This item is held in Loughborough University's Institutional Repository (https://dspace.Iboro.ac.uk/) and was harvested from the British Library's EThOS service (http://www.ethos.bl.uk/). It is made available under the following Creative Commons Licence conditions.

Attribution-NonCommercial-NoDerius 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:
BY:
Attribution. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes,

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Leqal Code (the full license).
Disclaimer 민

For the full text of this licence, please go to:

## COMPUTATIONAL AND ALGORITHMIC TECHNIQUES

## FOR THE SOLUTION OF ELLIPTIC AND PARABOLIC

## PARTIAL DIFFERENTIAL EQUATIONS

IN TWO AND THREE SPACE DIMENSIONS
by

Elias Anastasios Lipitakis, B.Sc.

A Doctoral Thesis<br>Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of the Loughborough University of Technology

February, 1978.

Supervisor: Professor D.J. Evans, Ph.D.,D.Sc. Department of Computer Studies.
© by Elias Anastasios Lipitakis, 1978.

## DECLARATION


#### Abstract

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.


DEDICATED TO

Christina and Alexandra

## ACKNOWLEDGEMENTS


#### Abstract

The author wishes to express his deep gratitude and appreciation to Professor D.J. Evans, Head of the Department of Computer Studies for introducing him to the subject as well as for his continuous advice, guidance and encouragement at every stage of this research work.


He also expresses his thanks to Dr. A. Benson, for valuable comments; to the staff of the Computing laboratory of Loughborough University of Technology, where the bulk of the computing was carried out, for their assistance and co-operation; and to Miss J.M. Briers for her expert typing of this thesis.

Finally, grateful acknowledgements are due to my Parents, Anastasios and Alexandra Lipitakis, and Father-in-Law, D.S. Georgacopoulos, for their affection, encouragement and help; and to my Wife, Christina, and Daughter, Alexandra, for more than can be expressed.

## CONTENTS

PAGE
CHAPTER 1: BASIC MATHEMATICAL CONCEPTS
1.1 Partial Differential Equations - Basic Concepts ..... 1
1.2 Finite Difference Approiimations to Derivatives ..... 5
1.3 Basic Matrix Properties and Concepts ..... 10
1.4 Vector and Matrix Norms ..... 13
1.5 Eigenvalues of a Matrix ..... 15
1.6 Basic Definitions ..... 16
CHAPTER 2: ELLIPTIC AND PARABOLIC EQUATIONS IN TWO AND THREESPACE DIMENSIONS - STANDARD METHODS OF SOLUTION
Section A: The Elliptic Problem
2.1 Introduction ..... 18
2.2 The Model Problems ..... 23
2.3 Some Point Iterative Methods ..... 26
2.4 The Conjugate Gradient Method ..... 40
Section B: The Parabolic Problem
2.5 A Parabolic 2D-Example ..... 45
2.6 Derivation of Finite Difference Formulae ..... 46
2.7 Explicit and Implicit Methods ..... 47
2.8 A Parabolic 3D-Example ..... 54
CHAPTER 3: BASIC ALGORITHMS FOR TWO AND THREE DIMENSIONAL P.D.E.'S
3.1 Introduction58
3.2 The Algorithm for the solution of large, un- symmetric, quindiagonal, sparse linear systems (The LUBOT Algorithm) ..... 58
3.3 The Normalized Algorithm for the solution of large, symmetric, quindiagonat, sparse linear systems (The NORMBAND Algorithm) ..... 65
3.4 The Algorithm for the solution of large, un- symmetric, seven diagonal, sparse linear systems (The Lubot-3D Algorithm) ..... 67
3.5 The Normalized Algorithm for the solution of large, symmetric, seven diagonal, sparse linear systems (The NB3D Algorithin) ..... 75
CHAPTER 4: APPROXIMATE ALGORITHMS FOR TWO AND THREE DIMENSIONAL P.D.E.'S
4.1 Introduction ..... 82
4.2 The Approximate Algorithm for the solution of large, unsymmetric, quindiagonal, sparse linear systems (The ALUBOT Algorithm) ..... 82
4.3 The Approximate Normalized Algorithm for the solution of large, symmetric, quindiagonal, sparse linear systems (The NOBAR Algorithm) ..... 85
4.4 The Approximate Algorithm for the solution of large, unsymmetric seven diagonal, sparse linear systems (The ALUBOT-3D Algorithm) ..... 89
4.5 The Approximate Normalized Algorithm for the solution of large, symmetric, seven diagonal, sparse linear systems (The NOBAR-3D Algorithm) ..... 94
CHAPTER 5: NORMALIZED IMPLICIT CONJUGATE GRADIENT METHODS
5.1 Introduction ..... 100
5.2 The Optimum Value of the "Fill-in" Parameter r for the 2D-Model Problem ..... 100
5.3 The Conjugate Gradient Method as an Iterative Method ..... 108
5.4 The Normalized Implicit Conjugate Gradient Method ..... 110
5.5 Computational Work and Experimental Results in Two and Three Space Dimensions ..... 114
CHAPTER 6: NORMALIZED SECOND ORDER METHODS
6.1 Introduction ..... 125
6.2 The Conjugate Gradient Method as a Second Degree Method ..... 125
6.3 The Normalized Implicit Conjugate Gradient Method as a Second Degree Method ..... 127
6.4 A Semi-Empirical Procedure for the Normalized Conjugate Gradient Second Degree Methods ..... 138
6.5 Derivation of Standard Normalized Implicit Methods and Standard Implicit Methods ..... 144
CHAPTER 7: THREE DIMENSIONAL ELLIPTIC AND PARABOLIC PROBLEMS
7.1 Introduction ..... 150
Section A: The Elliptic 3D-Problem
7.2 Statement of the Problem and Formation of the Difference Equations ..... 150
7.3 Computational Results ..... 152
Section B: The Parabolic 3D-Problem
7.4 Statement of the Problem and Formation of the Difference Equations ..... 156
7.5 Analytical Representation of the Parabolic 3D Problem ..... 160
7.6 Computational Results ..... 161
7.7 Concluding Remarks ..... 172
CHAPTER 8: ON THE SOLUTION OF MILDLY NON-LINEAR ELLIPTIC P.D.E'S
8.1 Introduction ..... 174
8.2 Linearization and Quasi-linearization Methods ..... 174
8.3 Formation of the Mildly Non-linear Elliptic Difference Equations ..... 176
8.4 Generalized Linear Methods: Derivation of Newton-LUBOT and Newton-ALUBOT Methods ..... 179
8.5 Iterative Procedures and Numerical Results ..... 181
CHAPTER 9: CONCLUDING REMARKS AND DISCUSSION ..... 191
REFERENCES ..... 193
APPENDIX 1: A TEST PROBLEM ..... 202
APPENDIX 2: PROGRAMS ..... 206

Chapter 1

BASIC MATHEMATICAL CONCEPTS

### 1.1 PARTIAL DIFFERENTIAL EQUATIONS - BASIC CONCEPTS

The numerical treatment of many problems in Mathematical Physics or Engineering, involving the rates of change of unknown quantities (derivatives) with respect to two, three or more independent variables, leads to the solution of a Partial Differential Equation (P.D.E.) or a set of such equations.

Let R denote a bounded, connected, plane region with a boundary C consisting of one or more differential curves and $\gamma(x, y)$ a function defined on $C$. Let $U(x, y)$ be a function continuous in $R+C$, twice differentiable in $R$ and satisfying in $R$ the general second order P.D.A.

$$
\begin{equation*}
a \frac{\partial^{2} U}{\partial x^{2}}+b \frac{\partial^{2} U}{\partial x \partial y}+c \frac{\partial^{2} U}{\partial y^{2}}+d \frac{\partial U}{\partial x}+e \frac{\partial U}{\partial y}+f . U=h \tag{1.1.1}
\end{equation*}
$$

and on $C$ the Dirichlet condition

$$
\begin{equation*}
U(x, y)=\gamma(x, y) \tag{1.1.2}
\end{equation*}
$$

(Alternatively on $C$, the inward normal derivative $\frac{\partial U}{\partial \zeta}$ (Neumann condition) or a linear combination of $U$ and $\frac{\partial U}{\partial \zeta}$ (Robbins condition) may be specified).

The exact solution $U(x, y)$ to the above P.D.E. in region $R$ satisfies the equation (1.1.1) at every point in $R$ and matches the 'boundary conditions' on $C$. The coefficients of (1.1.1) may be constants (including the value zero) or functions of independent variables $x$ and $y$ (linear case) or of the dependent variable $U$ and its first derivatives (non-linear case).

Equations of form (1.1.1) are conventionally classified with respect to the sign of the quantity of discriminant $\Delta \equiv \mathrm{b}^{2}-4 \mathrm{ac}$. Specifically, it is defined to be elliptic when $\quad b^{2}-4 a c<0$
parabolic when $b^{2}-4 a c=0$
hyperbolic when $b^{2}-4 a c>0$
for all $x, y, U$ in the region $R$ under consideration.

Typical and respective examples are:

$$
\begin{array}{ll}
\frac{\partial^{2} U}{\partial x^{2}}+\frac{\partial^{2} U}{\partial y^{2}}= \begin{cases}g(x, y), & \text { Poisson's equation } \\
0, & \text { Laplace's equation }\end{cases} \\
\frac{\partial U}{\partial t}-C_{1}^{2} \frac{\partial^{2} U}{\partial x^{2}}=0, & \begin{array}{l}
\text { Diffusion or Heat conduction } \\
\text { equation }
\end{array} \\
\frac{\partial^{2} U}{\partial t^{2}}-C_{2}^{2} \frac{\partial^{2} U}{\partial x^{2}}=0, & \text { Wave equation }
\end{array}
$$

If the quantity $b^{2}-4 a c$ depends only upon $(x, y)$, the type of equation at a point is completely determined by the co-ordinates of the point. When the coefficients of (1.1.1) depend upon the dependent variable $U$ and its first derivatives, the nature of the equations at a point $(x, y)$ depends not only upon the location of that point, but also upon the behaviour of the solution itself at that point. For example,
(a) The equation

$$
\begin{equation*}
(1-y) \frac{\partial^{2} U}{\partial x^{2}}+2 x \frac{\partial^{2} U}{\partial x \partial y}+(1+y) \frac{\partial^{2} U}{\partial y^{2}}=h \tag{1.1.5}
\end{equation*}
$$

is elliptic inside the unit circle $x^{2}+y^{2}=1$, parabolic on the boundary and hyperbolic outside the circle.
(b) The nature of the equation

$$
\begin{equation*}
\frac{\partial^{2} U}{\partial x^{2}}+U \frac{\partial^{2} U}{\partial y^{2}}=h \tag{1.1.6}
\end{equation*}
$$

at a point depends upon the sign of $U$ at that point and, in any specific problem, depends also upon the side conditions which serve to complete the formulation of that problem.

The two elliptic P.D.E's (1.1.4a)-(1.1.4b) are generally associated with steady-state or equilibrium problems. For example,
(i) the electric potential, associated with a two dimensional electron distribution of charge density, satisfies Poisson's
equation and is the mathematical expression of the total electric flux through any closed surface is equal to the total charge enclosed, whereas,
(ii) the velocity potential for the steady $f 10 w$ of incompressible nonviscous fluid satisfies Laplace's equation, stating that the rate at which such fluid enters any given region is equal to the rate at which it leaves it. Such problems are referred to as 'boundary value problems', since the dependent variable is usually specified on the boundary of the region under consideration, e.g. see system (1.1.1) (1.1.2). Boundary value problems frequently occur in applications such as reservoir problems, reactor studies, numerical weather forecasting, etc.

Parabolic and hyperbolic P.D.E's, in general, result from diffusion, equalization or oscillatory processes and the usual independent variables are time and space.

The simplest parabolic P.D.E. given by (1.l.4c) governs the flow of heat in a thin homogeneous bar or rod, assuming that the radiation and convection are neglected. The temperature distribution along the bar or rod is usually known at some instant in time. This is termed the 'initial condition'. The boundary conditions consist of appropriate end conditions, which are either the temperature given at two ends of the bar or rod or some measure of the diffusion from the ends. Such problems are called 'initial-boundary value problems' (sometimes only the terminology 'initial value problem' is used).

A major class of hyperbolic equations arise from vibration problems or those in which discontinuities persist in time. In the case of the simplest hyperbolic equation (1.l.4d), giving the transverse displacement at a given distance from one end of a
vibrating string of given length after a certain time, the initial conditions, usually $U$ and $\frac{\partial U}{\partial t}$, are given at some instant in time and the boundary conditions are given on two lines $(x=a, b)$.

For the solution of problems with arbitrarily shaped regions and general boundary conditions, an exact solution to a given P.D.E. is not usually possible to determine. Only in the simplest cases can a solution be analytical either in implicit form or that involving a finite formula.

The 'approximate methods', which have been developed to tackle this problem, can be divided into two groups:
(i) Analytical approximate methods, in which the approximate solution is in an analytical form, e.g. the truncating of an infinite series. They consist mainly of Fourier's method of solving boundary value problems in P.D.E's, where the exact solution is in the form of a certain infinite series and the approximate solution is the sum of the first few terms.
(ii) Numerical approximate methods, in which approximate values of the required solution can be found at various points of a region under consideration in a tabular form. The most widely used numerical method for solving P.D.E's, applicable to both linear and non-linear problems, is the 'finite difference' method, which will be exclusively used throughout this thesis. The method of 'characteristics' for solving hyperbolic equations and sets of equations is also essentially a finite difference method, only in this method the P.D.E. or set of P.D.E's is first reduced to an equivalent set of Ordinary Differential Equations (O.D.E's) or linear algebraic equations, which is then solved by numerical direct or iterative methods.

### 1.2 FINITE DIFFERENCE APPROXIMATIONS TO DERIVATIVES

We consider, without any loss of generality, that the problem is to solve the elliptic equation (1.1.4a) with independent variables $x, y$ in a connected region $R$, in the $X-Y$ plane. Let $\bar{R}=R+C$ be the closure of the considered region $R$ with boundary $C$. We overlay $\bar{R}$ with a system of rectangular mesh lines formed by two families of equally spaced straight lines, which are parallel to coordinate axes viz.,

$$
\begin{array}{ll}
x=x_{0}+i h & , \quad i=0, \pm 1, \pm 2, \ldots \\
y=y_{0}+j h & , \quad j=0, \pm 1, \pm 2, \ldots
\end{array}
$$

where $\left(x_{0}, y_{0}\right)$ is any conveniently chosen origin for the "mesh coordinates" i,j.

The intersection points are called mesh points (grid or lattice points, nodes, pivots) and the distances between the parallel lines are called mesh sizes (lengths).

A mesh point $P_{i, j}$ is called 'regular' if the four adjacent points $P_{i+1, j}, P_{i-1, j}, P_{i, j+1}, P_{i, j-1}($ see Fig.1.1) are also mesh points contained in $\bar{R}$; otherwise $P_{i, j}$ is called 'irregular'.


An approximate solution to the differential equation is then found at the $n$ mesh points $P_{1,1}, P_{1,2}, \ldots \ldots, P_{i, j}, \ldots$ and the problem is now reduced to the solution of n-algebraic equations (linear, if the differential equations are linear) involving approximate values of $U$ at the $n$-mesh points internal to $C$.

## Taylor Series Expansion

This method is probably the best known of all for deriving finite difference approximations.

We assume that the solution of (1.1.1) has the required continuous derivatives of higher order in a sufficiently large neighbourhood about a point ( $\mathrm{x}, \mathrm{y}$ ) and a constant mesh size h , then by Taylor's theorem the values at the surrounding mesh points can be determined, viz.,

$$
\begin{align*}
& U(x \pm h, y)=U(x, y) \pm h \frac{\partial U}{\partial x}+\frac{h^{2}}{2} \frac{\partial^{2} U}{\partial x^{2}} \pm \frac{h^{3}}{3!} \frac{\partial^{3} U}{\partial x^{3}}+\frac{h^{4}}{4!} \frac{\partial^{4} U}{\partial x^{4}} \pm \ldots  \tag{1.2.1}\\
& U(x, y \pm h)=U(x, y) \pm h \frac{\partial U}{\partial y}+\frac{h^{2}}{2} \frac{\partial^{2} U}{\partial y^{2}} \pm \frac{h^{3}}{3!} \frac{\partial^{3} U}{\partial y^{3}}+\frac{h^{4}}{4!} \frac{\partial^{4} U}{\partial y^{4}} \pm \ldots \tag{1.2.2}
\end{align*}
$$

Combining these formulae we obtain

$$
\begin{align*}
\frac{\partial U}{\partial x} & =\frac{U(x+h, y)-U(x-h, y)}{2 h}+o\left(h^{2}\right)  \tag{1.2.3}\\
\frac{\partial^{2} U}{\partial x^{2}} & =\frac{U(x+h, y)-2 U(x, y)+U(x-h, y)}{h^{2}}+o\left(h^{2}\right), \tag{1.2.4}
\end{align*}
$$

and

$$
\begin{align*}
\frac{\partial U}{\partial y} & =\frac{U(x, y+h)-U(x, y-h)}{2 h}+O\left(h^{2}\right)  \tag{1.2.5}\\
\frac{\partial^{2} U}{\partial y^{2}} & =\frac{U(x, y+h)-2 U(x, y)+U(x, y-h)}{h^{2}}+O\left(h^{2}\right) \tag{1.2.6}
\end{align*}
$$

where the 0 -notation denotes that if $S$ is any set and $f, \phi$ be real or complex functions defined on $S$ then,

$$
\begin{equation*}
f(t)=O(\phi(t)) \text { as } t \rightarrow a \text { with } t, a \varepsilon S \tag{1.2.7a}
\end{equation*}
$$

if there exists a positive number $P$, such that

$$
\begin{equation*}
|f(t)| \leqslant P|\phi(t)| \tag{1.2.7b}
\end{equation*}
$$

for all $t$ sufficiently close to $a$.
Let us denote $U(x, y)$ by $U_{i, j}$, for a general mesh point $(x, y)=(i h, j h)$. Then, Poisson's equation

$$
\frac{\partial^{2} U}{\partial x^{2}}+\frac{\partial^{2} U}{\partial y^{2}}=g(x, y)
$$

using the formulae $(1.2 .4),(1.2 .6)$, can be replaced at the point $\left(x_{i}, y_{j}\right)$ by

$$
\begin{align*}
& \frac{1}{h^{2}}\left\{U_{i+1, j}+U_{i-1, j}+U_{i, j+1}+U_{i, j-1}-4 U_{i, j}\right\}=g_{i, j}+ \\
&  \tag{1.2.8}\\
& +\frac{h^{2}}{12}\left\{\frac{\partial^{4} U^{4}}{\partial x^{4}}+\frac{\partial^{4} U}{\partial y^{4}}\right\}_{i, j}+\ldots
\end{align*}
$$

or equivalently,

$$
\begin{align*}
4 U_{i, j}-U_{i+1, j}-U_{i-1, j}-U_{i, j+1}-U_{i, j-1} & =-h^{2} g_{i, j}- \\
& -\frac{h^{4}}{12}\left\{\frac{\partial^{4} U}{\partial x^{4}}+\frac{\partial^{4} U}{\partial y^{4}}\right\}_{i, j}-\ldots \tag{1.2.9}
\end{align*}
$$

As "local truncation error" of the above formula are defined the terms on the right hand side of (1.2.9), excluding $\left\{-h^{2} g_{i, j}\right\}$, while the term $O\left(h^{4}\right)$ is defined as the 'principal part' of this error. The local truncation error in (1.2.9) is usually neglected and then scanning over the mesh points with such a formula, a set of simultaneous equations can be obtained, with unknown functions $u_{i, j}$, the finite difference approximation of the exact solution $U_{i, j}$ at the point (ih,jh).

The above set of equations can be written in matrix form as

$$
\begin{equation*}
\text { A. } \underline{u}=\underline{s} \tag{1.2.10}
\end{equation*}
$$

where $\underline{s}$ is a vector composed of the known values $-h^{2} g_{i, j}$ plus values of the finite difference approximate solution $u_{i, j}$ givin on the boundary $C$. Note that if $\left(x_{i}, y_{i}\right) \varepsilon C$ then $u_{i, j}=U_{i, j}$.

Assuming that the region under consideration is the unit
square and there are $n^{2}$ internal mesh points, then the coefficient matrix $A$ of (1.2.10) is of order $n^{2}$, while $\underline{u}$ and $\underline{s}$ are $\left(n^{2} \times 1\right)$ column vectors. From (1.2.9) it can be easily seen that the better accuracy is obtained as the mesh size $h$ tends to zero.

We consider now the following initial-boundary value problem defined by

$$
\begin{equation*}
\frac{\partial U}{\partial t}=\frac{\partial^{2} U}{\partial x^{2}} \tag{1.2.11}
\end{equation*}
$$

in the region

$$
0 \leqslant x \leqslant a ; \quad t \geqslant 0
$$

with initial conditions

$$
\begin{equation*}
U(x, 0)=f(x) ; \quad 0 \leqslant x \leqslant a \tag{1.2.11a}
\end{equation*}
$$

and boundary conditions consisting of

$$
\begin{equation*}
\mathrm{U}(0, \mathrm{t})=\mathrm{g}_{0}(\mathrm{t}), \quad \mathrm{U}(\mathrm{a}, \mathrm{t})=\mathrm{g}_{\mathrm{a}}(\mathrm{t}) ; \quad \mathrm{t} \geqslant 0 \tag{1.2.11b}
\end{equation*}
$$

We cover the region under consideration by a rectangular grid with spacings $h_{x}$ and $\Delta t$ in the $X$ and $t$ (time)-directions respectively (see Figure 1.2).


FIGURE 1.2
Let $U_{i, j}^{(k)}$ and $u_{i, j}^{(k)}$ denote the exact and finite difference solution respectively of equation (1.2.11) at the point (ih,$k \Delta t)$. Then, using Taylor series expansions, a simple replacement of (1.2.11)
leads to the following difference scheme:

$$
\begin{equation*}
\frac{U_{i}^{(k+1)}-U_{i}^{(k)}}{\Delta t}=\frac{U_{i+1}^{(k)}-2 U_{i}^{(k)}+U_{i-1}^{(k)}}{h_{x}^{2}}+O\left(\Delta t+h_{x}^{2}\right) \tag{1.2.12}
\end{equation*}
$$

which can be written as

$$
\begin{equation*}
u_{i}^{(k+1)}=(1-2 r) U_{i}^{(k)}+r\left(U_{i+1}^{(k)}+U_{i-1}^{(k)}\right)+0\left(\Delta t^{2}+\Delta t \cdot h_{x}^{2}\right) \tag{1.2.13}
\end{equation*}
$$

where $r=\frac{\Delta t}{h_{x}^{2}}$ is called the 'mesh ratio'. Note that $U_{i}^{(k+1)}$ is expressed in (1.2.13) solely in terms of the values of $U$ at the $k^{\text {th }}$-time level. Such schemes are called explicit (open) whereas schemes involving more than one point at the $(k+1)^{\text {th }}$ time level are called implicit (closed). Assuming that the local truncation error in (1.2.13) is neglected, then scanning over each mesh point in turn, in the interval $0<x<a$, we obtain a set of simultaneous equations, which can be expressed in matrix notation as

$$
\begin{equation*}
\underline{u}^{(k+1)}=A \cdot \underline{u}^{(k)}, \quad k \geqslant 0 . \tag{1.2.14}
\end{equation*}
$$

An analogous implicit scheme could be written as

$$
\begin{equation*}
A_{1} \cdot \underline{u}^{(k+1)}=A_{2} \cdot \underline{u}^{(k)}, \quad k \geqslant 0 . \tag{1.2.15}
\end{equation*}
$$

The numerical integration of the differential equation (1.2.11) is obtained by a step by step procedure represented by both equations (1.2.14),(1.2.15). This process is continued until the final solution $\underline{u}^{r}$ is obtained at some time $T=r . \Delta t$.

It can be seen that the application of finite difference method to the numerical solution of P.D.E's leads to the solution of a system of simultaneous equations, such as (1.2.10), (1.2.15). The coefficient matrix of these systems is usually a large "sparse" (with many zero elements) matrix, with a certain number of non-zero elements in each row, and with characteristic properties such as irreducibility, diagonal dominance, positive definiteness etc., which will be defined in the next section.

Note that the structure of the coefficient matrix is of major importance for the method of solution of the resulting system of simultaneous equations and several techniques for this purpose will be presented in Chapters 3 and 4.

In the following section a basic knowledge of matrix and linear algebra theory has been presupposed, with the subject being found in [3], [26], [29], [57], [61].

### 1.3 BASIC MATRIX PROPERTIES AND CONCEPTS

Definition 1.3.1
The matrix $A=\left(a_{i, j}\right)$ of order $n$ is said to be 'irreducible' if $\mathrm{n}=1$ or if $\mathrm{n}>1$ and given any two non-empty disjoint subsets S and T of $W$, the set of the first $n$-positive integers, such that $S+T=W$, there exists some $a_{i, j} \neq 0$ such that $i \varepsilon S$ and $j \varepsilon T$.

The following alternative definition is given in [67,(p.37)]: Definition 1.3.2

The matrix $A$ is irreducible if and only if there does not exist a permutation matrix $P$ such that $P^{-1} A P$ has the form

$$
\mathrm{P}^{-1} \mathrm{AP}=\left[\begin{array}{ll}
\mathrm{A}_{1} & 0 \\
\mathrm{~A}_{2} & \mathrm{~A}_{3}
\end{array}\right]
$$

where $A_{1}$ and $A_{3}$ are square matrices and where all elements of 0 vanish.

Note that a permutation matrix is a square matrix which in each row and each column has some one entry unity and all others zero.

Definition 1.3.3
An $(n \times n)$ matrix $A=\left(a_{i, j}\right)$ is said to be "diagonally dominant"
if

$$
\begin{equation*}
\left|a_{i, i}\right| \geqslant \sum_{\substack{j=1 \\ j \neq i}}^{n}\left|a_{i, j}\right|, \text { for all } i \varepsilon[1, n] \tag{1.3.1}
\end{equation*}
$$

The matrix $A$ is 'strictly diagonally dominant' if strict inequality is valid for all $\mathrm{i} \varepsilon[1, \mathrm{n}]$ in relationship (1.3.1).

If $A$ is irreducible and diagonally dominant, with strict inequality in (1.3.1) for at least one value of $i$, then $A$ is said to be "irreducibly diagonally dominant".

Definition 1.3.4
If $A$ is real matrix and $\underline{x}$ is complex, then $A$ is said to be 'positive definite' matrix if

$$
(\underline{x}, A \underline{x})>0, \quad \text { for all } \underline{x} \neq \underline{0} .
$$

(Note that if $\underline{x}, \underline{y}$ are complex, then

$$
(\underline{x}, \underline{y})=\sum_{i=1}^{n} x_{i} \bar{y}_{i} \text {, where } \bar{y}_{i} \text { is the }
$$

complex conjugate of $y_{i}$. The quantity ( $\underline{x}, \underline{y}$ ) is called the
inner-product of $\underline{x}, \underline{y}$ ).
Definition 1.3.5
If $A$ is real matrix and $\underline{x}$ is complex, then $A$ is said to be 'non-negative (or positive semi)-definite' matrix if

$$
(\underline{x}, A \underline{x}) \geqslant 0 \quad \text { for all } \underline{x} \neq 0
$$

with equality for at least one $\underline{x} \neq \underline{0}$.
The following theorem, given without proof, can be used as an alternative definition of positive/non-negative definite matrix.

Theorem 1.3.1
symmetric
A real matrix is positive/non-negative definite if and only
all its eigenvalues are positive/(non-
negative, with at least one eigenvalue equal to zero).
symmetric
Let $A$ be a positive definite matrix. The matrix $A$ can be
wrıtten as $A=\mathrm{GJG}^{-1}$, where $J$ is a positive diagonal matrix and $G$ can be taken to be an orthogonal matrix i.e., $G^{T}=G^{-1}$ (see [61](p.16)).

Then, by Theorem 1.3.1 we obtain that

$$
\begin{equation*}
A^{\frac{1}{2}}=G J^{\frac{1}{2}} G^{-1} \tag{1.3.2}
\end{equation*}
$$

is a positive definite, provided that $J^{\frac{1}{2}}$ is the diagonal matrix whose elements are the positive square roots of the corresponding elements of $J$. It can be easily seen that $\left(A^{\frac{1}{2}}\right)^{2}=\left(G J^{\frac{1}{2}} G^{-1}\right)^{2}=A$. Theorem 1.3.2

A real symmetric matrix $A$ is positive/non-negative definite if and only if it can be written in the form $A=G^{T} G$, where $G$ is some non-singular/singular matrix.

Proof
(I) Let us assume that $A=G^{T} G(\operatorname{det} G \neq 0)$.

Then, for all $\underline{x} \neq \underline{0}$, we have

$$
\underline{x}^{T} A \underline{x}=x^{T} G^{T} \underline{x}=(G \underline{x})^{T} G \underline{x}>0
$$

Hence, by Definition 1.3.4, the matrix $A$ is positive definite.
(II) Let A be real and positive definite matrix.

Since $A=A^{\frac{1}{2}} A^{\frac{1}{2}}$ and $A^{\frac{1}{2}}$ is symmetric, we have that $A=\left(A^{\frac{1}{2}}\right)^{T} A^{\frac{1}{2}}$, where $A^{\frac{1}{2}}$ is defined in (1.3.2) and is also positive definite. Hence $\operatorname{det} A^{\frac{1}{2}} \neq 0$ and then it can be easily seen that the choice $G=A^{\frac{1}{2}}$ leads to the required conclusion.

If $A$ is non-negative definite matrix, the proof can be similarly obtained.

Definition 1.3.6
A real $(n \times n)$ matrix $A=\left(a_{i, j}\right)$, with all its off-diagonal elements non-positive ( $a_{i, j} \leqslant 0$, for all $i \neq j$ ) is said to be:
(i) a "Stieltjes matrix" if $A$ is symmetric and positive definite, (ii) a "M-matrix" if $A$ is non-singular and $A^{-1} \geqslant 0$.

### 1.4 VECTOR AND MATRIX NORMS

Definition 1.t.1
If $\underline{x}=\left(x_{1}, x_{2}, \ldots, x_{n}\right)$ is $n$-component vector, then the following quantities are defined as the $L_{1}, L_{2}$ and $L_{\infty}$ norms of $\underline{x}$ respectively:

$$
\begin{align*}
& \|\underline{x}\|_{1}=\sum_{i=1}^{n}\left|x_{i}\right|  \tag{1.4.1}\\
& \|\underline{x}\|_{2}=\left(\sum_{i=1}^{n}\left|x_{i}\right|^{2}\right)^{\frac{1}{2}} \quad \text { (Euc1idean or } L_{2} \text { Norm or length of } \underline{x} \text { ) }  \tag{1.4.2}\\
& \|\underline{x}\|_{\infty}=\max _{i \in[1, n]}\left|x_{i}\right| \quad \text { (Maximum or Uniform Norm). } \tag{1.4.3}
\end{align*}
$$

Definition 1.4.2
A matrix norm is said to be 'compatible' with a vector norm $||\underline{x}||$ if

$$
\|\mathrm{Ax}\| \leq\|\mathrm{A}\| \cdot\|\underline{\mathrm{x}}\| \mid, \text { for all } \underline{x} \neq \underline{0} .
$$

Definition 1.4.3
A matrix norm is said to be "subordinate" to the corresponding vector norm, if it can be constructed in the following manner:

$$
\begin{equation*}
\|A\|=\sup _{\underline{x} \neq \underline{0}} \frac{\|A \underline{x}\|}{\|\underline{x}\|} \tag{1.4.4}
\end{equation*}
$$

or equivalently:

$$
\begin{equation*}
\|\mathrm{A}\|=\sup _{\|\underline{y}\|=1}\|\operatorname{Ay}\| \tag{1.4.5}
\end{equation*}
$$

with the matrix norm satisfying the compatibility relation $||A \underline{y}|| \leqslant||A|| .||\underline{y}||$.

Definition 1.4.4
Let $A$ be an $n \times n$ matrix with eigenvalues $\lambda_{i}, i \varepsilon[1, n]$ then the "spectral radius" of $A$ is defined as

$$
\rho(A)=\max _{i \varepsilon[1, \mathrm{n}]}\left|\lambda_{\mathrm{i}}\right| .
$$

It can be easily shown that for any matrix and any norm, we have

$$
\begin{equation*}
\|A\| \geqslant \rho(A) . \tag{1.4.6}
\end{equation*}
$$

## Proof

Let us assume that $A \underline{x}=\lambda_{i} \underline{x}, i \varepsilon[1, n]$ is the equation associating the non-zero eigenvector $\underline{x}$ with the eigenvalues $\lambda_{i}$ of $A$. Then, from Definition 1.4.2 and for any eigenvalue $\lambda_{i}$ of $A$, we have

$$
|\lambda| \cdot||\underline{x}||=\|\lambda \underline{x}| |=\| \underline{A} \underline{x}| | \leqslant||A|| \cdot| | \underline{x}| |,
$$

which leads to (1.4.6).

Definition 1.4.5
Let $A$ be a square matrix of order $n$. Then, the subordinate norms associated with $\mathrm{L}_{1}, \mathrm{~L}_{2}$ and $\mathrm{L}_{\infty}$ vector Norms are defined as:

$$
\begin{array}{ll}
\|A\|_{1}=\max _{j \varepsilon[1, n]} \sum_{i=1}^{n}\left|a_{i, j}\right| & \text { (Maximum absolute columis. sum) } \\
\|A\|_{2}=\left\{\rho\left(A^{T} A\right)\right\}^{\frac{1}{2}} & \text { (Spectral Norm) } \\
\|A\|_{\infty}=\max _{i \varepsilon[1, n]} \sum_{j=1}^{n}\left|a_{i, j}\right| & \text { (Maximum absolute row sum) } \tag{1.4.9}
\end{array}
$$

It can be easily shown from (1.4.8) that if $A$ is a symmetric $\mathrm{n} \times \mathrm{n}$ matrix, then we get

$$
\begin{equation*}
\|A\|_{2}=\rho(A) \tag{1.4.10}
\end{equation*}
$$

Proof
Given that A is symmetric we have,

$$
\begin{equation*}
\|A\|_{2}^{2}=\rho\left(A^{T} \cdot A\right)=\rho\left(A^{2}\right)=\rho^{2}(A), \tag{1.4.11}
\end{equation*}
$$

hence (1.4.10) follows.
Definition 1.4.6
Let $A=\left(a_{i, j}\right)$ be a positive definite $n \times n$ matrix and $x=\left(x_{1}, x_{2}, \ldots, x_{n}\right)$ be a n-component vector.

Then, the A-Norm of $\underline{x}$ is defined as

$$
\|\underline{x}\|_{A}=(A \underline{x}, \underline{x})^{\frac{1}{2}}
$$

### 1.5 EIGENVALUES OF A MATRIX

Theorem 1.5.1
Let $A$ le an arbitrary $n \times n$ complex matrix, then all the eigenvalues of A lie in the union of the disks

$$
\begin{equation*}
\left|z-a_{i, i}\right| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n}\left|a_{i, j}\right|, \quad i \varepsilon[1, n] \tag{1.5.1}
\end{equation*}
$$

The above theorem is due to Gerschgorin [30] and its proof may be found in [57],[35],[54].

## Corollary 1

If $A=\left(a_{i, j}\right)$ is an arbitrary $n \times n$ complex matrix and we have

$$
\begin{equation*}
v_{1}=\max _{i \varepsilon[1, n]} \sum_{j=1}^{n}\left|a_{i, j}\right| \tag{1.5.2}
\end{equation*}
$$

$$
\begin{equation*}
v_{2}=\max _{j \varepsilon[1, n]} \sum_{i=1}^{n}\left|a_{i, j}\right| \tag{1.5.3}
\end{equation*}
$$

then

$$
\begin{equation*}
\rho(A) \leqslant \min \left(v_{1}, v_{2}\right) \tag{1.5.4}
\end{equation*}
$$

The condition (1.5.4) is a direct consequence of the fact that $A$ and $A^{T}$ have the same eigenvalues.

Theorem 1.5.2
Let $A=\left(a_{i, j}\right)$ be an $n \times n$ strictly or irreducibly diagonally dominant complex matrix. Then the matrix $A$ is non-singular and if $a_{i, i}, i \varepsilon[1, n]$ are positive real numbers the eigenvalues $\lambda_{i}$ of A satisfy

$$
\begin{equation*}
\operatorname{Re}\left\{\lambda_{i}\right\}>0, \quad \mathrm{i} \varepsilon[1, \mathrm{n}] \tag{1.5.5}
\end{equation*}
$$

Proof
The proof can be easily obtained from Theorem 1.5.1 and can be found in [57](p.23).

Corollary 2
If $A=\left(a_{i, j}\right)$ is a symmetric $n \times n$ strictly or irreducibly
diagonally dominant matrix with $\mathrm{a}_{\mathrm{i}, \mathrm{i}}>0$, $\mathrm{i} \varepsilon[1, \mathrm{n}]$, then A is positive definite.

## Proof

Since a symmetric matrix has real eigenvalues the result follows from Theorems 1.3.1 and 1.5.2.

A useful lower bound for the smallest eigenvalue $\lambda_{1}$ of the matrix $A=\left(a_{i, j}\right)$ can be obtained from the following result due to Collatz [11],

$$
\begin{equation*}
\lambda_{1} \geqslant \min _{i \in[1, n]}\left(\left|a_{i, i}\right|-\sum_{\substack{j=1 \\ j \neq i}}^{n}\left|a_{i, j}\right|\right) . \tag{1.5.6}
\end{equation*}
$$

### 1.6 BASIC DEFINITIONS

Definition 1.6.1
Let $x_{i}, i=1,2, \ldots, n$ be a set of vectors in a unitary space $T$.
If $A$ is a positive definite operator on $T$, then the set of vectors
$\underline{x}_{i}$ is said to be "A-conjugate" or "A-orthogonal" if

$$
\begin{equation*}
\left(\underline{A x}_{i}, \underline{x}_{j}\right)=0, \quad \text { where } i \neq j \tag{1.6.1}
\end{equation*}
$$

Definition 1.6.2
Consider the equation $\mathrm{F}[\underline{\mathrm{x}}]=\mathrm{C}$, where C is a constant, which represents an ellipsoid in the $n$-dimensional space.

Let $\underline{x}=\left(x_{1}, x_{2}, x_{3}, \ldots, x_{n}\right)$ be a point on the surface of the above ellipsoid. Then, assuming that the function F is continuous and differentiable, the vector with components $\frac{\partial F}{\partial x_{i}}, i=1,2, \ldots, n$ is said to be the "Gradient" of F[x] (represented by Grad F or $\nabla F$ ).

At any point P the Gradient vector of F is normal to the surface $F[\underline{x}]=C$, which passes through $P$.

For a scalar point function $\Phi(x, y, z)$ in rectangular coordinates it is

$$
\begin{equation*}
\nabla \Phi=i \frac{\partial \Phi}{\partial x}+j \frac{\partial \Phi}{\partial y}+k \frac{\partial \Phi}{\partial z} \tag{1.6.2}
\end{equation*}
$$

where $i, j, k$ are unit vectors along the $X, Y, Z$ axes.

## Definition 1.6.3

Let $f_{r}\left(u_{1}, u_{2}, \ldots, u_{n}\right), r=1,2, \ldots, n$ be $n$-functions of the variables $u_{1}, u_{2}, \ldots, u_{n}$ and let each of the partial derivatives $\frac{\partial f_{r}}{\partial u_{s}}, r, s=1,2, \ldots, n$ be a continuous function of $u_{1}, u_{2}, \ldots, u_{n}$. Then, the Jacobian matrix $f_{r}^{\prime}$ of the functions $f_{1}, f_{2}, \ldots, f_{n}$ with respect to $u_{1}, u_{2}, \ldots, u_{n}$ is defined by:

$$
f_{r}^{\prime}=\left[\begin{array}{ccccc}
\frac{\partial f_{1}}{\partial u_{1}} & \frac{\partial f_{1}}{\partial u_{2}} & \cdots & \cdot & \cdot  \tag{1.6.3}\\
\frac{\partial f_{2}}{\partial u_{n}} \\
\frac{\partial f_{2}}{\partial u_{1}} & \frac{\partial f_{2}}{\partial u_{2}} & \cdots & \cdot & \frac{\partial f_{2}}{\partial u_{n}} \\
\cdot & \cdot & \cdots & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\dot{f_{n}} & \frac{\partial f_{n}}{\partial u_{1}} & & \cdots & \cdot \\
\partial u_{2} & \cdots & \cdot & \frac{\partial f_{n}}{\partial u_{n}}
\end{array}\right]
$$

## Definition 1.6.4

Let $R$ denote an open region of $n$-dimensional space ( $n=2$ or 3 ) with a boundary $C$ made up of one or more differenciable curves or surfaces.

Then, an operator $L$ is said to be "self adjoint" if for any two sufficiently smooth functions $\phi_{1}, \phi_{2}$ the integral

$$
\begin{equation*}
\iint_{R}\left[\phi_{2} \mathrm{~L}\left(\phi_{1}\right)-\phi_{1} L\left(\phi_{2}\right)\right] d x d y \tag{1.6.4}
\end{equation*}
$$

is a function of the values of $\phi_{1}, \phi_{2}$ and their derivatives on C alone,

Chapter 2

## ELLIPTIC AND PARABOLIC EQUATIONS IN TWO AND THREE

 SPACE DIMENSIONS - STANDARD METHODS OF SOLUTION
## SECTION A: THE ELLIPTIC PROBLEM

### 2.1 INTRODUCTION

We will be concerned first with the application of the method of finite differences to obtain an approximate solution for the 'selfadjoint' two dimensional problem defined by

$$
\begin{array}{r}
\frac{\partial}{\partial x}\left(C_{1}(x, y) \frac{\partial U}{\partial x}\right)+\frac{\partial}{\partial y}\left(C_{2}(x, y) \frac{\partial U}{\partial y}\right)+P(x, y) U(x, y)+Q(x, y)=0,  \tag{2.1.1}\\
(x, y) \in R,
\end{array}
$$

subject to the Dirichlet boundary conditions

$$
\begin{equation*}
U(x, y)=\gamma(x, y), \quad(x, y) \varepsilon C \tag{2.1.1a}
\end{equation*}
$$

where the region $R$ and its boundary $C$ aredefined in section 1.1 . The functions $C_{1}, C_{2}, P, Q$ are assumed to be 'sufficiently smooth' functions (continuous and differentiable) and satisfy in $\bar{R}=R+C$ the conditions

$$
\begin{equation*}
C_{1}(x, y)>0, \quad C_{2}(x, y)>0, \quad P(x, y) \leqslant 0 \tag{2.1.1b}
\end{equation*}
$$

Consequently, from (1.1.3a) it follows that equation (2.1.1) is of elliptic type.

In a similar way as in section 1.2, a network of straight lines of spacings $h_{x}$ and $h_{y}$, parallel to each of the co-ordinate axes, is superimposed over the region $R$. Then, using Taylor series, a discrete averaged centred finite difference approximation for the x-derivative of the form

$$
\begin{align*}
\frac{\partial}{\partial x}\left(C_{1}(x, y) \frac{\partial U}{\partial x}\right) & \simeq C_{1}\left(x+\frac{h_{x}}{2}, y\right)\left[\frac{u\left(x+h_{x}, y\right)-u(x, y)}{h_{x}^{2}}\right]- \\
& -C_{1}\left(x-\frac{h_{x}}{2}, y\right)\left[\frac{u(x, y)-u\left(x-h_{x}, y\right)}{h_{x}^{2}}\right] \tag{2.1.2}
\end{align*}
$$

with analogous expression for $\frac{\partial}{\partial y}\left(C_{2}(x, y) \frac{\partial U}{\partial y}\right)$ can be obtained,
provided that the $(x, y)$ is a regular mesh point. In the case of an irregular mesh point, the Taylor series expansion of $U(x, y)$ about such a point may still be used, giving approximations similar to (1.2.3)-(1.2.6) and (2.1.2). Substitution of the above finite difference approximations for the derivatives in (2.1.1), assuming that $h_{x}=h_{y}=h$, and $u_{i, j}$ denotes $u(i h, j h)$, leads to the following linear five-point finite difference equation at the point (ih,jh)

$$
\begin{equation*}
d_{i, j} u_{i, j}{ }^{-t_{i, j}}{ }_{i, j+1}^{-b_{i, j}} u_{i, j-1}^{-r_{i, j}} u_{i+1, j^{-l}}^{i, j} u_{i-1, j}=s_{i, j} \tag{2.1.2a}
\end{equation*}
$$

where $d_{i, j}, t_{i, j}, b_{i, j}, r_{i, j}, \ell_{i, j}>0$ and $d_{i, j} \geqslant t_{i, j}+b_{i, j}+r_{i, j}+\ell_{i, j}$ for all i,j with equality at interior point if $\mathrm{P}=0$.

Equation (2.1.2a) is generally represented by the 'molecule' (stencil or star) in Figure 2.1.


FIGURE 2.1
which when applied at each regular mesh point, assuming that a fixed labelling is considered, yields a set of inhomogeneous, linear, simultaneous, symmetric difference equations, which can be expressed in matrix notation as

$$
\begin{equation*}
\mathrm{A} \underline{u}=\underline{s} \tag{2.1.3}
\end{equation*}
$$

where the vectors $\underline{u}$ and $\underline{s}$ consist of the unknown approximate solution $u_{i, j}$ and the known boundary value plus the quantity $h^{2} Q(x, y)$ respectively.

The coefficient matrix $A$ is a sparse, real, quindiagonal, ( $n \times n$ ) matrix ( n is the number of unknown mesh points) with the following properties:
(i) Symmetric ,
(ii) Positive definite ,
(iii) Diagonally dominant ,
(iv) Irreducible ,
and
(v) It has positive diagonal and non-positive off diagonal elements.

If all the interior mesh points in the region R are regular, then A will be symmetric and by Corollary 2 will be positive definite. Property (v) results from the fact that the diagonal entries of $A$ are given by:

$$
\left[C_{1}\left(x+\frac{h}{2}, y\right)+C_{1}\left(x-\frac{h}{2}, y\right)+C_{2}\left(x, y+\frac{h}{2}\right)+C_{2}\left(x, y-\frac{h}{2}\right)-h^{2} P(x, y)\right]
$$

and the off-diagonal elements consist of the quantities:

$$
-C_{1}\left(x+\frac{h}{2}, y\right),-C_{1}\left(x-\frac{h}{2}, y\right),-C_{2}\left(x, y+\frac{h}{2}\right),-C_{2}\left(x, y-\frac{h}{2}\right),
$$

while properties (iii) and (iv) follow from Definition 1.3.3 and Theorem (p.20) in [57] respectively.

## Three Dimensional Case

We consider now a class of problems defined by the self-adjoint P.D.E. in three space dimensions, namely:

$$
\left.\begin{array}{rl}
\frac{\partial}{\partial x}\left(C_{1}(x, y, z) \frac{\partial U}{\partial x}\right)+\frac{\partial}{\partial y}\left(C_{2}(x, y, z) \frac{\partial U}{\partial y}\right)+\frac{\partial}{\partial z}\left(C_{3}(x, y, z) \frac{\partial U}{\partial z}\right) & + \\
+ & P(x, y, z) U(x, y, z)+Q(x, y, z)
\end{array}\right)=0,
$$

where $C_{1}, C_{2}, C_{3}$ are strictly positive functions, $P$ non-positive and $U(x, y, z) \varepsilon R$, where $R$ is the interior of a compact region subject to the general boundary conditions

$$
\begin{equation*}
a \cdot U+\beta \frac{\partial U}{\partial \zeta}=\gamma \tag{2.1.6}
\end{equation*}
$$

for $U \varepsilon C$, the exterior boundary of $R$.
On C, a and $\beta$ are positive, piecewise continuous, and $\zeta$ denotes the direction of the outward drawn normal. The coefficients of (2.1.5) are assumed to be 'sufficiently smooth' functions.

The region under consideration $R$ is covered by a volumetric grid system $R_{h}$, with spacings $h_{x}, h_{y}, h_{z}$ defined by:

$$
\begin{equation*}
R_{h}=\left\{\left(i h_{x}, j h_{y}, k h_{z}\right): 0 \leqslant i, j, k \leqslant M\right\} \tag{2.1.7}
\end{equation*}
$$

Assuming that ( $\mathrm{i}, \mathrm{j}, \mathrm{k}$ ) denotes the grid point ( $\mathrm{ih} \mathrm{X}_{\mathrm{x}}, \mathrm{jh} \mathrm{h}_{\mathrm{y}}, \mathrm{kh} \mathrm{z}_{\mathrm{z}}$ ) and $U_{i, j, k}$ denotes $U\left(i h_{x}, j h_{y}, k h_{z}\right)$, a discrete approximation for $x-$ derivative of the form:

$$
\begin{align*}
\frac{\partial}{\partial x}\left(C_{1}(x, y, z) \frac{\partial U}{\partial x}\right) & \simeq C_{1}\left(x+\frac{h}{2}, y, z\right)\left[\frac{u\left(x+h_{x}, y, z\right)-u(x, y, z)}{h_{x}^{2}}\right] \\
& -C_{1}\left(x-\frac{h_{x}}{2}, y, z\right)\left[\frac{u(x, y, z)-u(x-h x, y, z)}{h_{x}^{2}}\right] \tag{2.1.8}
\end{align*}
$$

with an analogous expression for $\frac{\partial}{\partial y}\left(C_{2}(x, y, z) \frac{\partial U}{\partial y}\right)$ and $\frac{\partial}{\partial z}\left(C_{3}(x, y, z) \frac{\partial U}{\partial z}\right)$ are used to derive a linear finite difference equation at point $\left(i h_{x}, j h_{y}, k h_{z}\right)$.

When the seven-point, three-dimensional molecule (Figure 2.2)
is used


FIGURE 2.2
the following finite difference equation is obtained

$$
\begin{array}{r}
D_{i, j, k} u_{i, j, k}^{-I} i, j, k u_{i, j, k-1}-L_{1, j, k} u_{i, j, k+1}-T_{i, j, k} u_{i, j+1, k}-B_{i, j, k} u_{i, j-1, k} \\
-R_{i, j, k} u_{i+1, j, k}-L_{i, j, k} u_{i-1, j, k}=s_{i, j, k} \tag{2.1.9}
\end{array}
$$

with $D_{i, j, k}>0, I_{i, j, k}, \square_{i, j, k}, T_{i, j, k}, B_{i, j, k}, R_{i, j, k}, L_{i, j, k}>0$
(representing mnemonic abbreviations for point Inplane, , utplane, Top, Bottom, Right and Left of the point ( $i h_{x}, j h_{y}, k h_{z}$ )) and

$$
\begin{equation*}
D_{i, j, k} \geqslant I_{i, j, k}+\square_{i, j, k}+T_{i, j, k}+B_{i, j, k}+R_{i, j, k}+L_{i, j, k} \tag{2.1.10}
\end{equation*}
$$

for all $\mathrm{i}, \mathrm{j}, \mathrm{k}$, with equality at interior point if $\mathrm{P}=0$.
Under the same considerations as in the 2D-case, the obtained set of symmetric difference equations may be expressed as the system (2.1.3), the coefficient matrix A being a sparse, real, seven-diagonal, square matrix with the properties (2.1.4).

Henceforth, throughout the thesis, the solution of the system $A \underline{u}=\underline{s}$ will be restricted to matricus $A$ with the properties (2.1.4b)-(2.1.4e) unless otherwise specified.

### 2.2 THE MODEL PROBLEMS

Let us consider the following 2D-model problem:

We seek to determine a continuous, twice differentiable function $u(x, y)$ in the region,

$$
R \equiv(0,1) \times(0,1) \text { with boundary } C,
$$

satisfying Laplace's equation

$$
\begin{equation*}
\frac{\partial^{2} U}{\partial x^{2}}+\frac{\partial^{2} u}{\partial y^{2}}=0, \quad(x, y) \varepsilon R \tag{2.2.1}
\end{equation*}
$$

subject to the Dirichlet boundary conditions

$$
\begin{equation*}
U(x, y)=0, \quad(x, y) \varepsilon C \tag{2.2.2}
\end{equation*}
$$

The region $R$ under consideration is covered by a rectilinear net with mesh spacing $h$ in the $X, Y$ directions and mesh points ( $x_{i}, y_{j}$ ), where $x_{i}=i h, i=0,1, \ldots, M ; y_{j}=j h, j=0,1, \ldots M$.

This problem is a special case $\left(C_{1}(x, y) \equiv C_{2}(x, y) \equiv 1\right.$, $P(x, y) \equiv Q(x, y) \equiv 0)$ of the general 2D-problem given in section 2.1 . Substituting the finite difference approximations for the derivatives in (2.2.1) the following five-point formula is obtained

$$
\begin{equation*}
4 u_{i, j} u_{i+1, j^{-u}}^{i-1, j^{-u_{i}}}{ }_{i+1}^{-u_{i, j-1}}=0 \tag{2.2.3}
\end{equation*}
$$

If we order the $(M-1)^{2}$ internal mesh points column-wise (Figure 2.3)


FIGURE 2.3
the coefficient matrix $A$ of the obtained system (2.1.3) is a real, square, quindiagonal, sparse matrix of order $m^{2}=(M-1)^{2}$ and of the general form,

where $I$, is the unit matrix of order $m$ and $A_{i}$, $i \varepsilon[1, m]$ matrices of order m, given by,

$$
A_{i}=\left[\begin{array}{cccccc}
4 & -1 & & & &  \tag{2.2.4b}\\
-1 & 4 & -1 & & 0 \\
& -1 & 4 & -1 & \\
& \ddots & \ddots & \\
& & \ddots & \ddots & \ddots & \\
& 0 & & \ddots & \ddots & -1 \\
& & & \ddots & \ddots & 4
\end{array}\right] \quad, i \varepsilon[1, \mathrm{~m}]
$$

The solution vector $\underline{u}$ and the right hand side vector $\underline{s}$ of (2.1.3) are $\left(m^{2} \times 1\right)$ column vectors defined as before in section 2.1 .

3D-mode1 Problem
We consider now the Laplace's equation in three space dimensions:

$$
\begin{equation*}
\frac{\partial^{2} U}{\partial x^{2}}+\frac{\partial^{2} U}{\partial y^{2}}+\frac{\partial^{2} U}{\partial z^{2}}=0, \quad(x, y, z) \varepsilon R \tag{2.2.5}
\end{equation*}
$$

subject to the Dirichlet boundary conditions

$$
\begin{equation*}
U(x, y, z)=0, \quad(x, y, z) \in C \tag{2.2.5a}
\end{equation*}
$$

where $R \equiv(0,1) \times(0,1) \times(0,1)$ and $C$ its boundary.
The region under consideration $R$ (unit cube) is covered by an equally spaced three-dimensional grid, defined by,

$$
\begin{equation*}
R_{h}=\{(i h, j h, k h): 0 \leqslant i, j, k \leqslant M\} \tag{2.2.5b}
\end{equation*}
$$

where M. $\mathrm{h}=1$ and h the grid spacing.
Discrete approximations to the derivatives in (2.2.5), in an analogous way to (2.1.8), leads to the following seven-point finite difference equation

$$
\begin{equation*}
6 u_{i, j, k}-u_{i+1, j, k}-u_{i-1, j, k}-u_{i, j+1, k}-u_{i, j-1, k}^{-u_{i, j, k+1}}-u_{i, j, k-1}=0 \tag{2.2.6}
\end{equation*}
$$

Ordering the $(M-1)^{3}$ internal points of $R_{h}$ with increasing values of j , then i , then k (Figure 2.4).


FIGURE 2.4
the coefficicnt matrix $A$ of the obtained system (2.1.3) is a real, square, seven-diagonal, sparse matrix of order $m^{3}=(M-1)^{3}$ and of the general form:

$$
A=\left[\begin{array}{llllll}
A_{1} & -I & & & & \\
-I & A_{2} & -I & & &  \tag{2.2.7a}\\
& -I & \ddots & \ddots & & \\
& & \ddots & \ddots & \ddots & \\
& & \ddots & \ddots & \ddots & -I \\
& & & \ddots & \ddots & \\
& & & & -I & A_{m}
\end{array}\right]
$$

where $I$ is the unit matrix of order $m^{2}$ and $A_{i}$, $i \varepsilon[1, m]$ are matrices of order $\mathrm{m}^{2}$ given by:

$$
A_{i}=\left[\begin{array}{llllll}
B_{j+1} & -J & & & &  \tag{2.2.7b}\\
-J & B_{j+2} & -J & & 0 \\
& -J & \ddots & \ddots & & \\
& & \ddots & \ddots & \ddots & \\
& 0 & \ddots & \ddots & \ddots & -J \\
& & \ddots & \ddots & \ddots_{0} \\
& & & \ddots & B_{j+m}
\end{array}\right], j=(i-1) m, i \in[1, m]
$$

where $J$ is now the unit matrix of order $m$ and $B_{i}, i \varepsilon[1, m]$ are matrices of order m given by

$$
B_{i}=\left[\begin{array}{cccccc}
6 & -1 & & & &  \tag{2.2.7c}\\
-1 & 6 & -1 & & 0 & \\
& -1 & \ddots & \ddots & & \\
& \ddots & \ddots & \ddots & \\
& 0 & \ddots & \ddots & -1 \\
& & & \ddots & \ddots & 6
\end{array}\right], i \in[1, m]
$$

The solution vector $\underline{u}$ and the right hand side vector $\underline{s}$ of (2.1.3), in this case, are ( $\mathrm{m}^{3} \times 1$ ) column vectors.

### 2.3 SOME POINT ITERATIVE METHODS

For the solution of the linear system (2.1.3), where the coefficient matrix $A$ is a large, sparse matrix with no more than five or seven non-zero elements in any row, e.g. formulae (2.2.3) or (2.2.6) respectively, iterative methods of successive approximations
are usually preferable to direct methods such as Gaussian elimination and Cramer's rule.

The iterative process, starts with an arbitrary initial approximation (can be chosen arbitrarily close to the exact solution of the difference equations) to the solution and then it is successively modified according to some predetermined rule.

Consider the system (2.1.3) and let $\underline{u}^{(n)}$ be a sequence, such that,

$$
\underline{u}^{(\mathrm{n})} \rightarrow A^{-1} \underline{s} \quad \text { as } n \rightarrow+\infty .
$$

Generally an iterative scheme is said to be of degree $v$ if $\underline{u}^{(n)}$ is a function of $A, \underline{s}^{,} \underline{u}^{(n-1)}, \underline{u}^{(n-2)}, \ldots, \underline{u}^{(n-\nu)}$.

Since computer storage requirements must be minimised the value of $v$ must be kept reasonably small. Consequently, for an iteration of first degree ( $v=1$ ), we obtain

$$
\begin{equation*}
\underline{u}^{(\mathrm{n})}=\mathrm{f}_{\mathrm{n}}\left(\mathrm{~A}, \underline{\mathrm{~s}}^{\left(\underline{u}^{(\mathrm{n}-1)}\right)} .\right. \tag{2.3.1}
\end{equation*}
$$

The iterative process (2.3.1) is said to be
(i) 'linear' if $f_{n}$ is a linear function of $\underline{u}^{(n-1)}$ and
(ii) 'stationary' if $f_{n}$ is independent of $n$.

A general iterative scheme of the form

$$
\left.\underline{u}^{(\mathrm{n}+1)}=\theta^{(\mathrm{n})}+\mathrm{f}_{\mathrm{n}} \underline{u}^{(\mathrm{n})}\right)
$$

is said to be 'non-stationary' if either $\theta$ or $f_{n}$ (or both) is (are) dependent on n .

Let $A=E-F$, so the system (2.1.3) can be written as

$$
\begin{equation*}
E \underline{u}=F \underline{u}+\underline{s} . \tag{2.3.2}
\end{equation*}
$$

Assuming that $\underline{u}^{(n)}$ denotes the $n^{\text {th }}$ approximation to the solution $\underline{u}$, then the following iterative scheme can be obtained

$$
\begin{equation*}
\underline{E u}^{(\mathrm{n}+1)}=\underline{\mathrm{Fu}}^{(\mathrm{n})}+\underline{\mathrm{s}}^{-} \tag{2.3.3}
\end{equation*}
$$

where the initial approximation (starting vector) $\underline{u}^{(0)}$ is an
arbitrary column vector.
The matrix $E$ (assuming that $\operatorname{det}(E) \neq 0$ and $E^{-1}$ exists) is chosen so that $\underline{u}^{(n+1)}$ can be easily obtained from (2.3.3), provided that $\underline{u}^{(n)}$ is known, and is the structure of $E$, which determines two important classes of iterative methods, specifically:
(i) if $E$ is a diagonal matrix, then the methods of 'Simultaneous displacements' [e.g. Jacobi, Richardson, etc.] are developed.
(ii) if $E$ is a lower triangular matrix the methods of 'Successive displacements' [e.g. Gauss-Seidel, Successive Over-relaxation (S.O.R.), etc.] arise.

Note that although in the 'Successive displacement' method the ordering of the mesh points is such that the latest estimate $u_{i}^{(n+1)}$ of the components of $\underline{u}$ is used in order to determine a new estimate of a component of $\underline{u}^{(n+1)}$, in the 'Simultaneous displacement' method the order in which the components $u_{i}^{(n+1)}$ are obtained is of no consequence.

Since in this thesis we are solely interested in 'Simultaneous displacement' methods, in the following we define the basic methods of this important class of Iterative Procedures.

Class of 'Simultaneous Displacement' Methods
Let us consider without loss of generality that the coefficient matrix $A$ of the system (2.1.3) can be written as

$$
\begin{equation*}
A=D-L-U \tag{2.3.4}
\end{equation*}
$$

where $D$ is a positive diagonal matrix whose elements are the diagonal elements of $A$ and $L, U$ are respectively lower and upper triangular matrices with zero diagonal elements.

Since $D$ is positive diagonal matrix and so $D^{-1}$ exists, let

$$
\begin{equation*}
B=D^{-1}(L+U) \quad \text { and } \quad \underline{c}=D^{-1} \underline{s} \tag{2.3.5}
\end{equation*}
$$

Combining (2.1.3), (2.3.4) and (2.3.5) we obtain

$$
\begin{equation*}
\underline{\mathrm{u}}=\mathrm{B} \cdot \underline{\mathrm{u}}+\underline{\mathrm{c}} . \tag{2.3.6}
\end{equation*}
$$

Then, the iterative scheme defined by

$$
\begin{equation*}
\underline{u}^{(n+1)}=B \cdot \underline{u}^{(n)}+\underline{c}, n \geqslant 0, \tag{2.3.7}
\end{equation*}
$$

is known as 'Jacobi' iteration and can be equivalently written as

$$
\begin{equation*}
\underline{u}^{(n+1)}=\underline{u}^{(n)}+\left({\left.\underline{c}-D^{-1} A \underline{u}^{(n)}\right) .}^{(n)}\right. \tag{2.3.8}
\end{equation*}
$$

The method of 'Simultaneous displacement' is defined by

$$
\begin{equation*}
\underline{u}^{(n+1)}=\underline{u}^{(n)}+a\left(\underline{c}^{\left(n-D^{-1} A\right.} \cdot \underline{u}^{(n)}\right) \tag{2.3.9}
\end{equation*}
$$

where a is an acceleration parameter (positive constant) 'chosen to speed up convergence.

Assuming, without loss of generality, that $D$ may be chosen as the identity matrix $I$, the Jacobi (2.3.7) and Simultaneous

Displacement (2.3.9) iterations can be re-written respectively as,

$$
\begin{align*}
& \underline{u}^{(n+1)}=(L+U) \underline{u}^{(n)}+\underline{s}^{(n)}  \tag{2.3.10}\\
& \underline{u}^{(n+1)}=\underline{u}^{(n)}+a \underline{s}^{\left(\underline{A} \underline{u}^{(n)}\right)} \tag{2.3.11}
\end{align*}
$$

and
or equivalently
and

$$
\begin{align*}
& \underline{u}^{(n+1)}=\underline{u}^{(n)}+\underline{r}^{(n)},  \tag{2.3.12}\\
& \underline{u}^{(n+1)}=\underline{u}^{(n)}+a \cdot \underline{r}^{(n)}=(I-a A) \underline{u}^{(n)}+a \cdot \underline{s} \tag{2.3.13}
\end{align*}
$$

where $\underline{r}$ is the residual vector defined by

$$
\begin{equation*}
\underline{r}^{(n)}=\underline{s}^{\left(n-A \underline{u}^{(n)}\right.} . \tag{2.3.14}
\end{equation*}
$$

Convergence of Simultaneous Displacement Methods
The iterative methods described previously, can be written in the general form,

$$
\begin{equation*}
\underline{\mathrm{u}}^{(\mathrm{n}+1)}=\tilde{\mathrm{M}} \underline{\mathrm{u}}^{(\mathrm{n})}+\underline{\mathrm{d}}, \mathrm{n} \geqslant 0 \tag{2.3.15}
\end{equation*}
$$

where $\tilde{M}$ is known as the 'iteration matrix' of the considered method and $d$ is a column vector of constants.

We consider now the system (2.1.3) and the iterative
scheme (2.3.15).
Definition 2.3.1
The 'consistency condition' for an iterative process is said to be satisfied if the solution $\underline{u}$ of system (2.1.3) is substituted for $\underline{u}^{(n)}$ in (2.3.15) then $\underline{u}^{(n+1)}$ is also the solution $\underline{u}$ of the system,i.e. the iterative procedure makes no further modification of successive iterates once the solution has been obtained.

Definition 2.3.2
An iterative method is said to be 'convergent', if for all initial vectors $\underline{u}^{(0)}$, each component of the successive iterates $\underline{u}^{(n)}$ tends to the corresponding component of the solution $\underline{u}$ of $\mathrm{A} \underline{u}=\underline{s}$, for any given $\underline{s}$.

## Theorem 2.3.1

An iterative method which can be expressed in the form of equation (2.3.15) converges if and only if (iff) $\rho(\tilde{M})<1$. Proof

$$
\text { Let } \underline{e}^{(n)}=\underline{u}^{(n)}-\underline{u} \text { be the error vector after } n \text {-iterative steps. }
$$ We assume that the iterative method is consistent, i.e.

$$
\begin{equation*}
\underline{u}=\tilde{M} \underline{u}+\underline{d} \tag{2.3.16}
\end{equation*}
$$

then from (2.3.15), (2.3.16) we obtain

$$
\begin{equation*}
\underline{e}^{(n+1)}=\tilde{M} \underline{e}^{(n)} \tag{2.3.17}
\end{equation*}
$$

and furthermore,

$$
\begin{equation*}
\underline{e}^{(n)}=\tilde{M}^{n} \underline{e}^{(0)}, \tag{2.3.18}
\end{equation*}
$$

where $\underline{e}^{(0)}$ is the error vector associated with the initial vector $\underline{u}^{(0)}$. Assuming that $\underline{e}^{(0)}$ is bounded, i.e. $\left|e_{i}^{(0)}\right|<\Sigma$, $i \varepsilon[1, N]$ for some constant $\Sigma$, then from (2.3.18) it follows that $\underline{e}^{(n)} \rightarrow \underline{0}$, as $\mathrm{n} \rightarrow+\infty$ if and only if $\tilde{\mathrm{M}}^{\mathrm{n}} \rightarrow \square$ (where $\square$ denotes the null matrix) as $n \rightarrow+\infty$.

By a theorem in [57] (p.82) (stating: if A is an arbitrary $n \times n$ matrix the $A$ is convergent iff $\rho(A)<1)$ this will be true if and only if $\rho(\tilde{M})<1$.

Note that if the coefficient matrix $A$ has the properties (2.1.4) it can be shown [63] (pp.13-14) that the Jacobi method converges.

## Rate of Convergence of Simultaneous Displacement Methods

The effectiveness of an iterative method is generally evaluated both from the computational work required per iteration and from the number of iterations required for convergence.

We usually say for practical purposes that an iterative method has converged when

$$
\begin{equation*}
\left\|\mid \underline{e}^{(n)}\right\| \leqslant \varepsilon\left\|\underline{e}^{(0)}\right\| \tag{2.3.19}
\end{equation*}
$$

where $e^{e}$ is the error vector, $\varepsilon$ is a predetermined positive factor and $\left\|\|\right.$ denotes the $L_{2}$ or spectral norm.

From (2.3.18) we obtain

$$
\begin{equation*}
\left\|\underline{\mathrm{e}}^{(\mathrm{n})}\right\|=\left\|\tilde{\mathrm{M}}^{\mathrm{n}} \underline{\mathrm{e}}^{(0)}\right\| \leqslant\left\|\tilde{\mathrm{M}}^{\mathrm{n}}\right\| \cdot\left\|\underline{\mathrm{e}}^{(0)}\right\| \tag{2.3.20}
\end{equation*}
$$

(Note that in the case of a non-stationary iteration the relationships (2.3.18), (2.3.20) become

$$
\begin{equation*}
\left\|\underline{e}^{(n)}\right\|=\tilde{M}_{n} \tilde{M}_{n-1} \cdots \tilde{M}_{1} \tilde{M}_{0} \cdot \underline{e}^{(0)} \tag{2.3.20a}
\end{equation*}
$$

and

$$
\begin{equation*}
\left\|e^{(n)}\right\|=\left\|\prod_{i=0}^{n} \tilde{M}_{i} \underline{e}^{(0)}| | \leqslant\right\| \prod_{i=0}^{n} \tilde{M}_{i}\|\cdot\| \underline{e}^{(0)} \| \tag{2.3.20b}
\end{equation*}
$$

respectively).
It is known that $\left\|\tilde{M}^{n}\right\| \rightarrow 0$ as $n \rightarrow+\infty$ iff $\rho(\tilde{M})<1$, so (2.3.19) can be satisfied choosing $n$ large enough such that

$$
\begin{equation*}
\left|\left|\tilde{M}^{n}\right|\right| \leqslant \varepsilon . \tag{2.3.21}
\end{equation*}
$$

Then, for all $n$ sufficiently large that $\left|\left|\tilde{\mathrm{M}}^{\mathrm{n}}\right|\right|<1,(2.3 .21)$ is equivalent to

$$
\begin{equation*}
\mathrm{n} \geqslant-\ln \varepsilon /\left(-\frac{1}{\mathrm{n}} \ln | | \tilde{\mathrm{M}}^{\mathrm{n}}| |\right) \tag{2.3.22}
\end{equation*}
$$

Consider the iterative method (2.3.15). Then the 'average rate of convergence' is defined by

$$
\begin{equation*}
\mathrm{R}_{\mathrm{n}}(\tilde{\mathrm{M}})=-\frac{1}{\mathrm{n}} \ell n| | \tilde{\mathrm{M}}^{\mathrm{n}}| | \tag{2.3.23}
\end{equation*}
$$

Definition 2.3.4
The 'asymptotic rate of convergence' is defined by

$$
\begin{equation*}
R(\tilde{M})=\lim _{n \rightarrow+\infty} R_{n}(\tilde{M})=-\ln [\rho(\tilde{M})] \tag{2,3.24}
\end{equation*}
$$

Henceforth, we shall refer to the $R(\tilde{M})$ simple as the 'rate of convergence'.

It can be shown $[61](p .87)$, that $\rho(\tilde{M})=\lim _{n \rightarrow+\infty}\left(| | \tilde{M}^{n}| |\right)^{1 / n}$, so equality (2.3.24) is a direct result of (2.3.23).

Note that from Definitions 2.3.2-2.3.4 we obtain that
(i) the number of iterations needed to reduce an initial error by a predetermined factor is approximately inversely proportional to the 'average rate of convergence'.
(ii) the rapidity of convergence depends on the rapidity with which the norm $\left|\left|\tilde{M}^{n}\right|\right|$ tends to zero, as $n \rightarrow+\infty$.
(iii) the smaller the value of $\rho(\tilde{M})$, the greater the rate of convergence $R(\tilde{M})$. Consequently, the iterative processes achieve their fastest rate of convergence when $\rho(\tilde{M})$ is reduced to a minimum.

Consider now the simultaneous iteration defin:d by (2.3.15).
It can be shown that the error vector satisfies:

$$
\begin{equation*}
\underline{e}^{(n+1)}=(I-a A) e^{(n)} \tag{2.3.25}
\end{equation*}
$$

or

$$
\begin{equation*}
\underline{e}^{(n+1)}=(I-a A)^{(n+1)} \underline{e}^{(0)} \tag{2,3.26}
\end{equation*}
$$

Note that the error operator is constant throughout the iteration (stationary, linear iterative procedure).

Let us assume that the matrix ( $\mathrm{I}-\mathrm{aA}$ ) has N -linearly independent eigenvectors $\mathrm{v}_{\mathrm{i}}$, associated with N -distinct eigenvalues $\lambda_{i}$ and let $\mu_{i}, i \varepsilon[1, N]$ be the eigenvalues of $A$. Given that an arbitrary vector $\underline{e}^{(0)}$ can be expressed as

$$
\begin{equation*}
\underline{e}^{(0)}=\sum_{i=1}^{N} c_{i} \underline{v}_{i}, \tag{2.3.27}
\end{equation*}
$$

where $c_{i}$, $i \varepsilon[1, N]$ constants, we have that

$$
\begin{equation*}
\underline{e}^{(n+1)}=\sum_{i=1}^{N} c_{i} \lambda_{i}^{(n+1)} \underline{v}_{i}=\lambda_{N}^{(n+1)}\left\{_{\left\{_{1}\right.}\left(\frac{\lambda_{1}}{\lambda_{N}}\right) \underline{v}_{1}+c_{2}\left(\frac{\lambda_{2}}{\lambda_{N}}\right)^{(n+1)} \underline{v}_{2}+\ldots\right\} . \tag{2.3.28}
\end{equation*}
$$

Hence, the iterative scheme converges if $\rho(\mathrm{I}-\mathrm{aA})<1$ i.e., the modulus of the largest eigenvalue of ( $\mathrm{I}-\mathrm{aA}$ ) must be less than unity.

Let $m, M$ be the extreme eigenvalues of $A$ e.g.,

$$
\begin{equation*}
0 \leqslant m \leqslant \mu_{i} \leqslant M<+\infty \quad, i \varepsilon[1, N] . \tag{2.3.29}
\end{equation*}
$$

Since

$$
\lambda_{i}=1-a \mu_{i}, \quad i \varepsilon[1, N]
$$

the necessary and sufficient condition for convergence gives

$$
\begin{equation*}
\left|1-\mathrm{a}_{\mathrm{i}}\right|<1, \quad \mathrm{i} \varepsilon[1, \mathrm{~N}], \tag{2.3.30}
\end{equation*}
$$

hence,

$$
\begin{equation*}
0<a<\frac{2}{M} . \tag{2.3.31}
\end{equation*}
$$

The choice of the parameter $a$ in such a way that $\rho(I-a A)$ is minimised, leads to the fastest rate of convergence. The optimal a is obtained for the smallest of $\max \{|1-\mathrm{am}|,|1-\mathrm{am}|\}$ and this value of a is given by:

$$
|1-\mathrm{am}|=-|1-\mathrm{am}|
$$

or equivalently

$$
\begin{equation*}
a=\frac{2}{m+M} \tag{2.3.32}
\end{equation*}
$$

With this choice of $a$, for all i, we obtain

$$
\begin{equation*}
\left|1-a \mu_{i}\right| \leqslant \frac{M-m}{M+m}<1 . \tag{2.3.33}
\end{equation*}
$$

From (2.3.24) the (asymptotic) rate of convergence $R_{s}$ is given by

$$
\begin{equation*}
\mathrm{R}_{\mathrm{S}}=-\ell n(\rho(\mathrm{I}-\mathrm{aA})) \tag{2.3.34}
\end{equation*}
$$

so that from (2.3.33) we have

$$
\begin{equation*}
R_{s} \geqslant \frac{2}{\frac{M}{m}} \quad, \text { for } \frac{M}{m} \gg 1 \tag{2.3.35}
\end{equation*}
$$

where the ratio $\frac{M}{m}$ is known as the $P$-condition number of $A$.
Note that the rate of convergence for the Simultaneous displacement method is independent of the ordering of the points.

Consider now the iterative scheme (2.3.15).

If a sequence of different factors $a_{n}$ (instead of the constant factors a) for each iteration is multiplied by each component of the residual vector $\underline{r}^{(n)}$ and then added to each component of the $\underline{u}^{(n)}$ to give the next value of $\underline{u}^{(n+1)}$, the following non-stationary iterative scheme, known as 'Richardson's Method' [51], is obtained:

$$
\begin{equation*}
\underline{u}^{(n+1)}=\underline{u}^{(n)}+a_{n} \underline{r}^{(n)}=\left(I-a_{n} A\right) \underline{u}^{(n)}+a_{n} \underline{s}^{(n)} \tag{2,3.36}
\end{equation*}
$$

The constants $a_{n}$ are either given in terms of the extreme eigenvalues of the matrix $A$ or calculated during the iteration from formulae involving $\underline{u}^{(n)}$. The error vector for the iterative scheme (2.3.36) is given by:

$$
\begin{equation*}
\underline{e}^{(n+1)}=\left(I-a_{n} A\right) e^{(n)} \tag{2.3.37}
\end{equation*}
$$

or

$$
\begin{equation*}
\underline{e}^{(n+1)}=\prod_{i=0}^{n}\left(I-a_{i} A\right) \underline{e}^{(0)} \tag{2.3.38}
\end{equation*}
$$

Note that the error operator changes for each iteration (Non-stationary iterative method).

Let

$$
\begin{equation*}
F_{n+1}(A)=\prod_{i=0}^{n}\left(I-a_{i} A\right) \tag{2.3.39}
\end{equation*}
$$

If $\mu_{i}$, $i \varepsilon[1, N]$ are the $N$-eigenvalues of the positive definite matrix $A$ and ${\underset{-i}{i}}$ are the corresponding eigenvectors, then the eigenvalues and eigenvectors of $\mathrm{F}_{\mathrm{n}+1}(\mathrm{~A})$ are $\mathrm{F}_{\mathrm{n}+1}\left(\mu_{\mathrm{i}}\right)$ and $\mathrm{v}_{\mathrm{i}}$ respectively. A combination of (2.3.38), (2.3.27) yields

$$
\underline{e}^{(n+1)}=F_{n+1}(A) \underline{e}^{(0)}=\sum_{i=1}^{N} F_{n+1}(A) c_{i} \underline{v}_{i}
$$

or

$$
\begin{equation*}
\underline{e}^{(n+1)}=\sum_{i=1}^{N} c_{i-i} v_{i} F_{n+1}\left(\mu_{i}\right) \tag{2.3.40}
\end{equation*}
$$

The minimization of the error vector is now equivalent to the following problem:

Find a polynomial $\mathrm{F}_{\mathrm{n}+1}(\mathrm{x})$ of degree $(\mathrm{n}+1)$ such that

$$
\begin{equation*}
\max _{x \in[\mathrm{~m}, \mathrm{M}]}\left|\mathrm{F}_{\mathrm{n}+1}(\mathrm{x})\right| \tag{2.3.41}
\end{equation*}
$$

is minimized, under the constraint

$$
\begin{equation*}
F_{n+1}(0)=1 . \tag{2.3.42}
\end{equation*}
$$

The polynomial satisfying (2.3.41), (2.3.42) is given in [41] to be:

$$
\begin{equation*}
F_{n+1}(x)=\frac{T_{n+1}\left(\frac{M+m-2 x}{M-m}\right)}{T_{n+1}\left(\frac{M+m}{M-m}\right)} \tag{2.3.43}
\end{equation*}
$$

where $T_{n}$ is the Chebychev polynomial of degree $n$ given by:

$$
\begin{equation*}
T_{n}(x)=\cos \left(n \cos ^{-1} x\right)=\frac{1}{2}\left\{\left[x+\sqrt{\left(x^{2}-1\right)}\right]^{n}+\left[x+\sqrt{x^{2}-1}\right]^{-n}\right\} \tag{2.3.43a}
\end{equation*}
$$

adjusted in the interval $[-1,1]$. The optimal choice of parameters $a_{i}$ is such that the zeros of $F_{n+1}(x)$ coincide with the set $\left\{a_{i}^{-1}\right\}$, provided that the extreme eigenva!ues $m, M$ and the value of the ratio $\frac{\left\|\underline{\underline{r}}^{(0)}\right\|}{\left\|\underline{\underline{r}}^{(n)}\right\|}$, at which the iteration is terminated, are given. From (2.3.39) the zeros of $\mathrm{F}_{\mathrm{n}+1}(\mathrm{x})$ are:

$$
\begin{equation*}
x_{i}=\left[a_{i}^{(n+1)}\right]^{-1}, \quad i \varepsilon[1, n+1], \tag{2.3.43b}
\end{equation*}
$$

while, from (2.3.43), the zeros of $T_{n+1}\left(\frac{M+m-2 x}{M-m}\right)$ and consequently of $\mathrm{F}_{\mathrm{n}+1}(\mathrm{x})$ are:

$$
\begin{equation*}
\frac{M+m-2 x_{i}}{M-m}=\cos \left\{\frac{(2 i-1) \pi}{2(n+1)}\right\}, \quad i \varepsilon[1, n+1] . \tag{2.3.43c}
\end{equation*}
$$

By equating the zeros of polynomials we obtain

$$
\begin{equation*}
a_{i}^{(n+1)}=\frac{2}{(M+m)-(M-m) \cos \left[\frac{(2 i-1) \pi}{2(n+1)}\right]}, \tag{2.3.44}
\end{equation*}
$$

and assuming that only $\nu$-iterations are to be performed then,

$$
\begin{equation*}
a_{i}^{(\nu)}=\frac{2}{(M+m)-(M-m) \cos \left[\frac{(2 i-1) \pi}{2 \nu}\right]}, i \varepsilon[1, \nu] . \tag{2.3.45}
\end{equation*}
$$

Since the maximum absolute value of the numerator of (2.3.43) is unity, then the maximum value of $\mathrm{F}_{\mathrm{n}+1}(\mathrm{x})$, as $\mathrm{n} \rightarrow+\infty$, is given by:

$$
\begin{equation*}
\max _{x \in[m, M]}\left|F_{n+1}(x)\right|=\left[T_{n+1}\left(\frac{M+m}{M-m}\right)\right]^{-1} \tag{2.3.46}
\end{equation*}
$$

Consider now the iterative scheme (2.3.36).

## Definition 2.3.5

The 'average convergence factor' for the first $v$-iterative steps is defined by the quantity $\left[\frac{\left\|\underline{e}^{(\nu)}\right\|}{\left\|\underline{e}^{(0)}\right\|}\right]^{1 / \nu}$

Definition 2.3.6
The 'average rate of convergence' for the first $v$-iterative steps is defined by the quantity $-\ln \left(\frac{\left\|\underline{e}^{(\nu)}\right\|}{\left\|\underline{e}^{(0)}\right\|}\right)^{1 / \nu}$.

It can be shown [27] that the average rate of convergence is bounded by:

$$
\begin{equation*}
-\frac{1}{v} \ln 2+\ln \left(z_{0}+\sqrt{z_{0}^{2}-1}\right) \tag{2.3.47}
\end{equation*}
$$

where $z_{0}=\frac{M+m}{M-m}>1$.

The rate of convergence, is the asymptotic value of (2.3.47) as $v \rightarrow+\infty$, and is given by $\quad \begin{aligned} \\ R_{R}=\ln \left(z_{0}+\sqrt{z_{0}^{2}-1}\right) .\end{aligned}$

Given that for most problems $z_{0} \simeq 1+2 \frac{m}{M}$, then we obtain

$$
\begin{equation*}
R_{R} \simeq \frac{2}{\sqrt{\frac{M}{m}}} \text { for } \quad \frac{M}{m} \gg 1 \tag{2,3.49}
\end{equation*}
$$

Assuming that the extreme eigenvalues $m, M$, the value of $v$ and the value of the ratio $\frac{\left\|\underline{r}^{(0)}\right\|}{\left\|\underline{r}^{(v)}\right\|}$ (at which the iteration is to be terminated) are given, the procedure is then simplified, to calculate the terms of the sequence $\left\{a_{i}\right\}$, $i \varepsilon[1, \nu]$, which is given by (2.3.44) and is the best possible choice if only $\nu$-iterations are to be performed.

Consider now the following linear stationary iteration of second degree, known as 'second order Richardson's iterative scheme'

$$
\begin{equation*}
\left.\underline{u}^{(n+1)}=\underline{u}^{(n)}+a\left(\underline{s}-\hat{u}^{(n)}\right)+\beta \underline{u}^{(n)} \underline{u}^{(n-1)}\right) \tag{2,3.50}
\end{equation*}
$$

where parameters $a, \beta$ remain constant throughout the iteration and are chosen to provide maximum convergence to the solution. The error vector for (2.3.50) satisfies

$$
\begin{equation*}
\underline{e}^{(n+1)}=[(1+\beta) I-a A] \underline{e}^{(n)}-\beta \underline{e}^{(n-1)} \tag{2.3.51}
\end{equation*}
$$

thus for each error mode we have

$$
\begin{equation*}
\underline{e}_{i}^{(n+1)}=\left[1+\beta-a \mu_{i}\right] e_{i}^{(n)}-\beta e_{i}^{(n-1)} . \tag{2.3.52}
\end{equation*}
$$

Let $\gamma_{i}$ be the eigenvalues of the matrix associated with the iterative process then

$$
\begin{equation*}
{\underset{-}{e}}_{(n+1)}^{(n)} \gamma_{i} e_{i}^{(n)}=\gamma_{i}^{2} e_{i}^{(n-1)} \tag{2.3.53}
\end{equation*}
$$

From (2.3.52), (2.3.53) we obtain
and combination of (2.3.53), (2.3.54) leads to

$$
\begin{equation*}
\gamma_{i}^{2}-\left(1+\beta-a \mu_{i}\right) \gamma_{i}+\beta=0 \tag{2.3.55}
\end{equation*}
$$

which yields

$$
\begin{equation*}
\gamma_{i}=\frac{\delta_{i} \pm \sqrt{\delta_{i}^{2}-4 \beta}}{2} \tag{2.3.56}
\end{equation*}
$$

where $\delta_{i}=1+\beta-a \mu_{i}$.
Choosing the parameters $a, \beta$ so that $\left(\delta_{i}^{2}-4 \beta\right)<0$ for all $\mu_{i}$ (which means that $\gamma_{i}$ will be complex) all $\left|\gamma_{i}\right|$ will be identical. Then, the choices

$$
\begin{align*}
& 1+\beta-a \cdot m=2 \sqrt{\beta}  \tag{2.3.57a}\\
& 1+\beta-a \cdot M=-2 \sqrt{\beta} \tag{2.3.57b}
\end{align*}
$$

lead to

$$
\begin{equation*}
a=\left[\frac{2}{\sqrt{m}+\sqrt{M}}\right]^{2} \quad, \quad B=\left[\frac{\sqrt{M}-\sqrt{m}}{\sqrt{M}+\sqrt{m}}\right]^{2} . \tag{2.3.58}
\end{equation*}
$$

The above choice of parameters, which makes the square root of (2.3.56) zero or negative, yields,

$$
\begin{equation*}
\left|\gamma_{i}\right|=\sqrt{\beta} . \tag{2.3.59}
\end{equation*}
$$

The eigenvalues $\gamma_{i}$ are complex but all have the same absolute value and thus all the error modes are decreased at the same rate. Note that $\left|\gamma_{\mathrm{i}}\right|$ is always smaller than the maximum of $\left|\lambda_{1}\right|,\left|\lambda_{N}\right|$ and consequently the procedure is always more rapidly convergent than the Simultaneous Iteration. In the case that the estimates of the extreme eigenvalues $m, M$ are not exact, then it is advisable to underestimate $m$ and overestimate $M$ assuring the $\gamma_{i}$ being complex and therefore all the modes decaying at the same rate. From (2.3.50) it can be seen that to commence the iterative process two iterates are required and although one initial iterate is available, only $\left|\gamma_{i}\right|$ is known and the second initial iterate is unobtainable. Thus the convergence rate indicated by (2.3.59) is asymptotically only approached as $\mathrm{i} \rightarrow+\infty$. Given that the spectral radius of the iteration is $\sqrt{\beta}$, then the rate of convergence is $\frac{2}{\sqrt{\frac{M}{m}}}$.

Note that for each iteration of the method at least two vectors need to be stored.

Another non-stationary second degree iteration, less sensitive to round-off errors, is the 'Chebychev second order method' defined by

$$
\begin{equation*}
\underline{u}^{(n+1)}=\underline{u}^{(n)}+a_{n}{\left.\left.\underline{s}-\underline{A} \underline{u}^{(n)}\right)+\beta_{n} \underline{u}^{(n)} \underline{u}^{(n-1)}\right), ~}_{\text {(n) }} \tag{2.3.60}
\end{equation*}
$$

where $a_{n}, \beta_{n}$ are varied with each iterative step.
If the parameters $a_{n}, \beta_{n}$ have to be computed during the iteration, the technique proves to be inefficient especially for large values of $N$. For pre-determined values of $a_{n}, \beta_{n}$ the computational work per iteration is not much more than that of the iteration for the simultaneous displacement method and the storage requirements are no greater. The values of parameters $a_{n}, \beta_{n}$ are given in [55] as follows:

$$
\begin{equation*}
a_{n}=\frac{4 T_{n}\left(\frac{M+m}{M-m}\right)}{(M-m) T_{n+1}\left(\frac{M+m}{M-m}\right)}, \quad \beta_{n}=\frac{T_{n-1}\left(\frac{M+m}{M-m}\right)}{T_{n+1}\left(\frac{M+m}{M-m}\right)} . \tag{2.3.61}
\end{equation*}
$$

The error vector is given by

$$
\begin{equation*}
\underline{e}^{(n+1)}=\frac{T_{n+1}\left(\frac{M+m-2 x}{M-m}\right)}{T_{n+1}\left(\frac{M+m}{M-m}\right)} e^{(0)} \tag{2.3.62}
\end{equation*}
$$

which leads to identical results to those obtained using Richardson's method for rate of convergence.

Since $a_{n}, \beta_{n}$ are less than unity round off-errors do not appear and the method is preferred to Richardson's method with the only inconvenient matter raised being the estimation of the extreme eigenvalues m,M.

Finally, for the 2D-model problem it can easily be shown
[27] (p.226) that the simultaneous displacement method of (2.3.13)
is equivalent to Jacobi method of (2.3.12).

The processes defined so far, belong to the class of point iterative methods, in which each component of $\underline{u}^{(n)}$ is expressed 'explicitly' i.e., can be expressed by itself using already computed approximate values of the other unknowns. Grouping the equations of the original system, according to a predetermined rule ${ }^{(\dagger)}$, subsets of the components can be solved at once, giving rise to the so-called 'Block iterative' methods.

With proper formulation, all the discussed point iterative methods give their counterpart block iterative methods. Since, in the following chapters, we are interested in Point terative methods, the discussion for the Block methods is terminated here.

### 2.4 THE CONJUGATE GRADIENT METHOD

Consider the system,

$$
\begin{equation*}
\text { A. } \underline{x}=\underline{s} \tag{2.4.1}
\end{equation*}
$$

assuming that the coefficient matrix $A$ is real, symmetric and positive definite. Let $\underline{x}_{i}$ be an arbitrary vector and $\underline{r}_{i}$ the corresponding residual, as defined in section 2.3.

The system (2.4.1) can be written as the gradient of a quadratic function

$$
\begin{equation*}
F[\underline{x}]=\frac{1}{2} \underline{x} \cdot A \underline{x}-\underline{s} \cdot \underline{x} . \tag{2.4.2}
\end{equation*}
$$

The solution of system (2.4.1) is equivalent to the minimization of function $F[\underline{x}]$, which defines an ellipsoid in the $N$-dimensional space of the elements of $\underline{x}$. For an arbitrary trial vector $\underline{x}_{i}$, the residual $\underline{r}_{i}$ is given by:

$$
\begin{equation*}
\underline{r}_{i}=\underline{s}-A \underline{x}_{i}=-\operatorname{Grad} \mathrm{F}\left[\underline{x}_{\mathrm{i}}\right] \tag{2.4.3}
\end{equation*}
$$

 can consist of all points on a row or colum of a rectangular mesh.
and it is normal to the surface of the ellipsoid defined by (2.4.2) (see Definition 1.6.2).

Consider the iterative scheme:

$$
\begin{equation*}
\underline{x}_{i+1}=\underline{x}_{i}+a_{i-i} \tag{2.4.4}
\end{equation*}
$$

## is an

where $a_{i}$ arbitrary constant dependent upon $i$ and $\underline{m}_{i}$ is an arbitrary direction.

The problem now is to choose an $a_{i}$ such that the quadratic function $F\left[\underline{x}_{i+1}\right]$ will be a minimum for a given direction $\underline{m}_{i}$. Consider now a plane defined by ${\underset{-}{i}}$ and $\underline{r}_{i}$. Then, the intersection of this $p l a n e$ and the surface $F[\underline{x}]=$ constant is represented by Figure 2.5 .

$F[x]=c_{1}$

FIGURE 2.5
Note that the tip of the vector $\underline{x}_{i}$ appears as a point in this plane. The minimum $F\left[\underline{x}_{i+1}\right]$ is tangent to the direction $\underline{m}_{-1}$ and so


From (2.4.3) we can easily obtain

$$
\begin{equation*}
\underline{r}_{i+1}-\underline{r}_{i}=\underline{s}_{-A x_{i+1}}-\left[\underline{s}-A \underline{x}_{i}\right]=A\left[\underline{x}_{i}-\underline{x}_{i+1}\right] \tag{2.4.5}
\end{equation*}
$$

and combining (2.4.4), (2.4.5) we have

$$
\begin{equation*}
\underline{r}_{i+1}=\underline{r}_{i}-a_{i} A_{i} . \tag{2.4.6}
\end{equation*}
$$

Then, using the orthogonality relation $\underline{r}_{i+1} \cdot \underline{m}_{i}=0$, equation (2.4.6) becomes:

$$
\begin{equation*}
\underline{r}_{i+1} \cdot \stackrel{m}{-1}^{i}=\underline{r}_{i} \cdot \underline{m}_{i}-a_{i}{ }_{-A m_{i}} \cdot \underline{m}_{i}=0, \tag{2.4.7}
\end{equation*}
$$

which leads to

$$
\begin{equation*}
a_{i}=\frac{\underline{r}_{i} \cdot \underline{m}_{i}}{\underline{m}_{i} \cdot \underline{m}_{-i}} \tag{2.4.8}
\end{equation*}
$$

The above choice of $a_{i}$ systematically reduces $F\left[\underline{x}_{i+1}\right]$ and the method converges for any given $\underline{m}_{i}$.

If the direction $\underline{I}_{i}$ is chosen as vector ${\underset{-}{\sigma}}$, directed from a point on the ellipse towards the center of the ellipses, then the 'conjugate directions' are defined by $\underline{\sigma}_{i}$ and a vector $\underline{t}_{i}$, tangent to the ellipse at $\underline{x}_{i}$. The vectors ${\underset{\sigma}{i}}$ and $\underline{t}_{i}$ are orthogonal with respect to the matrix $A$, that is

$$
\begin{equation*}
\underline{-}_{i} \cdot \mathrm{~A} \underline{\mathrm{t}}_{\mathrm{i}}=0 . \tag{2.4.9}
\end{equation*}
$$

The minimum $F\left[\underline{x}_{i}\right]$ occurs at the centre of a manifold of ellipses in the plane passing through ${\underset{r}{i}}$ and $\underline{t}_{i}$ (Figure 2.6).


FIGURE 2.6
for an $a_{i}$ given from (2.4.8) by

$$
\begin{equation*}
a_{i}=\frac{\underline{r}_{i} \cdot \underline{-}_{i}}{\underline{\sigma}_{i} \cdot A \underline{\sigma}_{i}} \tag{2.4.10}
\end{equation*}
$$

The iterative scheme (2.4.4) and the residual (2.4.6) become respectively:

$$
\begin{equation*}
\underline{x}_{i+1}=\underline{x}_{i}+a_{i} \underline{\sigma}_{i} \tag{2.4.11}
\end{equation*}
$$

and

$$
\begin{equation*}
\underline{r}_{i+1}={\underset{-}{r}}_{i}-a_{i} \underline{-}_{i} \tag{2.4.12}
\end{equation*}
$$

The residual $\underline{r}_{i+1}$ is normal to $F\left[\underline{x}_{i+1}\right]=c$ at the point $\underline{x}_{i+1}$, the centre of the ellipses in the plane at which the function $F[\underline{x}]$ is minimal and is also orthogonal to the plane, which is tangent to the surface at point $\underline{x}_{i+1}$, and to the vectors ${\underset{-}{\sigma}}, \underline{r}_{i}, \underline{t}_{i}$, which lie in this plane.

Choosing

$$
\begin{equation*}
\underline{\sigma}_{i}=\underline{r}_{i}+\beta_{i} \underline{t}_{i}, \tag{2.4.13}
\end{equation*}
$$

from (2.4.9) we obtain

$$
\begin{equation*}
\underline{\sigma}_{i} \cdot A \underline{t}_{i}=\underline{r}_{i} \cdot A \underline{t}_{i}+\beta \cdot \underline{t}_{i} \cdot A \underline{t}_{i}=0 \tag{2.4.14}
\end{equation*}
$$

or

$$
\begin{equation*}
\beta_{i}=-\frac{r_{i} \cdot A t_{i}}{\underline{t}_{i} \cdot A t_{i}} \tag{2.4.15}
\end{equation*}
$$

Considering that the tangent vector $t_{i+1}$ lies in the plane of $\underline{r}_{i}, \underline{t}_{i}$ and with the choice

$$
\begin{equation*}
\underline{t}_{i+1}={\underset{-}{-}}_{i} \tag{2.4.16}
\end{equation*}
$$

from (2.4.13), (2.4.15) we have respectively,

$$
\begin{equation*}
\underline{\sigma}_{\mathrm{i}}=\underline{\mathrm{r}}_{\mathrm{i}}+\beta_{\mathrm{i}} \underline{\sigma}_{\mathrm{i}-1} \tag{2.4.17}
\end{equation*}
$$

and

$$
\begin{equation*}
\beta_{i}=\frac{\underline{r}_{i} \cdot{ }_{-}^{A} \underline{\sigma}_{i-1}}{\underline{\sigma}_{i-1} \cdot{ }_{-1-1}} . \tag{2.4.18}
\end{equation*}
$$

The equations (2.4.10)-(2.4.12) and (2.4.17), (2.4.18) imply the following orthogonality relations:

$$
\begin{equation*}
\underline{\mathrm{r}}_{\mathrm{i}+1} \cdot \underline{\mathrm{r}}_{\mathrm{i}}=0 ; \quad \underline{\mathrm{r}}_{\mathrm{i}+1} \cdot \underline{-}_{\mathrm{i}}=0 \tag{2.4.19}
\end{equation*}
$$

and

$$
\begin{equation*}
\underline{r}_{i+1} \cdot \stackrel{\sigma}{i-1}=0 ; \quad \underline{-}_{i} \cdot{\underset{-}{-}}_{i-1}=0 . \tag{2.4.19a}
\end{equation*}
$$

Moreover, it can be shown [39] that all ${\underset{-}{i}}$ are mutually orthogonal and all ${\underset{-}{i}}$ are mutually conjugate, i.e.,

$$
\begin{equation*}
\underline{r}_{i} \cdot \underline{r}_{j}=0 \quad ; \quad \underline{\sigma}_{i} \cdot \underline{A \sigma}_{j}=0, \quad \text { for } i \neq j . \tag{2.4.20}
\end{equation*}
$$

The iterative scheme of Conjugate Gradient method begins
 Since equations (2.4.19) hold for all $\underline{r}_{i}$, then ${\underset{-}{o}}_{0}=\underline{r}_{0}$ i.e., $\beta_{0}=0$. The method is exact in $N$-iterative steps with $\left|\underline{r}_{i+1}\right|<\left|\underline{r}_{\mathrm{i}}\right|$ at each step.

The procedure amounts to passing N -mutually orthogonal planes throigh the ellipsoid $F[x]$ and finding the center of the resultant manifold of ellipses in each plane.

From equations (2.4.10)-(2.4.12) and (2.4.17),(2.4.18), a second order iterative scheme can be derived as follows:

$$
\begin{align*}
& \underline{r}_{i+1}=\left(1+\frac{a_{i} \beta_{i}}{a_{i-1}}\right) \underline{r}_{i}-\left(\frac{a_{i} \beta_{i}}{a_{i-1}}\right) \underline{r}_{i-1}-a_{i} \underline{A r}_{i},  \tag{2.4.21}\\
& \underline{x}_{i+1}=\left(1+\frac{a_{i} \beta_{i}}{a_{i-1}}\right) \underline{x}_{i}-\left(\frac{a_{i} \beta_{i}}{a_{i-1}}\right) \underline{x}_{i-1}+a_{i}\left(\underline{A x}_{i}-\underline{s}\right) \tag{2.4.22}
\end{align*}
$$

and

$$
\begin{array}{r}
\underline{\sigma}_{i+1}=\left(1+\beta_{i+1}\right) \underline{\sigma}_{i}-\beta_{i} \stackrel{\sigma}{i-1}-a_{i} \underline{-}_{-i}  \tag{2.4.23}\\
\text { with } \beta_{0}=0 .
\end{array}
$$

Note that equation (2.4.22) is of the same form as the second order Richardson and Chebyshev procedures given by (2.3.50) and (2.3.60) respectively, except that here the coefficients are variables (instead of constants in Richardson case) chosen to minimize the quadratic function $F[\underline{x}]$ instead to generate the polynomial. (Chebychev case).

Although the Conjugate Gradient method theoretically gives an exact answer in $N$-iterative steps, the residuals are not truly
orthogonal in the actual practice because of the round-off errors. So, generally we have $\underline{r}_{-1-1} \neq 0$ but it might be expected to be very close to zero. If the obtained solution $x_{i+1}$ becomes too inaccurate because of round-off errors then the iteration either terminated if $\underline{r}_{i}$ is sufficiently small or could be restarted with $\underline{x}_{i+1}$ as initial guess.

Note that the inner products needed for the calculation of parameters $a_{i}, \beta_{i}$ are considerable time-consuming operations and the method depends upon orthogonality (the error reduction may become very slow if orthogonality breaks down [32]).

## SECTION B: THE PARABOLIC PROBLEM

### 2.5 A PARABOLIC 2D-EXAMPLE

We consider the quasi-linear (see [12]) diffusion equation of the following form:

$$
\begin{equation*}
\frac{\partial U}{\partial t}=\frac{\partial}{\partial x}\left(\Lambda \frac{\partial U}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Lambda \frac{\partial U}{\partial y}\right), \quad \Lambda=\Lambda(x, y) \tag{2.5.1}
\end{equation*}
$$

in the region $\bar{R}=R \times[0<t \leqslant T]$
where

$$
R=\{(x, y) ; \quad 0<x, y<1\}
$$

with initial condition

$$
\begin{equation*}
\mathrm{U}(\mathrm{x}, \mathrm{y}, 0)=\mathrm{f}(\mathrm{x}, \mathrm{y}), \quad(\mathrm{x}, \mathrm{y}) \varepsilon \mathrm{R} \tag{2.5.2}
\end{equation*}
$$

and subject to the boundary condition

$$
\begin{equation*}
U(x, y, t)=\gamma(x, y, t), \quad(x, y, t) \varepsilon C \times[0 \leqslant t \leqslant T] \tag{2.5.3}
\end{equation*}
$$

where $C$ is the boundary of $R$.
The region $R$ is covered by a rectilinear net with mesh spacings $h_{x}, h_{y}, \Delta t$ in the $\lambda, Y, t$ directions respectively and mesh points $\left(x_{i}, y_{j}, t_{k}\right)$ where:

$$
\begin{array}{ll}
x_{i}=i h_{x}, & i \varepsilon[0, p] \\
y_{j}=j h_{y}, & j=[0, m] \\
t_{k}=k \cdot \Delta t, & k=[0, T / \Delta t] . \tag{2.5.4}
\end{array}
$$

Let $u_{i, j}^{(k)}$ and $U_{i, j}^{(k)}$ denote the approximate (finite difference) and exact solution respectively of equation (2.5.1) at the point $\left(i h_{x}, j h_{y}, k \Delta t\right)$.

### 2.6 DERIVATION OF FINITE DIFFERENCE FORMULAE

Definition 2.6.1
Consider a function $y=f(x)$ and a constant interval size $\mathrm{h}=\mathrm{x}_{\mathrm{n}+1} \mathrm{x}_{\mathrm{n}}$. Let $\mathrm{y}_{\mathrm{n}}=\mathrm{f}\left(\mathrm{x}_{\mathrm{n}}\right)$, then if $\Delta, \nabla, \delta, \mu$ be the forward, backward, central and average difference operators respectively, we have:
and

$$
\begin{align*}
& \Delta y_{n}=y_{n+1}-y_{n}, \quad \nabla y_{n}=y_{n}-y_{n-1} \\
& \delta y_{n}=y_{n+\frac{1}{2}}-y_{n-\frac{1}{2}}, \quad \mu y_{n}=\frac{1}{2}\left[y_{n+\frac{1}{2}}+y_{n-\frac{1}{2}}\right] . \tag{2.6.0}
\end{align*}
$$

We assume that any given partial derivatives of $U$ are continuous and uniformly bounded for all $x, y, t \varepsilon \bar{R}$.

The exact difference replacement of equation (2.5.1) can be obtained from the following Taylor expansion

$$
\begin{gather*}
U(x, y, t+\Delta t)=\left(1+\Delta t \frac{\partial}{\partial t}+\frac{(\Delta t)^{2}}{2!} \frac{\partial^{2}}{\partial t^{2}}+\frac{(\Delta t)^{3}}{3!} \frac{\partial^{3}}{\partial t^{3}}+\cdots\right) U\left(x, y^{\prime}, t\right) \equiv \\
\equiv \exp \left(\Delta t \cdot \frac{\partial}{\partial t}\right) U(x, y, t) \tag{2.6.1}
\end{gather*}
$$

It is well known [43],[33] that

$$
\begin{align*}
& \frac{\partial}{\partial x}=\frac{2}{h_{x}} \sinh ^{-1}\left(\frac{\delta_{x}}{2}\right)=\frac{1}{h_{x}}\left(\delta_{x}-\frac{1^{2}}{2^{2} \cdot 3!} \delta_{x}^{3}+\frac{1^{2} \cdot 3^{2}}{2^{4} \cdot 5!} \delta_{x}^{5}-\ldots\right)  \tag{2.6.2}\\
& \frac{\partial}{\partial y}=\frac{2}{h_{y}} \sinh ^{-1}\left(\frac{\delta}{2}\right)=\frac{1}{h_{y}}\left(\delta_{y}-\frac{1^{2}}{2^{2} \cdot 3!} \delta_{y}^{3}+\frac{1^{2} \cdot 3^{2}}{2^{4} \cdot 5!} \delta^{5}-\ldots\right) \tag{2.6.2a}
\end{align*}
$$

and

$$
\begin{align*}
& \frac{\partial^{2}}{\partial x^{2}}=\frac{1}{h_{x}^{2}}\left(\delta_{x}^{2}-\frac{1}{12} \delta_{x}^{4}+\frac{1}{90} \delta_{x}^{6}-\cdots\right)  \tag{2.6.3}\\
& \frac{\partial^{2}}{\partial y^{2}}=\frac{1}{h_{y}^{2}}\left(\delta_{y}^{2}-\frac{1}{12} \delta_{y}^{4}+\frac{1}{90} \delta_{y}^{6}-\cdots\right) \tag{2.6.3a}
\end{align*}
$$

where

$$
\begin{align*}
& \delta_{x} u_{i, j}^{(k)}=\left(u_{i+\frac{1}{2}, j}^{(k)}-u_{i-\frac{1}{2}, j}^{(k)}\right) ; \delta_{y} u_{i, j}^{(k)}=\left(u_{i, j+\frac{1}{2}}^{(k)}-u_{i, j-\frac{1}{2}}^{(k)}\right),  \tag{2.6.4}\\
& \delta_{x}^{2} u_{i, j}^{(k)}=\left(u_{i+1, j}^{(k)}-2 u_{i, j}^{(k)}+u_{i-1, j}^{(k)}\right) ; \delta_{y}^{2} u_{i, j}^{(k)}=\left(u_{i, j+1}^{(k)}-2 u_{i, j}^{(k)}+u_{i, j-1}^{(k)}\right) \tag{2.6.5}
\end{align*}
$$

and so on.

We consider now the following approximations

$$
\frac{\partial}{\partial x}\left(\Lambda \frac{\partial}{\partial x}\right)=\frac{1}{h_{x}^{2}} \delta_{x}\left(\Lambda \delta_{x}\right) ; \quad \frac{\partial y}{\partial y}\left(\Lambda \frac{a}{\partial y}\right)=\int_{h^{2}}^{j^{j}} y\left(\Lambda j_{y}\right)
$$

where

$$
\begin{align*}
& \delta_{x}\left(\Lambda \delta_{x}\right) u_{i, j}^{(k)}=\Lambda_{i+\frac{1}{2}, j}^{(k)}\left(u_{i+1, j}^{(k)}-u_{i, j}^{(k)}\right)-\Lambda_{i-\frac{1}{2}, j}^{(k)}\left(u_{i, j}^{(k)}-u_{i-1, j}^{(k)}\right)  \tag{2.6.7}\\
& \delta_{y}\left(\Lambda \delta_{y}\right) u_{i, j}^{(k)}=\Lambda_{i, j+\frac{1}{2}}^{(k)}\left(u_{i, j+1}^{(k)}-u_{i, j}^{(k)}\right)-\Lambda_{i, j-\frac{1}{2}}^{(k)}\left(u_{i, j}^{(k)}-u_{i, j-1}^{(k)}\right) \tag{2.6.8}
\end{align*}
$$

### 2.7 EXPLICIT AND IMPLICIT METHODS

It can be easily seen that equation (2.5.1) can be
written as

$$
\begin{equation*}
U_{i, j}^{(k+1)}=\exp \left\{\Delta t\left[\frac{\partial}{\partial x}\left(\Lambda \frac{\partial}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Lambda \frac{\partial}{\partial y}\right)\right]\right\} U_{i, j}^{(k)} \tag{2.7.1}
\end{equation*}
$$

Then, the substitution of relationships (2.6.6) into equation (2.7.1) and expansion of the right hand side of (2.7.1), neglecting terms of order $\Delta t^{2}$ and above, leads to the following standard explicit formula, involving five grid points at $t=k . \Delta t$,

$$
\begin{equation*}
u^{(k+1)}=\left\{1+r_{x} \delta_{x}\left(\Lambda \delta_{x}\right)+r_{y} \delta_{y}\left(\Lambda \delta_{y}\right)\right\} u^{(k)} \tag{2.7.2}
\end{equation*}
$$

where

$$
\begin{equation*}
r_{x}=\Delta t / h_{x}^{2}, \quad r_{y}=\Delta t / h_{y}^{2} \tag{2.7.3}
\end{equation*}
$$

are the mesh ratios in the $x, i \operatorname{directions}$ and $u^{(k)} \operatorname{denotes} i_{i, j}^{(k)}$.
Then, combination of (2.6.7),(2.6.8),(2.7.2) gives

$$
\begin{align*}
u_{i, j}^{(k+1)}= & {\left[1-r_{x}\left(\Lambda_{i+\frac{1}{2}, j}^{(k)}+\Lambda_{i-\frac{1}{2}, j}^{(k)}\right)-r_{y}\left(\Lambda_{i, j+\frac{1}{2}}^{(k)}+\Lambda_{i, j-\frac{1}{2}}^{(k)}\right)\right] u_{i, j}^{(k)}+} \\
& +r_{x}\left(\Lambda_{i+\frac{1}{2}, j}^{(k)} u_{i+1, j}^{(k)}+\Lambda_{i-\frac{1}{2}, j}^{(k)} u_{i-1, j}^{(k)}\right)+r_{y}^{\left(\Lambda_{i, j+\frac{1}{2}}^{(k)} u_{i, j+1}^{(k)}+\Lambda_{i, j-\frac{1}{2}}^{(k)} u_{i, j-1}^{(k)}\right)} \tag{2.7.4}
\end{align*}
$$

In order to find the local truncation error of (2.7.4), we consider the error vector at the mesh point (ih $x_{x}, j h_{y}, k \Delta t$ ) defined by:

$$
\begin{equation*}
e_{i, j}^{(k)}=u_{i, j}^{(k)}-u_{i, j}^{(k)} . \tag{2.7.5}
\end{equation*}
$$

Using Taylor series expansion (see section 1.2, equation
(2.6.1)) and from (2.7.4), (2.7.5) we have

$$
\begin{align*}
e_{i, j}^{(k+1)}= & {\left[1-r_{x}\left(\Lambda_{i+\frac{1}{2}, j}^{(k)}+\Lambda_{i-\frac{1}{2}, j}^{(k)}\right)-r_{y}\left(\Lambda_{i, j+\frac{1}{2}}^{(k)}+\Lambda_{i, j-\frac{1}{2}}^{(k)}\right)\right] e_{i, j}^{(k)}+} \\
& +r_{x}\left(\Lambda_{i+\frac{1}{2}, j}^{(k)} e_{i+1, j}^{(k)}+\Lambda_{i-\frac{1}{2}, j}^{(k)} e_{i-1, j}^{(k)}\right)+r_{y}^{\left(\Lambda \Lambda_{i, j+\frac{1}{2}}^{(k)} e_{i, j+1}^{(k)}+\Lambda_{i, j-\frac{1}{2}}^{(k)} e_{i, j-1}^{(k)}\right)} \\
& +\frac{(\Delta t)^{2}}{2}\left[\frac{\partial^{2} U}{\partial t^{2}}\right]_{i, j}^{k}-(\Delta t) \frac{h_{x}^{2}}{12}\left[\frac{\partial^{4} U_{+}}{\partial x^{4}} \frac{\partial^{4} U}{\partial y^{4}}\right]_{i, j}^{k}-(\Delta t) \frac{h^{2}}{12}\left[\frac{\partial^{4} U^{4}}{\partial x^{4}}+\frac{\partial^{4} U}{\partial y^{4}}\right]_{i, j}^{k}+\ldots \tag{2.7.6}
\end{align*}
$$

Assuming now that the adjacent points to an internal point ( $i, j$ ) belong to the region $R$ under consideration then from (2.7.6) we get that the local truncation error of (2.7.4) is

$$
\begin{equation*}
0\left(\Delta t^{2}+\Delta t\left(h_{x}^{2}+h_{y}^{2}\right)\right) \tag{2.7.7}
\end{equation*}
$$

Note that if the region $R$ under consideration is nonrectangular then a special treatment for the internal grid points adjacent to the boundary is required.


FIGURE 2.7

In such a case, let

$$
(i+1, j),\left(i-\varepsilon_{1}, j\right),\left(i, j+\varepsilon_{2}\right),(i, j-1)
$$

be the adjacent points to the internal point (i,j). The mesh points (i- $\left.\varepsilon_{1}, j\right),\left(i, j+\varepsilon_{2}\right)$ are not exactly distance $h_{x}, h_{y}$ away but $\varepsilon_{1} h_{x}$, $\varepsilon_{2} h_{y}$ respectively with $\varepsilon_{1}, \varepsilon_{2} \in(0,1]$, (Figure 2.7).
Then the following finite difference approximations are obtained

$$
\frac{\partial}{\partial x}\left(\Lambda \frac{\partial U}{\partial x}\right) \simeq \frac{1}{1+\varepsilon_{1}} \frac{2}{h_{x}}\left\{\Lambda_{i+\frac{1}{2}, j}\left[\frac{u_{i+1, j}-u_{i, j}}{h_{x}}\right]-\Lambda_{i-\varepsilon_{1} / 2, j}\left[\frac{u_{i, j}-u_{i-\varepsilon_{1}, j}}{\varepsilon_{1} h_{x}}\right]\right\}
$$

and the difference equation valid at an internal point such as ( $\mathrm{i}, \mathrm{j}$ ) in Figure 2.7, is given by an analogous equation to (2.7.4) as follows:

$$
\begin{gather*}
u_{i, j}^{(k+1)}=\left[1-\frac{2 r_{x}}{1+\varepsilon_{1}}\left(\Lambda_{i+\frac{1}{2}, j}^{(k)}+\frac{1}{\varepsilon_{1}} \Lambda_{i-\varepsilon_{1} / 2, j}^{(k)}\right)-\frac{2 r_{y}}{1+\varepsilon_{2}}\left(\frac{1}{\varepsilon_{2}} \Lambda_{i, j+\varepsilon_{2} / 2}^{(k)}+\Lambda_{i, j-\frac{1}{2}}^{(k)}\right] u_{i, j}^{(k)}+\right. \\
+\frac{2 r_{x}}{1+\varepsilon_{1}}\left(\Lambda_{i+\frac{1}{2}, j}^{(k)} u_{i+1, j}^{(k)}+\frac{1}{\varepsilon_{1}} \Lambda_{i-\varepsilon_{1} / 2, j}^{(k)} u_{i-\varepsilon_{1}, j}^{(k)}\right) \\
 \tag{2.7.8b}\\
+\frac{2 r_{y}^{1+\varepsilon_{2}}\left(\frac{1}{\varepsilon_{2}} \Lambda_{i, j+\varepsilon_{2} / 2}^{(k)} u_{i, j+\varepsilon_{2}}^{(k)}+\Lambda_{i, j-\frac{1}{2}}^{(k)} u_{i, j-1}^{(k)}\right)}{}
\end{gather*}
$$

Since this thesis is concerned only with Implicit methods, the Explicit method has been included only for completeness. It is well known [52], [43] that the latter method, in the case of constant coefficients ( $\Lambda(x, y) \equiv 1$ ) in equation (2.5.1), is conditionally stable against the growth of rounding errors only when $\frac{\Delta t}{h_{x}^{2}} \frac{1}{4}$ and $\frac{\Delta t}{h_{y}^{2}} \leqslant \frac{1}{4}$. Given that the application of Explicit
difference methods for the solution of initial boundary value problems, in two or more space dimensions, is limited because of poor stability properties, the Implicit methods with superior stability properties are always used.

For example, formula (2.7.1) can be written as

$$
\begin{equation*}
\exp \left\{-\Delta t\left[\frac{\partial}{\partial x}\left(\Lambda \frac{\partial}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Lambda \frac{\partial}{\partial y}\right)\right]\right\} U^{(k+1)}=\dot{U}^{(k)} \tag{2.7.9}
\end{equation*}
$$

where $U^{(k)}$ denotes $U_{i, j}^{(k)}$. Then, combination of (2.6.6), (2.7.9) leads to

$$
\begin{equation*}
\exp \left(-r_{x} \delta_{x}\left(\Lambda \delta_{x}\right)\right) \exp \left(-r_{y} \delta_{y}\left(\Lambda \delta_{y}\right)\right) u^{(k+1)} \simeq u^{(k)} \tag{2.7.9a}
\end{equation*}
$$

where the mesh ratios $r_{x}, r_{y}$ and $\delta_{x}\left(\Lambda \delta_{x}\right), \delta_{y}\left(\Lambda \delta_{y}\right)$ are given by (2.7.3) and (2.6.7), (2.6.8) respectively. The expansion of the left-hand side of equation (2.7.9a), neglecting terms of order $\Delta t^{2}$ and above, gives the following two dimensional 'fully implicit' scheme:

$$
\begin{equation*}
\left[1-r_{x} \delta_{x}\left(\Lambda \delta_{x}\right)-r_{y} \delta_{y}\left(\Lambda \delta_{y}\right)\right] u^{(k+1)}=u^{(k)}+0\left(\Delta t^{2}+\Delta t\left(h_{x}^{2}+h_{y}^{2}\right)\right) \tag{2.7.9b}
\end{equation*}
$$

Advantages of the above formula is stability but we go on to discuss the 'Crank-Nicolson' formula in detail, which is not only stable but also of higher order accuracy.

The exact difference replacement formula (2.7.1) can be re-written as

$$
\begin{align*}
& \exp \left\{-\frac{1}{2} \Delta t\left[\frac{\partial}{\partial x}\left(\Lambda \frac{\partial}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Lambda \frac{\partial}{\partial y}\right)\right]\right\} U^{(k+1)}= \\
&=\exp \left\{\frac{1}{2} \Delta t\left[\frac{\partial}{\partial x}\left(\Lambda \frac{\partial}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Lambda \frac{\partial}{\partial y}\right)\right]\right\} U^{(k)} \tag{2.7.10}
\end{align*}
$$

and assuming that the approximations (2.6.6) are substituted in (2.7.10) we get

$$
\begin{align*}
\exp \left(-\frac{1}{2} r_{x} \delta_{x}\left(\Lambda \delta_{x}\right)\right) & \exp \left(-\frac{1}{2} r_{y} \delta_{y}\left(\Lambda \delta_{y}\right)\right) u^{(k+1)}= \\
& =\exp \left(\frac{1}{2} r_{x} \delta_{x}\left(\Lambda \delta_{x}\right)\right) \exp \left(\frac{1}{2} r_{y} \delta_{y}\left(\Lambda \delta_{y}\right)\right) u^{(k)} \tag{2.7.10a}
\end{align*}
$$

If we expand both sides of equation (2.7.10a), neglecting terms of order $\Delta t^{2}$ and above, and average the operation over two time intervals, we obtain the following 'Crank-Nicolson' formulae in two space dimensions [13]:

$$
\begin{gather*}
{\left[1-\frac{1}{2} r_{x} \delta_{x}\left(\Lambda \delta_{x}\right)-\frac{1}{2} r_{y} \delta_{y}\left(\Lambda \delta_{y}\right)\right] u^{(k+1)}=\left[1+\frac{1}{2} r_{x} \delta_{x}\left(\Lambda \delta_{x}\right)+\frac{1}{2} r_{y} \delta_{y}\left(\Lambda \delta_{y}\right)\right] u^{(k)}+} \\
+0\left(\Delta t^{3}+\Delta t\left(h_{x}^{2}+h_{y}^{2}\right)\right) . \tag{2.7.10b}
\end{gather*}
$$

lt is well known [52] that the Crank-Nicolson implicit formula is unconditionally stable against the growth of the rounding errors.

A great number of implicit finite difference methods have been proposed for the solution of initial boundary value problems involving linear parabolic equations in the region $\overline{\mathrm{R}}$, [52],[27], [28],[21]. Some governing criteria for the choice of the best implicit finite difference method in a particular problem are:
(i) the nature of the coefficients in the parabolic equation,
(ii) the shape of the region $R$ under consideration,
(iii) the type of boundary condition on $C$ for $t>0$.

From relationships (2.6.6)-(2.6.8) and Crank-Nicolson finite difference formula (2.7.10b) the finite difference discretization of equation (2.5.1) on the chosen grid (see Figure 2.8), can be written as a series of five-point linear finite difference equations of the form:

$$
\begin{array}{r}
d_{i, j} u_{i, j-1, k+1}+a_{i, j} u_{i-1, j, k+1}+b_{i, j} u_{i, j, k+1}+c_{i, j} u_{i+1, j, k+1}+ \\
+e_{i, j} u_{i, j+1, k+1}=s_{i, j, k}  \tag{2.7.11}\\
\text { for } i \varepsilon[1, p-1], j \varepsilon[1, m-1]
\end{array}
$$

where,

$$
\begin{align*}
& d_{i, j}=-\frac{r^{y}}{2} \Lambda_{i, j-\frac{1}{2}} ; e_{i, j}=-\frac{r_{y}}{2} \Lambda_{i, j+\frac{1}{2}} ;  \tag{2.7.12}\\
& a_{i, j}=-\frac{r_{x}}{2} \Lambda_{i-\frac{1}{2}, j} ; c_{i, j}=-\frac{r_{x}}{2} \Lambda_{i+\frac{1}{2}, j} ;  \tag{2.7.13}\\
& b_{i, j}=1+\frac{r_{y}}{2}\left[\Lambda_{i, j-\frac{1}{2}}+\Lambda_{i, j+\frac{1}{2}}\right]+\frac{r_{x}}{2}\left[\Lambda_{i-\frac{1}{2}, j}+\Lambda_{i+\frac{1}{2}, j}\right] ; \tag{2.7.14}
\end{align*}
$$

and

$$
\begin{align*}
s_{i, j, k}=u_{i, j, k} & +\frac{r_{x}}{2}\left[\Lambda_{i+\frac{1}{2}, j}\left(u_{i+1, j, k}-u_{i, j, k}\right)-\Lambda_{i-\frac{1}{2}, j}\left(u_{i, j, k}-u_{i-1, j, k}\right)\right]+ \\
& +\frac{r_{y}}{2}\left[\Lambda_{i, j+\frac{1}{2}}\left(u_{i, j+1, k}-u_{i, j, k}\right)-\Lambda_{i, j-\frac{1}{2}}\left(u_{i, j, k}-u_{i, j-1, k}\right)\right] \tag{2.7.15}
\end{align*}
$$

with $\quad r_{x}=\Delta t / h_{x}^{2}, \quad r_{y}=\Delta t / h_{y}^{2}$.


FIGURE 2.8

Grouping the above system of finite difference equatıons into matrix form, we obtain a sparse, quindiagonal matrix of order $n=(m-1)(p-1)$ of the following form:

where $c_{q}=0$ if $q=t(m-1), 1 \leqslant t \leqslant p-2$

$$
a_{q}=0 \text { if } q=t(m-1)+1, \quad 1 \leqslant t \leqslant p-2
$$

and the obtained system

$$
\begin{equation*}
\mathrm{A} \underline{\mathrm{u}}=\underline{\mathrm{s}} \tag{2.7.17}
\end{equation*}
$$

has to be solved at each time step, to give the required solution to the problem.

Remark
A fully Implicit finite difference formula in two space dimensions is given in (2.7.9b). In such a case, the coefficients of the five-point linear finite difference equation (2.7.11) are given as follows:

$$
\begin{align*}
& d_{i, j}=-r_{y} \Lambda_{i, j-\frac{1}{2}} ; \quad e_{i, j}=-r_{y} \Lambda_{i, j+\frac{1}{2}} ;  \tag{2.7.18}\\
& a_{i, j}=-r_{x} \Lambda_{i-\frac{1}{2}, j} ; \quad c_{i, j}=-r_{x} \Lambda_{i+\frac{1}{2}, j} ;  \tag{2.7.19}\\
& b_{i, j}=1+r_{x}\left(\Lambda_{i+\frac{1}{2}, j}+\Lambda_{i-\frac{1}{2}, j}\right)+r_{y}\left(\Lambda_{i, j+\frac{1}{2}}+\Lambda_{i, j-\frac{1}{2}}\right) \tag{2.7.20}
\end{align*}
$$

and

$$
\begin{equation*}
s_{i, j, k}=u_{i, j, k} \tag{2.7.21}
\end{equation*}
$$

### 2.8 A PARABOLIC 3D-EXAMPLE

We consider now the application of the finite difference method to obtain an approximate solution for the parabolic three dimensional P.D.E. defined by:

$$
\begin{equation*}
\frac{\partial}{\partial x}\left(A(x, y, z) \frac{\partial U}{\partial x}\right)+\frac{\partial}{\partial y}\left(C(x, y, z) \frac{\partial U}{\partial y}\right)+\frac{\partial}{\partial z}\left(E(x, y, z) \frac{\partial U}{\partial z}\right)=F(x, y, z) \frac{\partial U}{\partial t}+Q(x, y, z) \tag{2.8.1}
\end{equation*}
$$

in the region $\bar{R}=R \times[0<t \leqslant T]$,
where $\quad R=\{(x, y, z) ; 0<x, y, z<1\}$,
with the initial condition,

$$
\begin{equation*}
U(x, y, z, 0)=f(x, y, z), \quad(x, y, z) \varepsilon R \tag{2.8.4}
\end{equation*}
$$

subject to the boundary conditions

$$
\begin{equation*}
U(x, y, z, t)=g(x, y, z, t), \quad(x, y, z, t) \varepsilon \tilde{C} \times[0 \leqslant t \leqslant T] \tag{2.8.5}
\end{equation*}
$$

where $\tilde{C}$ is the boundary of $R$, and $A, C, E, F$ are known coefficients (in general they are functions of position). $Q$ is a source term, $x, y, z$ are the position co-ordinates and $U$ is the dependent variable.

We cover the region $R$ with a volumetric grid system $R_{h}$, with spacings $h_{x}, h_{y}, h_{z}$ and $\ell$ in $X, Y, Z$ and $t$ directions respectively and mesh points $\left(x_{i}, y_{j}, z_{k}, t_{r}\right)$, defined by

$$
R_{h}=\left\{\left(i h_{x}, j h_{y}, k h_{z}, r \ell\right): \begin{array}{l}
0 \leqslant i \leqslant \zeta+1  \tag{2.8.6}\\
0 \leqslant j \leqslant \theta+1 \\
0 \leqslant k \leqslant \xi+1 \\
0 \leqslant r \leqslant T / \ell
\end{array}\right\}
$$

The three dimensional molecule is shown in Figure 2.9.


FIGURE 2.9

Assuming that $P(i, j, k)$ and $u_{i, j, k}$ denote. the grid point $P^{\prime}\left(i h_{x}, j h_{y}, k h_{z}\right)$ and $u\left(i h_{x}, j h_{y}, k h_{z}\right)$ respectively, discrete approximations to the partial derivatives in (2.8.1) are used to derive the following linear finite difference equations at the point $P(i, j, k)$

$$
\begin{align*}
& A\left(x+\frac{1}{2} h_{x}, y, z\right)\left[\frac{u\left(x+h_{x}, y, z\right)-u(x, y, z)}{h_{x}^{2}}\right]-A\left(x-\frac{1}{2} h_{x}, y, z\right)\left[\frac{u(x, y, z)-u\left(x-h_{x, y}, z\right)}{h_{x}^{?}}\right]+ \\
& +C\left(x, y+\frac{1}{2} h_{y}, z\right)\left[\frac{u\left(x, y+h_{y}, z\right)-u(x, y, z)}{h_{y}^{2}}\right]-C\left(x, y-\frac{1}{2} h_{y}, z\right)\left[\frac{u(x, y, z)-u\left(x, y-h_{y}, z\right)}{h_{y}^{2}}\right]+ \\
& +E\left(x, y, z+\frac{1}{2} h_{z}\right)\left[\frac{u\left(x, y, z+h_{z}\right)-u(x, y, z)}{h_{z}^{2}}\right]-E\left(x, y, z-\frac{1}{2} h_{z}\right)\left[\frac{u(x, y, z)-u\left(x, y, z-h_{z}\right.}{h_{z}^{2}}\right. \\
& -\frac{1}{\Lambda \uparrow} F(x, y, z)[u(x, y, z)-\tilde{u}(x, y, z)]=Q(x, y, z), \tag{2.8.7}
\end{align*}
$$

where $u(x, y, z)$ and $\tilde{u}(x, y, z)$ are the values of the dependent variable at the $(\ell+1)$ and $\ell$ time levels respectively and $\Delta t$ is the time step.

When the seven-point, three dimensional molecule (Figure 2.9) is used, we obtain the following seven-point finite difference equation:

$$
\begin{gather*}
D_{i, j, k} u_{i, j, k}+I_{i, j, k} u_{i, j, k-1}+\square_{i, j, k} u_{i, j, k+1}^{+F_{i, j}, k} u_{i, j+1, k}+B_{i, j, k} u_{i, j-1, k}^{+} \\
+R_{i, j, k} u_{i+1, j, k}+L_{i, j, k} u_{i-1, j, k}=S_{i, j, k},  \tag{2.8.8}\\
\text { for } i \varepsilon[1, \zeta], \quad j \in[1, \theta], \quad k \varepsilon[1, \xi],
\end{gather*}
$$

where

$$
\begin{align*}
& D_{i, j, k}=-\left(A_{i+\frac{1}{2}, j, k}+A_{i-\frac{1}{2}, j, k}\right) \frac{h_{y} h_{z}}{h_{x}}-\left(C_{i, j+\frac{1}{2}, k}+C_{i, j-\frac{1}{2}, k}\right) \frac{h_{x} h_{z}}{h_{y}}- \\
& -\left(E_{i, j, k+\frac{1}{2}}+E_{i, j, k-\frac{1}{2}}\right) \frac{h_{x} h_{y}}{h_{z}}-F_{i, j, k} \frac{h_{x} h^{h} h_{z}}{\Delta t} \text {; }  \tag{2.8.9}\\
& I_{i, j, k}=E_{i, j, k-\frac{1}{2}} \frac{h_{x} h}{h}{ }_{z} ; \square_{i, j, k}=E_{i, j, k+\frac{1}{2}} \frac{h^{h} y}{h_{z}} ;  \tag{2.8.10}\\
& T_{i, j, k}=C_{i, j+\frac{1}{2}, k} \frac{h_{x} h_{z}}{h_{y}} ; \quad B_{i, j, k}=C_{i, j-\frac{1}{2}, k-\frac{h_{x}}{h_{y}}}^{-} ;  \tag{2.8.11}\\
& R_{i, j, k}=A_{i+\frac{1}{2}, j, k} \frac{h_{y} h_{z}}{h_{x}} ; \quad L_{i, j, k}=A_{i-\frac{1}{2}, j, k} \frac{h_{y} h_{z}}{h_{x}} ;  \tag{2.8.12}\\
& \text { and } S_{i, j, k}=\left(Q_{i, j, k}-\frac{1}{\Delta t} F_{i, j, k} \tilde{u}_{i, j, k}\right) h_{x} h_{y} h_{z} .
\end{align*}
$$

We order the points of $R_{h}$ with increasing values of $j$, then $i$ and then $k$. If we order equation (2.8.8) in the same sequence then a set of difference equations is obtained which may be expressed in matrix notation as:

$$
\begin{equation*}
\Omega \underline{\mathrm{u}}=\underline{\mathrm{s}} \tag{2.8.14}
\end{equation*}
$$

where $\underline{u}$ and $\underline{s}$ are $[\zeta \theta \xi \times 1]$ column vectors, which consist of the unknown approximate solutions $u_{i, j, k}$ and the right hand side
quantity (2.8.13) plus the fnown boundary values respectively, and $\Omega$ is a seven diagonal, sparse matrix of order $n=\zeta \theta \xi$ and bandwidths $\mathrm{m}=\theta+1, \mathrm{p}=\theta \zeta+1$.

where: $c_{q}=0$, if $q=t \theta, 1 \leqslant t \leqslant \xi \zeta-1$

$$
a_{q}=0 \text {, if } q=t \theta+1,1 \leqslant t \leqslant \xi \zeta-1
$$

and which has to be solved to give the required solution to the problem at each time step.

## Chapter 3

## BASIC ALGORITHMS FOR TWO AND THREE <br> DIMENSIONAL P.D.E.'S

### 3.1 INTRODUCTION

In this Chapter we introduce algorithmic solution methods for the large sparse linear systems derived from finite-difference discretization of parabolic and elliptic p.d.e's in both two and three space dimensions. The coefficient matrix is shown to be factorized exactly to yield direct algorithmic procedures for the finite difference solution.
3.2 THE ALGORITHM FOR THE SOLUTION OF LARGE, UNSYMMETRIC, QUINDIAGONAL SPARSE LINEAR SYSTEMS (THE LUBOT ALGORITHM)

An investigation into the current implicit solution processes involving the diffusion equation with one space dimension reveals the fact that extensive use is made of the tridiagonal matrix algorithm [57] for solving the three term finite difference equations.

For the two dimensional case, the implicit methods lead to the requirement that large sparse matrices have to be solved for each time interval, a compact Gaussian elimination process without interchanges and an approximate elimination scheme have been derived [24] with the use of an echelon 'moving frame' process to devise an efficient algorithmic process by which the Gaussian factorization procedure is carried out.

In this section we introduce a sparse triangular factorization LU method for the solution of similar problems by use of a matrix 'bordering" technique [34], [26].

Let A (a large order, diagonally dominant, unsymmetric, quindiagonal, sparse matrix) be factorized into the product of two matrices L, U i.e.,

$$
\begin{equation*}
\mathrm{A}=\mathrm{L} . \mathrm{U} \tag{3.2.1}
\end{equation*}
$$

where $A$ is defined in (2.7.16)

L is a strictly lower triangular matrix and
U is a strictly upper triangular matrix with unit diagonal elements given below.

The coefficient matrix $A$ can be written in the following partitioned form:-


111



Replacing the right hand side of (3.2.1) by its partitioned
forms leads to

and on equating with the partitioned form of $A$ in (3.2.2) we obtain the identities:

$$
\begin{array}{ll}
\mathrm{A}_{11}=\mathrm{L}_{11} \mathrm{U}_{11} ; & \mathrm{A}_{12}=\mathrm{L}_{11} \mathrm{U}_{12} ; \\
\mathrm{A}_{21}=\mathrm{L}_{21} \mathrm{U}_{11} ; & \mathrm{A}_{22}=\mathrm{L}_{21} \mathrm{U}_{12}+\mathrm{L}_{22} \mathrm{U}_{22} . \tag{3.2.6b}
\end{array}
$$

The elements of $\mathrm{L}_{11}, \mathrm{U}_{11}$ are well known and may be easily obtained by applying the tridiagonal algorithm [57] as follows:

$$
\begin{align*}
& \omega_{1}=b_{1} ; \quad \beta_{1}=a_{2} ; \quad g_{1}=c_{1} / \omega_{1},  \tag{3.2.7a}\\
& \text { for } i=2,3, \ldots, m-2 \\
& \omega_{i}=b_{i}-\beta_{i-1} g_{i-1} ; \quad \beta_{i}=a_{i+1} ; \quad g_{i}=c_{i} / \omega_{i} \tag{3.2.7b}
\end{align*}
$$

and

$$
\begin{equation*}
\omega_{m-1}=b_{m-1}-\beta_{m-2} g_{m-2} . \tag{3.2.7c}
\end{equation*}
$$

The elements of $L_{21}, U_{21}, L_{22}$ and $U_{22}$ can be obtained from the following relations:
for $j=1,2, \ldots, n-m+1$

$$
\begin{equation*}
h_{1, j}=\tau_{j} / \omega_{j} ; \quad \gamma_{1, j}=v_{j+m-1} \tag{3.2.8}
\end{equation*}
$$

and $\quad g_{m+j-2}=c_{m+j-2} / \omega_{m+j-2} ; \quad \beta_{m+j-2}=a_{m+j-1}$,
whilst for $\mathrm{j} \leqslant \mathrm{m}-2$

$$
\begin{align*}
& h_{i, j}=-\beta_{i+j-2} h_{i-1, j} / \omega_{i+j-1} \text {, for } i=2,3, \ldots, m-j  \tag{3.2.10}\\
& \text { and } \\
& \gamma_{i, j}=-g_{i+j-2} \gamma_{i-1, j} \quad \text { for } i=2,3, \ldots, m-j \text {. } \tag{3.2.11}
\end{align*}
$$

Then, for all $j>1$

$$
\begin{equation*}
\gamma_{i, j}=-g_{i+j-2} \gamma_{i-1, j}-\sum_{k=1}^{i-1} \gamma_{k, j} h_{k-i+m, i+j-m} \tag{3.2.12}
\end{equation*}
$$

and

$$
\begin{equation*}
h_{i, j}=\left(-\beta_{i+j-2} h_{i-1, j}-\sum_{k=1}^{i-1} \gamma_{k-i+m, i+j-m} h_{k, j}\right) / \omega_{i+j-1}, \tag{3.2.13}
\end{equation*}
$$

for either $i=(m-j+1),(m-j+2), \ldots,(m-1)$ and all $j \leqslant m-1$
or $\quad i=2,3, \ldots, m-1$ for $j>m-1$.
Then, for $\mathrm{i}=\mathrm{m}-1$

$$
\begin{equation*}
\omega_{m+j-1}=b_{m+j-1}-\beta_{i+j-1} h_{i, j} \quad-g_{i+j-1} \gamma_{i, j}-\beta_{i+j-1} g_{i+j-1}-\sum_{k=1}^{i} \gamma_{k, j} h_{k, j} . \tag{3.2.14}
\end{equation*}
$$

The original linear system

$$
\begin{equation*}
\text { A. } \underline{u}=\underline{s} \tag{3.2.15}
\end{equation*}
$$

can then be solved by rewriting (3.2.15) as

$$
\begin{equation*}
\text { L.U. } \underline{u}=\underline{s} . \tag{3.2.16}
\end{equation*}
$$

This can be solved directly for $\underline{u}$ in terms of an auxiliary vector $\underline{y}$ where

$$
\begin{equation*}
\mathrm{U} \underline{\mathrm{u}}=\underline{\mathrm{y}} \quad \text { and } \quad \mathrm{L} \underline{y}=\underline{\mathrm{s}} . \tag{3.2.17}
\end{equation*}
$$

The initial processes yield the algorithm

$$
\begin{align*}
& y_{1}=s_{1} / \omega_{1} ;  \tag{3.2.18}\\
& y_{i}=\left(s_{i}-\beta_{i-1} y_{i-1}\right) / \omega_{i_{i}}, \text { for } i=2,3, \ldots, m-1  \tag{3.2.19}\\
& y_{i}=\left(s_{i}-\beta_{i-1} y_{i-1} \sum_{k=i-m+1} \gamma_{k-i+m, i-m+1} y_{k}\right) / \omega_{i}, \\
& \text { for } i=m, m+1, \ldots, n \tag{3.2.20}
\end{align*}
$$

and

Then a back substitution process yields $\underline{u}$ in terms of $\underline{y}$, the components being given by the equations:
and $\quad u_{i}=y_{i}-g_{i} u_{i+1}-\sum_{j=p}^{q} h_{i-j+m, j-m+1} y_{j}$,

$$
\begin{equation*}
u_{n}=y_{n} \tag{3.2.21}
\end{equation*}
$$

where $p, q$ are given by:
If $m>\left[\frac{n}{2}+1\right]$

| $p=i+1$, | $q=n$ | for $i=n-1, n-2, \ldots, m$ (except if $m=n$ ) |
| :--- | :--- | :--- |
| $p=m$, | $q=n$ | for $i=m-1, m-2, \ldots, n-m+1$ |
| $p=m$, | $q=i+m-1$ | for $i=n-m, n-m-1, \ldots, 2,1$ (except if $m=n)$. |

If $m<\left[\frac{n}{2}+1\right]$
If $n$ is even and $m=\frac{n}{2}$,
then

| $p=i+1$ | ,$q=n$ | for $i=n-1, n-2, \ldots, m+1$ |
| :--- | :--- | :--- |
| $p=m+1$ | ,$q=n-1$ | for $i=m$ |
| $p=m$ | ,$q=i+m-1$ | for $i=m-1, m-2, \ldots, 2,1$. |


| $p=i+1$ | ,$q=n$ | for $i=n-1, n-2, \ldots, n-m+1$ |
| :--- | :--- | :--- |
| $p=i+1$ | ,$q=i+m-1$ | for $i=n-m, n-m-1, \ldots, m$ |
| $p=m$ | ,$q=i+m-1$ | for $i=m-1, m-2, \ldots, 2,1$. |

With [s] we denote the greatest integer not exceeding $s$.
If $\tau_{i}=0, i=1,2, \ldots, n-m+1$ and $v_{i}=0, i=m, m+1, \ldots, n$,
the algorithm (henceforth called the LUBOT algorithm) reduces to the form of the common tridiagonal system [57].

Storage requirements and computational work
Although the $\beta, \omega, g$ vector stores can be overwritten the $\gamma, h$ arrays and $s$ vector stores have to be strictly preserved. Given that the memory space required for the $h$ array is $(m-1)(N-m+1)$ words, the total memory requirement for the LUBOT algorithm is $\simeq(2 m+4) \mathrm{N}$ words.

The amount of work involved is given by:
(i) decompose $A_{11}=L_{11} U_{11} \quad \rightarrow \quad(2 m-4)$ operations
(ii) form $U_{12}=L_{11}^{-1} A_{12} \quad \rightarrow \quad\left(m^{2}-2 m+1\right)$ operations
(iii) solve for $L_{21}: L_{21} U_{11}=A_{21} \quad \rightarrow \quad\left(\frac{m^{2}-3 m+2}{2}\right)$ operations
(iv) decompose $A_{22}=L_{21} U_{12}+L_{22} U_{22} \quad \rightarrow \quad \simeq\left(m^{2}+m\right) N$ operations

Therefore, the total operations for the factorization stage are $\simeq\left(m^{2}+m\right) N$ and given that the forward-backward substitution process requires $\simeq(2 m+1) N$ operations, the total operations for this algorithm are $\simeq\left(m^{2}+3 m+1\right) N$, for $m \ll N$.
3. 3 THE NORMALIZED ALGORITHM FOR THE SOLUTION OF LARGE, SYMMETRIC, QUINDIAGONAL, SPARSE LINEAR SYSTEMS (THE NORMBAND ALGORITHM)

It is well known [14] that for a symmetric and positive definite matrix A there exists a unique factorization of the form:

$$
\begin{equation*}
\mathrm{A}=\mathrm{D} \mathrm{~T}^{\prime} \mathrm{TD} \tag{3.3.1}
\end{equation*}
$$

where $D$ is a unique, positive diagonal matrix,
T is a unique, real, upper triangular matrix with
unit diagonal elements and $T$ ' denotes the transpose of $T$.
With A given by:

$D$ and $T$ in (3.3.1) are of the form:

$$
\begin{equation*}
D=\operatorname{diag}\left\{d_{i}, d_{2}, \ldots . ., d_{n}\right\} \tag{3.3.3}
\end{equation*}
$$



The following normalized algorithm [6], due to Benson and Evans, gives the elements of $D$ and $T$,

$$
\begin{equation*}
d_{1}=\sqrt{a_{1}} ; \quad d_{i}=\left\{a_{i}-\left(\frac{b_{i-1}}{d_{i-1}}\right)^{2}\right\}^{\frac{1}{2}} ; e_{i-1}=\frac{b_{i-1}}{d_{i-1} d_{i}}, \tag{3.3.5}
\end{equation*}
$$

Then,

$$
\text { for } i=2,3, \ldots, m-1 \text {. }
$$

for $j=1,2,3, \ldots, n-m+1$

$$
\begin{equation*}
x_{1}=\frac{c_{j}}{d_{j}} ; \quad v=\frac{b_{m+j-2}}{d_{m+j-2}} \tag{3.3.6}
\end{equation*}
$$

whilst for $j \leqslant m-2$

$$
\begin{equation*}
x_{i}=-e_{i+j-2} x_{i-1}, \quad \text { for } i=2,3, \ldots, m-j \tag{3.3.7}
\end{equation*}
$$

Then, for all $\mathrm{j}>1$

$$
\begin{equation*}
x_{i}=-e_{i+j-2} x_{i-1}-\sum_{k=1}^{i-1} x_{k} t_{k-i+m, i+j-m} \tag{3.3.8}
\end{equation*}
$$

for either $i=(m-j+1),(m-j+2), \ldots,(m-1)$ and all $j \leqslant m-1$
or $\quad i=2,3, \ldots,(m-1)$ for $j>m-1$.
Then,
and

$$
d_{m+j-1}=\left\{a_{m+j-1}-\sum_{k=1}^{m-2} x_{k}^{2}-\left(x_{m-1}+v\right)^{2}\right\}^{\frac{1}{2}} ; e_{m+j-2}=\frac{v}{d_{m+j-1}} \text { (3.3.9) }
$$

$$
\begin{equation*}
t_{i, j}=\frac{x_{i}}{d_{m+j-1}}, \quad i=1,2, \ldots, m-1 . \tag{3.3.10}
\end{equation*}
$$

The linear system (3.2.15) can then be solved by rewriting it in
the form

$$
\begin{equation*}
\text { DT'TD. } \underline{u}=\underline{s} \tag{3.3.11}
\end{equation*}
$$

or equivalently

$$
\begin{equation*}
T^{\prime} T(\underline{D u})=D^{-1} \underline{s} \tag{3.3.12}
\end{equation*}
$$

and introducing the auxiliary vectors $\underline{y}$ and $\underline{g}$ where

$$
\begin{equation*}
\underline{y}=D \underline{u} \quad ; \quad \underline{g}=D^{-1} \underline{s} \tag{3.3.13}
\end{equation*}
$$

the problem is reduced to solving

$$
\mathrm{T}^{\prime} \mathrm{T} \underline{y}=\underline{\mathrm{g}}
$$

then $\underline{y}$ is obtained in terms of an auxiliary vector $\underline{h}$ where

$$
\begin{equation*}
\mathrm{T} \cdot \underline{y}=\underline{\mathrm{h}} \quad \text { and } \quad \mathrm{T}^{\prime} \underline{h}=\underline{\mathrm{g}} \tag{3.3.14}
\end{equation*}
$$

i.e.,

$$
\begin{equation*}
h_{1}=g_{1} ; \quad h_{i}=g_{i}-e_{i-1} h_{i-1}, i=2,3, \ldots, m-1 . \tag{3.3.15}
\end{equation*}
$$

and

$$
\begin{equation*}
h_{i}=g_{i}-e_{i-1} h_{i-1}-\sum_{k=i-m+1}^{i-1} t_{k-i+m, i-m+1} h_{k}, i=m, m+1, \ldots, n \tag{3.3.16}
\end{equation*}
$$

A back substitution process, in a similar manner as in section
3.2 gives the $\underline{y}$ in terms of $\underline{h}$ and finally the solution $\underline{u}$ is obtained from $\underline{u}=D^{-1} \underline{y}$.

The total memory requirements for the NORMBAND algorithm is $\simeq(m+3) N$ words.

The amount of work involved in the factorization is $\simeq\left(\frac{(m-1)(m-2)}{2}+3 m-1\right) N$ mults +N square roots.

The normalization and forward-backward substitution processes require respectively N divisions and $\simeq 2 \mathrm{mN}$ mults +N divisions.

Therefore, the total number of operations for this algorithm is $\simeq\left(\frac{(m-1)(m-2)}{2}+5 m+1\right) N$ mults $+N$ square roots, for $m \ll N$.

Note that although the factorization $A=D T$ 'TD is preferable for theoretical purposes, it has the disadvantage of involving N square roots, which is relatively costly in computer time, and in this case the factorization $A=L . U$ may be computationally more desirable.

### 3.4 THE ALGORITHM FOR THE SOLUTION OF LARGE, UNSYMMETRIC, SEVEN DIAGONAL, SPARSE LINEAR SYSTEMS (THE LUBOT-3D ALGORITHM)

In this section, by extending the techniques of algorithmic construction developed in section 3.2, we introduce a sparse LU triangular factorization method for the solution of a seven-diagonal unsymmetric, linear system of bandwidth m and p (LUBOT-3D Algorithm) which can be regarded as a natural extension to the LUBOT-2D algorithm.

Such systems are derived from the application of finite difference method to the solution of parabolic p.d.e's in three space dimensions (see section 2.8).

Let $\Omega$ (a large, diagonally dominant, sevendiagonal, sparse matrix) be factorised into the product of the two matrices $\overline{\mathrm{L}}, \overline{\mathrm{U}}, \mathrm{i} . \mathrm{e}$.

$$
\begin{equation*}
\Omega=\overline{\mathrm{L}} \cdot \overline{\mathrm{U}} \tag{3.4.1}
\end{equation*}
$$

where $\Omega$ as given in (2.8.15)
$\overline{\mathrm{L}}$ is a strictly lower triangular matrix and
$\overline{\mathrm{U}}$ is a strictly upper triangular matrix with unit
diagonal elements.

The coefficient matrix $\Omega$ can be rewritten in the following partitioned form



(3.4.4)

Equating the partitioned form of $\bar{L} \bar{U}$, which can be obtained similarly as in (3.2.5), with the partitioned form of $\Omega$ in (3.4.2) we obtain the following identities:

$$
\begin{array}{ll}
\Omega_{11}=\overline{\mathrm{L}}_{11} \overline{\mathrm{U}}_{11} ; & \Omega_{12}=\overline{\mathrm{L}}_{11} \overline{\mathrm{U}}_{12} ; \\
\Omega_{21}=\overline{\mathrm{L}}_{21} \overline{\mathrm{U}}_{11} ; & \Omega_{22}=\overline{\mathrm{L}}_{21} \overline{\mathrm{U}}_{12}+\overline{\mathrm{L}}_{22} \overline{\mathrm{U}}_{22} . \tag{3.4.5b}
\end{array}
$$

The elements of $\overline{\mathrm{L}}_{11}, \overline{\mathrm{U}}_{11}$ are known and may be obtained by applying the LUBOT-2D algorithm for the solution of general five term linear system of order $(\mathrm{p}-1)$ and semibandwidth $m$ [see relations (3.2.7)-(3.2.14)].

The elements of $\overline{\mathrm{U}}_{12}, \overline{\mathrm{~L}}_{21}, \overline{\mathrm{~L}}_{22}, \overline{\mathrm{U}}_{22}$ may be obtained as follows: for $j=1,2, \ldots, n-p+1$.

$$
\begin{array}{ll} 
& t_{1, j}=r_{j} / \omega_{j} ; \quad f_{1, j}=s_{p+j-1} \\
\text { and } & g_{p+j-2}=c_{p+j-2} / \omega_{p+j-2} ; \beta_{p+j-2}=a_{p+j-1},
\end{array}
$$

whilst for $j \leqslant m-2$

$$
\begin{align*}
t_{i, j}=\beta_{i+j-2} t_{i-1, j} / \omega_{i+j-1} ; f_{i, j}=-g_{i+j-2} f_{i-1, j} \\
\quad \text { for } i=2,3, \ldots, m-j \tag{3.4.8}
\end{align*}
$$

then, if $j=1$ and $i=p-m+1$

$$
\begin{equation*}
t_{i, j} \leftarrow t_{i, j}+\tau_{i+j-1} / \omega_{i+j-1} ; f_{i, j} \leftarrow f_{i, j}+v_{p+j-1} \tag{3.4.9}
\end{equation*}
$$

If $i \geqslant p-j+1$ or $j>p-2$, then
and

$$
\begin{equation*}
t_{i, j}=\left(-\beta_{i+j-2} t_{i-1, j}-\sum_{i-1}^{i-1} f_{k-i+p, i+j-p} t_{k, j}\right) / \omega_{i+j-1} \tag{3.4.10}
\end{equation*}
$$

$$
\begin{equation*}
f_{i, j}=-g_{i+j-2} f_{i-1, j}-\sum_{k=1}^{i=1} f_{k, j} t_{k-i+p, i+j-p} \tag{3.4.11}
\end{equation*}
$$

if $i=p-m+1$, then

$$
\begin{equation*}
t_{i, j} \leftarrow t_{i, j}+\tau_{i+j-1} / \omega_{i+j-1} ; f_{i, j} \leftarrow f_{i, j}+v_{p+j-1} \tag{3.4.12}
\end{equation*}
$$

else

$$
\begin{align*}
& \text { if } i<m, \text { then } \\
& t_{i, j}=\left(-\beta_{i+j-2} t_{i-1, j}-\sum_{k=1}^{i-1} \gamma_{k-i+m, i+j-m} t_{k, j}\right) / \omega_{i+j-1}  \tag{3.4.13}\\
& f_{i, j}=-g_{i+j-2} f_{i-1, j}-\sum_{k=1}^{i-1} f_{k, j} h_{k-i+m, i+j-m} \tag{3.4.14}
\end{align*}
$$

else

$$
\begin{equation*}
t_{i, j}=\left(-\beta_{i+j-2} t_{i-1, j}-\sum_{k=1}^{m-1} \gamma_{k, i+j-m} t_{k+i-m, j}\right) / \omega_{i+j-1} \tag{3.4.15}
\end{equation*}
$$

and

$$
\begin{equation*}
f_{i, j}=-g_{i+j-2} f_{i-1, j}-\sum_{k=1}^{m-1} f_{k+i-m, j} h_{k, i+j-m} \tag{3.4.16}
\end{equation*}
$$

if $i=p-m+1$, then

$$
\begin{equation*}
t_{i, j} \leftarrow t_{i, j}+\tau_{i+j-1} / \omega_{i+j-1} ; f_{i, j} \leftarrow f_{i, j}+v_{p+j-1} \tag{3.4.17}
\end{equation*}
$$

for either $i=(m-j+1),(m-j+2), \ldots, p-1$ and all $j \leqslant m-1$
or

$$
i=2,3, \ldots, p-1 \quad \text { for } \quad j>m-1
$$

Then, for $i=p-1$

$$
\begin{equation*}
\omega_{p+j-1}=b_{p+j-1}-\beta_{1+j-1} t_{i, j}-\beta_{i+j-1} g_{i+j-1}-g_{i+j-1} f_{i, j}-\sum_{k=1}^{i} f_{k, j} t_{k, j} \tag{3.4.18}
\end{equation*}
$$

The standard notation " $\theta_{1} \leftarrow \theta_{2}$ ", see [37], where $\theta_{1}$ is variable and $\theta_{2}$ is variable or formula, means the value of $\theta_{1}$ is to be replaced by the value of $\theta_{2}$.

The system (2.8.14) can be solved by rewriting as before

$$
\begin{equation*}
\bar{L} \bar{U} \underline{u}=\underline{s} . \tag{3.4.19}
\end{equation*}
$$

Let

$$
\begin{equation*}
\underline{y}=\bar{U} \cdot \underline{u} \tag{3.4.20}
\end{equation*}
$$

then the problem is to solve

$$
\begin{equation*}
\overline{\mathrm{L}} \cdot \underline{y}=\underline{s} \tag{3.4.21}
\end{equation*}
$$

i.e.,

$$
\begin{align*}
& y_{1}=s_{1},  \tag{3.4.22}\\
& y_{i}=\left(s_{i}-\beta_{i-1} y_{i-1}\right) / \omega_{i}, \text { for } i=2,3, \ldots, m-1,  \tag{3.4.23}\\
& y_{i}=\left(s_{i}-\beta_{i-1} y_{i-1}-\sum_{k=i-m+1}^{\left.\gamma_{k-i+m, i-m+1} y_{k}\right) / \omega_{i},}\right. \\
& \quad \text { for } i=m, m+1, \ldots \ldots, p-1, \tag{3.4.24}
\end{align*}
$$

and

$$
y_{i}=\left(s_{i}-\beta_{i-1} y_{i-1}^{-} \sum_{k=i-m+1}^{i-1} \gamma_{k-i+m, i-m+1} y_{k}^{-} \sum_{k=i-p+1}^{i-1} f_{k-i+p, i-p+1} y_{k}\right) / \omega_{i},
$$

$$
\begin{equation*}
\text { for } i=p, p+1, \ldots, n \tag{3.4.25}
\end{equation*}
$$

followed by a back substitution process given by
i.e.,

$$
\begin{align*}
& \overline{\mathrm{U}} \underline{\mathrm{u}}=\underline{y} \\
& \mathrm{u}_{\mathrm{n}}=\mathrm{y}_{\mathrm{n}} \tag{3.4.26}
\end{align*}
$$

and

$$
\begin{equation*}
u_{i}=y_{i}-g_{i} y_{i+1}-\sum_{k=\tilde{p}}^{\tilde{q}} h_{i-k+m, k-m+1} y_{k}-\sum_{k=\bar{p}}^{\bar{q}} t_{i-k+p, k-p+1} y_{k} \tag{3.4.27}
\end{equation*}
$$

where the quantities $\tilde{p}, \tilde{q}, \bar{p}, \bar{q}$ are given by the following
'shorthand' representation.

If $m>[n / 2+1]$
if $\underline{p}>[n / 2+1]$ then
else

$$
\begin{array}{lll}
\tilde{p}=i+1, \quad \tilde{q}=n & \text { for } & i=n-1, n-2, \ldots, m^{+} \\
\tilde{p}=m, \quad \tilde{q}=n & \text { for } & i=m-1, m-2, \ldots, n-m+1 \\
\tilde{p}=m, \quad \tilde{q}=i+m-1 & \text { for } & i=n-m, n-m-1, \ldots, 1^{+}
\end{array}
$$

Note:
(+) denotes: "except if m=n"
(*) denotes: "except if $\mathrm{p}=\mathrm{n}$ "
If $m<[n / 2+1]$
if $\mathrm{p}>[\mathrm{n} / 2+1]$

$$
\text { if } n \text { is even and } m=n / 2 \text {, then }
$$

else


If $\mathrm{p}<[\mathrm{n} / 2+1]$
if $n$ is even and $p=n / 2$, then


if $p>n$ or $p \leq m$
if $n$ is even and $m=n / 2$, then

$$
\begin{array}{lll}
\tilde{p}=i+1, & \tilde{q}=n & \text { for } i=n-1, n-2, \ldots, m+1 \\
\tilde{p}=m+1, & \tilde{q}=n-1 & \text { for } i=m \\
\tilde{p}=m, & \tilde{q}=i+m-1 & \text { for } i=m-1, m-2, \ldots, 1 . \\
\frac{e 1 s e}{\widetilde{p}=i+1,} & \tilde{q}=n & \text { for } i=n-1, n-2, \ldots, n-m+1 \\
\tilde{p}=i+1, & \tilde{q}=i+m-1 & \text { for } i=n-m, n-m-1, \ldots, m \\
\tilde{p}=m, & \tilde{q}=i+m-1 & \text { for } i=m-1, m-2, \ldots, 1 .
\end{array}
$$

The total memory requirement for the LUBOT-3D algorithm is $\simeq(2 m+2 p+4) N$ words.

The amount of work involved in the factorization stage is $\simeq\left(p^{2}+p+3\right) N$ operations and given that the forward-backward substitution process requires $\simeq(2 m+2 p-1) N$ operations, the total operations for this algorithm are $\simeq\left(p^{2}+3 p+2 m+2\right) N$, for $m<p<N / 2$.

### 3.5 THE NORMALIZED ALGORITHM FOR THE SOLUTION OF LARGE, SYMMETRIC, SEVEN DIAGONAL, SPARSE LINEAR SYSTEMS (THE NB3D ALGORITHM)

In this section we introduce a sparse normalized factorization method for the solution of the seven-diagonal, symmetric, linear systems of bandwidth $m$ and $p$ (NB3D Algorithm), resulting from the application of the finite difference method to the solution of self adjoint p.d.e's in three space dimensions (see section 2.1 ).

We consider, as in section 3.3, the unique factorization:

$$
\begin{equation*}
\mathrm{A}=\overline{\mathrm{D}} \overline{\mathrm{~T}} \cdot \overline{\mathrm{~T}} \overline{\mathrm{D}} \tag{3.5.1}
\end{equation*}
$$

where $A$ is a large symmetric, positive definite, seven-diagonal sparse matrix, $\overline{\mathrm{D}}$ is a unique positive diagonal matrix, $\overline{\mathrm{T}}$ is a unique real upper triangular matrix with unit diagonal elements and $\overline{\mathrm{T}}$ ' denotes the transpose of $\overline{\mathrm{T}}$. Then, the coefficient matrix A of the system (2.1.3) can be written in the following partitioned form:

while $\overline{\mathrm{D}}, \overline{\mathrm{T}}$ in (3.5.1) are of the following form:


(3.5.4)

11

and on equating (3.5.5) with the partitioned form of $A$ in (3.5.2) we have the identities:

$$
\begin{align*}
& \tilde{A}=\tilde{D} \tilde{T} \cdot \tilde{T} \tilde{D} ; \quad \tilde{b}=\tilde{D} \tilde{T} \cdot r \tilde{d}  \tag{3.5.6}\\
& \tilde{a}=\tilde{d} r^{\prime} r \tilde{d}+\tilde{d} u u^{\prime} u \tilde{d} \tag{3.5.7}
\end{align*}
$$

The elements of $\tilde{D}, \tilde{T}$ are known and may be obtained by applying the normalized algorithm for the solution of symmetric, general five term linear systems of order $(p-1)$ and semibandwidth $m$ (see section 3.3 ).

In order to define the elements of the submatrices $r, u, \tilde{d}$ from the identities (3.5.6), (3.5.7), a test problem, with a simplified structure of the coefficient matrix $A$, has been worked out in Appendix 1.

The equations to determine the elements of $\overline{\mathrm{T}}$ matrix prove to be non-linear and a simple iterative Picard-type scheme was used in an inner loop to determine the values of $d_{p}, d_{p+1}, \ldots, d_{n}$ in an easy manner as the direct solution of these equations proved to be intractable. The elements of the submatrices $r, u, \tilde{d}$ may be obtained as follows:
for $j=1,2, \ldots, n-p+1$

$$
\begin{array}{ll}
d_{p+j-1}=d_{p+j-2} & \text { (Initial guess of } d_{p+j-1} \\
& \text { is taken as the adjacent } \\
& \text { value) } \tag{3.5.8}
\end{array}
$$

and

$$
\begin{equation*}
r_{1, j}=h_{j} / d_{j} d_{p+j-1} ; \quad e_{p+j-2}=b_{p+j-2} / d_{p+j-2} d_{p+j-1} \tag{3.5.9}
\end{equation*}
$$

whilst for $j \leqslant m-2$

$$
\begin{equation*}
r_{i, j}=-e_{i+j-2} r_{i-1, j}, \quad \text { for } i=2,3, \ldots, m-j \tag{3.5.10}
\end{equation*}
$$

then, if $j=1$ and $i=p-m+1$

$$
\begin{equation*}
r_{i, j} \leftarrow r_{i, j}+c_{i+j-1} / d_{m+j-2} d_{p+j-1} \tag{3.5.11}
\end{equation*}
$$

If $j>p-2$ or $i \geqslant p-j+1$, then

$$
\begin{equation*}
r_{i, j}=-e_{i+j-2} r_{i-1, j}-\sum_{k=1}^{i-1} r_{k-i+p, i+j-p} r_{k, j} \tag{3.5.12}
\end{equation*}
$$

then, if $i=p-m+1$

$$
\begin{equation*}
r_{i, j} \leftarrow r_{i, j}+c_{i+j-1} / d_{i+j-1} d_{p+j-1} \tag{3.5.13}
\end{equation*}
$$

else
if i<m, then

$$
\begin{equation*}
r_{i, j}=-e_{i+j-2} r_{i-1, j}-\sum_{k=1}^{i-1} r_{k, j} t_{k-i+m, i+j-m} \tag{3.5.14}
\end{equation*}
$$

else

$$
r_{i, j}=-e_{i+j-2} r_{i-1, j}-\sum_{k=1}^{m-1} r_{k+i-m, j} t_{k, i+j-m}
$$

if $i=p-m+1$, then

$$
\begin{equation*}
r_{i, j} \leftarrow r_{i, j}+c_{i+j-1} / d_{i+j-1} d_{p+j-1} \tag{3.5.16}
\end{equation*}
$$

for either $i=2,3, \ldots, p-1$ and all $j>m-1$
or

$$
i=(m-j+1),(m-j+2), \ldots, p-1 \text { for } j \leqslant m-1 .
$$

Then, for $i=p-1$

$$
\begin{equation*}
d_{p+j-1}=a_{p+j-1}^{\frac{1}{2}} /\left[1+\sum_{k=1}^{p-2} r_{k, j}^{2}+\left(e_{p+j-2}+r_{i, j}\right)^{2}\right]^{\frac{1}{2}} \tag{3.5.17}
\end{equation*}
$$

The linear system (2.1.3) can be written as:

$$
\begin{equation*}
\left(\overline{\mathrm{D}} \overline{\mathrm{~T}}^{\prime} \overline{\mathrm{T}} \overline{\mathrm{D}}\right) \underline{\mathrm{u}}=\underline{s} \tag{3.5.18}
\end{equation*}
$$

from which we obtain

$$
\begin{array}{ll} 
& (\overline{\mathrm{T}} ' \overline{\mathrm{~T}}) \underline{\bar{D}} \underline{u}=\overline{\mathrm{D}}^{-1} \underline{s} \\
\text { Let } & \underline{y}=\overline{\mathrm{D}} \underline{u} \quad \text { and } \quad \underline{g}=\overline{\mathrm{D}}^{-1} \underline{s} \tag{3.5.20}
\end{array}
$$

then the problem is reduced to solving the system

$$
\begin{equation*}
(\overline{\mathrm{T}}, \overline{\mathrm{~T}}) \underline{y}=\underline{g} \tag{3.5.21}
\end{equation*}
$$

This can be solved directly for $\underline{y}$ in terms of the auxiliary vector $h^{*}$, where

$$
\begin{equation*}
\overline{\mathrm{T}} \underline{y}=\underline{h}^{*} \quad \text { and } \quad \overline{\mathrm{T}}^{\prime} \underline{h}^{*}=\underline{\mathrm{g}} \tag{3.5.22}
\end{equation*}
$$

i.e.

$$
\begin{align*}
& h_{1}^{*}=g_{1} ;  \tag{3.5.23}\\
& h_{\dot{i}}^{*}=g_{i}-e_{i-1} h_{\dot{i}-1}^{*}, \quad i=2,3, \ldots, m-1 ;  \tag{3.5.24}\\
& h_{\dot{i}}^{*}=g_{i}-e_{i-1} h_{i-1}^{*} \sum_{\substack{k=i-m+1 \\
i-1}}^{t_{k-i}+m, i-m+1} h_{k}^{*}, i=m, m+1, \ldots, p-1  \tag{3.5.25}\\
& i-1
\end{align*}
$$

and

$$
h_{i}^{*}=g_{i}-e_{i-1} h_{i-1}^{*}-\sum_{k=i-m+1}^{i-1} t_{k-i+m, i-m+1} h_{k}^{*}-\sum_{k=i-p+1}^{i=1} r_{k-i+p, i-p+1} h_{k}^{*},
$$

A back substitution process yields $\underline{y}$ in terms of $\underline{h}^{*}$, the components being given by the equations

$$
\begin{equation*}
y_{n}=h_{n}^{*} \tag{3.5.27}
\end{equation*}
$$

and

$$
\begin{equation*}
y_{i}=h_{i}^{*}-e_{i} y_{i+1}-\sum_{j=\tilde{p}}^{\tilde{q}} t_{i-j+m, j-m+1} y_{j}-\sum_{j=\bar{p}}^{\vec{q}} r_{i-j+p, j-p+1} y_{j} \tag{3.5.28}
\end{equation*}
$$

where the values of $\tilde{p}, \tilde{q}, \bar{p}, \bar{q}$ are given in section 3.4 .
The final solution $\underline{u}$ is obtained from $\underline{u}=\bar{D}^{-1} \underline{y}$ an operation involving only one division per component.

The total memory requirement of the NB3D algorithm is $\simeq(m+p+3) N$ words. The amount of work involved for the factorization is $\approx\left(\frac{(p-1)(p-2)}{2}+2 p+4\right) N$ mults $+N$ square roots. Given that the normalization and the forward-backward substitution processes require $2 N$ divisions and $\simeq(2 m+2 p-2) N$ mults respectively, the total number of operations for this algorithm is: $\simeq\left(\frac{(p-1)(p-2)}{2}+4 p+2 m+4\right) N$ mults $+N$ square roots.

Note that if $h_{i}=0, i \varepsilon[1, n-p+1]$ or $c_{i}=0, i \varepsilon[1, n-m+1]$ the algorithm reduces to the normalized form of the quindiagonal system of bandwidth $m$ or $p$ respectively, [6], which is encountered usually in solving five point boundary value problems. Furthermore, if $h_{i}=0, i \varepsilon[1, n-p+1]$ and $c_{i}=0, i \varepsilon[1, n-m+1]$ the algorithm reduces to the normalized form of the common tridiagonal system [14], which is encountered in solving two point boundary value problems.

$$
\text { Chapter } 4
$$

APPROXIMATE ALGORITHMS FOR TWO AND THREE
DIMENSIONAL P.D.E'S

### 4.1 INTRODUCTION

For the solution of the class of problems discussed in Chapter 2, it can be easily seen that the storage requirements and computational work for the basic algorithms, as introduced in Chapter 3, is prohibitively high for computers with relatively limited core memory. In this Chapter we introduce approximate algorithmic solution methods in which the large, sparse matrix derived from the finite difference discretizations of parabclic and elliptic p.d.e's, in both two and three space dimensions, is approximately factorized to yield algorithmic procedures for use in iterative schemes for finite difference methods.

These procedures can be considered to be approximate counterparts of the algorithmic procedures given in Chapter 3.

The idea of the approximate factorization method was first proposed by Buleev [9] and Oliphant [47] and is based on the simple replacement of the coefficient matrix $A$ by a matrix $(A+B)$ such that

$$
A+B=L_{s} U_{s}
$$

where $L_{s}, U_{s}$ are sparse strictly lower triangular and upper triangular matrices. Obviously there is a large number of such matrices $B$, where the matrix $(A+B)$ can be factored in sparse triangular matrices.

In the following, we shall attempt to outline a strategy whereby $L_{S}$ and $U_{S}$ are easily determined.

### 4.2 THE APPROXIMATE ALGORITHM FOR THE SOLUTION OF LARGE, UNSYMMETRIC, QUINDIAGONAL, SPARSE LINEAR SYSTEMS (THE ALUBOT-2D ALGORITHM)

In this section, we present an approximate triangular factorization of the coefficient matrix of system (3.2.15), resulting from the application of finite difference methods to the solution of parabolic p.d.e's in two dimensions (see section 2.5).

Let A be defined as in section 3.2 , then we consider the approximate factorization

$$
\begin{equation*}
\mathrm{A} \approx \mathrm{~L}_{\mathrm{S}} \cdot \mathrm{U}_{\mathrm{S}} \tag{4.2.1}
\end{equation*}
$$

where the triangular matrices $\mathrm{L}_{\mathrm{S}}, \mathrm{U}_{\mathrm{S}}$ (sparse forms of $\mathrm{L}, \mathrm{U}$ given by (3.2.3), (3.2.4) respectively) are of the following form:



The approximate algorithm (henceforth called the ALUBOT Algorithm) which retains the $r$ outermost off-diagonal entries, can be given in a similar compact form as the LUBOT algorithm and is expressed as follows:

$$
\begin{equation*}
\tilde{\omega}_{1}=\mathrm{b}_{1} . ; \quad \tilde{\beta}_{1}=\mathrm{a}_{2} ; \quad \tilde{\mathrm{g}}_{1}=\mathrm{c}_{1} / \tilde{\omega}_{1} \text {, } \tag{4.2.3}
\end{equation*}
$$

for $i=2,3, \ldots, m-2$

$$
\begin{equation*}
\tilde{\omega}_{i}=b_{i}-\widetilde{\beta}_{i-1} \tilde{g}_{i-1} ; \quad \tilde{\beta}_{i}=a_{i+1} ; \quad \tilde{g}_{i}=c_{i} / \tilde{\omega}_{i} \tag{4.2.4}
\end{equation*}
$$

and

$$
\begin{equation*}
\tilde{\omega}_{m-1}=b_{m-1}-\tilde{\beta}_{m-2} \widetilde{g}_{m-2} \tag{4.2.5}
\end{equation*}
$$

For $j=1,2, \ldots, n-m+1$, we have,

$$
\begin{equation*}
\tilde{h}_{1, j}=\tau_{j} / \tilde{\omega}_{j} ; \quad \tilde{\gamma}_{1, j}=v_{j+m-1} ; \tag{4.2.6}
\end{equation*}
$$

and $\quad \tilde{g}_{m+j-2}=c_{m+j-2} / \tilde{\omega}_{m+j-2} ; \quad \tilde{\beta}_{m+j-2}=a_{m+j-1}$,
whilst for $\mathrm{j} \leqslant \mathrm{r}-1$

$$
\begin{equation*}
\tilde{h}_{i, j}=-\tilde{\beta}_{i+j-2} \tilde{h}_{i-1, j} / \tilde{\omega}_{i+j-1}, \quad \text { for } \quad i=2,3, \ldots, r-j+1 \tag{4.2.8}
\end{equation*}
$$

and

$$
\begin{equation*}
\tilde{\gamma}_{i, j}=-\tilde{g}_{i+j-2} \tilde{\gamma}_{i-1, j} \quad, \quad \text { for } i=2,3, \ldots, r-j+1 \tag{4.2.9}
\end{equation*}
$$

Then, for $\mathrm{j}>1$ and $\mathrm{r}>1$

$$
\begin{equation*}
\tilde{\gamma}_{i, j}=-\tilde{g}_{i+j-2} \tilde{\gamma}_{i-1, j}-\sum_{k=1}^{i-1} \tilde{\gamma}_{k, j} \tilde{h}_{k-i+r+1, i+j-r-1} \tag{4.2.10}
\end{equation*}
$$

and

$$
\begin{equation*}
\tilde{h}_{i, j}=\left(-\tilde{\beta}_{i+j-2} \tilde{h}_{i-1, j}-\sum_{k=1}^{i-1} \tilde{\gamma}_{k-i+r+1, i+j-r-1} \tilde{h}_{k, j}\right) / \tilde{\omega}_{i+j-1} \tag{4.2.11}
\end{equation*}
$$

for either $i=(r-j+2),(r-j+3), \ldots, r$ and all $j \leqslant r$
or $\quad i=2,3, \ldots, r$ for $j>r$.
Then, for $i=r$

$$
\begin{equation*}
\tilde{\omega}_{m+j-1}=b_{m+j-1}-\tilde{\beta}_{i+j-1} \tilde{h}_{i, j}-\tilde{g}_{i+j-1} \tilde{\gamma}_{i, j}-\tilde{\beta}_{i+j-1} \tilde{g}_{i+j-1}-\sum_{k=1}^{i} \tilde{\gamma}_{k, j} \tilde{h}_{k, j} \tag{4.2.12}
\end{equation*}
$$

An approximate solution of linear system (3.2.15) is now given in a similar manner to (3.2.16)-(3.2.22). The forward substitution process is expressed as

$$
\begin{equation*}
\tilde{y}_{1}=s_{1} / \tilde{\omega}_{1} \quad ; \quad \tilde{y}_{i}=\left(s_{i}-\tilde{\beta}_{i-1} \tilde{y}_{i-1}\right) / \tilde{\omega}_{i}, \quad i=2,3, \ldots, m-1 \tag{4.2.13}
\end{equation*}
$$

and

$$
\begin{equation*}
\tilde{y}_{i}=\left(s_{i}-\tilde{\beta}_{i-1} \tilde{y}_{i-1}-\sum_{k=i-m+1}^{i-m+r} \tilde{\gamma}_{k-i+m, i-m+1} \tilde{y}_{k}\right) / \tilde{\omega}_{i}, i=m, m+1, \ldots, n \tag{4.2.14}
\end{equation*}
$$

whilst the backward substitution process yields the solution $\underline{u}$ in terms of $\underline{\tilde{y}}$ as

$$
\begin{equation*}
u_{n}=\tilde{y}_{n} \tag{4.2.15}
\end{equation*}
$$

and $\quad u_{i}=\tilde{y}_{i}-\tilde{g}_{i} u_{i+1}-\sum_{j=p^{*}}^{q^{*}} \tilde{h}_{i-j+m, j-m+1} \tilde{y}_{j}$,
where the quantities $\mathrm{p}^{*}, \mathrm{q}^{*}$ can be easily obtained from those given in section 3.2.

The total memory requirement for the ALUBOT algorithm is $\simeq(2 r+6) N$ words.
The amount of work involved for the factorization is $\simeq\left(r^{2}+3 r+2\right) N$ operations.

Given that the forward-backward substitution process requires $\simeq(2 \mathrm{r}+3) \mathrm{N}$ operations, the total number of operations for this algorithm is $\simeq\left(r^{2}+5 r+5\right) N$.
4.3 THE APPROXIMATE NORMALIZED ALGORITHM FOR THE SOLUTION OF LARGL;

An approximate triangular factorization of the coefficient matrix of system (3.2.15), resulting from the application of the finite difference method to the solution of self adjoint p.d.e's in two space dimensions (see section 2.1 ) is now introduced.

Let $A$ be defined as in section 3.3 and we consider the approximate factorization

$$
\begin{equation*}
A \approx D_{s} T_{s}^{\prime} T_{s} D_{s} \tag{4.3.1}
\end{equation*}
$$

where $\mathrm{D}_{\mathrm{S}}, \mathrm{T}_{\mathrm{s}}$ are the sparse forms of $\mathrm{D}, \mathrm{T}$ matrices in section 3.3 and $T_{s}^{\prime}$ denotes the transpose of $T_{S}$.

With the coefficient matrix $A$ as in (3.3.2) $D_{S}, T_{S}$ have the
following form:



The approximate normalized algorithm (henceforth called NOBAR) obtained from retaining r-outermost off-diagonal entries can be expressed in the following compact form:

$$
\begin{align*}
& \tilde{d}_{1}=\sqrt{a} 1_{1} ; \quad \tilde{d}_{i}=\left\{a_{i}-\left(\frac{b_{i-1}}{\tilde{d}_{i-1}}\right)^{2}\right\}^{\frac{1}{2}} ; \quad \tilde{e}_{i-1}=\frac{b_{i-1}}{\tilde{d}_{i-1} \tilde{d}_{i}} \\
& \text { for } i=2,3, \ldots, m-1 . \tag{4.3.4}
\end{align*}
$$

Then for $j=1,2,3, \ldots, n-m+1$, we have

$$
\begin{equation*}
\tilde{x}_{1}=\frac{c_{j}}{\tilde{d}_{j}} ; \quad \tilde{v}=\frac{b_{m+j-2}}{\tilde{d}_{m+j-2}} \tag{4.3.5}
\end{equation*}
$$

and for $\mathrm{j} \leqslant \mathrm{r}-1$

$$
\begin{equation*}
\tilde{x}_{i}=-\tilde{e}_{i+j-2} \tilde{x}_{i-1}, \quad \text { for } i=2,3, \ldots, r+1-j . \tag{4.3.6}
\end{equation*}
$$

Then, for $\mathrm{j}>1$ and $\mathrm{r}>1$

$$
\begin{equation*}
\tilde{x}_{i}=-\tilde{e}_{i+j-2} \tilde{x}_{i-1}-\sum_{k=1}^{i-1} \tilde{x}_{k} \tilde{t}_{k-i+r+1, i+j-r-1} \tag{4.3.7}
\end{equation*}
$$

and for either $i=(r-j+2),(r-j+3), \ldots, r$ and all $j \leqslant r$

$$
\text { or } \quad i=2,3, \ldots, r \text { for } j>r \text {. }
$$

Then, we have

$$
\begin{align*}
& \tilde{\mathrm{d}}_{\mathrm{m}+\mathrm{j}-1}=\left\{\mathrm{a}_{\mathrm{m}+\mathrm{j}-1}-\sum_{\mathrm{k}=1}^{\mathrm{r}} \tilde{\mathrm{x}}_{\mathrm{k}}^{2}-\tilde{v}^{2}\right\}^{\frac{1}{2}}  \tag{4.3.8}\\
& \widetilde{\mathrm{e}}_{\mathrm{m}+\mathrm{j}-2}=\tilde{\mathrm{v}} / \tilde{\mathrm{d}}_{\mathrm{m}+\mathrm{j}-1} \tag{4.3.9}
\end{align*}
$$

and,

$$
\begin{equation*}
\tilde{t}_{i, j}=\tilde{x}_{i} / \tilde{d}_{m+j-1}, \quad \text { for } i=1,2,3 \ldots, r \tag{4.3.10}
\end{equation*}
$$

An approximate solution of the linear system (3.2.15) can then
be obtained by writing

$$
\begin{equation*}
D_{s} T_{s}^{\prime} T_{s} D_{s} \cdot \underline{u}=\underline{s}, \tag{4.3.11}
\end{equation*}
$$

or equivalently

$$
\begin{equation*}
\left(T_{s}^{\prime} T_{s}\right) D_{s} \cdot \underline{u}=D_{s}^{-1} \cdot \underline{s} \tag{4.3.12}
\end{equation*}
$$

By introducing the auxiliary vectors $\underline{y}$ and $\underline{g}$ where,

$$
\begin{equation*}
\underline{y}=D_{s} \cdot \underline{u} \quad ; \quad \underline{g}=D_{s}^{-1} \cdot \underline{s} \tag{4.3.13}
\end{equation*}
$$

the problem is reduced to solving the normalized system

$$
\begin{equation*}
\mathrm{T}_{\mathrm{s}}^{\prime} \cdot \mathrm{T}_{\mathrm{s}} \cdot \underline{y}=\underline{\mathrm{g}} \tag{4.3.14}
\end{equation*}
$$

and $\underline{y}$ is obtained in terms of an auxiliary vector $\underline{h}$, where

$$
\begin{equation*}
\mathrm{T}_{\mathrm{s}} \cdot \underline{y}=\underline{\mathrm{h}} \quad \text { and } \quad \mathrm{T}_{\mathrm{s}}^{\prime} \cdot \underline{h}=\underline{\mathrm{g}} \tag{4.3.15}
\end{equation*}
$$

i.e.

$$
\begin{align*}
& h_{1}=g_{1} ;  \tag{4.3.16}\\
& h_{i}=g_{i}-\tilde{e}_{i-1} h_{i-1}, \quad i=2,3, \ldots, m-1 ; \tag{4.3.17}
\end{align*}
$$

and

$$
\begin{equation*}
h_{i}=g_{i}-\tilde{e}_{i-1} h_{i-1}-\sum_{k=i-m+1}^{i-m+r} \tilde{t}_{k-i+m, i-m+1} h_{k}, i=m, m+1, \ldots, n . \tag{4.3.18}
\end{equation*}
$$

A back substitution process yields $\underline{y}$ in terms of $\underline{h}$, the components being given by the equations

$$
\begin{equation*}
y_{n}=h_{n} \tag{4.3.19}
\end{equation*}
$$

and

$$
\begin{equation*}
y_{i}=h_{i}-\tilde{e}_{i} y_{i+1}-\sum_{j=p}^{q} \tilde{t}_{i-j+m, j-m+1} y_{j} \tag{4.3.20}
\end{equation*}
$$

where $p, q$ are easily obtained from those given in section 3.2. The final solution $\underline{u}$ is obtained from (4.3.13) as $\underline{u}=D_{s}^{-1} \underline{y}$, an operation involving only one division per vector component.

The total memory requirements for this algorithm is $\simeq(r+4) N$ words.

The amount of work involved for the factorization process is given by $\simeq\left(\frac{r(r-1)}{2}+3 r+3\right) N$ mults $+N$ square roots.

The normalization and the forward-backward substitution processes require 2 N divisions and $\simeq(2 \mathrm{r}+2) \mathrm{N}$ multiplications respectively. Therefore, the total number of operations for the NOBAR algorithm is

$$
\begin{equation*}
\simeq\left(\frac{r(r-1)}{2}+5 r+7\right) N \text { mults }+N \text { square roots } \tag{4.3.21}
\end{equation*}
$$

Remark
It can be easily seen from (4.3.3), (3.3.4) and (4.3.4), (3.3.5)
that

$$
\begin{array}{ll}
\tilde{e}_{i}=e_{i}, & \text { for } i=1,2, \ldots, m-2 \\
\tilde{d}_{i}=d_{i}, & \text { for } i=1,2, \ldots, m-1 . \tag{4.3.23}
\end{array}
$$

By calculating exactly (as in (3.3.9)) the $\tilde{d}_{i}, i \varepsilon[m, n]$ and $\tilde{\mathrm{e}}_{\mathrm{i}}, \mathrm{i} \varepsilon[\mathrm{m}-1, \mathrm{n}-1]$ from relations (4.3.8) and (4.3.9) respectively i.e., $\tilde{d}_{i}=d_{i}, i \varepsilon[m, n]$ and $\tilde{e}_{i}=e_{i}, i \varepsilon[m-1, n-1]$, the following approximate factorization is obtained:

$$
\begin{equation*}
A \approx T_{S}^{\prime} T_{s} D \tag{4.3.24}
\end{equation*}
$$

where $D$ is given by (3.3.3) and (3.3.5), (3.3.9)

$$
\begin{aligned}
& \mathrm{T}_{\mathrm{s}} \text { is the sparse version of }(3.3 .4)(\mathrm{r}- \\
& \text { outermost off diagonal terms are retained), }
\end{aligned}
$$

and

$$
T_{S}^{\prime} \text { is the transpose of } T_{S} .
$$

Although the approximate factorization (4.3.24) is preferable for theoretical purposes and analysis, in practice it turned out to be computationally more efficient to use the factorization of the form (4.3.1).
4.4 THE APPROXIMATE ALGORITHM FOR THE SOLUTION OF LARGE, UNSYMMETRIC SEVEN DIAGONAL, SPARSE LINEAR SYSTEMS (THE ALUBOT-3D ALGORITHM)

We now introduce an approximate triangular factorization of the coefficient matrix of system (2.8.14) resulting from the application of finite difference methods to the solution of parabolic p.d.e's in three space dimensions (see section 2.8).

Let $\Omega$ be defined as in section 3.4 and consider the approximate factorization

$$
\begin{equation*}
\Omega \approx \overline{\mathrm{L}}_{\mathrm{S}} \cdot \overline{\mathrm{U}}_{\mathrm{S}} \tag{4.4.1}
\end{equation*}
$$

where the triangular matrices $\bar{L}_{S}, \bar{U}_{S}$ (sparse forms of $\bar{L}, \bar{U}$ given by (3.4.3), (3.4.4) respectively) are of the following form:


(4.4.3)

It can be easily seen that the elements of $\tilde{L}_{11}, \widetilde{U}_{11}$ are similar in structure to the factors in ALUBOT-2D and are known. These may be obtained by applying the ALUBOT-2D algorithm for the solution of a general five term linear system of order $(p-1)$, semibandwidth m, where $r_{1}$ outermost off-diagonal entries are retained.

The elements of the submatrices $\tilde{U}_{12}, \tilde{L}_{21}, \tilde{L}_{22}, \tilde{U}_{22}$ (cf. (4.4.2), (4.4.3)) may be obtained algorithmically as follows:

For $j=1,2, \ldots, n-p+1$

$$
\begin{equation*}
t_{1, j}=r_{j} / \omega_{j} \quad ; \quad f_{1, j}=s_{p+j-1} \tag{4.4.4}
\end{equation*}
$$

and

$$
\begin{equation*}
g_{p+j-2}=c_{p+j-2} / \omega_{p+j-2} ; \beta_{p+j-2}=a_{p+j-1} \tag{4.4.5}
\end{equation*}
$$

whilst if $j \leqslant r_{2}-1$

$$
\begin{equation*}
t_{i, j}=-\beta_{i+j-2} t_{i-1, j} / \omega_{i+j-1} \text {, for } i=2,3, \ldots\left(r_{2}+1-j\right) \tag{4.4.6}
\end{equation*}
$$

and

$$
\begin{equation*}
f_{i, j}=-g_{i+j-2} f_{i-1, j}, \text { for } i=2,3, \ldots,\left(r_{2}+1-j\right) \tag{4.4.7}
\end{equation*}
$$

then, if $j=1$ and $i=p-m+1$

$$
\begin{equation*}
t_{i, j} \leftarrow t_{i, j}^{+\tau_{i+j-1} / \omega_{i+j-1}} ; f_{i, j} \leftarrow f_{i, j}+v_{p+j-1} \tag{4.4.8}
\end{equation*}
$$

If $j>p-2$ or $i>p-j+1$, then

$$
\begin{equation*}
t_{i, j}=\left(-\beta_{i+j-2} t_{i-1, j}^{-} \sum_{k=1}^{i-1} f_{k-i+r_{2}+1, i+j-r_{2}-1} t_{k, j}\right) / \omega_{i+j-1} \tag{4.4.9}
\end{equation*}
$$

and

$$
\begin{equation*}
f_{i, j}=-g_{i+j-2} f_{i-1, j}-\sum_{k=1}^{i-1} f_{k, j} t_{k-i+r_{2}+1, i+j-r_{2}-1} \tag{4.4.10}
\end{equation*}
$$

if $i=p-m+1$, then

$$
\begin{equation*}
t_{i, j} \leftarrow t_{i, j}^{+\tau_{i+j-1} / \omega_{i+j-1}} ; f_{i, j} \leftarrow f_{i, j}+v_{p+j-1} \tag{4.4.11}
\end{equation*}
$$

else
if $i<m$, then
and

$$
\begin{align*}
& t_{i, j}=\left(-\beta_{i+j-2} t_{i-1, j}-\sum_{k=1}^{i-1} \gamma_{k-i+r_{1}+1, i+j-r_{1}-1} t_{k, j}\right) / \omega_{i+j-1} \\
& f_{i, j}=-g_{i+j-2} f_{i-1, j}-\sum_{k=1}^{i-1} f_{k, j} h_{k-i+r_{1}+1, i+j-r_{1}-1} \tag{4.4.12}
\end{align*}
$$

else

$$
\begin{equation*}
t_{i, j}=\left(-\beta_{i+j-2^{t}} t_{i-1, j}-\sum_{k=1}^{m-1} \gamma_{k, i+j-m^{\prime}}{ }_{k+i-m, j}\right) / \omega_{i+j-1}, \tag{4.4.17}
\end{equation*}
$$

and $\quad f_{i, j}=-g_{i+j-2} f_{i-1, j}-\sum_{k=1}^{m-1} f_{k+i-m, j} h_{k, i+j-m}$
if $i=p-m+1$, then

$$
\begin{equation*}
t_{i, j} \leftarrow t_{i, j}^{+\tau_{i+j-1} / w_{i+j-1} ; \quad f_{i, j} \leftarrow f_{i, j}+v_{p+j-1}, ~} \tag{4.4.16}
\end{equation*}
$$

for either $i=2,3, \ldots, r_{2}$ and all $j>r_{1}$
or

$$
i=\left(r_{1}-j+2\right),\left(r_{1}-j+3\right), \ldots, r_{2} \text { for } j \leqslant r_{1}
$$

Then, for $i=r_{2}$

$$
\begin{align*}
& \omega_{p+j-1}=b_{p+j-1}-\beta_{i+j-1} t_{i, j}-\beta_{i+j-1} g_{i+j-1}-g_{i+j-1} f_{i, j}-\sum_{k=1}^{r_{2}} f_{k, j} t_{k, j} \\
& -\sum_{k=1}^{r_{1}} \gamma_{k, j+p-m^{h}}{ }_{k, j+p-m} . \tag{4.4.17}
\end{align*}
$$

An approximate solution of the linear system (2.8.14) can then be obtained by writing,

$$
\begin{align*}
& \overline{\mathrm{L}}_{\mathrm{s}} \overline{\mathrm{U}}_{\mathrm{s}} \underline{\mathrm{u}}=\underline{s}  \tag{4.4.18}\\
& \underline{y}=\overline{\mathrm{U}}_{s} \underline{\mathrm{u}} \tag{4.4.19}
\end{align*}
$$

and if we let
the problem is reduced to solving the triangular system

$$
\begin{equation*}
\overline{\mathrm{L}}_{\mathrm{s}} \underline{y}=\underline{s} \tag{4.4.20}
\end{equation*}
$$

i.e.,

$$
\begin{align*}
& y_{1}=s_{1} ; \\
& y_{i}=\left(s_{i}-\beta_{i-1} y_{i-1}\right) / \omega_{i}, \quad \text { for } i=2,3, \ldots, m-1 \text {; }  \tag{4.4.22}\\
& i-m+r_{1} \\
& y_{i}=\left(s_{i}-\beta_{i-1} y_{i-1}^{-} \sum_{k=i-m+1} \gamma_{k-i+m, i-m+1} y_{k}\right) / \omega_{i} \text {, } \\
& \text { for } i=m, m+1, \ldots, p-1 \text {. }  \tag{4.4.23}\\
& i-m+r_{1} \quad i-p+r_{2} \\
& \text { and } \quad y_{i}=\left(s_{i}-\beta_{i-1} y_{i-1}^{-} \sum_{k=i-m+1} \gamma_{k-i+m, i-m+1} y_{k}-\sum_{k=i-p+1} f_{k-i+p, i-p+1} y_{k}\right) / \omega_{i} \text {, } \\
& \mathrm{i}=\mathrm{p}, \mathrm{p}+1, \ldots, \mathrm{n} \text {. }
\end{align*}
$$

The final solution is obtained from a back substitution
process given by

$$
\begin{equation*}
u_{n}=y_{n} \tag{4.4.25}
\end{equation*}
$$

and

$$
\begin{equation*}
u_{i}=y_{i}-g_{i} y_{i+1}-\sum_{k=\tilde{p}}^{\tilde{q}} h_{i-k+m, k-m+1} y_{k}-\sum_{k=\bar{p}}^{\bar{q}} t_{i-k+p, k-p+1} y_{k} . \tag{4.4.26}
\end{equation*}
$$

The quantities $\tilde{p}, \widetilde{q}, \bar{p}, \bar{q}$ can be easily obtained from those given in section 3.4 .

The above approximate algorithmic procedure (henceforth called the ALUBOT-3D algorithm) in which $r_{1}, r_{2}$ outermost off-diagonal entries are retained, has been expressed in a compact form, similar to those given in sections 3.4 and 3.5 respectively.

The total memory requirement for the ALUBOT-3D algorithm is $\simeq\left(2 r_{1}+2 r_{2}+8\right) N$ words.

The amount of work involved for the factorization process is
$\simeq\left(\mathrm{r}_{1}^{2}+\mathrm{r}_{2}^{2}+4 \mathrm{r}_{1}+4 \mathrm{r}_{2}+3\right) \mathrm{N}$ mults,
and the forward-backward substitution process requires $\simeq\left(2 \mathrm{r}_{1}+2 \mathrm{r}_{2}+3\right) \mathrm{N}$ mults. Therefore, the total operations for this algorithm are $\simeq\left(r_{1}^{2}+r_{2}^{2}+6 r_{1}+6 r_{2}+6\right) N$ mults.
4.5 THE APPROXIMATE NORMALIZED ALGORITHM FOR THE SOLUTION OF LARGE, SYMMETRIC, SEVEN DIAGONAL, SPARSE LINEAR SYSTEMS (THE NOBAR-3D ALGORITHM)

Finally, an approximate triangular factorization of the coefficient matrix of system (2.1.3), resulting from the application of finite difference method to the solution of self adjoint p.d.e's in three space dimensions (see section 2.1) is presented.

We consider the approximate factorization

$$
\begin{equation*}
A \approx \bar{D}_{s} \bar{T}_{s}^{\prime} \bar{T}_{s} \bar{D}_{s} \tag{4.5.1}
\end{equation*}
$$

where the coefficient matrix $A$ of system (2.1.3) is given by (3.5.2) and the matrices $\bar{D}_{S}, \vec{T}_{S}$ (sparse forms of $D, T$ given by (3.5.3), (3.5.4) respectively), are of the following form:


$\bar{T}_{S}^{\prime}$ denotes the transpose of $\bar{T}_{S}$.

Then the approximate normalized algorithmic procedure (henceforth called the NOBAR-3D algorithm), retaining $r_{1}$ and $r_{2}$ outermost offdiagonal entries, is developed as follows:-
the elements of submatrices $\delta_{1}$ and $\bar{\tau}$ (cf. (4.5.2), (4.5.3))
can be obtained by applying the NOBAR-2D algorithm for the solution of a symmetric five term linear system of order ( $p-1$ ), semibandwidth $m$, retaining $r_{1}$ outermost off-diagonal entries. The elements of submatrices $\overline{\mathrm{r}}, \overline{\mathrm{u}}, \delta_{2}$, may be obtained algorithmically as follows:
for $j=1,2, \ldots, n-p+1$

$$
\begin{align*}
& \bar{d}_{p+j-1}=\bar{d}_{p+j-2} ; \bar{r}_{i, j}=h_{i} / \bar{d}_{j} \bar{d}_{p+j-1} ;  \tag{4.5.4}\\
& \bar{v}=b_{p+j-2} / \bar{d}_{p+j-2} \bar{d}_{p+j-1}, \tag{4.5.5}
\end{align*}
$$

whilst for $j \leqslant r_{1}-1$

$$
\begin{equation*}
\bar{r}_{i, j}=-\bar{e}_{i+j-2} \overline{\mathrm{r}}_{i-1, j}, \quad \text { for } i=2,3, \ldots,\left(r_{1}+1-j\right) \tag{4.5.6}
\end{equation*}
$$

then, if $j=1$ and $i=p-m+1$

$$
\begin{equation*}
\overline{\mathrm{r}}_{i, j} \leftarrow \overline{\mathrm{r}}_{i, j}+c_{i+j-1} / \overline{\mathrm{d}}_{m+j-2} \overline{\mathrm{~d}}_{\mathrm{p}+\mathrm{j}-1} . \tag{4.5.7}
\end{equation*}
$$

If $i \geqslant p-j+1$ or $j>p-2$

$$
\begin{equation*}
\bar{r}_{i, j}=-\bar{e}_{i+j-2} \bar{r}_{i-1, j}-\sum_{k=1}^{i-1} \bar{r}_{k-i+r_{2}+1, i+j-r_{2}-1} \bar{r}_{k, j} \tag{4.5.8}
\end{equation*}
$$

if $i=p-m+1$, then

$$
\begin{equation*}
\bar{r}_{i, j} \leftarrow \bar{r}_{i, j}+c c_{i+j-1} / \bar{d}_{i+j-1} \bar{d}_{p+j-1} \tag{4.5.9}
\end{equation*}
$$

else
if $i<m$, then

$$
\begin{equation*}
\bar{r}_{i, j}=-\bar{e}_{i+j-2} \bar{r}_{i-1, j}-\sum_{k=1}^{i-1} \bar{t}_{k-i+r_{1}+1, i+j-r_{1}-1} \bar{r}_{k, j} \tag{4.5.10}
\end{equation*}
$$

else

$$
\begin{equation*}
\bar{r}_{i, j}=-\bar{e}_{i+j-2} \bar{r}_{i-1, j}-\sum_{k=1}^{m-1} \bar{t}_{k, i+j-m} \bar{r}_{k+i-m, j} \tag{4.5.11}
\end{equation*}
$$

if $i=p-m+1$, then

$$
\begin{equation*}
\bar{r}_{i, j} \leqslant \bar{r}_{i, j}+c{ }_{i+j-1} / \bar{d}_{i+j-1} \bar{d}_{p+j-1}, \tag{4.5.12}
\end{equation*}
$$

for either $i=2,3, \ldots, r_{2}$ and all $j>r_{1}$
or

Then,

$$
\begin{equation*}
\overline{\mathrm{d}}_{\mathrm{p}+\mathrm{j}-1}=\left\{a_{\mathrm{p}+\mathrm{j}-1} /\left[1+\sum_{\mathrm{k}=1}^{\mathrm{r}_{1}} \overline{\mathrm{t}}_{\mathrm{k}, \mathrm{j}+\mathrm{p}-\mathrm{m}}^{2} \sum_{\mathrm{k}=1}^{\mathrm{r}_{2}^{2}} \overline{\mathrm{r}}_{\mathrm{k}, \mathrm{j}}^{2}+\overline{\mathrm{v}}^{2}\right]\right\}^{\frac{1}{2}} \tag{4.5.13}
\end{equation*}
$$

and

$$
\begin{equation*}
\bar{e}_{p+j-2}=b_{p+j-2} / \bar{d}_{p+j-2} \bar{d}_{p+j-1} \tag{4.5.14}
\end{equation*}
$$

An approximate solution of the linear system (2.1.3) can now be obtained by writing,

$$
\begin{equation*}
\overline{\mathrm{D}}_{\mathrm{s}} \overline{\mathrm{~T}}_{\mathrm{s}}^{\prime} \overline{\mathrm{T}}_{\mathrm{D}} \overline{\mathrm{~s}}_{\mathrm{u}}=\underline{s} \tag{4.5.15}
\end{equation*}
$$

from which we obtain the normalized form

$$
\begin{align*}
& \left(\bar{T}_{s}^{\prime} \bar{T}_{s}\right) \bar{D}_{s} \underline{u}=\bar{D}_{s}^{-1} \underline{s}  \tag{4.5.16}\\
\text { with } \quad \underline{y} & =\bar{D}_{s} \underline{u} \quad \text { and } \quad \underline{g}=\bar{D}_{s}^{-1} \underline{s} . \tag{4.5.17}
\end{align*}
$$

The problem is then to solve

$$
\begin{equation*}
\overline{\mathrm{T}}_{\mathrm{s}}^{\prime} \overline{\mathrm{T}}_{\mathrm{s}} \underline{\mathrm{y}}=\underline{\mathrm{g}} \tag{4.5.18}
\end{equation*}
$$

which can be solved directly for $\underline{y}$ in terms of the auxiliary vector $\underline{h}^{*}$, where

$$
\begin{equation*}
\overline{\mathrm{T}}_{s} \underline{y}=\underline{\mathrm{h}}^{*} \quad \text { and } \quad \overline{\mathrm{T}}_{\mathrm{s}}^{\prime} \underline{h}^{*}=\underline{g} \tag{4.5.19}
\end{equation*}
$$

and are given by i.e.,
and

$$
\begin{aligned}
& \mathrm{h}_{1}^{*}=\mathrm{g}_{1} \quad ; \\
& h_{i}^{*}=g_{i}-\bar{e}_{i-1} h_{i-1}^{*}, \underset{i-m+r}{\text { for }} i=2,3, \ldots, m-1 \text {; } \\
& \left.h_{i}^{*}=g_{i}-\bar{e}_{i-1} h_{i-1}^{*} \sum_{\substack{k=i-m+1 \\
i-m+r}} \bar{t}_{k-i+m, i-m+1} h_{k}^{*}, \text { for } i=m, m+1, \ldots, p, p-1.22\right) \\
& h_{i}^{*}=g_{i}-\bar{e}_{i-1} h_{i-1}^{*}-\sum_{k=i-m+1}^{i-m+r} \bar{t}_{k-i+m, i-m+1} h_{k}^{*}-\sum_{k=i-p+1}^{i-p+r} \bar{r}_{k-i+p, i-p+1} h_{k}^{*}, \\
& \text { for } i=p, p+1, \ldots, n \text {. (4.5.23) }
\end{aligned}
$$

The back substitution process yields the final solution and is
expressed simply as
and

$$
y_{n}=h_{n}^{*}
$$

$$
\begin{equation*}
y_{i}=h_{i}^{*}-\bar{e}_{i} y_{i+1}-\sum_{j=\tilde{p}}^{\tilde{q}} \bar{t}_{i-j+m, j-m+1} y_{j}-\sum_{j=\bar{p}}^{\bar{q}} \bar{r}_{i-j+p, j-p+1}{ }_{j}, \tag{4.5.25}
\end{equation*}
$$

where $\tilde{p}, \tilde{q}, \bar{p}, \bar{q}$, can be easily defined from section 3.4 .
The final solution $\underline{u}$ is then obtained from

$$
\underline{u}=\bar{D}_{\mathrm{s}}^{-1} \underline{y}
$$

The total memory requirement for the NOBAR-3D algorithm is
$\simeq\left(r_{1}+r_{2}+5\right) N$ words. The amount of work involved for the factorization process is given by $\simeq\left(\frac{r_{1}\left(r_{1}-1\right)+r_{2}\left(r_{2}-1\right)}{2}+4 r_{1}+2 r_{2}+12\right) N$ mults $+2 N$ square roots. The normalization and forward-backward substitution processes require $2 N$ divisions and $\simeq\left(2 r_{1}+2 r_{2}+2\right) N$ mults respectively. Hence, the total operations for this algorithm are
$\simeq\left(\frac{r_{1}\left(r_{1}-1\right)+r_{2}\left(r_{2}-1\right)}{2}+6 r_{1}+4 r_{2}+16\right) N$ mults $+2 N$ square roots.

Remark
It should be noted at this point, that the given number of operations for the approximate algorithms i.e., ALUBOT-3D and NOBAR-3D, is not analogous to the corresponding number of operations for their exact counterparts, i.e., LUBOT3D and NB3D.

This is due to our main objective to design approximate 3Dalgorithms keeping the memory requirements rather than the computational work involved to a minimum.

## Chapter 5

NORMALIZED IMPLICIT CONJUGATE GRADIENT METHODS

### 5.1 INTRODUCTION

Several procedures for approximate factorization of the coefficient matrix of a large, sparse, linear system have appeared in the literature and several variants of factorization methods combined with iterative methods have been developed (see:[22],[58],[56],[18],[19],[5],[16], [23], [7], [53], [2], [42], [10], [31]).

In this chapter, we introduce a normalized implicit method for the iterative solution of large, sparse systems of algebraic linear equations, which arise from the discretization on a network of grid lines of self adjoint elliptic P.D.E's. In particular we consider the Conjugate Gradient (C.G.) method and the Normalized Implicit Conjugate Gradient (N.I.C.G.) method, which is a combination of the approximate factorization technique of sections $4.3,4.5$ and the Conjugate Gradient method, for solving the model problems of section 2.2.
5.2 THE OPTIMUM VALUE OF THE FILL-IN PARAMETER $r$ FOR THE 2D-MODEL PROBLEM

The normalized algorithm of section 3.3 was applied to many of
the standard P.D.E's of Mathematical Physics, involving two space dimensions. In this algorithm the elements $t_{i, j}$ of the upper triangular matrix T (see (3.3.4), satisfy the following theorem: Theorem 5.2.1

Let $A$ be an ( $n \times n$ ) matrix of bandwidth $m$, as is given by (3.3.2) with the properties (2.1.4) and consider the factorization $A \equiv D T^{\prime} T D$, where $D, T$ are given by (3.3.3), (3.3.4) respectively. Let $t_{i, j}$, $\mathrm{i} \varepsilon[1, \mathrm{~m}-1], \mathrm{j} \varepsilon[1, \mathrm{n}-\mathrm{m}+1]$ be the elements of T -matrix and r be the number of terms retained in bandwidth $m$. Then, the elements $t_{i, j}$
are monotonically decreased for $i \varepsilon[1, \mathrm{~m}-\mathrm{r}]$. (i.e., the sequence $t_{1, j}, t_{2, j}, \ldots . ., t_{m-r, j}, j \varepsilon[1, n-m+1]$ decreases monotonically). Proof

Because of the diagonally dominance of the coefficient matrix
A from (3.3.2) we generally obtain the inequality

$$
\begin{equation*}
a_{i}>\left|c_{i-m+1}\right|+\left|b_{i-1}\right|+\left|b_{i}\right|+\left|c_{i}\right| \tag{5.2.1}
\end{equation*}
$$

From the above relationship we can establish that no pivoting in the factorization is necessary for numerical stability. It can be easily seen that relations (3.3.5) and (3.3.9) give

$$
\begin{equation*}
\left|e_{i}\right|<1, \text { for } i=1,2, \ldots, n-1 \tag{5.2.2}
\end{equation*}
$$

Then, from (3.3.6), (3.3.10) we have

$$
\begin{equation*}
\left|t_{1, j}\right|=\frac{\left|x_{1}\right|}{d_{m+j-1}}=\frac{\left|c_{j}\right|}{d_{j} \cdot d_{m+j-1}}<1, \text { for } j=1,2, \ldots,(n-m+1) \tag{5.2.3}
\end{equation*}
$$

> A combination of (3.3.7), (3.3.10) and (5.2.2) yields the result

$$
\begin{align*}
& \left|t_{i, j}\right|=\frac{\left|x_{i}\right|}{d_{m+j-1}}=\frac{\left|-e_{i+j-2} x_{i-1}\right|}{d_{m+j-1}}=\left|-e_{i+j-2}\right| \cdot\left|t_{i-1, j}\right|<\left|t_{i-1, j}\right|, \\
& \text { for } i=2,3, \ldots, m-j \text { and }  \tag{5.2.4}\\
& j=1,2, \ldots, m-2 \text {. }
\end{align*}
$$

Similarly, from (3.3.8), (3.3.10) and (5.2.2) we obtain:

$$
\begin{gather*}
\left|t_{i, j}\right|=\frac{\left|x_{i}\right|}{d_{m+j-1}}=\frac{\left|-e_{i+j-2} x_{i-1}-\sum_{k=1}^{i-1} x_{k} t_{k-i+m, i+j-m}\right|}{d_{m+j-1}} \leqslant \frac{\left|-e_{i+j-2} x_{i-1}\right|}{d_{m+j-1}} \\
+M<\left|t_{i-1, j}\right|+M \tag{5.2.5}
\end{gather*}
$$

for either $i=(m-j+1),(m-j+2), \ldots,(m-1)$ and $j \leqslant m-1$
or $i=2,3, \ldots, m-1$ and $j>m-1$,
where

$$
\begin{equation*}
M=\frac{\left|\left(-\sum_{k=1}^{i-1} x_{k} t_{k-i+m, i+j-m}\right)\right|}{d_{m+j-1}}=0 . \tag{5.2.6}
\end{equation*}
$$

The quantity $\sum_{k=1}^{i-1} x_{k} t_{k-i+m, i+j-m}$ can be modelled as a geometric series of the form $\sum_{\lambda=1}^{n} x^{\lambda} t$ from which an upper bound can be easily determined.
(i.e. for the 2D model problem we can determine that

$$
\begin{equation*}
\sum_{k=1}^{i-1} x_{k} t_{k-i+m, i+j-m}^{<0.027} \tag{5.2.7}
\end{equation*}
$$

for either $i=(m-j+1),(m-j+2), \ldots,(m-1)$ and $j \leqslant m-1$
or $\quad i=2,3, \ldots, m-1$ for $j>m-1)$.
For values of $|t| \gg M$ and $M$ positive, from (5.2.5) we
immediately have the relation

$$
\begin{equation*}
\left|t_{i, j}\right|<\left|t_{i-1, j}\right|, \tag{5.2.8}
\end{equation*}
$$

for either $\mathrm{i} \varepsilon[\mathrm{m}-\mathrm{j}+1, \mathrm{~m}-\mathrm{r}]$ and $\mathrm{j} \varepsilon[\mathrm{r}+1, \mathrm{~m}-1]$
or $\quad i \varepsilon[2, m-r]$ and $j \varepsilon(m-1, n-m+1]$,
which guarantees the monotonicity of the terms in the T matrix. From the relationships (5.2.3), (5.2.4) and (5.2.8) the conclusion of the theorem easily follows.

The contents of the T and D matrix arrays were scrutinised
(i) for matrices of constant order ( $\mathrm{N}=50$ ) with varying bandwidth m ( $\mathrm{m}=10,15,20,30,40$ ) and
(ii) for matrices of varying order $(\mathrm{N}=50,100,200,300,400,500,1000)$ with constant bandwidth ( $\mathrm{m}=20$ ) .

In particular, the Euclidean error norms of the approximate solutions $u_{r}$ i.e., $\left[\sum\left(u_{r}\right)^{2}\right]^{\frac{1}{2}}$ which were obtained by including only r-terms in the bandwidth, are given in Figures 5.1,5.2 for the 2D-model problem.

A thorough examination of these results show that after $r=4$, the efficiency of the algorithm falls off rapidly and it requires a great deal of extra computational work, to achieve the required accuracy.


FIG.5.1 : The behaviour of the NOBAR-2D Algorithm when the coefficient matrix is of order $N=50$ and the bandwidth $m$ is varied.


FIG. 5.2: The behaviour of the NOBAR-2D Algorithm when the order $N$ of the coefficient is varied with constand bandwidth $(m=20)$.

Hence, for a certain class of 2D-model problems (see theorem 5.2.1) the NOBAR-2D algorithm can be efficiently applied with only the four outermost terms retained in the $\mathrm{T}_{\mathrm{S}}$ array. Remark

From Theorem 5.2.1 it can be easily seen that for values of $\left|t_{i, j}\right|$ such that

$$
\begin{equation*}
\frac{\left|t_{i, j}\right|-\left|t_{i-1, j}\right|}{M} \geqslant 1 \tag{5.2.9}
\end{equation*}
$$

for either $i \varepsilon[m-j+1, m-1]$ and $j \varepsilon[2, m-1]$
or $\quad i \varepsilon[2, m-1]$ and $j \varepsilon(m-1, n-m+1]$, the monotonicity relationship breaks down and does not apply. In particular it was noticed during the experimental investigation that in the case of narrow banded matrices the values of the elements of the T -matrix were not monotonic after a certain point ( $\mathrm{i}>\mathrm{m}-\mathrm{r}$ ). This is due to the values of $t_{i, j}$ becoming equal to or less than the magnitude as $M$.

Since in the examples chosen the above criteria for $M$ and $t_{i, j}$ were satisfied then we can safely adopt the procedure of neglecting the fill-in terms after r-terms.

Size effect of the coefficients of matrix A on the value of $r$
In a further investigation we examine how the size of the coefficients of matrix $A$ of the linear system (3.2.15) effect the value of the fill-in parameter $r$.

We consider experimentally the situation where the codiagonals $b_{i}$, $i \varepsilon[1, n-1]$ and $c_{i}$, $i \varepsilon[1, n-m+1]$ of (3.3.2) are of different size. Let the co-diagonals $b_{i}$ consist of larger (or strong) elements and the $m^{\text {th }}$-diagonals $c_{i}$ consist of small (or weak) elements. Then, we investigate the alternative case and by interchanging the above values i.e., the codiagonals $\mathrm{b}_{\mathrm{i}}$ consist
of weak elements and the $m^{\text {th }}$-diagonal $c_{i}$ consist of strong elements, we obtain the opposite situation. The error norms of the approximate solutions $u_{r}$, obtained by including only $r$-terms, are expressed graphically in Figure 5.3 for both cases.

From Figure 5.3 it is fairly obvious to establish that when the coefficient matrix has a strong co-diagonal the inclusion of more terms in the $\mathrm{T}_{\mathrm{s}}$ matrix produces a greater effect on the solution than if the co-diagonal is weak.

Consequently, if there are strong co-diagonal elements it is worthwhile to include as much r-terms as the computer storage requirements will permit, while in the case of weak co-diagonal elements the value of the fill-in parameter $\mathrm{r}=2$ or 3 is almost always the best and safest value to choose.

It can be observed that each of the illustrated curves of Figure 5.3 intersects the horizontal axis when $\mathrm{r}=\mathrm{m}-1$. Then the solution is obtained in one iteration, with the method being a direct one. The computational results demonstrating the above conclusions have been obtained for matrices of order $N=50$, bandwidth $\mathrm{m}=10$ for the following cases:
(i) co-diagonal elements strong $\left(b_{i}=-\frac{3}{8}, i \varepsilon[1, n-1]\right.$ and $m^{\text {th }}$ diagonal elenients weak ( $\mathrm{c}_{\mathrm{i}}=-\frac{1}{8}, \mathrm{i} \varepsilon[1, \mathrm{n}-\mathrm{m}+1]$ ),
(ii) standard case ( $\mathrm{b}_{\mathrm{i}}=-\frac{1}{4}, \mathrm{i} \varepsilon[1, \mathrm{n}-1]$ and $\left.\mathrm{c}_{\mathrm{i}}=-\frac{1}{4}, \mathrm{i} \varepsilon[1, \mathrm{n}-\mathrm{m}+1]\right)$
and
(iii) co-diagonal elements weak ( $b_{i}=-\frac{1}{8}, i \varepsilon[1, n-1]$ and $m^{\text {th }}$ diagonal elements strong ( $c_{i}=-\frac{3}{8}, i \varepsilon[1, n-m+1]$ ).

The solution vector $\underline{u}$ was chosen to be a unity vector and the error norm of the approximate solution $u_{r}$ has been taken to be the Euclidean error norm, i.e., $\left[\sum\left(u-u_{r}\right)^{2}\right]^{\frac{1}{2}}$.


FIG. 5.3 : case $i$ : Codiagonals strong $-m^{\text {th }}$ diagonals weak, case $\mathrm{ii}:$ Standard case
case iii:Codiagonals weak - $m^{\text {th }} d i a g o n a l s$ strong.

### 5.3 THE CONJUGATE GRADIENT METHOD AS AN ITERATIVE PROCEDURE

One of the currently favourite iterative methods for the solution of large, sparse linear systems with an arbitrary symmetric, positive definite coefficient matrix is the Conjugate Gradient (C.G.) method, previously discussed in section 2.4. The reasons for its popularity are:
(i) it is easy to program [49],
(ii) it does not require any estimation of acceleration parameters,
(iii) it takes advantage of the distribution of the eigenvalues of the iteration operator,
(iv) it is relatively efficient [50].

Next, we present a revised form of the Conjugate Gradient algorithm in order to achieve a more convenient form.

Consider the linear system (2.4.1), where the coefficient matrix A is symmetric, positive definite.

Given an initial approximation $\underline{x}_{0}$ to the solution $\underline{x}$ then, we form the residual ${\underset{\sim}{r}}_{0}$ such that

$$
\begin{equation*}
\underline{r}_{0}=\underline{s}-A \underline{x}_{0} \tag{5.3.1}
\end{equation*}
$$

and let the directional vector $\underline{\sigma}_{0}$ be denoted as

$$
\begin{equation*}
\underline{\sigma}_{0}=\underline{r}_{0} . \tag{5.3.2}
\end{equation*}
$$

Then the iterative scheme proceeds as follows. We calculate the scalar $a_{i}$, i.e.,

$$
\begin{equation*}
a_{i}=\frac{\left(\underline{r}_{i}, \underline{r}_{i}\right)}{\left(\underline{\sigma}_{i}, \mathrm{~A} \underline{\underline{i}}_{\mathrm{i}}\right)} \tag{5.3.3}
\end{equation*}
$$

or equivalently,

$$
\begin{equation*}
a_{i}=\frac{\left(\underline{r}_{i}, \underline{\sigma}_{i}\right)}{\left(\underline{\sigma}_{i}, \mathrm{~A} \underline{-}_{i}\right)} . \tag{5.3.3a}
\end{equation*}
$$

Then, we calculate the new iterate to minimize $F[\underline{x}]$ along $\underline{\sigma}_{i}$
(see section 2.4),

$$
\begin{equation*}
\underline{x}_{i+1}=\underline{x}_{i}+a_{i} \underline{\sigma}_{i} . \tag{5.3.4}
\end{equation*}
$$

Now the new residual $\underline{r}_{i+1}$ is formed such that

$$
\begin{equation*}
\underline{r}_{i+1}=\underline{r}_{i}-a_{i}{\underset{-i}{i}}, \tag{5.3.5}
\end{equation*}
$$

and the scalar $\beta_{i+1}$ is calculated,

$$
\begin{equation*}
\beta_{i+1}=\frac{\left(\underline{r}_{i+1}, \underline{r}_{i+1}\right)}{\left(\underline{r}_{i}, \underline{r}_{i}\right)} \tag{5.3.6}
\end{equation*}
$$

or equivalently,

$$
\begin{equation*}
\beta_{i+1}=\frac{-\left(\underline{r}_{i+1}, A \underline{\sigma}_{i}\right)}{\left(\underline{\sigma}_{i}, A \underline{\sigma}_{i}\right)} . \tag{5.3.6a}
\end{equation*}
$$

Then finally we calculate the new direction to be A-conjugate (see Definition 1.6 .1 ) to the preceding direction such that,

$$
\begin{equation*}
\underline{-}_{i+1}=\underline{\mathrm{r}}_{\mathrm{i}+1}+\beta_{\mathrm{i}+1} \underline{\sigma}_{\mathrm{i}} . \tag{5.3.7}
\end{equation*}
$$

Since the coefficient matrix A remains unmodified there is no need for it to be stored explicitly. The storage requirements and the number of multiplications per iteration required by the C.G. algorithm for the model problems are given in Tables 5.1 and 5.2.

Note that the C.G. algorithm can be efficiently applied for the solution of large, sparse systems of equations in the case where the matrix decomposition is considered to be impractical [46]. Furthermore, the algorithm can be applied when an approximate (relatively close) solution to the linear system is desirable (e.g. when the system is discrete approximation to a P.D.E.). Then, the level of correspondence of the model to the real system determines the required level of accuracy for the solution of the linear system and the algorithm could be terminated early [46].

Generally, the qualitative and quantitative behaviour of C.G. method is very well understood, [32], [38], [50], [20], [15].

### 5.4 THE NORMALIZED IMPLICIT CONJUGATE GRADIENT METHOD

In this section, we introduce the Normalized Implicit Conjugate Gradient (N.I.C.G.) method for solving the indicated model problems. As with the C.G. method, the N.I.C.G. method is very efficient, makes no assumptions about the structure of the coefficient matrix $A$, requires no estimation of iterative parameters and is easy to program.

Instead of considering the linear system (2.4.1) we now consider the linear system

$$
\begin{equation*}
\left(D_{s} T_{s}^{\prime} T_{s} D_{s}\right)^{-1} A \underline{x}=\left(D_{s} T_{s}^{\prime} T_{s} D_{s}\right)^{-1} \underline{s} \tag{5.4.1}
\end{equation*}
$$

where $D_{S} T_{S}^{\prime} T_{S} D_{S}$ is the known approximate factorization of $A$, as given in sections $4.3,4.5$ with $D_{S}, T_{s}$ defined as in (4.3.2), (4.3.3) and (4.5.2), (4.5.3) respectively.

Once the factorization has been computed, the system (5.4.1) is solved by the C.G. method.

Let $\underline{x}_{0}$ be an arbitrary initial approximation of the solution $\underline{x}$, form the residual

$$
\begin{equation*}
\underline{r}_{0}=\underline{s}-A \underline{x}_{0} . \tag{5.4.2}
\end{equation*}
$$

We proceed to solve

$$
\begin{align*}
& \left(\mathrm{D}_{\mathrm{s}} \mathrm{~T}_{\mathrm{s}}^{\mathrm{T}} \mathrm{~T}_{\mathrm{s}} \mathrm{D}_{\mathrm{s}}\right) \underline{r}_{0}^{+}=\underline{r}_{0},  \tag{5.4.3}\\
& \underline{\sigma}_{0}=\underline{r}_{0}^{+} . \tag{5.4.4}
\end{align*}
$$

Then, for $i=0,1,2, \ldots$ calculate the vectors $x_{i+1},{ }_{-1+1},{ }_{-}^{\sigma}-1$ and the scalar quantities $a_{i}, \beta_{i+1}$ as follows:

$$
\begin{align*}
& a_{i}=\frac{\left(\underline{r}_{i}, \underline{r}_{i}^{+}\right)}{\left(\underline{\sigma}_{i}, \underline{A}_{i}\right)},  \tag{5.4.5}\\
& \underline{x}_{i+1}=\underline{x}_{i}+a_{i} \underline{\sigma}_{i},  \tag{5.4.6}\\
& \underline{r}_{i+1}=\underline{r}_{i}-a_{i} A \underline{\sigma}_{i} . \tag{5.4.7}
\end{align*}
$$

and
Then we solve ( $D_{S}^{T} S_{s}^{\prime} S_{s} D_{s}$ ) $\underline{r}_{i+1}^{+}=\underline{r}_{i+1}$,
and evaluate

$$
\begin{equation*}
\beta_{i+1}=\frac{\left(\underline{r}_{i+1}, \underline{r}_{i+1}^{+}\right)}{\left(\underline{r}_{i}, \underline{r}_{i}^{+}\right)}, \tag{5.4.9}
\end{equation*}
$$

and

$$
\begin{equation*}
\underline{\sigma}_{i+1}=\underline{r}_{i+1}^{+}+\beta_{i+1} \underline{\sigma}_{i} . \tag{5.4.10}
\end{equation*}
$$

An equivalent, but more computational form of the iterative scheme (5.4.2)-(5.4.10) is given below:

$$
\begin{array}{ll}
\text { Form } & \underline{-}_{0}=\underline{s-A x} \underline{x}_{0}, \\
\text { set } & \underline{\underline{r}}_{0}=D_{s}^{-1} \underline{r}_{0}, \\
\text { solve } & \left(T_{s}^{\prime} T_{s}\right) \cdot \underline{r}_{0}^{*}=\underline{\underline{r}}_{-0}, \\
\text { and set } & \underline{\sigma}_{0}=\underline{r}_{-0}^{*},
\end{array}
$$

Then for $i=0,1,2,3, \ldots$ calculate the vectors $\underline{x}_{i+1}, \underline{\underline{\tilde{r}}}_{i+1}, \widetilde{\sigma}_{i+1}$ and the scalar quantities $\tilde{\mathrm{a}}_{\mathrm{i}}, \widetilde{\beta}_{i+1}$ as follows:

$$
\begin{align*}
& \tilde{a}_{i}=\frac{\left(\underline{\underline{r}}_{i}, \underline{r}_{i}^{*}\right)}{\left(\underline{\tilde{\sigma}}_{i}, A \tilde{\sigma}_{i}\right)},  \tag{5.4.15}\\
& \underline{x}_{i+1}=\underline{x}_{i}+\tilde{a}_{i} \tilde{\sigma}_{i},  \tag{5.4.16}\\
& \tilde{\underline{r}}_{i+1}=\tilde{\tilde{r}}_{i}-\tilde{a}_{i} \tilde{\sigma}_{i} .
\end{align*}
$$

and

$$
\begin{equation*}
\left(\mathrm{T}_{\mathrm{S}}^{\prime} \mathrm{T}_{\mathrm{S}}\right) \underline{\mathrm{r}}_{\mathrm{i}+1}^{*}=\underline{\underline{\boldsymbol{r}}}_{\mathrm{i}+1} \tag{5.4.18}
\end{equation*}
$$

and evaluate

$$
\begin{equation*}
\tilde{\beta}_{i+1}=\frac{\left(\tilde{\tilde{r}}_{i+1}, \underline{r}_{i+1}^{*}\right)}{\left(\underline{\tilde{r}}_{i}, \underline{\underline{r}}_{i}^{*}\right)}, \tag{5.4.19}
\end{equation*}
$$

and

$$
\begin{equation*}
\tilde{\underline{\sigma}}_{i+1}=\underline{r}_{i+1}^{*}+\widetilde{\beta}_{i+1} \tilde{\underline{\sigma}}_{i} . \tag{5.4.20}
\end{equation*}
$$

Geometrical representation of the N.I.C.G. method
Consider the linear system (2.4.1) and the corresponding quadratic function $F[\underline{x}]$, given by (2.4.2), which defines an ellipsoid $E$ in the $N$-dimensional space of the elements of $\underline{x}_{i}$ as is shown in Figure 5.4.


FIGURE 5.4

The ellipses $c_{i}, i=0,1,2, \ldots$ are the intersections of several planes defined by the vector ${\underset{\mathrm{t}}{\mathrm{i}}}$, tangent to the ellipse $\mathrm{c}_{\mathrm{i}}$ at $\underline{x}_{\mathrm{i}}$ and the residual ${\underset{\mathrm{r}}{\mathrm{i}}}$ for $\mathrm{i}=0,1,2, \ldots$, and the surface $\mathrm{F}[\underline{x}]=$ constant.

By """ we denote the projection of vectors and ellipses in the plane P .

Consider now the linear system (5.4.1) and the corresponding quadratic function $F_{\tau}[\underline{x}]$, given by

$$
\begin{equation*}
\mathrm{F}_{\tau}[\underline{x}]=\frac{1}{2} \underline{x} A^{*} \underline{x}-\underline{s}^{*} \underline{x} \tag{5.4.21}
\end{equation*}
$$

where

$$
\begin{equation*}
A^{*}=\left(D_{S} T_{s}^{\prime} T_{s} D_{s}\right)^{-1} A \quad \text { and } s^{*}=\left(D_{S} T_{s}^{\prime} T D_{s}\right)^{-1} \underline{s} \tag{5.4.22}
\end{equation*}
$$

[Note that, in the case of the approximate factorization $A \approx D_{s}^{1} T D_{S}$ (see remark of section 4.3 ), the values of $A^{*}$ and $s^{*}$ in the quadratic function (5.4.21) are given by

$$
\begin{equation*}
A^{*}=\left(T_{s}^{\prime T}\right)^{-1} T^{\prime} T \text { and } \underline{s}^{*}=\left(D T_{s}^{\prime} \mathrm{T}_{\mathrm{s}}\right)^{-1} \underline{s} \tag{5.4.23}
\end{equation*}
$$

respectively].

Then, a new ellipsoid $\mathrm{E}_{\tau}$ is defined by the transformed quadratic function $F_{\tau}[\underline{x}]$, in the $N$-dimensional space of the elements of $\underline{x}_{i}$ (see Figure 5.5), with the solution of the method occurring at the $\mathrm{x}_{\mathrm{k}}$.


FIGURE 5.5
The ellipses $\tilde{c}_{i}$, are in this case, the intersections of several planes defined by the vector $\tilde{\underline{t}}_{i}$, tangent to the ellipse $\tilde{c}_{i}$ at $\underline{x}_{i}$, and the transformed residual $\tilde{\underline{r}}_{i}$ for $i=0,1,2, \ldots$, and the surface $F_{\tau}[\underline{x}]=$ constant .

Since the minimization of the transformed quadratic function (5.4.21) is equivalent to the solution of the system (5.4.1), the problem now reduces to the minimization of the quadratic function $\mathrm{F}_{\tau}[\underline{x}]$, instead of minimizing the quadratic function $\mathrm{F}[\underline{x}]$ as given by (2.4.2) and corresponding to the system (2.4.1).

Note that for non-positive definite matrices the existence of the minimum of the quadratic functions $F[\underline{x}]$ and $F_{\tau}[\underline{x}]$ respectively is not unique.

Let an arbitrary vector ${\underset{\sim}{x}}_{0}$ be the initial guess (a point on the surface of the ellipsoid under consideration) and consider the hyperellipsoid $\mathrm{F}_{\tau}[\underline{\mathrm{x}}]=\mathrm{F}_{\tau}\left[\underline{\mathrm{x}}_{0}\right]$ passing through this point.

The minimum of $\mathrm{F}_{\tau}\left[\underline{x}_{0}\right]$ occurs at the centre of this hyperellipsoid. Then, in each iteration we move along the chord of the hyperellipsoid, which is normal to the surface at $\underline{x}_{0}$ (i.e., the direction of ${\underset{\sim}{\sigma}}_{0}$ ) in order to arrive at $\underline{x}_{1}$.

Note that $F_{\tau}\left[\underline{x}_{0}\right]=F_{\tau}\left[\underline{x}_{0}+2 \tilde{a}_{0} \tilde{\sigma}_{0}\right]$ and consequently $\underline{x}_{1}=\underline{x}_{0}+\tilde{a}_{0} \tilde{\sigma}_{0}$ is the mid-point of this normal chord. Then, consider the projection of the hyperellipsoid $\mathrm{F}_{\tau}[\underline{\mathrm{x}}]=\mathrm{F}_{\tau}\left[\underline{x}_{1}\right]$ in the ( $\mathrm{N}-1$ ) dimensional space, that is A-conjugate to $\tilde{\sigma}_{0}$ and the above procedure is repeated. The iterative process is continued, so that at each iterative step the dimension of the space is decreased by one until finally after N -iterations all the space is evacuated or terminated when a suitable convergence criterion is satisfied.

The Three Dimensional Case
The algorithms of sections 5.3 and 5.4 can be generalized to apply to the solution of the self-adjoint, second order Elliptic P.D.E. in three space dimensions. Then, proceeding in an analogous manner to the two dimensional case, we consider the C.G.3D and the N.I.C.G.3D, which is a combination of the approximate factorization technique of section 4.5 and the C.G. algorithm, methods to solve the 3D-model problem.

### 5.5 COMPUTATIONAL WORK AND EXPERIMENTAL RESULTS IN TWO AND THREE DIMENSIONS

A summary of the storage and number of multiplications per iteration required for both methods applied to the model and general problems (i.e., the case of the self adjoint P.D.E's with variable coefficients in the unit square) in two and three dimensions, are given in Tables 5.1 and 5.2 respectively.

| Equation | Vectors in Storage | Storage Required | Number of mults per iteration | Number of mults for the solution of ( $\mathrm{T}_{\mathrm{s}} \mathrm{T}_{\mathrm{s}}$ ) $\underline{r}_{i}^{*}=\widetilde{r}_{i}$ |
| :---: | :---: | :---: | :---: | :---: |
| Laplace: |  |  |  |  |
| C.G. | $\underline{\mathrm{r}}, \underline{\mathrm{x}}, \underline{\sigma}, \mathrm{A} \underline{\sigma}$ | 4 N | $6 \mathrm{~N}+2$ | 0 |
| N.I.C.G. | $\underline{\underline{\tilde{r}}}, \underline{x}, \underline{\tilde{\sigma}}, A \underline{\tilde{\sigma}}, \underline{a}, \underline{b}, \underline{s}, \underline{u}$ <br> plus $r$-vectors in $T_{s}$ matrix | $(8+r) N-r n$ | $(2 \mathrm{r}+8) \mathrm{N}-\mathrm{rn}-\mathrm{r}$ | $(2 \mathrm{r}+2) \mathrm{N}-\mathrm{rn}-\mathrm{r}-2$ |
| General: |  |  |  |  |
| C.G. | $\underline{r}, \underline{x}, \underline{\sigma}, \mathrm{~A} \underline{\sigma}, \underline{a}, \underline{b}, \underline{c}$ | $7 \mathrm{~N}-\mathrm{n}$ | $10 \mathrm{~N}-4 \mathrm{n}+2$ | 0 |
| N.I.C.G. | $\underline{\tilde{\tilde{x}}}, \underline{x}, \underline{\tilde{\sigma}}, A \underline{\tilde{\sigma}}, \underline{a}, \underline{b}, \underline{c}, \underline{s}, \underline{u},$ <br> plus $r$-vectors in $T_{s}$ matrix | $(9+r) N-r n$ | $(2 \mathrm{r}+12) \mathrm{N}-(\mathrm{r}+4) \mathrm{n}-\mathrm{r}$ | $(2 \mathrm{r}+2) \mathrm{N}-\mathrm{rn}-\mathrm{r}-2$ |

[^0]| Equation | Vectors in Storage | Storage Required | Number of mults per iteration | Number of mults for $t$ solution of $\left(T_{s}^{\prime} T_{s}\right) r_{i}^{*}=$ |
| :---: | :---: | :---: | :---: | :---: |
| Laplace: |  |  |  |  |
| C.G.3D | $\underline{\mathrm{r}}, \underline{\mathrm{x}}, \underline{\sigma}, \mathrm{A} \underline{\sigma}$ | 4N | $6 \mathrm{~N}+2$ | 0 |
| N.I.C.G.3D | $\underline{\underline{\tilde{r}}}, \underline{x}, \underline{\tilde{q}}, A \underline{\underline{\delta}}, \underline{a}, \underline{b}, \underline{s}, \underline{u}$ <br> plus $r_{1}$ and $r_{2}$ vectors in $T_{s}$ matrix | $\left(8+r_{1}+r_{2}\right) N-r_{2} n^{2}-r_{1} n$ | $\begin{array}{r} \left(2 r_{1}+2 r_{2}+8\right) N-r_{2} n^{2}-r_{1} n- \\ -r_{1}-r_{2} \end{array}$ | $\begin{array}{r} \left(2 r_{1}+2 r_{2}+2\right) N-r_{2} n^{2}-r_{1} n . \\ -r_{1}-r_{2}-2 \end{array}$ |
| General: |  |  |  |  |
| C.G.3D | $\underline{r}, \underline{x}, \underline{\sigma}, A \underline{\theta}, \underline{a}, \underline{b}, \underline{c}, \underline{h}$ | $8 \mathrm{~N}-\mathrm{n}^{2}-\mathrm{n}$ | $12 \mathrm{~N}-6 \mathrm{n}^{2}+2$ | 0 |
| N.I.C.G.3D | $\begin{aligned} & \underline{\tilde{r}}, \underline{x}, \underline{\theta}, A \underline{\tilde{\sigma}}, \underline{a}, \underline{b}, \underline{c}, \underline{h}, \underline{s}, \underline{u} \\ & \text { plus } r_{1} \text { and } r_{2} \text { vectors in } T_{s} \text { matrix } \end{aligned}$ | $\left(10+r_{1}+r_{2}\right) N-\left(1+r_{2}\right) n^{2}-\left(1+r_{1}\right) n$ | $\left(\begin{array}{c} \left(2 r_{1}+2 r_{2}+14\right) N-\left(r_{2}+6\right) n^{2}- \\ -r_{1} n-r_{1}-r_{2} \end{array}\right.$ | $\begin{gathered} \left(2 r_{1}+2 r_{2}+2\right) N-r_{2} n^{2}-r_{1} n- \\ -r_{1}-r_{2}-2 \end{gathered}$ |

[^1]It can be easily seen from Tables 5.1, 5.2 that an upper bound on the arithmetic work per iteration for the N.I.C.G. method is $0(20 N)$ multiplications when $\mathrm{r}=4$ and $O(14 \mathrm{~N})$ multiplications when $\mathrm{r}=1$, whilst for the N.I.C.G.3D method we obtain $O(18 N)$ multiplications when $r_{1}=r_{2}=1$ and $O(30 N)$ multiplications when $r_{1}=r_{2}=4$ respectively.

The numerical results for both methods on the model problems, are given in Tables 5.3 and 5.4 respectively.

The initial "guess" $\underline{x}_{0}$ was chosen to be the zero vector.
The solution vector was chosen to be:
(i) $x_{i}=0, i \varepsilon[1, N]$ with $x_{m+1}=1$, (case II)
$i \neq m+1$
and
(ii) a vector of N -pseudo-random numbers from a uniform (rectangular) distribution on the range ( 0,1 ), (case I).

The right hand side vector of equation (2.4.1) was obtained as the product of the solution vector $\underline{x}$ with the coefficient matrix $A$ of (2.4.1). Following Ginsburg [20] (p.68) and Reid [50] (p.243) the error was taken to be the maximum norm of the recursive residual. The iterative process was terminated when the error was less than the relative precision of the arithmetic, which has been chosen to be $10^{-6}$, $10^{-8}, 10^{-10}$ respectively. The behaviour of the following four error measures

$$
\left.\left\|r_{i}\right\|_{2},\left(r_{i},\left(T_{s}^{\prime} T_{s}\right)^{-1} r_{i}\right)\right)^{\frac{1}{2}},\left(r_{i},\left(x_{k}-x\right)\right)^{\frac{1}{2}},\left\|x_{k}-x\right\|_{2}
$$ where $x_{k}$ means the $\mathrm{k}^{\text {th }}$ approximant to the exact solution x , applied to the Laplacian matrices of order $(49 \times 49),(12 \times 12 \times 12)$ respectively, and for the model problems, is given in Figures 5.7-5.10.

The effect of number of equations on the rate of convergence for the 2D-model problem case II, is given in Figure 5.6. Tlie computations were performed on an ICL 1904 computer. In addition, extreme cases of the problems were computed on an ICL 1906A machine.

|  |  | Case I |  |  |  |  | Case II |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\varepsilon=10^{-6}$ | Mesh size$h^{-1}$ | C.G. | N.I.C.G. |  |  |  | C.G. | N.I.C.G. |  |  |  |
|  |  |  | $\mathrm{r}=1$ | 2 | 3 | 4 |  | $\mathrm{r}=1$ | 2 | 3 | 4 |
|  | 5 | 10 | 7 | 6 | 6 | 5 | 10 | 6 | 6 | 6 | 1* |
|  | 10 | 25 | 12 | 9 | 8 | 8 | 26 | 10 | 8 | 7 | 8 |
|  | 15 | 40 | 15 | 12 | 10 | 10 | 39 | 13 | 10 | 8 | 8 |
|  | 20 | 53 | 19 | 15 | 12 | 11 | 50 | 16 | 11 | 10 | 9 |
|  | 30 | 75 | 24 | 18 | 15 | 14 | 68 | 21 | 14 | 12 | 11 |
|  | 40 | 101 | 31 | 23 | 20 | 17 | 81 | 24 | 17 | 15 | 14 |
|  | 50 | >110 | 37 | 26 | 23 | 20 | >100 | 23 | 18 | 16 | 15 |
|  | 60 | >110 | 43 | 31 | 26 | 23 | >100 | 23 | 17 | 16 | 15 |


| $\varepsilon=10^{-8}$ | 5 | $\sim 10$ | 9 | 8 | 7 | 7 | $\sim 10$ | 9 | 7 | 7 | 1* |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | 10 | 31 | 14 | 11 | 11 | 10 | 29 | 12 | 10 | 9 | 10 |
|  | 15 | 47 | 20 | 15 | 13 | 13 | 46 | 16 | 12 | 11 | 11 |
|  | 20 | 63 | 25 | 18 | 16 | 16 | 59 | 19 | 15 | 13 | 13 |
|  | 30 | 91 | 35 | 23 | 21 | 20 | 84 | 26 | 20 | 17 | 15 |
|  | 40 | 123 | 40 | 30 | 27 | 24 | 108 | 32 | 23 | 19 | 17 |
|  | 50 | >130 | 49 | 34 | 32 | 26 | >120 | 39 | 27 | 22 | 20 |
|  | 60 | >130 | 58 | 40 | 37 | 30 | >120 | 45 | 31 | 25 | 23 |


| $\varepsilon=10^{-10}$ | 5 | $\sim 10$ | 11 | 9 | 9 | 8 | $\sim 10$ | 11 | 9 | 9 | 1 * |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | 10 | 35 | 18 | 13 | 13 | 13 | 32 | 15 | 11 | 11 | 12 |
|  | 15 | 53 | 23 | 17 | 16 | 16 | 51 | 18 | 14 | 13 | 13 |
|  | 20 | 71 | 28 | 21 | 19 | 18 | 68 | 22 | 18 | 15 | 15 |
|  | 30 | 105 | 40 | 28 | 24 | 23 | 99 | 31 | 24 | 20 | 19 |
|  | 40 | 140 | 52 | 36 | 31 | 28 | 127 | 39 | 30 | 25 | 23 |
|  | 50 | >140 | 63 | 44 | 36 | 33 | >130 | 47 | 35 | 29 | 25 |
|  | 60 | >140 | 73 | 51 | 42 | 38 | >130 | 55 | 41 | 33 | 28 |

TABLE 5.3: Number of iterations required to reduce the error to $1 \mathrm{E}-6,1 \mathrm{E}-8,1 \mathrm{E}-10$ respectively for the 2 D -model problem. The resulting sparse matrices, for the above considered mesh sizes, are of order 16,81,196, 361,841,1421,2401,3481 respectively.

|  |  | Case I |  |  | Case II |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Mesh <br> size $\mathrm{h}^{-1}$ | C.G.3D | N.I.C.G.3D |  | C.G.3D | N.I.C.G.3D |  |
|  |  |  | $\mathrm{r}_{1}=\mathrm{r}_{2}=1$ | $\mathrm{r}_{1}=\mathrm{r}_{2}=4$ |  | $\mathrm{r}_{1}=\mathrm{r}_{2}=1$ | $\mathrm{r}_{1}=\mathrm{r}_{2}=4$ |
| $\varepsilon=10^{-6}$ | 5 | 18 | 8 | 8 | 18 | 7 | 6 |
|  | 7 | 27 | 10 | 9 | 25 | 9 | 8 |
|  | 9 | 33 | 13 | 11 | 29 | 10 | 9 |
|  | 11 | 41 | 14 | 13 | 33 | 12 | 10 |
|  | 13 | 47 | 16 | 14 | 39 | 13 | 11 |


| $\varepsilon=10^{-8}$ | 5 | 22 | 10 | 9 | 22 | 9 | 8 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | 7 | 33 | 14 | 12 | 32 | 12 | 10 |
|  | 9 | 43 | 17 | 15 | 39 | 14 | 12 |
|  | 11 | 53 | 20 | 17 | 44 | 16 | 14 |
|  | 13 | 62 | 24 | 19 | 52 | 18 | 15 |


| $\varepsilon=10^{-10}$ | 5 | 26 | 13 | 11 | 25 | 12 | 10 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | 7 | 38 | 16 | 14 | 37 | 14 | 12 |
|  | 9 | 50 | 20 | 17 | 47 | 17 | 14 |
|  | 11 | 62 | 24 | 20 | 54 | 19 | 17 |
|  | 13 | 73 | 28 | 23 | 66 | 23 | 19 |

TABLE 5.4: Number of iterations required to reduce the error to $1 \mathrm{E}-6,1 \mathrm{E}-8,1 \mathrm{E}-10$ respectively for the 3D-model problem. The resulting sparse matrices, for the above considered mesh sizes, are of order $64,216,512,1000,1728$ respectively.


FIG. 5.6 : Effect of number of equations on the rate of convergence (2D-model problem, case II).


FIG. 5.7 : Behaviour of the Error measures applied to the Laplacian matrix of order $\mathrm{N}=24 \mathrm{Ol}$ (Grid $49 \times 49$ ).


FIG. 5.8 : Behaviour of the Error measures applied to the Laplacian matrix of order $\mathrm{N}=2401$ (Grid $49 \times 49$ ).


FIG. 5.9 : Behaviour of the Error measures applied to the Laplacian - -tain af nader N 1728 (Grid $12 \times 12 \times 12$ ).


FIG. $5.10: \quad$ Behaviour of the error measures applied to the
matrix of order $N=1728$ (Grid/ $2 \times 12 \times 12$ ).

Chapter 6

NORMALIZED SECOND ORDER METHODS

### 6.1 INTRODUCTION

In this chapter, the representation of the Conjugate Gradient method as a non-stationary second degree iteration method is given. We, then, develop the Normalized Implicit Conjugate Gradient method as a second order method, giving a proposal for certain values of the iterative parameters $\rho_{i}, \gamma_{i}$ involved in both two and three space dimensions. Next, we proceed to apply the approximate two dimensional algorithms in conjunction with the standard well known stationary and non-stationary iterative methods, i.e., the Simultaneous Displacement, Second order Richardson and Chebychev methods. An experimental estimation of the optimal iteration parameters involved are obtained, resulting in a substantial saving in computational work.
6.2 THE CONJUGATE GRADIENT METHOD AS A SECOND DEGREE METHOD (C.G.S.D.)

In the following, we derive an expression for $\underline{x}_{i+1}$ in terms of $\underline{x}_{i}$ and $x_{i-1}$. By (5.3.4) we have

$$
\begin{equation*}
\underline{x}_{i+1}=\underline{x}_{i}+a_{i}{ }_{-}-i \tag{6.2.1}
\end{equation*}
$$

and

$$
\begin{equation*}
\underline{x}_{i}=\underline{x}_{i-1}+a_{i-1} \underline{\sigma}_{i-1} \tag{6.2.2}
\end{equation*}
$$

$$
\begin{equation*}
\frac{\beta_{i}}{a_{i-1}} a_{i} \underline{x}_{i}=\frac{\beta_{i}}{a_{i-1}} a_{i-x_{i-1}}+a_{i} \beta_{i} \underline{\sigma}_{i-1} \tag{6.2.3}
\end{equation*}
$$

From (6.2.1) and (5.3.7) we obtain

$$
\begin{equation*}
\underline{x}_{i+1}=\underline{x}_{i}+a_{i}\left(\underline{r}_{i}+\beta_{i} \underline{\sigma}_{i-1}\right) \tag{6.2.4}
\end{equation*}
$$

and a combination of (6.2.3), (6.2.4) leads to

$$
\begin{equation*}
\underline{x}_{i+1}=\underline{x}_{i}+a_{i} \underline{r}_{i}+\frac{\beta_{i}}{a_{i-1}} a_{i} \underline{x}_{i}-\frac{\beta_{i}}{a_{i-1}} a_{i} \underline{x}_{i-1}, \tag{6.2.5}
\end{equation*}
$$

or equivalently,

$$
\begin{equation*}
\underline{x}_{i+1}=\underline{x}_{i-1}+\rho_{i+1}\left(\gamma_{i+1} \underline{r}_{i}+\underline{x}_{i}-x_{i-1}\right), \tag{6.2.6}
\end{equation*}
$$

where

$$
\begin{equation*}
\rho_{i+1}=1+\frac{a_{i}}{a_{i-1}} \beta_{i} \tag{6.2.7}
\end{equation*}
$$

$$
\begin{equation*}
\gamma_{i+1}=\frac{a_{i}}{\rho_{i+1}} \quad, \quad i=0,1,2, \ldots, k-1 \tag{6.2.8}
\end{equation*}
$$

with $k$ the smallest integer such that $\underline{r}_{k}=\underline{0}$.
The derivation of formulae for the evaluation of parameters $\rho_{i+1}$, $\gamma_{i+1}$ using the orthogonality relations (2.4.20) follows from equation (6.2.6) and by substituting

$$
\begin{equation*}
\underline{r}_{i}=\underline{s}-\underline{A x}_{i}, \quad i=0,1,2, \ldots \tag{6.2.9}
\end{equation*}
$$

we obtain the result

$$
\begin{equation*}
\underline{r}_{i+1}=\underline{r}_{i-1}^{+\rho}{ }_{i+1}\left(-\gamma_{i+1} \underline{A r}_{i}+\underline{r}_{i}-\underline{r}_{i-1}\right) . \tag{6.2.10}
\end{equation*}
$$

By forming the inner product of both sides of equation (6.2.10) with ${\underset{\mathrm{r}}{\mathrm{i}}}$ we have,

$$
\begin{equation*}
0=\rho_{i+1}\left(-\gamma_{i+1}\left(\underline{r}_{i}, \underline{A r}_{i}\right)+\left(\underline{r}_{i}, \underline{r}_{i}\right)\right) \tag{6.2.11}
\end{equation*}
$$

and from (6.2.11) assuming that $\rho_{i+1} \neq 0$ we get,

$$
\begin{equation*}
\gamma_{i+1}=\frac{\left(\underline{r}_{i}, \underline{r}_{i}\right)}{\left(\underline{r}_{i}, \underline{A r}_{i}\right)} \tag{6.2.12}
\end{equation*}
$$

Similarly, by taking the inner product of both sides of equation (6.2.10) with ${\underset{\mathrm{r}}{\mathrm{i}-1}}^{\text {yields: }}$

$$
\begin{equation*}
0=\left(\underline{r}_{i-1}, \underline{r}_{i-1}\right)+\rho_{i+1}\left(-\gamma_{i+1}\left(\underline{r}_{i-1}, \underline{A r}_{i}\right)-\left(\underline{r}_{i-1}, \underline{r}_{i-1}\right)\right) \tag{6.2.13}
\end{equation*}
$$

and replacing index $i$ by $i-1$ in (6.2.10) we obtain,

$$
\begin{equation*}
\underline{r}_{i}=\underline{r}_{i-2}+\rho_{i}\left(-\gamma_{i} \underline{A r}_{i-1}+\underline{r}_{i-1} \underline{r}_{i-2}\right) \tag{6.2.14}
\end{equation*}
$$

Taking the inner product of both sides of (6.2.14) with $\underline{r}_{\mathrm{i}}$ yields:

$$
\begin{align*}
& \left(\underline{r}_{i}, \underline{r}_{i}\right)=\rho_{i}\left(-\gamma_{i}\left(r_{i}, A r_{i-1}\right)\right)  \tag{6.2.15}\\
& \left(\underline{r}_{i-1}, A \underline{r}_{i}\right)=\left(\underline{r}_{i}, A \underline{r}_{i-1}\right)=-\frac{\left(\underline{r}_{i}, \underline{r}_{i}\right)}{\gamma_{i} \rho_{i}} \tag{6.2.16}
\end{align*}
$$

or
and then, combining (6.2.13) with $(6.2 .16)$ we obtain the result

$$
\begin{array}{r}
\rho_{i+1}=\left[1+\frac{\left(\frac{r}{i-1}, \underline{A r}_{i}\right)}{\left(\underline{r}_{i-1}, \underline{r}_{i-1}\right)} \gamma_{i+1}\right]^{-1}=\left[1-\frac{\left(\underline{r}_{i}, \underline{r}_{i}\right)}{\left(\underline{r}_{i-1}, \underline{r}_{i-1}\right)} \frac{r_{i+1}}{\gamma_{i}} \frac{1}{\rho_{i}}\right]^{-1} \\
\text { for } i=1,2,3, \ldots .
\end{array}
$$

(Note that $\rho_{1}=1$, since $\underline{x}_{1}=\underline{x}_{0}+a{ }_{0} \underline{r}_{0}$ ).
The formulae (6.2.12), (6.2.17) are used to generate the parameters $\rho_{i}, \gamma_{i}$ in a recursive form in the C.G. method. An alternative derivation of these formulae can be found in [62].
6.3 THE NORMALIZED IMPLICIT CONJUGATE GRADIENT METHOD AS A SECOND DEGREE METHOD (N.I.C.G.S.D.)

A similar three-term formula analogous to (6.2.6) for $x_{i+1}$, involving $\underline{x}_{i}$ and $\underline{x}_{i-1}$, can be derived as follows:

From (5.4.16) we obtain
and

$$
\begin{equation*}
\underline{x}_{i+1}=\underline{x}_{i}+\tilde{a}_{i} \tilde{\sigma}_{i} \tag{6.3.1}
\end{equation*}
$$

$$
\begin{equation*}
\underline{x}_{i}=\underline{x}_{i-1}+\tilde{a}_{i-1} \tilde{\sigma}_{i-1} \tag{6.3.2}
\end{equation*}
$$

or

$$
\begin{equation*}
\frac{\tilde{\beta}_{i}}{\tilde{a}_{i-1}} \tilde{a}_{i} x_{i}=\frac{\tilde{\beta}_{i}}{\widetilde{a}_{i-1}} \widetilde{a}_{i} x_{i-1}+\tilde{a}_{i} \tilde{\beta}_{i} \tilde{\sigma}_{i-1} . \tag{6.3.3}
\end{equation*}
$$

A combination of (6.3.1), (6.3.3) and (5.4.20) leads to

$$
\begin{equation*}
\underline{x}_{i+1}=\underline{x}_{i-1}+\tilde{\rho}_{i+1}\left(\tilde{\gamma}_{i+1} \underline{r}_{i}^{*}+\underline{x}_{i}-\underline{x}_{i-1}\right), \tag{6.3.4}
\end{equation*}
$$

where the parameters $\tilde{\rho}_{i}, \tilde{\gamma}_{i}$ are given by
and

$$
\begin{align*}
& \tilde{\rho}_{i+1}=1+\frac{\tilde{a}_{i}}{\tilde{a}_{i-1}} \tilde{\beta}_{i}  \tag{6.3.5}\\
& \tilde{\gamma}_{i+1}=\frac{\tilde{a}_{i}}{\widetilde{\rho}_{i+1}} \tag{6.3.6}
\end{align*}
$$

for $i=0,1,2,3, \ldots k-1$, where $k$ is the smallest integer, such that ${\underset{\underline{r}}{k}}^{\tilde{x}^{\prime}}$ (see equations (5.4.12), (5.4.17)).
Next, we derive the analogue formulae of (6.2.12), (6.2.17) by use of the orthogonality relations (2.4.20) and a combination of (5.4.12), (5.4.18), (6.2.9) yields
where

$$
\begin{equation*}
\underline{r}_{i}^{*}=\underline{\tilde{s}}-\tilde{\mathrm{G}}_{\mathrm{i}} \tag{6.3.7}
\end{equation*}
$$

From (6.3.4), (6.3.7) we obtain

$$
\begin{equation*}
\underline{r}_{i+1}^{*}=\underline{r}_{i-1}^{*}+\tilde{\rho}_{i+1}\left(-\tilde{\gamma}_{i+1} \tilde{G}_{i}^{*}+\underline{r}_{i}^{*}-\underline{r}_{i-1}^{*}\right) \tag{6.3.9}
\end{equation*}
$$

and forming the inner product of both sides of (6.3.9) with $\underline{r}_{i}^{*}$ we have

$$
\begin{equation*}
0=\tilde{\rho}_{i+1}\left(-\tilde{\gamma}_{i+1}\left(\underline{r}_{i}^{*}, \tilde{G}_{\underline{i}}^{*}\right)+\left(\underline{r}_{i}^{*}, \underline{r}_{i}^{*}\right)\right) \tag{6.3.10}
\end{equation*}
$$

Then, assuming that $\tilde{\rho}_{i+1} \neq 0$ we get

$$
\begin{equation*}
\tilde{\gamma}_{i+1}=\frac{\left(\underline{r}_{i}^{*}, \underline{r}_{i}^{*}\right)}{\left(\underline{r}_{i}^{*}, \tilde{\mathrm{G}}_{-i}^{*}\right)} \tag{6.3.11}
\end{equation*}
$$

Forming the inner product of both sides of (6.3.9) with $\underline{r}_{\mathrm{i}-1}^{*}$ we have

$$
\begin{equation*}
0=\left(\underline{r}_{i-1}^{*}, \underline{r}_{i-1}^{*}\right)+\tilde{\rho}_{i+1}\left(-\tilde{\gamma}_{i+1}\left(\underline{r}_{i-1}^{*}, \tilde{G}_{-i}^{*}\right)-\left(\underline{r}_{i-1}^{*}, \underline{r}_{i-1}^{*}\right)\right) \tag{6.3.12}
\end{equation*}
$$

or

Also from (6.3.9) we get

$$
\begin{equation*}
\tilde{\rho}_{i+1}=\left[1+\frac{\left(\underline{r}_{i-1}^{*}, \tilde{G r}_{-1}^{*}\right)}{\left(\underline{r}_{i-1}^{*}, \underline{r}_{i-1}^{*}\right)} \tilde{\gamma}_{i+1}\right]^{-1} \tag{6.3.13}
\end{equation*}
$$

$$
\begin{equation*}
\underline{r}_{i}^{*}=\underline{r}_{i-2}^{*}+\tilde{\rho}_{i}\left(-\tilde{\gamma}_{i} \tilde{G}_{-i-1}^{*}+\underline{r}_{i-1}^{*} \underline{r}_{i-2}^{*}\right) \tag{6.3.14}
\end{equation*}
$$

and taking the inner product of both sides with ${\underset{-}{r}}_{i}^{*}$ we obtain

$$
\begin{equation*}
\left(\underline{r}_{i}^{*}, \underline{r}_{i}^{*}\right)=\tilde{\rho}_{i}\left(-\tilde{\gamma}_{i}\left(\underline{r}_{i}^{*}, \widetilde{\operatorname{Gr}}_{i-1}^{*}\right)\right) \tag{6.3.15}
\end{equation*}
$$

or

$$
\begin{equation*}
\left(\underline{r}_{i-1}^{*}, \widetilde{G}_{-i}^{*}\right)=\left(\underline{r}_{i}^{*}, \widetilde{G}_{-1-1}^{*}\right)=-\frac{\left(\underline{r}_{i}^{*}, r_{-1}^{*}\right)}{\widetilde{\rho}_{i} \widetilde{\gamma}_{i}} . \tag{6.3.16}
\end{equation*}
$$

Then, the substitution of (6.3.16) in (6.3.13) leads to the recursive formulae

$$
\begin{equation*}
\tilde{\rho}_{i+1}=\left[1-\frac{\left(r_{i}^{*}, r_{i}^{*}\right)}{\left(r_{i-1}^{*}, r_{i-1}^{*}\right)} \frac{\tilde{\gamma}_{i+1}}{\tilde{\gamma}_{i}} \frac{1}{\tilde{\rho}_{i}}\right]^{-1} \quad \text { for } i=1,2,3, \ldots \tag{6.3.17}
\end{equation*}
$$

with $\tilde{\rho}_{1}=1$, since $\underline{x}_{1}=x_{0}+\tilde{a}_{0} \tilde{r}_{0}$.

The formulae (6.3.11), (6.3.17) can now be used for the generation of the parameters $\widetilde{\rho}_{i}, \widetilde{\gamma}_{i}$.

Alternatively, using the relations (5.4.3), (5.4.8), (6.2.9) we obtain
where

$$
\begin{equation*}
\underline{\mathrm{r}}_{\mathrm{i}}^{+}=\underline{\mathrm{s}}^{+}-\mathrm{G}^{+} \underline{\mathrm{x}}_{\mathrm{i}} \tag{6.3.18}
\end{equation*}
$$

Proceeding in a similar manner as above, we obtain the following formulae for $\rho_{i}^{+}, \gamma_{i}^{+}$:

$$
\begin{equation*}
\gamma_{i+1}^{+}=\frac{\left(\underline{r}_{i}^{+}, \underline{r}_{i}^{+}\right)}{\left(\underline{r}_{i}^{+}, G^{+} \underline{r}_{i}^{+}\right)} \tag{6.3.20}
\end{equation*}
$$

and

$$
\begin{equation*}
\underline{s}^{+}=G^{+} \underline{s} \text { and } G^{+}=\left(D_{S} T_{s}^{\prime} T_{s} D_{s}\right)^{-1} A \tag{6.3.19}
\end{equation*}
$$

$$
\begin{equation*}
\rho_{i+1}^{+}=\left[1-\frac{\left(\underline{r}_{i}^{+}, \underline{r}_{i}^{+}\right)}{\left(\underline{r}_{i-1}^{+}, \underline{r}_{i-1}^{+}\right)} \cdot \frac{r_{i+1}^{+}}{r_{i}^{+}} \cdot \frac{1}{\rho_{i}^{+}}\right]^{-1} \tag{6.3.21}
\end{equation*}
$$

It is easily seen that the values of parameters $\rho_{i}^{+}, \gamma_{i}^{+}$given by $(6.3 .21),(6.3 .20)$ are easily obtained and when $G^{+}$has the form

$$
\begin{equation*}
\mathrm{G}^{+}=\left(\mathrm{T}_{\mathrm{S}}^{\prime} \mathrm{T}_{\mathrm{S}}\right)^{-1} \mathrm{~T}^{\prime} \mathrm{T} \tag{6.3.22}
\end{equation*}
$$

which can be shown to be related to the approximate factorized form of A discussed earlier in Chapter 4.

## A Factor Affecting the Stability Conditions

It can be shown, [32] (p.420), that when the ratio $a_{i} / a_{i-1}$ is large (see equation (6.2.7)) the disturbance of the orthogonality relations in the C.G. method will be greatest. Consequently, the larger the ratio $\tilde{a}_{i} / \widetilde{a}_{i-1}$ (see equation (6.3.5)) the more rapidly rounding-off error will accumulate.

Since the scalars $a_{i},[32](p .421)$, and $\tilde{a}_{i}$ lie in the range $\left(\frac{1}{M}, \frac{1}{m}\right)$, where $m, M$ are the extreme eigenvalues of the coefficient matrix $A$, it follows that $\mathrm{M} / \mathrm{m}$ is an upper bound of the ratios $a_{i} / a_{i-1}$ and $\tilde{a}_{i} / \tilde{a}_{i-1}$ which directly affect the sensitivity to
round-off errors of the C.G. and N.I.C.G. processes respectively.
Consequently for any pre-assigned set of values of the scalars $a_{i}$ and $\tilde{a}_{i}$, such that $\left(a_{i} / a_{i-1}\right)<1$ and $\left(\tilde{a}_{i} / \tilde{a}_{i-1}\right)<1$ respectively, the stability of the method with respect to the growth of rounding-off errors can be satisfied (i.e., the algorithms are stable).

## Computational Results

Proceeding in an analogous way to the two dimensional case, we can now extend the C.G.S.D. and N.I.C.G.S.D. methods in three space dimensions, obtaining the C.G.3D.S.D. and N.I.C.G.3D.S.D. methods respectively.

Numerical results for C.G.S.D., N.I.C.G.S.D. and C.G.3D.S.D., N.I.C.G.3DS.D. methods for the model problems are given in Tables 6.1,6.2, including the values of the parameters $\rho_{n}, \gamma_{n}$ at the end of the $n^{\text {th }}$ iteration. For comparative purposes two cases of the N.I.C.G.S.D. $(r=1, r=4)$ and N.I.C.G.3D.S.D. $\left(r_{i}=r_{2}=1, r_{1}=r_{2}=4\right)$ methods have been considered during the experiments. The error was computed as in section 5.4 and the same termination criterion and initial guess vector used. The solution vector was chosen to be:

$$
\underset{i \neq m+1}{x_{i}}=0, \quad i \varepsilon[1, N] \text { with } \quad x_{m+1}=1
$$

and the right hand side vector of equation (2.4.1) was obtained as the product of the solution vector with the coefficient matrix $A$ of (2.4.1).

In order to observe the behaviour of the parameters $\rho_{n}, \gamma_{n}$ of the C.G.S.D., C.G.3D.S.D. methods and $\widetilde{\rho}_{n}, \widetilde{\gamma}_{n}\left(\rho_{n}^{+}, \gamma_{n}^{+}\right)$of the N.I.C.G.S.D., N.I.C.G.3D.S.D. methods, as the grid size of the model problems is increased, comparative figures are given in Figures 6.2-6.4, for both two and three dimensional cases.

Finally, a comparison of the results of the N.I.C.G.S.D., N.I.C.G.3D.S.D methods, as given in Tables $6.1,6.2$, with those of N.I.C.G., N.I.C.G.3D (case II), as given in Tables 5.3,5.4, shows that inspite of the increase in the computational work involved using second order methods, there is no substantial gain in the number of iterations of these methods against those of first order.

|  |  | C.G.S.D. |  |  | N.I.C.G.S.D. (r=1) |  |  | N.I.C.G.S.D. (r=4) |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\varepsilon=10^{-6}$ | Mesh Size | ${ }^{\circ} \mathrm{n}$ | $\gamma_{n}$ | No. of Iters. n | $\tilde{\rho}_{n}$ | $\tilde{\gamma}_{n}$ | n | $\tilde{\rho}_{n}$ | $\widetilde{\gamma}_{11}$ | n |
|  | 5 | 1.069053 | 0.298162 | 8 | 1.007466 | 0.287808 | 5 | 1.000000 | 0.275916 | 1* |
|  | 10 | 1.099267 | 0.223776 | 24 | 1.027916 | 0.294186 | 9 | 1.024744 | 0.307372 | 7 |
|  | 15 | 1.300436 | 0.264003 | 37 | 1.029364 | 0.369555 | 12 | 1.034481 | 0.332620 | 7 |
|  | 20 | 1.506623 | 0.249149 | 48 | 1.112857 | 0.358697 | 15 | 1.050707 | 0.360515 | 8 |
|  | 30 | 1.601015 | 0.259928 | 66 | 1.194302 | 0.422701 | 20 | 1.117637 | 0.390249 | 10 |
|  | 40 | 1.866241 | 0.243659 | 79 | 1.688252 | 0.508812 | 23 | 1.169254 | 0.398259 | 13 |
| $\varepsilon=10^{-8}$ | 5 | 1.069053 | 0.298162 | $\sim 8$ | 1.039101 | 0.319471 | 8 | 1.000000 | 0.275916 | 1* |
|  | 10 | 1.029541 | 0.249137 | 27 | 1.005562 | 0.297932 | 11 | 1.018946 | 0.305773 | 9 |
|  | 15 | 1.239204 | 0.263739 | 44 | 1.067375 | 0.351895 | 15 | 1.034711 | 0.331232 | 10 |
|  | 20 | 1.490244 | 0.267305 | 57 | 1.061609 | 0.357588 | 18 | 1.048635 | 0.329002 | 12 |
|  | 30 | 1.481521 | 0.247052 | 82 | 1.112554 | 0.398684 | 25 | 1.094541 | 0.378938 | 14 |
|  | 40 | 1.646100 | 0.253540 | 106 | 1.211116 | 0.470903 | 31 | 1.109891 | 0.379627 | 16 |
| $\varepsilon=10^{-10}$ | 5 | 1.069053 | 0.298162 | $\sim 8$ | 1.004009 | 0.291505 | 10 | 1.000000 | 0.275916 | 1* |
|  | 10 | 1.054674 | 0.275596 | 30 | 1.315973 | 0.407352 | 14 | 1.008881 | 0.309039 | 11 |
|  | 15 | 1.155045 | 0.253499 | 49 | 1.017133 | 0.329981 | 17 | 1.020284 | 0.312923 | 12 |
|  | 20 | 1.264603 | 0.249825 | 66 | 1.033903 | 0.367417 | 21 | 1.042171 | 0.356372 | 14 |
|  | 30 | 1.530361 | 0.245857 | 97 | 1.189517 | 0.460586 | 30 | 1.081147 | 0.378157 | 18 |
|  | 40 | 1.771570 | 0.246147 | 125 | 1.191915 | 0.449522 | 38 | 1.281877 | 0.434722 | 22 |

## TABLE 6.1

Parameters $\rho_{n}, \gamma_{n}, \tilde{\rho}_{n}, \tilde{\gamma}_{n}$ and work required to reduce the error to $1 E-6,1 E-8,1 E-10$
The resulting sparse matrices for the above
considered mesh sizes are of order $16,81,196,361,841,1421$ respectively.
(*) For $r=m$-1 the method becomes a direct one.

|  |  | C.G.3D.S.D. |  |  | N.I.C.G.3D.S.D. $\left(\mathrm{r}_{1}=\mathrm{r}_{2}=1\right)$ |  |  | N.I.C.G.3D.S.D. ( $\mathrm{r}_{1}=\mathrm{r}_{2}=4$ ) |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\varepsilon=10^{-6}$ | $\begin{aligned} & \text { Mesh } \\ & \text { Size } \\ & h^{-1} \end{aligned}$ | ${ }^{\circ} \mathrm{n}$ | $\gamma_{n}$ | No. of Iters. | $\tilde{\rho}_{n}$ | $\tilde{\gamma}_{n}$ | n | $\widetilde{\rho}_{n}$ | $\tilde{\gamma}_{n}$ | n |
|  | 5 | 1.098210 | 0.161351 | 17 | 1.00534510 | 0.18877492 | 6 | 1.00649336 | 0.18959059 | 5 |
|  | 7 | 1.349571 | 0.175824 | 24 | 1.01636366 | 0.20085404 | 8 | 1.02070535 | 0.21133891 | 7 |
|  | 9 | 1.383097 | 0.164590 | 28 | 1.04634084 | 0.21988940 | 9 | 1.04019646 | 0.22447190 | 8 |
|  | 11 | 1.387208 | 0.164679 | 34 | 1.06898420 | 0.23945461 | 11 | 1.06532638 | 0.23534745 | 9 |
| $\varepsilon=10^{-8}$ | 5 | 1.106920 | 0.133364 | 21 | 1.01005623 | 0.19751844 | 8 | 1.00659299 | 0.19161870 | 7 |
|  | 7 | 1.281821 | 0.164165 | 31 | 1.02800411 | 0.20962749 | 11 | 1.02042768 | 0.20104534 | 9 |
|  | 9 | 1.357018 | 0.156382 | 38 | 1.09045951 | 0.25259443 | 13 | 1.03682028 | 0.21623437 | 11 |
|  | 11 | 1.565861 | 0.184152 | 45 | 1.14026254 | 0.26781433 | 15 | 1.06795429 | 0.22639343 | 13 |
| $\varepsilon=10^{-10}$ | 5 | 1.089927 | 0.201645 | 24 | 1.11618224 | 0.21464544 | 11 | 1.00251448 | 0.18345553 | 9 |
|  | 7 | 1.192356 | 0.170759 | 36 | 1.02040246 | 0.21501709 | 13 | 1.00778659 | 0.19175601 | 11 |
|  | 9 | 1.243097 | 0.163397 | 46 | 1.03680503 | 0.20917582 | 16 | 1.01369925 | 0.20019031 | 13 |
|  | 11 | 1.303649 | 0.159903 | 55 | 1.12175850 | 0.23319397 | 18 | 1.05927287 | 0.24077135 | 16 |

[^2]



FIf $k \rho$ : Rohavinur of parameters $\ell_{n}, \gamma_{n}$ of CG3DSD method Non odel problem.



FIG. 6.3 : Behaviour of parameters $\tilde{e}_{n}, \tilde{\gamma}_{n}$ of NICG3DSD method - $\left.T_{s}^{\prime} T_{s} D_{3}\right]$ for the 3D model problem.


neters $e_{n}^{+} \gamma_{n}^{+}$of NICG3DSD method

### 6.4 A SEMI-EMPIRICAL PROCEDURE FOR THE NORMALIZED CONJUGATE GRADIENT SECOND DEGREE METHODS

A thorough examination of the values of the parameters $\tilde{\rho}_{i}, \tilde{\gamma}_{i}$, $i \varepsilon[1, n]$ (see Figures $6.5,6.6$ ) reveals that after a certain number of iterations the oscillating in above parameters subsides and the values remain almost stable.

Given that the values of parameters $\tilde{\rho}_{n}, \tilde{\gamma}_{n}$ are known (see Tables $6.1,6.2$ ), we assume now that after a certain number $k_{A}$ of iterations the iterative scheme proceeds with the values of parameters $\tilde{\rho}_{i}, \tilde{\gamma}_{i}$ constant and equal to $\widetilde{\rho}_{n}, \tilde{\gamma}_{n}$ respectively, i.e., $\tilde{\rho}_{i}=\tilde{\rho}_{n}, \tilde{\gamma}_{i}=\tilde{\gamma}_{n}$, $i=k, k+1, k+2, \ldots($ Case $A)$.

Since the previous outlined procedure pre-assumes the knowledge of parameters $\tilde{\rho}_{n}, \tilde{\gamma}_{n}$, we consider now a different approach, in which the values of $\tilde{\rho}_{i}, \tilde{\gamma}_{i}$ are retained constant after the first $k_{B}$-iterations, throughout the iterative process, and are equal to $\tilde{\rho}_{k_{B}}, \widetilde{\gamma}_{k_{B}}$ respectively, i.e., $\quad \tilde{\rho}_{i}=\tilde{\rho}_{k_{B}}, \tilde{\gamma}_{i}=\tilde{\gamma}_{k_{B}}, i=k, k+1, k+2, \ldots$ (Case B).

Since the formulation of the scalar products into the inner loop of the methods is an important time-consuming operation, substantial gains in the computational work involved can be obtained if these scalar products are not calculated at all after ( $n-k_{A}$ ) or ( $n-k_{B}$ ) iterations respectively.

With this in mind, we derive a semi-Empirical Procedure in two and three space dimensions (henceforth called N.I.C.G.S.D.S.E.P. and N.I.C.G.3D.S.D.-S.E.P. methods respectively) giving a tentative proposal for the numbers $k_{A}, k_{B}$ of initial iterations required for the parameters $\tilde{\rho}_{i}, \tilde{\gamma}_{i}$ to settle down, provided that the number of iterations of N.I.C.G.S.D.-S.E.P., N.I.C.G.S.D. and N.I.C.G.3D.S.D.-S.E.P., N.I.C.G.3D.S.D. methods is about the same.

At this point we indicate the following case worth further investigation. The determination of $k_{A}$ and $k_{B}$ such that the obtained number of iterations of N.I.C.G.S.D.-S.E.P., N.I.C.G.3D.S.D.-S.E.P. methods will be the minimum.

Experimental results for case $A$ and case $B$ are given in Tables 6.3, 6.4.




FIG. 6.6 : Behaviour of parameters $\tilde{e}_{1}, \tilde{\gamma}_{i}$ of NICG3DSD [A $\left.\simeq D_{s} T_{s}^{\prime} T_{s} D_{s}\right]$ method annlied to the lanlacian matrix of order $N=1000$ the 3D model problem.



[^3]Finally, we state that the determination of the optimum size of the subset of parameters is an interesting and open question and further work on this topic is planned at a future date.
6.5 DERIVATION OF STANDARD NORMALIZED IMPLICIT METHODS AND STANDARD IMPLICIT METHODS
6.5.a Derivation of standard Normalized Implicit Methods

We consider the approximate factorization (4.3.1) applied to the linear system (2.1.3) to yield the iteration

$$
\begin{array}{r}
\left(D_{s} T_{s}^{\prime} T_{S} D_{s}\right) \underline{u}_{i+1}=\left(D_{S}^{T} S_{s}^{\prime} D_{s}\right) \underline{u}_{i}+a\left(\underline{s}-A \underline{u}_{i}\right), \\
i=0,1,2, \ldots \tag{6.5.1}
\end{array}
$$

where $a$ is a predetermined acceleration parameter.
It can be easily seen from (2.3.13), (2.3.25), (2.3.26) and (6.5.1) that the error vector $\widetilde{\tilde{e}}_{i}$ satisfies:

$$
\begin{equation*}
\underline{\tilde{e}}_{i+1}=\mathrm{E} \cdot \widetilde{\underline{e}}_{\mathrm{i}} \tag{6.5.2}
\end{equation*}
$$

where $E$ is the error propagation matrix, given by

$$
\begin{equation*}
E=I-a\left(D_{s} T_{s}^{\prime} T_{s} D_{s}\right)^{-1} A \tag{6.5.3}
\end{equation*}
$$

The iteration (6.5.1) can be alternatively written as

$$
\begin{equation*}
\underline{u}_{i+1}=\underline{\underline{u}}_{i}+a \cdot \underline{r}_{i}^{*} \tag{6.5.4}
\end{equation*}
$$

where

$$
\begin{equation*}
\underline{\tilde{u}}_{i}=D_{s} \cdot \underline{u}_{i} \quad \text { and } \quad \underline{r}_{i}^{*}=\left(D_{S}^{T} T_{S}^{\prime} T^{-1} \underline{r}_{i}\right. \tag{6.5.5}
\end{equation*}
$$

or equivalently as

$$
\begin{equation*}
\mathrm{T}_{\mathrm{S}}^{\prime} \mathrm{T} \delta \underline{\mathrm{u}}_{\mathrm{i}+1}=\mathrm{a} \cdot \underline{\underline{r}}_{\mathrm{i}} \tag{6.5.6}
\end{equation*}
$$

$$
\begin{equation*}
\delta \underline{u}_{i+1}=\underline{\tilde{u}}_{i+1}-\underline{\tilde{u}}_{i} \quad \text { and } \quad{\underset{-}{r}}_{i}=D_{s}^{-1} \cdot \underline{r}_{i} \tag{6.5.7}
\end{equation*}
$$

where

The iterative scheme (6.5.4) or (6.5.6), is an analogue of the Simultaneous Displacement Iteration given by (2.3.13) and defined as the 'Normalized Implicit Simultaneous Displacement' (N.I.S.D.) method.

We consider now the following linear stationary iteration of second degree

$$
\begin{equation*}
\left(D_{s} T_{s}^{\prime} T_{s} D_{s}\right) \underline{u}_{i+1}=\left(D_{s} T_{s}^{\prime} T_{s} D_{s}\right) \underline{u}_{i}+a\left(\underline{s}-A \underline{u}_{i}\right)+\beta\left(\underline{u}_{i}-\underline{u}_{i-1}\right) \tag{6.5.8}
\end{equation*}
$$

which can be written as

$$
\begin{equation*}
\tilde{\underline{u}}_{i+1}=\underline{\tilde{u}}_{i}+\underline{a r}_{i}^{*}+\left(T_{s}^{\prime} T_{s}\right)^{-1} \tilde{\beta}\left(\tilde{u}_{i}-\tilde{u}_{i-1}\right), \tag{6.5.9}
\end{equation*}
$$

where ${\underset{\underline{u}}{i+1}}, \underline{r}_{i}^{*}$ are given by (6.5.5), or equivalently,

$$
\begin{equation*}
\mathrm{T}_{\mathrm{S}}^{\prime} \mathrm{T}_{\mathrm{S}} \delta \underline{\underline{u}}_{\mathrm{i}+1}=\mathrm{a} \cdot \underline{\tilde{r}}_{\mathrm{i}}+\tilde{\beta} \cdot \delta \underline{\underline{u}}_{\mathrm{i}} \tag{6.5.10}
\end{equation*}
$$

where $\delta \underline{\underline{u}}_{i+1}, \tilde{\underline{r}}_{i}$ are given by (6.5.7) and $a, \tilde{\beta}$ are preconditioned acceleration parameters.

The iterative scheme (6.5.9) or (6.5.10) is an analogue of the second order Richardson's method given by (2.3.50) and is defined as the 'Normalized Implicit Richardson's' (N.I.R.) method. The parameters $a, \tilde{\beta}$ remain constant throughout the iteration and are chosen to provide maximum convergence to the solution (see relationships (2.3.58)).

Consider now the iterative scheme (6.5.8), with the parameter sequences $a_{n}, \beta_{n}$ instead of the fixed value parameters $a, \beta$. Then the 'Normalized Implicit Chebychev' (N.I.Ch.) method, analogue of Chebychev second order method given by (2.3.60), is a non-stationary second degree iteration defined by

$$
\begin{equation*}
\underline{\tilde{u}}_{i+1}=\underline{\tilde{u}}_{i}+a_{n} \underline{r}_{i}^{*}+\left(T_{s}^{\prime}{ }_{s}\right)^{-1} \tilde{\beta}_{n}\left(\tilde{u}_{i}-\underline{\tilde{u}}_{i-1}\right) \tag{6.5.11}
\end{equation*}
$$

where $\tilde{\underline{u}}_{i+1}, \underline{r}_{i}^{*}$ are given by (6.5.5), or equivalently,

$$
\begin{equation*}
\left(T_{s}^{\prime} T_{s}\right) \delta \underline{u}_{i+1}=a_{n} \tilde{\underline{r}}_{i}+\tilde{\beta}_{n} \delta \underline{\underline{u}}_{i}, \tag{6.5.12}
\end{equation*}
$$

where $\delta \underline{u}_{i+1}, \underline{\tilde{r}}_{i}$ are given by (6.5.7) and $a_{n}, \tilde{\beta}_{n}$ are the sequence of acceleration parameters which vary with each iterative step (see relationships (2.3.61)).

The analysis and convergence conditions of the derived N.I.S.D., N.I.R., N.I.Ch. methods can be easily obtained from section 2.3, using properly the corresponding modified iteration matrix for each of the normalized implicit methods. It can be shown that since the parameter sequences are similar to those given in section 2.3 then the N.I.Ch. method is less sensitive to round-off errors than the N.I.R. method.

## 6.5.b Derivation of Standard Implicit Methods

Several iterative procedures for solving non-symmetric linear systems have appeared in the literature, [25], [60], [36], [16], [40].

We now introduce a class of iterative procedures for solving numerically the linear system (2.7.17) associated with the P.D.t. (2.5.1). We consider the approximate factorization (4.2.1) applied to the above linear system to yield the following iterative method (Simultaneous Displacement)

$$
\begin{equation*}
L_{s} U_{S} \delta \underline{u}_{i+1}=a \underline{r}_{i} \tag{6.5.13}
\end{equation*}
$$

where $L_{S}$ and $U_{s}$ are the known triangular matrices given by (4.2.2), a is a predetermined acceleration parameter, $\underline{r}_{i}=\underline{s}-\underline{A}_{i}$ and $\delta \underline{u}_{i+1}=\underline{u}_{i+1}-\underline{u}_{i}$. For a fixed choice of parameter a and an initial guess of the solution $\underline{u}_{0}$, a sequence of approximate solutions $\underline{u}_{1}, \underline{u}_{2}, \ldots, \underline{u}_{k}$ can be obtained using the ALUBOT algorithm.

Similarly the following second order iterative procedure can be formulated for the Richardson and Chebychev methods respectively,
and

$$
\begin{align*}
& L_{s} U_{s} \delta \underline{u}_{i+1}=\operatorname{ar}_{i}+\beta \delta \underline{u}_{i}  \tag{6.5.14}\\
& L_{s} U_{s} \delta \underline{u}_{i+1}=a_{n} \underline{r}_{i}+\beta_{n} \delta \underline{u}_{i} . \tag{6.5.15}
\end{align*}
$$

Note that for the parabolic problem of section 2.5 since the solution only differs slightly from plane to plane then the obtained solution at the $\ell^{\text {th }}$ time step is a good initial approximation to commence the iterative scheme $(6.5 .13)$ at the $(\ell+1)^{\text {th }}$ time step.

## Remark

Since the problem under investigation has been solved in many different ways a complete knowledge of the eigenspectrum of the iteration matrix was known.

Hence, for the Chebychev method, guesses which were close to the actual values were used.

For problems without this a priori knowledge then an attempt to calculate the spectral radius of $\left(\mathrm{L}_{\mathrm{s}} \mathrm{U}_{\mathrm{S}}\right)^{-1} \mathrm{LU}$ (or the spectral radius of ( $D_{S} T_{s} T_{s} D_{s}$ ) ${ }^{-1} D_{D T}$ 'TD for the case of section 6.5.a) must be made.

Computational Work and Numerical Experiments
Numerical results for the Normalized Implicit methods i.e., N.I.S.D., N.I.R. and N.I.Ch. methods, on the 2D-model problem and the experimental estimation of the optimal iterative parameters involved, are given in Table 6.5.

For comparative reasons the numerical results obtained applying the ALUBOT-2D algorithm in conjunction with the Simultaneous Displacement, Second Order Richardson and Chebychev methods, on the 2D-model problem, are presented and the values of the optimal iterative parameters involved are experimentally estimated.

The initial vector was chosen to be the zero vector and the solution vector was chosen: $x_{i}=0, i \varepsilon[1, N]$ and $x_{m+1}=1$, where $m$ is the $i \neq \mathrm{m}+1$
bandwidth of the coefficient matrix of linear systems (2.4.1). The right hand side vector of (2.4.1) was obtained as the product of the solution vector $\underline{x}$ with the coefficient matrix A of (2.4.1). The error was computed as
error $=\left\|\frac{x_{i}^{(k+1)}-x_{i}^{(k)}}{x_{i}^{(k)}+1}\right\|_{\infty}$
and the iteration process terminated when the error was reduced to $10^{-6}$.

The cases 1 and 2 of the Normalized Implicit methods are referred to as the approximate factorizations $A \approx D T T_{S} T_{S} D$ and $A \approx D_{S}^{T} T_{S}^{T} D_{S}$ respectively.

Note that the calculation of the "true" residual i.e., $\underline{r}_{i}=\underline{S}_{i}-\underline{A}_{i}$, into the inner loop of the Normalized Implicit (N.I.S.D., N.I.R., N.I.Ch.) methods, requires the normalization of the solution vector $\underline{u}_{i}$ at each iterative step.

Consequently, the above methods as far as the computational work is concerned, are unfavourable when compared with the N.I.C.G. procedure, since in the latter method the residuals are computed recursively (see section 5.4 ).

| Method | $\begin{gathered} \text { Mesh } \\ \text { Size } \\ h^{-1} \\ \hline \end{gathered}$ | Normalized Method (NOBAR-2D Algorithm) $\qquad$ |  | Normalized Method (NOBAR-2D Algorithm) Case 2 |  | LU-factorization (ALUBOT-2D Algorithm) |  | $\begin{aligned} & \text { No } \\ & \text { fit } \end{aligned}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Parameters: $\widetilde{\beta}$ | No. of Iters | $\text { a } \quad \text { Parameters: } \underset{\tilde{\tilde{B}}}{ }$ | $\begin{aligned} & \text { No.of } \\ & \text { liters } \end{aligned}$ | a Par | ameters: <br> $\beta$ |  |
| Simultaneous Displacement | 10 | 1.20 | 11 | 1.13 0 | 10 | 1.07 | 0 |  |
|  | 15 | 1.320 | 16 | 1.20 0 | 14 | 0.98 | 0 |  |
|  | 20 | 1.400 | 21 | 1.260 | 17 | 0.94 | 0 |  |
|  | 30 | 1.50 0 | 126 | 1.340 | 21 | 0.75 | 0 |  |
| 2nd Order Richardson | 10 | $1.13 \quad 0.008$ | 7 | 1.110 .004 | 7 | 1.04 | 0.025 |  |
|  | 15 | 1.24 0.015 | 10 | 1.150 .008 | 9 | 1.01 | 0.015 |  |
|  | 20 | $1.31 \quad 0.022$ | 12 | 1.220 .010 | 10 | 0.98 | 0.011 |  |
|  | 30 | 1.39 0.030 | 14 | $1.29 \quad 0.014$ | 13 | 0.80 | 0.010 |  |
|  |  | $\begin{array}{ll} a_{n} \text { Parameters: } & \tilde{\beta}_{n} \\ \hline \end{array}$ |  | $\mathrm{a}_{\mathrm{n}} \quad$ Parameters $_{\dot{\mathrm{B}}_{\mathrm{n}}}$ |  | $\mathrm{a}_{\mathrm{n}} \quad$ Par |  |  |
| 2nd Order Chebychev | 10 | $\begin{aligned} & \left\{\begin{aligned} 1.139 & \downarrow_{1.130} \end{aligned}\right\}\left\{\begin{array}{ll} 0.016 & \downarrow_{0.008} \end{array}\right\} \\ & \left\{\begin{aligned} 1.263 & \iota_{1.245} \end{aligned}\right\}\left\{\begin{array}{ll} 0.030 & \downarrow_{0.015} \end{array}\right\} \\ & \left\{\begin{aligned} 1.338 & \iota_{1.310} \end{aligned}\right\}\left\{\begin{array}{ll} 0.044 & \iota_{0.022} \end{array}\right\} \\ & \left\{\begin{array}{ll}  & \downarrow_{1.365} \end{array}\right\}\left\{\begin{array}{ll} 0.060 \\ & \iota_{0.030} \end{array}\right\} \end{aligned}$ | 810 | $\begin{aligned} & \left\{\begin{array}{ll} 1.114 & \downarrow_{1.110} \end{array}\right\}\left\{\begin{array}{ll} 0.008 & \downarrow_{0} \\ & \downarrow_{0.004} \end{array}\right\} \\ & \left\{\begin{array}{ll} 1.159 & \downarrow_{1.150} \end{array}\right\}\left\{\begin{array}{ll} 0.016 & \iota_{0.008} \end{array}\right\} \end{aligned}$ | 810 | $\left\{\begin{array}{ll} 1.065 & \downarrow_{1.040} \end{array}\right\}\left\{\begin{array}{ll} 0.050 & \downarrow_{0.025} \\ & \end{array}\right\}$ |  |  |
|  | 15 |  |  |  |  | $\begin{cases}1.020 & l_{1.005}\end{cases}$ | $\left\{\begin{array}{ll}0.030 & \\ & \pm 0.015\end{array}\right\}$ |  |
|  | 20 |  | 12 | $\left\{\begin{array}{ll}1.232 & \\ & \downarrow_{1.220}\end{array}\right\}\left\{\begin{array}{ll}0.020 & \downarrow_{0.010}\end{array}\right\}$ | 10 | $\begin{cases}0.986 & \\ & t_{0.975}\end{cases}$ | $\left\{\begin{array}{ll}0.022 & 10.011\end{array}\right\}$ |  |
|  | 30 |  | 16 | $\left\{\begin{array}{ll}1.308 & \downarrow_{1.290}\end{array}\right\}\left\{\begin{array}{ll}0.028 & t_{0.014}\end{array}\right\}$ | 14 | $\left(\begin{array}{ll}0.808 \\ & \downarrow 0.800 \\ \hline\end{array}\right.$ | $\left\{\begin{array}{ll}0.020 & \\ & 0.010\end{array}\right\}$ |  |

TABLE 6.5
The performance of Normalized Implicit (N.I.S.D., N.I.R., N.I.Ch.) methods and the application of ALUBOT-2D algorithm to Simultaneous Displacement, 2nd order Richardson and Chebychev methods for solving the 2D-model problem. The resulting sparse matrices for the above considered mesh sizes are of order $81,196,361,841$, respectively.

Chapter 7

THREE DIMENSIONAL ELLIPTIC AND PARABOLIC PROBLEMS

### 7.1 INTRODUCTION

In this chapter, the application of the algorithmic procedures of Chapter 3 to three dimensional Elliptic and Parabolic problems is presented with accompanying experimental results to verify the applicability and usefulness of the methods.

## SECTION A: THE ELLIPTIC 3D-PROBLEM

### 7.2 STATEMENT OF THE PROBLEM AND FORMATION OF THE DIFFERENCE EQUATIONS

We consider the solution of the Laplace's equation in three
variables:

$$
\begin{equation*}
\frac{\partial^{2} U}{\partial x^{2}}+\frac{\partial^{2} U}{\partial y^{2}}+\frac{\partial^{2} U}{\partial z^{2}}=0, \quad(x, y, z) \varepsilon R, \tag{7.2.1}
\end{equation*}
$$

where $R$ is the unit cube as given in (2.8.3).
The following Dirichlet boundary conditions are considered in connection with the approximate solution of equation (7.2.1):

$$
\begin{align*}
& U(0, y, z)=U(1, y, z)=0, \quad 0 \leqslant y, z \leqslant 1  \tag{7.2.2}\\
& U(x, 0, z)=U(x, 1, z)=\sin \pi x \cdot \sin \pi z, \quad 0 \leqslant x, z \leqslant 1  \tag{7.2.3}\\
& U(x, y, 0)=U(x, y, 1)=0, \quad 0 \leqslant x, y \leqslant 1 . \tag{7.2.4}
\end{align*}
$$

The unit cube is now covered by a three dimensional grid, defined by:

$$
\begin{equation*}
\widetilde{R}_{h}=\left\{\left(i h_{x}, j h_{y}, k h_{z}\right): 0 \leqslant i, j, k \leqslant M\right\} \tag{7.2.5}
\end{equation*}
$$

where $h_{x}, h_{y}, h_{z}$ are the lengths in the $X, Y$ and $Z$ directions respectively. j

$$
\text { Let } x=i h_{x}, y=i h_{y}, z=k_{z} \text { be the co-ordinates of }(M-1)^{3}
$$

internal grid points and assume that $u_{i, j, k}$ denotes $u\left(i h_{x}, j h_{y}, k h_{z}\right)$.
The fully implicit difference approximations to the partial
derivatives of equation (7.2.1) i.e.,

$$
\begin{equation*}
\left[\frac{\partial^{2} U}{\partial x^{2}}\right]_{(i, j, k)}^{(n+1)} \simeq \frac{u_{i-1, j, k}^{(n+1)} u_{i, j, k}^{(n+1)}+u_{i+1, j, k}^{(n+1)}}{h_{x}^{2}} \tag{7.2.6}
\end{equation*}
$$

and analogous expressions for $\frac{\partial^{2} U}{\partial y^{2}}, \frac{\partial^{2} U}{\partial z^{2}}$ are used to derive the following finite difference analogue to (7.2.1) at the point $\left(i h_{x}, j h_{y}, k h_{z}\right)$

where the local truncation error $\mathrm{R}_{\mathrm{L}}$ of (7.2.7), assuming that only the second order differences are retained is given by

$$
\begin{equation*}
R_{L}=\frac{-1}{12}\left[h_{x}^{2} \frac{\partial^{4} U_{1}^{4}}{\partial x^{4}}+h_{y}^{2} \frac{\partial^{4} U^{4}}{\partial y^{4}} h_{z}^{2} \frac{\partial^{4} U}{\partial z^{4}}\right]_{(i, j, k)}^{(n+1)} . \tag{7.2.8}
\end{equation*}
$$

Ordering the ( $\mathrm{M}-1)^{3}$ internal grid points of $\widetilde{R}_{h}$ with increasing values of $j$, then $i$, then $k$ (see Figure 2.4) the following system of equations can be obtained in matrix notation:

$$
\begin{equation*}
\Omega \cdot \underline{u}^{(\mathrm{n}+1)}=\underline{\mathrm{s}} \tag{7.2.9}
\end{equation*}
$$

where $\Omega$ is a real, square, symmetric, seven-diagonal, sparse matrix of the following form:

where $\Gamma$ is the following diagonal matrix of order $m^{2}=(M-1)^{2}$

$$
\begin{equation*}
\Gamma=\operatorname{diag}\left\{\frac{1}{h_{z}^{2}}, \frac{1}{h_{z}^{2}}, \ldots \ldots, \frac{1}{h_{z}^{2}}\right\} \tag{7.2.11}
\end{equation*}
$$

and $A_{i}, i \varepsilon[1, m]$ are matrices of order $m^{2}$, given by:
where $\Delta$ is the following diagonal matrix of order $m$.

$$
\begin{equation*}
\Delta=\operatorname{diag}\left\{\frac{1}{h_{y}^{2}}, \frac{1}{h_{y}^{2}}, \ldots \ldots, \frac{1}{h_{y}^{2}}\right\} \tag{7.2.12}
\end{equation*}
$$

and $B_{i}$, $i \varepsilon\left[1, m^{2}\right]$ are matrices of order $m$, given by:
where $C=2\left[\frac{1}{h_{x}^{2}}+\frac{1}{h_{y}^{2}}+\frac{1}{h_{z}^{2}}\right]$ and $\theta=\frac{1}{h_{x}^{2}}$.
The solution vector $\underline{u}^{(n+1)}$ and the right hand side vector s of the system (7.2.9) are ( $m^{3} \times 1$ ) column vectors. Note that the components of the latter vector depend on the boundary values of $u(x, y, z)$ at the grid points on the boundary planes of the unit cube.

### 7.3 COMPUTATIONAL RESULTS

The analytic solution of the problem can be verified by separation of variables and is given by

$$
\begin{equation*}
U(x, y, z)=\operatorname{sech} \frac{\pi}{\sqrt{2}} \cdot \sin \pi x \cdot \cosh \left\{\sqrt{2} \pi\left(y-\frac{1}{2}\right)\right\} \cdot \sin \pi z . \tag{7.3.1}
\end{equation*}
$$

Values of $U$ calculated from the theoretical solution (7.3.1) are given in Tables 7.1, 7.2, 7.3.

For a numerical solution, the implicit finite difference scheme defined in (7.2.7), was used as an approximation to the original Elliptic P.D.E. (7.2.1). Since the coefficient matrix A in (7.2.9) is positive definite, diagonally dominant and sevendiagonal, the resulting system (7.2.9) was solved using the LUBOI' 3 D Algorithm.

Numerical results are presented in the accompanying Tables 7.1,7.2,7.3, where the values of the approximate and analytic solution at the $(M-1)^{3}$ internal mesh points are given in groups of five decimal figures assuming that $h_{x}=h_{y}=h_{z}=\frac{1}{M}$.

Since the solution is symmetric:
(i) about the planes which pass through the points $x=0.5$, $y=0.5, z=0.5$ and are parallel to the co-ordinate planes $(Y, Z),(X, Z),(X, Y)$ respectively and
(ii) about the plane which pass through the ( $0.5,0.5,0.5$ ) point and the co-ordinate axis $Y$, the values of the solution is given only for

$$
\begin{equation*}
x \leqslant 0.5, \quad y \leqslant 0.5, \quad z \leqslant 0.5 \tag{7.3.2a}
\end{equation*}
$$

with $\quad z \leqslant x$.
The modulus, relative and maximum modulus errors for the test problem are given in Table 7.4 for mesh sizes $M=4,5,6,7,8$ respectively.


TABLE 7.1
Analytic ( + ) and Computed ( ++ ) solution of the E1liptic 3 -problem for a grid ( $3 \times 3 \times 3$ ).

| $\downarrow$ Z | $\pm Y \quad \vec{x}$ | 0.167 | 0.333 | 0.500 |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0.167 | 0.167 | $0.1239295735^{(+)}$ | 0.2146523179 | 0.24785 | 91471 |
|  |  | $0.1273196573^{(++)}$ | 0.2205241152 | 0.25463 | 93146 |
|  | 0.333 | 0.0689733511 | 0.1194653485 | 0.13794 | 67022 |
|  |  | 0.0728697233 | 0.1262140457 | 0.14573 | 94266 |
|  | 0.500 | 0.0535959381 | 0.0928308878 | 0.10719 | 18762 |
|  |  | 0.0574905309 | 0.0995418795 | 0.11494 | 10619 |
| 0.333 | 0.167 |  | 0.3717887206 | 0.42930 | 46358 |
|  |  |  | 0.3819589719 | 0.44104 | 82305 |
|  | $\begin{aligned} & 0.333 \\ & 0.500 \end{aligned}$ |  | 0.2069200533 | 0.23893 | 06970 |
|  |  |  | 0.2186091399 | 0.25242 | 80915 |
|  |  |  | 0.1607878143 | 0.18566 | 17757 |
|  |  |  | 0.1724115928 | 0.19908 | 37590 |
| 0.500 | 0.167 |  |  | 0.49571 | 82941 |
|  |  |  |  | 0.50927 | 86291 |
|  | 0.333 |  |  | 0.27589 | 34045 |
|  |  |  |  | 0.29147 | 88531 |
|  | 0.500 |  |  | 0.21438 | 37523 |
|  |  |  |  | 0.22988 | 21237 |

TABLE 7.2
Analytic (+) and Computed (++) solution of the Elliptic 3D- problem for a grid ( $5 \times 5 \times 5$ )

| $\downarrow$ Z | $\overbrace{+Y} \vec{X}$ | 0.125 | 0.250 | 0.375 | 0.500 |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0.125 | 0.125 | $0.0860303367^{(+)}$ | 0.1589633345 | 0.2076956057 | 0.2248081036 |
|  |  | 0.08702 $83982^{(++)}$ |  |  |  |
|  |  | $0.0870283982^{(++)}$ | 0.1608075117 | 0.2101051393 | 0.2274161641 |
|  | 0.250 | 0.0528370098 | 0.0976300638 | 0.1275598256 | 0.1380697603 |
|  |  | 0.0541087565 | 0.0999799453 | 0.1306300938 | 0.1413929947 |
|  | 0.375 | 0.0363631228 | 0.0671902898 | 0.0877888344 | 0.0952141911 |
|  |  | 0.0376642501 | 0.0695944596 | 0.0909295435 | 0.0984214287 |
|  | 0.500 | 0.0313957736 | 0.0580118253 | 0.0757961025 | 0.0820411102 |
|  |  | 0.0326878251 | 0.0603992251 | 0.0789153906 | 0.0854174033 |
| 0.250 | 0.125 |  | 0.2937259424 | 0.3837714381 | 0.4153912113 |
|  |  |  | 0.2971335376 | 0.3882236758 | 0.4202102787 |
|  | 0.250 |  | 0.1803968353 | 0.2356998240 | 0.2551196511 |
|  |  |  | 0.1847388503 | 0.2413729400 | 0.2612601876 |
|  | 0.375 |  | 0.1241514671 | 0.1622117089 | 0.1755766885 |
|  |  |  | 0.1285937936 | 0.1680158883 | 0.1818590870 |
|  | 0.500 |  | 0.1071918762 | 0.1400529355 | 0.1515922051 |
|  |  |  | 0.1116032157 | 0.1458166284 | 0.1517830781 |
| 0.375 | 0.125 |  |  | 0.5014215480 | 0.5427347726 |
|  |  |  |  | 0.5072386769 | 0.5490311876 |
|  | 0.250 |  |  | 0.3079566609 | 0.3333298878 |
|  |  |  |  | 0.3153689441 | 0.3413528853 |
|  | 0.375 |  |  | 0.2119398113 | 0.2294019987 |
|  |  |  |  | 0.2195233371 | 0.2376103479 |
|  | 0.500 |  |  | 0.1829879787 | 0.1980647609 |
|  |  |  |  | 0.1905186064 | 0.2062158536 |
| 0.500 | 0.125 |  |  |  | 0.5874518847 |
|  |  |  |  |  | 0.5942670751 |
|  | 0.250 |  |  |  | 0.3607936707 |
|  |  |  |  |  | 0.3694777006 |
|  | 0.375 |  |  |  | 0.2483029341 |
|  |  |  |  |  | 0.2571875873 |
|  | 0.500 |  |  |  | 0.2143837523 |
|  |  |  |  |  | 0.2232064315 |

TABLE 7.3
Analytic $(+)$ and Computed $(++)$ solution of the Elliptic 3 -problem for a grid $(7 \times 7 \times 7)$.

| Grid | Modulus Error ${ }^{(+)}$ | Relative Error ${ }^{(+)}$ | Maximum Modulus <br> Error |
| :---: | :---: | :---: | :---: |
| $3^{\times} 3^{\times} 3$ | 0.0337898604 | 0.1576139051 | 0.0337898604 |
| $4^{\times} 4^{\times} 4$ | 0.0200310081 | 0.0938804662 | 0.0200310082 |
| $5^{\times} 5^{\times} 5$ | 0.0154983714 | 0.0722926584 | 0.0155854487 |
| $6^{\times} 6^{\times} 6$ | 0.0109321870 | 0.0510574037 | 0.0109321870 |
| $7 \times 7^{\times} 7$ | 0.0088226791 | 0.0411536743 | 0.0088846531 |

TABLE 7.4

The Modulus, Relative and Maximum Modulus Errors for mesh sizes $M=4,5,6,7,8$. The resulting sparse matrices are of order $\mathrm{N}=27,64,125,216,343$ respectively.
${ }^{(+)}$Note that the Modulus and Relative Errors are given at the centre (even mesh size) or at one of the eight grid points near the centre (odd mesh size) of the unit cube.

## SECTION B: THE PARABOLIC 3D-PROBLEM

7.4 STATEMENT OF THE PROBLEM AND FORMATION OF THE DIFFERENCE EQUATIONS

We consider the solution of the boundary-value problem,

$$
\begin{equation*}
\frac{\partial U}{\partial t}=a^{2}\left\{\frac{\partial^{2} U}{\partial x^{2}}+\frac{\partial^{2} U}{\partial y^{2}}+\frac{\partial^{2} U}{\partial z^{2}}\right\}, \quad(x, y, z) \varepsilon R_{\lambda} \tag{7.4.1}
\end{equation*}
$$

where a is a known constant and
$R_{\lambda}$ is a cube of side $\lambda[0<x, y, z<\lambda]$,
subject to the following boundary conditions:

$$
\begin{equation*}
\left.U\right|_{x=0}=\left.U\right|_{x=\lambda}=\left.U\right|_{y=0}=\left.U\right|_{y=\lambda}=\left.U\right|_{z=0}=\left.U\right|_{z=\lambda}=0,0<t<+\infty \tag{7.4.2}
\end{equation*}
$$

and initial condition

$$
\begin{equation*}
\left.U\right|_{t=0}=\Phi_{0}, \quad 0<x<\lambda, \quad 0<y<\lambda, \quad 0<x<\lambda \tag{7.4.3}
\end{equation*}
$$

With $\Phi_{0}=$ constant, the parabolic P.D.E. (7.4.1) together with boundary and initial conditions (7.4.2),(7.4.3), describes the temperature of a cube, uniformly heated at the initial time $t=0$.

It is easily seen that equation (7.4.1) can be obtained from equation (2.8.1) of the Parabolic 3D-example given in section 2.8 , when the coefficients of the latter equation are $A=C=E \equiv a^{2}$ and $F \equiv 1$, the source term is $Q \equiv 0$, the initial condition $f \equiv \Phi_{0}$ and boundary conditions $g \equiv 0$. Then, proceeding in a similar manner as in section 2.8 , we derive the following fully implicit finite difference discretization form of (7.4.1), i.e.,

$$
\begin{align*}
& \frac{U_{i, j, k}^{(\ell+1)}-U_{i, j, k}^{(\ell)}}{\Delta t}+0(\Delta t)= \\
& +a^{2}\left\{\begin{array}{l}
U_{i-1, j, k^{-2 U_{i, j, k}^{(\ell+1)}+U_{i+1, j, k}^{(\ell+1)}}}^{h_{x}^{2}}+ \\
\left.+\frac{U_{i, j-1, k}^{(\ell+1)}-2 U_{i, j, k}^{(\ell+1)}+U_{i, j+1, k}^{(\ell+1)}}{h_{y}^{2}}+\frac{U_{i, j, k-1}^{(\ell+1)}-2 U_{i, j, k}^{(\ell+1)}+U_{i, j, k+1}^{(\ell+1)}}{h_{z}^{2}}\right\} \\
\\
+0\left(h_{x}^{2}+h_{y}^{2}+h_{z}^{2}\right)
\end{array}\right. \tag{7.4.4}
\end{align*}
$$

where $U_{i, j, k}^{(\ell+1)}$ and $U_{i, j, k}^{(\ell)}$ are the values of the dependent variable at the $(\ell+1)$ and $\ell$ time levels respectively (see Figure 2.9).

Note that the volumetric grid system $R_{h}$, which covers the region $R_{\lambda}$, is now defined by

$$
R_{h}=\left\{\left(i h_{x}, j h_{y}, k h_{z}, r \ell\right): \begin{array}{ll} 
& 0 \leqslant i \leqslant M \\
& 0 \leqslant j \leqslant M \\
& 0 \leqslant k \leqslant M \\
& 0 \leqslant r \leqslant T / \ell
\end{array}\right\}
$$

Then, grouping the system of finite difference equations (7.4.4) in matrix form ordering the $(M-1)^{3}$ internal mesh points as in section 2.8 (see also Figure 2.4), we obtain:

$$
\begin{equation*}
\Omega \cdot \underline{u}^{(\ell+1)}=\underline{u}^{(\ell)} \tag{7.4.5}
\end{equation*}
$$

where $\Omega$ is a real, square, seven-diagonal matrix of order $N$, semibandwidths $m$ and $p$ and is given by (7.2.10), where $\Gamma$ now is the following diagonal matrix of order $m^{2}=(M-1)^{2}$

$$
\begin{equation*}
\Gamma=\operatorname{diag}\left\{a^{2} \frac{\Delta t}{h_{z}^{2}}, a^{2} \frac{\Delta t}{h_{z}^{2}}, \ldots, a^{2} \frac{\Delta t}{h_{z}^{2}}\right\} \tag{7.4.6}
\end{equation*}
$$

and $A_{i}$, $i \varepsilon[1, m]$ are matrices of order $m^{2}$ given by (7.2.12) where $\Delta$, in this case, is given by the following diagonal matrix of order m

$$
\begin{equation*}
\Delta=\operatorname{diag}\left\{a^{2} \frac{\Delta t}{h_{y}^{2}}, a^{2} \frac{\Delta t}{h_{y}^{2}}, \ldots \ldots, a^{2} \frac{\Delta t}{h_{y}^{2}}\right\} \tag{7.4.7}
\end{equation*}
$$

and $B_{i}, i \varepsilon\left[1, m^{2}\right]$ are matrices of order $m$ given by (7.2.14), where

$$
\begin{equation*}
\mathrm{C}=1+2 \mathrm{a}^{2}\left[\Delta \mathrm{t}\left(\frac{1}{h_{x}^{2}}+\frac{1}{h_{y}^{2}}+\frac{1}{h_{z}^{2}}\right)\right] \text { and } \theta=a^{2} \frac{\Delta t}{h_{x}^{2}} . \tag{7.4.8}
\end{equation*}
$$

Note that $\underline{u}^{(\ell+1)}$ and $\underline{u}^{(\ell)}$ of system (7.4.5) are respectively [ $\left.m^{3} \times 1\right]$ column vectors, consisting of the unknown approximate solutions $u_{i, j, k}$ at the $(\ell+1)$ time level and the known boundary values given by (7.4.2), plus the known approximate solutions $u_{i, j, k}$ at the $\ell$ time level.

Alternatively, the Crank-Nicolson implicit scheme can be derived and is given by

$$
\begin{align*}
& +\frac{U_{i, j-1, k}^{(\ell+1)}-2 U_{i, j, k}^{(\ell+1)}+U_{i, j+1, k}^{(\ell+1)}}{h_{y}^{2}}+\frac{U_{i, j-1, k}^{(\ell)}-2 U_{i, j, k}^{(\ell)}+U_{i, j+1, k}^{(\ell)}}{h_{y}^{2}} \\
& \left.+\frac{U_{i, j, k-1}^{(\ell+1)}-2 U_{i, j, k}^{(\ell+1)}+U_{i, j, k+1}^{(\ell+1)}}{h_{z}^{2}}+\frac{U_{i, j, k-1}^{(\ell)}-2 U_{i, j, k}^{(\ell)}+U_{i, j, k+1}^{(\ell)}}{h_{z}^{2}}\right\} \\
& +O\left(h_{x}^{2}+h_{y}^{2}+h_{z}^{2}\right) \quad . \tag{7.4.9}
\end{align*}
$$

Then, the system of finite difference equations so obtained can be expressed in matrix form, namely,

$$
\begin{equation*}
\Omega_{1} \underline{u}^{(\ell+1)}=\Omega_{2} \underline{u}^{(\ell)} \tag{7.4.10}
\end{equation*}
$$

with $\Omega_{1}, \Omega_{2}$ defined as follows:
$\Omega_{1}$ is of the same type and form of matrix $\Omega$ (defined in (7.2.10)-(7.2.15)) and its expression can be obtained by replacing in (7.2.10) $\Omega$ by $\Omega_{1}$.

In this case, $\Gamma$ is given by the following diagonal matrix of order $m^{2}=(M-1)^{2}$

$$
\begin{equation*}
\Gamma=\operatorname{diag}\left\{\frac{\mathrm{a}^{2}}{2} \frac{\Delta \mathrm{t}}{\mathrm{~h}_{\mathrm{z}}^{2}}, \frac{\mathrm{a}^{2}}{2} \frac{\Delta \mathrm{t}}{\mathrm{~h}_{\mathrm{z}}^{2}}, \ldots, \frac{\mathrm{a}^{2}}{2} \frac{\Delta \mathrm{t}}{\mathrm{~h}_{\mathrm{z}}^{2}}\right\} \tag{7.4.11}
\end{equation*}
$$

and $A_{i}$, $i \varepsilon[1, m]$ are matrices of order $\mathrm{m}^{2}$, given by (7.2.12) where $\Delta$, is given by the following diagonal matrix of order $m$

$$
\begin{equation*}
\Delta=\operatorname{diag}\left\{\frac{\mathrm{a}^{2}}{2} \frac{\Delta \mathrm{t}}{\mathrm{~h}_{\mathrm{y}}^{2}}, \frac{\mathrm{a}^{2}}{2} \frac{\Delta \mathrm{t}}{\mathrm{~h}_{\mathrm{y}}^{2}}, \ldots, \frac{\mathrm{a}^{2}}{2} \frac{\Delta \mathrm{t}}{\mathrm{~h}_{\mathrm{y}}^{2}}\right\} \tag{7.4.12}
\end{equation*}
$$

and $B_{i}$, $i \varepsilon\left[1, m^{2}\right]$ are matrices of order $m$, given by (7.2.14) where

$$
\begin{equation*}
C=1+a^{2}\left[\Delta t\left(\frac{1}{h_{x}^{2}}+\frac{1}{h_{y}^{2}}+\frac{1}{h_{z}^{2}}\right)\right] \quad \text { and } \theta=\frac{a^{2}}{2} \frac{\Delta t}{h_{x}^{2}} . \tag{7.4.13}
\end{equation*}
$$

Similarly $\Omega_{2}$ is of the same form and type as $\Omega_{1}, \Omega$ matrices and its expression can be obtained by replacing $\Omega$ by $\Omega_{2}$ in (7.2.10). Then, $\Gamma$ is given by the diagonal matrix of order $m^{2}=(M-1)^{2}$

$$
\begin{equation*}
\Gamma=\operatorname{diag}\left\{-\frac{a^{2}}{2} \frac{\Delta t}{h_{z}^{2}},-\frac{a^{2}}{2} \frac{\Delta t}{h_{z}^{2}}, \ldots,-\frac{a^{2}}{2} \frac{\Delta t}{h_{z}^{2}}\right\} \tag{7.4.14}
\end{equation*}
$$

and $A_{i}$, $i \varepsilon[1, M]$ are matrices of order $m^{2}$, given by (7.2.12), where $\Delta$, is given by the diagonal matrix of order $m$ :

$$
\begin{equation*}
\Delta=\operatorname{diag}\left\{-\frac{a^{2}}{2} \frac{\Delta t}{h_{y}^{2}},-\frac{a^{2}}{2} \frac{\Delta t}{h_{y}^{2}}, \ldots,-\frac{a^{2}}{2} \frac{\Delta t}{h_{y}^{2}}\right\} \tag{7.4.15}
\end{equation*}
$$

and $B_{i}$, $i \varepsilon\left[1, m^{2}\right]$ are matrices of order $m$, given by (7.2.14) where

$$
\begin{equation*}
C=1-a^{2}\left[\Delta t\left(\frac{1}{h_{x}^{2}}+\frac{1}{h_{y}^{2}}+\frac{1}{h_{z}^{2}}\right)\right] \quad \text { and } \theta=-\frac{a^{2}}{2} \frac{\Delta t}{h_{x}^{2}} . \tag{7.4.16}
\end{equation*}
$$

### 7.5 ANALYTICAL REPRESENTATION OF THE PARABOLIC 3D-PROBLEM

The analytic solution of this problem can be verified [8] by separation of variables and is given by:

$$
\begin{align*}
U(x, y, z, t)= & \left(\frac{4}{\pi}\right)^{3} \Phi_{0} \sum_{k, m, n=0}^{+\infty} \frac{A_{k, m, n}}{(2 k+1)(2 m+1)(2 n+1)} \cdot \\
& \cdot \sin \frac{(2 k+1) \pi x}{\lambda} \cdot \sin \frac{(2 m+1) \pi y}{\lambda} \cdot \sin \frac{(2 n+1) \pi z}{\lambda}  \tag{7.5.1}\\
\text { here } A_{k, m, n}= & e^{-\frac{a^{2} \pi^{2}}{\lambda^{2}}\left[(2 k+1)^{2}+(2 m+1)^{2}+(2 n+1)^{2}\right] t} \tag{7.5.2}
\end{align*}
$$

The time at which a steady-state solution will occur at the centre of the cube with relative accuracy $\varepsilon>0$, (where $\varepsilon$ is an arbitrary positive quantity) can be easily seen to be:

$$
\begin{equation*}
U\left(\frac{\lambda}{2}, \frac{\lambda}{2}, \frac{\lambda}{2}, t\right)=\left(\frac{4}{\pi}\right)^{3} \Phi_{0}\left\{\sum_{k=0}^{+\infty}(-1)^{k} \frac{e^{\frac{-\pi^{2} a^{2}(2 k+1)^{2}}{\lambda^{2}}} \cdot t}{2 k+1}\right\}^{3} . \tag{7.5.3}
\end{equation*}
$$

Since the series on the right hand side of (7.5.3) satisfies the conditions of Leibnitz's theorem on alternating series, then the remainder of this series does not exceed the value of the first of the residual terms in absolute value, i.e.,

$$
\begin{array}{r}
\left|R_{n}\left(\frac{\lambda}{2}, \frac{\lambda}{2}, \frac{\lambda}{2}, t\right)\right|=\left|\left(\frac{4}{\pi}\right)^{3} \Phi_{0} \sum_{k=n+1}^{+\infty} \frac{(-1)^{k}}{(2 k+1)} e^{-\frac{\pi^{2} a^{2}(2 k+1)^{2}}{\lambda^{2}}} \cdot t\right| \leqslant \\
\leqslant\left(\frac{4}{\pi}\right)^{3} \Phi_{0} \frac{e^{-\frac{\pi^{2} a^{2}(2 n+3)^{2}}{\lambda^{2}}} \cdot t}{2 n+3} \tag{7.5.4}
\end{array}
$$

The ratio of the sum of all terms of the series (7.5.3), excluding the first and the first term of the same series enables us to estimate the time taken for the steady-state solution, i.e.,

$$
\begin{equation*}
\frac{\left|R_{0}\left(\frac{\lambda}{2}, \frac{\lambda}{2}, \frac{\lambda}{2}, t\right)\right|}{\left(\frac{4}{\pi}\right)^{3} \Phi_{0} e^{-\frac{\pi^{2} a^{2}}{\lambda^{2}} \cdot t}}=\frac{1}{3} e^{-\frac{8 \pi^{2} a^{2}}{\lambda^{2}} \cdot t} \leqslant \tilde{\varepsilon} \tag{7.5.5}
\end{equation*}
$$

where $\tilde{\varepsilon}>_{0}$ is an arbitrary positive quantity.
From (7.5.5) now, we can easily obtain

$$
\begin{equation*}
t \geqslant t^{*}=-\frac{\lambda^{2}}{8 \pi^{2} a^{2}} \ln 3 \tilde{\varepsilon} \tag{7.5.6}
\end{equation*}
$$

Then, for all $t$ satisfying the inequality (7.5.6), where $\tilde{\varepsilon}$ is less than the least of the numbers 1 and $\varepsilon / 9$, it is given in [8] that a steady-state solution will occur at the centre of the cube with relative accuracy $\varepsilon>0$.

### 7.6 COMPUTATIONAL RESULTS

For a numerical solution, the implicit finite difference schemes defined in (7.4.4),(7.4.9) were used as an approximation to the original P.D.E. (7.4.1), subject to the conditions (7.4.2), (7.4.3) and the obtained systems (7.4.5), (7.4.10) respectively were solved using the LUBOT-3D Algorithm.

For simplicity, without loss of generality, we choose $\mathrm{a} \equiv 1$, $\Phi_{0} \equiv 1$ and $h_{x}=h_{y}=h_{z}=\frac{1}{M}$. The side $\lambda$ of the cube was fixed $\lambda=1$ and $a$ relative accuracy $\varepsilon=10^{-6}$ was used.

The time step $\Delta t$ is chosen such that $\Delta t=h_{x}^{2}, \Delta t=h_{x}^{3}$ respectively and the values of the theoretical solution $U(x, y, z, t)$ are obtained from (7.5.1), where the first ten terms of the infinite series have been kept.

Numerical results are presented in the accompanying Tables 7.5-7.12, where the values of the analytic and approximate (for both implicit finite difference schemes given in (7.4.4), (7.4.9)) at the internal mesh points of a $(7 \times 7 \times 7)$ grid are given in groups of five decimal figures, for the first four time steps.

Since a great deal of symmetry is involved [specifically the solution is symmetric besides the cases (i), (ii) of section 7.3 and
(iii) about the plane passing through the points

$$
(0,0,0),(0.5,0.5,0),(0.5,0.5,0.5)]
$$

the values of the approximate and analytic solution are given only for

|  | $x \leqslant 0.5, \quad y \leqslant 0.5, \quad z \leqslant 0.5$, | (7.6.1a) |
| :--- | :--- | :--- |
| with | $z \leqslant x$ and $y \geqslant x$. | $(7.6 .1 b)$ |

The modulus and maximum modulus error for the test problem at a grid point, after a number of time steps, are given in Tables 7.13, 7.14.

|  |  | Time step l - Time $\simeq 0.016$ |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\downarrow$ z | $\stackrel{\vec{x}}{ }$ | 0.125 | 0.250 | 0.375 | 0.500 |
| 0.125 | $\begin{aligned} & 0.125 \\ & 0.250 \\ & 0.375 \\ & 0.500 \end{aligned}$ | $\begin{aligned} & 0.1410113288(\mathrm{AS}) \\ & 0.0331748376 \\ & (\mathrm{FI}) \\ & 0.2282979664 \\ & 0.0909444228 \\ & 0.2616263273 \\ & 0.1153432303 \end{aligned}$ | 0.3696102851 <br> 0.3571080484 <br> 0.42356 | $\begin{array}{ll} 0.48540 & 34416 \\ 0.51637 & 69231 \\ 0.49794 & 25949 \\ 0.53545 & 66523 \end{array}$ | $\begin{aligned} & 0.5108056652 \\ & 0.5557075426 \end{aligned}$ |
| 0.250 | $\begin{gathered} 0.250 \\ 0.375 \\ 0.500 \end{gathered}$ |  | $\begin{array}{ll} 0.59839 & 23773 \\ 0.46262 & 94953 \\ 0.68574 & 94284 \\ 0.56215 & 04584 \\ & \\ 0.70346 & 40067 \\ 0.58436 & 95832 \end{array}$ | $\begin{array}{ll} 0.78585 & 94068 \\ 0.68848 & 71595 \\ 0.80616 & 00696 \\ 0.71738 & 08452 \end{array}$ | $\begin{aligned} & 0.82698 \\ & 0.74812 \end{aligned} 61889$ |
| 0.375 | $\begin{aligned} & 0.375 \\ & 0.500 \end{aligned}$ |  |  | 0.9005840639 <br> 0.74397 <br> 05864 <br> 0.92384 <br> 0.77673 | $\begin{array}{ll} 0.94771 & 35976 \\ 0.81165 & 23989 \end{array}$ |
| 0.500 | 0.500 |  |  |  | $\begin{array}{ll} 0.97219 & 53484 \\ 0.82648 & 39111 \end{array}$ |

TABLE 7.5

Analytic (AS) and Computed (FI) [using the Fully Implicit Finite Difference Scheme] solutions of the parabolic 3Dproblem for a grid $(7 \times 7 \times 7)$ at the first time step $\left[\Delta t=h_{x}^{2}=\left(\frac{1}{8}\right)^{2}\right]$.


## TABLE 7.6

Analytic (AS) and Computed (FI) [using the Fully Implicit, Finite Difference Scheme] solutions of the parabolic 3Dproblem for a grid $(7 \times 7 \times 7)$ at the second time step $\left[\Delta t=h_{x}^{2}=\left(\frac{1}{8}\right)^{2}\right]$.

|  |  | ime step 3 - Time $\simeq 0.047$ |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\downarrow 2$ | $\overrightarrow{\mathrm{x}}$ | 0.125 | 0.250 | 0.375 | 0.500 |
| 0.125 |  | $\begin{aligned} & 0.0306291713^{(\mathrm{AS})} \\ & 0.02537 \\ & 07732 \end{aligned} \text { (FI) }$ | $\begin{array}{ll} 0.10219 & 86356 \\ 0.10222 & 52556 \\ 0.13198 & 73367 \\ 0.13449 & 32947 \\ & \\ 0.14217 & 13994 \\ 0.14491 & 10314 \end{array}$ | 0.17045 88027 <br> 0.17702 83688 | $\begin{aligned} & 0.197778636,7 \\ & 0.2056478440 \end{aligned}$ |
| 0.250 | $\begin{aligned} & 0.250 \\ & 0.375 \\ & 0.500 \end{aligned}$ |  | 0.1866809592 0.1715144635 0.2410944381 0.2258945803 0.2596971384 0.2436746894 | $\begin{aligned} & 0.3113682741 \\ & 0.2976195389 \\ & \\ & 0.33539 \\ & 0.32109 \end{aligned}$ | $\begin{aligned} & 0.3612720011 \\ & 0.34648 \\ & 56263 \end{aligned}$ |
| 0.375 | $\begin{aligned} & 0.375 \\ & 0.500 \end{aligned}$ |  |  | $\begin{aligned} & 0.40212 \\ & 0.34197 \\ & 0.36100 \end{aligned} 27728$ | $\begin{aligned} & 0.4665750084 \\ & 0.4207751595 \end{aligned}$ |
| 0.500 | 0.500 |  |  |  | $\begin{aligned} & 0.5025756525 \\ & 0.4432575625 \end{aligned}$ |

TABLE 7.7
Analytic (AS) and Computed (FI) [using the fully Implicit Finite Difference Scheme] solutions of the parabolic 3Dp oblem for a grid $(7 \times 7 \times 7)$ at the third time step $\left[\Delta t=h_{x}^{2}=\left(\frac{1}{8}\right)^{2}\right]$.

|  |  | Time step $4-$ Time $\simeq 0.063$ |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| +2 | $+Y \cdot \vec{X}$ | 0.125 | 0.250 | 0.375 | 0.500 |
| 0.125 | $\begin{aligned} & 0.125 \\ & 0.250 \\ & 0.375 \\ & 0.500 \end{aligned}$ |  | 0.06272 52011 <br> 0.06527 64324 <br> 0.08167 72438 <br> 0.08600 31499 <br> 0.08828 25476 <br> 0.09309 11252 | 0.10635 55323 <br> 0.11332 22849 <br>   <br> 0.11495 65890 <br> 0.12266 59672 | $\begin{array}{ll} 0.12425 & 32200 \\ 0.13278 & 64416 \end{array}$ |
| 0.250 | $\begin{aligned} & 0.250 \\ & 0.375 \\ & 0.500 \end{aligned}$ |  | 0.11551 03530 <br> 0.11457 96713 <br> 0.15041 11123 <br> 0.15095 29028 <br> 0.16257 49788 <br> 0.16344 61036 | 0.1958569263 <br> 0.1988893910 <br> 0.2116960320 <br> 0.21535 | $\begin{array}{ll} 0.22881 & 60589 \\ 0.23319 & 33786 \end{array}$ |
| 0.375 | $\begin{aligned} & 0.375 \\ & 0.500 \end{aligned}$ |  |  | $\begin{array}{ll} 0.25503 & 39199 \\ 0.24866 & 75181 \\ 0.27565 & 87163 \\ 0.26930 & 28962 \end{array}$ | $\begin{array}{ll} 0.29795 & 14565 \\ 0.29166 & 04024 \end{array}$ |
| 0.500 | 0.500 |  |  |  | $\begin{array}{ll} 0.32204 & 70283 \\ 0.31044 & 26938 \end{array}$ |

TABLE 7.8
Analytic (AS) and Computed (FI) [using the Fully Implicit Finite Difference Scheme] solutions of the parabolic 3Dproblem for a grid $(7 \times 7 \times 7)$ at the fourth time step $\left[\Delta t=h_{x}^{2}=\left(\frac{1}{8}\right)^{2}\right]$

|  |  | Time step l - Time $\simeq 0.002$ |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\downarrow 2$ | $1{ }^{1} \quad \vec{X}$ | 0.125 | 0.250 | 0.375 | 0.500 |
| 0.125 | $\begin{gathered} 0.125 \\ 0.250 \\ 0.375 \\ 0.500 \end{gathered}$ |  | 0.9593391177 <br> 0.7562045480 <br> 0.70657 <br> 04549 <br> 0.95918 <br> 0.81638 <br> 67604 <br> 0.78891 <br> 46167 <br> 0.97200 <br> 36940 <br> 0.82124 <br> 41023 <br> 0.79291 04586 | 0.95903 55735 <br> 0.88469 20535 <br> 0.87845 11729 <br> 0.97184 99061 <br> 0.89042 88769 <br> 0.88293 71840 | 0.9848354596 <br> 0.8962854085 <br> 0.8874662005 |
| 0.250 | $\begin{gathered} 0.250 \\ 0.375 \\ 0.500 \end{gathered}$ | , | 0.98568 51994 <br> 0.82306 03300 <br> 0.79375 24478 <br> 0.98552 92470 <br> 0.89264 98765 <br> 0.88391 07969 <br> 0.99869 75796 <br> 0.89857 38761 <br> 0.88845 41062 | 0.98537 33192 <br> 0.97205 74746 <br> 0.98222 22015 <br> 0.99853 95684 <br> 0.97909 06592 <br> 0.98733 90425 | $\begin{array}{ll} 1.01188 & 17.408 \\ 0.98628 & 02112 \\ 0.99250 & 76754 \end{array}$ |
| 0.375 | $\begin{aligned} & 0.375 \\ & 0.500 \end{aligned}$ |  |  | 0.98521 74160 <br> 0.98063 15189 <br> 0.98793 75819 <br> 0.99838 15821 <br> 0.98787 13157 <br> 0.99311 55662 | 1.01172 16438 <br> 0.99527 49114 <br> 0.99834 64881 |
| 0.500 | 0.500 |  |  |  | $\begin{array}{ll} 1.02523 & 99508 \\ 0.99608 & 32861 \\ 0.99865 & 27983 \end{array}$ |

TABLE 7.9

Analytic (AS) and Computed [using the Fully Implicit (FI) and the Crank-Nicolson (CN) Implicit Schemes respectivelyl solutions of the parabolic 3D-problem for a grid ( $7 \times 7 \times 7$ ) at the first time step $\left[\Delta t=h_{x}^{3}=\left(\frac{1}{8}\right)^{3}\right]$.

|  |  | Time step $2-\mathrm{Time} \simeq 0.004$ |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\downarrow 2$ | $\pm Y \quad X$ | 0.125 | 0.250 | 0.375 | 0.500 |
| 0.125 | $\begin{aligned} & 0.125 \\ & 0.250 \\ & 0.375 \\ & 0.500 \end{aligned}$ | $\begin{aligned} & 0.6008850413 \\ & 0.0219314898 \\ & 0.0162486965 \\ & 0.70827 \\ & \text { (FI) } \\ & \text { (CN) } \end{aligned}$ | $\begin{array}{ll} 0.83486 & 43907 \\ 0.59292 & 95464 \\ 0.62124 & 65962 \\ 0.83893 & 88563 \\ 0.67967 & 25275 \\ 0.71558 & 31892 \\ 0.84014 & 85306 \\ 0.68994 & 18860 \\ 0.72331 & 78770 \end{array}$ |   <br> 0.84303 32068 <br> 0.78255 65820 <br> 0.82322 70364 <br> 0.84424 87849 <br> 0.79509 89785 <br> 0.83222 87847 | $\begin{array}{ll}0.84546 & 61157 \\ 0.80801 & 71995 \\ 0.84137 & 21391\end{array}$ |
| 0.250 | $\begin{aligned} & 0.250 \\ & 0.375 \\ & 0.500 \end{aligned}$ |  | $\begin{array}{ll} 0.98407 & 50252 \\ 0.69521 & 55716 \\ 0.72567 & 36314 \\ 0.98887 & 76971 \\ 0.80171 & 40220 \\ 0.83503 & 30128 \\ 0.99030 & 35698 \\ 0.81489 & 18758 \\ 0.84423 & 50288 \end{array}$ | 0.99370 38080 <br> 0.92871 27081 <br> 0.96015 20884 <br> 0.99513 66396 <br> 0.94488 24337 <br> 0.97089 42304 | $\begin{array}{ll} 0.99657 & 15372 \\ 0.96156 & 16118 \\ 0.98181 & 27148 \end{array}$ |
| 0.375 | $\begin{aligned} & 0.375 \\ & 0.500 \end{aligned}$ |  |  | 0.99855 34722 <br> 0.94976 86238 <br> 0.97270 91968 <br> 0.99999 32965 <br> 0.96665 50915 <br> 0.98366 76839 | $\begin{array}{ll} 1.00143 & 51969 \\ 0.98407 & 39405 \\ 0.99480 & 78518 \end{array}$ |
| 0.500 | 0.500 |  |  |  | $\begin{aligned} & 1.0028791765 \\ & 0.9866706006 \\ & 0.9957454806 \end{aligned}$ |

TABLE 7.10

Analytic (AS) and Computed [using the Fully Implicit (FI) and the Crank-Nicolson (CN) Implicit Schemes respectively] solutions of the parabolic 3D-problem for a grid ( $7 \times 7 \times 7$ ) at the second time step $\left[\Delta t=h_{x}^{3}=\left(\frac{1}{8}\right)^{3}\right]$.

|  |  | Time step 3 - Time $\simeq 0.006$ |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\downarrow$ 2 | , x | 0.125 | 0.250 | 0.375 | 0.500 |
| 0.125 | $\begin{gathered} 0.125 \\ 0.250 \\ 0.375 \\ 0.500 \end{gathered}$ | $\begin{array}{ll} 0.42507 & 22749 \end{array}{ }^{\text {(AS) }} \text { (FI) }$ | $\begin{array}{ll} 0.72065 & 37686 \\ 0.48022 & 44775 \\ 0.55181 & 85284 \\ 0.73568 & 33011 \\ 0.57620 & 97777 \\ 0.65278 & 09583 \\ 0.73617 & 44213 \\ 0.59094 & 08475 \\ 0.66382 & 90271 \end{array}$ | 0.75102 62807 <br> 0.69398 00086 <br> 0.77194 03736 <br> 0.75152 76435 <br> 0.71243 83981 <br> 0.78516 40644 | $\left\lvert\, \begin{array}{ll} 0.75202 & 93409 \\ 0.73161 & 33666 \\ 0.79867 & 90358 \end{array}\right.$ |
| 0.250 | $\begin{gathered} 0.250 \\ 0.375 \\ 0.500 \end{gathered}$ |  | 0.93833 64856 <br> 0.60065 50853 <br> 0.66819 96534 <br> 0.95790 58812 <br> 0.72485 68130 <br> 0.79048 93980 <br> 0.95854 53507 <br> 0.74462 11195 <br> 0.80414 89379 | 0.97788 34048 <br> 0.87797 11128 <br> 0.93516 55897 <br> 0.97853 62106 <br> 0.90283 15140 <br> 0.95156 00525 | $\begin{aligned} & 0.9791894522 \\ & 0.9286959432 \\ & 0.9683283041 \end{aligned}$ |
| 0.375 | $\begin{aligned} & 0.375 \\ & 0.500 \end{aligned}$ |  |  | 0.99827 75678 <br> 0.91253 72512 <br> 0.95517 02848 <br> 0.99894 39881 <br> 0.93888 40398 <br> 0.97204 09843 | $\begin{array}{ll} 0.99961 & 08534 \\ 0.96631 & 08887 \\ 0.98930 & 00123 \end{array}$ |
| 0.500 | 0.500 |  |  |  | $\begin{array}{ll} 1.00027 & 81638 \\ 0.97152 & 88156 \\ 0.99118 & 24008 \end{array}$ |

TABLE 7.11
Analytic (AS) and Computed [using the Fully Implicit (FI) and the Crank-Nicolson (CN) Implicit Schemes respectively] solutions of the parabolic 3D-problem for a grid ( $7 \times 7 \times 7$ ) at the third time step $\left[\Delta t=h_{x}^{3}=\left(\frac{1}{8}\right)^{3}\right]$.

|  |  | Time step $4-$ Time $\simeq 0.008$ |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\downarrow 2$ | $+y^{\vec{x}}$ | 0.125 | 0.250 | 0.375 | 0.500 |
| 0.125 | $\begin{gathered} 0.125 \\ 0.250 \\ 0.375 \\ 0.500 \end{gathered}$ | $\begin{array}{ll} \hline 0.31819 & 11854 \\ 0.03894 & 09041 \\ \text { (FI) } \\ 0.02962 & 61535 \\ 0.44486 & 81769 \\ 0.11830 & 24208 \end{array}$ | 0.62197 73104 <br> 0.40002 29509 <br> 0.49476 87672 <br> 0.64986 90742 <br> 0.49650 38752 <br> 0.59869 34751 <br> 0.65155 53356 <br> 0.51440 25609 <br> 0.61256 59588 | 0.67901 16178 <br> 0.61794 44490 <br> 0.72453 56226 <br> 0.68077 34973 <br> 0.64080 66017 <br> 0.74151 43610 | 0.68253 99485 <br> 0.66474 45068 <br> 0.75897 24771 |
| 0.250 | $\begin{gathered} 0.250 \\ 0.375 \\ 0.500 \end{gathered}$ |  | 0.86959 63963 <br> 0.52891 29012 <br> 0.61929 80110 <br> 0.90859 23293 <br> 0.65961 51.330 <br> 0.74987 27773 <br> 0.91094 99181 <br> 0.68458 36479 <br> 0.76760 83011 | 0.94933 69848 <br> 0.82476 12631 <br> 0.90832 20484 <br> 0.95180 02967 <br> 0.85674 92738 <br> 0.93008 12300 | $\begin{array}{ll} 0.95427 & 00003 \\ 0.89028 & 97563 \\ 0.95247 & 24790 \end{array}$ |
| 0.375 | $\begin{gathered} 0.375 \\ 0.500 \end{gathered}$ |  |  | 0.99190 87819 <br> 0.87221 55182 <br> 0.93599 19914 <br> 0.99448 25578 <br> 0.90663 51897 <br> 0.95858 69830 | $\begin{array}{ll} 0.99706 & 30122 \\ 0.94274 & 74400 \\ 0.98184 & 42759 \end{array}$ |
| 0.500 | 0.500 |  |  |  | $\begin{array}{ll} 0.99965 & 01622 \\ 0.95114 & 65365 \\ 0.98495 & 34627 \end{array}$ |

TABLE 7.12
Analytic (AS) and Computed [using the Fully Implicit (FI) and the Crank-Nicolson (CN) Implicit Schemes respectively] solutions of the parabolic 3D-problem for a grid ( $7 \times 7 \times 7$ ) at the fourth time step $\left[\Delta t=h_{x}^{3}=\left(\frac{1}{8}\right)^{3}\right]$.

| Time <br> Step | t | Modulus Error <br> at $\left(\frac{1}{8}, \frac{1}{8}, \frac{1}{8}\right)$ | Modulus Error <br> at $\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$ | Maximum Modulus <br> Error |
| :---: | :---: | :---: | :---: | :---: |
| 1 | 0.016 | 0.1078384500 | 0.1457114373 | 0.3082823113 |
| 2 | 0.031 | 0.0225869100 | 0.1317112925 | 0.1317112925 |
| 3 | 0.047 | 0.0052583980 | 0.0593180900 | 0.0593180900 |
| 4 | 0.063 | 0.0006273487 | 0.0116043345 | 0.0116043345 |
| 5 | 0.078 | 0.0008056422 | 0.0114154882 | 0.0134033480 |
| 10 | 0.156 | 0.0007280248 | 0.0129342865 | 0.0129342865 |
| 15 | 0.234 | 0.0001712647 | 0.0030554197 | 0.0030554197 |
| 20 | 0.313 | 0.0000321005 | 0.0005727818 | 0.0005727818 |
| 25 | 0.391 | 0.0000054867 | 0.0000979028 | 0.0000979028 |
| 30 | 0.469 | 0.00000089512 | 0.0000159722 | 0.0000159722 |
| 35 | 0.547 | 0.0000001423 | 0.0000025385 | 0.0000025385 |
| 40 | 0.625 | 0.0000000223 | 0.0000003972 | 0.0000003972 |
| 45 | 0.703 | 0.0000000034 | 0.0000000616 | 0.0000000616 |
| 50 | 0.781 | 0.0000000005 | 0.0000000095 | 0.0000000095 |
| 55 | 0.859 | $0.82635 \times 10-10$ | 0.0000000014 | 0.0000000014 |
| 60 | 0.938 | $0.12496 \times 106$ | 0.0000000002 | 0.0000000002 |

TABLE 7.13
Modulus and Maximum Modulus errors of the parabolic 3Dproblem for a grid $(7 \times 7 \times 7)$ using the Fully Implicit Difference Formula (7.4.4) $\left[\Delta t=h_{x}^{2}=\left(\frac{1}{8}\right)^{2}\right]$.

| 1 | 0.002 | 0.89848 | 65963 | 0.02915 | 66648 |
| ---: | :--- | :--- | :--- | :--- | :--- |
| 2 | 0.004 | 0.5789535516 | 0.0162085760 | 0.9330963163 |  |
| 3 | 0.006 | 0.3933023882 | 0.0287493482 | 0.5809622618 |  |
| 4 | 0.008 | 0.2792502814 | 0.0485036258 | 0.4779291371 |  |
| 5 | 0.010 | 0.2051584340 | 0.0715757047 | 0.3983982716 |  |
| 10 | 0.020 | 0.0608427025 | 0.1662381288 | 0.2417815310 |  |
| 15 | 0.029 | 0.0256646452 | 0.1820765008 | 0.1843064484 |  |
| 20 | 0.039 | 0.0136686769 | 0.1566190681 | 0.1566190681 |  |
| 25 | 0.049 | 0.0084375872 | 0.1224657529 | 0.1224657529 |  |
| 30 | 0.059 | 0.0056527425 | 0.0918221288 | 0.0918221288 |  |
| 35 | 0.068 | 0.0039442912 | 0.0674996002 | 0.0674996002 |  |
| 40 | 0.078 | 0.0028037383 | 0.0491294750 | 0.0491294750 |  |
| 45 | 0.088 | 0.0020081858 | 0.0355615508 | 0.0355615508 |  |
| 50 | 0.098 | 0.0014417402 | 0.0256480082 | 0.0256480082 |  |
| 55 | 0.107 | 0.0010348844 | 0.0184458111 | 0.0184458111 |  |
| 60 | 0.117 | 0.0007417705 | 0.0132316045 | 0.0132316045 |  |

TABLE 7.14
dulus errors of the parabolic 3D-
$7 \times 7$ ) using $_{3}$ the Fully Implicit
4.4) $\left[\Delta t=h_{x}^{3}=\left(\frac{1}{8}\right)^{3}\right]$.

### 7.7 CONCLUDING REMARKS

The main purpose of the two problems was to test the effectiveness of the derived algorithmic solution process (see section 3.4 ) by comparing the numerical with the analytic solution, finding the appropriate errors both at each internal mesh point (Elliptic problem) and at each time step (Parabolic problem).

Let $\Lambda_{p}$ be the exact solution of the P.D.E., $\Lambda_{D}$ be the exact solution of the difference equation and $\Lambda_{N}$ be the numerical solution of the difference equation (computed solution). Then, the quantity $\left(\Lambda_{\mathrm{P}}-\Lambda_{\mathrm{N}}\right)$, which represents the total error can be expressed as follows:

$$
\begin{equation*}
\left(\Lambda_{\mathrm{P}}-\Lambda_{\mathrm{N}}\right)=\left(\Lambda_{\mathrm{P}}-\Lambda_{\mathrm{D}}\right)+\left(\Lambda_{\mathrm{D}}-\Lambda_{\mathrm{N}}\right) \tag{7.7.1}
\end{equation*}
$$

where

$$
\begin{aligned}
& \left(\Lambda_{P}-\Lambda_{D}\right) \text { is the truncation error caused by } \\
& \text { finite distances between points of } \\
& \text { the difference mesh and } \\
& \left(\Lambda_{D}-\Lambda_{N}\right) \text { is the numerical error caused mainly } \\
& \\
& \text { by round-off (computer in exact arithmetic). }
\end{aligned}
$$

Convergence and Stability analysis conditions are satisfied when $\Lambda_{D} \rightarrow A_{p}$ as $h_{x}, h_{y}, h_{z} \rightarrow 0$ and when $\left(\Lambda_{D}-\Lambda_{N}\right)$ can be considered a "negligible" quantity respectively. Since with the implicit method the stability condition is satisfied [52], a large value of ( $\Lambda_{\mathrm{P}}-\Lambda_{\mathrm{N}}$ ) is caused by truncation errors.

In particular, from a comparison of the analytic and numerical solutions of the parabolic 3D-problem the quantity ( $\Lambda_{\mathrm{P}}-\Lambda_{N}$ ) was seen to be considerably large for most values of $\Delta t$ and $h_{x}, h_{y}, h_{z}$.

In this case, as in [44], the truncation error proved to be the main factor of the above growth of total error.

Since the absolute value of the mesh size affects the
truncation error, good results could be expected only for sufficiently small values of $h_{x}, h_{y}, h_{z}$ (Elliptic problem) and $\Delta t$ (Parabolic problem). The experiments verified that reasonable results can be obtained with intervals of this size.

Consequently, the algorithmic solution process proves to be an effective and efficient method, given that the discrepancies between numerical and analytic solutions were caused mainly by truncation errors. The applicability and usefulness of the method can be verified by the presented results for both problems in Tables 7.1-7.14.

Finally, we state that the algorithmic procedure can be used for the solution of the "one-phase" oil reservoir flow problem [59], [45].

## Chapter 8

## ON THE SOLUTION OF MILDLY NON-LINEAR ELLIPTIC P.D.E.'S

### 8.1 INTRODUCTION

In this final Chapter we present inner and outer iterative procedures for which both the exact and approximate factorization processes developed in sections 3.2, 4.2, respectively are used and a strategy for the solution of mildly non-linear elliptic P.D.E.'s is described.

The applicability of the above factorization processes is considered by introducing the following two iterative procedures,
(i) Procedure I, where the outer iteration is a Picard or Newton iteration and the inner iteration is carried out directly by the LUBOT algorithm. In this case, the usual two level 'inner-outer' iterative scheme reduces to an equivalent one-level iteration.
(ii) Procedure II, where a two-level iteration is involved. The outer iteration is a Picard or modified Newton iteration and the inner iteration is performed by a modified strongly Implicit procedure (the ALUBOT algorithm).

### 8.2 LINEARIZATION AND QUASI-LINEARIZATION METHODS

We consider the system of non-linear equations

$$
\begin{equation*}
L u=f(u), \quad u \in R \tag{8.2.1}
\end{equation*}
$$

where $L$ is an elliptic partial differential operator (of at least second order), $f$ is an n-component column vector given by

$$
\begin{equation*}
f(u)=\left(f_{1}(u), f_{2}(u), \ldots, f_{n}(u)\right), \tag{8.2.2}
\end{equation*}
$$

and $R$ is a finite, connected, two-dimensional region subject to certain boundary conditions on $C$ the boundary of $R$.

The method of linearization solves the system (8.2.1) as the limit of the solutions of a sequence of linear equations of the same form as (8.2.1), using the method of successive approximations.

The simplest variant of the method is the Picard's iteration. Let the sequence of functions $\left\{u^{(k)}\right\}$ satisfy the sequence of linear P.D.E's

$$
\begin{equation*}
L u^{(k+1)}=f\left(u^{(k)}\right), \quad u^{(k)}, u^{(k+1)} \varepsilon R \tag{8.2.3}
\end{equation*}
$$

where $u^{(0)}$ indicates the initial guess, and the same boundary conditions specified for $u$.

When the sequence $\left\{u^{(k)}\right\}$ converges, let say

$$
\begin{equation*}
\lim _{k \rightarrow+\infty} u^{(k)}=\xi_{0} \text { with } f\left(\xi_{0}\right)=0, \quad \xi_{0} \varepsilon R \tag{8.2.4}
\end{equation*}
$$

the convergence is linear, i.e.,

$$
\begin{equation*}
u^{(k+1)}-u \equiv 0\left[u^{(k)}-u\right], \text { as } k \rightarrow+\infty \tag{8.2.5}
\end{equation*}
$$

The main disadvantages of the method are:
(i) the convergence is relatively slow, geometric at best and
(ii) an initial guess $u^{(0)}$ sufficiently close to the solution $u$ is required for convergence.

Assuming now that $f$ is continuous and differentiable with respect to $u$, the right hand side of (8.2.3) can be replaced by the expansion about $u^{(k)}$ i.e.,

$$
\begin{equation*}
f\left(u^{(k)}\right)+\left[u^{(k+1)}-u^{(k)}\right] f^{\prime}\left(u^{(k)}\right) \tag{8.2.6}
\end{equation*}
$$

Then, from (8.2.3),(8.2.6) we get the Newton's iteration:

$$
\begin{equation*}
L u^{(k+1)}=f\left(u^{(k)}\right)+\left[u^{(k+1)}-u^{(k)}\right] f^{\prime}\left(u^{(k)}\right) \tag{8.2.7}
\end{equation*}
$$

where $f^{\prime}$ denotes the Jacobian of $f$ with respect to $u^{(k)}$, (see Definition 1.6.3).

The method of Quasilinearization replaces the non-linear equations (8.2.1) by the sequence of linear equations:

$$
L u^{(k+1)}-f^{\prime}\left(u^{(k)}\right) u^{(k+1)}=f\left(u^{(k)}\right)-f^{\prime}\left(u^{(k)}\right) u^{(k)}(8.2 .8)
$$

where (8.2.8) is subject to the same boundary conditions as (8.2.1).

When the sequence $\left\{u^{(k)}\right\}$ converges (see (8.2.4)), assuming that there are the required higher order derivatives of $f(u)$ and the Jacobian $f$ ' be non-singular, tlaen the convergence of the Newton's method is quadratic, i.e.

$$
\begin{equation*}
u^{(k+1)}-u \equiv 0\left[\left(u^{(k)}-u\right)^{2}\right] \text {, as } k \rightarrow+\infty . \tag{8.2.9}
\end{equation*}
$$

## Definitions 8.4.1

The iterative scheme

$$
\begin{equation*}
\mathrm{A}\left(\underline{\mathrm{u}}^{\left.(\mathrm{k}+1) \underline{\mathrm{u}}^{(\mathrm{k})}\right)=-\mathrm{f}\left(\underline{\mathrm{u}}^{(\mathrm{k})}\right), \quad \mathrm{k}=0,1,2, \ldots,, ~}\right. \tag{8.2.10}
\end{equation*}
$$

where $A$ is a non-singular matrix to be defined later, is said to be 'locally convergent' if assuming that $\underline{u}^{(0)}$ is sufficiently close to a solution $\underline{\xi}_{0}$ of $f(\underline{u})=0$, then $\underset{k \rightarrow+\infty}{\lim } \underline{u}^{(k)}=\xi_{0}$.

## Note that:

(i) when $f^{\prime}\left(\underline{\xi_{0}}\right)$ exists, the sufficient and necessary condition for local convergence is

$$
\begin{equation*}
\rho\left(I-A^{-1} f^{\prime}\left(\underline{\xi}_{0}\right)\right)<1 \tag{8.2.11}
\end{equation*}
$$

where $\rho$ denotes the spectral radius of the matrix.
It is well known [48] that the smaller the spectral radius, the more rapid the convergence.
 iterative scheme $(8.2 .10)$ becomes $\left.\underline{u}^{(k+1)}=G \underline{u}^{(k)}\right)$ (Picard's iteration) while, for $A=f^{\prime}\left(\underline{u}^{(k)}\right.$ ) we obtain

$$
\left.f^{\prime}\left(\underline{u}^{(k)}\right)\left(\underline{u}^{(k+1)} \underline{-}^{(k)}\right)=-f \underline{u}^{(k)}\right) \text { (Newton iteration). }
$$

### 8.3 FORMATION OF THE MILDLY NON-LINEAR ELLIPTIC DIFFERENCE EQUATIONS

We consider a class of mildy non-linear elliptic boundary value problems in two space dimensions of the form,

$$
\begin{equation*}
\frac{\partial}{\partial x}\left(\Lambda(x, y) \frac{\partial U}{\partial x}\right)+\frac{\partial}{\partial y}\left(\tilde{\Lambda}(x, y) \frac{\partial U}{\partial y}\right)=f(x, y, U), \quad\left(x_{i}, y_{j}\right) \equiv\left(i h_{x}, j h_{y}\right) \varepsilon R \tag{8.3.1}
\end{equation*}
$$

where $\Lambda, \tilde{\Lambda}$ are non-negative and $R$ is a finite, connected, twodimensional region subject to the general boundary condition,

$$
\begin{equation*}
a \cdot U+\beta \frac{\partial U}{\partial \zeta}=\gamma, \quad\left(x_{i}, y_{j}\right) \varepsilon C \tag{8.3.2}
\end{equation*}
$$

where $C$ is the exterior boundary of $R, \zeta$ denotes the direction of the outward drawn normal and $a, \beta$ are positive, piecewise continuous on C .

If a rectilinear network of mesh spacings $h_{x}, h_{y}$ in the $X, Y$ directions respectively is superimposed over the region $R$, then the central finite difference analogue of (8.3.1) at the point

$$
\begin{align*}
& (i, j) \equiv\left(i h_{x}, j h_{y}\right) \text { is } \\
& \qquad \quad\left[\delta_{x}\left(\Lambda_{i, j} \delta_{x}\right)+\delta_{y}\left(\tilde{\Lambda}_{i, j} \delta_{y}\right)\right] u_{i, j}=f\left(x_{i}, y_{j}, u_{i, j}\right),\left(x_{i}, y_{j}\right) \in R \tag{8.3.3}
\end{align*}
$$

where $\delta_{x}, \delta_{y}$ denote the usual central difference operators with respect to $x, y$ respectively (see equations (2.6.7)-(2.6.8)).

The solution of the set of non-1inear algebraic equations (8.3.3) can be obtained by
(i) the linearized Picard iteration [1] defined by,

$$
\begin{equation*}
\left[\delta_{x}\left(\Lambda_{i, j} \delta_{x}\right)+\delta_{y}\left(\tilde{\Lambda}_{i, j} \delta_{y}\right)\right] u_{i, j}^{(k+1)}=f\left(x_{i}, y_{j}, u_{i, j}^{(k)}\right),\left(x_{i}, y_{j}\right) \varepsilon R, \tag{8.3.4}
\end{equation*}
$$

(ii) the quasilinearized Newton second-order iteration [4] given by,

$$
\begin{align*}
& {\left[\delta_{x}\left(\Lambda_{i, j} \delta_{x}\right)+\delta_{y}\left(\tilde{\Lambda}_{i, j} \delta_{y}\right)\right] u_{i, j}^{(k+1)}-f^{\prime}\left(x_{i}, y_{j}, u_{i, j}^{(k)}\right) u_{i, j}^{(k+1)}=} \\
& \quad=f\left(x_{i}, y_{j}, u_{i, j}^{(k)}\right)-f^{\prime}\left(x_{i}, y_{j}, u_{i, j}^{(k)}\right) u_{i, j}^{(k)}, \quad\left(x_{i}, y_{j}\right) \in R \tag{8.3.5}
\end{align*}
$$

where $f^{\prime}$ denotes the Jacobian of $f$ with respect to $u_{i, j}^{(k)}$. The resulting large order, sparse, linear systems are of the form,

$$
\begin{equation*}
\mathrm{A}^{(\mathrm{k}+1)}=\mathrm{s}\left(\underline{\mathrm{u}}^{(\mathrm{k})}\right), \tag{8.3.6}
\end{equation*}
$$

(where the coefficient matrix $A$ is of the same type and form as in (3.2.2)) and have to be solved many times.

Similarly we consider now a class of mildly non-linear elliptic boundary value problems in three space dimensions of the form:

$$
\begin{gather*}
\frac{\partial}{\partial x}\left(\Lambda(x, y, z) \frac{\partial U}{\partial x}\right)+\frac{\partial}{\partial y}\left(\tilde{\Lambda}(x, y, z) \frac{\partial U}{\partial y}\right)+\frac{\partial}{\partial z}\left(\tilde{\Lambda}(x, y, z) \frac{\partial U}{\partial z}\right)=f(x, y, z, U), \\
\left(x_{i}, y_{j}, z_{k}\right) \equiv\left(i h_{x}, j h_{y}, k h_{z}\right) \varepsilon R, \tag{8.3.7}
\end{gather*}
$$

where $\Lambda, \tilde{\Lambda}, \widetilde{\tilde{\Lambda}}$ are non-negative and R is a finite, compact, three dimensional region subject to the general boundary conditions (8.3.2) with $\left(x_{i}, y_{j}, z_{k}\right) \varepsilon C$, the exterior boundary of $R$.

Proceeding in a similar manner as in the two-dimensional case the following central finite difference analogue of (8.3.7) is obtained at the point ( $i, j, k) \equiv\left(i h_{x}, j h_{z}, k h_{z}\right)$,

$$
\begin{gather*}
{\left[\delta_{x}\left(\Lambda_{i, j, k} \delta_{x}\right)+\delta_{y}\left(\tilde{\Lambda}_{i, j, k} \delta_{y}\right)+\delta_{z}\left(\tilde{\tilde{\Lambda}}_{i, j, k} \delta_{z}\right)\right] u_{i, j, k}=f\left(x_{i}, y_{j}, z_{k}, u_{i, j, k}\right),} \\
\left(x_{i}, y_{j}, z_{k}\right) \varepsilon R, \tag{8.3.8}
\end{gather*}
$$

where $\delta$ denotes the usual central difference operator.
Then, the linearized Picard iteration is defined by:

$$
\begin{array}{r}
{\left[\delta_{x}\left(\Lambda_{i, j, k} \delta_{x}\right)+\delta_{y}\left(\tilde{\Lambda}_{i, j, k} \delta_{y}\right)+\delta_{z}\left(\tilde{\tilde{\Lambda}}_{i, j, k} \delta_{z}\right)\right] u_{i, j, k}^{(n+1)}=f\left(x_{i}, y_{j}, z_{k}, u_{i, j, k}^{(n)}\right)} \\
\left(x_{i}, y_{j}, z_{k}\right) \varepsilon R ; \tag{8.3.9}
\end{array}
$$

and the quasilinearized Newton iteration is given by:

$$
\begin{gather*}
{\left[\delta_{x}\left(\Lambda_{i, j}, k \delta_{x}\right)+\delta_{y}\left(\tilde{\Lambda}_{i, j, k} \delta_{y}\right)+\delta_{z}\left(\tilde{\tilde{\Lambda}}_{i, j, k} \delta_{z}\right)\right] u_{i, j, k}^{(n+1)}-f\left(x_{i}, y_{j}, z_{k}, u_{i, j, k}^{(n)}\right) u_{i, j, k}^{(n+1)}=} \\
=f\left(x_{i}, y_{j}, z_{k}, u_{i, j, k}^{(n)}\right)-f^{\prime}\left(x_{i}, y_{j}, z_{k}, u_{i, j, k}^{(n)}\right) u_{i, j, k}^{(n)},\left(x_{i}, y_{j}, z_{k}\right) \varepsilon R \tag{8.3.10}
\end{gather*}
$$

where $f^{\prime}$ denotes the Jacobian of $f$ with respect to $u_{i, j, k}^{(n)}$.
Both cases lead to the solution of large order, sparse,
linear systems of the form,

$$
\begin{equation*}
\Omega \underline{u}^{(n+1)}=s\left(^{(n)}\right) \tag{8.3.11}
\end{equation*}
$$

(where the coefficient matrix $\Omega$ is of the same type and form as in (3.4.2)) and have to be solved many times.

Several techniques [17], [1] have appeared in the literature dealing with the solution of (8.3.4), (8.3.5), (8.3.9), (8.3.10). In the following section the LUBOT algorithm is used for a direct method of solution of the inner iteration in (8.3.6) and after convergence of the outer iteration the desired approximate solution is obtained.

Furthermore, the ALUBOT algorithm in conjunction with the standard stationary iteration method i.e., Simultaneous Displacement method, is incorporated as an inner iteration into (8.3.6) and an experimental estimation of the optimal iterative parameter involved is obtained.

### 8.4 GENERALIZED LINEAR METHODS: DERIVATION OF NEWTON-LUBOT AND NEWTON-ALUBOT METHODS

Utilizing the LUBOT-ALUBOT algorichms in connection with mildly non-linear equations, we consider the Newton-LUBOT and Newton-ALUBOT composite iterative schemes, where Newton's method is the outer (primary) iteration and the inner (secondary) iteration is carried out by LUBOT or ALUBOT algorithm.

Let us assume that the coefficient matrix $A$ of the linear system

$$
\begin{equation*}
\mathrm{A} \underline{u}=\underline{s} \tag{8.4.1}
\end{equation*}
$$

can be decomposed or split as $A=B-R$, where $B$ is non-singular and under the assumption that the linear system

$$
\text { B. } \underline{u}=\underline{s}^{*}
$$

can be 'easily' solved. It is known that an iterative scheme can
be defined by

$$
\begin{equation*}
\underline{\mathrm{Bu}}_{(\mathrm{r}+1)}=\underline{R u}_{(\mathrm{r})}+\underline{s}, \quad \mathrm{r}=0,1,2, \ldots \tag{8.4.2}
\end{equation*}
$$

or

$$
\begin{equation*}
B\left(\underline{u}(r+1)^{-\underline{u}}(r)\right)=\underline{s}-A \underline{u}(r), \quad r=0,1,2, \ldots \tag{8.4.3}
\end{equation*}
$$

where appropriate splittings of matrices $B$ and $R$ lead to well
known standard iterative methods i.e., Simultaneous Displacement, S.O.R., A.D.I., etc.

Let us assume that $B=L_{S} U_{S}$ and $R=L_{S} U_{S}-L U$, where $L, U$ and $L_{s}, U_{S}$ are the lower and upper triangular matrices in their full and sparse forms given by (3.2.3),(3.2.4) and (4.2.2) respectively. Then, the iterative scheme (8.4.3) can be written as

$$
\begin{equation*}
L_{s} U_{s}\left(\underline{u}_{(r+1)}^{-\underline{u}}(r)\right)=\underline{r}_{(r)}, r=0,1,2, \ldots \tag{8.4.4}
\end{equation*}
$$

where

$$
\begin{equation*}
\underline{r}_{(r)}=\underline{s}-A \underline{u}(r) . \tag{8.4.5}
\end{equation*}
$$

The derivation of the composite non linear-linear methods can now be considered for any linear iterative method of the form (8.4.4) and any non-linear iterative method given by

$$
\begin{equation*}
\left.A_{k} \underline{u}^{(k+1)}-\underline{u}^{(k)}\right)=-\underline{f}^{\left(\underline{u}^{(k)}\right)}, \quad k=0,1,2, \ldots, \tag{8.4.6}
\end{equation*}
$$

where the matrix $A_{k}$ can be split as

$$
\begin{equation*}
A_{k}=B_{k}-R_{k} \tag{8.4.7}
\end{equation*}
$$

Then, provided that the matrix $B_{k}$ is non-singular we get

$$
\begin{equation*}
A_{k}^{-1}=\left(I-B_{k}^{-1} R_{k}\right)^{-1} B_{k}^{-1}=\left(I+H_{k}+H_{k}^{2}+\ldots+H_{k}^{\left(\mu_{k}-1\right)}\right) B_{k}^{-1} \tag{8.4.8}
\end{equation*}
$$

where the first $\mu_{k}$ terms have been retained in the expansion of $\left(I-B_{k}^{-1} R_{k}\right)^{-1}$ and $I$ is the identity matrix. Combination of (8.4.6), (8.4.8) leads to the following iterative scheme,

$$
\begin{equation*}
\underline{u}^{(k+1)} \underline{u}^{(k)}=-\left(\mathrm{I}+\mathrm{H}_{\mathrm{k}}+\ldots+\mathrm{H}_{\mathrm{k}}^{\left(\mu_{k}-1\right)}\right) \mathrm{B}_{\mathrm{k}}^{-1} \mathrm{f}\left(\underline{u}^{(\mathrm{k})}\right), \mathrm{k}=0,1,2, \ldots \tag{8.4.9}
\end{equation*}
$$

where $H_{k}=B_{k}^{-1} R_{k}, k=0,1,2, \ldots$.
The explicit procedure (8.4.9) is a composite iteration in which, at the $k^{\text {th }}$ stage starting from $\underline{u}^{(k)}, \mu_{k}$ steps of the inner linear iteration are carried out in order to approximate a solution
of the outer iteration.
The choice,

$$
\begin{equation*}
A_{\bar{k}} f^{\prime}\left(\underline{u}^{(k)}\right), \quad B_{k}=L_{s} U_{s} \text { and } \quad R_{k}=L_{s} U_{s}-L U, \tag{8.4.10}
\end{equation*}
$$

in which only the first term in the expansion in (8.4.8) is retained, leads to the following first order Newton-ALUBOT iterative scheme,

$$
\begin{equation*}
L_{s} U_{s}\left(\underline{u}^{(k+1)} \underline{\underline{u}}^{(k)}\right)=-f\left(\underline{u}^{(k)}\right), \quad k=0,1,2, \ldots, \tag{8.4.11}
\end{equation*}
$$

whilst retaining the first two terms in the expansion in (8.4.8) a second order Newton-ALUBOT iteration can be obtained, viz.,

$$
\begin{equation*}
L_{s} U_{s}\left(\underline{u}^{(k+1)}-\underline{u}^{(k)}\right)=-\left[I-\left(L_{s} U_{s}\right)^{-1}\left(L U-L_{s} U_{s}\right)\right] f\left(\underline{u}^{(k)}\right) . \tag{8.4.12}
\end{equation*}
$$

The Newton-LUBOT iterative scheme can be easily derived from (8.4.6), assuming that $A_{k}$ can be decomposed as $A_{k}=L U$ and is given by

$$
\begin{equation*}
\operatorname{LU}\left(\underline{\mathrm{u}}^{(\mathrm{k}+1)} \underline{\mathrm{u}}^{(\mathrm{k})}\right)=-\mathrm{f}\left(\underline{\mathrm{u}}^{(\mathrm{k})}\right), \mathrm{k}=0,1,2, \ldots . \tag{8.4.13}
\end{equation*}
$$

### 8.5 ITERATIVE PROCEDURES AND NUMERICAL RESULTS

We consider the non-linear elliptic P.D.E.

$$
\begin{align*}
& \frac{\partial^{2} U}{\partial x^{2}}+\frac{\partial^{2} U}{\partial y^{2}}=e^{U} \quad, \quad(x, y) \varepsilon R  \tag{8.5.1}\\
& R=\left\{(x, y), \begin{array}{ll}
0 \leqslant x \leqslant x_{\max } \\
0 \leqslant y \leqslant y_{\max }
\end{array}\right\}, \tag{8.5.1a}
\end{align*}
$$

where
subject to the Dirichlet boundary conditions

$$
\begin{equation*}
U \equiv 0 \text { and } U \equiv 10 \tag{8.5.2}
\end{equation*}
$$

respectively, with $(x, y) \varepsilon C$, exterior boundary of $R$.
The equation (8.5.1) arises in magnetohydrodynamics, (specifically it is of physical interest in diffusion-reaction problems, Vortex problems and electric space charge considerations) and the existence and uniqueness of the solutions to the above boundary value problem are assured by the classical theory.

The linearized Picard and Quasilinearized Newton iterations are outer iterative schemes of the form,

$$
\begin{equation*}
\left(\delta_{x}^{2}+\delta_{y}^{2}\right) \underline{u}^{(k+1)}=e^{u^{(k)}},\left(x_{i}, y_{j}\right) \equiv\left(i h_{x}, j h_{y}\right) \varepsilon R, \tag{8.5.3}
\end{equation*}
$$

and
respectively.

$$
\left(\delta{\underset{x}{ }}_{2}^{+\delta} \delta_{y}^{2}\right) \underline{u}^{(k+1)}-\left[e^{u^{(k)}} \cdot \underline{u}^{(k+1)}\right]=\left[1-\underline{u}^{(k)}\right] \cdot e^{u} \underline{u}^{(k)},
$$

For notational simplification we assume that $h_{x}=h_{y}=h$ and we denote the resulting set of $\left(\frac{x_{\max }}{h_{x}}-1\right) \cdot\left(\frac{y_{\max }}{h_{y}}-1\right)$ numbers $u_{i, j}^{(k)}$, which are the approximations to the solution at (ih $x_{x}, j h_{y}$ ) at the $k^{\text {th }}$ Picard or Newton iteration, by $u^{(k)}$.

Then, a columnwise ordering of the points is introduced and the resulting large order, sparse, linear system (8.3.6) is obtained which can be written in the following matrix form,

$$
\left[\begin{array}{llllll}
\mathrm{B}_{1} & \mathrm{C}_{1} & & & &  \tag{8.5.5}\\
A_{2} & B_{2} & C_{2} & & & \\
\\
& A_{3} & B_{3} & C_{3} & & \\
\\
& & \ddots & \ddots & & \\
& & \ddots & \ddots & & \\
& & \ddots & \ddots & & \\
& & & \ddots & \ddots & \\
& & & & \ddots & \ddots \\
& & & & \ddots & \\
& & & & & A_{n-1} \\
C_{n-2} & \\
B_{n-1} & C_{n-1} \\
& & & & & \\
A_{n} & B_{n}
\end{array}\right] \cdot \underline{u}^{(k+1)}=s\left(\underline{u}^{(k)}\right)
$$

where $B_{i}$, is the $n \times n$ tridiagonal matrix
and $C_{i}, A_{i}$ are the diagonal matrices defined by,

$$
\begin{equation*}
C_{i} \equiv \operatorname{diag}\left\{\tau_{\ell+1}, \tau_{\ell+2}, \ldots, \tau_{\ell+n}\right\} \quad, i=1,2, \ldots, n-1, \tag{8.5.7}
\end{equation*}
$$

and

$$
\begin{equation*}
A_{i} \equiv \operatorname{diag}\left\{v_{\ell+1}, v_{\ell+2}, \ldots, v_{\ell+n}\right\} \quad, i=2,3, \ldots, n, \tag{8.5.8}
\end{equation*}
$$

with

$$
\ell=\left\{\begin{array}{l}
0, \quad \text { if } i=1  \tag{8.5.9}\\
n \times(i-1), \text { if } i \neq 1 .
\end{array}\right.
$$

Equivalently the system (8.3.6) can be written as,

where $\quad c_{\rho(m-1)}=0, \quad a_{\rho(m-1)+1}=0, \quad \rho \varepsilon[1, m-2]$.
For the Picard iteration we have

$$
\begin{equation*}
b_{i}=1, i \varepsilon[1, N] ; c_{i}=-1 / 4 ; \quad i \varepsilon[1, N-1] ; a_{i}=-1 / 4, i \varepsilon[2, N] ; \tag{8.5.11a}
\end{equation*}
$$

$$
\begin{equation*}
\tau_{i}=-1 / 4, \quad i \varepsilon[1, N-m+1] ; \quad v_{i}=-1 / 4, i \varepsilon[m, N] ; \tag{8.5.11b}
\end{equation*}
$$

and

$$
\begin{equation*}
s\left(u_{i}^{(k)}\right)=-\frac{h^{2}}{4} e^{u_{i}^{(k)}}, i \varepsilon[1, N], \tag{8.5.11c}
\end{equation*}
$$

whilst for the Newton iteration we get

$$
\begin{align*}
& b_{i}=1, i \varepsilon[1, N] ; \quad c_{i}=\frac{1}{\left(4+h^{2} e^{u_{i}^{(k)}}\right)}, i \varepsilon[1, N-1] ;  \tag{8.5.12a}\\
& a_{i}=\frac{1}{\left(4+h^{2} e^{u_{i}^{(k)}}\right)}, \quad i \varepsilon[2, N] ; \quad \tau_{i}=\frac{1}{\left(4+h^{2} e^{\left.u_{i}^{(k)}\right)}\right)}, i \varepsilon[1, N-m+1] ;
\end{align*}
$$

$$
\begin{gather*}
v_{i}=1 /\left(4+h^{2} e^{u_{i}^{(k)}}\right), \quad i \varepsilon[m, N] \text { and }  \tag{8.5.12b}\\
s\left(u_{i}^{(k)}\right)=-h^{2} e^{u_{i}^{(k)}}\left(1-u_{i}^{(k)}\right) /\left(4+h^{2} e^{u_{i}^{(k)}}\right), i \varepsilon[1, N] . \tag{8.5.12c}
\end{gather*}
$$

An efficient solution of system (8.5.10) can be obtained as follows:

## Procedure I

We apply the LUBOT algorithm, solving accurately the system (8.5.10) for each value of $k$. This combination of a direct method for the solution of the inner iteration and the use of Picard or Newton iteration for the solution of the outer iteration yields the advantages of a direct method and on the other hand the technique proves to be a one-level iteration. The criterion for terminating the outer iteration was that the maximum of the absolute value of the relative change between consecutive iterations of the functional values be no greater than $\varepsilon_{\square}=10^{-6}$, i.e.,

$$
\begin{equation*}
\max _{i}\left|\frac{u_{i, j}^{(k+1)}-u_{i, j}^{(k)}}{u_{i, j}^{(k+1)}}\right| \leq \varepsilon_{\square} \tag{8.5.13}
\end{equation*}
$$

The heuristic character of this termination criterion is particularly useful, if the solution of the considered problem or effective bounds on it, are not known.

For the set of boundary conditions $U \equiv 0$, the initial guess was fixed to be $u^{(0)}=0$, while for $U \equiv 10$ on the boundary the initial guess ${ }_{u}^{(0)}=6$ was chosen.

The numerical experiments were carried out for the non-linear elliptic P.D.E's $\nabla^{2} U=e^{U}$ and $\nabla^{2} U=U^{2}$, with $h=\frac{1}{64}$ and $x_{\max }=\frac{1}{2}, y_{\max }=\frac{1}{4}$.

Numerical results for the Procedure I, with the same accuracy used in both cases, are presented in Table 8.5.1.

| $\begin{aligned} & \text { Non linear } \\ & \text { P.D.E. } \end{aligned}$ | Method - <br> Outer <br> Iteration | No. of Iterations | $\begin{aligned} & \text { Value of } \mathrm{u}^{(\mathrm{k})} \\ & \text { at }\left(\frac{1}{4}, \frac{1}{8}\right) \end{aligned}$ | $\begin{aligned} & \text { Value of } u^{(k)} \\ & \text { at }\left(\frac{1}{64}, \frac{1}{64}\right) \end{aligned}$ |
| :---: | :---: | :---: | :---: | :---: |
| $\nabla^{2} \mathrm{U}=\mathrm{e}^{\mathrm{U}}$ | B.C. $\quad \mathrm{U}=0$ |  |  |  |
|  | Picard | 4 | -0.00707 0716588 | -0.00041 00262 |
|  | Newton | 3 | -0.00707 0716559 | -0.00041 00262 |
|  | B.C. $\mathrm{U} \equiv 10$ |  |  |  |
|  | Picard | (1)SC) | - | - |
|  | Newton | 6 | 5.6599502703 | 8.8197540932 |
| $\nabla^{2} \mathrm{U}=\mathrm{U}^{2}$ | B.C. $\mathrm{U} \equiv \mathrm{O}$ |  |  |  |
|  | Picard | 4 | 0 | 0 |
|  | Newton | 4 | 0 | 0 |
|  | B.C. $\mathrm{U} \equiv 10$ |  |  |  |
|  | Picard | 7 | 9.3589418443 | 9.9606922215 |
|  | Newton | 4 | 9.3589414802 | 9.9606922144 |

## TABLE 8.5.1

Procedure I. Numerical results for Picard and Newton iterations for the non-linear P.D.E's $\nabla^{2} U=e^{U}$ and $\nabla^{2} U=U^{2}$. The values of the solution at the centre and near the origin of the considered rectangular region are given.
(OSC) In the case of $\nabla^{2} U=e^{U}$, with $U \equiv 10$ on the boundary, although a sufficiently close initial guess to the solution was given, it was found that the conditions for local convergence of the Picard method are not satisfied.

Procedure II

In this case the composite iterative scheme is given by

$$
\begin{equation*}
L_{s} U_{s}\left(\underline{u}_{(r+1)}^{(k+1)}-\underline{u}(k)\right)=a \underline{r}_{(r)}^{(k+1)} \tag{8.5.14}
\end{equation*}
$$

where the superscript $k$ denotes the outer iteration index and the subscript $r$ the inner iteration index, $L_{S}$ and $U_{S}$ are respectively lower and upper sparse triangular matrices of the form (4.2.2), appropriate to the particular method (Picard or Newton method), $\underline{(r)}(k+1)$ is the well known residual factor $\underline{r}_{(\underline{r})}^{(k+1)}=\underline{s}-\underline{\operatorname{Au}}(\mathrm{r})(\mathrm{r})$ ) and a is a pre-determined acceleration parameter. The inner iteration is now carried out by the ALUBOT algorithm given a sufficient near approximate solution. At this stage, it should be pointed out that
(i) if the approximation to the solution obtained by ALUBOT is not sufficiently good i.e., an improper choice of convergence criteria or value of parameter $a$, then $a$ good approximation to the original non-linear elliptic P.D.E. cannot be achieved from (8.5.14),
(ii) if $\underline{u}^{(k+1)}$ is not a good approximation to the solution of equation (8.5.1), there is no need to obtain a very accurate solution of (8.5.14).
(iii) the better the inner iteration process, the better the overall method.

The termination criterion for the outer loop was the same as in Procedure I, while as stopping criterion for the inner loop the mixed error test was used, i.e.,

$$
\begin{equation*}
\frac{\left|\underline{u}_{(\mathrm{r}+1)}^{(\mathrm{k}+1)}-\underline{u}_{(\mathrm{r})}^{(\mathrm{k}+1)}\right|}{\left|\underline{u}_{(\mathrm{r})}^{(\mathrm{k}+1)}\right|+1}<\varepsilon_{\mathrm{I}} \tag{8.5.15}
\end{equation*}
$$

where the accuracy $\varepsilon_{\mathrm{I}}$ was initially taken $\varepsilon_{\mathrm{I}}=10^{-2}$ and then was decreased at each iterative step by $\varepsilon_{\mathrm{I}} / 10$ up to $10^{-6}$ where it remained constant during the next iterative steps.

For the Picard iteration, in the case of boundary conditions $\mathrm{U} \equiv 0$ an optimal $\mathrm{a}=1.02$ was obtained with the overall number of iterations 21 , the number of outer iterations 12 (denoting how many times the criterion (8.5.13) has been applied) and the value at the point $(1 / 4,1 / 8)-0.0070707124$ to eight significant figures. Starting the iterative process with the accuracy $\varepsilon_{\mathrm{I}}=10^{-6}$ and then constant throughout the iteration, the same optimum value of parameter a was obtained, with the overall number of iterations 27 , the number of outer iterations 9 and the value at the central point -0.00707 07203. For the Newton iteration the optimal $\mathrm{a} \simeq 1.01$ was obtained with overall number of iterations 22, number of outer iterations 13 and value at the central point -0.00707 07141.

In the case of boundary conditions $\mathrm{U} \equiv 10$, for the Newton iteration the optimal parameter $\mathrm{a}=0.96$ was found, with overall number of iterations 21 and number of outer iterations 6. The value at the central point was found 5.65995 08873. For the Picard iteration, as in Procedure I, the conditions for local convergence are not satisfied.

Numerical results for Procedure II are presented in Table 8.5.2.

| $\begin{gathered} \text { Parameter } \\ \mathrm{a} \end{gathered}$ | No. of Overall Iterations | No. of Outer Iterations | Value of $u^{(k)}$ at $\left(\frac{1}{4}, \frac{1}{8}\right)$ | Value of $u^{(k)}$ at $\left(\frac{1}{64}, \frac{1}{64}\right)$ |
| :---: | :---: | :---: | :---: | :---: |
| B.C. U\#O , PICARD METHOD |  |  |  |  |
| 1.27 | 89 | 48 | -0.00707 0716405 | -0.00041 0026195 |
| 1.26 | 80 | 43 | -0.00707 0716766 | -0.00041 0026206 |
| 1.22 | 56 | 31 | -0.00707 0716730 | -0.00041 0026201 |
| 1.18 | 43 | 25 | -0.00707 0716502 | -0.00041 0026204 |
| 1.14 | 34 | 21 | -0.00707 0716670 | -0.00041 0026196 |
| 1.10 | 28 | 18 | -0.00707 0716647 | -0.00041 0026195 |
| 1.08 | 26 | 17 | -0.00707 0716602 | -0.00041 0026196 |
| 1.06 | 23 | 15 | -0.00707 0716072 | -0.00041 0026202 |
| 1.02 | 21 | 12 | -0.00707 0712353 | -0.00041 0026130 |
| 1.00 | 22 | 13 | -0.00707 0713317 | -0.00041 0026144 |
| 0.98 | 22 | 13 | -0.00707 0711667 | -0.00041 0026116 |
| 0.96 | 23 | 14 | -0.00707 0712618 | -0.00041 0026132 |
| B.C. U三0 , NEWTON METHOD |  |  |  |  |
| 1.20 | 48 | 27 | -0.00707 0716717 | -0.00041 0026199 |
| 1.16 | 38 | 23 | -0.00707 0716673 | -0.00041 0026197 |
| 1.14 | 34 | 21 | -0.00707 0716665 | -0.00041 0026196 |
| 1.12 | 31 | 19 | -0.00707 0716538 | -0.00041 0026206 |
| 1.10 | 28 | 18 | -0.00707 0716642 | -0.00041 0026195 |
| 1.08 | 25 | 16 | -0.00707 0716451 | -0.00041 0026208 |
| 1.06 | 23 | 15 | -0.00707 0716007 | -0.00041 0026200 |
| 1.02 | 22 | 13 | -0.00707 0714120 | -0.00041 0026157 |
| 1.00 | 22 | 13 | -0.00707 0712872 | -0.00041 0026136 |
| 0.96 | 23 | 13 | -0.00707 0712151 | -0.00041 0026124 |
| 0.92 | 24 | 14 | -0.00707 0711067 | -0.00041 0026105 |
| 0.90 | 25 | 14 | -0.00707 0711867 | -0.00041 0026119 |

TABLE 8.5.2
Procedure II. Numerical results of Picard and Newton iterations for the non-linear P.D.E. $\nabla^{2} U=e^{U}$, when $U \equiv 0$ on the boundary. The values of the solution at the centre and near the origin of the considered rectangular region are given.

| Parameter <br> a | No. of Overall Iterations | No, of Outer Iterations | Value of $u^{(k)}$ at $\left(\frac{1}{4}, \frac{1}{8}\right)$ | Value of $u^{(k)}$ at $\left(\frac{1}{64}, \frac{1}{64}\right)$ |
| :---: | :---: | :---: | :---: | :---: |
|  | B.C. Uミ10 |  | NEWTON METHOD |  |
| 1.28 | 42 | 6 | 5.6599502715 | 8.8197541595 |
| 1.24 | 37 | 6 | 5.6599502570 | 8.8197541075 |
| 1.20 | 33 | 6 | 5.6599502903 | 8.8197540193 |
| 1.16 | 30 | 6 | 5.6599502821 | 8.8197541197 |
| 1.12 | 25 | 6 | 5.6599503298 | 8.8197540804 |
| 1.08 | 25 | 6 | 5.6599503729 | 8.8197540815 |
| 1.04 | 22 | 6 | 5.6599504826 | 8.8197540870 |
| 1.00 | 22 | 6 | 5.6599506378 | 8.8197540920 |
| 0.96 | 21 | 6 | 5.6599508873 | 8.8197540953 |
| 0.92 | 23 | 6 | 5.6599506605 | 8.8197540947 |
| 0.88 | 23 | 6 | 5.6599509940 | 8.8197540968 |
| 0.84 | 25 | 6 | 5.6599508730 | 8.8197540990 |

TABLE 8.5.3
Procedure II. Numerical results of Newton iteration for the non-linear P.D.E. $\nabla^{2} U=e^{U}$, when $U \equiv 10$ on the boundary.

From a comparison of numerical results obtained by application of a composite Newton-S.O.R. iterative scheme given in [4] and experimental results obtained by applying the Newton-ALUBOT iterative scheme (see Tables $8.5 .2,8.5 .3$ ) it can be easily seen that for optimal parameters, the former iterative scheme takes up to two times as many iterations as the latter composite method. This result must be considered in the light of the computational effort required to complete the inner iteration which is nearly twice for the latter than the former.

Note that for the Picard iteration the coefficient matrix of system (8.5.10) is factorized once, while the Newton iteration requires the
coefficient matrix to be factorized every iterative step. A reasonable approach, dictating further experimentation, to tackle this problem is the consideration that the coefficient matrix is factorized after a certain number of iterations.

Remarks
(i) The applicability of the LUBOT-ALUBOT algorithms in the nonlinear equations can be furthermore extended deriving analogous combined methods, in which the inner iteration is represented by one of LUBOT, ALUBOT algorithms and the outer iteration is one of the Secant or Steffensen methods.
(ii) It is expected that a non-linear ALUBOT iteration can be derived (a reversed composite scheme to Newton-ALUBOT method), in which the one-dimensional Newton method can be used as inner iteration and the ALUBOT as outer iteration.
(iii) It can be easily seen that for a class of mildly non-linear self-adjoint P.D.E.'s the NOBAR algorithm can be incorporated as an inner iteration into (8.3.6), giving rise to a NewtonNOBAR method, (similarly the Secant-NOBAR, Steffensen-NOBAR methods can be derived).
(iv) Finally the extension of the foregoing combined methods to three space dimensions can be considered using the LUBOT-3D, ALUBOT-3D and NB3D,NOBAR-3D (for the self-adjoint case) algorithmic procedures for the solution of the linear system (8.3.11).

## Chapter 9

## CONCLUDING REMARKS AND DISCUSSION

In this thesis, the aim was to derive direct and iterative algorithmic procedures for the solution of Elliptic and Parabolic P.D.E.'s, in both two and three space dimensions using finite difference approximations.

The particular strategies of the derived algorithmic procedures are that more outermost off-diagonal entries are retained in the solution process in comparison with known variants of Stone's strongly implicit procedure, and which enables one to obtain faster convergence rates.

These new algorithmic procedures are the central framework for the efficient solution of boundary value problems as already discussed in Chapters 2,7 and 8.

The researches have shown that extremely powerful convergent methods for the solution of large scale problems in scientific computing can be derived. To offset against these advantages are the extra computational work involved in the algorithmic procedures.

Generally speaking we can say that the application of these new computational and algorithmic techniques can be extended to the solution of more general boundary value problems under different boundary conditions in which the region under consideration is also of general shape.

Finally, it should be mentioned that further research work on these topics includes:
(i) Extension of the derived direct and iterative algorithmic procedures for the finite difference of Elliptic and Parabolic P.D.E.'s in $n$ dimensions ( $n>3$ ).
(ii) Extension of the presented algorithmic procedures such that more complicated banded matrices derived from finite element discretization, i.e.,

can be factorized exactly and approximately to yield direct and iterative algorithmic procedures for the finite element solution of boundary value problems.
(iii) Combinations of the so-obtained algorithmic procedures with linearized and quasilinearized iterative schemes for the solution of complicated non-linear problems in both two and three space dimensions.

REFERENCES
[1] Ames, W.F.(1965): "Nonlinear partial differential equations in engineering", Academic Press, Inc., New York.
[2] Axelsson, 0. (1974): "On preconditioning and convergence acceleration in sparse matrix problems", report CERN 74-10 of the CERN European Organization for Nuclear Research, Data Handling Division, Laboratory 1.
[3] Bellman, R. (1960): "Introduction to Matrix Analysis", McGraw-Hill Book Co., Inc., New York.
[4] Bellman, R., Juncosa, M.L., Kalaba, R. (1961): "Some numerical experiments using Newton's method for nonlinear parabolic and elliptic boundary-value problems", C.A.C.M., Vol.4, pp.187-191.
[5] Benson, A., Evans D.J. (1976): "A normalized algorithm for the solution of positive definite symmetric quindiagonal systems of Zinear equations", A.C.M. Trans. on Math. Soft., Vol.3, pp.96-103.
[6] Benson, A., Evans, D.J.(1977): "A normalized algorithm for the solution of symmetric general five-term linear systems of semibandwidth $m^{\prime \prime}$, to be published.
[7] Bracha-Barak, A., Saylor, P.E. (1973): "A symmetric factorization procedure for the solution of elliptic boundary value problems", SIAM J.Num.Anal., Vol. 10, No.1, pp.190-206.
[8] Budak, B.M. Samarskii, A.A., Tikhonov, A.N. (1964): "A collection of problems on Mathematical Physics", Pergamon Press.
[9] Buleev, N.I. (1960): "A numerical method for the solution of two-dimensional and three dimensional equations of diffusion", Math. Sbornik, Vol.51, No.2, pp.227-238.
[10] Chandra, R., Eisenstat, S.C., Schultz, M.H. (1975): "Conjugate gradient methods for partial differential equations", Adv. in Comp.Meth. for P.D.E.'s, A.I.C.A., pp.60-64.
[11] Collatz, L. (1942): "Einschliessungenssatz fur die characteristischen zahlen von matrizen", Math.z. 48, pp.221-226.
[12] Collatz, L. (1960): "The numerical treatment of differential equations", Springer-Verlag, Berlin.
[13] Crank, J., Nicolson, P. (1947): "A practical method for numerical evaluation of partial differential equations of the heat conduction type", Proc.Camb.Phil.Soc., Vol.43, pp.50-67.
[14] Cuthill, E.H. Varga, R.S. (1959): "A method of normalized block iteration', J.Assoc.Comp.Mach.,Vol.6, pp.236-244.
[15] Daniel, J.W. (1967): "The conjugate gradient method for Iinear and nonlinear operator equations", SIAM J., Num.Anal., Vol.4, pp.10-26.
[16] Diamond, M.A. (1972): "An economical algorithm for the solution of elliptic difference equations independent of user supplied parameters", Doctoral Thesis, University of Illinois, Urbana-Champaign, U.S.A.
[17] Douglas, J. Jr. (1961): "Alternating direction iteration for mildly nonlinear elliptic difference equations", Numer.Math., Vol.3, pp.92-98.
[18] Dupont, T. (1968): "A factorization procedure for the solution of elliptic difference equations", SIAM J.Num.Anal., Vol.5, pp.753-782.
[19] Dupont, T., Kendall, R.P., Rachford, H.H. (1968): "An approximate factorization procedure for solving selfadjoint elliptic difference equations", SIAM J.Numer.Anal., Vol.5, No.3, pp.559-573.
[20] Engeli, M., Ginsburg, T. Rutishauser, H., Stieffel, E.(1959): "Refined Iterative methods for computation of the solution and the eigenvalues of self-adjoint boundary value problems", Birkhłuser Verlag, Basel/Stuttgart.
[21] Evans, D.J. (1966): "Simplified implicit methods for the finite difference solution of parabolic and hyperbolic partial differential equations", J.Inst.Maths.Applics., Vol.2, pp.76-96.
[22] Evans, D.J. (1967): "The use of preconditioning in iterative methods for solving linear equations with symmetric positive definite matrices", J.Inst.Maths.Applics., Vol.4, pp.295-314.
[23] Evans, D.J. (1973): "The analysis and applications of sparse matrix algorithms in the finite element method", pp.427-447 of the book "The Mathematics of Finite Elements and Applications" edit. J.R. Whiteman, Academic Press, London and New York.
[24] Evans, D.J. (1974): "Iterative sparse matrix algorithms", in Software for Numerical Mathematics, edit. D.J. Evans, pp.49-84.
[25] Fadeev, D.K., Fadeeva, V.N. (1963): "Computational methods of linear algebra", W.H. Freeman and Co., San Francisco.
[26] Fadeeva, V.N. (1950): "Vychislitel' nye metody lineinoi algebry", Gostekhizdat.
(Computational methods of linear algebra)-translated by Curtis D. Benster (1959), Dover Publications Inc., New York.
[27] Forsythe, G.E., Wasow, W.R. (1960): "Finite difference methods for partial differential equations", Wiley, New York, London.
[28] Fox, L. (1962): "Numerical solution of ordinary and partial differential equations", Pergamon Press.
[29] Fox, L. (1964): "An introduction to numerical linear algebra", Oxford University Press.
[30] Gerschgorin, S. (1931): "Uber die abrenzung der eigenwerte einer matrix", Izv. Akad.Nauk. SSSR, Ser.Mat.7.
[31] Gustafsson, I. (1977): "A class of first order factorization methods", Research report 77.04R, Comp.Sc., Chalmers University of Technology and University of Göteborg.
[32] Hestenes, M.R., Stiefel, E. (1952): "Methods of conjugate gradients for solving linear systems", J.Research, Nat. Bur. Standards, Vo1.49, No.6, pp.409-436.
[33] Hildebrand, F.B. (1968): "Finite difference equations and simuZations", Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
[34] Householder, A.S. (1964): "The theory of matrices in numerical analysis", Blaisdell, New York.
[35] Isaacson, E., Keller, H.B. (1967): "Analysis of numerical methods", John Wiley and Sons Ltd.
[36] Kincaid, D.R. (1968): "On complex second-degree iterative methods", Center for Numerical Analysis, University of Texas, Austin.
[37] Knuth, D.E. (1968): "The art of computer programming", Vol.1/ "Fundamental Algorithms", Edits. R.S. Varga, M.A. Harrison, Addison-Wesley Co., Inc.
[38] Lanczos, C. (1952): "Solution of systems of Zinear equations by minimized iterations", J.Res. N.B.S., Vol.49, pp.33-53.
[39] Lee, E.L., Stone, P.M. (1958): "Numerical methods for solving linear systems and applications to elliptic difference equations", Report LA-2314, Los Alamos Scientific Laboratory, University of California, Los Alamos, New Mexico.
[40] Manteuffel, T.A. (1975): "An iterative method for solving nonsymmetric linear systems with dynamic estimation of parameters", Ph.D. Thesis, University of Illinois, UrbanaChampaign, U.S.A.
[41] Markoff, W. (1916): "Über polynome, die in einem gegebenen intervalle möglichst wenig von null abweichen", Math.Ann., Vol.77, (Translated by Grossmann J. from Russian original of 1892).
[42] 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. ut Computation, Vol.31, No.137, pp.148-162.
[43] Mitchell, A.R. (1969): "Computational methods in partial differential equations", John Wiley and Sons Ltd., Aberdeen University Press.
[44] O'Brien, G.G., Hyman, M.A., Kaplan, S. (1951): "A Study of the numerical solution of partial differential equations", J.Maths.Phys., Vol.29, pp.223-251.
[45] 0'Dell, P.M. (1974): "Numerical reservoir simulation review and state of the art', Proc. of A.I.Ch.E., $76^{\text {th }}$ Nat. Meeting.
[46] O'Leary, D.P. (1976): "Hybrid conjugate gradient algorithms", Ph.D. dissertation, Stanford University, U.S.A.
[47] Oliphant, T.A. (1962): "An extrapolation procedure for solving Zinear systems", Quarterly of Applied Math., Vol.20, No.3, pp.257-267.
[48] Ortega, J.M., Rheinboldt, W.C. (1970): "Iterative solution of nonlinear equations in several variables", Academic Press, Inc., New York.
[49] Reid, J.K. (1970): "A Fortran subroutine for the solution of large sparse sets of linear equations by conjugate gradients", Research group report, Theoretical Physics Division, Atomic Energy Research Establishment, Harwell, G.B.
[50] Reid, J.K. (1971): "On the method of conjugate gradient for the solution of large sparse systems of linear equations", Proc. Conference on "Large Sparse sets of Linear Equations", Academic Press, New York.
[51] Richardson, L.F. (1910): "The approximate arithmetical solution by finite differences of physical problems involving differential equations with application to the stress in a masonary dam", Philos.Trans.Royal.Soc., London, Sev.A, Vol. 210.
[52] Richtmyer, R.D., Morton, K.W. (1967): "Difference methods for initial value problems", Interscience (sec. edition).
[53] Saylor, P.E. (1974): "Second order strongly implicit symmetric factorization methods for the solution of elliptic difference equations", SIAM J.Num.Anal, Vol.11, No.5, pp.894-908.
[54] Smith, G.D. (1965): "Numerical solution of partial differential equations", Oxford University Press, Ely House, London.
[55] Stiefe1, E.L. (1958): "Kernel polynomials in linear algebra and their numerical application", Nat.Bur.Stan.App.Math. Ser. 49.
[56] Stone, H.L. (1968): "Iterative solution of implicit approximations of multidimensional partial differential equations", SIAM J.Num.Anal. Vol.5, pp.530-558.
[57] Varga, R.S. (1962): "Matrix Iterative Analysis", Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
[58] Wachspress, E.L. (1966): "Iterative solution of elliptic systems" Prentice Hall, Inc., Englewood Cliffs, New Jersey.
[59] Weinstein, H.G., Stone, H.L., Kwan, T.V. (1969): "Iterative procedure for solution of systems of parabolic and elliptic equations in three dimensions", I. and E.C. Fundamentals, Vol.8, No.2, pp.281-287.
[60] Wrigley, H.E. (1963): "Accelerating the Jacobi method for solving simultaneous equations by Chebychev extrapalation when the eigenvalues of the iteration matrix are complex", Comput.J. Vol.6, pp.169-176.
[61] Young, D.M. (1971): "Iterative solution of Zarge Linear systems", Academic Press, New York.
[62] Young, D.M. (1975): "Notes on the conjugate gradient method", unpublished.
[63] Young, D.M., Frank, T.G. (1963): "A survey of computer methods for solving elliptic and parabolic partial differential equations", ICC Bulletin, Vol.2, No.11, pp.3-61.

APPENDIX 1

## A TEST PROBLEM

In order to define the elements of the submatrices $r, u$ and $\tilde{d}$ from the identities $(3.5 .6),(3.5 .7)$, in the case of the NB3D algorithm (section 3.5), let us consider the following simplified structure of the coefficient matrix A of order 10 .

where
$\overline{\mathrm{D}} \equiv \operatorname{diag}\left\{\mathrm{d}_{1}, \mathrm{~d}_{2}, \ldots, \mathrm{~d}_{10}\right\}$
and
$\overline{\mathrm{T}} \equiv$



On equating (Al.4) we obtain the elements of the submatrix $r$ as follows:-

$$
\mathrm{r}_{11}=\mathrm{h}_{1} / \mathrm{d}_{1} \mathrm{~d}_{8}
$$

$$
\begin{aligned}
& r_{21}=-e_{1} r_{11} \\
& r_{31}=-e_{2} r_{21}
\end{aligned}
$$

$$
\begin{aligned}
r_{12} & =h_{2} / d_{2} d_{9} \\
r_{22} & =-e_{2} r_{12}
\end{aligned}
$$

$$
\begin{aligned}
& r_{13}=\mathrm{h}_{3} / \mathrm{d}_{3} \mathrm{~d}_{10} \\
& r_{23}=-\mathrm{e}_{3} r_{13} \\
& r_{33}=-e_{4} r_{23}-\sum_{3}^{4} t_{k 1} r_{k-2,3}
\end{aligned}
$$

$$
\begin{aligned}
& r_{43}=c_{6} d_{6} d_{10} e_{5} r_{33^{-}} \sum_{2}^{4} t_{k 2^{r}} r_{k-1,3} \\
& r_{53}=-e_{6} r_{43} \sum_{1}^{4} t_{k 3} r_{k 3}
\end{aligned}
$$



SUBROUTINE LUBOT (N,M,B,A, C, U,V)
THIS SUBRUUTINE IS A FACTORIZATION OF A SQUARE MATRIX OF OKDER N. THE COEFFICIENT MATRIX IS NON-SYMMETRICIPOSITIVE DEFIHITE AND QUINDIAGONAL OF BANDWIDTH M ( $3<M<N$ ).
B, C, AIU,V ARE VECTORS CONTAINING RESPECTIVELY THE DIAGONAL, UPPER AND LCIWER : CODIAGONAL AND MOTH DIAGONAL ELEMENTS. THE COEFFICIENT MATRIX IS FACTORIZED INTO L U WHERE LIU PWITH UNIT DIAGONAL ELEMENTSS ARE STRICTLY LOWER AND UPPER. TRIANGULAR MATRICES WITH NON-ZERO ELEMENTS IN SUB, SUPEK DIAGONAL RESPECTIVELY AND RETAINING MI=M-1 UUTERMOST OFF-DIAGONAL ENTRIES. THREE KESULT VECTORS OF LENGTH N AND TWO ARRAYS OF DIMENSION [M-1,N-M+1] ARE USED AS WORKSPACE.
$N 1=N-M+1$.
DIMENSION B(N), A(N),C(N),U(N),V(N)
COMION /BLOCK2/G(N),W(N), D(N)
CUMHON / BLOCK4/H(M1,N1);E(M1,N1)
C FACTORIZATION OF TRIDIAGONAL MATKIX AII
$W(1)=B(1)$
$D(1)=A(2)$
$G(1)=C(1) / W(1)$
DO 5 I m (9M2
$W(I)=B(I)-D(I-1) * G(1-1)$
$D(I)=A(I+1)$
$5 \quad G(I)=C(I) / W(I)$
$W(M-1)=B(M-1)-D(M-2) * G(M-2)$
CALCULATION OF ELEMENYS OF SUBMATRICES U12.L21.U22.L22.
DO O J=1:N-H+1
$E(1, J) V(J+H-1)$
$H(1, J) m \cup(J) / W(J)$
$G(M+J-2)=C(M+J-2) / W(M+J-2)$
$D(M+J-2)=A(M+J-1)$
If (J.GT.M"2) GU TO 22
DO 7 Im2.M—」
$E(I, J)=-G(!+J-2) * E(I-1, J)$
$7 \quad H(I, J)=-D(I+J-2) * H(I-1 ; J) / W(I+J-1)$
22 IF (J.EQ.I) GOTO 23
IF (J,GT,M-1) IP=2
IF $(J, b E, M-1) \quad I P=M-J+1$
DU 8 Im1P, M-1
$2=0$
DO 9 K $=1.1-1$
$9 \quad Z=Z+E(K, J) * H(K-I+M, I+J-M)$
$E(1, J)=-G(I+J-2) \star E(I-1, J)-2$
$z=0$
DO $12 K=1,1-1$
$12 \quad Z=Z+E(K-I+M, I+J-M) \neq H(K, J)$
$H(I, J)=(\rightarrow D(I+J-2) \neq H(I-1, J)-2) / W(I+J-1)$
8 CUNTINUE
$23 \quad I=M-1$
$Z=0$
DO $13 \mathrm{~K}=111$
$Z=Z+E(K, J) * H(K, J)$
13 CUNTINUE
$W(H+J-1)=B(M+J-1)-D(I+J-1) * H(I, J)-G(I+J-1) * E(I, J)-D(I+J-q) *$

* $G(I+J-1)-2$

6
CONTINUE

RETURN
END

```
    SUBROUTINE FBSUBS (N,H,S)
C THIS SUBRUUTINE SOLVES THE SET (LU)Y=S OF N LINEAR EQUATIONS.
C WHERE L,U (WITH UNIT DIAGONAL ELEMENTS) ARE STRICTLY LOWER
C AND UPPER TRIANGULAR MATRICES OF BANDWIDTH M. THE NON-ZERO
C ELENENTS ARE ON SUB, SUPER DIAGONAL ELEMENTS RESPECTIVELY AND
C RETAIN MI=M-1 OUTERIOST OFF-DIAGONAL ENTRIES. THE SOLUTION
C IS EFFECTED BY A FORWARDMBACKWARD SUBSTITUTION PRUCESS WHERE
C THE INPUT VECTOR S IS OVERWRITTEN SUCCESSIVELY BY THE
C INTERMEDIATE SOLUTION (OBTAINED BY FORWARD SUBSTITUTION) AND
C THE FINAL SOLUTION (OETAINED BY BACK SUBSTITUTION). NYZN-M+1.
    DIMENSION S(N)
    COMHON /BLOCK2/G(N),W(N),B(N)
    COMHON /BLOCK4/T(M1,N1),E(M1,N1)
    \(S(1)=S(1) / W(1)\)
    DU \(7 \quad 1=2 \% \mathrm{M}-1\)
    \(7 \quad S(I)=(S(I)-B(I-1) * S(I-1)) / W(I)\)
        DU \(8 \quad I=H_{1}^{\prime \prime} N\)
        \(2=0\)
        DO \(9 K=I m M+1, I-1\)
        \(9 \quad Z=Z+E(K-1+M, I-11+1) * S(K)\)
    \(8 \quad S(I)=(S(I)-B(I-1) * S(I-1)-2) / W(I)\)
        IF (M.LT.N/2+1) GO TO 34
        IF (M.EQ.N) GO TO 35
CP-m CASE: \(M>[N / 2+1]\).
    DO 10 II \(=1, N-M\)
        \(!=N-I 1\)
        \(Z=0\)
        0U \(11 J=1+1, N\)
        \(Z=Z+T(I-d+H, J-H+1) * S(J)\)
    11 CONTINUE
    \(10 S(I)=S(1)-G(I) * S(I+1)-2\)
    \(350012 \quad I=1,2 * M-N-1\)
        \(1=M-11\)
        \(Z=0\)
        DO \(13 \quad J=M, N\)
        \(Z=Z+T(I-J+M, J-M+1) * S(J)\)
    13 CONTINUE
    \(12 S(I)=S(I)-G(I) * S(I+1)-2\)
        IF (M,EO.N) GO TO 36
        DU \(14 \quad I I=1, N-M\)
        \(I=N-M+1-I I\)
        Z=0
            DO \(15 \mathrm{~J}=\mathrm{M}_{1} \mathrm{I}+\mathrm{M}-1\)
        \(Z=Z+T(I-J+M, J-M+1) * S(J)\)
        5 CONTINUE
    \(14 S(I)=S(I)-G(I) * S(I+1)-2\)
        GO TO 36
    34 IF ( \(((N-N / 2 * 2), E Q, 0)\).AND. (M,EQ.N/2)) GOTO 37
Cm- CASE: \(M<[N / 2+1]\).
        DO \(16 \quad 11=1, \mathrm{M}-1\)
        \(I=N-I I\)
        \(2=0\)
        DO \(17 \mathrm{~J}=1+1, N\)
        \(Z=Z+T(I-J+M, J-M+1) * S(J)\)
    CONTINUE
```

```
    16 S(I)=S(I)-G(I)*S(I+1)-Z
        D0 1& II=1,N-2*M+1
    I=N-M+1-II
    2=0
    DO 19 J=I+1,I+M-1
    Z= Z+T(I-J*H,J-M+1)*S(J)
    19 CUNTINUE
    18S(I)=S(I) mG(I)*S(1+1) mZ
    DO 20 \I=1,M=1
    I=MmII
    Z=0
    DO 21 J=M,I+M-1
    Z= Z+T(Imd+H,J-M+1)*S(J)
    21 CONTINUE
    20 S(I)=S(I)-G(I)*S(I+1)-2
    GO TO 36
C-m CASE:N IS EVEN AND M=N/2.
    37 DO 22 |I=1,N-H-1
    1=N-11
    Z=0
    DO 23 JmI*1,N
    23 Z= Z+T(I-J+M,J-H+1)*S(J)
    22.S(I)=S(I)-G(I)*S(I+1)-Z
    I=M
    z=0
    DO 24 JmM+1,N-1
    24 2= Z+T(1-N*H,J-H+1)*S(J)
    S(I)=S(I)-G(I)*S(I+1)-z
    DO 25 II=1,M-1
    I=11-II
    Z=0
    DO 26 J=M,I*M-1
    26 Z= Z+T(I-J+M,J-M+1)*S(d)
    25 S(I)=S(I)-G(I)*S(I+1)-Z
    36 CONTINUE
        RETURN
    END
```

SUBROUTINE LUBOT3D(N,P,M,A,B,C,U,V,R,S)
THIS SUBRUUTINE IS A FACTORIZATION OF A SQUARE MATRIX OF OKDER N. THE COEFFICIENT MATRIX IS NON-SYMMETRIC, POSITIVE DEFINITE, SEVEN DIAGDNAL UF BANUWIDTHS M AND P ( $4<M \angle P<N$ ). B, C, A, U,V,R,S ARE VECTORS CONTAINING RESPECTIVELY THE DIAGONAL.UPPER AND LOWER: CO DIAGONALIM-TH DIAGONAL AND P-TH DIAGUNAL ELEMENTS. THE COEFFICIENT MATRIX IS FACTORIZED INTU L U WHERE L, U (WITH UNIT DIAGONAL ELEMENTS) ARE STRICTLV LUWLR AND UPPER TRIANGULAR MATRICES WITH NON-ZERO ELEMENTS IN SUL, SUPER DIAGONAI. RESPECTIVELY,RETAINING M1=M-1; M2=P-1 OUTERMOST OFF=DIAGONAL ENTRIES. THREE RESULT VECTORS OF LENGTH N AND FOUR ARRAYS OF UIMENSIUN [M-1,N-M+1] AND
[P-1, N-P*1] ARE USED AS WORKSPACE, N1=N-M+1, N2=N-P+1. $D I M_{E} N S I O N A(N), B(N), C(N), U(N), V(N), R(N), S(N)$
INTEGER P
CUMHON /BLOCK2/G(N), O(N),W(N)
CUMHON H (M1,N1),T(M2,N2),E(M1,N1),F(M2,N2)
$W(1)=B(1)$
$D(1)=A(2)$
$G(1)=C(1) / W(1)$
DU $5 \quad I=2: \mathrm{M}-2$
$W(I)=B(I)-D(I-1) * O(I-1)$
$D(I)=A(I+1)$
$G(I)=C(I) / W(I)$
$W(M-1)=B(M-1)-D(M-2) * G(M-2)$
DO 0 小=1:"p-11
$E(1, J)=V(J+M-1)$
$H(1, J)=U(J) / W(J)$
$G(M+J-2)=C(11+J-2) / W(H+J-2)$
$D(M+J-2)=A(H+J-1)$
IF (J,GT.M-2) GOTO4
DO $71=2, M-J$
$E(1, J)=-G(I+J-2) * E(I-1, J)$
$7 H(I, J)=-D(I+J-2) * H(I-1, J) / W(I+J-1)$
4 IF (J.EQ.1) GO TO 18
IF (J.GT.Mm1) IP=2
IF $(J \cdot L E \cdot M-1) \quad I P=M-J+1$
DU 8 IDIP, M-1
$Z=0$
DO $3 \mathrm{~K}=1 \mathrm{~T}=1$
$3 \quad Z=Z+E(K, J) * H(K-I+M, I+J-M)$
$E(1, J)=-G(1+J-2) * E(1-1, J)-2$
$Z=0$
DO $12 K=1,1-1$
$12 \quad 2=2+E(K-I+M, I+J-M) * H(K, J)$
$H(1, J)=(-D(1+J-2) \neq H(I-1, J)-L) / W(I+J-1)$
8 CUNTINUE
$18 \quad I=M-9$
Z=0
DO $13 K=1$,I
$132=2+E(K, j) \neq H(K, J)$
$W(M+J-1)=B(H+J-1)-D(I+J-1) * H(1, J)-G(I+J-1) * E(I, J)-D(I+J-1) *$

* $G(I+J-1)-2$

CONTINUE
GENERATION DF ELEMENTS OF SUBMATRICES U12.L21.
DU $22 J=1, N-P+1$

```
    T(1,J)=R(J)/W(J)
    G(P+J-2)=C(P+J-2)/W(P+J-2)
    F(1,J)=S (P+J-1)
    D(P+J-2)=A(P+J-1)
    IF (J.GT.YM-2) GO TO 17
    DO 23 I=2,H-J
    T(I,J) }=-D(I+J-2)*T(I-1,J)/W(I+J-1
    F(I,J)m-G(I+J-2)*F(I-1,J)
    IF (J.E(R.1,AND.I.EQ.P-M+1) GO TO 55
    GU TO 23
    55 T(I,J)=T(I,J)+1I(I+J-1)/U(I+J-1)
    F(I,J)=F(I,J)+V(P+J-1)
    23 CUNTINUE
    17 IF (J,GT,M-1) IP=2
    Ir (J.LE.M-1) IPEM-J+1
    DO 24 I =IP,P-1
    IF (I.GE.PMJ+1.OR.J.GT.Pm2) GO TO 66
    z=0
    Z1=0
    IF (I.LT.M) GO TO 25
    DU 28 K=1,M-1
    Z=Z+E(K,I+J-M)*T(K+I-M,J)
    28 2i=L\+F(K+I-M,J)*H(K,I+JmM)
    GO TO 27
    25 DO 26 K=1,I-1
    Z=Z+E(K-I+N,I+J-M)*T(K,J)
    26 21=L1+F(K,J)*H(K-I+M,I+J-M)
    27T(I,J)=(-D(I+J-2)*T(I-1,J)-2)/W(I+J-1)
        F(I,J)--G(I+J-2)*F(I-1,J)-<1
        IF (I.EQ:M-M+ 1) GO TO 53
        GO TO 24
    53T(I,J)=T(I,J)+U(I+J-1)/W(I+J-1)
        F(I,J) - F (I,J) +V (P*J-1)
        GO TO 24
G GENERATION OF ELEMENTS OF SUBMATRICES L22,U22.
    60 2=0
        21=0
        DO 29 K=1,I-1
        Z=Z+F(K-I+P,I+J-P)*T(K,J)
    29 21=21+F(K,J)*T(K-I+P,I+J mP)
        T(I,J)=(-D(I+J-2)*T(I-1,J)-Z)/W(I+J-1)
        F(I,J)=-G(I+J-2)*F(I-1,J)-L1
        IF (I.EQ.PmH+1) GO TO 2
        GO TO 24
    2T(I,J)=T(I,J)+U(I+J-1)/W(I+J-1)
    F(I,J)=F(I,J)+V(P*J-1)
    24 CONTINUE
    33 I=P-1
        Z=0
        DO 31 K=1,1
    31 Z=2+F(K,J)*T(K,J)
        W(P+J-1)=B(P+J-1)-D(I+J-1)*T(I,J)-D(I+J-1)*G(I+J-1)-
        *G(I+J-1)*F(I,J)-Z
        CUNTINUE
        RETURN
        END
```

SUBROUTINE FBSUBS3D(N,P,M,S)
THI'S SUBRUUTINE SOLVES THE SET (IU)Y=S OF N LINEAR EQUATIONS,
WIIERE L,U (WITH UNIT DIAGONAL EIEMENTS) ARE STRICTLY LOWER
AIND UPPER TRIANGULAK MATRICES OF BANDWIDTHS M AND P. THE
NUN-ZERG ELEMENTS ARE ON SUB, SUPER DIAGONAL ELEMENTS
RESPECTIVELY, RETAINING M1 $=M-1$ AND $12=P-1$ OUTERMOST OFF-
DIAGONAL ENTRIES. THE SOLUTION IS EFFECTED BY A FORWARD-
BACKWARD SUBSTITUTIUN PROCESS WHERE THE INPUT VECTOR S IS
OVERWKITTEN SUCCESSIVELY BY THE INTERMEDIATE SOLUTION
(UBTAINED BY FORWARD SUBSTITUTIUN) AND THE FINAL SOLUTIUN
(UBTAINED BY BACK SUBSTITUYION). $N 1=N-M+1, N 2=N-P+1$.
DIMENSIUN S(N)
COMMON /BLOCK2/GS(N),B(N),W(N)
COMMON H (M1,N1), T(M2,N2),E(M1,N1),F(M2,N2)
I HTEGER P
$S(1)=S(1) / W(1)$
DO $44 \quad t=2, M_{-1}$
$44 \quad S(1)=(S(I) m B(I-1) * S(I-1)) / W(1)$
If (M.EQ.N) GO TO 396
DO 78 I $=M, \quad P=1$
$2=0$
DO $79 \quad K=I-M+1, I-1$
$79 \quad 2=2+E(K-I+M, I-M+1) * S(K)$
$78 S(1)=(S(I)-B(1-1) * S(I-1)-2) / W(I)$
396 DO 87 d= P,N
2-1)
DU $88 \quad K=I-M+1, I-1$
$88 \quad Z=Z+E(K-I * M, I-M+9) * S(K)$
$21=0$
D0 $89 K=I=\quad P+1, I m 1$
$\begin{array}{ll}89 & Z 1=21+F(K-I+P, I-P+1) * S(K) \\ 87 & S(I)=(S(I)-B(I-1) * S(I-1)-\end{array}$
IF (M.LT: $(N / 2+1))$ GOTO 64
IF (M,EQ.N) GO TO 65
IF (P,GT. (N/2+1)) GO TO 183
Cr-- CASE: M>[N/2+1], P DOES NOT EXIST
DO 47 II $=9, N-M$
1=N-11
$2=0$
DO $48 \quad J=I+1, N$
$Z=L+H(I-J+M, J-M+1) * S(J)$
$47 S(I)=S(I)-G S(I) * S(I+1)-Z$
65 DO 49 d $I=1,2 * M-N-1$
$1=M-11$
$Z=0$
DO $51 \quad \lambda=M, N$
$z=2+H(d-J+M, J-M+1) * S(J)$
$\begin{array}{ll}51 & S(1)=S(I)-G S(1) \star S(1+1)-2\end{array}$
IF (M,EQvN) GO TO 66
DO 52 II $=1, \mathrm{~N}=\mathrm{M}$
$I=N-M+1-I I$
$z=0$
DO $53 \mathrm{~J}=\mathrm{M}, I+\mathrm{M}-1$
$53 \quad 2=2+H(I-J+M, J-M+1) * S(J)$
$52 S(I)=S(I)-G S(I) * S(I+1)-2$
GU TO 60
Cr-- $C A S E: \|>[N / 2+1] ; P>[N / 2+1]$
183 IF (P,EQ.N) GO TO 252
DO $231 I I=1, N-P$

```
    I=N-I|
    2=0
    21=0
    DU 232 Jal+1,N
    Z=2+H(I-J+M,J-M+1)*S(J)
    232 21=21+Y(1-J+P,J-P+1)*S(J)
    231 S(I)=S(1)-GS(I)*S(1+1)-2-21
    252 IF (M.EQ.N) GO TO 253
    DU 233 11=1,P-M
    I=0-II
    z=0
    DO 234 J=1+1,N
    234 2=2+H(1-j+M,J-M+1)*S(J)
    21=0
    DO 23) J=ANN
    235 21=<1+T(I-N+P,J-P+1)*S(J)
    233 S(I)=S(I)-GS(I)*S(I+1)-2-21
    253 DO<36 II =N-M+1,M-1
    I=N-II
    z=0
    21=0
    DO 237 J=H1N
    Z=2+H(I-j+M,J-M+1)*S(J)
    237 21=21*T(I-J*P,J-P*1)*S(J)
    236 S(I)=S(I)-GS(I)*S(I+1)-2-21
        IF (M.EQ.N) GO TO 66
        DO 238 II=H,P-1
    I=N-II
    z=n
    DO 239 J=A,I I M-1
    2392=2+H(I-J+M,J-M+1)*S(J)
    Z1=0
    DO 241 J=M,N
    241 21=21*T(I-J+P,J-P*1)*S(J)
    238 S(I)=S(I)-GS(I)*S(I+1)-2-21
    IF (P.EQ.N) GO TO 60
    DO 242 II=P,N-1
    I=N-1!
    Z=0
    z1=0
    DO 243 J=M,I+M-1
    Z=Z+H(I-J+M,J-M+1)*S(J)
    243 21=21+T(1-J+P,J-P+1)*S(J)
    242 S(I)=S(I)-GS(I)*S(I+1)-2-21
    gu ro 6o
    64 IF (P.GT&N.OR.P.LE.M) GO TU 821
    IF (P .LT.(N/2+1)) riO TO 444
    IF(((N-N/2*2),EQ,0).AND.(M,EQ,N/2)) GO TO 67
CH-- CASE: M<[N/2+1]; P>[N/2+1].
    If (P.EQ.N) GO TO 254
    DU 94 II=1,N- P
    I=N-II
    z=0
    21=0
    DU 95 J=1+1,N
    zi=21+T(1-J+ P,J% P+1)*S(J)
    95 z=z+H(I-j+M,J-M+1)*S(J)
    04 S(1)=S(I)-GS(I)*S(I+1)-z-<1
    254 DU 96 11=N- P+1%M-1
    I=N-II
```

```
    z=0
    DO 97 J=1+1,N
    97 Z=2+H(1-J+M,J-M+1)*S(J)
    z1=0
    DO 98 J= P,N
    98 2:=21+T(1-J* P,Jm P+1)*S(J)
    96 S(I)=S(I)-GS(I)*S(I+1)-2-21
    DU 101 II =M,N-M
    I=N-1I
    z=0
    Du 102 J=I+1,I+H-1
    102 2=2+H(1-J+M,J-M+1)*S(J)
    21=0
    DO 10S y= P,N
    103 21=21+T(I-J+ P,Jm P+1)*S(J)
    101 S(I)= S(I)mGS(I)*S(I+1)-z-21
    DU 104 II=N-M+1, P-1
    I=N-11
    z=0
    DO 105 J=M,I+M-1
105 z=z*H(1-j+M,J-M+1)*S(J)
    2'1=0
    DO 106 J= P,N
106 21=21+T(I-J* P,Jm P+1)*S(J)
    104 S(I)=S(I)mGS(I)*S(I+1)-2-21
    IF (P,EQ.N) GO TO 60
    D0 121 1I= P,N-1
    I=N-II
    z=0
    DO 122 J=M,I+M-1
    122 z=2+H(I-J+M,J-M+1)*S(J)
    Z1=0
    DO 123 J= P,I+ P-1
    123 21=21+T(I-J* P,j- P+1)*S(J)
    121 S(I)=S(I)-GS(I)*S(I+1)-2-21
    GO TO 66
C&-- CASE:M=N/2 ,P>[N/2+1].
    67 IF (P,EQgN) GO TO 256
    DU o1 1IF1,N-P
    I=N-Id
    z=0
    21=0
    DO 62 J=I+1,N
    Z=2+H(I-J+M,J-M+1)*S(J)
    o2 Z1=21+T(I-J+P,J-P+1)*S(J)
    61 S(1)=S(1)-GS(I)*S(1+1)-2-21
    256 DO 141 II=1,P-M-1
    I=P-1I
    z=0
    DU 142 JmI+1,N
    142 z= z= z+H(I-JN+M,J-M+1)*S(J)
    DO 145 JmP,N
    145 21=21+T(1-j+P,J-p+1)*S(J)
    141 S(I)=S(I)-0S(I)*S(I+1)-2-21
        1=M
    z=0
    DO 140 J=M+1,N-1
146 Z = Z +H(J-J+M,J-M+1)*S(J)
    21=0
```

```
    DO 147 J=P,N
    147 21= L1+T(I-d*P,J-p*1)*S(J)
    S(I)=S(I)-GS(I)*S(I+1)-Z-21
    DO 14% II=1,M+P=N=1
    I=M-1\
    Z=0
    DO 149 J=N, 1+M-1
    149 2=2+H(I-J+M,J-M+1)*S(J)
    Z1=0
    DU 158 JmP,N
    158 21=21+T(I-J+P,J-P+1)*S(J)
    148 S(I)=S(I)-GS(I)*S(I+1)-Z-21
    IF (P,EQ:N) GO TO 66
    DO 152 II=P,N-1
    I=N-1\
    z=0
    DO 156 J=H,I+M-1
    156 2=Z+H(I-J +M,J-M+1)*S(J)
        21=0
    DO 154 J=F,I +P-1
    154 21=21+T(I-J*P,J-P+1)*S(J)
    152S(1)=S(I)-GS(I)*S(I +1)-Z-21
    GO TO 66
C=-m CASE: M<&N/2+1], P<[N/2+1].
    44 IF (((N-N/2*2).EQ.0).AND.( P.EQ.N/2)) GO TO 107
    P=I+1,Q=N,\quadP1=I*1,Q1=N
    DO 54 II=1,M=1
    I=N-II
    z=0
    Z1=0
    DO 55 J=I+1,N
    Z=Z+H(I-J+M,J-M+1)*S(J)
    21=21+T(1-J* P;J* P+1)*S(J)
    55 CONTINUE
    54 S(I)=(S(I)-GS(I)*S(I+1)-2-Z1)
    DO 56 IImM, p-1
    I=N-I!
    Z=0
    DO 57 J=I+1,I+M-1
    57 Z Z Z + H(I-J+M,J-M+1)*S(J)
    57 CONTINUE
        21=0
    0091 v=1+1,N
    91 21=21+T(I-J* P,j* p+1)*S(J)
    50 S(I)=S(I)mGS(I)*S(I+1)-Z-Z1
    DO124 II= P,N-P
    I=N-I|
    2=0
    D0 125 JmIt+1,I+Mm1
    125 2= Z+H(I-J+H,J-M+1)*S(J)
    21=0
    DO120 J=I*1,I+ P-1
    126 21=21+T(1-J+ P,J- P+1)*S(J)
    124 S(I)=S(I)mGS(I)*S(I+1)-2-21
    DO 12% 1I=N- P+1,N-M
    I=N-II
    z=0
    DO 128 J=I*1,I+M-1
128Z=Z+H(I-J+M,JmM+1)*S(J)
    Z1=0
```

```
        DU \(124 \mathrm{~J}=\mathrm{P}, I+\mathrm{I}+\mathrm{P}\)
    129 21=21+T(I-J+ p,j-p+1)*S(J)
    127S(1)=S(1)-GS(I)*S(1+1)-2-21
        D0 58 II \(=1,11-1\)
        \(1=\mathrm{Mm} 11\)
        \(z=0\)
        DU \(59 \mathrm{~J}=\mathrm{M}, 1+\mathrm{M}-1\)
    \(54 \quad Z=2+H(1-J+M, J-M+9) * S(J)\)
        \(21=0\)
        DU \(93 \mathrm{~J}=\mathrm{P} 1\) I Pm
    \(9321=21+T(1-J+p, j=p+1) * S(J)\)
    \(58 \quad S(1)=S(1)-G S(1) * S(1+1)-2-21\)
    GU TO 60
Co-- CASE: \(\quad \mathrm{K}\langle[\mathrm{N} / 2+1], \mathrm{P}=\mathrm{N} / 2\)
    1U7 DO 108 II=1,N- \(p-1\)
        \(1=\mathrm{N}-11\)
        \(z=0\)
        Z1=0
        DO \(109 \mathrm{~J}=\mathrm{I}+1\), N
        21=21+T(I-J+ P, J= \(p+1) * S(J)\)
    \(109 z=2+H(1-j+M, J-M+1) * S(J)\)
    \(108 \mathrm{~S}(\mathrm{I})=\mathrm{S}(\mathrm{I})-\mathrm{GS}(\mathrm{I}) * S(\mathrm{I}+1)-2-21\)
        \(1=p\)
        \(z=0\)
        po \(111 \mathrm{Jal}+1, \mathrm{~N}\)
    \(1112=2+H(1-J+M, J-M+1) * S(J)\)
        \(21=0\)
        DU \(112 \mathrm{~J}=\mathrm{P}+1, \mathrm{~N}=1\)
    112 21=21+T(I-J+ PoJ- \(\rho+1\) )
    \(S(I)=S(I)-G S(I) * S(I+1)-2-21\)
    D0 \(113 \mathrm{IFN} \quad \mathrm{P}+1, \mathrm{~N}-\mathrm{M}-1\)
    \(I=N-I I\)
    Z=0
    DO \(114 \mathrm{~J}=\mathrm{I}+1\), N
    114. \(2=2+H(1-J+M, J-M+1) * S(J)\)
    \(Z 1=0\)
    DU \(115 \mathrm{~J}=\mathrm{PaI}+\mathrm{P}-1\)
```



```
    \(113 \mathrm{~S}(\mathrm{I})=\mathrm{S}(\mathrm{I}) \mathrm{mS}(\mathrm{I}) * S(1+1)-2-21\)
        \(1=1\)
    \(z=0\)
    DU \(110 \mathrm{~J}=\mathrm{M}+1, \mathrm{~N}-1\)
    \(1162=2+H(d-J+M, J-M+1) * S(J)\)
    \(21=0\)
    DO i17 J= Polt P-1
    117 21:21+T(1-J* \(P, J-P+1) * S(J)\)
    \(S(I)=S(I)-G S(I) * S(I+1)-2-21\)
    DO \(118 \quad 1 I=N-M+1, N-1\)
    \(1=\mathrm{N}-\mathrm{I}\) I
    \(\mathrm{z}=0\)
    DO \(119 \mathrm{~J}=\mathrm{M}, 1+\mathrm{M}-1\)
    \(119 \mathrm{z}=2+\mathrm{H}^{(1-J+M, J-M+1) * S(J)}\)
    Z1 \(=0\)
    DO \(131 \mathrm{~J}=\mathrm{Pil}+\mathrm{pm}\)
    \(13121=21+T(1-J+\quad\) P, J. \(\quad\) +1)*S(J)
    \(118 \mathrm{~S}(\mathrm{I})=\mathrm{S}(\mathrm{I})-\mathrm{GS}(\mathrm{I}) * S(1+1)-2-21\)
    60 ro 60
    821 IF ( \((N-N / 2 * 2) . E Q, O \cdot A N D \cdot H \cdot E Q \cdot N / 2) G 0\) TO 822
Cr-- CASE: M<[N/2+1], P DOES NOT EXIST
    DO 381 II \(=1, M-1\)
```

```
    I=N-II
    Z=0
    DO 38< J={+1,N
    382 Z=2+H(I-J+H,J-M+1)*S(J)
    381 S(I)=S(I)-GS(I)*S(I+1)-2
    DU 38S I I =1,N-2*M+1
    I=N-M+1-1I
    Z=0
    DU 384 J=1*1,1+11*1
    384 Z=Z+H(I-J+M,J-M+1)*S(J)
    333 S(I)=S(I)-GS(I)*S(I+1)-2
    DU 380 II=1,M-1
    I=M-II
    Z=0
    DU 387 J=M,I+M-1
    387 2=2+H(I-J+M,J-M+1)*S(J)
    386 S(I)=S(I)-GS(I)*S(I+1)-Z
    GO TO 66
C-m- CASE: M=N/2 , P DOES NOT EXIST.
    82? DO 388 II=1,NmM-1
        I=N-II
        2=0
        DO 389 J=I*1,N
    389 z=Z+H(I-J*M,J-M+1)*S(J)
    388 S(I) =S(I)-GS(I)*S(I+1)mZ
        I=M
        z=0
        DO 391 J=I+1,N-1
    391 2=2+H(I-J+M,J-M+1)*S(J)
        S(I)=S(I)-GS(I)*S(I*1)-2
        DO 392 II=1,M-1
        I=11-1\
        z=0
        DO }393\textrm{Jm}\mp@subsup{M}{1}{\prime}I+M-
    393 Z=Z+H(I-J+M,J-M+1)*S(J)
    392S(I)=S(I)-GS(I)*S(I+1)-2
    6 6 ~ C O N T I N U E ~
        RETURN
        END
```

SUBROUTINE NBSD(N,M,P,A,B,C,H,S)
this subruutine solves the set wxes of n linear equations ( N $>$ \&) (JHERE THE COEFFICIENT MATRIX IS SYMMETRIC,POSITIVE DEFINITE AND SEVEN DIAGONAL OF BANDWIDTH MOP ( $\angle<M<P<N$ ). A, B, C, $H, ~ S$ ARE VECTORS CONTAINING RESPECTIVELY The DIAGONAL , CO DIAGONAL,M-TH DIAGONAL"P畀TH DIAGONAL AND CONSTANT ELEMENTS . THE MATRIX IS FACTORISED INTO --DTTTD=WIIERE D IS A DIagunal Matrix and tis a realoupper triangular Matrix with unit diagonal elementsononezero elements in SUPER DIAGONAL AND RETAINING R1=M-1 AND R2=P-1 OUTERMOST off-diagonal entries. tt denutes the transpose of t. the infut vectars $A, B$ are overwritten during the factorization stage so that vector a contains the diagonal elements of matrix d and vector b contains the co diagonal elements of matrix t two result vectors uid of bengit n and two arrays OF DIMENSION [Pmi,N-M+I] ARE USED AS WORKSPACE. THE SOLUTION IS EFFECTED BY A FORWARDMBACKWARD SUBSTITUTION PROCESS. THE NORIIALIZED R.H.S. VECTOR IS OVERWRITTEN SUCCESSIVELY EY THE INTERIEDIATE SOLUTIUN (OBTAINED BY FORWARD SUBSTITUTION) AND the final solution (ObTAINED by back SUbStitution). the VECTORS A,B,CiH,S ARE GENERAIED by the CALLING SUBROUTINE GEVECT. N1=P-1, $N 2=N-M+1, N S=N-P+1$.
DIMENS!ON A(N),B(N);C(N2)OH(N3),S(N),D(N)OU(N)
DIMENSION F(N1, H2), G(H9,N2)
INTEGER P
CALL GEVECT(N,P,M,A,B,C,H,S)
test fur the structure of the coefficient matrix. If (M.NE:N) GO TO 871
MM $=1 /$
MPDMM M+1
GO TO 173
871 WEM
MNEH
$1 \mathrm{C}=0$
DO 151 I:1 $1 N m+1$
IF (C(1):EQ,O) IC ElC +1
151 CONTINUE
IND=0
DO 171 1:1, N- $P+1$
IF (H(I).EQ,O) INDZIND+1
171 CUNTINUE
ISaIC+IND

IF (IS.EQ.ISUM) GO TO 999
IF (IC.NE.N-M+1) GO TO 172
C CASE : NORMBAND WITH BANDWIDTH P.
$M P=N-P+1$
$11=\mathrm{P}$
IKF EIRG
HIIFP
DO 155 Ia1, $N-P+1$
$153 \mathrm{C}(1)=\mathrm{H}(1)$
GU TO 173
172 IF (IND.NE,N-P+1) GO TO 973
C
CASE : NORIGBAND WITH BANDWIDTH M.
MP=N-141

```
        H112II
    C STAGE I: FACTORISATION.
    \(173 \mathrm{~A}(1)=\operatorname{SQRT}(A(1))\)
        DO 1 l-2";MH-1
        \(2=B(1-1) / A(1-1)\)
        A(1) =SQRT(A(1) \(-2 * 2)\)
    1 B(I-1) m/A(I)
        IF (M.EQ.N) GO TO 177
        IF (IC.NE.N~W+1.OR.IND.NE.N-P+1) MP=PMH
    177 DO 6 J円1:MP
        \(U(1)=C(J) / A(J)\)
        \(V=B(M+J-2) / A(M+J-2)\)
        If (J.GTr(l|-2)) GO TO 7
        DU 2 【の2" 110 J
        \(2 \quad U(1)=-B(1+J-2) * U(I-1)\)
        7 [f (J.E(as1) GO TO 8
            IF (J.GT. (H-1)) IR=2
            if (J.he. (llmi)) i \(\quad=11-J+1\)
            DO 3 Inipill-1
            \(z=0\)
```



```
    \(4 \quad 2=2+U(K) * F(K-I+11,1+J-11)\)
    \(3 \quad U(1)=-B(1+J-2) * U(1-1)-2\)
    \(8 \quad 2=0\)
    DO 5 KaliMme2
    \(5 \quad 2=2+U(K)+U(K)\)
        \(A(M+J-1)=\operatorname{SQRT}(A(M+J-1)-2-(U(M-1)+V) * * 2)\)
        \(B(M+J-2)=V / A(M+J-1)\)
        DO 6 In1:IRF
    \(6 \quad F(I, J) m u(I) / A(H+J M 1)\)
        IF (M.EQ.N) GO TO 311
        IF (IC.EQ.N-W+1.OR.IND.EQ.N-P+1) GO TO 656
        DO 27 !=1,N
    27 D(1)aA(1)
        \(U(P-1)\) हD \((P n i)\)
        \(M 1=P-M+1\)
        DO \(12 \mathrm{~J}=1, N-P+1\)
        \(A(P+J-1)=A(P+J-2)\)
    \(562 G(1, J) \oplus H(J) /(A(J) * A(P+J-1))\)
        \(V=B(P+J-2) /(A(P+J-2) \star A(P+J-1))\)
        IF (J.GT: (H-2)) GO TO 31
        DO 33 \&a2, 1 mJ
        \(G(1, J)=-B(1+J-2) * G(I-1, J)\)
        IF (J.EQ, \(1 . A N D . I, E Q .111) G(1, J)=G(I, J)+C(I+J-1) /(A(I+J-1) *\)
        * \(A(P+J-1))\)
    33 cuntinue
        generation of elements of matrices u and r (froh the
C GENERATION OF ELEMENTS OF MATRICES
C PARTITIONED COEFFICIENT MATRIX A).
    31 IF (J.GT. \(\left.\left(M_{m} 1\right)\right)\) iP=2
        IF (J.bET(H-1)) IPAM-J+1
        DO 16 Ia』PiPal
        IF (I.OEIPOJ+1.OR.J.GT.P-2) GO TO 654
        2:0
        IF (I.GT:M) GO TO 23
        DO \(17 \mathrm{~K}=1, \mathrm{Hm}=1\)
        \(172=2+G(K+I-H, J) * F(K, I+J-H)\)
        GO TO 24
    23 DO 26 Ka1,l-1
    \(262=2+G(K, j)\) क \(F(K-1+11, I+J-11)\)
    \(24 G(I, J)=B(I+J-2) * G(I-1, J)-2\)
```

```
    IF (I.EQ.M9 ) G(I,J)=G(I,J)+C(I+J-1)/(A(I+J-1)*A(P+J-1))
    GO TO 16
    654
    DO 21 K=1.Im1
    21 Z=z+G(K-I*P,I+J-P)*G(K, J)
    G(I,J)=пВ(I+J-2)कG(I-1,J)-2
    IF (I,EQ,H1 ) G(I,N)=G(I,J)+C(I+J-1)/(A(I+J-1)*A(P+J-1))
    10 cONTINUE
    I=R-1
    W=V+G(l,J)
    15 2=0
    DO 161 K=1,P-2
    161 2=2+G(K,J)*G(K,J)
    A(P+J=1) mSQRT(D(P+J-1)/(1+Z+W**2))
    IF (ABS(A(P+J-1)-U(P+J-2)),LT.1.OE-12) GO TO 191
    U(P+J-2) #A(P+Jm1)
    GO TO S62
    191 IF (J,EQv1) U(P-1)=D(P-1)
    U(P+J-q)=A(p+J-1)
    12B(P+J-2)=B(P+J-2)/(A(P+J-2)*A(P+J-1))
        A(P-1)=#D(P@1)
    656 IF (IC.NE.N-W+1) GO TO 311
        M=W
    DO 314 I=1,p-1
    DO 314 Ja1,NmM+1
        G(I,J)禺F(I,J)
    314 F(I,J)=0
C STAGE II: NORMALIZATION.
    311 DO 37 \=1,N
    37 S(I)AS(IS/A(!) SOLUTION (FURWARDMBAGKWARD SUBSTITUTION).
    DO 44 I=2,H-1
    44 S(I)=S(I)= B(I-1) *S(Im)
    IF (M,EQGN) GO TO 396
    DO }78\mathrm{ l=M, P-1
    2=0
    DO 79 K=Im II+1,I-1
    79 2=2+F(K-I+H,I-M+1)*S(K)
    78 S(1)ES(1)-B(I-1)*S(I-1)m2
    396 DO 87 & = P,N
        2=0
        DO }88\quadK=1-11+1,I-
    88 2=2+F(K-I+M,I-M+1)*S(K)
        21=0
        DO 89 Kalm P+1,Im1
    89 Z1=21*G(K-l* P,Im P+1)*S(K)
    87 S(I)=S(I)-B(I-1)*S(I-1)-2-21
        IF (M,HTw(N/2+1)) GO TO 64
        IF (M,EQNN) GO TO 65
        IF (P,GTL(N/2+1)) GO rO 18s
CHON CASE: M\[N/2+1}, P DOES NOT EXIST
    0047 1IEI,NmM
    I=N-1I
    Z=0
    DO 48 J=1+1,N
    48 2= 2+F(1~J+H(J-H+1)*S(N)
    47 S(1)=S(1)% B(I)*S(1+1)m2
    65 DO 49 IIm1,2*H-NmI
        I=Mmld
        2=0
```

```
            DO 51 J=M,N
    51 Z=Z+F(I-J+M,J-14+1)*S(J)
    49 S(I)=S(I)-D(I)*S(I+1)-2
        IF (M.EQ.N) GO TO 66
        DU 52 |I=1, NmM
        I=NmM+1-II
        Z=0
            DO 53 JmM,I+M-1
    53 2=2+F(| - J +11,J-M+1)*S(J)
    52S(I)=S(I) ~B(I)*S(I+1)-2
        GO TO 66
CP-m CASE: |1>{N/2+1]: P>{N/2+1]
    183 IF (P,EQ%N) GO TO 252
        DO 231 11#1,NmP
        I=N-II
        Z=0
        21=0
        DO 232 J m l +1,N
        Z=Z+F(I-j+M,JmM+1)*S(J)
    232 21=21+G(I-J+P,J~P+1)*S(J)
    231 S(I) =S(I) -B(I)*S(1+1)-2-21
    252 IF (M,EQ,N) GO TO 253
        DU 233 II#9, PmM
        I=Pm!l
        Z=0
        OO 234 Jml*1,N
    234 Z=Z+F(|\inftyJ*||,JmM+1)*S(J)
    21=0
    DO 235 JmM,N
    235 21=21+G(I~J*P,J-P*1)*S(J)
    233 S(1)=S(1)-B(1)*S(1+q)-2-29
    253 DO 236 IIENmM+1,M=1
        I=NmI!
    Z=0
    21=0
    DU 237 J=M,N
    Z=2+F(l-J+M,JmM+1)=S(J)
237 21=21+G(I-J+P,J-P+1)*S(J)
236 S(I) =S(I)-B(I)*S(I+1)-2-21
    IF (M, EQyN) GO TO 66
    DO 238 II=H,P-1
    I=N-18
    Z=0
    00239 JaM,l+M-1
239 2&2+F(J-J+M,J-M+1)*S(J)
    21=0
    DO 249 JuH,N
249 21=21+G(1-J+P,N-P+1)*S(J)
238S(1)=S(1) -B(1)*S(1+1)-2mZ1
    IF (P,EQ,N) GO TO 66
    DO 242 II=P,N-1
    I#Nm1\
    2=0
    21=0
    DO243 JmH1I+M-1
    Z=2+F(I-J+11,J-M+1)*S(J)
243 Z1mZ1+G(ImJ+P,J-P+1)*S(J)
242S(I)=S(I)mB(I)*S(I+1)-2-Z1
    GO T0 66
64 IF (P,GT,N,OR,P.LE.M) GO TU 821
```

```
    IF ( P .LT,(N/2+1)) GO TO 444
    IF (((N-N/2*2),EO.O).AND.(M.EQ,N/2)) GO TO 67
Cm=- CASE: M<[N/2+1], P>[N/2+1].
    IF (P.EQ.N) GO TO 254
    DO 94 1I=1,N- P
    I=N-11
    2=0
    21=0
    DU 95 J=1+1,N
    21-21+G(I-j+ P,Jm P+1)*S(J)
    95 2=2+F({-J+M,J-M+1)*S(J)
    94 S(1)=S(1)-E(1)*S(I+1)-2-21
    25400 96 \IaNm p+9":M-1
        I=N-1!
        2=0
        DO 97 J=I+1,N
    97 z=2+F(d-J+H,J-M+1)*S(J)
        21=0
        DU 98 J= P,N
    98 21=z1+G(Im,* P,Jm P+1)*S(J)
    96 S(I)=S(1)mB(I)*S(I+1)m-2-21
            DO 101 1IFH,N-M
            I=N-!!
            2=0
            DO 102 Jml*1.1+|m1
    102 z=2+F(f-j+M,j-M+9)*S(J)
            21=0
            DO 103 jaP ,N
    103 21=21+G(ImJ+ P,Jm P+1)*S(J)
    101 S(I)=S(1)=B(!)*S(1+1)-2m21
            DO 104 II=N-M+1, P-1
            I=N-II
            z=0
            DO 105 JmM,I+M-1
    105 2=2+F(f-J+11,J-M+1)*S(J)
            21=0
            DO 100 J= P,N
    106 21=21+G(ImJ* P,Jm P+1)*S(J)
    104 S(I)&S(1)mB(I)*S(l+1)-2m29
            IF (P,E(G;N) GO TO 66
            DO 121 IIE P ,N-1
            | FHm|
            z=0
            DO 122 JmM,I+M-1
    122 2=2+F(f-J+M,J-M+1)*S(J)
            21=0
            DO 123 J#PII+P-1
    123 21=21+G(1-J+P,J-P+1)*S(J)
    121 S(1)=S(1)-B(1)*S(1+1)-2-29
            GO TO }0
CT-- CASE: MEN/2 ,P>[N/2+1].
    67 If (P,EQ,N) GO TO 256
            DO ó IIa1,N-P
            l=N-II
            2=0
            21=0
            DO 62 Jal+1,N
            z=2+F(i-j+M,J=N+1)*S(J)
62 21=21+G(1-j+p,J-p+1)*S(J)
61 S(1)=S(1)=B(1)*S(1+1)-2-21
```

```
    256 DO 14, II=1,P-N-1
        I=P-II
        z=0
        DO 142 JEl+1,N
    142 2=2+F(|-J+M,J-M+1)*S(J)
        z1=0
        DO 145 JaPMN
    145 21=21+G(1-J+P,J-P+1)*S(J)
    149 S(I)=S(I)-B(!)*S(!+1)-2-21
        l=M
        z=0
        00 146 J=M+1,N-1
    146 z=2+F(f-J+M,J-M+1)*S(J)
        z1=0
        DO 147 J=P,N
    147 21=21+G(l-d+P,J-P+1)*S(J)
        S(1)=S(15mB(1)*S(1+1)-2-29
        DO 148 IIEq,M+PmN=1
        I=M-I!
        z=0
    DO 144 JaH/It+M-1
    149 Z=2+F(f-j+M,J-M+1)*S(J)
    21=0
    DO 158 JaP,N
    158 21=21*G(1"J+P,J-p*1)*S(J)
    148 S(1)=S(1)-B(1)*S(1+1)-2-21
        IF (P,EQyN) GO TO 60
        DU 152 IIEP,N-1
        I=NmII
        2=0
        DO 156 j=11, I+M-1
    156 z=2+F(I-J*M,J-M+1)*S(J)
    21=0
    DO 154 JaP,1+P-1
    154 21=21+G(I-J*P,JmP*1)*S(J)
    152 S(1)=S(I)-B(I)*S(l+1)-2-21
    GO TO OO
Cr-m CASE: H<[N/2+1], P<{N/2+1]
    444 IF (((N-N/2*2),E(Q,O).AND.S P.EO.N/2)) GO TO 107
        DO 54 |I#1,||=1
        I=N-1!
        z=0
        21=0
        DO 55 J=1+1,N
        z1=29+G(1-j+ POJm P+1)*S(J)
    55 2=2+F(I-j+M,J-N+1)*S(J)
    54 S(1)=S(I')-8(1)*S(1+1)-2-21
    DO 56 &IaM, P-1
    I=N-Il
    z=0
    DO 57 J=1+1,I+M-1
    57 2=2+F(I-J+M,J-M+1)*S(J)
    21=0
    DO 91 J=1+1,N
    91 21=21*(j(i-j+ P,Jm p+1)*S(J)
    50 S(I)ES(15-B(I)*S(I+1)-2-21
        DO 124 IIE PON- P
        I=N-11
        2=0
    DO 125 Jma*1.1+11m1
```

```
    1252=22+F(1-J+M,J-M+1)*S(J)
        21=0
    DO120 J=1*1.l+ R-1
    126 21=21+G(I-J+ P,Jm P+1)*S(J)
    124 S(1)=S(1)-B(I)*S(1+1)-2-21
        DO 127 IIFNm P+1,N-M
        I=N-II
        2=0
        DO 128 JaI*1,I+11-1
    128 Z=Z+F(|-J+H,J-M+1)*S(J)
        21=0
        DO 12% J= P.l+ P-1
    129 21=21+G(I-J+ P,Jr p+1)*S(J)
    127 S(I)=S(I)-B(I)*S(I+1)-2-29
        DO 58 |I=1,||=1
        I=M-I!
        2=0
        DO 59 \aM, 8+M-1
    59 2=2+F(I-J+M,J-M+1)*S(J)
        21=0
        DU 93 J= P.l* Pr1
    9321=21*G(1mJ* P,Jп p+1)*S(J)
    58 S(I)=S(I)-B(I)*S(I+1)=2-21
        GO TO 66
CF-- CASE: M<{N/2+1],P=N/2
    107 DO 108 IIF1,Nm PM1
        I=N-1!
        2=0
        z1=0
        DO 109 JmI+1,N
        z1=z1*G(I-d* P,Jmp+1)*S(J)
    109 2=2+F(f-J+M,J-M+1)*S(J)
    108 S(I)ES(I)-B(I)*S(I+1)-2-21
        I~P
    2=0
    DO 111 J=1*1.N
    111 2=22+F(ImJ+M,JmM+1)*S(J)
    21=0
    DO 112 Jm p+1,Nm1
    112 21#21+G(Imd+ P,Jm P+1)
        S( 1)mS(1)mB( 1)*S( 1+1)-2-21
        DO 113 IImN- P+1,N-M-1
        I=N-1d
        2F0
    DO 114 Jml*1,N
    114 Z=2+F(ImJ+M,Jm H+1)*S(J)
        21=0
        DO 115 Jm P,It P-1
115 21= z1+G(ImJ* P,Jr P+1)*S(J)
113S(1)=S(1)-B(1)*S(1+1)-2-21
    I=M
    2=0
    DO 116 JmM+1,N-1
    116 2=2+F(fmJ*M,J~M+1)*S(J)
    21=0
    DO 117 Jm Pilt P~1
117 21=21*G(ImJ* P,Jr P+1)*S(J)
    S(1)=S(15-B(1)*S(1+1)-2-29
    DO 118 II FNmM+1,Nm1
    I=N-I!
```

```
        z=0
        DO 119 JaM,I+M=1
    119 z=2+F(f-J+H,J-M+1)*S(J)
        21m0
        DO 131 J= PoI+ P=1
    139 21=21+G(l-J+ P,J= p+1)*S(J)
    118 S(I)=S(I)-B(I)*S(I+1)m2-21
    GO TO 66
    821 IF ((NMH/2由2).EQ.O.AND.H.EQ.N/2) GO TO 822
CF-- CASE: M<[N/2+1] , P DOES NOT EXIST
    DO 381 II=1,M-1
    I~N-II
    Z=0
    DO 382 Jal$1,N
    382 2=2+F(I-J+M,J-M+1)*S(J)
    381 S(1)=S(1)-B(1)*S(l+1)-2
    DO 383 1!=1,N-2*M+1
    I=N-M+1-11
    2-0
    DU 384 Jml*1.l+11-1
    384 2F2+F(I-J+M,J-M+1)*S(J)
    383 S(1)=S(1)-B(I)*S(1+1)-Z
    DO 380 11=1,M-1
    I= M-I!
    2=0
    DO 387 JaM, I+H-1
    387 Z=2+F(f-')+M,J-M+q)*S(J)
    386 S(1)=S(1)-B(1)*S(1+1)-2
        GO TO 66
C+~= CASE: HEN/2, P DOES NUT EXIST.
    822 DO 388 11#1,N-M-1
        I=Nm!!
        2=0
        DO 389 J=I*9,N
    389 2=2+F(I-J+M,J-M+1)*S(J)
    388 S(1)mS(!g-B(1)*S(!+1)-2
        IFM
        z=0
        D0 391 JmI$1,Nm9
    391 Z=Z+F(f-J+M,J-M+1)*S(J)
    S(I)=S(I)mB(1)*S(I+1)mZ
    DO 392 1IE9,11-1
    I=M-1!
    Z=0
    DO 393 JaH,l+M-1
    393 2=2+F(|-J+1H,J-M+9)*S(J)
    392 S(1)=S(1)-B(I)*S(I+1)-2
    66 DO 76 &=1,N
    76 S(I)=S(15/A(I)
        RETURN
    999 WKITE (2':800)
    800 FORHAT\9A 15X,'***GIVEN MATRIX IS A TRIDIAGONAL MATRIX***'1)
        RETURN
        END
```

SUBROUTINE ALUBOT（N，M，IR，B，A，C，U，V）
C THIS SUBROUTINE IS AN APPRUXIMATE FACTORIZATION OF A SQUARE
MATRIX DF ORDER N, THE COEFFICIENT IIATRIX IS NON-SYMMETRIC.
PUSITIVE UEFINITE AND QUINDIAGONAI. OF BANDWIDTH M ( $3<4<N$ ).
R, C, AIU,V ARE VECTDRS CONTAINING RESPECTIVELY THE DIAGONAL.
UPPER AND LOWER CO DIAGONAL AND METH DIAGONAL ELEMENTS.
THE CUEFFIGIENT MATRIX IS FACTORIZED INTO LS US WHERE LS.
US (WITH UNIT DIAGONAL ELEMENTS) ARE STRICTLY LOWER AHD UPPEK
TRIANGULAR IIATRICES WITH NON-ZERO ELEHENTS IN SUBISUPER
DIAGONAL RESPECTIVELY AND REIAINING IR E!qoll-qJ OUTERMOST
OFFmDIAGONAL ENTRIES. THREE RESULT VECTORS OF LENGTH N AND
TWO ARRAYS OF DIMENSION [IR'ON-M+IJ ARE USED AS WORKSPACE.
H1ョH~Mわ1.
$D I M E N S$ §ON $B(N), A(N), C(N), U(N), V(N)$
COMIION /BLOCK2/G(N), W(N) OD(N)
COMHON /BLOCKG/H(IR,H1),E(IR,N1)
C
FACTOKIZATION OF TRIDIAGONAL MATRIX AII
W(1) mb(1)
$D(9)=A(2)$
G(1) $\mathrm{AC}(1) / W(1)$

$W(I)=B(I)-D(I-1) * G(I-1)$
$D(I) E A(I+1)$
$5 \quad G(I)=C(I) / W(I)$
$W(M-1) \equiv B(M-1)-D(M \sim 2) \star G(M-2)$
C CALCULATIUN OF ELEMENTS OF SUBMATRICES U92.L21.U22.L22.
DO 6 JM1iN Nmill
$E(1, J) F V(J \oplus 11-1)$
$H(1, j) m \cup(J) / W(J)$
$G(M+J-2)=C(11+J-2) / W(M+J-2)$
$D(M+J-2) \equiv A(1+J-1)$
IF (IKmJ巾1.LT.2) ©0 TO 22
DO 7 In2"IRmJ+1
$E(I, J)=-G(\|+J \sim 2) \star E(1 \infty 1, J)$
$7 \quad H(I, J)=\oplus D(!+J=2) \oplus H(I-1, J) / W(I+J-1)$
22 IF (J.EQ.i IOR.IR.EQ. 1 ) GO 1023
IF (J.GT.IR) IP=2
IF (J, $\subset E \| R) \quad I P=\| R-J+2$
I $R 1=1 R+1$

2:0

ZロZ+E(K,J) H H (K=I+IR1,I+J-IR1)
$E(J, J) \operatorname{Bn} G(\|+J m 2) \oplus E(I-1, J)=2$
2:0
DO 12 Kaq, I-1
$2=2+E\left(K_{m} I+I R 1, I+J+I R 1\right) * H(K, J)$
$H(I, J) A(\mp D(I+J-2) 由 H(I-1, J)-2) / W(I+J=1)$
8 CONTINUE
23 IE1R
Z=0
DO 13 Km : 1
ZЕZ+E(K,J) \#H(K,J)
13 CONTINUE
$W(M+J=1)=B(H+J-1) \sim D(I+J-1) * H(I, J)-G(I+J-1)+E(I, J)-D(I+J-1)$ *
$-G(I+J-1)=z$

6 CUNTINUE RETURN END

```
    SUBROUTINE FBSUBS(N,M,IR,S)
    THIS SUGBRUTINE SULVES THE SET (LS US)YES Of N-LINEAR
    EQUATIUNS, THE SOLUTION IS EFFECTED BY A FORWARDOBACKWARD
    SUBSTITUTION PROCESS WHERE THE INPUT VECTOR S IS OVERWRITTEN
        SUCCESSIVELY BY THE INTERHEDIATE SOLUTIONCOBTAINED BY FORWARID
        SUBSTITUTION) AND THE FINAL SOLUTION (OBTAINED BY BACK
        SUBSTITUTION). N1mN-M+1.
        DIMENSION S(H)
        COMHON /BLDCKZ/G(N),W(N),B(N)
        COMIHON/BLOCK4/T(IR,NQ),E(IR,NQ)
        S(1)=S(1)/W(1)
        DO 7 dm2"NM1
    7 S(l)=($(I)= B(lm1)*S(l-1) )/W(l)
        DO 8 Im|INN
        z=0
```



```
    9 2= Z+E(K-I*|,I-H+1)*S(K)
    8 S(I)=(S(I)mB(Im1)+S(Im1)mz )/W(I)
        IF (H.LT.N/2+1) GO TO 34
        IF (M,EQ,N) GO TO 35
CM=- CASE: |l> [N/2+1].
        DO 10 1!#1,NmM
        I=N-1!
        2=0
        DO 11 J=I*q,N
        IF (I-J+11.GT.IR) OO TO 11
        2= 2+T(Imd+||, (-11+1)*5(J)
        11 cONTINUE
        10 S(1)= S(1)mG(I)*S(1+1)mZ
        35 DO 12 1121,2*|I-N-1
        I=Nm|l
        2=0
        DO 13 JaM,N
        IF (ImJ+M.GT.IR) GO TO 13
        Z=2+T(I-J+M,JmM+1)*S(J)
        13 CONTINUE
    12 S(I)= S(I)mG(I)*s(I+1)-2
        IF (M,EQ.N) GO TO 36
        DO 14 &Im1,N-M
        I=N-M+9-1!
        2=0
            DO 1b JNH!I+M=1
            IF (I~J+M.GT.lR)GO TO 15
            2#2+T(d-J+|,J-NH+1)*S(J)
        15 CONTINUE
    14 S(1)= S(1)-G(1)*S(1+1)-2
    GO TO 36
    34 IF (((NmN/2*2),EQ,0),AND.(M,EQ,N/2)) GO TO 37
Crom CASE : II< [N/2+1〕.
        DO 16 IIMI,Mm1
    I=Nm|!
    2=0
    DO 17 J=I+9,N
    IF(I-J+M.GT.IR) GOTO 17
```

```
        Z=Z*T(I-J*M,J-|+1)*S(J)
    17 CONTINUE
    16S(I)=S(I)~G(I)*S(I+1)*Z
        DO 18 II=9,N-2*H*1
        I=N-M+9-1\
        2口O
        DO 19 J=I*9,I*H=1
        IF (I~J+H.GT.IR) GO TO 19
        Zロ Z+T(I~J+||,d-|+I)*S(J)
    19 CONTINUE
    18 S(I)=S(I)-G(I)*S(I+1)-2
        DO 20 \I#1,|m1
        | AM=I|
        2口0
        DO 21 JaM1I+Mm1
        IF(ImJ+M,GT.IR) (iO TO 21
        Z=Z+T(I+J*||J-M*I)*S(J)
    21 CONTINUE
    20 S(I)=S(I)mG(I)*S(I+1)=Z
        GO TO 36
Cmmm CASE : N IS EVEN AND MZN/2.
    37 DO 22 &Im1,N-H-1
        I#Nm!1
        2#0
        DO 23 Jml*9.N
    23 2= 2+T(I~J+||Jm|+1)*S(J)
    22S(I)ES(I)mG(I)*S(I+1)-2
        I#M
        Z口0
        DO 24 JaM+1,N-1
        2= Z+T(I-J*||, J-||q)*S(J)
        S(I)= S(I)mG(I)* S(I+1)-2
        DO 25 |Ia1,Hm1
        I=M=1!
        2=0
        DO 26 JचH,l+M=1
    2= 2+T(I-dw|,J-|l+1)*S(J)
    26 Sn S(I)mS(I)mG(I)*S(I+1)=Z
    36 CONTINUE
        RETURN
        END
```

SUBROUTINE ALUBOTYD（N，PIM，A＇B，C，U，V＇GR，S）
YHIS SUBRUUTINE IS AN APPROXIMATE FACTORIZATION OF A SQUARE MATRIX OF ORDER N．THE COEFFICIENT MATRIX IS NONGSYMMETRIC． POSITIVE DEFINITE SEVEN DIAGONAL OF BANOWIDTHS M AND P （L＜UI＜P＜NS．B，CIA，W，VIRIS ARE VECTORS CONTAINING RESPECTIVELY THE DIAGONAL ，UPPER AND LOWER ICO DIAGONAGIMOTH OIAGONAL AND P円TH DIAGONAL ELEMENTS．THE COEFFICIENT MATRIX IS APPROXIMATELLY FACTORIZED INTU LS US WHERE LS，US IWITH UNII DIAGONAL ELEMENTSY ARE STRICTLY LOWER ANO UPPER TRIANGULAR MATRICES WITH NONEZERO ELEMENTS IN SUB，SUPER DIAGONAL RESPECTIVELY ，RETAINING IRI E\｛1，M－IJ，IR2 E【I，P－IJ OUTERMOST OFFMDIAGONAL ENTRIES．THREE RESULT VECTORS OF LENGTH N AND FOUR ARRAYS OF DIMENSION【IRI，N－M＋IJAND 【IR2，N－P＋I〕ARE USEO AS WOKKSPACE，N1EN－M＋1，N2EN－P＋1． COMIION／BLOCK2／G（N），D（N），W（N）

```
DIMENSION A(N),B(N),C(N),U(N),V(N),R(N),S(N)
```

INTEGER P
COMIION /BLOCKT/IRIIIR2
COMION H(IR1,N1),T(IR2,N2),E(IR1,N1):F(IR2,N2)
IRYS=IR1*1
IR2Sm1R2*1
W(1) = 日 (1)
$D(1)=A(2)$
G(1) $=\mathrm{C}$ (1)/W(1)
DO 5 Im2"Mm2
$W(I) \equiv B(I) m D(I-1) \oplus G(I-1)$
$D(I)=A(I+1)$
$5 \quad G(I)=C(I) / W(I)$
$W(M-1)$ B $(M-1)-D(M-2) * G(M-2)$

$E(1, J) \operatorname{L} V(J 由 H=1)$
$H(1, J)$ 觬 $(J) / W(J)$
$G(M+J-2)=C(11+J-2) / W(M+J-2)$
$D(M+J=2) \equiv A(11+J m 1)$
IF (IR9.EQ,1) GO TO 18
IF (J,GT:IRImI) GOTO4

$E(I, J) H-G(\|+J-2) \neq E(I-1, J)$
$7 \quad H(I, J)$ M-D $(I+J-2) \oplus H(I-1, J) / W(I+J-1)$
4 IF (J.EQ.1.OR,IR1,EQ.1) GOTO 18
IF (J,GT.IR1) \{Pm2
IF (J.WE.dRI) IP由IRI-J+2
DO 8 IMIPI \&R9
Z $=0$
DO 3 KMri:Im9
$Z=Z+E(K, J) \neq H(K-I+I R 1 S, I+J=I R 1 S)$
$E(I, J) E=G(I+J m 2) \oplus E(I \sim 1, J)=2$
$2=0$
DO $12 K=1,1=1$
$122=2+E(K-I+I R 9 S, I+J-I R 1 S) \oplus H(K, J)$
$H(I, J)$ ( $-D(I+J-2) \neq H(I=1, J)-2) / W(I+J=1)$
8 CONTINUE
18 ImIR1
$2=0$
DO 13 Kai 11
IF (K,GTHIR1) GOTO 13

```
            2口z+E(K,j)*H(K,J)
13 CONTINUE
            W(M+J-1)=B(II+J-1)~D(I+J-1)*H(I,J)-G(I+J~1)*E(I,J)-D(I+J-1)*
        *G(I+J-1)=z
    6 CONTINUE
    GENERATION OF U12
    DC 22 J=1,NmP+1
    T(1,j)ar(J)/W(J)
    G(P+J-2)=C(P+J-2)/W(P+J-2)
    F(1,J) ms(p+Jmq)
    D(P+J-2)=A(p+J-1)
    If (IKZ.EQ,1) GO TO 33
    If (J.GT,\R2-1) GOT017
    DO 23 da2,&R2SmJ
    T(I,J)月mD({+J-2)由T(I-q,J)/W(I+J-1)
    F(I,J)EmG(f*J-2)*F(I-1,J)
    IF (J.EQ:1,AND.I.EQ.P-M+1) GO TO 55
    GO TO }2
    55T(I,J)mT(I,J)+U(I+J-1)/W(I+J-1)
    F(I,J)HF(I,J)*V(P&J-1)
    23 CONTINUE
    17 IF (J.EQ.1,OR.IRZ.EQ.1) GO TO 33
    IF (J.GT:IRZ) IP=2
    IF (J,GE:IR2) IP=IR2-J+2
    DO 24 Im|PIIR2
    IF (I.GE%PMJ+1,OR,J.GT,P~2) GO TO 66
    z=0
    21=0
    IF (I.LT.M) GO TO 25
    DO 28 Kaq,Mm1
    IF (K,GT.JR1) GO TO 28
    Z=2+E(K,I+JmM)*T(K+I-M,J)
    21m21+F(K+ImM,J)*H(K,I+Jm|)
    28 CONTINUE
    GO TO 27
    25 DO 26 K=1,Im1
    IF (K-I+IR\S.LT.I.OR.I+J#IRIS.LT:1)GO T0 26
    2=2+E(K-l+|R1S,I+J-1R1S)#T(KOJ)
```



```
    26 CONTINUE
    27T(I,j)m(nD(I+J-2)*T(I-1,J)-2)/W(I+J-1)
    F(I,J):-G(I+J-2)*F(I-1,J)-29
    IF (I,EQ:PMM+1.AND.J.NE.1) GO TO 53
    GO TO 24
    53 T(I,J)nT(I,J)+U(I+J-1)/W(I+J-1)
    F(I,j)=F(I,J)+V(P*J-1)
    GO TO 24
C
    GENERATION OF U12.U22
    65 IF (J.EQ.1) GO TO 33
    66 2=0
    21=0
    DO 29 Km1,1-1
    IF (K-1&|R2S.LT.1) GO TO 29
    2=2+F(K-I+IR2S,i+J=1R2S)*T(K,J)
    z1*29+F(K_d)*T(Kml+1R2S,I+J-1R2S)
    29 CONTINUE
    T(I,J)m(-D (I+J-2)*T(I-1,j)-2)/W(I+J-1)
    F(I,J)M-G(I+Jm2)*F(I-1,J)=21
    IF (I,NE:PR|l+1) GU TO 24
    T(I,j)ET(I,J)+U(I*J-1)/W(I+J-1)
```

```
            F(I,J)=F(I,J)+V(P+J-q)
            24 CONTINUE
    33 I=IR2
            Z=0
            21=0.
            DO 31 K=1.!
            IF (K,GT,IR2) GO TO 31
            Z=Z+F(K,J)\oplusT(K,J)
            31 CONTINUE
            IF (IR2,GE,P-M+1) GU TO 822
            EFFECT OF E(I,N),H(I,J),I E[1,RIJOJ E{PmM+1,N] ON
            C EFFECT OF E(I,N),H(I,J):IEE{1.
            JJ=J+Pm|l
            II=IR1
            DO 32 K=1, & I
            IF (K,GTGIR1) GO TO 32
            21=21+E(K,JJ)*H(K,JJ)
    32 CONTINUE
    822W(P+Jmq)mB(P+Jm1)mD(I+Jmq)*T(I,J)~D(I+J~q)*G(I+J-1)-
    *G(I+J~q)*F(IOJ)=Z-Z1
    22 CONTINUE
            RETURN
            END
            SUBROUTINE FBSUBSSD(N,P,M,S)
C THIS SUBROUTINE SOLVES THE SET (LS USIVIMS OF NOLINEAR
    EQUATIONS, THE SOLUTION IS EFFECTED BY A FORWARD-BACKWARD
    SUBSTITUTION PROCESS WHERE THE INPUT VECTOR S IS OVERWRITTEN
    SUCCESSIVELY BY THE INTERMEDJATE SOLUTIONGORTAINED BY FORWARO
    SUBSTITUTION) AND THE FINAL SOLUTION (OBTAINED BY BACK
    SUBSTITUTION),N1=NmM+1,N2#N-P+1.
    DIMENSION S(N)
    COMHON/BLOCKZ/GS(N),B(N):W(N)
    COHHON H(IR1,N1),T(IR2,N2),E(IR1,N1),F(IR2,N2)
    COHMON /BLOCK7/IRY,IRZ
    INTEGER P
    S(1)=S(1)/W(1)
    DO 44 I=2,Hm1
    44 S(I)=(S(1)NB(Im1)*S(I-1) )/W(I)
    DO 78 ImM, Pm1
    2=0
    DO 79 K=ImM+1,ImM*IRI
    79 2=2+E(K-I*H,I-H+1)*S(K)
    78 S(I)=(S(I)mG(I-1)&S(I-1)mZ)/W(I)
    DO 87 la P,N
    Z=0
    DO 88 K=ImM+1,I-M+IR1
    88 2=2+E(K-1+H,I-M+1)*S(K)
    21=0
```



```
    89 21=21+F(Km!+P,! - P*1)*S(K)
    87S(I)#(S(I)-B(I-1)*S(I-1)-Z-Z1)/W(I)
    IF (M, LT,(N/2+1)) GOTO 64
    IF (11,EQ.N) GO TO 65
    IF (P.GT.(N/2+1)) GO TO 183
CFO- CASE: MD{N/2+1J, P DOES NOT EXIST
    DO47 II=1,NmM
    I|Nm|l
```

```
    2=0
    DU 48 J=1$1,N
    IF (Imj+H.GT.JR1) GO TO 48
    z= z+H(Ind+||, |-|+|)*S(J)
    48 CONTINUE
    47 S(I)r S(I)mGS(I)* S(I+1)=2
    65 DO 49 dI"1,2*M-N-1
    I=M-Id
    200
    DO 51 JEM,N
    IF (InJ+M.GT.IR1) GO TO 51
    Z=Z*H(I~J+M,J-M+1)*S(J)
    51 CONTINUE
    49 S(I)a S(1)mGS(1)*S(1+1)-2
    IF (H.EQ,N) GO TO 66
    DO 52 |Ia1,NmM
    I=NmH*I-I!
    2=0
        DO 53 JmM,I+M-1
        IF (ImJ+M.GT.IRI) GO TO 53
        Z=2*H(I~J+N|,JmN+1)*S(J)
    53 CONTINUE
    52S(I)a S(I) ~GS(I)*S(I+1)m2
        GO TO 66
Crom CASE: H>[N/2+1]:P>[N/2+1]
    183 1F (P,EQ,N) GO TO 252
        DO 231 II=1,NmP
        I=N-I!
        2=0
        21=0
        DO 232 JEl\phi1,N
        IF (Imd+M,GT.IR1) GO TO 932
        Z=2+H(I-j*M,JmM+1)*S(J)
    932IF(ImJ*P.GT.IR2) GO TO 232
        z1=z1+T(ImJ+P&Jmp*1)*S(J)
    232 CONTINUE
    239 S(I)=S(15-GS(I)*S(I+1)-2-21
    252 IF (M.EQrN) GO TO 253
    DO233 1fF1,PmM
        I=Pm|!
        2=0
        DO 234 Jaldq|N
        IF (IFJ+M.GT.IR1) GO TO 234
        Z=Z+H(I-J*M,J-N+9)*S(J)
    234 CONTINUE
    21:0
    DO 235 JuM,N
    IF (I-J+P,GT.IR2) GU TO 235
    z1=z1+T(ImJ*P,Jmp*1)*S(J)
    235 CONTINUE
    233 S(I)mS(I)-GS(I)*S(I+1)-2-21
    253 DO 236 IIaNmM+1,Mm1
    I=N-I!
    z=0
    21=0
    DU 237 JmM&N
    IF (ImJ+M.GT.IR1) GO TO 93%
    2=2+H(|mJ+M,JmM+1)*S(J)
    937 IF (INJ+P,GT.IR2) GO TO 237
    Z1mz1+T(I-J+P,J-P*1)*S(J)
```

```
    237 CONTINUE
    236 S(I)=S(I)-GS(I)*S(I+1)=2m21
        IF (M.EQ.N) GO TO 66
        DO 238 II=M,P=1
        I=N-1I
        2=0
        DO 239 JmM,I+H-1
    IF (ImJ+N,GT.IR1) GO TO 239
    Z=2+H(I-J+M,J-M+1)*S(J)
    239 CONTINUE
        21=0
    DO 241 J=M,N
    IF (IFJ+P,GT,IR2) GO TO 241
    z1=21*P(1-J*P,Jmp*1)*S(J)
    241 CONTINUE
    238 S(1)ES(I)mGS(1)*S(1+1)-2-21
        IF (P,EQ.N) GO TO 66
    DO 24Z IIFP,Nm1
    IMN-1!
    z=0
    21:0
    DO 243 JmM!I+M-9
    IF (ImJ+M,GT.IR1) GO TO 94S
    2=2+H(f~j+M,j~M+1)*S(J)
    943 IF (ImJ+P.GT.IR2) GO TO 24S
    21=21+T(I-d+P,J-p+1)*S(J)
    243 CONTINUE
    242 S(I)=S(1)-GS(I)*S(1+1)-2-21
    GO TO 66
    64 IF (P.GT.N.OR.P.LE,M) GO TO ४21
    IF (P..LT,(N/2+Ig) GOTO 444
    IF (((N~N/Z*2),EQ,O).AND.(M,EQ,N/Z)) GO TO 67
C-m= CASE: M<{N/2+1}, P>[N/2+1].
    If (P,EQ,N)GO TO 254
    DO 94 1I=1,N- P
    I=N-1】
    2=0
    21=0
    DO 95 Jm&+1,N
    IF (ImJ+D.GT.IR2) GO TO 49S
    21=21+T(Imj* Pojm p+1)*S(J)
    4 9 5 ~ I F ~ ( I N J + M . G T . I R 1 ) ~ G O ~ T O ~ 9 5 ~
    Z=2+H(ImJ* Hojmil+1)*S(J)
    95 CONTINUE
    94 S(1)a S(1)mGS(1)*S(1+1)-2-21
    254 D0 96 |ImNm P+1,M-1
        I=Nm!\
        2-0
        DO 97 del+1,N
        IF (ImJ+M,GT.IR1) GO TO 97
        z=z+H(j-j+H,j-M+1)*S(J)
    97 CONTINUE
        z1=0
    DO 98 J= P,N
    IF (I~J+P,GT.IR2) GOTO 98
    z1=21+T(1mj+ P!JN P+1)*S(J)
    98 CONTINUE
    96 S(1)mS(1)mGS(1)*S(1+1)-2-21
    DO 101 1!EM,N-M
    I=N-II
```

```
    2=0
DO \(102 \mathrm{~J}=1\) 中1．l＋11－1
IF（ImJ＋M．GT．IR1）GO TO 102
```



```
102 CONTINUE
29：0
DO \(103 \mathrm{iJ}=\mathrm{P}, \mathrm{N}\)
IF（I～J＋P，GT，IR2）GO TO 105
```



```
103 CONTINUE
\(101 \mathrm{~S}(1)=\mathrm{S}(1) \operatorname{mS}(1) * S(1+1) \mathrm{m}-21\)
DO 104 1I \(=\mathrm{Nm} M+9\) ，\(P=1\)
\(1=\mathrm{Na}\)－ 1
\(2=0\)
DO \(105 \mathrm{Jam} \mathrm{M}_{1}+\mathrm{M-1}\)
IF（Im I M．GT．IR1）GO TO 105
\(2=2+H(I-J+M, j-M+1) * S(J)\)
105 CONTINUE
z1 \(=0\)
DO 106 J．\(P, N\)
1F（ImJ＋P．GT．IR2）GO TO 900
z1＝21＊T（I－J＋PoJm P＋1）＊S（J）
106 CONTINUE
104 S（I）ES（I）nGS（I）＊S（1＋1）m2－21
IF（P，EQ；N）GO TO 66
DO 121 IIF P，N－1
ImN－II
200
DO \(122 \mathrm{JmHII}+\mathrm{Mm} 1\)
IF（ImJ＋M．GT．IR1）GO TO 122
\(2=2+H(1 \sim J+H, J m+1) * S(J)\)
122 CONTINUE
\(21=0\)
DO 123 Jmp pilt \(p=1\)
IF（ImJ＋R．GT．IR2）GO TO 125
21＝21＋T（ImJ＋P．Jッ P＋1）＊S（J）
123 CONTINUE
\(121 \mathrm{~S}(1)=\mathrm{S}(1) \operatorname{mS}(1)+5(1+1)-2-21\)
GO TO 66
CPWO CASE：MEN／2 ，P\\｛N／2＋1〕．
67 IF（P，EQ：N）GO TO 256
DO 61 IImin \(N=P\)
1nNaId
2：0
\(21=0\)
DO 62 Jal \(1, N\)
1F（IMJ＋M．GT．IR1）GOTO 281
\(z=z+H(j-j+M, j=M+1) * S(J)\)
281 IF（Imj＋P，GT．IR2）GO TO 62
21m21中T（I－J＋P，J－P＊1）＊S（J）
62 CONTINUE
```



```
256 DO 141 11』9，PaMm 1
\(I=R-1 d\)
2：0
DO \(142 \mathrm{~J}=1+1, \mathrm{~N}\)
IF（I－J＋M，GT．IRI）GO TO 14く
\(2 \cdot 2+H(J-j+M, J=M+1) * S(J)\)
142 CONTINUE
\(21=0\)
```

    DO 14S JaPMN
    IF (I-J+P,GT.IR2) GO TO 14%
    21=21+T(I-J*P,J-P*1)*S(J)
    145 CONTINUE
    141-S(I)mS(I)mGS(I)*S(I+1)-2m21
    I=M
    2=0
    DO 146 Jm|t*1,Nm1
    IF (I~J+M.GT.IR1) GO YO 940
    Z=2+H(I-J+H,J-M+1)*S(J)
    146 CONTINUE
    21=0
    DO 147 JmP,N
    IF (ImJ*P.GT.IR2) GU TO 141
    ```

```

    147 CONTINUE
    S(I)xS(I)mGS(I)*S(I+1)-2-21
    DO 148 IIロq,M+P-Nm1
    I=M-II
    z=0
    DO 149 J=N!I+H-1
    IF (lmJ+M,GT.lR1) GO TO 149
    Z=Z+H(I-J*H,J-M+1)*S(J)
    149 CONTINUE
    21>0
    DO 158 JmPiN
    IF (ImJ+P.GT.IR2) GO TO 15%
    21=21*T(1mJ*P,Jmp*1)*S(J)
    158 CONTINUE
148 S(I)mS(IymGS(I)*S(I+1)-2m21
IF (P,EQYN) GO TO 66
DO 15% 1JFP,N=1
IaNm!!
2\#0
DO 156 J\#M,I*M-1
IF (ImJ+M,GT.IR1) GO TO 156
Z=Z+H(|mJ+M,JmM+I)*S(J)
156 CONTINUE
21=0
DO 154 JmPII+P-1
IF (Imd+P,GT.IRZ) GO TO 154
Z1\#Z1+T(Imj*p,jmp*1)*S(J)
154 CONTINUE
152 S(1)-5S(I)=GS(I)*S(I+1)-2m21
GO TO 66
Cf:= CASE: M<[N/2+1], P<[N/2+1]
444 IF (((N-N/Z*2),EQ,O),AND.( P.EQ.N/2)) GO TO 107
DO 54 \I=1,M-1
I=Nm!!
2=0
21=0
NO 55 J=1+1,N
IF (I-j+P,GT.IR2) GO TO 348
z1=21+T(1-J* P,JN p+1)*S(J)
348 [F (ImJ+M.GT.IR1) GO TO 55
Z=2+H(I~J+M,J~M+1)*S(J)
55 CONTINUE
54 S(1)= S(I)nGS(I)*S(1+1)-2-29
DO 56 |ImM, P=1
I=N-I!

```
```

    2=0
    DO $57 \mathrm{~J}=1+1,1+11-1$
if（I－J＋M．GT．IR1）GO TO 57
$2 \exists 2+H(1 m j+M, J-11+1)=S(J)$
57 CONTINUE
$21=0$
00 $91 \mathrm{~J}=1+1, \mathrm{~N}$
If（I－j＋p，GT．IR2）GO TO 91
z1：21＋T（I～J＊Pojn $p+1)+S(J)$
91 CONTINUE
$56 \mathrm{~S}(\mathrm{I})=\mathrm{S}(\mathrm{I}) \mathrm{mGS}(\mathrm{I}) * S(1+1)-2-21$
DO 124 I！ P ．N－ P
IaNmII
2＝0
DO $125 \mathrm{Jmd}+1,1+\mathrm{Hm}=1$
IF（ImJ＋ll．GT．IR1）GO TO 12S
$2=2+H(d-J+M, J=M+1) * S(J)$
125 CONTINUE
21：0
DO $126 \mathrm{Jml中1.l}+\mathrm{H}=1$
IF（InJ＋P，GT．IR2）GO TO 126
21＝29＋T（1～J＊PoJm P＋1）＊S（J）
126 CONTINUE
124 S（I） A S（1）mGS（I）＊S（1＋1）－2－21
DO 127 IIFNm $P+1, N-M$
$1=\mathrm{N}=1$ I
2•0
DO 128 J＝1＊1．l＋11m
IF（ImJ＋N．GT．IR1）GO TO 128
$z=2+H(j-J+M, J-M+1) * S(J)$
128 CONTINUE
$21=0$
DO $129 \mathrm{Jm} \quad \mathrm{P}, \mathrm{I}+\mathrm{P}=9$
IF（ImJ＋P，GT，JR2）GOTO 129
21：21＊T（1m」 P，Jm p＋1）＊S（J）
129 CONTINUE
127 S（I）mS（1）日GS（I）＊S（1＋1）－2－21
DO 58 1Im1，Mm1
10MmI！
200
DO 59 JaM，$\downarrow+M=1$
IF（ImJ＋H．GT．IR1）GO TO 59
$2=2+H(1-J+M, J \sim M+1)$＊S（J）
59 CONTINUE
$21=0$
DO $93 \mathrm{~J}=\mathrm{P}_{\mathrm{I}} \mathrm{I}+\mathrm{P}=1$
IF（Im $1+P$ GT．IRZ）GO TO 93
z1＝21＋T（I－J＋Pojr p＋i）＊S（J）
93 CONTINUE
$58 \mathrm{~S}(1)=\mathrm{S}(\mathrm{f})=\mathrm{GS}(\mathrm{I}) \mathrm{m}(1+1)-2-21$
GO TO 66
Crime CASE： $11<[N / 2+1\}, P=N / 2$
107 DO 108 1IF1，N PE1
$1 \mathrm{FN}-1$ I
2＝0
$21=0$
DO 109 Jal 1，N
IF（I－J＋P．GT．JR2）GO TO 404

```

```

409 IF（Im $1+11, G T . I R 1)$ GU TO 104

```
```

    Z=Z+H(I-'J*H,JmM+1)*S(J)
    109 CONTINUE
    108 S(I)= S(I)mGS(I)*S(1+1)m2-21
    IEP
    2=0
    DO 111 Jml$1,N
    If (ImJ+M.GT.lR1) GO TO 111
    Z=2+H(I-J+H,JmH+1J*S(J)
    111 CONTINUE
    21=0
    DO 112 J= P+1,N+1
    IF (ImJ+P,GT.IR2) GO TO 112
    21=21+T(Imj+ P,jn p+1)
    112 CONTINUE
S(I)=S(IVmGS(I)*S(I+1)-2mZ1
DO 113 1/mN- P+I,N-Hm1
[=N-1】
2=0
DO 114 Jal\&1,N
IF(Imd+M.GT.IR1) GO TO 114
2=2+H(I~J+M,j~M+1)*S(J)
114 CONTINUE
21=0
DO 115 Jm Pil+ Rm1
IF (ImJ+P.GT.IR2) GO TO 115
21m29+T(ImJ* PaJ. P+1)*S(J)
115 CONTINUE
113 S(I)mS(I)mGS(I)*S(I+1)m2m29
I=M
2:0
DO 116 J=M*1,Nm1
IF (ImJ+M,GT.IR1) GO TO 116
ZaZ+H(Imj+H,j~M+g)*S(J)
116 CONTINUE
21m0
D0 117 Jm Polt Rm1
IF (Im\&+p,GT.IR2) GO TO 11/
z1mz1*T(1mJ* P,j% p+1)*S(J)
117 CONTINUE
S(1)= S(l)mgS(l)*s(I+1)mz-z1
DO 118 I\mNmM+1,Nm1
IENmId
2-0
DO 119 JmM,I+M=1
IF (Imd+M.GT.IR1) GO TO 119
Zп2+H(j~j*M,jmM+1)*S(J)
119 CONTINUE
21=0
DO 131 J= P1f+ P-1
IF (ImJ+P,GY,IR2) GO TO 131
z1=21+T(ImJ* P,J% P+1)*S(J)
131 CONTINUE
118 S(I)=S(1)nGS(I)*8(I+1)-2-21
GO TO 66
821 IF ((N@N/2*2).EQ.O.AND.H.EQ,N/2) GO TO 822
Criam CASE: M<[N/2+1] , P DOES NOT EXIST
DO 381 1!51,M=1
I=N-1!
2:0
DO 382 J=1*1,N

```
```

    IF (I-J+M.GT.IRI) GO TO 382
    Z=Z+H(I-J+M,J~M+1)*S(J)
    382 CONTINUE
    389 S(I) =S(I)-GS(I)*S(I+1)-Z
        DO 383 II=1,Nm2*M+1
        I=N-M+1-1!
        2=0
    DO 384 Jmd由1,1+11m1
    IF(Imd+M,GT.IR1) GO TO 384
    Z#Z+H(I~J+M,J~M+1)*S(J)
    384 CONTINUE
    383S(I) =S(I)-GS(I)*S(I+1)-2
    DO 386 \1п1,Mन1
    I=MmI!
    2:0
    DO 387 JmM,I*M=1
    IF (ImJ+M,GT.IR1) GO TO 381
    Z=Z+H(|mJ+M,J~M+1) & S(J)
    387 CONTINUE
    386 S(I)mS(I)mGS(1)*S(1+1)m2
    GO TO 66
    Crma CASE: MEN/2, P DOES NUT EXIST.
822 DO 388 IIF1,N-M=1
I=N-I】
2:0
DO 389 JE|由1,N
IF (ImJ+M,GT.IR1) GO TO 384
ZaZ+H(I-J+M,JmM+1)*S(J)
389 CONTINUE
388 S(I)mS(I)~GS(I)*S\!+1)mZ
l=M
2:0
00 391 Jmd+1,N-1
IF (I*N+M,GT.IR1) GO TO 391
Z=2+H(| - J +M, JmM+1)*S(J)
391 CONTINUE
S(1)=S(1)mGS(I)*S(1+1)=2
00392 IIE1,M-1
IaMm|!
2\#0
DO 393 JmM,I+11-1
IF (INN+M.GT.IRI) GO TO 393
ZaZ+H(|mJ*H,JmM+9)*S(J)
393 CONTINUE
392S(I) \#S(I) -GS(1)*S(I*1)~2
6 6 ~ C O N T I N U E ~
RETURN
END

```
C
THE NORMALISED IHPLICIT CONJUGATE GRADIENT METHOD IN TWO SPACE DIMENSIONS .
AN ACCELERATED HETHOD FOR SYMMETRIC SYSTEMS,DERIVED FROM LINEAR LAPLACE IN UNIT SQUARE, W*XIS OF N LINEAR EQUATIUNS Where the coefficient matrix w is symmetricipositive definite OF SEMIBANDWIDTH M (3<M<N). THE APPROXIHATE FACTORIZATIUN ALGORITHM NOBAR (I,E.,SUBROUTINES IAPNOFA. GEDRHS , FBSUBS) RETAINING R OUTERMOST OFFDDIAGONAL ENTRIESIIS USED AND THE ITERATIVE PROCESS IS ACCELERATED (BY THE CONJUGATE GRADIENT METHOD ) THE FOLLOWING CALLING SUBROUTINES ARE USED: GEVECT GENERATES THE DIAGONAL VECTORS A,B,C OF THE COEFFICIENT MATRIX AND THE R.H.S. VECTOR S .
RESID ICALCULATES THE PRUDUCT WWX IF IXEI品THERWISE CALCULATES THE RESIDUAL RI=SOW*X.
ERMES : CALCULATES THE ERRUR MEASURES
the n.a.g. sUbroutine gosaaf generates the n pseudomrandon
NUHBERS BETWEEN [0,1]. N1aN-M+1.
\(Y\) IS A VEGTOR OF DIMENSION N CONTAINING THE COHPUTED SOLUTIDN. VS , SI ARE WORKING VECTORS EACH OF DIMENSION N. DIMENSION \(A(N), B(N), C(N), S(N), Y(N), V S(N), R I(N), S I(N)\) COMHON/BLOCKI/N:M,MS,ANUM COMMON /BLOCKG/X(N)
INTEGER Q READ (1.10) N,M,RIMS IIPAR
90 FORIIAT (515)
IH aMS:1
EXACT SOLUTION
IF IPAR IITHE SOLUTION IS A VECTOR OF NGRANDOM NUMBERS IN [0.1]. OTHERWISE THE SOLUTION IS X(I)MOII.NE.M+1,X(M+1)=1. IF (IPARrEQ.1) GO TO 12
DO 448 Im1iN
\(448 \times(1)=0\).
\(x(M+1)=1 *\)
GO TO 666
\(120013 \quad 1=1, N\) 22mG05AAF(YY)
\(13 x(1)=22\) WRITE (2",310)
310 FORMAT (9H 5 SXIEXACT SOLUTIONIII)
DO 14 hatidH

14 WRITE (2"170) (X(今J-1)*1H+1),I=1;1H)
666 WKITE (2!159) N,M,R,MS OIPAR
159 FORHATSTH1,5X, 'ORDER OF MATRIX NEI./5//6X,ISEMIBANDHIUTH MEI
*':IS//OX, 10UTERMOST OFF DIAGONAL ENTRIES RMI.15//6X.
*IMESH SIZE IISE', IS//6X,IIPAREI,15////I
EPS=9.0Em6
550 WRITE (2"; 270\()\) EPS
270 FURIIAT (1H , 5X,'ACCURACY EPS \(=1, E 20,11 / /)\)
\(k=0\)
\(Q=0\)
KII IS AN UPPER LIMIT ON THE NUMBER OF ITERATIONS -
\(\mathrm{KM}=60\)
IX \(=0\)
GVㅁo. O
DO 17 i= \(1, N\)
\(17 \quad Y(I)=G V\)
Y－ッHOLDS－m THE INITIAL APPROXIMATE SOLUTIUN YO－
CALL GEVECT（N，M，MS，IX，A，B，C，S）
CALL RESID（YOIX，S）
C S－HOLDS－－RO
DO 24 IE9，N
24 RI（I）ms（I）
C RIm－HOLDSORO
CALL APNOFA（N，M，R，A，B，C）
CALL GEDRHS（N，A，S）
CALL FBSUBS（NOM，R，B，S）
C \(\quad\) Sm－HOLDSm－RNO DO 28 1＝1，N
28 SI（I）mS（1）
C \(\quad\) SI－HOLDS \(-m R * 0\)
C CALCULATION OF AI，Y，RI．
IX＝1

C + －
C CALCULATION OF PRODUCT W＊SI．
25 CALL RESBD（SI，IX，VS）
C SIm－HOLDSmSO，VSm－HOLDSm－W＊SO．
Q－0
C CALCULATION OF AI
ANUMEO
DENOMEO
DO 41 lation
ANUM EANUH＊RI（I）＊S（I）
C RImmHOLDSmer \(O\) ，SI－WHOLDS＝－R＊O
41 DENOM MDENOH＋SI（I）＊VS（I）
C VSm－HOLDS＂FW＊SO
IF（DENOM．WE．O）GO TO 55
AI aANUII／DENOM
WRITE（2＂゙210）AI
210 FORHAT（1H 5 SXISCALAR AIE 1，E20，111）
C CALCULATION OF Y＂RI．
DO 19 8：1，N
Y（I）\(\quad\) Y（I）中AI＊SI（I）
19 RI（I）MRI（I）नAI＊VS（I）

C test of the convergence with the recursive residuab．
DO 23 Iaq，N
IF（ABS（RI（I））．LT．EPS）Q■Q＋1
23 CONTINUE
CALL ERHES（RI，Y，S1，S2，S3，S4）
WRITE（2．290）S1，S2．S3．S4
 －E20．11／5x．＇LOGく108 OF EUCL．NORM OF RESIOUALS RI（I）＝．
＊E20．11／5X，1LOGく10I OF R＊（XK－X）E1，E20．11／5X．
＊iLOG＜1O＞OF EUCLM NORM OF（XK－X）EFE20．11／／）
\(K=K+1\)
IF（R．EQ．NN）GO TO 35
IF（K，GEyKH）GO TO 45
DO 21 Imin
\(21 \mathrm{VS}(\mathrm{I})=\mathrm{RI}(1)\)
C
VS－HOLDS－mR（I＋1）
CALL FBSUBS（N，M，RIB，RI）
C RIn－HULDS～MR＊（I＋I）
C CALCULATION OF BI
```

        ANU|II =0
        DO 22 I=1,N
    22
        BI #ANUMII / ANUH
        WHITE (2";220) BI
    220 FORIIAT (1H,5X,'SCALAR BI =1,E20.19/10X,IERROR HEASURES'1)
    CALCULATIUN OF SII=am
    DO 42 l=1,N
    42 SI(I)=RI(I)*BI*SI(I)
    C SIm-HOLDS-mS(I+1), RI=-HOLDS=-R*(I+1)
DO 31 I=1,N
S(I)aRI(I)
31 RI(I)EVS(I)
S(1)-mHOLDSM-R*(I*1), RI(I)--HOLDS--R (I+1)
GO TO 25
Cr-m
Cym- TERHINATION OF INNER LOOPNOM.
C%-0
35 WRITE (2;9190) K
190 FORHAT\1H 05X,'NUMBER OF ITERATIONS KEI.I5///6X;
*'fINAL SOLUTION Y(l) '///)
DO 46 bai,IH
J=|HOL+1
46 WRITE(2,170) (Y((J-1)*IH+I),I=1,IH)
170 FURMAT(1H 192F10.6)
IF (R.EQ.1.OR.R.EQ.4) GO TO 944
GO TO 945
944 WRITE (2'0゙260)
260 FORHAT (9H ,5X,'rECURSIVE RESIDUAL RI(I)'///)
DO 48 L=9.1H
JEIHmL*1
48 WRITE (2;170) (RI((J-1)*IH+1),I=1,IH)
945 EPS EEPS/100.0
IF (EPS.LT.1.0Em10) GO TO 65
GO TO 550
65 R =R+1
IF (R,LE:4) GO TO }66
STOP 11
4 5 ~ W R I T E ~ ( 2 " 6 0 ) ~ K M ~
60 FORIIAT(1H 15X,'NO CONVERGENCE AFTER '185.' (TERATIONS')
STOP 111
55 WRITE (2:230)
230 FORHAT(1H /5X,'THE COEF: HATRIX IS NOT POSITIVE DEFINITE'/)
STOP 1111
END

```
        SUBROUTINE APNOFA(N,M,IR,A,B,C)
    THIS SUBROUTINE IS AN APPRUXIMATE NORMALIZED FACTORIZATION
    of a SQare matrix of order n. the coefficient matrix is
    SYMIIETRIC, POSITIVE DEFINITE, QUINDIAGONAL OF SEMIBANDHIDTA
    M ( \(3<M<N\) ) , A,B,C ARE VECTORS CONTAINING RESPECTIVELY THE
    DIAGONAL CO DIAGONAL AND M-TH DIAGONAL ELEHENTS. THE MATRIX
    IS FACTORIZED INTO DS TST TS DS , WHERE DS IS A DIAGUNAL
    MATRIX AND TS IS A REAL UPPER TRIANGULAR MATRIX WITH UNIT
    DIAGONAL ELEMENTS, NON-ZERO ELEMENTS IN SUPER OIAGONAL.
    RETAINING IR E[1"M-1] OUTERMOST OFF-DIAGONAL ENTRIES:
    TST DENOTES THE TRANSPOSE OF TS. ONE RESULT VECTOR OF LENGTK
    (M-1) AND ONE ARRAY OF DIMENSION (IR,N-H+1〕 ARE USED AS
```

    WURKSPACE , N1¥H-M+1 , M9:11-9.
        DIMENSION A(N), B(N),C(N1),U(M1)
        COMIION T(IR,N1)
        \(A(1)=\operatorname{SQRT}(A(1))\)
        DO 1 Im \(2^{\prime \prime \prime} M m 1\)
        \(Z=B(I-1) / A(I-1)\)
        \(A(I) E \operatorname{SQRT}(A(I)-2 * Z)\)
    \(1 B(I-1)\) (2/A(I)
        \(z=0\)
        DU 5 Jmi!nmill
        \(U(1)=C(J) / A(J)\)
        \(V=B(M+J-2) / A(M+J=2)\)
        IF (J. QT.IRmI) GO TO 32
        \(002 \downarrow=21 / R+1-J\)
    \(2 U(I)=-B(1+J-2) * U(I-1)\)
    32 IF (J.EQ.1.OR.IR.EQ.1) GO TO 33
IF (J.GT: $\| R$ ) $\| P=2$
IF (J. $\mid E=\| R$ ) IP=』R-J+2
IK1 = IR由1
DO 3 ImIPIIR
$2=0$
DO $53 \mathrm{~K}=1$ 1 I m 1
$53 \quad 2=2+U(K) * T(K-I+I R 11 I+J-I R I)$
$U(I)=-B(I+J-2) * U(I-1)=Z$
CONTINUE
00 4 K—1"Mm2
$Z \approx Z+U(K) * U(K)$
CONTINUE
$6 \quad A(M+J-9)=S \cap R T(A(M+J-1)-2-(U(M-1)+V) *+2)$
$B(M+J m 2)=V / A(M+J-1)$
DO 5 dn1: 1 R
$T(I, J)=U(I) / A(M+J=1)$
RETURN
END

```
    \(33 \quad 2=0\)
    SUBROUTINE GEDRHS(N,A,S)
    THIS SUBROUTINE FORMS THE VECTOR Z,WHERE D*Z.S.
    D IS A DIAGONAL MATRIX OF ORDER NO WHOSE NON-ZERO DIACONAL
    ENTRIES ARE STORED IN A VECTOR A OF NEELEMENTS. THE INPUT
    VECTOR S IS OVERWRITTEN BY THE RESULT.
    DIMENSION A(N), S(N)
    DO 6 In \({ }^{\prime \prime} \mathrm{N}\)
6

SUBROUTINE FBSUBS(N,M,IR,B,S)
THIS SUBROUTINE SOLVES THE SET (TST TS) EVES OF NOLINEAR EQUATIONS, WHERE TS IS AN UPPER TRIANGULAR MATKIX OF BANDUIOIM M WITH UNIT ELEIIENTS ON THE DIAGONAL. TST DENOTES THE TRAHSPUSE OF TS. THE NON-ZERU ELEMENTS ARE ON SUPER DIAGONAL AND IR E\{IIII-IJ OUTERMOST UFF-DIAGONAL ENTRIES AKE RETAINED. THE SOLUTION IS EFFECTED BY A FORWARDOBACKWARD SUBSTITUTION PROCESS WHERE THE INPUT VECTUR S IS OVERWRITTEN SUCCESSIVELY BY THE INTERMEDIATE SOLUTION(OBTAINED BY FORWARD SUBSTITUYION)
```

C AND THE FINAL SOLUTIOH(OBTAINED BY BACK SUBSTITUTION).
C N1=N-N+1
DIMENS\&ON B(N),S(N)
COMHON T(IR,N1)
DO 7 1=2'MM-1
7 S(I)= S(I)* B(I-1) *S(I-1)
DO 8 IEHT゙N
2=0
DO 9 Kal-M+1,I-II+IR
9 z= Z+T(K-l +M, I-1|+1)*S(K)
3 S(1)=S(1)mB(I-1)*S(1-1)-2
IF (M.LT.(N/2+1)) GO TO 34
IF (M,EQ.NN) GO TO 35
CHE= CASE: H> [N/2+1].
DO 10 II=1,N-M
I=N-II
2=0
DO 11 J=I+9,N
IF (ImJ+M.GT.IR) GO TO 19
z= 2+T(I-J+||j-1|+I)*g(J)
11 CONTINUE
10 S(I)=S(1)-B(I)*S(I+1)-2
35 DO 12 II=1,2*M-N-1
I=M-1I
2=0
DO 13 JaM,N
IF (I-J+M.GT.lR) GO TO 13
Z=Z+T(I-J+M,j-M+1)*S(J)
13 CONTINUE
12 S(I)= S(1)ma(I)*S(1+1)m2
IF (M,EQ.N) GO TO 36
DO 14 II=1,NmM
I=N-M+1-1!
2=0
DO 15 JmM,I+M-1
IF (INJ+9.GT.IR)GO TO 15
Z=2+T(I-J*M少-M+1)*S(J)
15 CONTINUE
14 S(I)=S(I)mb(I)*S(I+I)-Z
GO TO 36
34 IF (((N-N/2*2),EQ,O).AND.(M,EQ.N/2)) GO TO 37
CHO- CASE:M<[N/2+1]:
DO 16 1/=1,M-1
I=N-II
Z=0
DO 17 Jal+1,N
IF (ImJ+M,GT.IR) GO TO 17
2= 2+T(1-J+M,J-M+1)*S(J)
17 CONTINUE
16 S(I)=S(I)nB(I)*S(I+1)-2
DO 18 II=1,N-2*M*1
IaN-M+9-1!
2=0
DO 19 J=1+1,I+N-1
IF (ImJ+M.GT.IR) GO TO 19
z= z+T(I~J* |, J-||+1)*S(J)
19 CONTINUE
98 S(I)=S(1) mB(I)*S(1+1)-2
DO 20 1Im1,H-1
I=M-1]

```

    GU TO 35
    15 IF (M.EG.N/2) GO TO 25
Cr-a CASL : H<N/2.
    DO 5 I=2,N-1
    5 S(I)=S(I)-(B(I-1)*x(I-1)+A(I)*x(I)+B(I)*x(I+1)+C(I)*x(N+I-1))
        DU O I = 1%N-N+1
        S(I) =S(I)-(C(I-11+q)*x(I-H1+1)+H(1-1)*x(I-1)+A(I)*x(I)*
        *B(I)*X(I+1)+C(I)*X(I+11-1))
    6 CUNTINUE
        DO 7 ImN-N+2,N-1
        S(I)=S(I)-(C(I-11+1)*x(I-M+1)+B(1-1)*x(I-1)+A(1)*x(1)*
        *B(I)*X(1*1))
    7 cuntinue
        GU TO 35
C--- CASE : H=N/2.
    25 DU 8 IN2'NN-H-1
```



```
        DU& 1aN-H,N-M+1
        S(I)=S(I)-(C(I-N+M+1)*X(I-N+M+1)+B(I-1)*x(I-1)+A(I)*x(I)*
        *B(I)*x(I+1)+C(I)*x(I+1I-1))
        cONTINUE
        DU 11 l=N-N+2,N-1
        S(I)=S(I)-(C(I-N+M+1)*X(I-N+M+1)+B(I-1)*X(I-1)+A(I)*x(!)*
    *B(I)*X(I+1))
    11 CONTINUE
    35 S(N)=S(Ny- (C(N-M*1)*X(N-11+q)+B(N-1)*X(N-1)+A(N)*X(N))
        IF (IND.EU.1) GO TO ?O
        RETURN
    20 DU 12 }l=1,
    12 S(I)=-S(I)
        RETURN
        END
```



KII IS AN UPPER LIMIT ON THE NUMBER OF ITERATIONS. $K H=00$
$1 K=0$
$G V=0 . U$
DO $17 \quad I=1 \cdot \mathrm{~N}$
$17 \quad V(I)=G V$
C Y-FHOLDSE- THE INITIAL APPROXIMATE SOLUTIUN Y O -
C.ALL GEVECT (N,P,MS,IX,A,B,C,H, S)

CALL RESIDSD(Y,IX,S)
C $\quad S=-H O L D S=-R \quad 0$
DU $24 \quad 1=1, N$
$24 R 1(1) E S(I)$
RI- $-11 O L D S=-R 0$
CALL APNUFASD(N,P,H,IRF,IRG,A,R,C,H)
CALL GEDRHS (N, A,S)
CALL FBSUBSZD(N,P,M,B,S)
$S=-110 L D S=R * 0$
DO $28 \quad\{=1, N$
$28 \quad S I(1)=S(1)$
C $\quad$ SI- $-H U L D S=-R * 0$
C CALCULATIUN OF AI,Y,RI.
IXEI

## C-m

C - $-\infty$ THE INNER LOOP $-\infty$
C
C CALCULATIUN OF PRODUCT W*SI.
25 CALL RESID3D(SI,IX,VS)
C $S I-=H U L D S=-\operatorname{SO} \quad$ VS=-HOLDS-W*SO.
C CALCULATIUN OF AI
ANU\|=O
DLNOHEO
DU $49 \quad 8=1, N$
ANUII $=A N U M+R I(I) * S(I)$
C RI =-HOLDSmmRO, SI=-HOLDS=-R*O
41 DENOM TDENOH+SI(I)*VS(I)
C VS=-HULDS $\quad$ - $-W * S O$.
IF (DENOM.WE.O) GO TO 55
AI =ANUII /DENOM
IIRITE(2.210) AI
210 FORMAT (9H,5X, SCALAR AI = .E20.91/)
C CALCULATIUN OF Y,RI.
DO 19 I=1, N
$Y(I)=Y(I) \oplus A!\star S I(I)$
99 RI(I) ARI(I)-AI*VS(I)
C RI=-HOLDS-mR(I+1)
C TEST IF THE CONVERGENCE IITH THE RECURSIVE RESIDUAL.
$00231=1, N$
IF (ABS(RI(I)).LT,EPS) $Q=Q+1$
23 CONTINUE
CALL ERIIES(RI,Y,S1,S2,S3,S4)
WRITE (2;130) S1;S2.S3 ©S4

*F20.11/5X,'LOG<10> OF EUCL. NORM OF RESIOUALS RI(I)E.
*E20.11/5X,1LOG<10> OF R* (XK-X) $=1 . E 20.11 / 5 X$.

* 1 LOG<1O〉 UF EUCL. NORII OF $(X K-X)=: \because E 20.11 / /)$ $K=K+1$
IF (R.EQ.N) GO TO 35
IF (K,GE.KM) GO TO 45
DO $21: I=1, N$
c 21 VS(1)=RI(I)
C VS-HULDS-MR(I+1)
CALL FBSUBS3D (N,P,H,B,RI)
C RIm-HOLDS-R $*(I+1)$
C CALCULATION OF BI
Allull =0
DO $22 \quad d=1, N$
22 ANUII RANUHY + VS(1)*RI(I)
BI =ANUHI /ANUM
WRITE (2;220) BI
220 FURIIAT ( $9 \mathrm{H}, 5 \mathrm{X}, \mathrm{S}$ SALAR BI $=\mathrm{e}$, E<0.11///) calculation of Si.
D) $42 \quad 1=1, N$

42 SI(I) RRI(I) +BI*SI(I)
c SI-HOLDS-GS(I+1), K(I)-HULDS-R*(I+1)
DO $31 \quad \mathrm{i}=\mathrm{I}$, N $S(I)=R I(I)$
$31 R I(I)=V S(1)$
C S(I)--HOLDS-R* (I+1),RI(I)-HOLDS--R(I+9) GU TO 25
C-m TER TERMINATION OF INNER LOOP-.-C.-.

35 WRITE (2',190) K

WRITE (2.260)
260 FURIIAT(1H,5X, 'NORMALISED RESIDUAL RI(1)'/I) DO $48 \mathrm{~L}=1$ 1/ H 1 $J=1+11-6+9$
48 WRITE (2:170) (RI((J-1) $+1 H+1) .1=1.1 H)$
170 FORIIAT ( $1 \mathrm{H}, 12 F 10.6$ )
C Y-HOLDS THE FINAL SOLUTION --
WRITE (2;190)

DO $01 \mathrm{~L}=\mathrm{F}, \mathrm{BH}$
$J=1 H 9-h+1$
61 WRITE (2.170) (Y((J-1)*IH+1),I=1,1H)
EPS $=E P S / 100.0$
If (EPS.LT.1.0E-10) GO TO 65
GO TO 550
$65 \quad 1 K G=I R G+3$
IF (IKG.LE,2) GO TO 666
$1 R F=1 R F+3$
IF (IRF.LE,4) GO TO 666
STOP 19
45 WRITE $(2,00)$ KM
60 FORMAT(1H1,5X,'HO CUNVERGENCE AFTER ',IS.' ITERATIUNS') stop 191
55 WRITE (2"230)
230 fURIIAT(1H1,5X, THE CUEF. MATRIX IS NOT POSITIVE DEFINITE'/I) END

```
    M-th diagunal and p-th diagunal elements. the matkix is
    FACTOKIZED INTO DS T个S TS DS , WHERE DS IS A DIAGONAL MATRIX
    AND TS IS A REAL UPPER TRIANGULAR MATRIX OF BANDWIDTHS M AND
        P. WITH UNIT DIAGONAL ELEMENTS, NON-ZERO ELEMENTS IN SUPER
        DIAGONAL,RETAINING IRF E\{1,M-1] AND IRG E[1, P-1J OUTERMOST
        OFF-DIAGQNAL ENTRIES. TTS DENOTHS THE TRANSPOSE OF TS.
        three result vectors of lengih n and tho arrays of dimensiun
        [IRF,NmPI+1〕,[IRG"NmP+1] ARE USED AS WORKSPACE.
        \(N 1=N-M+1, N 2=N-P+1\).
        DIIENSION \(A(N), B(N), C(N), H(N), U(N), X X(N), Y Y(N)\)
        CUMIION F(IRFON1): Y(IRGON2)
        INTEGER P
        i \(\mathrm{X}=1\)
        IRF1=1RF+1
        \(1 R G 1=1 R G+1\)
        CALL GEVECT (N,P,M IX,A,B,C,H,U)
        DO 11 IzP,N
        XX(I-1)=B(1-1)
    \(11 \mathrm{YY}(\mathrm{I})=\mathrm{A}(1)\)
        \(A(1)=\operatorname{SQRT}(A(1))\)
        DO 1 Im2.Mm1
        2天B(1-1)/A(I-1)
        \(A(I)=S Q R T(A(I)-Z * Z)\)
        B(I-1) mz/A(I)
        D() 6 Jm1'Nmil 1
        U(1) =C(Jy/A(J)
        \(V=B(M+J-2) / A(M+J-2)\)
        If (J.GT.IRF-1) GO TO 7
        DO 2 1m2;idRImJ
        2
    U(I) \(=-B(I+J-2) * U(I-1)\)
    If (J.EQ.1.OR.IRF.EQ.1) GO TO 8
    IF (J.GT.IRF ) IP=2
    IF (J.|EE.IRF) IP=IRF=J+く
    DU 3 IalpilRF
    \(z=0\)
    D0 4 Ka1:! 1 m
    \(Z=2+U(K)+F(K-l+I R F 1, l+J-!R F 1)\)
\(3 \quad U(1)=-B(!+j-2) * U(l-1)-2\)
\(8 \quad 2=0\)
    DO 5 K. 1 "Mr2
    \(5 \quad Z=z+U(K) * U(K)\)
    \(A(M+J-1)=\operatorname{SQRT}(A(M+J-1)-Z=(U(M-1)+V) * * 2)\)
    \(B(11+J-2)=\mathrm{V} / A(11+J-1)\)
    DO 6 In1: IRF
    \(6 \quad F(1, J)=(J(I) / A(M+J=1)\)
    14 CONTINUE
    DU \(18 \quad 1=\mathrm{D}-1, \mathrm{~N}=1\)
    \(18 \quad B(1)=X X(1)\)
    OU 19 InP, N
    \(19 \mathrm{~A}(\mathrm{I})=\mathrm{YY}(\mathrm{I})\)
    DO 13 ! EI , N
    13 U(1)EA(I)
    \(x \times(p-1)=1(p-1)\)
\(957 \mathrm{M1}=\mathrm{pm} 11\) \$1
    DO 12 Jai, \(N m p+1\)
    JJ \(=\mathrm{J}+\mathrm{Pm}\) /l
C INITIAL GUESS VALUE OF \(D(P+J-1)=D(P+J-2)\)
    \(A(P+J-1)=A(P+J-2)\)
\(502 \mathrm{Y}(1, J) \pm H(J) /(A(J) * A(P+J \sim 1))\)
    \(V=B(P+J \sim 2) /(A(P+J+2) * A(P+J-1))\)
```

```
    972 IF (J.GT.IRF-1) GO Y0 31
        DO 33 \=2, (RFI-J
        Y(I,J)R-B(I+J-2)*Y(I-1,J)
        IF (J.EQ.1,AND.I.EQ.M1) Y(I;J)=Y(I,J)+C(I+J-I)/(A(I+J-1)*
    * A(P+J-q))
    33 CONTINUE
    31 IF (J.EQ.1.OR.IRG.EQ.1) GO TU 1S
    IF (J,GT.IRF ) IP=2
    IF (J.LE.IRF ) IP=IRF-J+2
    DO 16 l=|P,IRG
    IF(I.GE.PmJ+1.OR,J.GT.Pa2) GO TO 335
222 2=0
    IF (I.LT.M) GO TO 23
    DO 17 K=1,Mm1
    IF (K,GT.lRF) GO TO 17
    2=Z+Y(K+In|,J)*F(K_1+J-11)
17 CONTINUE
    GO TO 24
23 D0 26 K=1,Im1
    Z=2+Y(K,J)*F(K-I+IRFI,I+J-IRF1)
26 CONTINUE
24 Y(I,J)=-B(f+J-2)*Y(1-1,J)-2
    If (I,ER.M1) Y(I,N)=Y(I,J)+C(I+J-1)/(A(I+J-q)*A(P+J-1))
    GO TO 16
C
335 2:0
    DO 21 KE1,I-1
    221 2=2+Y(K-I+IRG1,I+\-IRG1)*Y(K ,J)
    21 CONTINUE
    Y(I,J)= - B (I+J-2) % Y(I-1,J)-2
    IF (I,EQ.119) Y(I,J)=Y(I,J)+C(I+J-1)/(A(I+J-1)*A(P+J-1))
    715 CONTINUE
    16 CONTINUE
    15 2=0
    WaV
    IF (I,EQ.P#G) W=V+Y(I,J)
    DO 161 KM1,Pm2
    IF (K.GT.IRG) GO TO 161
    Z=Z+Y(K,J)*Y(K,J)
161 CONTINUE
    22=0
    D0 717 Ka1,11-2
    IF (K.GT.lRF) GO TO 717
    z2=z2+F(k,JJ)*F(K,JJ)
717 CUNTINUE
974 A(P+J-1)=SQRT(U(P+J-1)/(1+2+22+W**2))
    IF (ABS(A(P+J-1) -XX(P+J-2)),LT.1.0E-6) GO TO 191
    XX(P+J-2)=A(P+J-1)
    GO TO S62
191 IF (J.ER.1) XX(P-1) Eu(P-1)
    XX(P+J-1)=A(P+J-1)
    121 B(P+J-2)=B(P+J-2)/(A(P+J-2)*A(P+J-1))
    12 cONTINUE
        RETURN
        END
    SUBROUTINE GEDRHS(N,A,S)
C THIS SUBROUTINE FORHS THE VECTUR ZOWHERE D.Z=S.
```

```
    C D IS A DIAGONAL MATRIX OF UROER N. WHUSE NON-ZERO DIAGONAL ENTRIES ARE STORED IN A VECTOR A OF N-ELEHEVTS. THE INPUT VECTOK S IS OVERWRITTEN BY The reSUlt. DIMENSION A(N),S(N) DU 2 IF1.N
S(I) \(=S(1) / A(I)\)
RETURN
END
```

SUBROUTINE FBSUES $3 D(N, P, M O B, S)$
C THIS SUBROUTINE SOLVES THE SET (TTS TS) \& YES OF N-LINEAR
EQUATIONS, WHERE TS IS AN UPPER TRIANGULAR HATRIX OF GANDWIDTHS II AND P, WITH UNIT EIEMENTS ON THE DIAGONAL. tts denotes the transpose of ts. the non-lero elements are OH SUPER DIAGUNAL AND IRF E(1,M-I],IRG E(1,P-1) OUTERMOST OFF-DIAGONAL ENTRIES ARE RETAINED.THE SOLUTION IS EFFECIED BY A FURWARD-BACKWARD SUBSTITUTION PROCESS WHERE THE INPUT VECTOR S IS OVERWRITTEN SUCCESSIVFLY BY THE INTERHEDIATE SULUTION (OBTAINED BY FORWARD SUBSTITUTION) AND THE EINAL
SOLUTION (OBTAINED BY BACK SUBSTITUTION), NIEN-II+1, ON2EN-P+I.
DIHENSION B(N),S(N)
CUIIION F(IRF,N1), G(IRGON2)
COMHION /BLOCKS/IRFIIRG
INTEGER P
DO 44 l=2, M-1
$44 \quad S(I)=S(I)-B(I-1) * S(I-1)$
IF (M.ER,N) GO TO 396
DO 78 dचM, P-1
$2=0$
DU $79 K=I-M+1, I m+I R F$
$79 \quad 2=2+F(K-I+M, I-11+1) * S(K)$
$78 \quad S(I)=S(I)-B(I-1) * S(I-1)-2$
396 DO $87 \mathrm{l}=\mathrm{P}, \mathrm{N}$
$2=0$
DO $88 K=I-M+1, I-M+I K F$
$80 \quad 2=2+F(K-1+1,1-M+1) * S(K)$
21=0
DO $39 K=1-P+1, I r P+1 R G$
$89 \quad 21=21+G(K-1+p, 1 \sim p+1)+S(K)$
$87 S(1)=S(1)-B(1-1)+S(1-1)-2-21$
IF (M.LT. (N/2+1)) GO TO 64
IF (H.EQ.N) GO TO 65
IF (P.GT. (N/2+1)) G( TO 18S
C.-. CASE: M>[N/2+1], P DOES NOT EXIST
$0047 \quad 11=1, N m$
1aN-I!
Z=0
DO $48 \mathrm{~J}=1+1, \mathrm{~N}$
IF (I-J+M.GT.IRF) GO TO 48
$z=2+F(I-J+11, J-11+1) * S(J)$
48 CONTINUE
$47 \quad S(1)=S(1) \in B(1) * S(1+1)-2$
65 DO 49 dI=1,2*M-N-1
I= リーII
Z: 0
DU 51 JamoN
IF (I-J+M.GT.IRF) GO TO 51
$2=2+F(I-J+11, J=M+1)=S(J)$

```
    51 CONTINUE
    49 S(I)=S(I)mB(I)*S(I+1)-2
        IF (H,EQ.N) GU TO 66
        DU 52 11=1,N-11
        I=N-M+{-1!
        2=0
        DO 53 J=M!I+M-1
        IF (I-J+M.GT.IRF) GO TO 53
        Z=2+F(I-j+11,j-1H+9)*S(J)
    53 cuntinue
    52 S(I)=S(I) m(I)*S(I+1)-2
        GO TO 66
C--- CASE: H> (N/2+1], P>[N/2+1]
    183 IF (P.ER.N) GO TO 252
        DO 231 11=1,NmP
        I=N-1I
        z=0
        2.1=0
        DO 232 Jad+1,N
        IF (I-J+M.GT.IRF) GO TO 261
        z=Z+F(|-J+||,J-M+1)*S(J)
    269 IF (I-J+P.GT.IRG) GO TO 23<
    z1=21+G(1-d+P(Jmp+1)*S(J)
    232 CONTINUE
    231 S(1)=S(1)-B(1)*S(1+1)-2-21
    25? IF (M.EQ.N) GO TO 253
        DU 233 11=1,P-M
        I=P-11
        Z=0
        DO 234 Jml$1,N
        lf (Im\+M.GT.IRF) GU TO 234
        Z=2+F(I-J+M,J-M+9)*S(J)
    234 CUNTINUE
        21=0
        DO 235 J=H,N
        IF (I-J+P,GT,IRG) GU TO 2SS
        21=<1+G(1~J+P,J-p+1)*S(J)
235 CONTINUE
233S(1)=S(I)-B(1)*S(1+1)-2-21
253 DU 236 11=NmM+1.Mm1
    I=Nm!1
    Z=0
    2.1=0
    DO 237 JaM,N
    IF (ImJ+M.GT.IRF) GU TO 200
    z=2+F(d-j+H,j-1H+q)*S(J)
206 IF (I-J+P.GT.IRG) GO TO 23I
    z1=21+G(I~J+P,J-p+1)*S(J)
237 CUNTINUE
236 S(1)=S(1)-B(1)*S(1+1)-2-21
    IF (M,ER,N) GO TO 66
    DO 238 II=H,P-1
    I=N-I!
    2:0
    DO 239 JaM,I+M-1
    IF (ImJ+M.GT.IRF) GO TO 234
    Z=2+F(I-J+M,J-M+q)*S(J)
239 cONTINUE
    21=0
    DO 241 JEM,N
```

```
        IF (I-J+P.GT.IRG) GU TO 241
        z1=21+G(I-J*P,J-P*1)*S(J)
    241 CONTINUE
    23B S(1)=S(1)-B(1)*S(1+1)-2-21
        IF (P,EQ:N) GO TO 60
        DU 242 II#P,N-1
        I=N-11
        2=0
        7.1=0
        00 245 Jmll,I+H-1
        IF (I-J+M.GT.IRF) GU TO 271
        Z=2+F(I-j+M,j-M+1)*s(J)
        271 If (I-J+P.GT.IRG)GO TO 24S
        21=21+G(I-J+P,J-P+1)*S(J)
    24.3 CONTINUE
    242 S(I)=山(I)-B(I)*S(I+1)-2-21
        GO TO OG
    64 IF (P.GT.N.DR.P.LE.M) GO TO &21
    If (P.LT,(N/2+1)) GO TO 444
    If (((N-N/Z*2).EQ,O).AND.(M,EQ.N/Z)) GO TO 67
Cr-m CASE: M<{N/2+1j% P>{N/2+1].
    IF (P.EQ.N) GO TO 254
    DO 94 \I=1,N- P
    I=N-II
    Z=0
    21=0
    DO O5 J=1+1,N
    IF (I-J+P,GT.IRG) GO TO 211
    21=21+G(1-J+ P,Jm b+1)*S(J)
    219 IF (ImJ+M.GT.IRF) GO TO 95
    z=z+F(I-j+H,J-M+q)*S(J)
    95 CONTINUE
    94 S(I)=S(1)mB(I)*S(1+1)-2-21
    254 00 96 11ENm p+1:Mm-1
    I=N-11
    2=0
    DO 97 J=I+1,N
    If (INJ+M.GT.IRF) GO TO 97
    Z=2+F(I-J+11,J-M+1)*S(J)
    9% CONTINUE
    21=0
    DO 98 J= P,N
    IF (ImJ+P,GT.IRG) GO TO 98
    z1=z1+G(I-J+ P,jm P+I)*S(J)
    9d CONTINUE
    90 S(I)=S(I)-B(I)*S(I+1)-2-21
    DO 101 1IFH,N-M
    I=N-II
    Z=0
    DO 10< J=!+1.l+||=1
    IF (Im\+M.GT.IRF) GO TO 90&
    z=2+F(i-j+11,j-11+q)*S(J)
102 CONTINUE
    21=0
    DO 10S Ja P,N
    IF (I-J+P,GT.IRG) GOTO IOS
    21=21+G(1-J+ P,Jm P+1)*S(J)
    103 CONTINUE
    101 S(I) =S(I)-B(I)*S(1+1)-2-21
    DU 104 II =N-M+1, P-1
```

```
    I=N-Il
    Z=0
    DO 10S J=M,I+M-1
    IF (I~N+M.GT.IRF) GO TO 10S
    2=2+F(l-J+M,J-H+1)*S(J)
    105 CONTINUE
    21=0
    DU 100 J= P,N
    IF (I-J+R.GT.IRG) GO TO 100
    21=L1+G(I-J* P,Jm P+1)*S(J)
    106 CUNTINUE
    104 S(I)=S(I)-B(I)*S(I+1)-2-21
    IF (P.EQ.N) GO TO 66
    DO 121 II= P,N-1
    I=N-II
    Z=0
    DO 12< J=M, 1+M-1
    IF (I~J+H.GT.IRF) GO TO 122
    Z=Z+F(I-J+|!,J-11+q)*S(J)
    122 GUNTINUE
        21=0
        DO 123 J=P.I+ P=1
        IF (ImJ+P.GT.IRG) GU TO 12S
        Z1=L1+G(I-J* P,Jm P+1)*S(J)
    123 CONTINUE
    121S(I)=S(I)-B(I)*S(I+1)-2m21
        GO TO 66
C=-m CASE: M=N/2 ,P>[N/2+1].
    67 IF (P.EQ.N) GO TO 256
        DU O1 II=1,NmP
        I=N-II
        Z=0
        21=0
        DO 62 N=I+1,N
        IF (I-J+M,GT.IRF) GO TO 281
        Z=Z+F(|-J+M,J-M+1)*S(J)
    281 IF (ImJ+P.GT.IRG) GO TO 62
        Z1=29+G(I-J+P.J~P+1)*S(J)
    62 CONTINUE
    61 S(I)=S(I) - B (I)*S(I+1)-Z-21
    256 DU 144 II=9,P-11-1
        I=P-1!
        Z=0
    DO 142 Jmat+1,N
    IF (ImJ+M,GT.IRF) GO TO 14L
    Z=2+F(J-j+M,J-M+1)*S(J)
    142 CONTINUE
    21=0
    DU 143 JmPNN
    IF(ImJ+P.GT.IRG) GOTO I4S
    Z1=21+G(I-J+P,J-P+1)*S(J)
    145 CONTINUE
    141S(I)=S(I)-B(I)*S(I+1)-2-21
        I=M
        Z=0
        DU 146 JaM*1,N=1
        IF (ImJ+M.GT.IRF) GO TO 140
        Z=Z+F(lmJ+M,J-H+1)*S(J)
    146 CONTINUE
        Z1=0
```

```
    DO 14T JEP,N
    If (I-J+P.GT.lRG) GU TO 14!
    Z1=21+G(1-J+P,J-P+1)*S(J)
    147 CONTINUE
    S(I)=S(I)-B(I)*S(I+1)-2-21
    D0 14४ II=9,M+P-N-1
    1=1月-11
    z=0
    DO 144 JmH,I IH-1
    IF (ImJ+M.GT.IRF) GO TO 144
    z=Z+F(I-J*H,J-M+1)*S(J)
    149 cuntiNuE
    z1=0
    DO 158 JEP,N
    IF (I-J+P.GT.JRG) GO TO 158
    21=21+G(1-J+P,J-P*1)*S(J)
    938 CUNTINUE
    148 5(1)=S(1)-8(1)*S(1+1)-2-21
    If (P.EQ.N) GO TO 66
    DO 15< II=P,N-1
    I=N-1d
    Z=0
    D() 150 J=N,I+N-1
    IF (ImJ+M,GT.IRF) GO YO 950
    z=Z+F(I-J+M,J-H+1)*S(J)
    156 CONTINUE
    z1=0
    DU 154 J=P!l+p-1
    IF (I-J+P,GT,IRG) GO TO 1>4
    21=21+(G(I-J+P,J-p+1)*S(J)
    154 CONTINUE
    152 S(I)=S(1)-B(I)*S(1+1)-2-21
    gu TO 66
Cr-- CASE: M<[N/2+1], P<{N/2+1]
    444 If (((N-N/Z*2).EQ,O).AND.( P.ER.N/2)) 6OTO 107
    D(1 54 |Im1,|-1
    I=N-II
    z=0
    7.1=0
    DO 55 J=1*1,N
    IF (I-J+P.GT.IRG) GU TO 231
    71=21+G(ImJ+ P,Jm p+1)*S(J)
    231 IF (IrJ+M,GT.IRF) GO TO 55
    Z=2+F(I-J+M,j-M+1)*S(J)
    55 CUNTINUE
    54 S(1)=S(1)mB(1)*S(1+1)-2-21
    DO 56 11:M, P-1
    I=N-II
    z=0
    n() 57 Jal*1,I+M-1
    If (ImJ+M.GT.IRF) GO TO 57
    z=2+F(I-J*H,J-M+1)*S(J)
    57 CuNTINUE
    21=0
    DO 91 \aI+1,N
    IF (I-J+P.GT.IRG) GO TO 91
    z1= (1+G(1-J* P,J- P+1)*S(J)
91 CONTINUE
50 S(I)=S(1)-B(I)*S(I+1)-2-21
    DO 124 II= P,N= P
```

```
    |=Nm||
    Z=0
    DU 125 J=1*1.I+||=1
    IF (ImJ+M.GT.IRF) GO rO 12S
    z=Z+F(i-j*M,j-M+1)*S(J)
    125 CONTINUE
        21=0
        DO 12G J=l*1.lt p-1
        If (ImJ+P.GT.IRG) G() TO 120
    Z1=Z1+G(ImJ+ P,Jm P+1)*S(J)
    126 CONTINUE
    124 S(I)=S(I)-B(I)*S(I+1)-2-21
    DO 12% II=Nm P+1,N-11
    I=N-II
    2:0
    DO 128 Jal+1,I+||=1
    lF (Im\+M,GT.lRF)GU TO 12ठ
    z=z+F(I-J+11,j=M+q)*S(J)
    128 contINUE
    21=0
    D0 129 Jm Polt P-1
    IF (I~J+P.GT.IRG) GU TO 129
    z1=21+G(Imj+ P,Jm P+1)*S(J)
    129 CUNTINUE
    127 S(I)=S(I)mB(I)*S(I+1)m2-21
        DO 58 \I=1,||=1
        1FM-1!
        2=0
        DO }59 J=M,\+M-
        IF (ImJ+M.GT.IRF) GO TO 59
        Z=Z+F(I-J+M,J-M+1)*S(J)
    59 continue
        z1=0
        DU 93 Ja Pilt Pr1
        IF (IPJ+P.GT.IRG) GO TO 93
        z1=z1+G(Imj+ P!jm p+1)*S(j)
    93 CUNTINUE
    58 S(1)=S(1)mB(1)*S(1+1)-2-29
    G0 T0 66
Cra- CASE: M<{N/2+1},PzN/2
    107 DO 108 II#1,N- Pm1
        IaN-II
        2=0
        21=0
        DO 109 Jal* 1,N
        IF (I~J+P,GT.IRG) GO TO 251
        z1=z1+G(ImJ+ P,Jm P+1)*S(J)
    251 IF (I-J+M.GT.IRF)G0 TO 10Y
    z=2+F(d-'J+H,JmN+1)*S(J)
    107 CONTINUE
    103 S(1)=S(1)-B(I)*S(I+1)-2-21
        I=p
        z=0
    DO 119 JmI*1,N
    IF (ImJ+M,GT.IRF) GO TO 111
    Z=2+F(I-J+M,J-M+1)*S(J)
    111 CUNTINUE
    21=0
    DO 112 J= P+1,N-1
    IF (ImJ+P.GT.IRG) GO TO 11<
```

```
    21=21+G(I-J* P.Jm P+1)
    112 CONTINUE
        S(1)=S(1)=B(1)*S(1+1)-2-21
    D0 113 IIMN- P+1,N-M=1
    1FN-1!
    2=0
    D0 114 JmI+1,N
    IF (ImJ+M.GT.IRF) GO TO 114
    Z=Z+F(I-J*11,j-M+1)*S(J)
    114 CONTINUE
    Z1=0
    DO 115 Ja P,It P=1
    IF (I-J+P,GT.IRG) GO TO 11S
    z1=21+G(I-J+ P,Jm P+1)*&(J)
    115 cONTINUE
    113 S(1)=S(I)mB(I)*S(1+1)=2-21
        IFM
    2口0
    DO 116 JaH+1,N-1
    IF (ImJ+M.GT.IRF) GO TO 116
    Z=Z+F(d-'J+M,JmM+1)*S(J)
    116 CONTINUE
        z1=0
        DO 117 Jm Polt R-1
        IF (ImJ+P.GT.IRG) GO TO 111
        z1=21+G(1-j+ P,J% P+I)*S(J)
    117 CONTINUE
    S(I)=S(I)\-B(I)*S(I+1)-2-21
    DO 118 IIPNmM+1,Nm1
    \ENm!\
    2=0
    DO 119 JmM1I+M-1
    IF (I~J+M.GT.IRF) GO TO 119
    Z=Z+F(|-J+M,J-M+1)*S(J)
    119 CUNTINUE
    21=0
    DO 131 J= P,It P=1
    IF (ImJ+P,GT.IRG)GO TO {31
    21=29+G(1mJ+ P,Jm p+1)*S(J)
    131 CONTINUE
    118 S(I)=S(15-B(I)*S(l+1)-2m21
    G0 TO 66
    821 IF ((NmN/2.2).EQ.O.AND.IN.EQ.N/2) 6O TO 822
Crm= CASE: M<[N/2+1] , P DOES NOT EXIST
    D0 381 1!口1,M-1
    IaNm!!
    2=0
    DO 382 J=1*1.N
    IF (ImJ+M.GT.IRF) GO TO 382
    Z=Z+F(I-'J*M,J-M+q)*S(J)
382 CONTINUE
381 S(I)=S(1)-B(1)*S(I+1)-2
    D0 383 1!=1,N-2*M*1
    I=N-M+1-1!
    2:0
    DO 384 JaI*1,I+||=1
    IF(ImJ+M,GT.IRF)GOTO 384
    Z=Z+F(I-J*H,JmM+1)*S(J)
384 CONTINUE
383 S(I)=S(I)-B(1)*S(1+1)-Z
```

```
        DU 380 11m9,M-1
            12Mm1I
            200
            DO 387 JaM! I+H=1
            IF (I-J+H.GT.IRF) GO YO 38/
            Z=Z+F(I-J+M,J-M+1)*S(J)
    387 CONTINUE
    386 S(I)=S(I)m-B(I)*S(I+1)-2
            GO TO 66
    CP-= CASE: MEN/2 , P DOES NUT EXIST.
    822 DO 38K II=9,NmH-1
            I=N-1!
            2=0
            DO 389 Jal$1,N
            IF (I~J+M,GT.IRF) GU TO 384
            Z=Z+F(I-j* (1,J-M+1)*S(J)
    339 CONTINUE
    388 S(1)=S(I)-B(1)*S(1+1)-2
            1=1
            Z=0
            DO 391 Jm!+1,N-1
            IF (ImJ+M.GT.IRF) GO TO 391
            ZF2+F(ImJ+M,JmM+1)*S(J)
    391 CONTINUE
            S(1)=S(1)-B(1)*S(8+1)-2
            D0 392 1!m1,Mm1
            1aMm11
            2=0
            DO 393 JmM,l+M-1
            IF (ImJ+M.GT.IRF) GO TO 39S
            z=2+F(I-J+M,J~M+1)*S(J)
    393 CONTINUE
    392 S(I)=S(1)-B(I)*S(1*1)-2
    6 6 ~ C O N T I N U E ~
        RETURN
        END
    SUBROUTINE RESIDSO (X,IND,S)
    THIS SUbroutine calc|llates the corresponding TO The set woxas
    RESIDUALS WHERE THE COEFFICIENT MATRIX IS OF ORDER NO
    SYMHETRIC, POSITIVE DEFINITE,SEVEN DIAGONAL OF BANDHIDTMS N
    AND P (4<H<P<N/2) AND A,B,C,H,S ARE VECYORS CONTAINING
    RESPECTIVELY THE DIAGONAL, CO DIAGONAG, M-TH DIAGONAL.
    P-TH DIAGUNAL AND CONSTANT ELEMENTS. THE RESIDUAL IS
    OVERWRITTEN GY VECTOR S` IF INDEX INDGI THEN THE SUBROUTINE
    CALCULATES THE PRQDUCT S OF THE GIVEN SPARSE MATRIX Y EY PME
    INPUT VECTOR X.
    DIMENSION A(N),B(N)OC(N),H(N),S(N),X(N)
    INTEGER P
    COMIION /BLOCKI/N:P,N
    CALL GEVECT(N, P, M,IND;AOB,COH,S)
    S(1)=S(1)-(A(q)*X(1)*B(1)*X(2)+C(1)*X(H)*H(q)*X( P))
    DO 5 lm2';M=1
    S(I)=S(I)-(B(I-1)*x(I-1)*A(I)* X(I)+B(I)*x(I+1)*
    *C(I)*X(I+1-1)*H(I)*X(I+P-1))
    CONTINUE
    DO 13 & =M, P=1
    S(I)=S(I)~(C(I-11+1)*X(I-M+1)&B(I-1)*X(I-1)*A(I)*X(I)*
```

*B(I) $\left.\mathrm{H}_{\mathrm{X}}(1+1)+\mathrm{C}(1) * X(1+H-1)+H(1) * X(1+P-1)\right)$
13 CONTINUE DO $14 \mathrm{l}=\mathrm{P}, \mathrm{N}-\mathrm{P}+1$
S(I) $=S(1)=(H(I-p+1) * X(I-p+1)+C(1-M+1) * X(1-M+1)+B(1-1) *$ $* X(1-1)+A(1) * X(1)+B(1) * X(1+1)+C(1) * X(1 * M-1)+H(1)+X(1+p-1))$ CONTINUE DO $15 \quad \mathrm{I}=\mathrm{N}-\mathrm{P}+2, \mathrm{~N}-\mathrm{M}+1$
$S(1)=S(I)-(H(I-P+1) \star x(I-P+1)+C(1-M+1) * x(I-M+1)+B(I-1) *$

* $X(1-1)+A(1) * X(I)+B(I) * X(I+1)+C(1) * X(I * M-1))$


## CONTINUE

no 16 I $=N-M+2, N-1$
$S(1)=S(I)-(H(1-p+1) \star X(1-p+1)+C(1-M+1) * X(1-M+1)+B(1-1) *$
$* X(I-1) \neq A(I) * X(I)+B(I) * X(I+1))$
16 CONTINUE
$35 S(N)=S(N)-(H(N-P+1) * X(N-P+1)+C(N-M+1) * X(N-M+1) * B(N-1)$ * $X(N-1)$ * $A(N) * X(N))$

IF (IND.EQ,1) GO TO 20
RETURN
20 DO 12 \|=1,N
12 S(1)=-S(1)
RETURN
END


[^0]:    TABLE 5.1: Two dimensional case.
    Storage requirements and number of multiplications for $N \times N\left(N=n^{2}\right.$ ) systems, where $n=m-1$.

[^1]:    TABLE 5.2: Three dimensional case.
    Storage requirements and number of multiplications for

[^2]:    TABLE 6.2
    Parameters $\rho_{n}, \gamma_{n}, \tilde{\rho}_{n}, \tilde{\gamma}_{n}$ and work required to reduce the error to $1 E-6$,
    $1 E-8,1 E-10$ respectively for the $3 D$-model problem. The resulting
    sparse matrices for the above considered mesh sizes are of order 64 ,
    216,512,1000 respectively.

[^3]:    TABLE 6.4
    Computational work required to reduce the error to $1 \mathrm{E}-6,1 \mathrm{E}-8,1 \mathrm{E}-10$ respectively for the 3 D -model problem in
    cases A and B . The percentage of the gain in computational work (C.W.) is given as in Table 6.3 .
    The resulting sparse matrices for the above considered mesh sizes are of the order $64,125,512$ respectively.

