3.13), $f(x,y)$ in (7.3.11) have the form,

$$f(x,y) = 2(\cos 2\pi y - 1) [8 - \alpha(x^2 - x)] - 8\pi^2(x^2 - x)\cos 2\pi y,$$

and with $\alpha=2$, we have

$$f(x,y) = 4(\cos 2\pi y - 1) [8 - (x^2 - x)] - 8\pi^2(x^2 - x)\cos 2\pi y. \quad (7.3.14)$$

Since the partition of the \mathbb{R} is assumed to take place as in Example 7.3.1(ii), the application of the 5-point formula would enable us to replace equation (7.3.11) under the boundary conditions (7.3.12) by the difference equation,

$$4u_{i+1,j} + 4u_{i-1,j} - u_{i,j+1} + u_{i,j-1} - (10 + 2h^2)u_{i,j} = h^2 f_{i,j}, \quad i,j=1,2,\dots,m, \quad (7.3.15)$$

where $f_{i,j}$ is the discretised form of $f(x,y)$ in (7.3.14), i.e.

$$f_{i,j} \equiv f(ih, jh) = f(x_i, y_j), \quad h=1/(m+1).$$

Then, equation (7.3.15) immediately yields a block tridiagonal system of the form,

$$\begin{bmatrix} B & C & & & 0 \\ & C & & & \\ & & 0 & & \\ & & & C & \\ 0 & & & & C & B \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix} \quad (7.3.16)$$

where B, C and z_i are defined as follows,

$$B = \begin{bmatrix} k & -4 & & & 0 \\ -4 & & & & \\ & & & & \\ 0 & & & & \\ & & & & -4 \\ & & & -4 & k \end{bmatrix}_{(m \times m)}, \quad k = 10 + 2h^2, \quad C = -I, \quad (7.3.17a)$$

and $\underline{z}_i \equiv [z_{i,1}, z_{i,2}, \dots, z_{i,m}]^T$ such that, $\left. \begin{aligned} z_{i,j} &= -h^2 f_{i,j}, \quad i, j = 1, 2, \dots, m. \end{aligned} \right\} \quad (7.3.17b)$

Example 7.3.2(ii)

We now reconsider the problem of Example 7.3.2(i) with the periodic boundary conditions in the y-direction, i.e. we have, equation (7.3.11) subject to the boundary conditions,

$$\left. \begin{aligned} u(0,y) &= u(1,y) = 0, \quad 0 \leq y \leq 1, \\ u(x,0) &= u(x,1) \\ \frac{\partial u}{\partial y}(x,0) &= \frac{\partial u}{\partial y}(x,1) \end{aligned} \right\}, \quad 0 < x < 1. \quad (7.3.18)$$

Under these conditions the solution of the grid points 31, ..., 35 (Figure 7.3.1), is the same at the points 1, ..., 5 respectively, (on assumption $m=5$, $h=1/6$).

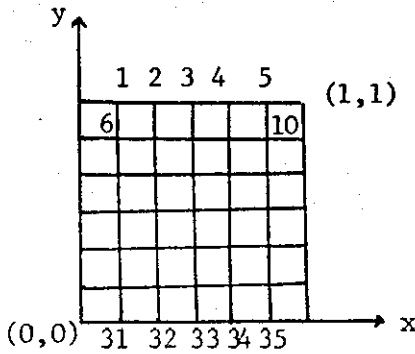


FIGURE 7.3.1: Periodic boundary conditions in the y-direction

Thus, by following the procedure of the previous example, instead of the system (7.3.16), we should obtain,

$$\begin{bmatrix} B & C & & 0 & C \\ & C & & & \\ & & & 0 & \\ & & & & C \\ C & & 0 & C & B \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} \quad (7.3.19)$$

where $N=m+1$, B and C as given in (7.3.17a), and \underline{z}_i is defined as

$$\underline{z}_i = [z_{i,1}, z_{i,2}, \dots, z_{i,m}]^T,$$

and

$$z_{i,j} = -h^2 f_{i,j}, \quad i=1,2,\dots,N, \quad j=1,2,\dots,m \quad (\text{c.f. (7.3.17b)}).$$

Example 7.3.2(iii)

We consider the Elliptic Equation expressed by

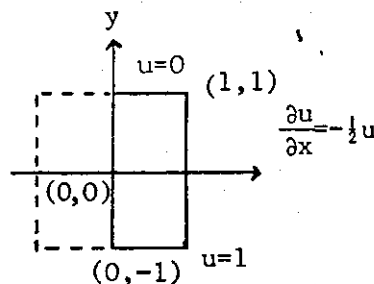
$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - 32u = 0, \quad (7.3.20)$$

on a square region bounded by $-1 \leq x \leq 1$, $-1 \leq y \leq 1$, with boundary conditions of the form,

$$\left. \begin{array}{l} \text{(i) } u=0 \text{ on } y=1, \quad -1 \leq x \leq 1, \\ \text{(ii) } u=1 \text{ on } y=-1, \quad -1 \leq x \leq 1, \\ \text{(iii) } \frac{\partial u}{\partial x} = -\frac{1}{2}u \text{ on } x=1, \quad -1 < y < 1 \\ \text{(iv) } \frac{\partial u}{\partial x} = \frac{1}{2}u \text{ on } x=-1, \quad -1 < y < 1 \end{array} \right\} \quad \begin{array}{l} \\ \\ \text{(Robin's conditions)} \\ \end{array} \quad (7.3.21)$$

(Smith (1978)).

Since the problem (as indicated in the above reference) is symmetric at $x=0$, the solution is considered only on half of the region (i.e. Figure 7.3.2, the dotted-region is ignored).



Solution is symmetric
at $x=0$

FIGURE 7.3.2

The current problem was treated with both 5-point and 9-point finite-difference formulae (see Section 3.5) which respectively yield the following difference equations,

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - (4+32h^2)u_{i,j} = 0, \quad \begin{matrix} i=1,2,\dots,m, \\ j=1,2,\dots,N, \end{matrix} \quad (5\text{-point formula}) \quad (7.3.22)$$

and

$$4(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) + (u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1}) - 20(32+6h^2)u_{i,j} = 0, \quad \begin{matrix} i=1,2,\dots,m, \\ j=1,2,\dots,N. \end{matrix} \quad (9\text{-point formula}) \quad (7.3.23)$$

where $h=1/(m-1)$, $N=2m-3$.

Any of the two equations (7.3.22) or (7.3.23) under the appropriate boundary conditions in (7.3.21) [bearing in mind that $\frac{\partial u}{\partial x}$ at $x=1$ is approximated by $(u_{i+1,j} - u_{i-1,j})/2h$] yields a block tridiagonal system of the form,

$$\begin{bmatrix} B & C & & 0 \\ C & & & \\ & 0 & & \\ & & C & B \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}, \quad (7.3.24)$$

where B, C and z_i are defined as below.

(a) for the finite-difference equation (7.3.22), we have

$$B = \begin{bmatrix} k & -2 & & 0 \\ -1 & k & -1 & \\ & 0 & & \\ & & -1 & k & -1 \\ & & & -2 & k+h \end{bmatrix}_{(m \times m)}, \quad k = 4+32h^2, \quad C = I_{(m \times m)} \text{ (I unit matrix)}$$

and $z_i = 0$, $i=1,2,\dots,N-1$, the m components of z_N are 1's.

and (b) for the finite-difference equation (7.3.23), we have,

$$B = \begin{bmatrix} k & -8 & & & \\ -4 & k & -4 & & \\ & & & 0 & \\ & 0 & & & \\ & & & & \\ & & & -4 & k & -4 \\ & & & & & -8 & k+4h \end{bmatrix}_{(m \times m)}, \quad C = \begin{bmatrix} -4 & -2 & & & \\ -1 & -4 & -1 & & \\ & & & 0 & \\ & 0 & & & \\ & & & & \\ & & & -1 & -4 & -1 \\ & & & & & -2 & -4+h \end{bmatrix}_{(m \times m)}$$

and $\underline{z}_1 = \underline{0}, i=1,2,\dots,N; \underline{z}_N = [6,6,\dots,6,6-h]^T$.

Example 7.3.3 (Fourth-order elliptic p.d.e.)

Consider the *Biharmonic Equation* expressed as,

$$\nabla^4 u = \frac{\partial^4 u}{\partial x^4} + \frac{2\partial^2 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4} = r(x,y), \quad (7.3.25)$$

in the *unit square* R , subject to the condition,

$$\underline{u} = \underline{n} \cdot \nabla \underline{u} = 0, \text{ on the boundaries } \partial R, \quad (7.3.26)$$

where \underline{n} is the *unit vector normal* and $r(x,y)$ is a prescribed function on R . For

$$r(x,y) = 8[3y^2(1-y)^2 + 3x^2(1-x)^2 + (6x^2 - 6x + 1)(6y^2 - 6y + 1)], \quad (7.3.27a)$$

the exact solution is known to be,

$$u = x^2(1-x)^2 y^2(1-y)^2, \quad (7.3.27b)$$

and for

$$r(x,y) = (2\pi)^4 [4\cos 2\pi x \cos 2\pi y - \cos 2\pi x - \cos 2\pi y], \quad (7.3.28a)$$

the exact solution is given by,

$$u = (1 - \cos 2\pi x)(1 - \cos 2\pi y). \quad (\text{Bauer and Reiss, (1972)}) \quad (7.3.28b)$$

By partitioning the region R into m^2 equal squares, each of length h (i.e. $h=1/(m+1)$), equation (7.3.25) can be replaced at the grid points by the difference equation derived from applying the 13-point finite-difference formula (see Section 3.5), i.e.,

- (*1) In any off-diagonal of L_1 in (7.4.1), the elements are repeated in a cycle of length 7 (or in general, of length equivalent to the order of the relevant sub-matrices involved in the coefficient matrix as in (7.3.4b)).
- (*2) The off-diagonal elements of L_1 are all less than 1 in modulus.
- (*3) As shown in Table 7.4.1, $\ell_{1,i}$ and $\ell_{7,i}$, $i=1,2,\dots,7$, have the largest magnitudes amongst the remaining off-diagonals; or in general, the off-diagonals which match those of the non-zero elements in the original matrix, possess the largest value in modulus (note that even in the first off-diagonal $\ell_{1,7}$ is the only element of low order which matches the zero elements in the original matrix, denoted by x in (7.3.4a)).
- (*4) The diagonal of U_1 in (7.4.2) is of largest element (in magnitude) than the off-diagonals (see d_i in Table 7.4.1).

In Example 7.3.1(ii) and Example 7.3.2(i) since the coefficient matrix is symmetric, remark (*1) is applicable; whilst in Example 7.3.2(iii), the sub-matrices B (or both B and C) in (7.3.24) are non-symmetric, no regular repetition occurs in the elements of the factor matrices. This is applied to Example 7.3.3 due to the sub-matrices B_1 and B_2 in (7.3.30). However, the main conclusions to draw from applying FIRML in these examples are:

- (**1) The factorisation procedure to yield the two factor matrices was achieved with very high accuracy (i.e. ϵ_f (see Section 6.1) was obtained up to $0(10^{-10})-0(10^{-12})$). For instance, this accuracy was achieved in 56, 30 iterations for the system (7.3.16) and (7.3.24) whose coefficient matrices are of order 121, 135 respectively; whilst for the order 25, 35 respectively the relevant number of iterations reduce to nearly half. The slowest convergence was observed in Example 7.3.3 in which the factorisation procedure to

yield $\varepsilon_f = 0(10^{-10})$ requires 195 iterations since the coefficient matrix was of order 121.

(**2) A further illustration related to the remarks (*2)-(*4) can be pointed out as follows:

The maximum magnitude of elements of L_1 ($\max |\ell_{i,j}|$, say) does not grow much as N (the order of L_1) increases; whilst the minimum magnitude of the diagonal elements of U_1 ($\min |u_{1,i}|$, say) does not decrease as much. For instance, in Example 7.3.2(i), $\max |\ell_{i,j}| \leq 0.56$ and $\min |u_{1,i}| \geq 7.7$ for $N=25$, whilst for $N=121$ $\max |\ell_{i,j}| \leq 0.6$ and $\min |u_{1,i}| \geq 7$. Moreover, the values $u_{1,i}$ oscillate in a rather 'narrow' range whose upper bound is slightly larger or smaller than the maximum elements of the diagonal of the original matrix.

Example 7.3.3 is excluded since for $N=121$, $10 \leq |u_{1,i}| \leq 17.8$.

The above remarks may lead to conclude that FIRM1 is applicable to the block systems and it is more efficient than its applications in Chapter 6.

On the other hand, in general, both the BANDSOLVER (FIRM1) and the appropriate BLOCKSOLVER yield very close results for the same example. For instance, the equivalent systems (7.3.4a) and (7.3.4b) were solved by FIRM1 and FIRM2 respectively and their numerical results are given in Table 7.4.2. In this table the results of both solvers agree up to at least 8 significant figures; in Table 7.4.3 which belongs to Example 7.3.2(iii) the solutions given by the two solvers for the system (7.3.24) (for $h=1/8$, using the 9-point formula) agree up to 9 or 10 significant figures. This was the case, generally, in the remaining examples that were tested.

A further discussion of the results is based on theoretical principles (see Section 3.5), that is, an improvement in the computed solution may be obtained by,

- (a) reducing the mesh size, or
- (b) applying a high-order finite-difference approximation.

Let for a given mesh h $\tau(h)$ be the pointwise error which is defined as the difference between the exact and the computed values, divided by the maximum value of the exact solution.

According to part (a) above, we have the following results: in Example 7.3.2(i), $\tau(h_1)=2.546 \times 10^{-2}$, $\tau(h_2)=6.217 \times 10^{-3}$, $\tau(h_3)=3.485 \times 10^{-3}$ for $h_1=1/6$, $h_2=1/12$, $h_3=1/16$. In Example 7.3.2(ii), $\tau(h_1)=8.27 \times 10^{-3}$, $\tau(h_2)=2.06 \times 10^{-3}$, for $h_1=1/10$, $h_2=1/20$. The ratio of $\tau(h_1):\tau(h_2)$ in the two examples is 4.1 and 4.01 respectively; where in theory it must be 4 since the truncation error of the 5-point finite-difference approximation is of $O(h^2)$ in both examples. The truncation error of the 13-point formula has the same order as used in Example 7.3.3, and the appropriate ratios, for $h_1=1/6$, $h_2=1/12$, for the problem (7.3.27) is 4.07 whilst for the problem (7.3.28) it is 4.01. Moreover, for the former problem $\tau(1/26)=1.13\%$ and for the latter problem $\tau(1/26)=0.98\%$ as given in Bauer and Reiss (1972), in which both may be expected for $h=1/12$, 5.30% and 4.60% respectively, whilst the corresponding results obtained by FIRMI for the same mesh are 5.29% and 4.69%.

In Example 7.3.2(iii), the two solutions obtained for $h=1/4$, $1/8$ coincide with one decimal place at least (see Table 7.4.3). Also, in this example the 9-point formula where truncation error is of $O(h^4)$ was used, and since no exact solution is given, the only check possible was made by substituting the obtained results in the finite difference equations which were all satisfied to quite good accuracy. Since the 9-point formula for the Laplace equation has the truncation error of $O(h^6)$ (see Section 3.5, equation (3.5.23)), then in Example 7.3.1(ii) using this formula the obtained

values for $h=1/6$ or $h=1/12$ to at least 3 and 5 decimal places respectively with the analytical solution (see Table 7.4.4).

It should be stressed in referring to the *inner* linear systems involved in the rectangular factorisation, i.e. the systems (4.4.45a), (5.2.23) and (5.4.22) which relate to FIRM1, FIRM2 and FIRM3 respectively, show no problem arises in solving them. In particular, their coefficient matrices (for the above tested examples and the eigenproblems mentioned below) possess a property such that the diagonal elements have the largest values in modulus.

Finally, the BANDSOLVER (FIRM1) has been applied to eigenproblems which involve block matrices, and has showed remarkable results in comparison to its application in Section 6.5, in a sense that no restrictions or modifications for the considered matrix are required. Moreover, the ordinary shifting, $A - \rho I$, $\rho \geq 0$ is applicable here with suitable values ρ . For example, using the ^{IPM} ~~IMP~~ procedure (6.5.1) for the matrix (a) A_1 of Example 7.2.1(i) and (b) A_1 of Example 7.2.1(iv) where eigenvalues are given by the relations (7.2.1) and (7.2.2) respectively, the related numerical results are: for (a) where the matrix A_1 is 64×64 and its submatrices are of order 4, at step $s=35$ (see procedure (6.5.1), and $s=16$ for $\rho=0$ and $\rho=0.3$ respectively, $\lambda_{\min} = 0.416019810, 0.416019811$ which agree up to 8,9 significant figures with λ_{11} of (7.2.1) (noting that the appropriate factorisation is achieved in 30,62 iterations respectively to yield $\epsilon_f = 0(10^{-12})$, see Section 6.1). And for (b) where A_1 is (100×100) and its submatrices are of order 10, for $\rho=0$, only $s=13$ iterations were required to give $\lambda_{\min} = 0.96560535201$ which coincide up to 8 with λ_{11} given by (7.2.2) (noting that $\epsilon_f = 0(10^{-12})$ is obtained in 64 iterations).

i	$\ell_{i,1}$	$\ell_{i,2}$	$\ell_{i,3}$	$\ell_{i,4}$	$\ell_{i,5}$	$\ell_{i,6}$	$\ell_{i,7}$	d_i
1	-3.16×10^{-1}	-3.54×10^{-1}	-3.55×10^{-1}	-3.52×10^{-1}	-3.41×10^{-1}	-2.99×10^{-1}	-2.03×10^{-3}	3.31
2	-1.55×10^{-2}	-1.89×10^{-2}	-1.91×10^{-2}	-1.70×10^{-2}	-1.13×10^{-2}	-4.75×10^{-3}	-4.05×10^{-3}	3.25
3	-7.09×10^{-3}	-8.62×10^{-3}	-7.95×10^{-3}	-5.11×10^{-3}	-9.40×10^{-3}	-9.78×10^{-3}	-7.68×10^{-3}	3.24
4	-3.54×10^{-4}	-3.86×10^{-3}	-2.54×10^{-3}	-1.89×10^{-2}	-2.04×10^{-2}	-1.97×10^{-2}	-1.55×10^{-2}	3.24
5	-1.61×10^{-3}	-1.28×10^{-3}	-4.16×10^{-2}	-4.45×10^{-2}	-4.45×10^{-2}	-4.24×10^{-2}	-3.48×10^{-2}	3.26
6	-5.48×10^{-4}	-1.04×10^{-1}	-1.09×10^{-1}	-1.10×10^{-1}	-1.09×10^{-1}	-1.05×10^{-1}	-8.98×10^{-2}	3.33
7	-3.02×10^{-1}	-3.08×10^{-1}	-3.09×10^{-1}	-3.09×10^{-1}	-3.07×10^{-1}	-3.01×10^{-1}	-2.70×10^{-1}	3.70

TABLE 7.4.1: (Example 7.3.1(i))

	u_1	u_2	u_3	u_4	u_5	u_6	u_7
a	0.353006809	0.913176676	2.010311261	4.295717748	9.153168404	19.663176677	43.210149666
b	0.353006808	0.913176673	2.010311258	4.295717744	9.153168400	19.663176674	43.210149665
c	0.3530	0.9132	2.0103	4.2957	9.1531	19.6631	43.2101
	u_8	u_9	u_{10}	u_{11}	u_{12}	u_{13}	u_{14}^*
a	0.498850563	1.289388634	2.832350620	6.019391327	12.653779192	26.289388634	53.3177421992
b	0.498850561	1.289388631	2.832350616	6.019391322	12.653779188	26.289388631	53.177421990
c	0.4988	1.2894	2.8323	6.0193	12.6537	26.2893	53.1774

TABLE 7.4.2

a - The solution obtained by BANDSOLVER (FIRM1)
b - " " " " BLOCKSOLVER (FIRM2)
c - " " is given in the indicated reference

* $u_{15} = u_1, u_{16} = u_2, \dots, u_{21} = u_7$

3/4	0.0000562995	0.0000558657	0.0000543862	0.0000514268	0.0000465857	a
	0.0000562995	0.0000558657	0.0000543862	0.0000514268	0.0000465857	b
	0.000090	0.000090	0.000087	0.000082	0.00075	c
1/2	0.0002403647	0.0002387827	0.0002331488	0.0002212026	0.0002006908	a
	0.0002403647	0.0002387827	0.0002331488	0.0002212026	0.0002006908	b
	0.00036	0.00036	0.00035	0.00033	0.00030	c
1/4	0.0009691313	0.0009641157	0.0009450124	0.0009007474	0.0008190410	a
	0.0009691313	0.0009641157	0.0009450124	0.0009007474	0.0008190410	b
	0.00137	0.00136	0.00133	0.00127	0.00115	c
0	0.0038902621	0.0038761131	0.0038166768	0.0036599547	0.0033383168	a
	0.0038902621	0.0038761131	0.0038166768	0.0036599547	0.0033383168	b
	0.00513	0.00511	0.00503	0.00482	0.00439	c
-1/4	0.0155959289	0.0155626382	0.0154018550	0.014880837	0.0136429355	a
	0.0155959289	0.0155626382	0.0154018556	0.0148880833	0.0136429355	b
	0.01920	0.01916	0.01895	0.01831	0.01677	c
-1/2	0.0624606771	0.0624031560	0.0620691542	0.0606442325	0.0560123561	a
	0.0624606770	0.0624031560	0.0670691542	0.0606442325	0.0560122561	b
	0.07175	0.07168	0.07130	0.06968	0.06439	c
-3/4	0.2499687334	0.2499134754	0.2495246098	0.247068277	0.2321389585	a
	0.2499687334	0.2499134753	0.2495246097	0.247068276	0.2321389584	b
	0.26791	0.26785	0.26745	0.26514	0.25006	c
$y_i \backslash x_i$	0	1/4	1/2	3/4	1	

a - Solution of the system (7.3.14) for the 9-point formulae, obtained by

BLOCKSOLVER(FIRM2) for $h=\frac{1}{8}$

b - Solution of the system (7.3.14) for the 9-point formulae, obtained by

BANDSOLVER(FIRM1) for $h=\frac{1}{8}$

c - Solution of the system (7.3.14) for the 9-point formulae, obtained by

FIRM1 or FIRM2, for $h=\frac{1}{4}$

TABLE 7.4.3 (Example 7.3.2(iii))

5/6	0.006123393	0.010599873	0.012236138	0.010599873	0.006123393	a
	0.006123269	0.010599666	0.012235904	0.010599666	0.006123269	b
	0.006121436	0.010596554	0.012232351	0.01096554	0.006121436	c
4/6	0.013973997	0.024172722	0.027894559	0.024172722	0.013973997	a
	0.013973699	0.024172249	0.027894038	0.024172249	0.013973699	b
	0.013969347	0.024165125	0.027886084	0.024165125	0.013969347	c
3/6	0.025793079	0.044524883	0.051328646	0.044524882	0.025793079	a
	0.025792445	0.044524016	0.051327760	0.044524016	0.025792445	b
	0.025783326	0.044510830	0.051314127	0.044510830	0.025783326	c
2/6	0.045085576	0.077357653	0.088932984	0.077357653	0.045085876	a
	0.045084311	0.077356220	0.088931771	0.077356220	0.045084311	b
	0.045061685	0.077333954	0.088913354	0.077333954	0.045061685	c
1/6	0.078205588	0.131793333	0.150461130	0.131793333	0.078205588	a
	0.078200647	0.131791677	0.150460097	0.131791677	0.078200647	b
	0.078121486	0.131767811	0.150445413	0.131767811	0.078121486	c
y_i / x_i	1/6	2/6	3/6	4/6	5/6	

a - Analytical solution obtained from (7.3.6)

b - Solution of (7.3.9) using the 9-point formula and solved by FIRM1
(with $h=\frac{1}{12}$)

c - Solution of (7.3.9) using the 9-point formula and solved by FIRM1
(with $h=\frac{1}{6}$)

TABLE 7.4.4: (Example 7.3.1(ii) Solution)

CHAPTER 8

CONCLUSIVE REMARKS AND FURTHER INVESTIGATIONS

PART (A)

The main conclusions associated with the new algorithmic solvers for the banded linear systems of equations are:

1. The factorisation involved in the pointwise procedures (i.e. the BANDSOLVERS) differ from the conventional and well-known direct techniques (such as LU-Decomposition) in a sense that the elements of the factor matrices are obtained by an iterative scheme.
2. In relation to the preceding point, the solution of a special set of non-linear equations derived from equating the elements of the factor matrices' product to the corresponding elements of the original matrix, is accomplished iteratively (precisely, by adopting the so-called Gauss-Seidel-Newton scheme (sub-section 4.3.6)).
3. The factorisation techniques involved in the block-case (i.e. in the BLOCKSOLVERS), explicitly seem to be a direct procedure to obtaining the sub-matrices of the block-factor matrices. Implicitly, an iterative process (precisely, the Newton's method) is involved to compute the required matrix square roots. In this respect, it should be emphasised that computing the matrix square root as accurate as possible is essential, since it is a vital step in the factorisation procedure.
4. The factor matrices possess the property of preserving the structure of the original matrix (the bandwidth and the sparsity (if any)), i.e. no 'fill-in' is created beyond the outer off-diagonals of the given matrix. Although this is true for the BANDSOLVERS when they are applied to the block matrices, but the factor matrices produced, in contrast to the original matrix, their elements in-between the diagonal and the outer off-diagonal (which

correspond to the zero elements of the original matrix) do 'fill-in' with small numbers.

5. The formulation of the BANDSOLVERs raises the question of adopting High-Order Discretization techniques for the second-order non-linear (or linear) 2-point boundary value problems. This seems to be practical for these problems under periodic conditions, in particular with the existence of the new algorithms (i.e. FICM1 and FICM2)
6. Generally speaking, both the BANDSOLVER (FIRM1) and the BLOCKSOLVERs have shown quite satisfactory results from their application to partial differential equations (Chapter 7). Also, the applications of the former have been extended to eigenproblems (Section 6.5, 7.4); whilst the latter in the light of factorisation results is believed to be applicable to eigenproblems associated with block-matrices (as the types given in Chapter 7).
7. The BANDSOLVERs and BLOCKSOLVERs (in Chapters 4,5 respectively) associated with skew-matrices involve mainly a modification which transfers the considered skew-matrix to a symmetric one and then the solution procedure is pursued as in the relevant solver for a symmetric matrix.
8. Will the new algorithms (or some of them) have a superiority or are they competitive with other methods? In fact, it is inadequate to judge the question on abstract grounds alone, just on the basis of the obtained results from the tested numerical examples or other 'artificial' examples. The answer, however, is connected with the other factors, such as the amount of storage required and the running-time which both for programming reasons have not been considered, included or measured in this work. In particular,

by noting that all the solvers' procedures have been programmed in a generalised form which consequently requires a considerable storage area, notably this can be reduced considerably for certain cases where the given matrix possesses a special structure.

Finally, the new factorisation techniques which include, obtaining pseudo-inverse rectangular matrices iteratively, matrix square roots, etc., may require further theoretical justification and the continuation of further related studies may take place in the light of the points outlined in part (B) below.

PART (B)

It is suggested that the following points be considered for further investigation:

1. The solution of the set of non-linear systems of equations involved in the BANDSOLVERS may be obtained by a direct solution or strategies to improve the rate of the convergence of the present iterative scheme, or to use other faster alternatives.
2. Referring to point 4 of part (A), it may be interesting to pursue improving the BANDSOLVER (FIRM1) when it is applied to block-matrix systems so that the indicated 'fill-in' can be alleviated or overcome.
3. In the Biharmonic equation (Example 7.3.3, Chapter 7) in order to apply a 25-point finite difference formula, it is suggested that the appropriate BLOCKSOLVER (FIRM3) may be extended to a septa-diagonal block solver, or probably to matrices of wider bandwidth. This might be proceeded by treating the non-linear matrix equations obtained in an analogous manner to the pointwise non-

linear equations involved in FIRM1 (i.e. GITRM procedure for $r \geq 3$).

4. Can any of the new algorithms be applied on the new parallel processing machines? If they can, this may increase the creditability of the algorithm(s); in particular for those algorithms which take a considerable time in converging to the solution such as the case of the Biharmonic equation using BANDSOLVER (FIRM1) or in other steps of the solution procedure.

REFERENCES

AMES, W.F., (1969), *"Numerical Methods for Partial Differential Equations"*, Nelson.

AUDISH, S.E. (1978), *"Deferred Correction Method for Boundary Value Problems"*, a dissertation submitted for the degree of M.Sc., University of Liverpool.

AUDISH, S.E., & EVANS, D.J. (1980), *"General Methods for Solving a System of Special Symmetric Periodic Matrix"*, Computer Studies 86, Internal Report, Dept. of Computer Studies, Loughborough University of Technology.

AZIZ, A.K., (1975), *"Numerical Solutions of Boundary-Value Problems for Ordinary Differential Equations"*, Academic Press, New York.

BATHE, K.J. & WILSON, E.L. (1976), *"Numerical Methods in Finite Element Analysis"*, Prentice-Hall, Inc.

BAUER, L. and REISS, E.L., (1972), *"Block Five Diagonal Matrices and the Fast Numerical Solution of the Biharmonic Equation"*, Math. Comp. 26, No. 118, pp. 311-321.

BERG, L., (1981), *"On Real Factorizations of Symmetric Circulant Sparse Matrices"*, Computing, 26, pp. 265-270.

BERMAN, A. & PLEMMONS, R.J., (1979), *"Non-negative Matrices in the Mathematical Sciences"*, Academic Press.

BJORCK, A. & PEREYRA, V., (1970), *"Solution of Vandermonde Systems of Equations"*, Math. Comp. 24, pp. 893-903.

BLANCH, G. (1964), *"Numerical Evaluation of Continued Fraction"*, SIAM Rev. Vol. 6, No. 4, pp. 383-419.

BRIGHAM, E.O. (1974), *"The Fast Fourier Transform"*, Prentice-Hall Inc.

BROYDEN, C.G., (1975), *"Basic Matrices"*, M.

BROWN, R.R., (1962), *"Numerical Solution of Boundary Value Problems Using Non-uniform Grids"*, J.Soc.Indust.Appl.Math., Vol.10, No.3, pp.475-495.

BUCKLEY, A., (1977), *"On the Solution of Certain Skew Symmetric Linear Systems"*, SIAM J.Numer.Anal., Vol.14, No.3, pp.566-570.

BUDAK, B.M., SAMASKII, A.A. and TIKHONOV, A.N., (1964), *"A Collection of Problems on Mathematical Physics"*, (Translated by Brink, D.M.),

Pergamon Press.

BUZBEE, B.L., GOLUB, G.H. and NIELSON, C.W., (1970), *"On Direct Methods for Solving Poisson's Equations"*, SIAM J.Numer. Anal., Vol.7, No.4, pp.627-656.

CONRAD, V. and WALLACH, Y., (1979), *"Alternating Methods for Sets of Linear Equations"*, Num.Math., 32, pp.105-108.

and

CONTE DE BOOR, (1972), *"Elementary Numerical Analysis"*, McGraw-Hill.

COOLEY, J.W., & TUKEY, J.W., (1965), *"An Algorithm for Machine Calculation of Complex Fourier Series"*, Math.Computation, Vol.19, pp.279-301.

CUTHILL, E.H. & VARGA, R.S. (1959), *"A Method of Normalized Block Iteration"*, J.A.C.M., Vol.6, pp.236-244.

DAHLQUIST, G. and BJORCK, A., (1969), *"Numerical Methods"*, Prentice-Hall Inc.

DANIEL, J.W. & MARTIN, A.J. (1977), *"Numerov's Method with Deferred Correction for Two-Point Boundary-Value Problems"*, SIAM J.Numer.Anal., Vol. 14, No.6, pp.1033-1050.

DEMIDOVICH, B.P. & MARON, I.A., (1976), *"Computational Mathematics"*, MIR Publishers, Moscow.

- EVANS, D.J., (1971), *"Numerical Solution of the Fourth Boundary Value Problem for Parabolic Differential Equations"*, J.Inst.Maths.Applics., Vol.7, pp.61-75.
- EVANS, D.J. (1972), *"An Algorithm for the Solution of Certain Tridiagonal Systems of Linear Equations"*, Comp.J., Vol.15, No.4, pp.356-359.
- EVANS, D.J., (1973), *"The Solution of Certain Systems of Linear Equations Occurring in Periodic Problems by Cyclic Factorization"*, Unpublished Manuscript.
- EVANS, D.J., (Ed.), (1974), *"Iterative Sparse Matrix Algorithms"*, in Software for Numerical Mathematics, Academic Press, London.
- EVANS, D.J., (1977), *"On the Use of Fast Methods for Solving Boundary Value Problems"*, Comp.J., Vol.20, pp.181-184.
- EVANS, D.J., (1979), *"Direct Methods of Solution of Partial Differential Equations with Periodic Boundary Conditions"*, Math.&Comp.Sim. XXI, pp.270-275.
- EVANS, D.J., & HADJIDIMOS, A., (1979), *"On the Factorization of Special Symmetric Parabolic and Non-Periodic Quindagonal Matrices"*, Computing 21, pp.259-266.
- EVANS, D.J. & OKOLIE, S.O. (1979), *"A Generalised Sparse Factorization Method for the Solution of Periodic Tridiagonal Systems"*, Comp. & Maths. with Applics., Vol.5, pp.211-216.
- EVANS, D.J. (1980), *"On the Solution of Certain Toeplitz Tridiagonal Linear Systems"*, SIAM J.Numer.Anal. Vol.17, No.5, pp.675-680.

- FADDEEVA, V.N. (1959), *"Computational Methods of Linear Algebra"*,
(translated from the Russian), Dover Publications Inc., New York.
- FENNER, R.T., (1974), *"Computing for Engineers"*, Macmillan.
- FENNER, R.T. (1975), *"Finite Element Methods for Engineers"*, Macmillan.
- FOX, L., (1957), *"The Numerical Solution of Two-Point Boundary Problems in Ordinary Differential Equations"*, Oxford University Press.
- FOX, L., (1962), *"Numerical Solution of Ordinary and Partial Differential Equations"*, Pergamon Press.
- FOX, L., (1964), *"An Introduction to Numerical Linear Algebra"*, Oxford University Press.
- FOX, L. & MAYERS, D.F. (1977), *"Computing Methods for Scientists and Engineers"*, Clarendon Press.
- FRANK, E., (1962), *"On Continued Fraction Expansions for Binomial Quadratic Grids"*, Numerische Mathematik, 4, pp.303-309.
- FROBERG, C.E., (1974), *"Introduction to Numerical Analysis"*, Addison-Wesley.
- GAWAIN, T.H., & BALL, R.E., (1977), *"Improved Finite Difference Formulas for Boundary Value Problems"*, Tech.Report NP567Gn77051A, Naval Postgraduate School, Monterey, C.A.
- GAWAIN, T.H., & BALL, R.E., (1978), *"Improved Finite Difference Formulas for Boundary Value Problems"*, Int.J.Num.Meth. Engng., Vol. 12, pp.1151-1160.
- GERALD, C.F., (1970), *"Applied Numerical Analysis"*, Addison-Wesley.

GERALD, C.F., (1978), *"Applied Numerical Analysis"*, Addison Wesley.

GOULT, R.J., HOKINS, R.F., MILNER, J.A., PRATT, M.J., (1974),
"Computational Methods in Linear Algebra", Stanley Thames.

GREGORY, R.T. & KARNEY, D.L., (1969), *"A Collection of Matrices for Testing Computational Algorithms"*, Wiley Interscience.

GUSTAFSSON, I., (1979), *"On Modified Incomplete Cholesky Factorization Methods for the Solution of Problems with Mixed Boundary Conditions and Problems with Discontinuous Notional Coefficients"*, Int.J.Num. Meth.Engng., Vol.14, pp.1127-1140.

HALL, G. & WATT, J.M.(Ed.), (1976), *"Modern Numerical Methods for Ordinary Differential Equations"*, Oxford University Press.

HENDERSON, D.S. & WASSYNG, A. (1978), *"A New Method for the Solution of $Ax=b$ "*, Num.Math., 29, pp.287-289.

HENRICI, P., (1964), *"Elements of Numerical Analysis"*, John Wiley.

HENRICI, P., (1962), *"Discrete Variable Methods in Ordinary Differential Equations"*, Wiley.

HOCKNEY, R.W., (1965), *"A Fast Direct Solution of Poisson's Equation Using Fourier Analysis"*, J.Ass.Comp.Mach., Vol.12, No.1, pp.95-113.

HOHN, F., (1973), *"Elementary Matrix Algebra"*, Macmillan Co., New York.

ISAACSON, E. & KELLER, H.B., (1966), *"Analysis of Numerical Methods"*, Wiley.

JENNINGS, W., (1964), *"First Course in Numerical Methods"*, Macmillan Co., New York.

KELLER, H.B., (1968), *"Numerical Methods for Two-Point Boundary-Value Problems"*, Blaisdell.

KELLER, H.B., (1975), *"Numerical Solutions of Boundary Value Problems for Ordinary Differential Equations: Survey and Some Recent Results on Difference Methods"*, in AZIZ (1975), pp.27-88.

KING, H.H., (1976), *"A Poisson Equation Solver for Rectangular or Annular Regions"*, Int.J.Num.Meth.Engng., Vol.10, pp.799-807.

KOLMOGOROV, A.N., FOMIN, S.V., (1970), *"Introductory Real Analysis"*, (Trans. by R.A. Silverman), Dover, New York.

KUNZ, K.S., (1957), *"Numerical Analysis"*, McGraw-Hill.

LAASONEN, P., (1958), *"On the Iterative Solution of the Matrix Equation $Ax^2 - I = 0$ "*, Math.Tables Aids Comp., Vol.12, pp.109-116.

LANCASTER, P., (1969), *"Theory of Matrices"*, Academic Press.

LIEBERSTEIN, H.M., (1968), *"A Course in Numerical Analysis"*, Harper & Row.

MACNAGHTEN, A.M. & HOARE, C.A.R., (1977), *"Fast Fourier Transform Free From Tears"*, Comp.J., Vol.20, pp.78-83.

MARTIN, H.C. & CAREY, G.F., (1973), *"Introduction to Finite Element Analysis"*, Theory and Application, TATA McGraw-Hill.

MEIJERINK, J.A. & VAN DER VORST, H.A., (1977), *"An Iterative Solution Method for Linear Systems of Which the Coefficient Matrix is a Symmetric M-Matrix"*, Math.Comp., Vol.31, No.137, pp.148-162.

- MEYER, C.D., Jr., PLEMMONS, R.J., (1977), *"Convergent Powers of a Matrix with Applications to Iterative Methods for Singular Linear Systems"*, SIAM J.Numer.Anal., 14, pp.699-705.
- MEYER, G.H. (1973), *"Initial Value Methods for Boundary Value Problems"*, Academic Press.
- MITCHELL, A.R. & WAIT, R., (1977), *"The Finite Element Method in Partial Differential Equations"*, Wiley.
- NEUMANN, M. & PLEMMONS, R.J., (1978), *"Convergent Non-negative Matrices and Iterative Methods for Consistent Linear Systems"*, Numer.Math. 31, pp.265-279.
- NOBLE, B., (1969), *"Applied Linear Algebra"*, Prentice-Hall Inc.
- OKOLIE, S.O. (1978), *"The Numerical Solution of Sparse Matrix Equations by Fast Methods and Associated Computational Techniques"*, Ph.D. Thesis Loughborough University of Technology.
- ORTEGA, J.M. & RHEINBOLDT, W.C., (1970), *"Iterative Solution of Non-linear Equations in Several Variables"*, Academic Press.
- OSBORNE, M.R., (1969), *"On Shooting Methods for Boundary Value Problems"*, J.Math.An.&Appl. 27, pp.417-433.
- OSTROWSKI, A.M., (1966), *"Solution of Equations and Systems of Equations"*, Academic Press.
- PASSAS, S., (1979), *"Data Transmission using Multi-level Quadrature Amplitude Modulation"*, Dept. Electronic and Electrical Engineering, Loughborough University.

PEREYRA, V., (1966), *"On Improving an Approximate Solution of a Functional Equation by Deferred Correction"*, Numer.Math., 8, pp.376-391.

PEREYRA, V., (1967), *"Iterated Deferred Correction for Non-Linear Operator Equations"*, Numer.Math., 10, pp.316-323.

PEREYRA, V., (1968), *"Iterated Deferred Correction for Non-linear Boundary Value Problems"*, Numer.Math. 11, pp.111-125.

PEREYRA, V. (1973), *"High Order Finite Difference Solution of Differential Equations"*, Computer Science Dept. Rep. STAN-CS-73-348, Stanford University, Stanford, C.A.

RALSTON, A., (1965), *"A First Course in Numerical Analysis"*, McGraw-Hill Kogakusha.

RICE, J.R., HOUSTIS, E.M. & DYKSEN, W.R., (1980), *"A Population of Linear 2nd Order Elliptic Partial Differential Equations on Rectangular Domains - Part I"*, M.R.C. Report 2078, Math. Research Centre, University of Wisconsin.

SALVADORI, M.G. & BARON, M.L., (1955), *"Numerical Methods in Engineering"*, Prentice-Hall.

SCARBOROUGH, J.B., (1955), *"Numerical Mathematical Analysis"*, Oxford University Press.

SCOFIELD, D.F., (1973), *"A Note of Lowdin Orthogonalization and the Square Root of a Positive Self-Adjoint Matrix"*, Inter.J. of Quantum Chem., Vol. VII, pp.561-568.

- SHOOSMITH, J.N., (1973), "*A Study of Monotone Matrices with an Application of a Linear, Two-Point Boundary-Value Problem*", A Dissertation for Ph.D. (University of Virginia).
- SHOOSMITH, J.N., (1975), "*A High-Order Finite-Difference Method for the Solution of Two-Point Boundary-Value Problems on a Uniform Mesh*", in AZIZ (1975), pp.355-369.
- SMITH, G.D. (1978), "*Numerical Solution of Partial Differential Equations: Finite Difference Methods*", 2nd Edition, Oxford University Press.
- SPATH, H., (1967), "*Algorithm 298: Determination of the Square Root of a Positive Definite Matrix [F1]*", Com. A.C.M., Vol.10, No.3, pp.182.
- STRANG, G., (1976), "*Linear Algebra and Its Application*", Academic Press.
- SWEET, R.A. (1974), "*A Generalized Cyclic Reduction Algorithm*", SIAM J. Numer.Anal., Vol.11, No.3, pp.506-510.
- SWEET, R., (1977), "*A Cyclic Reduction Algorithm for Solving Block Tri-diagonal Systems of Arbitrary Dimension*", SIAM, Vol.14, No.4.
- SZIDAROVSKY, F. and YAKOWITZ, S., (1978), "*Principles and Procedures of Numerical Analysis*", Plenum Press.
- VARAH, J.M. (1972), "*On the Solution of Block-Tridiagonal Systems Arising from Certain Finite-Difference Equations*", Math.Comp., Vol.26, No.120, pp.859-868.
- VARGA, R.S. (1962), "*Matrix Iterative Analysis*", Prentice-Hall, Englewood Cliff.

WALL, H.S., (1948), *"Analysis Theory of Continued Fractions"*, D. Van Nostrand Co.

WENDROFF, B., (1966), *"Theoretical Numerical Analysis"*, Academic Press Inc., New York.

WILKINSON, J.H., (1955), *"The Use of Iterative Methods of Finding the Latent Roots and Vectors of Matrices"*, MTAC, Vol.9, pp.184-191.

WILKINSON, J.H., (1961), *"Error Analysis of Direct Methods of Matrix Inversion"*, J.Assoc.Comput.Mach., Vol.8, pp.281-330.

WILKINSON, J.H., (1963), *"Rounding Errors in Algebraic Processes"*, H.M.S.O.

WILLIAMS, P.W., (1973), *"Numerical Computation"*, Nelson.

WOOD, W.L., (1971), *"Periodicity Effects on the Iterative Solution of Elliptic Difference Equations"*, SIAM J.Numer.Anal., Vol.8, No.2.

YOUNG, D.M., (1954), *"Iterative Methods for Solving Partial Differential Equations of Elliptic Type"*, Trans.Amer.Math.Soc., 76, pp.92-111.

APPENDIX A

We shall rewrite the system (4.2.4) in the form given in Berg (1981).

For this, we assume $n=r+1$, $r \geq 1$ (as defined in Section 4.2), and define x_i, a_i , $i=1, \dots, n$ as follows,

$$\left. \begin{aligned} \alpha_{i-1} &= x_i \\ c_i &= a_i \end{aligned} \right\} \quad i=1, 2, \dots, n, \quad (\text{A.1})$$

where α_{i-1} , c_{i-1} , $i=1, 2, \dots, n$ are the elements of the system (4.2.4).

From (A.1) and taking $a_1=1$ (Berg (1981)) the system (4.2.4) becomes,

$$\left. \begin{aligned} x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2 &= 1 \\ x_1 x_2 + x_2 x_3 + \dots + x_{n-1} x_n &= a_2 \\ \hline x_1 x_{n-1} + x_2 x_n &= a_{n-1} \\ x_1 x_n &= a_n \end{aligned} \right\} \quad (\text{A.2})$$

The direct method suggested by Berg (1981) to solve the system (A.2) to yield real solutions is discussed below.

For the special case, if the values a_2, a_3, \dots, a_n satisfy the following equality, i.e.,

$$a_{k+1} = \frac{1}{n+1} \left[(n-k) \cos \frac{k\pi}{n+1} + \frac{\sin \frac{k+1}{n+1} \pi}{\sin \frac{\pi}{n+1}} \right], \quad k=1, 2, \dots, n-1, \quad (\text{A.3})$$

and

$$a_2 > 0, \quad (\text{A.4})$$

then the solution of the system (A.2), i.e. x_1, \dots, x_n ($x_1 > 0$) is uniquely determined and has the form,

$$x_i = \sqrt{\frac{2}{n+1}} \sin \frac{i\pi}{n+1}, \quad i=1, 2, \dots, N. \quad (\text{A.5})$$

(N.B. if a_2 in (A.4) is negative, then a_t , $t=3, \dots, n$ in (A.3) should be multiplied by $(-1)^{t-1}$ and hence x_i , $i=1, 2, \dots, n$ in (A.5) is multiplied by $(-1)^{i-1}$ as well).

The derivation of (A.5) and (A.3) is as follows.

The method of Lagrange is used to determine the extreme value of a_2 under the first equation of (A.2) as a condition; thus from the first and

the second equations of (A.2) we may write,

$$F \equiv x_1 x_2 + x_2 x_3 + \dots + x_{n-1} x_n - \lambda (x_1^2 + x_2^2 + \dots + x_n^2) .$$

Now to determine the stationary values of F we start with differentiating F with respect to x_1, x_2, \dots, x_n successively to obtain the result

$$\sum_{t=1}^n \frac{\partial F}{\partial x_t} = x_2 + (x_1 + x_3) + (x_2 + x_4) + \dots + (x_{n-2} + x_n) + x_{n-1} - 2\lambda(x_1 + x_2 + \dots + x_n) = 0, \quad (\text{A.6'})$$

if we now define

$$x_0 = x_{n+1} = 0, \quad (\text{A.6''})$$

then (A.6'') can be written as a single difference equation, i.e.,

$$x_{t+1} + x_{t-1} - 2\lambda x_t = 0, \quad t=1, \dots, n. \quad (\text{A.7})$$

Berg (1981) indicates that this eigenvalue problem possesses n solutions. These can be determined in the following manner.

Let
$$x_t = A\theta^t \quad (A \text{ is a constant}), \quad (\text{A.8})$$

and substitute in (A.7) to yield the result

$$\theta^2 - 2\lambda\theta + 1 = 0, \quad (\text{A.9})$$

The quadratic equation (A.9) possesses two roots, θ_1, θ_2 (say); hence from (A.8) we have the general solution,

$$x_t = A_1 \theta_1^t + A_2 \theta_2^t, \quad (\text{A.10})$$

where A_1 and A_2 are constants.

Consequently from (A.10) and conditions (A.6'') we obtain the relations

$$0 = A_1 + A_2$$

and
$$0 = A_1 \theta_1^{n+1} + A_2 \theta_2^{n+1} .$$

From these two equations we have

$$0 = A_1 (\theta_1^{n+1} - \theta_2^{n+1})$$

or
$$\left(\frac{\theta_1}{\theta_2}\right)^{n+1} = 1 = e^{i2j\pi}, \quad \text{where } j=1, 2, \dots, n, \quad i=\sqrt{-1}. \quad (\text{A.11})$$

Therefore, from (A.11) we obtain

$$\frac{\theta_1}{\theta_2} = e^{i2j\pi/(n+1)} \quad (\text{A.12})$$

and from the quadratic equation (A.9) we have

$$\theta_1 \theta_2 = 1 \quad (\text{A.13})$$

$$\text{and} \quad \theta_1 + \theta_2 = 2\lambda. \quad (\text{A.14})$$

Hence from (A.12) and (A.13) we have

$$\text{and} \quad \left. \begin{aligned} \theta_1 &= e^{ij\pi/(n+1)} \\ \theta_2 &= e^{-ij\pi/(n+1)} \end{aligned} \right\} \begin{aligned} j &= 1, 2, \dots, n \\ i &= \sqrt{-1} \end{aligned} \quad (\text{A.15})$$

Then, from (A.14) and (A.15) λ can be expressed as

$$\begin{aligned} \lambda &= \frac{1}{2}(\theta_1 + \theta_2) = \frac{1}{2}(e^{ij\pi/(n+1)} + e^{-ij\pi/(n+1)}) \\ &= \frac{1}{2}(\cos \frac{j\pi}{n+1} + i \sin \frac{j\pi}{n+1} + \cos \frac{j\pi}{n+1} - i \sin \frac{j\pi}{n+1}) \end{aligned}$$

hence the n eigenvalues are given by

$$\lambda_j = \cos \frac{j\pi}{n+1}, \quad j=1, 2, \dots, n,$$

whilst the s^{th} component of the eigenvector (corresponding to the j^{th} eigenvalue) is given by

$$\begin{aligned} x_s^{(j)} &= A_1 \theta_1^s + A_2 \theta_2^s \\ &= A_1 (e^{isj\pi/(n+1)} - e^{-isj\pi/(n+1)}) \\ &= 2iA_1 \sin \frac{js\pi}{n+1}, \quad s=1, 2, \dots, n. \end{aligned} \quad (\text{A.16})$$

$$\text{Now since,} \quad \sum_{s=1}^n \sin^2 \frac{js\pi}{n+1} = \frac{n+1}{2}, \quad (\text{A.17})$$

then by substituting $x_s^{(j)}$ into the first equation of (A.2) we have

$$(2iA_1)^2 \sum_{s=1}^n \sin^2 \frac{js\pi}{n+1} = 1$$

which yields the result by virtue of (A.17),

$$\begin{aligned} (2iA_1)^2 &= \frac{2}{n+1} \\ \text{or} \quad (2iA_1) &= \sqrt{\frac{2}{n+1}} \end{aligned}$$

Hence substitution this result in (A.16) we obtain

$$x_s^{(j)} = \sqrt{\frac{2}{n+1}} \sin \frac{js\pi}{n+1}, \quad s=1, 2, \dots, n. \quad (\text{A.18})$$

Further, if we substitute $x_s^{(j)}$ given by (A.18) into the second, the third etc. up to the n^{th} equation of (A.2) and apply the following formula,

$$\sum_{s=1}^{n-k} \sin s x \sin(s+k)x = \frac{n-k}{2} \cos kx - \frac{\cos(n+1)x \sin(n-k)x}{2 \sin x}$$

we obtain for $x=j\pi/(n+1)$ the following relations,

$$a_{k+1} = \frac{1}{n+1} \left[(n-k) \cos \frac{kj\pi}{n+1} + \frac{\sin \frac{k+1}{n+1} j\pi}{\sin \frac{j\pi}{n+1}} \right], \quad (\text{A.19})$$

Thus, by setting $j=1$, the relations (A.18) and (A.19) implies (A.5) and (A.3) respectively.

On the other hand, for $k=1$ (A.19) yields the result,

$$a_2 = \cos \frac{j\pi}{n+1}, \quad j=1,2,\dots,n. \quad (\text{A.20})$$

Since the maximum of a_2 in (A.20) is at $j=1$, then Berg (1981) concludes that the necessary condition for solving (A.2) in the real domain is

$$|a_2| \leq \cos \frac{\pi}{n+1}. \quad (\text{A.21})$$

Further, he points out that for the remaining values of a_k , $k=3,4,\dots,n$, it is necessary that,

$$|a_{k+1}| \leq \cos \frac{\pi}{N+2} \quad (\text{A.22})$$

with $N = [\frac{n-1}{k}]$ for $k \geq 2$ so that (A.2) is solvable in a real domain. He adds also that both (A.21) and (A.22) are not sufficient for the real solvability of (A.2).

However, for the case $n=3$ (i.e. the coefficient matrix is circulant and quindagonal Berg (1981) gives the sufficient and necessary conditions, which are equivalent to those given by Evans and Hadjidimos (1979),

$$|a_2| \leq a_3 + \frac{1}{2} \quad \text{for } \frac{1}{2} \leq a_3 \leq \frac{1}{6},$$

$$|a_2| \leq \sqrt{4a_3(1-2a_3)} \quad \text{for } \frac{1}{6} \leq a_3 \leq \frac{1}{2}.$$

By inverting these inequalities we obtain the necessary and sufficient conditions, so that (A.2) possesses real solutions,

$$|a_2| \leq \frac{1}{2} \leq a_3 \leq \frac{1}{4}(1+\sqrt{1-2a_3^2}) \quad \text{for } 0 \leq |a_2| \leq \frac{2}{3},$$

$$\frac{1}{4}(1-\sqrt{1-2a_3^2}) \leq a_3 \leq \frac{1}{4}(1+\sqrt{1-2a_3^2}) \quad \text{for } \frac{2}{3} \leq |a_2| \leq \frac{1}{\sqrt{2}}$$

Thus, the maximum of $|a_2|$ and $|a_3|$ are $\frac{1}{\sqrt{2}}$ and $\frac{1}{2}$ respectively and which coincide with the results obtained from (A.21) and (A.22) (with $N=1$) respectively.

Finally, back to the system (4.2.1), after its normalization, the coefficient matrix will be left with unity values on the diagonal and the non-zero elements become c_i/c_0 , $i=1, \dots, r$. In this case, the factorization (4.2.3a) implies a non-linear system similar to (A.2) (taking the assumption into account). Thus, the conditions (A.21) and (A.22) may be considered as necessary conditions for the iterative method (GITRM, Subsection 4.2.2), to obtain the real solution of (4.2.4a).

APPENDIX B

The algorithm FIRMI for the tridiagonal case, i.e. $r=1$ can be reduced as follows.

From the relations (4.4.30), we have

$$\left. \begin{aligned} g_j &= (-\gamma_{1,j}/\gamma_{0,j+1})g_{1,j+1} \\ \text{with } g_{1,N+1} &= -1, \\ \text{and } \tilde{z}_j &= (-\gamma_{1,j}/\gamma_{0,j+1})\tilde{z}_{j+1} + z_j \\ \text{with } \tilde{z}_{N+1} &= 0. \end{aligned} \right\} j=N, N-1, \dots, 1. \quad (\text{B.1})$$

$$\left. \begin{aligned} \text{and } \tilde{z}_j &= (-\gamma_{1,j}/\gamma_{0,j+1})\tilde{z}_{j+1} + z_j \\ \text{with } \tilde{z}_{N+1} &= 0. \end{aligned} \right\} j=N, N-1, \dots, 1. \quad (\text{B.2})$$

Then, (4.4.32) implies

$$\left. \begin{aligned} y_j &= \frac{1}{\gamma_{0,j}}(\tilde{z}_j - y_{N+1}g_{1,j}) \\ &\equiv \phi_j + \psi_{1,j}y_{N+1} \end{aligned} \right\} j=1, 2, \dots, N. \quad (\text{B.3})$$

The system (4.4.28b) becomes,

$$\begin{bmatrix} 1 & & & & & \\ \alpha_1 & 1 & & & & \\ & \alpha_2 & 1 & & & 0 \\ & & \alpha_3 & \ddots & & \\ & & & \ddots & \ddots & \\ & 0 & & & \alpha_{N-1} & 1 \\ & & & & & \alpha_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad (\text{B.4})$$

(where $\alpha_{0,i}=1$, $i=1, 2, \dots, N$, as in (4.4.29b) and we take $\alpha_i=\alpha_{1,i}$, $i=1, 2, \dots, N$).

From (B.4) we can obtain the following relation

$$\left. \begin{aligned} x_1 &= y_1 \\ x_2 &= y_2 - \alpha_1 x_1 \\ x_j &= y_j - \alpha_{j-1} x_{j-1}, \quad j=2, 3, \dots, N \end{aligned} \right\} \quad (\text{B.5a})$$

$$\text{and } \alpha_N x_N = y_{N+1} \quad (\text{B.5b})$$

Now, from (B.5a) we obtain after substitution,

$$\begin{aligned}
x_1 &= y_1 \\
x_2 &= y_2^{-\alpha_1} x_1 = y_2^{-\alpha_1} y_1 \\
x_3 &= y_3^{-\alpha_2} x_2 = y_3^{-\alpha_2} y_2^{\alpha_2} y_1 \\
&\vdots \\
x_N &= y_N^{-\alpha_{N-1}} y_{N-1}^{\alpha_{N-1}} y_{N-2}^{\alpha_{N-2}} \dots + (-1)^{N+1} \alpha_{N-1} \alpha_{N-2} \dots \alpha_1 y_1
\end{aligned} \tag{B.6}$$

Multiply (B.6) by α_N , then from (B.5b) we have

$$y_{N+1} = \alpha_N x_N = m_N y_N^{-m_{N-1}} y_{N-1}^{m_{N-1}} y_{N-2}^{m_{N-2}} \dots + (-1)^{N+1} m_1 y_1 \tag{B.7}$$

where

$$m_N = \alpha_N$$

$$m_{N-1} = \alpha_N \alpha_{N-1} = \alpha_{N-1} m_N$$

so

$$m_j = \alpha_j m_{j+1}, \quad j=1, 2, \dots, N \text{ (with } m_{N+1}=1) \tag{B.8}$$

If we multiply y_j by m_j in (B.3), then we have

$$m_j y_j = m_j \phi_j + m_j \psi_{1,j} y_{N+1}, \quad j=1, 2, \dots, N.$$

When these are substituted in (B.7) and rearranged then we obtain

$$y_{N+1} = \sum_{j=1}^N m_j \phi_j + \sum_{j=1}^N m_j \psi_{1,j} y_{N+1}$$

or

$$y_{N+1} = \sum_{j=1}^N m_j \phi_j / (1 - \sum_{j=1}^N m_j \psi_{1,j}) \tag{B.9}$$

and hence y_j , $j=1, 2, \dots, N$ can be obtained from (B.3) and the solution vector x_j , $j=1, 2, \dots, N$ from (B.5).

Thus, the amount of work is reduced to the order of: $7N$ multiplications and $4N$ additions (including the normalization of (4.4.3a)); this can be reduced further if the coefficient matrix is constant and symmetric (see the system (4.4.8a)).

APPENDIX C

The attached procedures are written in ALGOL 68 (and the tested programs have been run on the ICL 1904S in the Computer Centre, at Loughborough University). These are:

C1: (Algorithm FICM1)

GENSYM - solves the system (4.2.1), and involves the following steps:

(i) the procedure GITRM to solve the appropriate non-linear equations,

and (ii) procedure BACKFORD to perform the forward and backward substitution schemes.

C2: (Algorithm FICM2)

PRDSYS - solves the system (4.3.1) which includes the relevant factorisation and elimination procedures.

C3: (Algorithm FIRMI)

NONPDUL - performs the factorisation (4.4.2)

BACKFORD - solves the system (4.4.4)

C4: (Algorithm FICM3)

GENSYSBLK - solves the system (5.1.1) and involves two procedures:

(i) GITRMBLK - computes the submatrices Q_0 and Q_1 of the system (5.1.4),

and (ii) BACKFORDBLK - solves the two systems (5.1.9a) and (5.1.9b).

C5: The Iterative Deferred Correction procedure (Chapters 3 and 6) for

2-point boundary value problems with periodic conditions in which either PRDSYS (i.e. FICM2) or GENSYM (i.e. FICM1) is used. The program is an extended and modified form to the one given in Audish (1978).

C1

```

'BEGIN'
'PROC'GITRM=('REF'[]'REAL'C,BETTA)'VOID':
'BEGIN'
  'INT'N='UPB'C;
  [0:N]'REAL'ALPHA,
      CC,EPS;
  CC:=C;
  'REAL'SS:=0,PP:=0;
  'INT'NN:=(N-('ODD'N!1!0))/'2;
  'FOR'I'TO'N'DO'SS'PLUS'C[I];
  'FOR'I'TO'NN'DO'PP'PLUS'C[2*I-1];
  'REAL'Z1,Z2;
  Z1:=SQRT(C[0]+2*SS);
  Z2:=SQRT(C[0]+2*(SS-2*PP));
  C[0]:=(Z1+Z2)/2;
  C[1]:=(Z1-Z2)/2;
  'PROC'MAXNM=('REF'[]'REAL'X,'REF' 'REAL'MAX)'VOID':
  (MAX:=0.0;'FOR'I'FROM'LWB'X'TO'UPB'X'DO'
    'IF'ABS'X[I]>'ABS'MAX'THEN'MAX:=X[I]'FI');
  'INT'NUM:=0;
  'BOOL'ACTIVE:='TRUE';
  'REAL'SUM,
      SUMBT,
      SUMBTT,
      EPSMAX,
      TT;
  'WHILE'ACTIVE'DO'
  'BEGIN'
    NUM'PLUS'1;
    PRINT((NEWLINE,"L E V E L *****",NUM,NEWLINE))
    'IF'N=1
      'THEN'ALPHA[0]:=BETTA[0]:=C[0]; ALPHA[1]:=BETTA[1]:=C[1]
    'ELSE'
      'FOR'I'FROM'0'TO'N'DO'ALPHA[I]:=BETTA[I]; 'CLEAR'BETTA;

    BETTA[N]:=C[N]/ALPHA[0]; SUMBT:=0;
    SUMBT:=BETTA[N];
    SUMBTT:=( 'ODD'N!BETTA[N]!0);
    'FOR'J'FROM'N-1'BY'-1'TO'2'DO'
    'BEGIN'
      SUM:=0;
      'FOR'I'FROM'J+1'TO'N'DO'SUM'PLUS'ALPHA[I-J]*BETTA[I];
      BETTA[J]:=(C[J]-SUM)/ALPHA[0];
      SUMBTT'PLUS'('ODD'J!BETTA[J]!0);
      SUMBT'PLUS'BETTA[J]
    'END';
    BETTA[1]:=C[1]-SUMBTT;
    BETTA[0]:=C[0]-(SUMBT-SUMBTT)
    'FI';
    EPSMAX:=0;
    'FOR'I'FROM'0'TO'N'DO'EPS[I]:=BETTA[I]-ALPHA[I];
    MAXNM(EPS,EPSMAX);
    PRINT((NEWLINE,"ALPHA BETTA& EPS ARE: ",NEWLINE,ALPHA,NEWLIN

```

```

      BETTA,NEWLINE,EPS,NEWLINE));

'FOR'K'FROM'0'TO'N'DO'
(TT:=0; 'FOR'J'FROM'K'TO'N'DO'TT'PLUS'BETTA[J-K]*BETTA[J];
  PRINT((NEWLINE,TT,"  ",CC[K],"  ",C[K],NEWLINE))
);
'IF'NUM>60'OR''ABS'EPSMAX<10&-12'THEN'ACTIVE:='FALSE''FI'
'END'
'END';'C' O F  G I T R M 'C'

```

```

'PROC'BACKFORD=('REF'[]'REAL'ALPHA,'REF'[]'REAL'Z,X)'VOID':
'BEGIN'
  'INT'R='UPB'ALPHA;
  'INT'N='UPB'X;
  [1:R,1:N+R]'REAL'F;
  [1:N+R]'REAL'E;
  [1:R]'REAL'M;
  'FOR'I'TO'R'DO'M[I]:=-ALPHA[I]/ALPHA[0];

  'FOR'I'TO'R'DO'
  'FOR'J'TO'R-1'DO' F[I,J]:=(J>I!0!ALPHA[I-J]);
  'FOR'I'TO'R'DO''FOR'J'TO'R'DO'F[I,N+J]:=(I=J!-ALPHA[0]!0);
  'FOR'I'TO'R'DO'E[N+I]:=0;

  'REAL'SUM,SUME;
  'FOR'J'FROM'N'BY'-1'TO'R'DO'
  'BEGIN'
    SUME:=0;
    'FOR'K'TO'R'DO'
    'BEGIN'
      SUM:=0;
      SUME'PLUS'M[K]*E[J+K];
      'FOR'I'TO'R'DO'SUM'PLUS'M[I]*F[K,J+I];
      F[K,J]:=SUM+(J=K!ALPHA[0]!0)
    'END';
    E[J]:=SUME+Z[J]
  'END';

  'FOR'I'TO'R-1'DO'
  'BEGIN'
    'FOR'K'TO'R'DO'
    'BEGIN'
      SUM:=0;
      'FOR'J'TO'R-I'DO'SUM'PLUS'M[I+J]*F[K,R+J];
      F[K,R-I]:=SUM+F[K,R-I]
    'END';
    SUME:=0;
    'FOR'J'TO'R-I'DO'SUME'PLUS'M[I+J]*E[R+J];
    E[R-I]:=SUME+Z[R-I]
  'END';

```



```

'BOOL'OK:='TRUE';
'INT'COUNT:=0,KHAT;
[1:R-1,1:R-1]'REAL'RATIO;
'CLEAR'RATIO;
  'WHILE'OK'DO
    'BEGIN'
      COUNT'PLUS'1;
      KHAT:=0;
      KHAT:=R+1-COUNT;
      'FOR'K'TO'R-COUNT'DO
        'BEGIN'
          RATIO[COUNT,K]:=-F[KHAT,K]/F[KHAT,KHAT];
          'FOR'J'TO'R-COUNT'DO'F[J,K]'PLUS'RATIO[COUNT,K]*F[J,KHAT
T];
          E[K]'PLUS'RATIO[COUNT,K]*E[KHAT]
        'END';
      'IF'COUNT=R-1'THEN'OK:='FALSE''FI'
    'END';
'REAL'SUMY;
[1:N]'REAL'GAMMA;
[-R+1:N]'REAL'Y;
'CLEAR'GAMMA; 'CLEAR'Y;
'FOR'I'TO'N'DO'GAMMA[I]:=(I>R!ALPHA[0]!F[I,I]);
'FOR'J'TO'N'DO'
  'BEGIN'
    'INT'KR:=0;
    SUMY:=0;
    'FOR'K'TO'R'DO'(KR:=-R+K+J-1;
      SUMY'PLUS'(J>R!Y[K]*F[K,J]!Y[KR]*(KR>0!F[KR,J]!
    ));
    Y[J]:=(E[J]-SUMY)/GAMMA[J]
  'END';

'REAL'SUMK;
[1:N]'REAL'YH;
[1:N+R]'REAL'ETAH;
'FOR'I'TO'N'DO'YH[I]:=Y[N-I+1];
'FOR'I'TO'R'DO'ETAH[N+I]:=0;
  'FOR'J'FROM'N'BY'-1'TO'R'DO'
    (SUME:=0;'FOR'K'TO'R'DO'SUME'PLUS'M[K]*ETAH[J+K];
      ETAH[J]:=SUME+YH[J]);
  'FOR'I'TO'R-1'DO'
    (SUME:=0;'FOR'J'TO'R-I'DO'SUME'PLUS'M[I+J]*ETAH[R+J];
      ETAH[R-I]:=SUME+YH[R-I]);
'INT'NUM:=0;
'BOOL'ACTIVE:='TRUE';
  'WHILE'ACTIVE'DO
    'BEGIN'
      NUM'PLUS'1;
      'FOR'T'TO'R-NUM'DO'ETAH[T]'PLUS'RATIO[NUM,T]*ETAH[R+1-NUM]
      'IF'NUM=R-1'THEN'ACTIVE:='FALSE''FI'
    'END';
[1:N]'REAL'K;

```

```

      'FOR'I'TO'N'DO'K[I]:=ETAH[N-I+1];
[1:N+R]'REAL'XX; 'CLEAR'XX;
'REAL'SUMXX;
'FOR'J'FROM'N'BY'-1'TO'1'DO'
'BEGIN'
  SUMXX:=0;
  'FOR'I'FROM'R'BY'-1'TO'1'DO'
    SUMXX'PLUS'(J<N-R+1!XX[N+I-R]*F[R+1-I,N+1-J]
                !XX[J+I]*F[I,N+1-J]);
    XX[J]:=(K[J]-SUMXX)/GAMMA[N+1-J];
    X[J]:=XX[J]
  'END'
'END';'C'OF BACKFORD 'C'

```

```

'PROC'GENSYM=('REF'[]'REAL'C,'REF'[]'REAL'Z,X)'VOID':
'BEGIN'
  'INT'R='UPB'C;
  [0:R]'REAL'ALPHA;
  'FOR'I'FROM'3'TO'R'DO'ALPHA[I]:=1;
  GITRM(C,ALPHA);
  [1:R]'REAL'BB;
  'FOR'I'TO'R'DO'BB[I]:=ALPHA[I]/ALPHA[0];
  PRINT((NEWLINE,"RATIOS ALPHA[I]/ALPHA[0],I=1,2,...M ARE",NEWLINE
    BB,NEWLINE,NEWLINE,NEWLINE));
  BACKFORD(ALPHA,Z,X)
'END';'C'OF GENSYM'C'

```

C2

```

'PROC'PRDSYS=('REF'[,]'REAL'A','REF'[]'REAL'Z,X)'VOID':
'BEGIN'
  'INT'N=2'UPB'A,
    R=1'UPB'A;
  'INT'T:='ENTIER'(N/R);
  'INT'RM:=N-T*R;
  [1:R,1:N]'REAL'L,
    EPS;
  [1:R+1,1:N]'REAL'U;
  [1:N]'REAL'EPSX;
  [1:R]'REAL'VECEPS;
  'REAL'EPSMAX;
  'REAL'SUM,SUM;
  'INT'NUM:=0;
  'BOOL'OK:='TRUE';
  'FOR'K'TO'R'DO'
    'C'
    'FOR'I'TO'N'DO'L[K,I]:=A[K,I]/A[0,I];
    'C'
    'FOR'I'TO'N'DO'L[K,I]:=10000;
    'C'INITIAL VALUES FOR L[1,...L[R-1,] 'C'

    'FOR'I'TO'N'DO'U[R+1,I]:=A[R,I];
  'WHILE'OK'DO'
  'BEGIN'
    NUM'PLUS'1;
    PRINT((NEWLINE,"L E V E L *****",NUM,NEWLINE));
    'FOR'K'FROM'R'BY'-1'TO'1'DO'
      'FOR'I'TO'N'DO'
        (SUM:=0; 'FOR'J'TO'R-K+1'DO'SUM'PLUS'L[J,I]*U[K+J,INT(N,
-J));
          U[K,I]:=A[K-1,I]-SUM);
        'FOR'K'FROM'R'BY'-1'TO'1'DO'
          (
            'FOR'I'TO'N'DO'
              (SUM:=0; 'FOR'J'TO'R-K+1'DO'SUM'PLUS'L[K+J-1,I]*U[J,INT(N,I-
(K+J-1))]);
                EPS[K,I]:=A[-K,I]-SUM);
              'FOR'I'TO'N'DO'L[K,I]'PLUS'EPS[K,I]/U[1,INT(N,I-K)]
            );
          'FOR'K'TO'R'DO'
            ( 'CLEAR'EPSX; EPSMAX:=0;
              EPSX:=EPS[K,];
              MAXNM(EPSX,EPSMAX);
              VECEPS[K]:=EPSMAX
            );

    PRINT((NEWLINE,"L LNEW&EPS ARE",NEWLINE,L,NEWLINE,EPS,NEWLINE)
;
    PRINT((NEWLINE,"U VALUES ARE",NEWLINE,U,NEWLINE));
    MAXNM(VECEPS,EPSMAX);
    'IF'NUM>61 'OR''ABS'EPSMAX<10&-12'THEN'OK:='FALSE''FI'
  'END';

```

```

'PROC 'TRNGMTX= ('REF' [,] 'REAL' F, FS, 'REF' [] 'REAL' E, 'REF' [] 'REAL' ES) 'VOI
D':
'BEGIN'
  'INT' R=1 'UPB' F,
    N='UPB' ES;
  'BOOL' OK:='TRUE';
  'INT' COUNT:=0, KHAT;
  [1:R-1, 1:R-1] 'REAL' RATIO;
  'CLEAR' RATIO;
  'WHILE' OK 'AND' R>1 'DO'
    'BEGIN'
      COUNT 'PLUS' 1;
      KHAT:=0;
      KHAT:=R+1-COUNT;
      'FOR' K 'TO' R-COUNT 'DO'
        'BEGIN'
          RATIO [COUNT, K] :=-F [KHAT, K] / F [KHAT, KHAT];
          'FOR' J 'TO' R-COUNT 'DO' F [J, K] 'PLUS' RATIO [COUNT, K] * F [J, KHAT];
          E [K] 'PLUS' RATIO [COUNT, K] * E [KHAT]
        'END';
      'IF' COUNT=R-1 'THEN' OK:='FALSE' 'FI'
    'END';
  FS:=F;
  'FOR' I 'TO' N 'DO' ES [I] := E [I]
'END'; 'C' OF TRNGMTX 'C'

[1:N] 'REAL' ZY; ZY:=Z;
[1:N] 'REAL' Y;
[1:R, -R+1:N] 'REAL' F;
[-R+1:N] 'REAL' E, ZZ;

'FOR' I 'TO' R 'DO'
  'FOR' J 'TO' R-1 'DO' F [I, N-J+1] := (J=I!!1! : J<I!!L [I-J, N-J+1]!0);
'FOR' J 'TO' R 'DO'
  'FOR' K 'TO' R 'DO' F [K, -J+1] := (J=K!!-1!0);
'FOR' I 'TO' R 'DO' E [-I+1] := 0;
  'FOR' I 'TO' N-R+1 'DO'
    'BEGIN'
      'FOR' K 'TO' R 'DO'
        (SUM:=0;
          'FOR' J 'FROM' R 'BY' -1 'TO' 1 'DO'
            SUM 'PLUS' (-L [J, I] * F [K, I-J]);
            F [K, I] := SUM + (I+K=N+1!!1!0)
          );
        SUME:=0;
        'FOR' J 'FROM' R 'BY' -1 'TO' 1 'DO' SUME 'PLUS' (-L [J, I] * E [I-J]);

```

```

      E[I]:=Z[I]+SUME
    'END';
    'FOR' I 'FROM' R 'BY' -1 'TO' 2 'DO'
    'BEGIN'
      'FOR' K 'TO' R 'DO'
        (SUM:=F[K,N+2-I];
        'FOR' J 'TO' I-1 'DO' SUM'PLUS' (-L[R-I+1+J,N+2-I]*F[K,N-R+1-J
);
        F[K,N+2-I]:=SUM
      );
      SUME:=Z[N+2-I];
      'FOR' J 'TO' I-1 'DO' SUME'PLUS' (-L[R-I+1+J,N+2-I]*E[N-R+1-J]
;
      E[N+2-I]:=SUME
    'END';

[1:R,1:N] 'REAL' FF,FS; 'CLEAR' FF; 'CLEAR' FS;
[1:N] 'REAL' ZS;
  'FOR' K 'TO' R 'DO'
    'FOR' I 'TO' N 'DO' FF[K,I]:=F[K,N-I+1];
    'FOR' I 'TO' N 'DO' ZZ[I]:=E[N-I+1];
  TRNGMTX(FF,FS,ZZ,ZS);
  'FOR' K 'TO' R 'DO'
    'FOR' I 'TO' N 'DO' F[K,I]:=FS[K,N-I+1];
    'FOR' I 'TO' N 'DO' Z[I]:=ZS[N-I+1];
  'REAL' SUMY;
[1:N] 'REAL' BB;
'CLEAR' Y;
  'FOR' I 'TO' R 'DO' BB[N+1-I]:=F[I,N+1-I];

  'FOR' I 'TO' N-R 'DO' BB[I]:=1;

  'FOR' J 'FROM' N 'BY' -1 'TO' 1 'DO'
    (SUMY:=0;
    'FOR' I 'FROM' R 'BY' -1 'TO' 1 'DO' SUMY'PLUS' Y[N+I-R]*F[R+1-I,J];
    Y[J]:=(Z[J]-SUMY)/BB[J]
    );

'FOR' I 'TO' N 'DO'
(SUM:=Y[I]; 'FOR' K 'TO' R 'DO' SUM'PLUS' L[K,I]*Y[INT(N,I-K)];
PRINT((NEWLINE,SUM," ",ZY[I]));
[1:R,1:N+R] 'REAL' G,GG;
[1:N+R] 'REAL' YY; 'CLEAR' YY;
[1:R] 'REAL' W;

'FOR' I 'TO' R 'DO'
  'FOR' J 'TO' R-1 'DO' G[I,J]:=(J<=I!U[I-J+1,J]!0);
'FOR' I 'TO' R 'DO' 'FOR' J 'TO' R 'DO' G[I,N+J]:=(J=I!-U[1,I]!0);
'FOR' I 'TO' R 'DO' YY[N+I]:=0;
  'FOR' J 'FROM' N 'BY' -1 'TO' R 'DO'
    'BEGIN'
      'FOR' K 'TO' R 'DO'
        (SUM:=0;
        'FOR' I 'TO' R 'DO'

```

```

      SUM'PLUS'((-U[I+1,J]/U[1,INT(N,I+J)])*G[K,I+J]);
      G[K,J]:=SUM+(K=J!U[1,R]!0);
      SUME:=0;
      'FOR'I'TO'R'DO'SUME'PLUS'(-U[I+1,J]/U[1,INT(N,I+J)])*YY[I+
];
      YY[J]:=Y[J]+SUME
'END';
'FOR'I'TO'R-1'DO'
'BEGIN'
  'FOR'K'TO'R'DO'
    (SUM:=G[K,R-I];
    'FOR'J'TO'R-I'DO'SUM'PLUS'(-U[I+J+1,R-I]/U[1,R+J])*G[K,R-
J];
      G[K,R-I]:=SUM);
      SUME:=Y[R-I];
      'FOR'J'TO'R-I'DO'SUME'PLUS'(-U[I+J+1,R-I]/U[1,R+J])*YY[R+J
;
      YY[R-I]:=SUME
'END';
GG:=G;
  TRNGMTX(GG,G,YY,Y);
'REAL'SUMX;
'CLEAR'BB;'CLEAR'X;
  'FOR'I'TO'R'DO'BB[I]:=G[I,I];
  'FOR'I'FROM'R+1'TO'N'DO'BB[I]:=U[1,I];
  'FOR'J'TO'N'DO'
    (SUMX:=0;
    'FOR'K'TO'R'DO'SUMX'PLUS'X[K]*G[K,J];
    X[J]:=(Y[J]-SUMX)/BB[J]
    )
'END';'C' O F P R D S Y S 'C'

```

C3

```

'PROC'NONPDUL=('REF'[,]'REAL'U,'REF'[,]'REAL'L,'REF'[,]'REAL'A)'VOID
:
'BEGIN'
  'INT'N=2'UPB'A,
    R=1'UPB'A;
  'INT'T:='ENTIER'(N/R);
  'INT'RM:=N-T*R;
  [1:R]'INT'SPAREVEC;
  'PROC'GETSPVEC=('INT'RM,'REF'[]'INT'X)'VOID':
  'BEGIN'
    'INT'COUNT:=1,K,R,RK;
    R:='UPB'X;
    RK:=(RM=0!R!RM);
    X[1]:=RK;
    'FOR'J'FROM'RK-1'BY'-1'TO'1'DO'(COUNT'PLUS'1;X[COUNT]:=J);
    'IF'COUNT<R'THEN'
      'FOR'J'FROM'R'BY'-1'TO'RM+1'DO'(COUNT'PLUS'1;X[COUNT]:=J)
    'FI'
  'END';'C'OF GETSPVEC 'C'
  GETSPVEC(RM,SPAREVEC);

  [1:R,1:N]'REAL'EPS;
  [1:N]'REAL'EPSX;
  [1:R]'REAL'VECEPS;
  'REAL'EPSMAX;
  'REAL'SUM;
  'INT'NUM:=0;
  'BOOL'OK:='TRUE';
  'FOR'K'TO'R'DO'
    'C'
    'FOR'I'TO'N'DO'L[K,I]:=A[K,I]/A[0,I];
    'C'
    'FOR'I'TO'N'DO'L[K,I]:=10;
    'C'INITIAL VALUES FOR L[1,...L[R,] 'C'

    'FOR'I'TO'N-R'DO'U[R+1,I]:=A[R,I];
    'FOR'J'TO'R'DO'U[R+1,N-J+1]:=U[R+1,SPAREVEC[J]];
  'WHILE'OK'DO'
  'BEGIN'
    NUM'PLUS'1;
    PRINT((NEWLINE,"L E V E L *****",NUM,NEWLINE));

```

```

'FOR'K'FROM'R'BY'-1'TO'1'DO'
  ('FOR'I'TO'N-K+1'DO'
    (SUM:=0; 'FOR'J'TO'R-K+1'DO'SUM'PLUS'U[K+J,I]*L[J,I+K-1]
      U[K,I]:=A[K-1,I]-SUM);
    'FOR'J'TO'K-1'DO'U[K,N-J+1]:=U[K,SPAREVEC[J]]
  );
'FOR'K'FROM'R 'BY'-1'TO'1'DO'
  (
    'FOR'I'TO'N-K'DO'
      (SUM:=0;'FOR'J'TO'R-K+1'DO'SUM'PLUS'L[K+J-1,I]*U[J,I+K];
        EPS[K,I]:=A[-K,I+K]-SUM);
    'FOR'I'TO'N-K'DO'L[K,I]'PLUS'EPS[K,I]/U[1,I+K]
  );
  'FOR'S'FROM'0'TO'R-1'DO''FOR'J'TO'R-S'DO'
    (L[R-S,N-J+1]:=L[R-S,SPAREVEC[J]]);
    EPS[R-S,N-J+1]:=EPS[R-S,SPAREVEC[J]]);
  'FOR'K'TO'R'DO'
    ( 'CLEAR'EPSX; EPSMAX:=0;
      EPSX:=EPS[K,];
      MAXNM(EPSX,EPSMAX);
      VECEPS[K]:=EPSMAX
    );
  PRINT((NEWLINE,"L LNEW&EPS ARE",NEWLINE,L,NEWLINE,EPS,NEWLINE)
;
  PRINT((NEWLINE,"U VALUES ARE",NEWLINE,U,NEWLINE))
;
  MAXNM(VECEPS,EPSMAX);
  'IF'NUM>61 'OR''ABS'EPSMAX<10&-12'THEN'OK:='FALSE''FI'
  'END'
'END';'C'OF NONPDUL 'C'

```

```

'PROC'BACKFORD=('REF'[,]'REAL'GAH,ALH,'REF'[]'REAL'X,Z)'VOID':
'BEGIN'
  'INT'R=1'UPB'GAH,
    N=2'UPB'GAH;
  [0:R,1:N+R]'REAL'GA,
    AL;
  [1:N,1:R]'REAL'PSI;
  [1:N]'REAL'PHI;
  [1:R,1:N+R]'REAL'G;
  'CLEAR'G;
  [1:N+R]'REAL'ZZ;
  [1:R]'REAL'M;
  'CLEAR'GA;'CLEAR'AL;'CLEAR'ZZ;
  'FOR'K'FROM'0'TO'R'DO'(GA[K,1:N]:=GAH[K,];AL[K,1:N]:=ALH[K,]);
  'FOR'T'TO'R'DO'(G[T,N+T]:=-1;GA[0,N+T]:=1);
  'FOR'J'FROM'N'BY'-1'TO'1'DO'
    'BEGIN'
      'FOR'K'TO'R'DO'M[K]:=-GA[K,J]/GA[0,J+K];
      ZZ[J]:=Z[J];
    'END'
  'END'

```



```

      'FOR'K'TO'R'DO'
      'BEGIN'
        ZZ[J]'PLUS'M[K]*ZZ[J+K];
        'FOR'S'TO'R'DO'G[K,J]'PLUS'M[S]*G[K,J+S]
      'END'
    'END';
    'FOR'J'TO'N'DO'
    'BEGIN'
      PHI[J]:=ZZ[J]/GA[0,J];
      'FOR'K'TO'R'DO'PSI[J,K]:=-G[K,J]/GA[0,J]
    'END';
    [1:N+R,0:R]'REAL'T;
    [1:N+R,0:N]'REAL'C;
    'INT'D,
      S;

    'FOR'J'TO'N+R'DO'
      'FOR'K'TO'R'DO'
        'IF'J-K>0'AND'J-K<=N
        'C'
        'THEN'T[J,K]:=-AL[K,J-K]/AL[0,J-K]
        'C'
        'THEN'T[J,K]:=-AL[K,J-K]
        'FI';
      C[1,0]:=T[1,0]:=1;
      'FOR'J'FROM'2'TO'N+R'DO'
      'BEGIN'
        C[J,0]:=T[J,0]:=1;
        'FOR'I'TO'(J<=N+1!J-1!N)'DO'
          (
            C[J,I]:=0;
            'IF'J<=N+1
            'THEN'S:=(I<R!I!R);
              'FOR'K'TO'S'DO'C[J,I]'PLUS'T[J,K]*C[J-K,I-K]
            'ELSE'D:=N+R+1-J;
              S:=(I<D!I!D);
              'FOR'K'TO'S'DO'C[J,I]'PLUS'T[J,R-D+K]*C[N+1-K,I-K]
            'FI'
          )
        'END';
      [1:R,1:R]'REAL'CPSI;
      [1:R]'REAL'CPHI,
        YH;
      'REAL'SUM;
      'CLEAR'CPSI;'CLEAR'CPHI;
      [1:N]'REAL'Y;
      'FOR'I'TO'R'DO'
        'FOR'J'TO'R'DO'
          'FOR'K'TO'N'DO'CPSI[I,J]'PLUS'C[N+I,N+1-K]*PSI[K,J];
        'FOR'I'TO'R'DO'CPSI[I,I]'PLUS'1;
        'FOR'I'TO'R'DO'
          'FOR'K'TO'N'DO'CPHI[I]'PLUS'-C[N+I,N+1-K]*PHI[K];
      PRINT((NEWLINE,"CPSI & CPHI ARE",NEWLINE,CPSI,NEWLINE,CPHI));
      'IF'R=1'THEN'YH[1]:=CPHI[1]/CPSI[1,1]

```

```

      'ELSE' 'C' SOLVE LINEAR SYSTEM (A V=B), WHERE A=CPSI, V=YH, B=CPI
I'C'
      SOLVESYS(CPSI,CPHI,YH)
      'FI';
      'FOR' S 'TO' N 'DO'
      'BEGIN'
      Y[S]:=PHI[S];
      'FOR' K 'TO' R 'DO' Y[S] 'PLUS' PSI[S,K]*YH[K]
      'END';

[1:N] 'REAL' YK;
      'FOR' I 'TO' N 'DO'
      'BEGIN'
      YK[I]:=Y[I];
      'IF' I > 1 'THEN' 'FOR' J 'FROM' I-1 'BY' -1 'TO' 1 'DO' YK[I] 'PLUS' Y[I-J]
C[I,J]
      'FI';
      'C'
      X[I]:=YK[I]/AL[0,I]
      'C'
      X[I]:=YK[I]
      'END'
      'END'; 'C' O F B A C K F O R D 'C'

```

```

'PROC'BACKFORDBLK=('REF'[,],'REAL'Q0,Q1,'REF'[,],'REAL'Z,X)'VOID':
'BEGIN'
  'INT'M=1'UPB'Z,
    N=2'UPB'Z;
  [1:M,1:N,1:N]'REAL'F;
  [1:N,1:N]'REAL'MX,E,
    QI,QK,
    B,
    S,
    ZEROMTX;
  [1:M,1:N]'REAL'YK,
    YH,
    ZK,
    ZH,
    Y;
  [1:N,1:1]'REAL'VV,
    V;
  [1:N]'REAL'VR,
    VL;
  'CLEAR'ZEROMTX;
  'CLEAR'E;'FOR'I'TO'N'DO'E[I,I]:=1;
    INVMTX(Q0,QI);'C' INVERT MATRIX Q0,QI=INVERSE OF Q0 'C'
    QK:=PRODUCT(QI,Q1);
    'FOR'I'TO'M'DO'
      ( V[,1]:=Z[I,];
        V:=PRODUCT(QI,V);
        ZK[I,]:=V[,1]
      );
    F[M,,]:=QK;MX:=SUBMTX(ZEROMTX,QK);ZH[M,]:=ZK[M,];
    'FOR'I'FROM'M-1'BY'-1'TO'1'DO'
      ( S:=F[I+1,,];
        S:=PRODUCT(MX,S);
        F[I,,]:=S;
        V[,1]:=ZH[I+1,];
        V:=PRODUCT(MX,V);
        VV[,1]:=ZK[I,];
        V:=ADDMTX(V,VV);
        ZH[I,]:=V[,1]
      );
      S:=F[1,,];
      B:=ADDMTX(E,S);
      F[1,,]:=B;
    VR:=ZH[1,];B:=F[1,,];
    SOLVESYS(B,VR,VL);'C' SOLVE LINEAR SYSTEM B(VL)=VR 'C'
    Y[1,]:=VL;
    'FOR'K'FROM'2'TO'M'DO'
      ( V[,1]:=Y[1,];
        B:=F[K,,];
        V:=PRODUCT(B,V);
        VV[,1]:=ZH[K,];
        V:=SUBMTX(VV,V);
        Y[K,]:=V[,1]
      );
    'FOR'I'TO'M'DO'
      ( V[,1]:=Y[I,];

```

```

      VV:=PRODUCT(QI,V);
      YK[1,:]=VV[,1]
    );

    YH[1,:]=YK[1,:];
    'FOR'K'FROM'2'TO'M'DO'
    ( V[,1]:=YH[K-1,:];V:=PRODUCT(MX,V);
      VV[,1]:=YK[K,:];V:=ADDMTX(V,VV);
      YH[K,:]=V[,1]
    );

    VR:=YH[M,:];B:=F[1,:];
    SOLVESYS(B,VR,VL);'C' SOLVE LINEAR SYSTEM B(VL)=VR 'C'
    X[M,:]=VL;
    'FOR'K'FROM'M-1'BY'-1'TO'1'DO'
    ( V[,1]:=X[M,:];
      B:=F[M-K+1,:];
      V:=PRODUCT(B,V);
      VV[,1]:=YH[K,:];
      V:=SUBMTX(VV,V);
      X[K,:]=V[,1]
    )
  'END';'C'OF BACKFORDBLK'C'

```

```

'PROC'GITRMBLK=('REF'[,]'REAL'B,C,Q0,Q1)'VOID':
'BEGIN'

```

```

  'INT'N=1'UPB'B;
  [1:N,1:N]'REAL'BB,CC,D;
  D:=ADDMTX(C,C);
  BB:=ADDMTX(B,D);
  CC:=SUBMTX(B,D);
  BB:=SQRTMTX(BB);
  CC:=SQRTMTX(CC);
  Q0:=ADDMTX(BB,CC);
  Q1:=SUBMTX(BB,CC);

```

5)

```

  'FOR'I'TO'N'DO''FOR'J'TO'N'DO'(Q0[I,J]'TIMES'0.5;Q1[I,J]'TIMES'0
  'END';'C'OF GITRMBLK'C'

```

```

'PROC'GENSYSBLK=('REF'[,]'REAL'B,C,'REF'[,]'REAL'Z,X)'VOID':
'BEGIN'

```

```

  'INT'M=1'UPB'Z,
  N=2'UPB'Z;
  [1:N,1:N]'REAL'Q0,Q1;
  [1:N,1:N]'REAL'E,D;
  'REAL'S1,S2;
  GITRMBLK(B,C,Q0,Q1);
  INFMTXNM(Q0,S1);INFMTXNM(Q1,S2);
  'IF'S2>S1'THEN'E:=Q0;Q0:=Q1;Q1:=E'FI';
  BACKFORDBLK(Q0,Q1,Z,X)

```

```

  'END';'C'OF GENSYSBLK'C'

```

C5

'BEGIN'

'C' 2-POINT B.V.P.WITH PERIODIC CONDITIONS 'C'

'INT' PROBNO;

```
[1:4] 'PROC' ('REAL', 'REAL', 'REAL') 'REAL' FNV,
                                     DFZV,
                                     DFYV;
```

```
[1:4] 'PROC' ('REAL') 'REAL' EXACTYV,
                                     EXACTDYV;
```

```
[1:4] 'REAL' XAV,
                                     XBV;
```

'C' PROBLEM 1: $D^2Y - Y - Y^2 - \exp(\sin 2\pi X) [4\pi^2 (\cos^2 \pi X - \sin^2 \pi X) - \exp(2\sin 2\pi X) - 1]$ 'C'

```
FNV[1] := ('REAL' X, Y, YD) 'REAL': ( Y + Y*Y + EXP(SIN(2*PI*X)) * (4*PI*PI * (
COS(2*PI*X)^2 - SIN(2*PI*X)) - EXP(2*SIN(2*PI*X)) - 1) ) ;
```

```
DFZV[1] := ('REAL' X, Y, YD) 'REAL': (0) ;
```

```
DFYV[1] := ('REAL' X, Y, YD) 'REAL': (1 + 3*Y*Y) ;
```

```
EXACTYV[1] := ('REAL' X) 'REAL': (EXP(SIN(2*PI*X))) ;
```

```
XAV[1] := 0.0 ;
```

```
XBV[1] := 1.0 ;
```

'C' PROBLEM 2: $D^2Y = Y^3 - \sin(X) (1 + \sin(X)^2)$ 'C'

```
FNV[2] := ('REAL' X, Y, YD) 'REAL': (Y^3 - SIN(X) * (1 + SIN(X)^2)) ;
```

```
DFZV[2] := ('REAL' X, Y, YD) 'REAL': (0) ;
```

```
DFYV[2] := ('REAL' X, Y, YD) 'REAL': (3.0*Y^2) ;
```

```
EXACTYV[2] := ('REAL' X) 'REAL': (SIN(X)) ;
```

```
XAV[2] := 0 ;
```

```
XBV[2] := 2.0*PI ;
```

'C' PROBLEM 3: $D^2Y + 4Y = 3\sin(X)$ 'C'

```
FNV[3] := ('REAL' X, Y, YD) 'REAL': (4*Y - 4*SIN(2*X) - 5*SIN(X)) ;
```

```
DFZV[3] := ('REAL' X, Y, YD) 'REAL': (0) ;
```

```
DFYV[3] := ('REAL' X, Y, YD) 'REAL': (4) ;
```

```
EXACTYV[3] := ('REAL' X) 'REAL': (3.5 * (SIN(2*X) + 2*SIN(X))) ;
```

```
XAV[3] := -0.5*PI ;
```

```
XBV[3] := 1.5*PI ;
```

'C' PROBLEM 4: $D^2Y - (1 - Y^2)YD - 4Y = -5\sin(X) - \cos(X)^3$ 'C'

```
FNV[4] := ('REAL' X, Y, YD) 'REAL': ((1 - Y*Y)*YD + 4*Y - 5*SIN(X) - COS(X)^3) ;
```

```
DFZV[4] := ('REAL' X, Y, YD) 'REAL': (1 - Y*Y) ;
```

```
DFYV[4] := ('REAL' X, Y, YD) 'REAL': (-2*Y*YD + 4) ;
```

```
EXACTYV[4] := ('REAL' X) 'REAL': (SIN(X)) ;
```

```
XAV[4] := 0.0 ;
```

```
XBV[4] := 2.0*PI ;
```

'PROC' PVAND = ('REF' [] 'REAL' ALPHA, X, B) 'VOID':

'BEGIN'

```
'INT' N = 'UPB' ALPHA ;
```

```
'FOR' K 'FROM' 0 'TO' N 'DO' X[K] := B[K] ;
```

```
'FOR' K 'FROM' 0 'TO' N-1 'DO'
```

```
  'FOR' J 'FROM' N 'BY' -1 'TO' K+1 'DO'
```

```
    X[J] 'PLUS' (-ALPHA[K] * X[J-1]) ;
```

```
'FOR' K 'FROM' N-1 'BY' -1 'TO' 0 'DO'
```

```
'BEGIN'
```

```

    'FOR'J'FROM'K+1'TO'N'DO'X[J] 'DIV'(ALPHA[J]-ALPHA[J-K-1]);
    'FOR'J'FROM'K'TO'N-1'DO'X[J] 'PLUS' (-X[J+1])
  'END'
'END'; 'C' OF PVAND 'C'

'PROC'RECMULT=('INT'N,'REF''REAL'X)'REAL':
  ( (N=3!1 ! X*RECMULT(N-1,X) ) );

'PROC'INT=('INT'N,I)'INT':('INT'K:=I;K'PLUS'(I<1!N!:I>N!-N!0));
'PROC'MAXNM=('REF'[]'REAL'X,'REF''REAL'MAX)'VOID':
(MAX:=0;'FOR'I'FROM'LWB'X'TO'UPB'X'DO'
  'IF''ABS'X[I]>'ABS'MAX'THEN'MAX:=X[I]'FI');

```

```

'WHILE' READ(PROBNO);PROBNO#0'DO'
'BEGIN'
  'PROC'('REAL','REAL','REAL')'REAL'FN:=FNV[PROBNO];
  'PROC'('REAL','REAL','REAL')'REAL'DFZ:=DFZV[PROBNO];
  'PROC'('REAL','REAL','REAL')'REAL'DFY:=DFYV[PROBNO];
  'PROC'('REAL')'REAL'EXACTY:=EXACTYV[PROBNO];
  'REAL'XA:=XAV[PROBNO],
  XB:=XBV[PROBNO];

```

```

'C' MAIN LOOP 'C'

```

```

'CHAR'CHAR;
'INT'N,
  RMAX;
'INT'RB;
'WHILE'READ((NEWLINE,CHAR));CHAR#"\" 'DO'
'BEGIN'
  'REAL'EPS;
  READ((EPS,RMAX));
  [0:RMAX]'INT'RC,

```

```

      RE,
      QQ;
'FOR' I 'TO' RMAX 'DO' READ((RE[I], RC[I], QQ[I]));
RE[0] := RC[0] := 0;
QQ[0] := 1;
'WHILE' READ(N); N#0 'DO'
'BEGIN'
  [-20:N+20] 'REAL' X;
  PRINT((NEWPAGE, "PROBLEM NUMBER", PROBNO, NEWLINE));
  'REAL' H := (XB-XA)/N;
  PRINT((NEWLINE, "STEP SIZE TAKEN IS: ", H, NEWLINE));
  'FOR' I 'FROM' -20 'TO' N+20 'DO' X[I] := XA + I * H;

RB := 0;
'WHILE' RB < 5 'DO'
'BEGIN'
  'REAL' TRICK;
  RB 'PLUS' 1;
  [-RB:RB, 1:N] 'REAL' GAMMAI,
  BI;
  'PROC' FINDCF3PTS = ('INT' I, 'REF' [,] 'REAL' A, B) 'VOID':
  'BEGIN'
    [0:2*RB] 'REAL' ALPHAPVD, BD1Y, BD2Y, R1, R2;
    'CLEAR' BD1Y; 'CLEAR' BD2Y;
    BD2Y[2] := 2;
    BD1Y[1] := 1;
    'FOR' K 'FROM' RB-1 'BY' -1 'TO' 0 'DO' ALPHAPVD[RB-1-K] := X[I+K+
1]-X[I];
    'FOR' K 'FROM' 0 'TO' RB-1 'DO' ALPHAPVD[RB+1+K] := -(X[I]-X[I-K
-1]);
    ALPHAPVD[RB] := 0;
    PVAND(ALPHAPVD, R2, BD2Y);
    PVAND(ALPHAPVD, R1, BD1Y);
    'FOR' K 'FROM' -RB 'TO' RB 'DO' (A[K, I] := R2[RB-K]; B[K, I] := R1[R
B-K])
  'END'; 'C' O F FINDCF3PTS 'C'
  'PROC' SPX = ('INT' I, 'REF' [] 'REAL' Y, 'REF' [,] 'REAL' A, 'REF' 'REAL'
T) 'VOID':
  'BEGIN'
    'INT' J; T := 0;
    'FOR' K 'FROM' -RB 'TO' RB 'DO' (J := I+K; J 'PLUS' (J < 1!N! : J > N!-N!0)
;
    T 'PLUS' Y[J] * A[K, I])
  'END'; 'C' O F SPX 'C'

  'FOR' I 'TO' N 'DO' FINDCF3PTS(I, GAMMAI, BI);
  'C'
  PRINT((NEWLINE, "GAMMAI IS", NEWLINE, GAMMAI, "BI IS ***", NEWLI
NE,
  BI, NEWLINE));
  'C'
  PRINT(("NUMBER OF STEPS IS ", N, NEWLINE));

  'PROC' DIFFCORR = ('INT' R, 'REF' [] 'REAL' Y, DIFCORVEC) 'VOID':

```

```

'BEGIN'
  [0:2*R+2]'REAL'ALPHA,
              C2OFDCPVD,
              C1OFDCPVD,
              BD2Y,
              BD1Y;
  'CLEAR'BD2Y;'CLEAR'BD1Y;
  BD2Y[2]:=2;BD1Y[1]:=1;
  'FOR'I'TO'N'DO'
  'BEGIN' 'C' FIND DIFFCOR OF 1ST & 2ND DIF 'C'
    'REAL'S:=0,
      YD:=0,
      TT1:=0,
      TT2:=0;
    'CLEAR'C2OFDCPVD;'CLEAR'C1OFDCPVD;
    'FOR'K'FROM'0'TO'2*R+2'DO'
      ALPHA[K]:=X[I-R-1+K]-X[I];
      PVAND(ALPHA,C2OFDCPVD,BD2Y);
      PVAND(ALPHA,C1OFDCPVD,BD1Y);

    'FOR'J'FROM'-R-1'TO'R+1'DO'
    'BEGIN'
      'INT'T=(I+J>N!I+J-N! :
        I+J<1!I+J+N!
        I+J);
      S'PLUS'Y[T ]*C2OFDCPVD[J+R+1];
      YD'PLUS'Y[T ]*C1OFDCPVD[J+R+1]
    'END';
    SPX(I,Y,BI,TT1);
    SPX(I,Y,GAMMAI,TT2);

    DIFCORVEC[I]:=-S-FN(X[I],Y[I],TT1)+FN(X[I],Y[I],YD)+TT2
    'END'
  ;DIFCORVEC[0]:=DIFCORVEC[N]
  ;PRINT(("DIF. CORRECTION OF O R D E R ",2*R+2,"ARE:",NEWLIN
E,
      DIFCORVEC,NEWLINE))
'END' ;

'PROC'JACOBIMTX=('REF'[]'REAL'Y,'REF'[,]'REAL'A)'VOID':
'BEGIN'
  'REAL'Y DASH,DFZI;
  'FOR'I'TO'N'DO'
  'BEGIN'
    Y DASH:=0;
    SPX(I,Y,BI,Y DASH);
    DFZI:=DFZ(X[I],Y[I],Y DASH);
    'FOR'K'FROM'-RB'TO'RB'DO'A[K,I]:=GAMMAI[K,I]-BI[K,I]*DFZI
  ;
    A[0,I]'PLUS'-DFY(X[I],Y[I],Y DASH)
  'END'
'END';'C' O F JACOBIMTX 'C'

'PROC'FFORNEWT=('REF'[]'REAL'Y,VECF,DIFCORVEC)'VOID':

```



```

'BEGIN'
  'REAL' TT1, TT2;
  'FOR' I 'TO' N 'DO'
    'BEGIN'
      TT1:=TT2:=0;
      SPX(I, Y, GAMMAI, TT2);
      SPX(I, Y, BI, TT1);
      VECF[I] := TT2 - FN(X[I], Y[I], TT1) - DIFCORVEC[I]
    'END'
  'END';

'PROC' NEWTSOL = ('REF' [,] 'REAL' Y, DIFCORVEC, 'REF' [,] 'REAL' S, 'INT'
R,
ID':
                                'REAL' EPS) 'VO

'BEGIN'
  [0:N] 'REAL' DELY;
  [1:N] 'REAL' VECF;
  'REAL' T:=MAXREAL,
    S:=MAXREAL;
  'FOR' QQ 'TO' 'IF' CHAR="C" 'THEN' 20 'ELSE' (R=0!20!1) 'FI'
  'WHILE' S>N*EPS^2 'AND' (S>EPS 'OR' S<T) 'DO'
    'BEGIN'
      T:=S;
      FFORNEWT(Y, VECF, DIFCORVEC);
      'IF' R=0 'OR' CHAR#"X" 'THEN' JACOBI MTX(Y, B) 'FI';
      'IF' 'ODD' RB
      'THEN' 'FOR' I 'TO' N 'DO' VECF[I] 'TIMES' -1;
        'FOR' K 'FROM' -RB 'TO' RB 'DO' 'FOR' I 'TO' N 'DO' B[K, I] 'TIM
ES' -1
      'FI';
      PRDSYS(B, VECF, DELY[1:N]);
      'C' OR SOLVE THE SYSTEM BY GENSYM, I.E. FICM1, IF POSSIBLE '
      C'
      DELY[0] := DELY[N];
      S:=0;
      'FOR' I 'FROM' 0 'TO' N 'DO'
        'BEGIN'
          S 'PLUS' DELY[I]^2;
          Y[I] 'MINUS' DELY[I]
        'END';
      PRINT((NEWLINE, "NEWTON ITERATION", QQ, "NORM OF DELTA Y ",
        Sqrt(S/(N+1))))
    'END'
  'END';

'C' I N N E R L O O P 'C'

[0:N] 'REAL' Y,
      DIFCORVEC,
      YY;
'REAL' ZZZ,
      SSS;

```

```

[-RB:RB,1:N]'REAL'A;

'FOR'R'FROM'0'TO'RMAX'DO'
'BEGIN'
  'REAL'TT:=MAXREAL,
  SS:=MAXREAL/3;
  'FOR'Q1'TO'QQ[R]
  'WHILE'SS>N*EPS^2      'AND'SS<TT/2      'DO'
  'BEGIN'
    TT:=SS;
    'IF'R=0
    'THEN'
      'FOR'I'TO'N'DO'
      'BEGIN'
        DIFCORVEC[I]:=0;
        'C'INITIAL VALUES FOR NEWTON,S PROCEDURE 'C'
        Y[I]:=1
        'END'; Y[0]:=Y[N]
        ;PRINT(("Y I S : ",NEWLINE,Y,NEWLINE))
      'ELSE'DIFFCORR(RC[R]+RB-1,YY,DIFCORVEC)
      'FI';
      'IF'R>0
      'THEN'
        PRINT((NEWLINE,NEWLINE,"DIFFERENCE CORRECTON ERROR O
RDER",
              2*RC[R]+2,"(TERMS UP TO DELTA",2*RC[R]+1,"AND"
              2*RC[R]+2," )",NEWLINE,"N=",N,NEWLINE))
      'FI' ;
      NEWTSOL(Y,DIFCORVEC,A,R,EPS); SSS:=0;
      PRINT((NEWLINE,"
CORRECTION                                ERROR",NEWLINE));
      'FOR'I'FROM'0'TO'N'DO'
      'BEGIN'
        'IF'R>0'THEN'YY[I]:=Y[I]-YY[I]'FI';
        ZZZ:=EXACTY(X[I])-Y[I];
        SSS'PLUS'ZZZ^2;
        PRINT(("X[" ,I,"]=" ,X[I] ,Y[" ,I,"]=" ,Y[I]));
        'IF'R>0

        'THEN'PRINT(YY[I])
        'ELSE'PRINT("
" )
        'FI';
        PRINT((ZZZ,NEWLINE))
      'END';

      PRINT((NEWLINE,"ERROR IN Y HAS NORM*****",
              Sqrt(SSS/(N+1)),NEWLINE));
      PRINT((NEWLINE,"EXPECTED ERROR IS:  ",RECMULT(2*R+4,
              NEWLINE));
      'IF'R>0'AND'Q1>=1
      'THEN'SS:=0;

```

APP.SOLN

H),

```

      'FOR'I'FROM'0'TO'N'DO'SS'PLUS'(YY[I]^2);
      PRINT((NEWLINE,"CORRECTION IT. NO.",Q1,
        "NORM OF COR. ",SQRT(SS/(N+1))))
    'FI';
    PRINT((NEWLINE,"EXACT SOLN. IS:",NEWLINE));
    'FOR'I'FROM'0'TO'N'DO'PRINT((EXACTY(X[I])));
    PRINT((NEWLINE,NEWLINE,NEWLINE));
    YY:=Y
  'END' ;
  PRINT(("*****",NEWL
INE))
    'END' 'C'OF INNER LOOP 'C'
    'END' 'C'OF LOOP WHICH INCREASES R3 'C'
    'END' 'C' OF LOOP WHICH READS N 'C'
    'END' 'C' OF MAIN LOOP 'C'
    'END' 'C' OF PROBNO LOOP 'C'
  'END'
'FINISH'

```