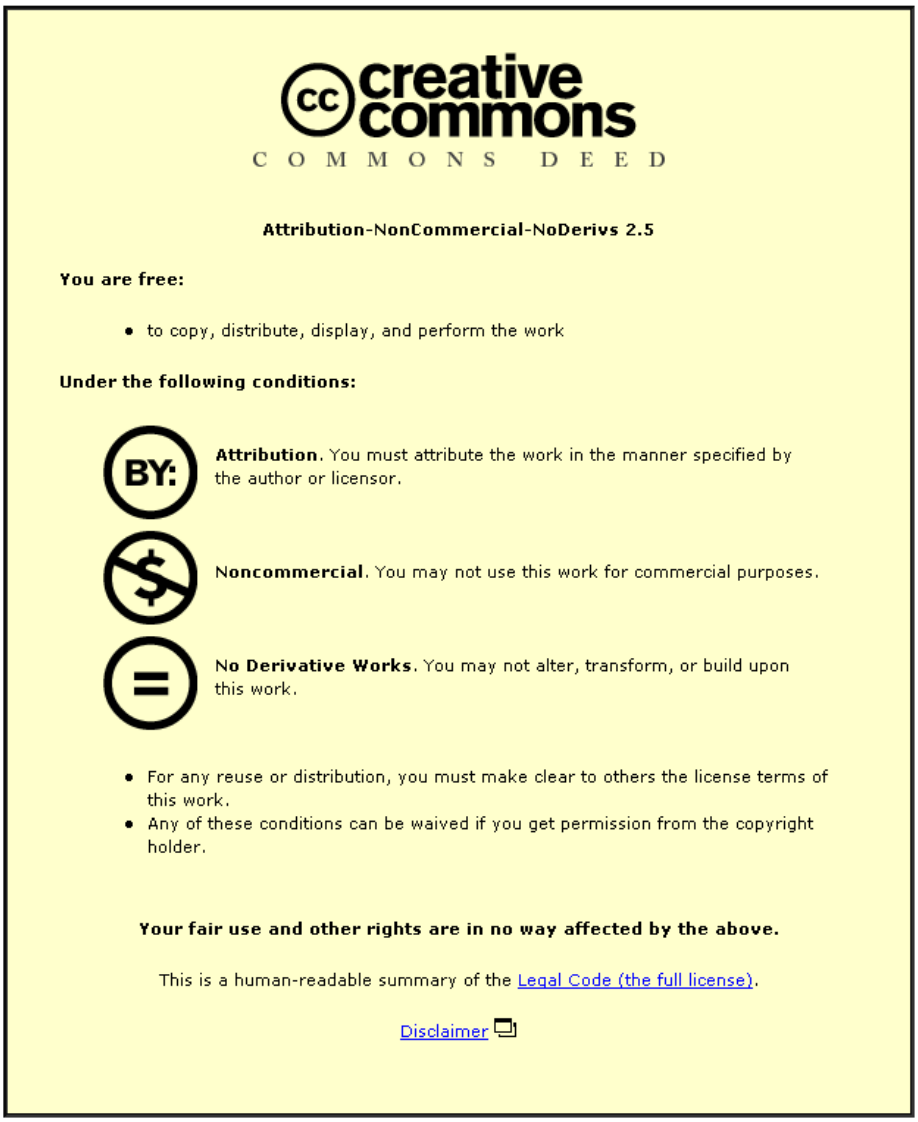
Loughborough University

This item is held in Loughborough University's Institutional Repository (https://dspace.lboro.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.


creative commons

COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

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

Under the following conditions:

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.

No Derivative Works. You may not alter, transform, or build upon this work.

- 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 Legal Code (the full license).

Disclaimer

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

THE SPLINE APPROACH TO THE NUMERICAL SOLUTION OF PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS

BY

NASHAT IBRAHIM KADHUM, B.Sc.,M.Sc.

A Doctoral Thesis

submitted in partial fulfilment of the requirements for

the award of Doctor of Philosophy of the

Loughborough University of Technology

August, 1988.

Supervisor: PROFESSOR D.J. EVANS, PH.D., D.SC.,
Department of Computer Studies

© by Nashat Ibrahim Kadhum, 1988.

DECLARATION

The work contained in this thesis (except where otherwise stated) is original research by the author and has not been submitted in full or part to this or any other institution for degree purposes.

NASHAT IBRAHIM KADHUM.

ACKNOWLEDGEMENTS

My sincere gratitude to my supervisor, Professor D.J. Evans, for his direction and guidance through the work to the completion of this thesis.

My thanks to the Ministry of Higher Education and Scientific Research in IRAQ for sponsoring my studies.

I would like to express my great regards for my parents for their moral support and understanding.

Finally, my thanks to the staff in the Computer Centre at Loughborough University for their help.

الى الجندي العراقي الباسل

To the brave Iraqui soldier

ABSTRACT

This thesis is concerned with the Numerical Solution of Partial Differential Equations.

Initially some definitions and mathematical background are given, accompanied by the basic theories of solving linear systems and other related topics. Also, an introduction to splines, particularly cubic splines and their identities are presented. The methods used to solve parabolic partial differential equations are surveyed and classified into explicit or implicit (direct and iterative) methods. We concentrate on the Alternating Direction Implicit (ADI), the Group Explicit (GE) and the Crank-Nicolson (C-N) methods.

A new method, the Splines Group Explicit Iterative Method is derived, and a theoretical analysis is given. An optimum single parameter is found for a special case. Two criteria for the acceleration parameters are considered; they are the Peaceman-Rachford and the Wachspress criteria. The method is tested for different numbers of both parameters. The method is also tested using single parameters, i.e. when used as a direct method. The numerical results and the computational complexity analysis are compared with other methods, and are shown to be competitive. The method is shown to have good stability property and achieves high accuracy in the numerical results.

Another direct explicit method is developed from cubic splines; the Splines Group Explicit Method which includes a parameter that can be chosen to give optimum results. Some analysis and the computational complexity of the method is given, with some numerical results shown

to confirm the efficiency and compatibility of the method.

Extensions to two dimensional parabolic problems are given in a further chapter.

In this thesis the Dirichlet, the Neumann and the periodic boundary conditions for linear parabolic partial differential equations are considered.

The thesis concludes with some conclusions and suggestions for further work.

CONTENTS

CHAPTER ONE

INTRODUCTION

1.1 INTRODUCTION

Many scientific and engineering problems lead to what are called, Partial Differential Equations, in which the dependent variable is expressed in terms of several independent variables.

The partial differential equations, or as abbreviated *p.d.e.*, can be classified, according to their physical meaning, into equations which are generally associated with a) equilibrium problems, and b) diffusion or propagation problems.

The first of these are termed elliptic kind. They are defined on closed regions with known boundary conditions at all points of the region's boundary, so also they are called boundary value problems. The second kind are termed either parabolic or hyperbolic type. They are defined on regions on which initial conditions are given on part of the region, and the boundary conditions are given on remaining parts. These regions are generally open from one side at least. The problems are also called initial, or initial boundary value problems [ABDULLAH, A.R.B., 1983, AMES, W.F., 1977, SMITH, D.G., 1978, YOUSIF, W.S., 1984].

1.2 CLASSIFICATION OF PARTIAL DIFFERENTIAL EQUATIONS

Some problems which lead to p.d.e.'s are of a different nature, (like; steady temperature distribution, steady voltage distribution,..., natural frequency problems in vibrations,..., propagation of pressure waves in a fluid, propagation of heat,...). This consequently shows different mathematical formulation, and thus another way of classification is due to the characteristic equation [AMES, W.F., 1977].

Let us consider the general linear equation of second order (with two independent variables, x,y),

$$a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} + e = 0 , \qquad (1.2.1)$$

where a,b,c and e are functions of $x,y,u,\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial y}$. Here we will find that there are two directions in which the integration of the p.d.e. at any point of the x-y plane, reduces to the integration of an equation involving total differentials only [SMITH, D.G., 1978].

Let the derivatives in equation (1.2.1) be denoted by,

$$\frac{\partial u}{\partial x} = p, \quad \frac{\partial u}{\partial y} = q, \quad \frac{\partial^2 u}{\partial x^2} = r, \quad \frac{\partial^2 u}{\partial x \partial y} = s \text{ and } \frac{\partial^2 u}{\partial y^2} = t .$$

Let C be a curve in the x-y plane on which the values of u,p,q,r,s and t satisfy (1.2.1). (The initial values of u,p and q are not given on C). Therefore the differentials of p and q in directions tangential to C satisfy the equations,

$$dp = \frac{\partial p}{\partial x} dx + \frac{\partial p}{\partial y} dy = rdx + sdy , \qquad (1.2.2)$$

and

$$dq = \frac{\partial q}{\partial x} dx + \frac{\partial q}{\partial y} dy = sdx + tdy , \qquad (1.2.3)$$

where,

$$ar + bs + ct + e = 0 . \qquad (1.2.4)$$

Using equation (1.2.2) and (1.2.3) to eliminate r and t from equation (1.2.4) gives,

$$\frac{a}{dx}(dp-sdy) + bs + \frac{c}{dy}(dq-sdx) + e = 0 , \qquad (1.2.5)$$

and multiplying by $-\frac{dy}{dx}$ gives,

$$s\left\{a\left(\frac{dy}{dx}\right)^2 - b\,\frac{dy}{dx} + c\right\} - \left\{a\,\frac{dp}{dx}\,\frac{dy}{dx} + c\,\frac{dq}{dx} + e\,\frac{dy}{dx}\right\} = 0 . \qquad (1.2.6)$$

Now choose the curve C so that the slope of the tangent at every point on it is a root of the equation,

$$a\left(\frac{dy}{dx}\right)^2 - b\,\frac{dy}{dx} + c = 0 . \qquad (1.2.7)$$

On such a curve the original differential equation is equivalent to setting the second part of (1.2.6) to zero, i.e.,

$$a\,\frac{dp}{dx}\,\frac{dy}{dx} + c\,\frac{dq}{dx} + e\,\frac{dy}{dx} = 0 . \qquad (1.2.8)$$

This shows that at every point of the solution domain there are two directions, given by the roots of equation (1.2.7), along which there is a relationship between the total differentials dp and dq, given by equation (1.2.8).

The directions given by the root $\frac{dy}{dx}$ of equation (1.2.7) are called characteristics of the differential equation, and the p.d.e. is said to be parabolic, hyperbolic or elliptic according to whether the equation has one, two or no real roots. This can be judged by the value of the discriminant b^2-4ac, i.e.,

$$\text{if } b^2-4ac \begin{cases} > 0 \\ = 0 \\ < 0 \end{cases} \text{ then } \begin{matrix} \text{hyperbolic} \\ \text{parabolic} \\ \text{elliptic} \end{matrix}$$

[GERALD, C.F., 1978].

1.3 BOUNDARY CONDITIONS

The importance of boundary conditions for p.d.e.'s can be seen quite easily, by noticing that there is a number of equations which have the same form, but they have different boundary conditions, which characterises them to be different problems.

The main types of boundary conditions are:

1. The Dirichlet Problem, in which the solution u is known on the boundary of the region, for example,

$$u|_s = f ,$$

where s is the boundary. The physical meaning of this is that the temperature on the boundary is given.

2. The Neumann Problem, in which the solution u is presented by its normal derivative on the boundary, and is written as,

$$\frac{\partial u}{\partial \nu}\Big|_s = f .$$

This means that the heat flow through the boundary is given.

3. The Mixed or Robins' Problem, in which the solution is presented combined with its derivative on s, and written as,

$$\left(\frac{\partial u}{\partial \nu} + \alpha u\right)\Big|_s = f ,$$

where α is the coefficient of thermal conductivity. The particular case $\alpha=0$ gives the Neumann problem.

4. The Periodic Problem, in which the solution satisfies the periodicity condition, this can be written as,

$$u(x) = u(x+\ell), \text{ for any } x,$$

where ℓ is the period.

1.4 AN INTRODUCTION TO SPLINES

Definition 1.4.1

Let $x_0,x_1,\ldots,x_k$ be a strictly increasing set of real numbers. Then a spline S of degree n with knots at $x_0 < x_1 < \ldots < x_k$ is a function possessing the following two properties:

1. In each of the k+2 intervals, $(-\infty,x_0),[x_0,x_1),\ldots,[x_{k-1},x_k)$, $[x_k,\infty)$, S is a polynomial of degree n or less.
2. S and its derivatives of order 1,2,...,n-1, are continuous in $(-\infty,\infty)$, i.e. $S \in C^{n-1}(-\infty,\infty)$.

The spline is a different polynomial in each of the k+2 intervals, and the continuity constraint $C^{n-1}(-\infty,\infty)$ imposes maximal continuity on this piecewise defined function, without it degenerating into the same polynomial everywhere.

Definition 1.4.2

The spline S is said to interpolate the values $u_0,u_1,\ldots,u_k$ at the points $x_0,x_1,\ldots,x_k$ if,

$$S(x_i) = u_i \ , \ i=0,1,\ldots,k.$$

If

$$u_i = u(x_i) \ ,$$

where u is a given function, then S is said to interpolate u at the points $x_0,x_1,\ldots,x_k$.

S is said to be a cubic spline on [a,b] with knots x_i's, i=0,1, ...,k, where,

$$a = x_0 < x_1 < \ldots < x_k = b,$$

if $S \in C^2[a,b]$ and in each of the intervals $[x_{i-1},x_i]$, i=1,2,...,k, S is a cubic polynomial.

In order to construct a cubic S interpolating the values $u_i=u(x_i)$, $i=0,1,\ldots,k$ such that,

$$S(x_i) = u_i \ , \ i=0,1,\ldots,k, \tag{1.4.1}$$

we use the second derivatives of S, namely $M_i = S^{(2)}(x_i)$, $i=0,1,\ldots,k$.

In the interval $[x_{i-1},x_i]$, S is a polynomial of degree three and hence $S^{(2)}$ is a linear polynomial. Therefore, if the values $M_i = S^{(2)}(x_i)$, $i=0,1,\ldots,k$, are known S can be obtained in $[x_{i-1},x_i]$ by integrating,

$$S^{(2)}(x) = \frac{1}{h_i} [M_{i-1}(x_i-x) + M_i(x-x_{i-1})] \ ,$$

twice, where,

$$h_i = x_i-x_{i-1}, \ i=1,\ldots,k$$

$$S^{(1)}(x) = -M_{i-1} \frac{(x_i-x)^2}{2h_i} + M_i \frac{(x-x_{i-1})^2}{2h_i} + A \tag{1.4.2}$$

$$S(x) = M_{i-1} \frac{(x_i-x)^3}{6h_i} + M_i \frac{(x-x_{i-1})^3}{6h_i} + Ax + B \ . \tag{1.4.3}$$

Evaluating equation (1.4.3) at the knots x_{i-1} and x_i gives,

$$S(x_{i-1}) = \frac{h_i^2}{6} M_{i-1} + Ax_{i-1} + B = u_{i-1} \ , \tag{1.4.4}$$

$$S(x_i) = \frac{h_i^2}{6} M_i + Ax_i + B = u_i \ . \tag{1.4.5}$$

Subtracting (1.4.4) from (1.4.5) gives,

$$u_i-u_{i-1} = \frac{h_i^2}{6} [M_i-M_{i-1}] + A(x_i-x_{i-1}) \ ,$$

thus,

$$A = \frac{1}{h_i} (u_i-u_{i-1}) - \frac{h_i}{6} [M_i-M_{i-1}] \ , \tag{1.4.6}$$

and,

$$S^{(1)} = \frac{1}{2h_i}\left\{M_i(x-x_{i-1})^2 - M_{i-1}(x_i-x)^2\right\} + \frac{1}{h_i}(u_i-u_{i-1}) - \frac{h_i}{6}(M_i-M_{i-1}) \tag{1.4.7}$$

To evaluate B let,

$$Ax + B = C(x-x_{i-1}) + D(x_i-x) \quad ,$$

which is a straight line equation. At x_{i-1} and x_i equation (1.4.3) gives,

$$S_i = u_i = M_i \frac{h_i^2}{6} + Ch_i \quad ,$$

$$C = \frac{u_i}{h_i} - M_i \frac{h_i}{6} \quad ,$$

and,

$$S_{i-1} = u_{i-1} = M_{i-1} \frac{h_i^2}{6} + Dh_i \quad ,$$

$$D = \frac{u_{i-1}}{h_i} - M_{i-1} \frac{h_i}{6} \quad ,$$

thus,

$$Ax+B = (x-x_{i-1})\left\{ \frac{u_i}{h_i} - M_i \frac{h_i}{6}\right\} + (x_i-x)\left\{ \frac{u_{i-1}}{h_i} - M_{i-1} \frac{h_i}{6}\right\} .$$

Substituting A as in (1.4.6) we get,

$$B = (x-x_{i-1})\left\{ \frac{u_i}{h_i} - M_i \frac{h_i}{6} \right\}+ (x_i-x)\left\{\frac{u_{i-1}}{h_i} - M_{i-1} \frac{h_i}{6}\right\} - Ax \cdot$$

Substituting this in (1.4.3) gives,

$$S(x) = \frac{(x_i-x)^3}{6h_i} M_{i-1} + \frac{(x-x_{i-1})^3}{6h_i} M_i + \frac{(x-x_i)}{h_i} \left\{u_i-M_i \frac{h_i^2}{6}\right\}+ \frac{(x_i-x)}{h_i} \left\{u_{i-1}-M_{i-1}- \frac{h_i^2}{6}\right\} \cdot (1.4.8)$$

An important relation is derived from equation (1.4.7) where we have,

$$S^{(1)}(x_{i-}) = \frac{h_i}{6} M_{i-1} + \frac{h_i}{3} M_i + \frac{1}{h_i}(u_i-u_{i-1}), \ i=1,\ldots,k, \quad (1.4.9)$$

$$S^{(1)}(x_{i+}) = - \frac{h_{i+1}}{3} M_i - \frac{h_{i+1}}{6} M_{i+1} + \frac{1}{h_{i+1}} (u_{i+1}-u_i), \ i=0,\ldots,k-1, \quad (1.4.10)$$

and from the continuity condition of the cubic splines we know that,

$$S_{i-}^{(1)} = S_{i+}^{(1)} \quad ,$$

thus (1.4.9) and (1.4.10) give,

$$\frac{h_i}{6} M_{i-1} + (\frac{h_i}{3} + \frac{h_{i+1}}{3}) M_i + \frac{h_{i+1}}{6} M_{i+1} = \frac{u_{i-1}}{h_i} - (\frac{1}{h_i} + \frac{1}{h_{i+1}}) u_i + \frac{u_{i+1}}{h_{i+1}} , \qquad (1.4.11)$$

for equally spaced knots, i.e. when $h=h_i$, $i=1,\ldots,k$, (1.4.11) will be,

$$M_{i-1} + 4M_i + M_{i+1} = \frac{6}{h^2} \{u_{i-1} - 2u_i + u_{i+1}\}, \quad i=1,\ldots,k-1. \qquad (1.4.12)$$

Another way of representing S is in terms of its first derivatives $m_i = S^{(1)}(x_i)$, $i=0,1,\ldots,k$. By using the continuity condition of the second derivatives at the knots leads to another important relation, for equally spaced knots,

$$m_{i-1} + 4m_i + m_{i+1} = \frac{3}{h}\{u_{i+1} - u_{i-1}\}, \quad i=1,2,\ldots,k-1, \qquad (1.4.13)$$

[AHLBERG, J.H., NILSON, E.N. and WALSH, J.L., 1967].

The main cubic spline relationships are given here,

$$m_{i+1} - m_i = \frac{h}{2} (M_i + M_{i+1}) \quad , \qquad (1.4.14)$$

$$m_i = \frac{h}{6} (2M_i + M_{i-1}) + \frac{u_i - u_{i-1}}{h} \quad , \qquad (1.4.15)$$

or

$$m_i = -\frac{h}{6} (2M_i + M_{i+1}) + \frac{u_{i+1} - u_i}{h} \quad , \qquad (1.4.16)$$

and,

$$M_i = \frac{2}{h}(m_{i-1} + 2m_i) - 6 \frac{u_i - u_{i-1}}{h^2} \quad , \qquad (1.4.17)$$

or

$$M_i = -\frac{2}{h} (2m_i + m_{i+1}) + 6 \frac{u_{i+1} - u_i}{h^2} \quad . \qquad (1.4.18)$$

CHAPTER TWO

THEORETICAL BACKGROUND

2.1 MATRIX ALGEBRA

Definition 2.1.1

A vector is a set of n numbers which are written in the form,

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} .$$

Definition 2.1.2

A matrix is a rectangular array of numbers. The size of a matrix is described by specifying the number of rows and columns, $(m \times n)$.

The numbers a_{ij} of a matrix A are called *elements* of A,

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \\ \vdots & & & \\ a_{m1} & \cdots & & a_{mn} \end{bmatrix}$$

A matrix A is said to be *singular* if $|A|=0$, where $|A|$ is the *determinant* of A, i.e. there is no matrix A^{-1} so that $A^{-1}A=I$, where A^{-1} is the *inverse* of A.

If $a_{ij}=a_{ji}$ then A is *symmetric*, i.e. $A^T=A$, where A^T is the *transpose* of A.

A matrix is *diagonal* when $a_{i,j}=0$ for $i \neq j$, and $a_{ii} \neq 0$,

it is *tridiagonal* if $a_{i,j}=0$ for $|i-j|>1$,

and *diagonally dominant* if $|a_{ii}| \geqslant \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$ for all i.

A is called *block diagonal* if,

$$A = \begin{bmatrix} B_1 & & & O \\ & B_2 & & \\ & & \ddots & \\ O & & & B_s \end{bmatrix},$$

and B_i, i=1,2,...,s is a square submatrix not necessarily all of the same order.

Two matrices A and B are *similar* if they have the same eigenvalues.

If AB=BA then they *commute*.

A *Hermitian* matrix $A^H=A$ is such that $a_{ij}=\overline{a_{ji}}$.

A real matrix A is said to be *positive definite* if $v^HAv>0$ for all vectors $v\neq 0$ and *positive semi-definite* if $v^HAv\geq 0$.

Conjugation of a matrix means that all elements are conjugated and is written as A*.

A symmetric matrix has:

(i) n real eigenvalues; and

(ii) n mutually orthogonal eigenvectors.

If A and B are symmetric and AB=BA, then AB is symmetric.

If (v,Av)>0 for all complex v, then A is symmetric.

If A is symmetric and positive definite, then its eigenvalues are all positive.

A^TA has positive eigenvalues.

2.2 NORMS

The norm is a pure number used to measure the magnitude of vectors and matrices comparatively.

The vector norm has to fulfil the following conditions,

1. $\|v\|>0$ for all $v\neq 0$ and $\|0\|=0$. (2.2.1a)

2. $\|cv\|=|c|\ \|v\|$ for any complex number c. (2.2.1b)

3. $\|v+u\| \leq \|v\|+\|u\|$. (2.2.1c)

The most common vector norms are the:

1. Maximum norm $\|v\|=\max_i |v_i|$. (2.2.2a)

2. Absolute norm $\|v\|= \sum_i |v_i|$. (2.2.2b)

3. Euclidean norm $\|v\|=(\sum_i |v_i|^2)^{\frac{1}{2}}$. (2.2.2c)

All these norms are special cases of the more general ℓ_p-norm, defined as,

$$\|v\| = \left\{ \sum_i |v_i|^p \right\}^{1/p} , \tag{2.2.3}$$

for $p=\infty$, $p=1$ and $p=2$, respectively.

Analogously, a matrix norm, has to fulfil the following conditions:

1. $||A||>0$ if $A\neq 0$ and $\|0\|=0$. (2.2.4a)

2. $||cA||=|c|\ ||A||$ for any complex c. (2.2.4b)

3. $||A+B|| \leq ||A||+||B||$. (2.2.4c)

4. $||AB|| \leq ||A||\ ||B||$. (2.2.4d)

The most common matrix norms are:

1. The maximum norms,

$$||A|| = ||A||_\infty = \max_i \sum_j |a_{ij}| , \tag{2.2.5a}$$

$$||A|| = ||A||_1 = \max_j \sum_i |a_{ij}| . \tag{2.2.5b}$$

2. The Euclidean norm,

$$N(A) = \left[\sum_{i,j} |a_{ij}|^2 \right]^{\frac{1}{2}} \quad . \tag{2.2.6}$$

3. The Hilbert or spectral norm,

$$||A||_2 = \sqrt{\lambda_1} \quad , \tag{2.2.7}$$

where λ_1 is the largest eigenvalue of $A^H A$.

Since matrices and vectors appear together often, a relation between their norm is needed to be introduced.

A matrix norm is said to be *compatible* with a given vector norm if,

$$\|Av\| \leq ||A|| \, \|v\| \text{, for all } v \neq 0 \quad , \tag{2.2.8}$$

and is defined as,

$$||A|| = \sup_{v \neq 0} \frac{\|Av\|}{\|v\|} \quad , \tag{2.2.9}$$

a matrix defined in this way is said to be subordinate to the vector norm under discussion.

2.3 EIGENVALUES AND EIGENVECTORS

For any matrix A there is a set of numbers λ_i and a set of vectors v_i such that,

$$(A-\lambda_i I)v_i = 0 \ , \qquad (2.3.1)$$

where λ_i, i=1,2,...,n, are called eigenvalues and v_i, i=1,2,...,n, are eigenvectors. These eigenvalues are the roots of the characteristic equation,

$$(A-\lambda I) = 0 \ . \qquad (2.3.2)$$

It is of importance to mention that of all eigenvalues the greatest modulus is called the spectral radius and is denoted as $\rho(A)$.

If the equation $Av=\lambda v$ is premultiplied by A, we get $A^2v=\lambda Av=\lambda^2 v$ and, analogously $A^m v=\lambda^m v$ and $A^{-1}v=\frac{1}{\lambda}v$.

If A is the tridiagonal matrix,

$$\begin{bmatrix} a & b & & & \\ c & a & b & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & b \\ & & & c & a \end{bmatrix}$$

where a,b and c are real and bc>0, then the eigenvalues of A are given by,

$$\lambda_s = a+2\sqrt{bc}\cos\frac{s\pi}{n+1} \ , \quad s=1,2,...,n. \qquad (2.3.4)$$

Theorem 2.3.1: (Gerschgorin)

The eigenvalues of a matrix A lie in the union of the discs,

$$|\lambda-a_{i,i}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{i,j}| \ , \text{ or } \ |\lambda-a_{i,i}| \leq \sum_{\substack{i=1 \\ i\neq j}}^{n} |a_{i,j}| \qquad (2.3.5)$$

Thus, $$\rho(A) \leq \min\left(\max_i \sum_j |a_{i,j}|;\ \max_j \sum_i |a_{i,j}|\right). \qquad (2.3.6)$$

Theorem 2.3.2

For any matrix A, $||A|| \geq \rho(A)$, where $||A||$ is any norm of A.

Proof:

For any eigenvalue λ, and associate eigenvector v of A, it follows that,

$$|\lambda| \ ||v|| = ||\lambda v|| = ||Av|| \leq ||A|| \ ||v||,$$

and so, $|\lambda| \leq ||A||$ for any eigenvalue of A.

Thus, $$\rho(A) \leq ||A|| \ . \qquad (2.3.7)$$

Different types of matrices have different types of eigenvalues and eigenvectors. In Table (2.3.1) some matrices are listed with their corresponding eigenvalues and eigenvectors.

Theorem 2.3.3

For any real matrix A, $||A||_2 = [\rho(A^T A)]^{\frac{1}{2}}$, where $||A||_2$ is the spectral norm of A.

Proof:

$A^T A$ is symmetric and positive definite. Let $\{v^{(i)}\}$, $(i=1,2,\ldots,n)$ be an orthonormal set of real eigenvectors of $A^T A$, i.e.,

$$A^T A v^{(i)} = \lambda_i v^{(i)}, \quad (0<\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n) \ ,$$

with, $$v^{(i)T}.v^{(j)} = 0 \ (i \neq j) \ ,$$

and $$v^{(i)T}.v^{(i)} = 1 \ (1 \leq i \leq n) \ .$$

Any other non-zero vector v in the space spanned by $v^{(i)}$, $(i=1,2,\ldots,n)$ can be expressed as,

$$v = \sum_{i=1}^{n} c_i v^{(i)} ,$$

and so,

$$\left(\frac{||Av||_2}{||v||_2}\right)^2 = \frac{(Av,Av)}{(v,v)} = \frac{v^T A^T A v}{v^T v}$$

$$= \frac{(\sum_i c_i v^{(i)})^T (\sum_i \lambda_i c_i v^{(i)})}{(\sum_i c_i v^{(i)})^T (\sum_i c_i v^{(i)})} = \frac{\sum_i \lambda_i |c_i|^2}{\sum_j |c_j|^2}$$

This gives the result,

$$0 \leq \lambda_1 \leq \left(\frac{||Av||_2}{||v||_2}\right)^2 \leq \lambda_n .$$

But $v=v^{(n)}$ shows that equality is possible on the right, and so,

$$||A||_2^2 = \max_{||v||_2 \neq 0} \frac{||Av||_2}{||v||_2} = \lambda_n = \rho(A^T A) .$$

Theorem 2.3.4

If A is symmetric, $||A||_2 = \rho(A)$.

Proof:

$$||A||_2^2 = \rho(A^T A) = \rho(A^2) = \rho^2(A) .$$

Thus the result follows, [MITCHELL, A.R. 1961].

TYPE OF MATRIX	DEFINITION	EIGENVALUES	EIGENVECTORS
Hermitian	$A^H=A$	λ real	v complex
Real, symmetric	$A^T=A^*=A$	λ real	v real
Anti-Hermitian	$A^H=-A$	λ purely imaginary	v complex
Anti-symmetric	$A^T=-A$		If $\lambda\neq 0$ then $v^Tv=0$
Real, anti-symmetric	$A^T=-A;A^*=A$	λ purely imaginary or $\lambda=0$	v complex (except when $\lambda=0$)
Orthogonal	$A^T=A^{-1}$		If $\lambda\neq\pm 1$, then $v^Tv=0$
Real, orthogonal	$A^T=A^{-1};A^*=A$	$\lvert\lambda\rvert=1$	v complex
Unitary	$A^H=A^{-1}$	$\lvert\lambda\rvert=1$	v complex

TABLE 2.3.1

[FROBERG, 1979, p.60].

2.4 FINITE DIFFERENCE APPROXIMATION

In many applications of mathematics, the governing equations relate discrete changes (differences) of variables, rather than rates of changes (derivatives), defined by limited processes. [HILDEBRAND, 1968,p.1]. When solving partial differential equations, these differences are actually used to simulate derivatives and the resulting equations are known as the finite difference equations.

The change in a function U(x), corresponding to an increase in the argument x by a positive amount h, is called the *forward difference* of U(x), relative to the increment h, and is denoted by $\Delta U(x)$:

$$\Delta U(x) = U(x+h) - U(x) \ . \qquad (2.4.1)$$

To show how this approximates a derivative, let the function U and its derivative be single-valued, finite and continuous functions of x, then by Taylor's theorem, we have,

$$U(x+h) = U(x)+h'U(x) + \frac{1}{2}h^2U''(x)+ \frac{1}{6}h^3U'''(x) + \ldots, \qquad (2.4.2)$$

and

$$U(x-h) = U(x)-h'U(x) + \frac{1}{2}h^2U''(x)- \frac{1}{6}h^3U'''(x) + \ldots \ . \qquad (2.4.3)$$

Adding these two equations gives,

$$U(x+h) + U(x-h) = 2U(x) + h^2U''(x) + O(h^4) \ , \qquad (2.4.4)$$

where $O(h^4)$ denotes the remaining terms which are of fourth and higher order of h.

Neglecting these terms of higher power of h, as they are small in comparison with lower powers of h, we get,

$$U''(x) \cong \frac{1}{h^2}\{U(x+h)-2U(x)+U(x-h)\} \ , \qquad (2.4.5)$$

with leading error of order h^2, i.e. $O(h^2)$.

Subtracting (2.4.3) from (2.4.2) and neglecting terms of order h^3 gives,

$$U'(x) \cong \frac{1}{2h}\{U(x+h)-U(x-h)\}, \qquad (2.4.6)$$

with an error of order h^2.

Equation (2.4.6) is called a central difference approximation. Using equations (2.4.2) and (2.4.3), considering the second and higher powers of h are negligible, leads us to the forward difference and backward difference approximations respectively:

$$U'(x) \cong \frac{1}{h}\{U(x+h)-U(x)\}\ , \qquad (2.4.7)$$

$$U'(x) \cong \frac{1}{h}\{U(x)-U(x-h)\}\ , \qquad (2.4.8)$$

with leading error of order h.

Our dealings with the parabolic partial differential equations will be through the applications of these differences.

We will consider the simplest possible form of the one dimensional parabolic p.d.e.:

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}\ , \qquad (2.4.9)$$

where U(x,t) is a function of two independent variables t and x. This equation describes different kinds of problems; one of these is heat conduction, in which U represents the temperature at a distance x from one end of a thin uniform bar after a time t.

For this initial boundary value problem, assume that the length of the bar is 1 and the temperatures at the end points of the bar are known and the temperature of the bar at time zero is U_0 (either maximum or minimum). For a numerical solution we use equation (2.4.7) to approximate the left hand side of equation (2.4.9), and equation

(2.4.5) for the right hand side. This leads to the finite difference equation:

$$\frac{U(x,t+\Delta t)-U(x,t)}{\Delta t} \cong \frac{U(x+\Delta x,t)-2U(x,t)+U(x-\Delta x,t)}{(\Delta x)^2} . \qquad (2.4.10)$$

It is obvious that after Δt time the solution U at some point x is given by,

$$U(x,t+\Delta t) \cong U(x,t) + \frac{\Delta t}{(\Delta x)^2}(U(x+\Delta x,t)-2U(x,t)+U(x-\Delta x,t)), \qquad (2.4.11)$$

i.e., in terms of U at the previous time level. This can be illustrated in Fig.(2.4.1),

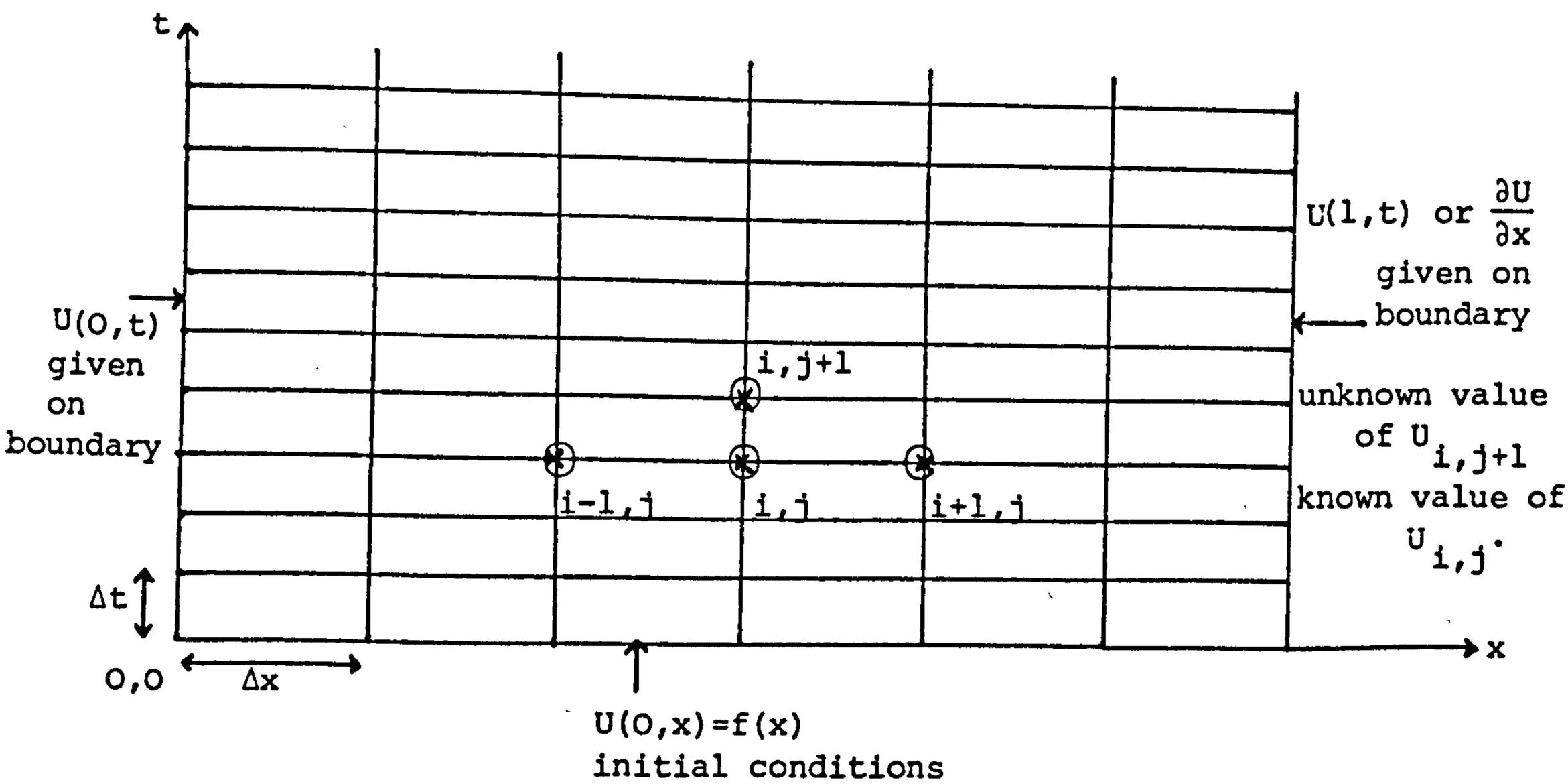


FIGURE 2.4.1

where the spatial variable is represented on the x-axis, the time variable on the y-axis and U on the mesh points created by the intersections of parallel lines to both axes. These grid points are usually chosen to be equally spaced for they are easier and faster to calculate than those which are unequally spaced.

Setting,

$$t_j = j\times k;\ \text{where } k=\Delta t,\ j=0,1,\ldots ,$$

$$x_i = i\times h;\ \text{where } h=\Delta x,\ i=0,1,\ldots ,$$

and $$r = k/h^2.$$

Equation (2.4.10) can be written as,

$$u_{i,j+1} = u_{i,j} + r(u_{i+1,j}-2u_{i,j}+u_{i-1,j}) \quad . \qquad (2.4.12)$$

This formula is known as the explicit formula. It expresses one unknown value on the (j+1)th time level directly in terms of known values on the jth time level.

There are many different schemes of finite difference, made up by using combinations of equations (2.4.5), (2.4.6), (2.4.7) and (2.4.8). Each one has its own properties concerning, for instance, stability, accuracy and complexity.

To demonstrate some other solution schemes of finite difference formula, let us consider the, so-called, weighted average formula,

$$\frac{u_{i,j+1}-u_{i,j}}{\Delta t} = \theta\,\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta x)^2} + (1-\theta)\,\frac{u_{i+1,j+1}-2u_{i,j+1}+u_{i-1,j+1}}{(\Delta x)^2} \qquad (2.4.13)$$

[CRANDALL, 1955], as an approximation to equation (2.4.9), where $0\leq\theta\leq1$. This was constructed by using equation (2.4.5) in the two time levels (j) and (j+1) for the r.h.s. of (2.4.9), and (2.4.7) for the l.h.s. of (2.4.9). If $\theta=1$, we get the classical explicit formula described above. For other values of θ we get the well known *Implicit scheme*, e.g. $\theta=0$ gives the classical implicit formula,

$$-ru_{i+1,j+1} + (1+2r)u_{i,j+1} - ru_{i-1,j+1} = u_{i,j}\ , \qquad (2.4.14)$$

See Fig.(2.4.2a),

and when $\theta=\frac{1}{2}$, (2.4.13) gives the Crank-Nicolson formula,

$$-ru_{i-1,j+1} + (2+2r)u_{i,j+1} - ru_{i+1,j+1} = ru_{i-1,j} + (2-2r)u_{i,j} + ru_{i+1,j}, \quad (2.4.15)$$

See Fig.(2.4.2b).

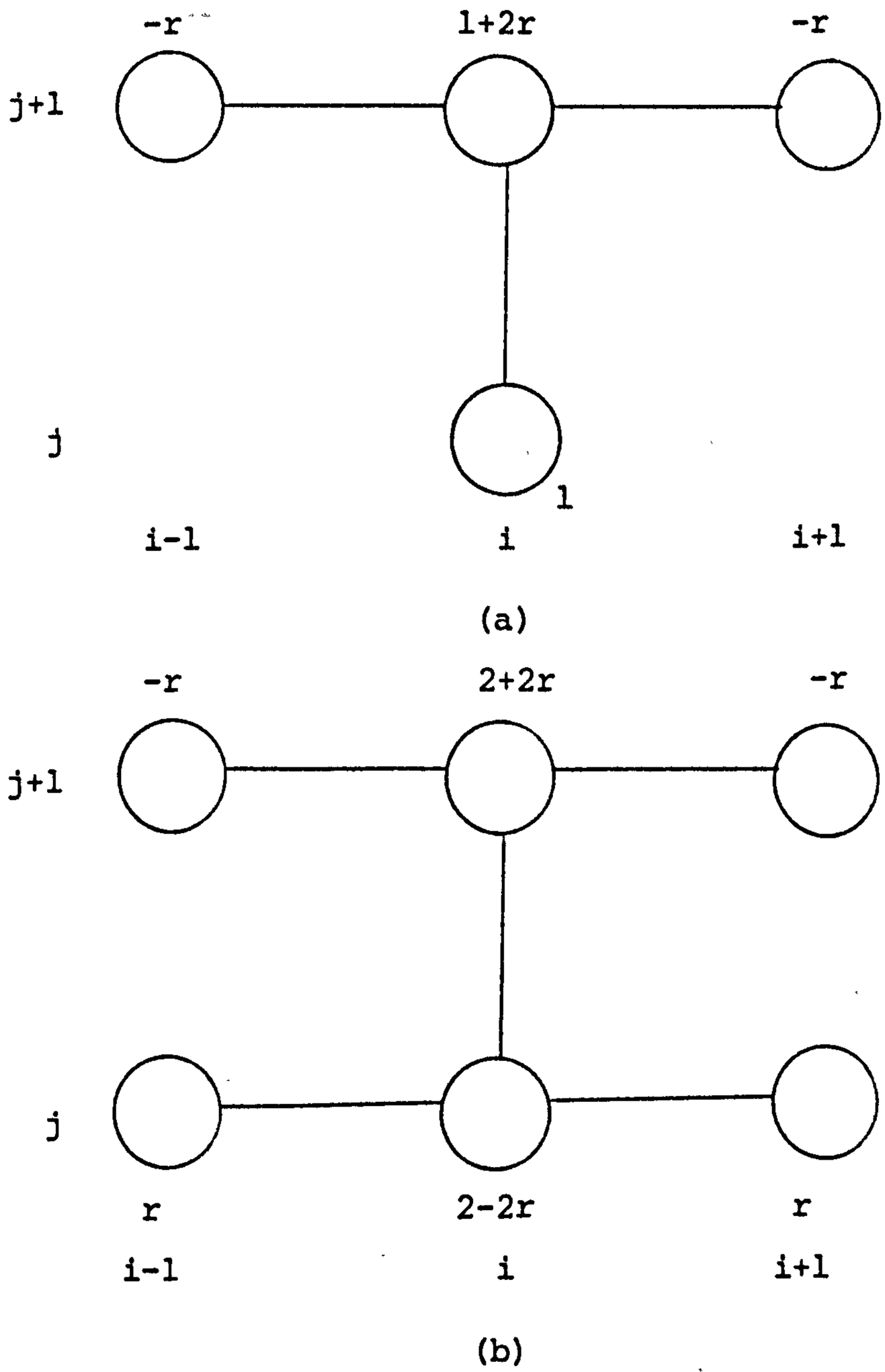


FIGURE 2.4.2: Molecular representation of:

a. Classical implicit formula,

b. Crank-Nicolson formula.

Applying either of the two equations (2.4.14) and (2.4.15) on every non-boundary points of the jth time level will result in the system of linear equations,

$$Au_{j+1} = Bu_j + c_j, \quad j=0,1,\ldots \quad . \tag{2.4.16}$$

For the classical implicit formula (2.4.14),

$$A = \begin{bmatrix} 1+2r & -r & & & \\ -r & 1+2r & -r & & O \\ & \ddots & \ddots & \ddots & \\ & O & -r & 1+2r & -r \\ & & & -r & 1+2r \end{bmatrix} \tag{2.4.17}$$

and B is the identity matrix. While for the Crank-Nicolson formula (2.4.15),

$$A = \begin{bmatrix} 2+2r & -r & & & \\ -r & 2+2r & -r & & O \\ & \ddots & \ddots & \ddots & \\ & O & -r & 2+2r & -r \\ & & & -r & 2+2r \end{bmatrix} \tag{2.4.18}$$

and,

$$B = \begin{bmatrix} 2-2r & r & & & \\ r & 2-2r & r & & O \\ & \ddots & \ddots & \ddots & \\ & O & r & 2-2r & r \\ & & & r & 2-2r \end{bmatrix} \tag{2.4.19}$$

In both cases c represents the vector associated with the boundary conditions.

2.5 ALGORITHMS FOR SOLVING SPECIAL SYSTEMS OF EQUATIONS

From the implicit schemes, systems of equations are produced, where more than one unknown value is expressed in terms of known values.

These systems have different forms and properties, depending on the difference formula used and the boundary conditions.

Here are some of the common matrix systems that appear by using finite difference methods, and some algorithms to solve them.

Our system of equations will, in general, be:

$$Au = f \ , \tag{2.5.1}$$

where A is the n×n matrix of coefficients, u is the vector of the unknown values to be found and f is the vector of the known values with their coefficients, including the boundary conditions.

The matrix A can be tridiagonal,

$$A = \begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix} \tag{2.5.2}$$

periodic,

$$A = \begin{bmatrix} b_1 & c_1 & & & a_1 \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ c_n & & & a_n & b_n \end{bmatrix} \tag{2.5.3}$$

quin-diagonal,

$$A = \begin{bmatrix} c_1 & d_1 & e_1 & & & & & & \\ b_2 & c_2 & d_2 & e_2 & & & & & \\ a_3 & b_3 & c_3 & d_3 & e_3 & & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & a_{n-2} & b_{n-2} & c_{n-2} & d_{n-2} & e_{n-2} & & \\ & & & a_{n-1} & b_{n-1} & c_{n-1} & d_{n-1} & & \\ & & & & a_n & b_n & c_n & & \end{bmatrix}, \quad (2.5.4)$$

or they may be symmetric, i.e. $a_{i+1} = c_i$, in equations (2.5.2)

or $a_{i+1} = e_i$ and $b_{i+1} = d_i$ in equation (2.5.4).

The theoretical solution of equation (2.5.1) is,

$$u = A^{-1} f \ , \quad (2.5.5)$$

as long as there exists an inverse to A.

But following that scheme broadly, i.e., finding the inverse of a matrix, needs many arithmetical operations, especially for large order matrices, where the number of operations increases rapidly with the order of the matrix.

Direct and iterative methods were developed to solve equation (2.5.1). From the direct method we have the Gauss elimination method, and other methods which are different versions of Gauss's method like triangularization (L-U Decomposition) and Crout's method. These methods are for general matrices, but can be modified to solve tridiagonal systems by saving the extra arithmetical work needed in the general case.

For the tridiagonal matrix (2.5.2) this algorithm was developed in algorithmic form to solve the system (2.5.1). Let

$$w_1 = \frac{c_1}{b_1} \quad ; \quad w_i = \frac{c_i}{b_i - a_i w_{i-1}} \quad , \tag{2.5.6}$$

and

$$g_1 = \frac{f_1}{b_1} \quad , \quad g_i = \frac{f_i - a_i g_{i-1}}{b_i - a_i w_{i-1}} \ , \ i=2,3,\ldots,n. \tag{2.5.7}$$

The components u_i of the solution vector u are then found by backward substitution,

$$u_n = g_n \ ; \quad u_i = g_i - w_i u_{i+1} \ , \ i=n-1,n-2,\ldots,1. \tag{2.5.8}$$

An iterative method for solving systems of equations is one in which a first approximation to the solution is used to calculate a second approximation. This is repeated many times until convergence is achieved. From these methods we name the Jacobi, Gauss-Seidel and Successive Over-relaxation methods.

Let $u_i^{(1)}, u_i^{(2)}, \ldots$ be the first, second,..., approximations to u_i, then for a general case, assuming that A is a full $n \times n$ matrix, the Jacobi scheme, after m iterations, is written as,

$$u_i^{(m+1)} = \frac{1}{a_{ii}} \left\{ f_i - \sum_{j=1}^{i-1} a_{ij} u_j^{(m)} - \sum_{j=i+1}^{n} a_{ij} u_j^{(m)} \right\},$$

$$i=1,2,\ldots,n. \tag{2.5.9}$$

This scheme is applicable for a tridiagonal system with less amount of work, but has slow convergence.

In using both direct and iterative methods different sources of inaccuracy arise. One of them is the round-off error. It may happen that it is necessary during the computation to cope with the disappearance of significant digits. This may cause a lower

order of accuracy in the results. In certain cases, the coefficients are such that the results are particularly sensitive to round-off; such systems are called ill-conditioned.

In the design of iterative methods, the iteration matrix is formed by splitting the l.h.s. coefficient matrix and transfer one part to the r.h.s. of the equation.

The main idea of this process is to get an easily solvable matrix on the l.h.s., a triangular, a diagonal or other easy matrix.

The new r.h.s. matrix will be associated with the same vector of the unknowns that is on the l.h.s. Therefore in order to make the system solvable, i.e. with only l.h.s. vector of unknowns, numerical values will have to be given to the new unknowns on the r.h.s. These values are either the initial conditions if available or any convenient guess near to the expected solution.

As a logical consequence, the nearer the guess is to the solution, the lower number of iterations are required. It is obvious that an initial guess solution is given only once, at the beginning of the iterative process.

We shall remember that, if the diagonal elements are to be split, then not more than half of each element to be transferred to the other side. This is to keep the method convergent.

For example, the matrix A in equation (2.5.1) can be split into,

$$A = I-G \ , \qquad (2.5.10)$$

where I is the identity matrix. Equation (2.5.1) will be,

$$Iu = Gu + f \ . \qquad (2.5.11)$$

As described above, in order to solve this system we need to

have the r.h.s. vector u known, so the general form of (2.5.11) will be,

$$u^{(n+1)} = Gu^{(n)} + f \ , \ n=0,1,\ldots \ . \qquad (2.5.12)$$

For n=0 $u^{(0)}$ will be the initial guess values.

Let the error in the nth approximation to the exact solution be,

$$e^{(n)} = u-u^{(n)} \ , \qquad (2.5.13)$$

so it follows by the subtraction of equation (2.5.12) from equation (2.5.11) that,

$$e^{(n+1)} = Ge^{(n)} \ .$$

Therefore,

$$e^{(n)} = Ge^{(n-1)} = G^2e^{(n-2)} = \ldots = G^ne^{(0)} \ . \qquad (2.5.14)$$

The sequence of iterative values $u^{(1)},u^{(2)},\ldots,u^{(n)},\ldots$ will converge to u as n tends to infinity if,

$$\lim_{n\to\infty} e^{(n)} = 0 \ . \qquad (2.5.15)$$

Since $u^{(0)}$ and hence $e^{(0)}$ is arbitrary it follows that the iteration will converge if and only if,

$$\lim_{n\to\infty} G^n = 0 \ . \qquad (2.5.16)$$

Let us assume now that the matrix G is of order m, and has m linearly independent eigenvectors v_s, s=1,2,...,m. These vectors can be used as basis for an m dimensional vector space, so that the error vector $e^{(0)}$ can be expressed uniquely as a linear combination of them, namely,

$$e^{(0)} = \sum_{s=1}^{m} c_s v_s \ , \qquad (2.5.17)$$

where c_s, s=1,2,...,m are scalars. Hence,

$$e^{(1)} = Ge^{(0)} = \sum_{s=1}^{m} c_s Gv_s \quad . \tag{2.5.18}$$

But $Gv_s=\lambda_s v_s$, where λ_s, s=1,2,...,m are the eigenvalues of G.

Hence,

$$e^{(1)} = \sum_{s=1}^{m} c_s \lambda_s v_s \quad . \tag{2.5.19}$$

Similarly,

$$e^{(n)} = \sum_{s=1}^{m} c_s \lambda_s^n v_s \quad . \tag{2.5.20}$$

Therefore $e^{(n)}$ will tend to the null vector as n tends to infinity if and only if $|\lambda_s|<1$ for all s. In other words, the iteration will converge for any arbitrary $u^{(0)}$ if and only if the spectral radius $\rho(G)<1$.

This results in that a sufficient condition for convergence is $||G||<1$.

Whereas we have $Gv_s=\lambda_s v_s$. Hence,

$$||Gv_s|| = ||\lambda_s v_s|| = |\lambda_s| \ ||v_s|| \ .$$

But for any matrix norm that is compatible with a vector norm $||v_s||$

$$||Gv_s|| \leq ||G|| \ ||v_s|| \quad .$$

Therefore,

$$|\lambda_s| \ ||v_s|| \leq ||G|| \ ||v_s|| \quad ,$$

so,

$$|\lambda_s| \leq ||G|| \quad , \ s=1,2,...,m.$$

This means that $||G||<1$ will be sufficient condition for convergence but not necessary because the norm of G can exceed one even when $\rho(G)<1$.

2.6 CONVERGENCE, STABILITY AND CONSISTENCY

From the finite difference formulae obtained there are some certain tests that they must pass. These tests are:

2.6.1 Convergence

The difference between the theoretical solutions of the partial differential equations U, and the difference equation u, at a grid point (i,j) is called the *discretization error*.

If this error converges to zero as h and k tend to zero, then the difference scheme is said to be convergent.

Let us consider, as an example, the classical explicit formula (2.4.12). In order to test its convergence we will need to keep its leading error terms to test them. Then equation (2.4.12) will be,

$$u_{i,j+1} = u_{i,j} + r(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + O(k^2 + kh^2). \qquad (2.6.1)$$

Let,

$$z_{i,j} = U_{i,j} - u_{i,j}\ , \qquad (2.6.2)$$

represent the discretization error at a point (i,j).

From equation (2.6.1) this satisfies the equation,

$$z_{i,j+1} = z_{i,j} + r(z_{i+1,j} - 2z_{i,j} + z_{i-1,j}) + O(k^2 + kh^2)\ . \qquad (2.6.3)$$

If $\frac{\partial^2 U}{\partial t^2}$ and $\frac{\partial^4 U}{\partial x^4}$ in the leading part of the local truncation error remains bounded, we find by taking the modulus of (2.6.3) that,

$$|z_{i,j+1}| \leq |z_{i,j}| + r(|z_{i+1,j}| - 2|z_{i,j}| + |z_{i-1,j}|) + A(k^2 + kh^2). \qquad (2.6.4)$$

Let,

$$|z_j| = \max_i |z_{i,j}|\ ,$$

equation (2.6.4) gives,

$$|z_{j+1}| \leq |z_j| + A(k^2+kh^2) \quad . \tag{2.6.5}$$

Since the initial data are the same for the difference and the differential equation, we have,

$$|z_o| = 0 \ ,$$

therefore,

$$|z_1| \leq |z_o| + A(k^2+kh^2) = A\ (k^2+kh^2)$$

$$|z_2| \leq |z_1| + A\,(k^2+kh^2) \leq |z_o| + 2A(k^2+kh^2) = 2A(k^2+kh^2)$$

$$\vdots$$

$$|z_m| \leq mA(k^2+kh^2) \ .$$

As time (denoted T) equals m×k, we get,

$$|z_m| \leq (m\times k)A(k+h^2) \ ,$$

$$\lim_{\substack{h\to 0\\ k\to 0}} |z_m| \leq \lim_{\substack{h\to 0\\ k\to 0}} TA(k+h^2) = 0 \ . \tag{2.6.6}$$

Convergence is now satisfied.

The discretization error can usually be diminished by decreasing Δx and Δt, but this leads to an increase in the number of equations to be solved, and this is limited subject to the cost of computations and computer storage capacity, etc.

2.6.2 Stability

As we have seen from the above topic that beside the discretization, there is another source of error due to the fact that in solving the finite difference equation each calculation is carried

out to a finite number of decimal places or significant figures. This is called the rounding error.

It is the difference between the theoretical solution and the numerical solution of the finite difference equation.

It would be useful if one could control and make this error as small as possible. So the aim is to find the conditions under which this error tends to zero as the calculations proceed.

There are two relatively simple methods which are commonly used for examining the stability of a finite difference scheme.

The Von Neumann Method

This method expresses an initial line of errors in terms of a finite Fourier series, and considers the growth of a function that reduces to this series for t=O [SMITH, G.D., 1978, p.92]. The Fourier series can be formulated in terms of complex exponentials, i.e. $\sum A_n e^{\hat{i}n\pi x/\ell}$, where $\hat{i}=\sqrt{-1}$ and ℓ is the interval which the function is defined on. Let us denote the error when t=O at the grid points by,

$$E(x_i) = \sum_{n=O}^{N} A_n e^{\hat{i}\beta_n x_i} \quad , i=O,\ldots,N \ , \tag{2.6.7}$$

where $\beta_n=n\pi/\ell$. The N+1 equations in (2.6.7) are sufficient to determine the unknowns $A_O, A_1, \ldots, A_N$ uniquely , Showing that an arbitrary distribution of initial errors can be expressed in this complex exponential form.

To investigate the propagation of this error as t increases, it is necessary to find a solution of the finite difference equation which reduces to $e^{\hat{i}\beta x}$ when t=O. Let,

$$E_{i,j} = e^{\hat{i}\beta x} e^{\alpha t} = e^{\hat{i}\beta x} e^{\alpha mk} = e^{\hat{i}\beta x}\xi^m \ , \tag{2.6.8}$$

where $\xi=e^{\alpha k}$, α is, in general, a complex constant and m is the number of time steps. This will reduce to $e^{\hat{i}\beta x}$ when m=0. Therefore, the error will not increase as t increases if,

$$|\xi| \leq 1 \quad . \tag{2.6.9}$$

This method applies to linear difference equations with constant coefficients, and strictly speaking only to initial value problems with periodic initial data, since it neglects the boundary conditions.

The Matrix Method

Although this method is more difficult than the first, since it involves matrix algebra, it is still more convenient when solving an initial boundary value problem. It includes the boundary conditions automatically, where the first one ignores them.

Consider equation (2.4.16),

$$Au_{j+1} = Bu_j + c \quad , \tag{2.6.10}$$

where B is the coefficient matrix of the right hand side, u_j is the vector of functions at time level j and c is a vector associated with the boundary conditions. For the linear parabolic equation with constant coefficients, A and B are constant matrices.

Equation (2.6.10) can be written as,

$$u_{j+1} = A^{-1}Bu_j + \hat{c} \quad , \tag{2.6.11}$$

where $\hat{c}=A^{-1}c$. Due to the errors mentioned earlier, the vector u is calculated as,

$$\tilde{u}_{j+1} = A^{-1}B\tilde{u}_j + \hat{c} \quad . \tag{2.6.12}$$

Let, $$e_{j+1} = \tilde{u}_{j+1} - u_{j+1} \ , \tag{2.6.13}$$

be the error vector. Subtracting equation (2.6.11) from equation (2.6.12) gives,

$$e_{j+1} = A^{-1}Be_j \ , \tag{2.6.14}$$

which leads to,

$$e_{j+1} = (A^{-1}B)^{j+1}e_o \ , \tag{2.6.15}$$

from which it follows that,

$$||e_{j+1}|| \leq ||(A^{-1}B)^{j+1}||\ ||e_o|| \ , \tag{2.6.16}$$

where $||.||$ is a suitable norm. The necessary and sufficient condition for the stability of the calculation is,

$$||(A^{-1}B)^{j+1}|| \leq 1 \ , \tag{2.6.17}$$

for all time steps j.

From the matrix norm property of equation (2.2.4d) we have,

$$\begin{aligned} ||(A^{-1}B)^{j+1}|| &\leq ||(A^{-1}B)^{j}||\ ||(A^{-1}B)|| \\ &\leq ||(A^{-1}B)^{j-1}||\ ||A^{-1}B||\ ||A^{-1}B|| = ||(A^{-1}B)^{j-1}||\ ||A^{-1}B||^2 \\ &\vdots \\ &\leq ||A^{-1}B||\ ||A^{-1}B||...||A^{-1}B|| = ||A^{-1}B||^{j+1}. \end{aligned}$$

Hence, from equation (2.6.17),

$$||A^{-1}B||^{j+1} \leq 1 \ ,$$

and that only happens when,

$$||A^{-1}B|| \leq 1 \ . \tag{2.6.18}$$

By Theorem 2.3.2 we have,

$$||A^{-1}B|| \geq \rho(A^{-1}B) \ . \qquad (2.6.19)$$

Thus,

$$\rho(A^{-1}B) \leq 1 \ ,$$

is a necessary and sufficient condition for stability.

2.6.3 Consistency

It may happen sometimes that an approximating finite difference formula converges to the solution of a different differential equation as Δx and Δt tend to zero. Such a formula is said to be *inconsistent* with the partial differential equation.

The importance of this concept of consistency can be seen through Lax's theorem which states that: 'Given a properly posed linear initial value problem and a finite difference approximation to it that satisfies the consistency condition, stability is the necessary and sufficient condition for convergence'. (The 'properly posed problem' term will be discussed later in this chapter).

To define consistency let us consider the partial differential equation,

$$L(U) = 0 \ . \qquad (2.6.20)$$

We are required to find the solution in the region $R=\{0<x<1\}\times\{0<t<T\}$, given the initial conditions,

$$U(x,0) = f(x), \text{ for } t=0 \ , \qquad (2.6.21)$$

and boundary conditions,

$$\left.\begin{aligned} U(0,t) &= \phi_1(t), \text{ at } x=0, \ t>0 \\ U(1,t) &= \phi_2(t), \text{ at } x=1, \ t>0 \end{aligned}\right\} \ , \qquad (2.6.22)$$

where L is an operator. Let $L_{h,k}$ denote an approximation to L

on the grid points of the region as in Fig.(2.4.1).

Then the difference scheme $L_{h,k}(u)$ is said to be consistent with the initial-boundary value problem (2.6.20)-(2.6.22), if,

$$||L_{h,k}(u) - L(U)|| \to 0 \ , \tag{2.6.23}$$

as $h \to 0$ and $k \to 0$, for a suitable norm $||.||$.

The value of equation (2.6.23) is called the error of the approximation.

We demonstrate with the simple example of equation (2.4.11), which is an approximation to,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \ , \tag{2.6.24}$$

i.e., $L = \frac{\partial}{\partial t} - \frac{\partial^2}{\partial x^2}$,

$$||L_{h,k}(u)-L(U)|| = ||(\frac{u_{i,j+1}-u_{i,j}}{k} - \frac{u_{i+1,j}-2u_{i,j}+u_{i+1,j}}{h^2}) - (\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2})|| \tag{2.6.25}$$

From the Taylor series expansion about the point (i,j), and by substituting the leading terms, the truncated error terms on the right hand side of equation (2.6.25) give,

$$||L_{h,k}(u)-L(U)|| = ||\frac{k}{2}\frac{\partial^2 U}{\partial t^2} - \frac{h^2}{6}\frac{\partial^4 U}{\partial x^4}|| \to 0 \ , \tag{2.6.26}$$

as $k \to 0$ and $h \to 0$, provided that $\frac{\partial^2 U}{\partial t^2}$ and $\frac{\partial^4 U}{\partial x^4}$ are bounded at every point of R. This shows that the approximation (2.4.11) is consistent with the initial-boundary problem (2.6.24). It also shows that the consistency test of a certain difference approximation to a parabolic equation is,

$$\frac{\text{Principle Part of Local Truncation Error}}{k} \to 0 \ , \qquad (2.6.27)$$

as $h \to 0$ and $k \to 0$.

Now that we have seen an example of consistent formula, we now show how a formula can be inconsistent.

Let us consider the well known Du-fort Frankel formula [DU-FORT AND FRANKEL, 1953], in three time levels:

$$\frac{u_{i,j+1} - u_{i,j-1}}{2k} = \frac{u_{i+1,j} - u_{i,j+1} - u_{i,j-1} + u_{i-1,j}}{h^2} \ . \qquad (2.6.28)$$

This is a stable scheme and has local truncation error of,

$$\text{L.T.E.} = \frac{1}{6} k^3 \frac{\partial^3 U}{\partial t^3} - \frac{h^2}{12} \frac{\partial^4 U}{\partial x^4} + \frac{k^3}{h^2} \frac{\partial^2 U}{\partial t^2} + \ldots \ . \qquad (2.6.29)$$

From equation (2.6.27) we have,

$$\frac{\text{L.T.E.}}{k} \to \frac{k^2}{h^2} \cdot \frac{\partial^2 U}{\partial t^2} \text{ as } h \to 0 \text{ and } k \to 0 \ ,$$

and if $\frac{k}{h} = \alpha$ then we see that the scheme is consistent with the hyperbolic equation,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \alpha^2 \frac{\partial^2 U}{\partial t^2} \ . \qquad (2.6.30)$$

However if $k \to 0$ faster than h (e.g. when $k = O(h^2)$), then (2.6.28) is consistent with the parabolic equation (2.6.24).

2.7 PROPERLY POSED PROBLEMS

A physical problem is said to be properly (well) posed if its solution exists, is unique and depends continuously on the auxiliary data.

Existence and uniqueness are an affirmation of the principle of determinism without which experiments could not be repeated with the expectation of consistent data. The continuous dependence criteria is an expression of the stability of the solution, that is, a small change in any of the problem's auxiliary data should produce only a correspondingly small change in the solution [AMES, W.F., 1977, p.41).

It is important to know whether the problem under attack is well posed or not. As an example, consider the Laplace equation,

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 , \tag{2.7.1}$$

defined in the semi-strip $0<y$, $-\pi/2\leq x\leq\pi/2$, with the boundary conditions,

$$\begin{aligned} U(-\tfrac{\pi}{2},y) &= U(\tfrac{\pi}{2},y) = 0 \\ U(x,0) &= 0 \\ \frac{\partial U}{\partial y}(x,0) &= \phi(x) , \end{aligned} \tag{2.7.2}$$

with

$$\phi(-\tfrac{\pi}{2}) = \phi(\tfrac{\pi}{2}) = 0 .$$

If we put $\phi(x)=0$, then the solution of the problem is $U(x,y)\equiv 0$. If we put,

$$\phi(x) = e^{-\sqrt{2n+1}} \cos(2n+1)x , \tag{2.7.3}$$

then we get the unique solution,

$$U(x,y) = \frac{1}{2n+1} e^{-\sqrt{2n+1}} \cos(2n+1)x \sinh(2n+1)y . \tag{2.7.4}$$

For sufficiently large n, the function ϕ and its derivatives differ from O by an arbitrary small amount. Yet for any non-zero y and large n, the function U has the form of a cosine function of arbitrarily large amplitude. Consequently, for sufficiently large n, the function U differs by an arbitrary great amount from the zero solution [MIKHLIN, 1967, p.19].

2.8 RATE OF CONVERGENCE

Assume that the iteration matrix G has m linearly independent eigenvectors v_s corresponding to the eigenvalues λ_s and that $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_m|$. By equation (2.5.15) the error vector e^n can be expressed as,

$$e^{(n)} = \lambda_1^n \{c_1 v_1 + (\frac{\lambda_2}{\lambda_1})^n c_2 v_2 + \ldots + (\frac{\lambda_m}{\lambda_1})^n c_m v_m\} .$$

For large values of n we get,

$$e^{(n)} \cong \lambda_1^n c_1 v_1 . \qquad (2.8.1)$$

Similarly,

$$e^{(n+1)} \cong \lambda_1^{n+1} c_1 v_1 ,$$

thus,

$$e^{(n+1)} \cong \lambda_1 e^n .$$

Let $e_i^{(n)}$ denote the ith component of e^n, then we see that,

$$\frac{|e_i^{(n)}|}{|e_i^{(n+1)}|} \cong \frac{1}{|\lambda_1|} = \frac{1}{\rho(G)} , \qquad (2.8.2)$$

gives an indication of the amount by which the error is decreased by each convergent iteration. The logarithm of $1/\rho(G)$ i.e. $\log_{10}(1/\rho)$ or $-\log_{10}\rho$ shows the number of digits, say p, by which the error is reduced.

For convergence we need $0<\rho<1$ and it is easy to see that p increases as ρ decreases, and thus $(-\log_{10}\rho)$ provides a measure for the rate of convergence of the iterative methods. Therefore, the quantity $(-\log_e\rho)$ is defined as the asymptotic rate of convergence which is denoted as $R_\infty(G)$.

The average rate of convergence $R_n(G)$ after n iterations is defined as,

$$R_n(G) = -1/n \log_e ||G^n||_2 , \qquad (2.8.3)$$

where $||G^n||_2$ is the spectral norm of G^n.

It is shown [YOUNG, D.M., 1971, p.87] that a crude estimate of the number of iterations needed to reduce the error $||e^{(0)}||$ to $\alpha||e^{(n)}||$, where α is a fraction, is obtained by,

$$n \cong -\log_e \alpha/R(G) . \qquad (2.8.4)$$

CHAPTER THREE

METHODS FOR SOLVING PARABOLIC P.D.E.'S

3.1 INTRODUCTION

In this chapter we present some finite difference methods for approximating the solution of partial differential equations of parabolic type. Differential equations of this type describe diffusion processes of many kinds. In recent years much progress has been made in developing more efficient finite difference procedures and in methods for determining whether or not the numerical solutions are indeed good approximations to the solutions of the partial differential equations. Clearly, the best method should be presented for the use of the applied scientist. However it is not always easy to say which method is better. There are a number of comparison points to be considered, and it is found, by experience for any specific scheme, that an improvement in any of these points results in worsening one or more of the other points. It seems like a natural law of balance. However each scheme is used where and when it is more useful and efficient.

3.2 A GENERAL TWO TIME-LEVEL FINITE DIFFERENCE APPROXIMATION FOR ONE-SPACE DIMENSIONAL P.D.E.'s

The general form of a linear parabolic partial differential equation is,

$$\sigma(x,t)\frac{\partial U}{\partial t} = \frac{\partial}{\partial x}(a(x,t)\frac{\partial U}{\partial x}) - c(x,t)U \qquad (3.2.1)$$

defined within some prescribed region R of the (x,t) plane. Within this region the functions $\sigma(x,t)$, $a(x,t)$ are strictly positive and $c(x,t)$ is non-negative.

We concentrate our attention on the constant coefficient case, and when $\sigma(x,t)=a(x,t)=1$ and $c(x,t)=0$ to give,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}\ ,\ 0\leq x\leq 1,\ 0\leq t\leq T, \qquad (3.2.2)$$

subject to the initial-boundary conditions,

$$\begin{aligned} U(x,0) &= f(x),\ 0<x<1\ , \\ U(0,t) &= g(t),\ 0<t\leq T, \\ U(1,t) &= h(t),\ 0<t\leq T. \end{aligned} \qquad (3.2.3)$$

A general finite difference approximation of (3.2.2) is

$$\frac{u_{i,j+1}-u_{i,j}}{k} = \frac{1}{h^2}(\theta_1\delta_x u_{i+\frac{1}{2},j+1}-\theta_2\delta_x u_{i-\frac{1}{2},j+1} + \theta_1'\delta_x u_{i+\frac{1}{2},j}-\theta_2'\delta_x u_{i-\frac{1}{2},j})\ , \qquad (3.2.4)$$

under the conditions, on θ_i,θ_i', i=1,2, that,

$$0 \leq \theta_i,\theta_i' \leq 1,\ \ i=1,2, \qquad (3.2.5a)$$

$$\sum_{i=1}^{2}(\theta_i+\theta_i') = 2\ , \qquad (3.2.5b)$$

$$-\theta_1-\theta_1'+\theta_2+\theta_2' = 0\ ,\ \text{[Abdullah, 1983]} \qquad (3.2.5c)$$

Most of the difference schemes for solving (3.2.2) are obtained from

the formula (3.2.4), by substituting θ_i, θ_i', i=1,2, with different values governed by the conditions (3.2.5a)-(3.2.5c).

Now we derive some of the well known formulae:

a) If $\theta_i'=1$, $\theta_i=0$, i=1,2 we get the classical explicit formula, Fig.(3.2.1a),

$$u_{i,j+1} = r(u_{i-1,j}+u_{i+1,j})+(1-2r)u_{i,j} , \qquad (3.2.6)$$

which is stable for $r\leq\frac{1}{2}$, and has a principal truncation error of order $(\Delta t+(\Delta x)^2)$, (SMITH, 1978).

b) If $\theta_i'=0$, $\theta_i=1$, i=1,2, we get,

$$-ru_{i-1,j+1}+(1+2r)u_{i,j+1}-ru_{i+1,j+1} = u_{i,j} , \qquad (3.2.7)$$

which is known as the fully-implicit formula. This is an unconditionally stable scheme for r>0 and has a principal truncation error of order $(\Delta t+(\Delta x)^2)$. Its molecular diagram is shown in Fig.(3.2.1b).

c) If $\theta_i=\theta_i'=\frac{1}{2}$, i=1,2, then (3.2.4) yields,

$$-ru_{i-1,j+1}+(2+2r)u_{i,j+1}-ru_{i+1,j+1} = ru_{i-1,j}+(2-2r)u_{i,j}+ru_{i+1,j} \qquad (3.2.8)$$

which is the Crank-Nicolson formula Fig.(3.2.1c). This formula is unconditionally stable for all r>0 and has a principle truncation error of $O((\Delta t)^2+(\Delta x)^2)$.

d) If $\theta_i=\alpha/2$, $\theta_i'=1-\alpha/2$, i=1,2, where α is a free parameter, we get the formula due to Saul'yev [1964: 91, eq.8.16], Fig.(3.2.1d),

$$-\alpha ru_{i-1,j+1}+2(1+\alpha r)u_{i,j+1}-\alpha ru_{i+1,j+1} = (2-\alpha)ru_{i-1,j}+2(1-2r+\alpha r)u_{i,j}+(2-\alpha)ru_{i+1,j} , \qquad (3.2.9)$$

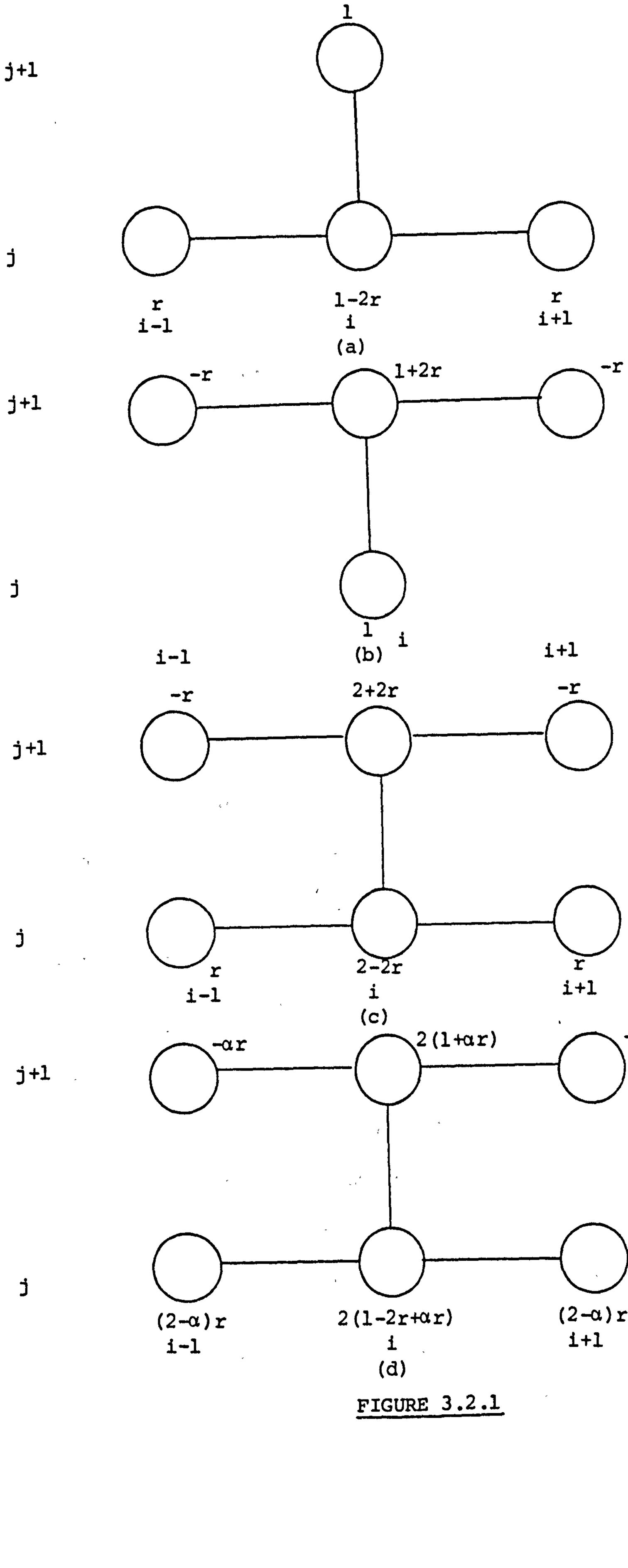

1
j+1
j
r
i-1
1-2r
i
r
i+1
(a)
-r
1+2r
-r
j+1
j
1
i
i-1
(b)
i+1
-r
2+2r
-r
j+1
j
r
i-1
2-2r
i
r
i+1
(c)
-αr
2(1+αr)
-αr
j+1
j
(2-α)r
i-1
2(1-2r+αr)
i
(2-α)r
i+1
(d)

FIGURE 3.2.1

The previous three formulae (3.2.6), (3.2.7) and (3.2.8) are actually special cases of (3.2.9). This formula is stable for r such that,

$$r \leq \frac{1}{2(1-\alpha)} \quad , \tag{3.2.10}$$

and a principal truncation error of,

$$O(\Delta t+(\Delta x)^2), \text{ for } \alpha \neq 1 \quad , \tag{3.2.11}$$
$$O((\Delta t)^2+(\Delta x)^2), \text{ for } \alpha=1.$$

e) If $\theta_i=\alpha-1/6r$, $\theta_i'=1-(\alpha-1/6r)$, i=1,2, where α, again, is a free parameter. This will give the cubic spline formula of Papamichael and Whiteman [1973],

$$(1-6\alpha r)(u_{i+1,j+1}+u_{i-1,j+1})+2(2+6r\alpha)u_{i,j+1} = (1-6r(1-\alpha))$$
$$(u_{i-1,j}+u_{i+1,j})+2(2-6r(1-\alpha))u_{i,j}. \tag{3.2.12}$$

The formula has unconditional stability when $\frac{1}{2} \leq \alpha \leq 1$ and is stable for $r \leq \frac{1}{6(1-2\alpha)}$ when $0 \leq \alpha \leq \frac{1}{2}$. Its principal truncation error is $O(\Delta t+(\Delta x)^2)$, but for $\alpha=\frac{1}{2}$, the principal truncation error order improves to $((\Delta t)^2+(\Delta x)^2+(\frac{\Delta t}{\Delta x})^2)$. The molecular diagram of this formula is shown in Fig.(3.2.2).

f) If $\theta_1=\theta_2'=1$ and $\theta_2=\theta_1'=0$, then we get the formula due to Saul'yev (1964:32, eq.3.10) as shown in Fig.(3.2.3a). This formula has a principal truncation error of order $(-\frac{\Delta t}{\Delta x}-\Delta t\Delta x)$, and it is unconditionally stable for r>0. It is written as,

$$-ru_{i+1,j+1}+(1+r)u_{i,j+1} = ru_{i-1,j}+(1-r)u_{i,j} \quad , \tag{3.2.13}$$

g) Saul'yev's alternative formula is in an opposite direction to (3.2.13). It is formed by letting $\theta_1=\theta_2'=0$ and $\theta_2=\theta_1'=1$, and

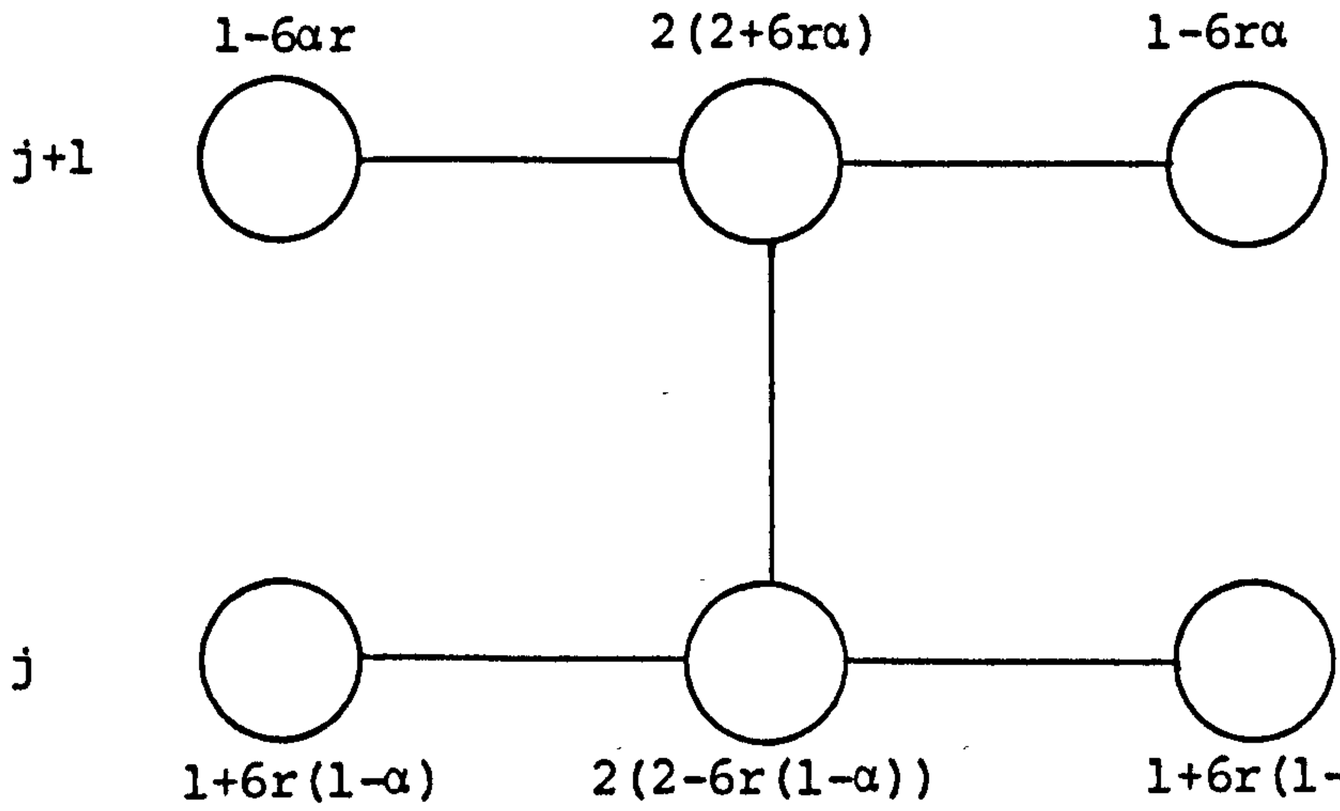


FIGURE 3.2.2

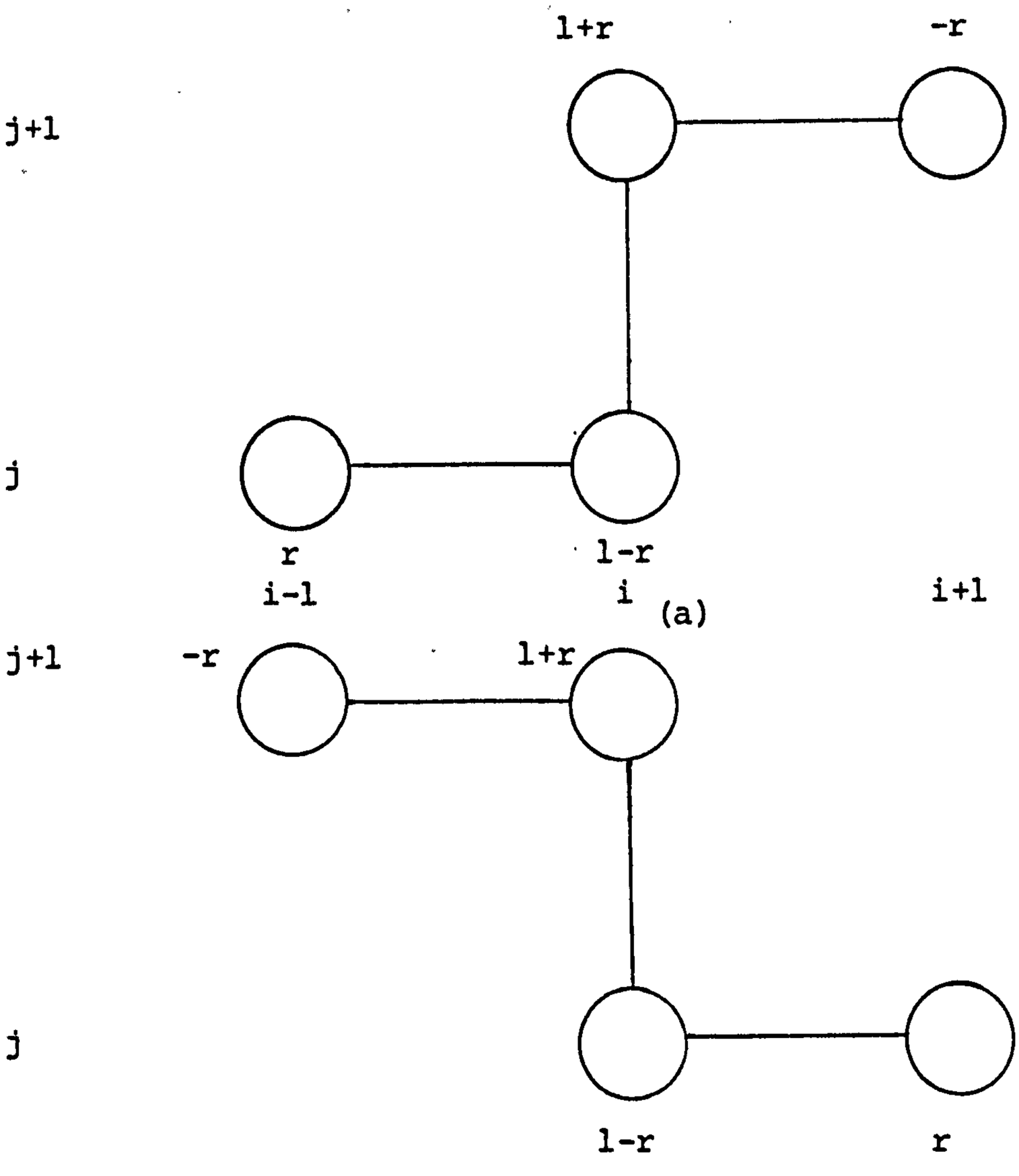


FIGURE 3.2.3

has the same properties except that its principal truncation error has an opposite sign of (3.2.13). It is written as (SAUL'YEV, 1964),

$$-ru_{i-1,j+1}+(1+r)u_{i,j+1} = ru_{i+1,j}+(1-r)u_{i,j} \quad . \qquad (3.2.14)$$

Fig.(3.2.3b) represents the molecular diagram of equation (3.2.14).

All the formulae mentioned above are of three point type. In Abdullah (1983:162) a four point formula is presented. It is a combination of equations (3.2.13) and (3.2.14), which result in a 2×2 coefficient matrix that is easy to invert. This formula is unconditionally stable for $r \leq 1$ and has a principal truncation error, for the 2 point group, of order $(\frac{\Delta t}{\Delta x} - \Delta t \Delta x)$ and $(-\frac{\Delta t}{\Delta x} + \Delta t \Delta x)$. The molecular diagram is shown in Fig.(3.2.4).

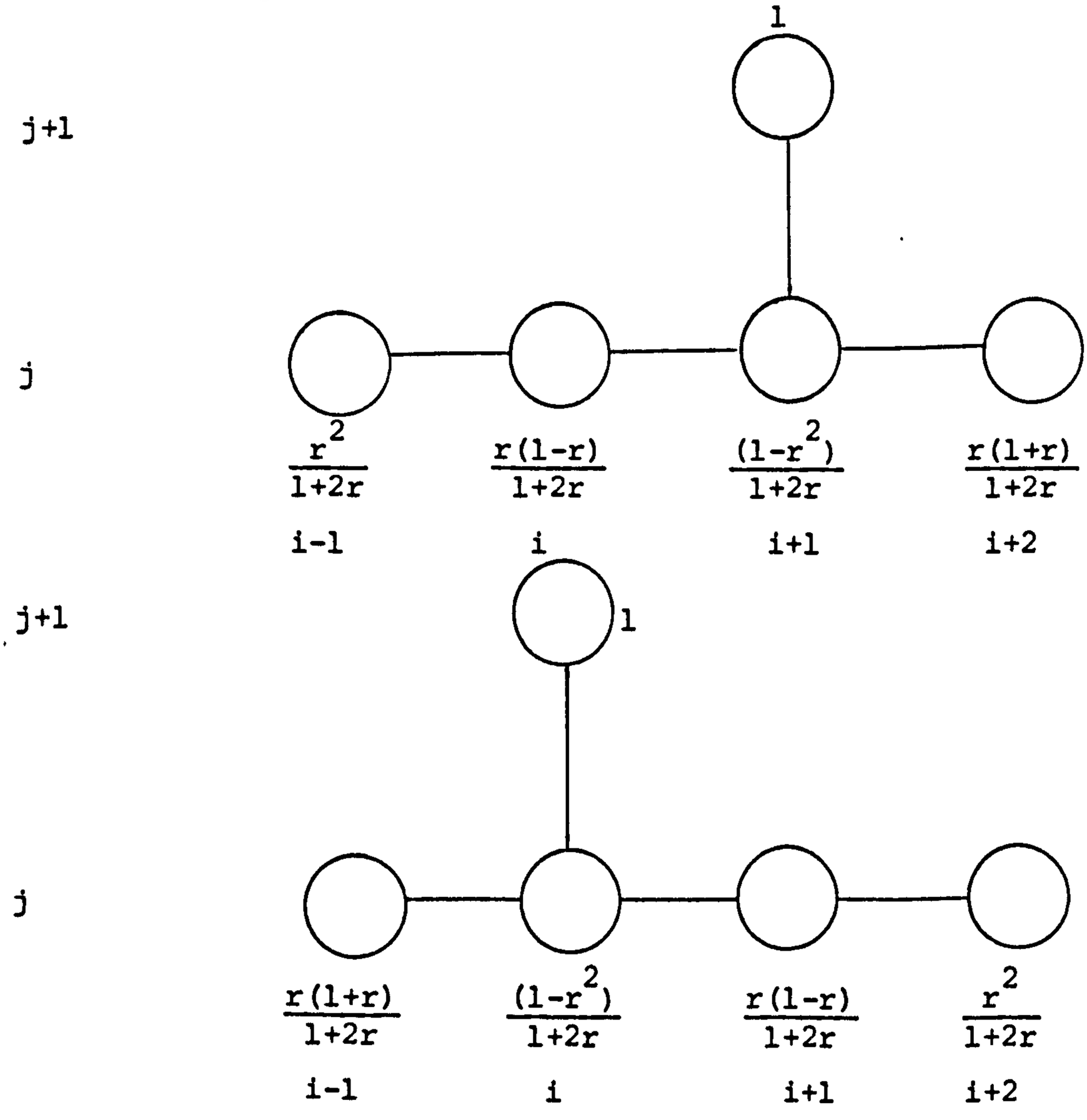


FIGURE 3.2.4

This scheme is called the Group Explicit (G.E.) method and is written as:

$$u_{i,j+1} = \frac{1}{1+2r}(r(1+r)u_{i-1,j}+(1-r^2)u_{i,j}+r(1-r)u_{i+1,j}+r^2u_{i+2,j}), \tag{3.2.15}$$

and,

$$u_{i+1,j+1} = \frac{1}{1+2r}(r^2u_{i-1,j}+r(1-r)u_{i,j}+(1-r^2)u_{i+1,j}+r(1+r)u_{i+2,j}). \tag{3.2.16}$$

3.3 TRUNCATION ERROR OF THE GENERAL APPROXIMATION

The general formula (3.2.4) can be written as,

$$-r\theta_1 u_{i+1,j+1} + [1+r(\theta_1+\theta_2)]u_{i,j+1} - r\theta_2 u_{i-1,j+1} = r\theta_1' u_{i+1,j}$$

$$+[1-r(\theta_1'+\theta_2')]u_{i,j} + r\theta_2' u_{i-1,j} \; . \qquad (3.3.1)$$

Using Taylor's series expansion about the point $(i,j+\frac{1}{2})$, the truncation error of equation (3.3.1) is,

$$rh\,\alpha\frac{\partial u}{\partial x} + k\,\frac{\partial u}{\partial t} - \frac{rh^2\beta}{2}\,\frac{\partial^2 u}{\partial x^2} + \frac{rhk\gamma}{2}\,\frac{\partial^2 U}{\partial x \partial t}$$

$$+ \frac{rh^3\alpha}{6}\,\frac{\partial^3 u}{\partial x^3} + \frac{rh^2 k\Delta}{4}\,\frac{\partial^3 u}{\partial x^2 \partial t} + \frac{rhk^2\alpha}{8}\,\frac{\partial^3 u}{\partial x \partial t^2}$$

$$+ \frac{k^3}{24}\,\frac{\partial^3 u}{\partial t^3} - \frac{rh^4\beta}{24}\,\frac{\partial^4 u}{\partial x^4} + \frac{rh^3 k\gamma}{12}\,\frac{\partial^4 u}{\partial x^3 \partial t}$$

$$- \frac{rh^2k^2\beta}{16}\,\frac{\partial^4 u}{\partial x^2 \partial t^2} + \frac{rhk^3\gamma}{48}\,\frac{\partial^4 u}{\partial x \partial t^3} + O(k^{\xi_1}, h^{\xi_2}) = O, \qquad (3.3.2)$$

where,

$$\alpha = -\theta_1+\theta_2-\theta_1'+\theta_2' \; , \qquad \beta = -\theta_1-\theta_2-\theta_1'-\theta_2' \; ,$$

$$\gamma = -\theta_1+\theta_2+\theta_1'-\theta_2' \; , \qquad \Delta = -\theta_1-\theta_2+\theta_1'+\theta_2' \; ,$$

$$\xi_1+\xi_2 = 5.$$

If the conditions (3.2.5a), (3.2.5b) and (3.2.5c) are fulfilled, then $\alpha=0$, $\beta=2$. Dividing through by k, (3.3.2) will become,

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} + \frac{k\gamma}{2h}\,\frac{\partial^2 u}{\partial x \partial t} + \frac{k\Delta}{4}\,\frac{\partial^3 u}{\partial x^2 \partial t} + \frac{k^3}{24}\,\frac{\partial^3 u}{\partial t^3}$$

$$- \frac{h^2}{12}\,\frac{\partial^4 u}{\partial x^4} + \frac{kh\gamma}{12}\,\frac{\partial^4 u}{\partial x^3 \partial t} - \frac{k^2}{8}\,\frac{\partial^4 u}{\partial x^2 \partial t^2} + \frac{k^3\gamma}{48h}\,\frac{\partial^4 u}{\partial x \partial t^3}$$

$$+ \frac{1}{k}\,O(k^{\xi_1}, h^{\xi_2}) = O \; . \qquad (3.3.3)$$

As we are dealing with the equation (3.2.2), the first two terms of (3.3.3) will vanish. Further, in order to estimate the truncation error of the formulae (3.2.6)-(3.2.9) and (3.2.12)-(3.2.14), we have to satisfy their conditions, concerning the free parameters θ_i and θ_i', i=1,2,. Now substituting θ_i and θ_i', i=1,2, in equation (3.3.3) we get,

$$T(3.2.6) = \frac{k}{2}\frac{\partial^3 u}{\partial x^2 \partial t} + \frac{k^2}{24}\frac{\partial^3 u}{\partial t^3} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4} - \frac{k^2}{8}\frac{\partial^4 u}{\partial x^2 \partial t^2} + \frac{1}{k}O(k^{\xi_1}, h^{\xi_2}) , \quad (3.3.4)$$

$$T(3.2.7) = -\frac{k}{2}\frac{\partial^3 u}{\partial x^2 \partial t} + \frac{k^2}{24}\frac{\partial^3 u}{\partial t^3} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4} + \frac{k^2}{8}\frac{\partial^4 u}{\partial x^2 \partial t^2} + \frac{1}{k}O(k^{\xi_1}, h^{\xi_2}) , \quad (3.3.5)$$

$$T(3.2.8) = \frac{k^2}{24}\frac{\partial^3 u}{\partial t^3} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4} - \frac{k^2}{8}\frac{\partial^4 u}{\partial x^2 \partial t^2} + \frac{1}{k}O(k^{\xi_1}, h^{\xi_2}) \quad (3.3.6)$$

$$T(3.2.9) = \frac{k(1-\alpha)}{2}\frac{\partial^3 u}{\partial x^2 \partial t} + \frac{k^2}{24}\frac{\partial^3 u}{\partial t^3} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4} - \frac{k^2}{8}\frac{\partial^4 u}{\partial x^2 \partial t^2} + \frac{1}{k}O(k^{\xi_1}, h^{\xi_2}) , \quad (3.3.7)$$

$$T(3.2.12) = \frac{k}{2}(1-2(\alpha-1/6r))\frac{\partial^3 u}{\partial x^2 \partial t} + \frac{k^2}{24}\frac{\partial^3 u}{\partial t^3} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4} - \frac{k^2}{8}\frac{\partial^4 u}{\partial x^2 \partial t^2} + \frac{1}{k}O(k^{\xi_1}, h^{\xi_2}) , \quad (3.3.8)$$

$$T(3.2.13) = -\frac{k}{h}\frac{\partial^2 u}{\partial x \partial t} + \frac{k^2}{24}\frac{\partial^3 u}{\partial t^3} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4} - \frac{kh}{6}\frac{\partial^4 u}{\partial x^3 \partial t} - \frac{k^2}{8}\frac{\partial^4 u}{\partial x^2 \partial t^2} - \frac{k^3}{24h}\frac{\partial^4 u}{\partial x \partial t^3} + \frac{1}{k}O(k^{\xi_1}, h^{\xi_2}) , \quad (3.3.9)$$

and,

$$T(3.2.1\ 4) = \frac{k}{h}\frac{\partial^2 u}{\partial x \partial t} + \frac{k^2}{24}\frac{\partial^3 u}{\partial t^3} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4} + \frac{kh}{6}\frac{\partial^4 u}{\partial x^3 \partial t} - \frac{k^2}{8}\frac{\partial^4 u}{\partial x^2 \partial t^2} + \frac{k^3}{24h}\frac{\partial^4 u}{\partial x \partial t^3} + \frac{1}{k}O(k^{\xi_1}, h^{\xi_2}), \quad (3.3.10)$$

respectively.

3.4 STABILITY ANALYSIS FOR THE GENERAL APPROXIMATION

The stability conditions of the general approximation formula (3.3.1) will be investigated, using the Fourier series method, where the error function at any point (i,j) is expressed as,

$$E_{i,j} = e^{\hat{i}\, i\beta h}\, e^{\alpha jk}, \text{ where } \hat{i}=\sqrt{-1},\ h=\Delta x,\ k=\Delta t,\ \alpha \text{ is complex}$$
$$= e^{\hat{i}\, i\beta h}\xi^{j},\quad \xi=e^{\alpha k}\ .$$

Substituting the error function into equation (3.3.1), gives,

$$-r\theta_1\, e^{\hat{i}\beta h(i+1)}\xi^{j+1}+[1+r(\theta_1+\theta_2)]e^{\hat{i}\beta hi}\xi^{j+1}\ -r\theta_2 e^{\hat{i}\beta h(i-1)}\xi^{j+1}$$
$$= r\theta_1'\, e^{\hat{i}\beta h(i+1)}\xi^{j}+[1-r(\theta_1'+\theta_2')]e^{\hat{i}\beta hi}\xi^{j}+r\theta_2' e^{\hat{i}\beta h(i-1)}\xi^{j} \qquad (3.4.1)$$

dividing by $e^{\hat{i}\beta hi}\xi^{j}$ gives,

$$\xi = \frac{r\theta_1' e^{i\beta h}+[1-r(\theta_1'+\theta_2')]+r\theta_2' e^{-i\beta h}}{-r\theta_1 e^{\hat{i}\beta h}+[1+r(\theta_1+\theta_2)]-r\theta_2 e^{-\hat{i}\beta h}} \qquad (3.4.2)$$

$$= \frac{1-r(1-\cos\beta h)(\theta_1'+\theta_2')+\sqrt{-1}\ r(\theta_1'-\theta_2')\sin\beta h}{1-r(\cos\beta h-1)(\theta_1+\theta_2)-\sqrt{-1}r(\theta_1-\theta_2)\sin\beta h}\ . \qquad (3.4.3)$$

For stability we need $|\xi|\leq 1$. Substituting the values of θ_i', θ_i, i=1,2, of section 3.2 into equation (3.4.3) leads us to expressions which verify the stability conditions of the schemes in equations (3.2.6)-(3.2.9) and (3.2.12)-(3.2.14) as below:

For i=1,2

a) $\theta_i'=1$, $\theta_i=0$, equation (3.4.3) becomes,

$$\left|\frac{1-2r(1-\cos\beta h)}{1}\right| \leq 1\ , \qquad (3.4.4)$$

as $1-\cos\beta h=2\sin^2\frac{\beta h}{2}$, we get,

$$|1-4r\sin^2\frac{\beta h}{2}| \leq 1\ .$$

i) $1-4r\sin^2\frac{\beta h}{2} \leq 1$,

$-4r\sin^2\frac{\beta h}{2} \leq 0$,

$-r \leq 0$, i.e. for all $r>0$, where,

$\max \sin^2\frac{\beta h}{2} = 1.$

ii) $-1 \leq 1-4r\sin^2\frac{\beta h}{2}$,

$4r\sin^2\frac{\beta h}{2} \leq 2$,

$\therefore \quad r \leq \frac{1}{2}$. (3.4.5)

As this is the lower bound of stability, the scheme will only be stable at values of r under this value.

b) $\theta_i'=0$, $\theta_i=1$, then (3.4.3) becomes,

$$\left|\frac{1}{1+2r(1-\cos\beta h)}\right| \leq 1 , \tag{3.4.6}$$

or

$$\left|\frac{1}{1+4r\sin^2\frac{\beta h}{2}}\right| \leq 1 .$$

i) $1 \leq 1+4r\sin^2\frac{\beta h}{2}$

$0 \leq 4r$

$0 \leq r$. (3.4.7)

ii) $-1-4r\sin^2\frac{\beta h}{2} \leq 1$

$-4r\sin^2\frac{\beta h}{2} \leq 2$

$-2r \leq 1$

$r \geq -\frac{1}{2}$ neglected.

c) $\theta_i'=\theta_i=\frac{1}{2}$, (3.4.3) gives,

$$\left|\frac{1-r(1-\cos\beta h)}{1+r(1-\cos\beta h)}\right| \leq 1$$

or

$$\left|\frac{1-2r\sin^2\frac{\beta h}{2}}{1+2r\sin^2\frac{\beta h}{2}}\right| \leq 1 \tag{3.4.8}$$

which is valid for all $r>0$.

d) $\theta'_i=\alpha/2$, $\theta_i=1-\alpha/2$, (3.4.3) leads to,

$$\left|\frac{1-\alpha r(1-\cos\beta h)}{1+(2-\alpha)r(1-\cos\beta h)}\right| \leq 1 ,$$

$$\left|\frac{1-2\alpha r \sin^2\frac{\beta h}{2}}{1+2r(2-\alpha)\sin^2\frac{\beta h}{2}}\right| \leq 1$$

i) $1-2\alpha r \sin^2\frac{\beta h}{2} \leq 1+2r(2-\alpha)\sin^2\frac{\beta h}{2}$

$0\leq 4r \Rightarrow 0<r$

ii) $-1-2r(2-\alpha)\sin^2\frac{\beta h}{2} \leq 1-2\alpha r\sin^2\frac{\beta h}{2}$

$-4r+2\alpha r \sin^2\frac{\beta h}{2} \leq 2-2\alpha r \sin^2 \beta h/2$

$-4r(1-\alpha)\sin^2\frac{\beta h}{2} \leq 2$

i.e.,

$$r \leq \frac{1}{2(\alpha-1)} \qquad (3.4.9)$$

e) $\theta_i=\alpha-1/6r$, $\theta'_i=1-(\alpha-1/6r)$, then (3.4.3) gives,

$$\left|\frac{1-2r(1-\alpha+1/6r)(1-\cos\beta h)}{1+2r(\alpha-1/6r)(1-\cos\beta h)}\right| \leq 1 ,$$

or

$$\left|\frac{1-2r(1-\alpha+1/6r)(2\sin^2\frac{\beta h}{2})}{1+2r(\alpha-1/6r)(2\sin^2\frac{\beta h}{2})}\right| \leq 1 ,$$

i) $1-4r(1-\alpha+1/6r)\sin^2\frac{\beta h}{2} \leq 1+4r(\alpha-1/6r)\sin^2\frac{\beta h}{2}$

$-4r\leq 0 \Rightarrow \quad r>0$

ii) $-1-4r(\alpha-1/6r)\sin^2\frac{\beta h}{2} \leq 1-4r(1-\alpha+1/6r)\sin^2\frac{\beta h}{2}$

$-4\alpha r+2/3+4r-4\alpha r+2/3 \leq 2,$

$4r-8\alpha r \leq 2-4/3 = 2/3 ,$

$r(2-4\alpha) \leq 1/3 ,$

$$r \leq \frac{1}{3(2-4\alpha)} = \frac{1}{6(1-2\alpha)} . \qquad (3.4.10)$$

For $\alpha=\frac{1}{2}$ the stability condition (3.4.10) will be

$$r \leq \infty ,$$

i.e. the cubic spline formula (3.2.12) will be unconditionally stable for all $r>0$. For this same value of α, the principal truncation error of (3.2.12) rises from $O(k,h^2)$ to $O(k^2,h^2)$. For equations (3.2.13) and (3.2.14), we substitute the values of θ_i' and θ_i in equation (3.4.3) to get, for the first equation, where $\theta_1=\theta_2'=1$ and $\theta_1'=\theta_2=0$

$$\xi = \frac{1-r(1-\cos\beta h)-\sqrt{-1}\ r\ \sin\beta h}{1+r(1-\cos\beta h)-\sqrt{-1}\ r\ \sin\beta h} \quad . \tag{3.4.11}$$

Now
$$|\xi|^2 = \frac{[(1-r)+r\cos\beta h]^2+r^2\sin^2\beta h}{[(1+r)-r\cos\beta h]^2+r^2\sin^2\beta h} \quad , \tag{3.4.12}$$

$$= \frac{(1-r)^2+2r(1-r)\cos\beta h+r^2\cos^2\beta h+r^2\sin^2\beta h}{(1+r)^2-2r(1+r)\cos\beta h+r^2\cos^2\beta h+r^2\sin^2\beta h} \quad , \tag{3.4.13}$$

$$= \frac{(1-r)^2+2r(1-r)(\cos^2\frac{\beta h}{2}-\sin^2\frac{\beta h}{2})+r^2}{(1+r)^2-2r(1+r)(\cos^2\frac{\beta h}{2}-\sin^2\frac{\beta h}{2})+r^2} \quad , \tag{3.4.14}$$

$$= \frac{(1-r)^2+2r(1-r)(1-2\sin^2\frac{\beta h}{2})+r^2}{(1+r)^2-2r(1+r)(1-2\sin^2\frac{\beta h}{2})+r^2} \quad , \tag{3.4.15}$$

$$= \frac{1-2r+r^2+2r-2r^2-4rs+4r^2s+r^2}{1+2r+r^2-2r-2r^2+4rs+4r^2s+r^2} \quad , \tag{3.4.16}$$

$$= \frac{1-4rs+4r^2s}{1+4rs+4r^2s} \leq 1, \text{ for } r\geq 0 \ . \tag{3.4.17}$$

Hence $|\xi|\leq 1$ for $r\geq 0$, where $s=\sin^2\frac{\beta h}{2}$.

For the remaining Saul'yev equation (3.2.14) where $\theta_1'=\theta_2=1$ and $\theta_1=\theta_2'=0$, equation (3.4.3) gives,

$$\xi = \frac{1-r(1-\cos\beta h)+\sqrt{-1}\ r\ \sin\beta h}{1+r(1-\cos\beta h)+\sqrt{-1}\ r\ \sin\beta h} \quad . \tag{3.4.18}$$

This ξ is the complex conjugate of ξ given by (3.4.11) and hence

$$|\xi|^2 \leq 1 \text{ for } r\geq 0.$$

Thus,

$$|\xi| \leq 1 \quad \text{for } r \geq 0 \; . \tag{3.4.19}$$

Therefore, the Saul'yev scheme is stable for all $r \geq 0$.

The stability conditions for the (G.E.) system of equations (3.2.15) and (3.2.16) is proven, by the matrix method, to be unconditionally stable for $r \leq 1$. [Abdullah, 1983; p.179].

3.5 EQUATIONS OF INCREASED ACCURACY

The principal criteria of any finite-difference scheme, for the numerical approximation of an equation are listed as follows according to their importance:

1. Stability
2. Order of error of approximation
3. Simplicity (SAUL'YEV, 1964, p.83).

The importance of stability of a numerical scheme is clearly seen through Lax's theorem [see Section 2.6.3], where it states that stability is the necessary and sufficient condition for convergence. This puts it on top priority of any numerical scheme. However unstable schemes will not be considered in this thesis as they are not convergent. Convergence itself, which depends on the order of error of approximation occurs in various forms. Slower convergence requires more computational work and is impractical while faster convergence needs less work and is desirable. This shows that the rate and order of convergence is important.

In most cases the order of error of the numerical solution is similar to that of the error of approximation which shows the importance of the order of error [SAUL'YEV, 1964, p.85].

Let LU=O be a p.d.e. and let a corresponding f.d.e. be denoted by $L_h u_{i,j}=O$. Then for a sufficiently smooth function W(x,t) satisfying the equation LW=O, the expression,

$$L_h W_{i,j} = O(h^{\alpha}) \ , \quad \alpha>O \ , \tag{3.5.1}$$

denotes that the order of error incurred by approximating the operator L by L_h at the point (ih,jk) is α for the class of functions satisfying the equation LW=O [SAUL'YEV, 1964,p.84]. In accordance

with this α is a measure of the category of the finite difference scheme, that is:

$$\left.\begin{array}{ll} \text{a)} & \alpha \leq 0\text{: Divergent schemes} \\ \text{b)} & 0<\alpha<2\text{: Low accuracy schemes} \\ \text{c)} & \alpha=2\text{: Standard accuracy} \\ \text{d)} & 2<\alpha<6\text{: High accuracy} \\ \text{e)} & \alpha=6\text{: Extreme accuracy} \end{array}\right\} \qquad (3.5.2)$$

Examples of these types will be presented in this section.

The third criterion (i.e. simplicity) is clearly concerned with the amount of computational work, i.e. arithmetical operations involved to get the solution. The simpler the scheme is the less work and time that is required.

i) Asymmetric Finite Difference Equations

The general versions of the two equations (3.2.13) and (3.2.14) have been introduced by Saul'yev (1964,p.31). They are of O(h) and given by,

$$(1+\theta r)u_{i,j+1}-\theta r u_{i-1,j+1} = (1-\theta)r u_{i-1,j}+[1+(\theta-2)r]u_{i,j}+r u_{i+1,j}, \qquad (3.5.3)$$

and,

$$(1+\theta r)u_{i,j+1}-\theta r u_{i+1,j+1} = r u_{i-1,j}+[1+(\theta-2)r]u_{i,j}+(1-\theta)r u_{i+1,j}, \qquad (3.5.4)$$

where $0\leq\theta\leq1$. Figs.(3.5.1a) and (3.5.1b) show the molecular diagram of (3.5.3) and (3.5.4). Both formulae generate the classical explicit formula when $\theta=0$, and they reduce to the formulae (3.2.13) and (3.2.14) respectively when $\theta=1$.

These two formulae can be used explicitly if they are applied from left to right for the first formula and in the opposite direction

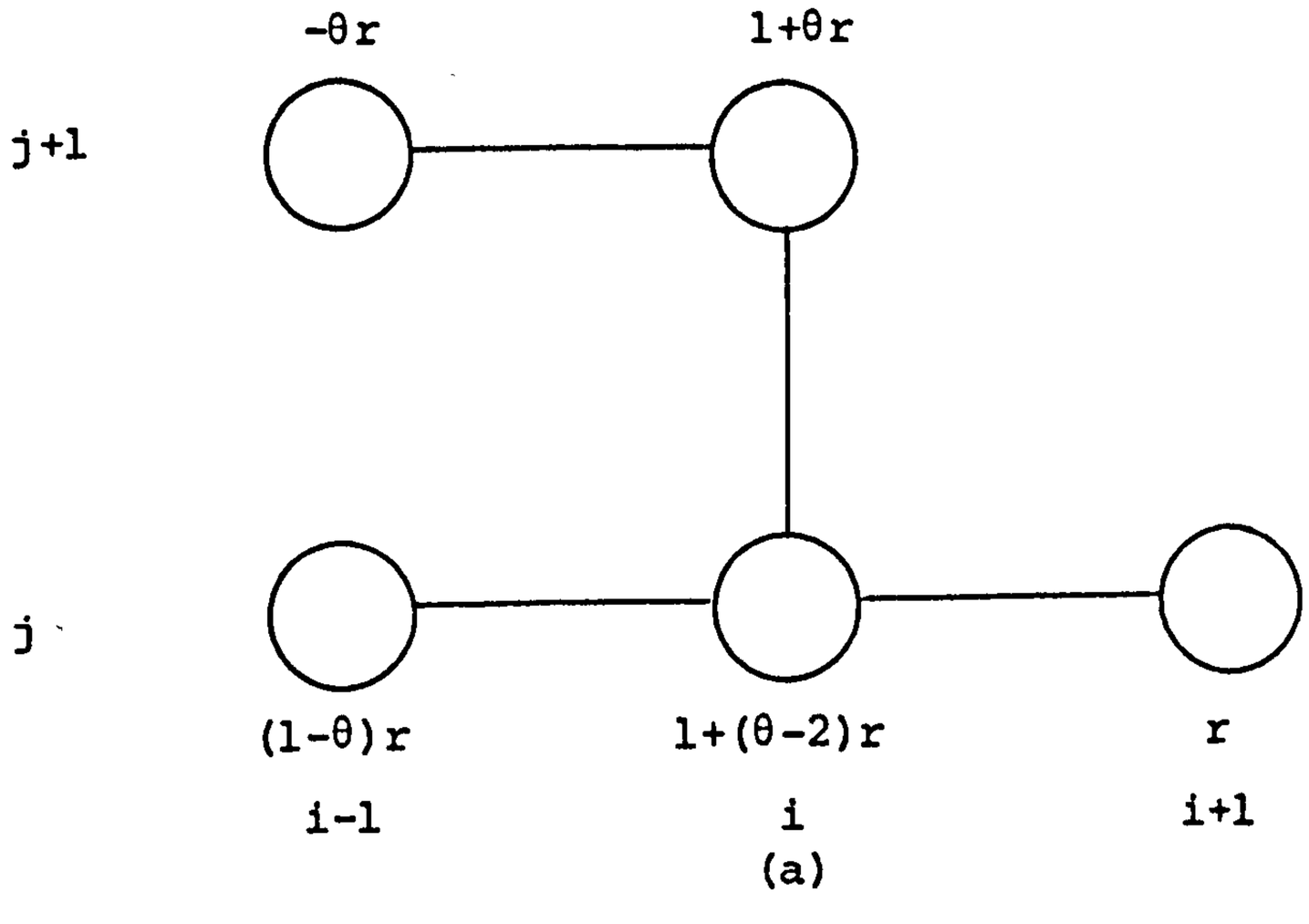


FIGURE 3.5.1

for the second. Accordingly this applies to equations (3.2.13) and (3.2.14), respectively, as well. For example, if formula (3.5.3) is applied on the first interior point i from the left boundary at the time level 0, then the functions at the points (i-1,0), (i,0) and (i+1,0) are known from the initial conditions. The function at (i-1,1) is known from the boundary conditions. Hence the only point where the function is unknown is (i,1) and that can be found by using equation (3.5.3) as,

$$(1+\theta r)u_{i,j+1} = \theta r u_{i-1,j+1} + (1-\theta) r u_{i-1,j} + [1+(\theta-2)r]u_{i,j} + r u_{i+1,j}. \tag{3.5.5}$$

This will be applied at the second interior point as now, the first point is known, and so on to the end of the line. Similarly with equation (3.5.4), (3.2.13) and (3.2.14) according to their direction. However, these equations are not accurate enough to be used, by themselves, for the numerical solution of parabolic equations. As their order of error is $O(h+k)$ whereas the classical explicit and the fully implicit formulae are of $O(h^2+k)$. These formulae are of the type b) of the classification (3.5.2). It was mentioned by A.F. Filipov (SAUL'YEV, 1964, p.34) that in the case of Cauchy's problem, a necessary and sufficient condition for the stability of equations (3.5.3) and (3.5.4) is that,

$$r \leq \frac{1}{2(1-\theta)} \quad . \tag{3.5.6}$$

This verifies the stability condition of equations (3.2.13) and (3.2.14), i.e. when $\theta=1$ we get $r \leq \infty$. As the order of error of formulae (3.5.3) and (3.5.4) does not encourage their use, Saul'yev (1964, p.43) has suggested the use of both formulae alternatively at the time steps. In this alternative strategy, the two equations seem to support each other at the weak side of their asymmetry by having their low order of error $O(h)$ diminished. This alternating method can be written as,

$$u_{i,j+1} = \frac{1}{1+\theta r} [\theta r u_{i-1,j+1} + (1-\theta) r u_{i-1,j} + \{1+(\theta-2)r\}u_{i,j} + r u_{i+1,j}], \tag{3.5.7}$$

and

$$u_{i,j+2} = \frac{1}{1+\theta r} [\theta r u_{i+1,j+2} + r u_{i-1,j} + \{1+(\theta-2)r\}u_{i,j} + (1-\theta) r u_{i+1,j}], \tag{3.5.8}$$

where $j=0,2,4,\ldots$ and $i=1,2,\ldots,n-1$ for (3.5.7) and $i=n-1,n-2,\ldots,1$

for (3.5.8). This method is then comparable with the classical explicit and fully implicit methods.

Further, if we put equations (3.5.3) and (3.5.4) in matrix form, i.e.,

$$\left.\begin{aligned} Au_{j+1} &= (A+C)u_j + b \\ A^T u_{j+1} &= (A^T+C)u_j + b \end{aligned}\right\} \qquad (3.5.9)$$

and

respectively, where,

$$A = \begin{bmatrix} 1+\theta r & & & & \\ -\theta r & 1+\theta r & & & \\ & -\theta r & \ddots & & \\ & & \ddots & 1+\theta r & \\ & & & -r\theta & 1+\theta r \end{bmatrix}, \quad C = \begin{bmatrix} -2r & r & & & \\ r & -2r & r & & \\ & r & \ddots & & \\ & & \ddots & -2r & r \\ & & & r & -2r \end{bmatrix} \qquad (3.5.10)$$

and b is the vector containing the associated boundary conditions. Then at every time step, we define the average value of u from both equations, i.e.,

$$u_{i,j+1} = \frac{\bar{u}_{i,j+1} + \bar{\bar{u}}_{i,j+1}}{2} \quad , \qquad (3.5.11)$$

where,

$$\left.\begin{aligned} A\bar{u}_{j+1} &= (A+C)u_j + b \\ A^T\bar{\bar{u}}_{j+1} &= (A^T+C)u_j + b \end{aligned}\right\} \quad . \qquad (3.5.12)$$

and

Substituting (3.5.12) in (3.5.11) we can write $u_{i,j+1}$ as,

$$u_{i,j+1} = \left\{I + \frac{A^{-1}+(A^T)^{-1}}{2} C\right\}u_j + \{A^{-1}+(A^T)^{-1}\}b \quad , \qquad (3.5.13)$$

where,

$$\frac{A^{-1}+(A^T)^{-1}}{2} = \frac{r}{2(1+\theta r)} \begin{bmatrix} 2 & a & a^2 & \cdots & a^{n-2} \\ a & 2 & a & \cdots & a^{n-3} \\ a^2 & a & 2 & \cdots & a^{n-4} \\ \vdots & \vdots & \vdots & & \vdots \\ a^{n-2} & a^{n-3} & a^{n-4} & \cdots & 2 \end{bmatrix}$$

where $$a = \frac{\theta r}{1+\theta r} \quad .$$

The formula (3.5.13) is stable for all values of r where,

$$r \leq \frac{1}{2(1-\theta)} \quad \text{(SAUL'YEV 1964, p.53)} \quad . \tag{3.5.14}$$

This method and the alternating method have an error of order (almost) $O(h^2)$.

This was an example of how a scheme of category (b) can be improved gradually to category (c), according to the classification (3.5.2).

ii) Formulae with Choice of Parameters

Consider the two equations (3.5.3) and (3.5.4). If the combination of them is to be an approximation to

$$LU = 0 \quad , \tag{3.5.15}$$

and is defined as $L_{\Delta x}u$, then we have,

$$L_{\Delta x}u_{i,j} = -\theta r(u_{i-1,j+1}+u_{i+1,j+1})+2(1+\theta r)u_{i,j+1}-r(2-\theta)(u_{i-1,j}+u_{i+1,j})$$

$$-2[1+r(\theta-2)]u_{i,j} = 0 \quad . \tag{3.5.16}$$

This implicit equation has an optimal accuracy, with error of order $O(h^2)$ i.e. it belongs to type (c) of (3.5.2) but it is no longer explicit. The following theorem will show how equation (3.5.15) can

have a higher order of accuracy for different choices of parameters.

Theorem 3.5.1: (SAUL'YEV 1964, P.92)

If the solution of equation (3.5.15) has derivatives up to the 8^{th} order which are bounded in absolute magnitude throughout D, then the following relations hold in D:

$$L(u_{i,j})-L_{\Delta x}(u_{i,j}) \begin{cases} O(h^2) \text{ if } \theta \neq 1-\frac{1}{6r}, & \theta \geq 1-\frac{1}{2r} \\ O(h^4) \text{ if } \theta = 1-\frac{1}{6r}, & r \neq \frac{1}{2\sqrt{5}} \\ O(h^6) \text{ if } \theta = 1-\frac{1}{6r}, & r = \frac{1}{2\sqrt{5}} \end{cases} \qquad (3.5.17)$$

where $L(u_{i,j})$ and $L_{\Delta x}(u_{i,j})$ are the differential and difference expressions respectively.

Proof:

In the proof two subjects will be considered, the stability and the accuracy of equation (3.5.16).

1. To investigate the stability we will use the matrix method. Hence equation (3.5.16) can be written in matrix form as,

$$A_1 u_{i,j+1} = A_2 u_{i,j} + b$$

or
$$u_{i,j+1} = A_1^{-1}(A_2 u_{i,j}+b) , \qquad (3.5.18)$$

where $A_1=\theta C-2I$, $A_2=-2I+(\theta-2)C$,

C is the tridiagonal matrix as given in (2.5.10) and I is the identity matrix. Since the eigenvalues of C are given by,

$$\lambda_s(C) = -4r \sin^2 \frac{s\pi}{2n}, \quad s=1,2,\ldots,n-1 \qquad (3.5.19)$$

therefore,
$$\lambda_s(A_1) = -4\theta r \sin^2 \frac{s\pi}{2n} - 2 , \qquad (3.5.20)$$

and
$$\lambda_s(A_2) = -2-4(\theta-2)r\sin^2 \frac{s\pi}{2n} , \qquad (3.5.21)$$

For the stability of (3.5.16) we need,

$$\left| \frac{\lambda_s(A_2)}{\lambda_s(A_1)} \right| = \left| \frac{2+4(\theta-2)r \sin^2 \frac{s\pi}{2n}}{2+4\theta r \sin^2 \frac{s\pi}{2n}} \right| \leq 1, \qquad (3.5.22)$$

or $$-2-4\theta r \sin^2 \frac{s\pi}{2n} \leq 2+4(\theta-2)r\sin^2 \frac{s\pi}{2n} \leq 2+4\theta r\sin^2 \frac{s\pi}{2n} \qquad (3.5.23)$$

The righthand side of this inequality is always fulfilled and for the l.h.s. we have,

$$8r(\theta-1)\sin^2 \frac{s\pi}{2n} \geq -4 ,$$

or $$r \leq \frac{1}{2(1-\theta)} , \qquad (3.5.24)$$

which is the stability condition for equation (3.5.16). Thus, for any value of θ there is a specific value of r that satisfies the stability of (3.5.16).

2) For the accuracy of (3.5.16) we apply a Taylor series expansion to replace the values of u involved in (3.5.16) in the neighbourhood of (ih,(j+1)k) to obtain,

$$L_{\Delta x}(u_{i,j+1}) - L(u_{i,j+1}) = \frac{-h^2}{12}(6r-6\theta r-1) \frac{\partial^2 u}{\partial t^2}_{i,j+1} - \frac{h^4}{360}(120r^2+1-30r+15\theta r-90\theta r^2) \frac{\partial^3 u}{\partial t^3}_{i,j+1} + O(h^6). \qquad (3.5.25)$$

This general formula expresses the accuracy of (3.5.16). When $\theta \neq 1-1/6r$, then no change will happen and the first condition of (3.5.17) is held. For $\theta=1-1/6r$, the coefficient of h^2 in (3.5.25) vanishes, and hence the second condition of (3.5.17) holds.

For the third condition of (3.5.17) to hold, we need the coefficient of h^4 in (3.5.25) to vanish, i.e. the term in the brackets

$$120r^2 + 1-30r + 15\theta r - 90\theta r^2 = 0 ,$$

which is satisfied for $\theta=1-1/6r$ and $r=\frac{1}{2\sqrt{5}}$. Hence an accuracy of $O(h^6)$ has been achieved. This shows how the scheme with accuracy of type (c) in classification (3.5.2), can be moved to type (d) and up to type (e)

with a proper choice of parameters.

Remarks on Theorem 3.5.1

1. For $\theta=0$ and $r=1/6$, an explicit equation with an accuracy $O(h^4)$ is obtainable. The scheme,

$$u_{i,j+1} = \frac{1}{6}(u_{i-1,j}+u_{i+1,j}) + \frac{2}{3}u_{i,j}\ , \qquad (3.5.26)$$

was considered by Milne (1953, p.134) and D. Yu Panow (1955, p.125).

2. For $\theta=1-1/6r$, the formula of Douglas is obtained with accuracy of $O(h^4)$ when $r\neq \frac{1}{2\sqrt{5}}$ and $O(h^6)$ when $r = \frac{1}{2\sqrt{5}}$. The formula is written as,

$$(1-6r)(u_{i-1,j+1}+u_{i+1,j+1})+(10+12r)u_{i,j+1} = (1+6r)(u_{i-1,j}+u_{i+1,j}) + (10-12r)u_{i,j}\ . \qquad (3.5.27)$$

3. The stability condition (3.5.14) for the explicit equation (3.5.13) coincides with the condition (3.5.24) of the implicit equation (3.5.16).

4. The identity

$$\frac{\partial^s u}{\partial t^s} = \frac{\partial^{s+\alpha} u}{\partial t^{s-\alpha}\partial x^{2\alpha}}\ , \quad s=1,2,\ldots \quad \alpha=1,2,\ldots,s,$$

has been used for the diffusion equation in Theorem 3.5.1 but it cannot be extended directly to the cases of variable coefficient and multidimensional equations.

We have already seen different ways of achieving a higher order of accuracy:

a) Altering the direction in space, at every time step, when solving the equation.

b) Taking the arithmetic mean of solutions in different directions.

c) Imposing a more strict choice on the free parameters.

Other methods of increasing accuracy are as follows.

iii) Alternating Method

In this method the two classical implicit and explicit formulae are used alternatively at every time step. Although the explicit scheme is only stable for small time steps compared with the implicit scheme, this combination seems to cancel this restriction and is valid for any time step.

Theorem 3.5.2

The scheme,

$$\frac{u_{i,2j+1}-u_{i,2j}}{k} = \frac{u_{i-1,2j+1}-2u_{i,2j+1}+u_{i+1,2j+1}}{h^2} \qquad (3.5.28a)$$

$$\frac{u_{i,2j+2}-u_{i,2j+1}}{k} = \frac{u_{i-1,2j+1}-2u_{i,2j+1}+u_{i+1,2j+1}}{h^2} \qquad (3.5.28b)$$

is absolutely stable, if the time step Δt (i.e. k) is constant or changes after an even number of time steps [SAUL'YEV, 1964, p.23].

Proof:

The two equations (3.5.28a) and (3.5.28b) can be written in matrix form as,

$$Au_{2j+1} = u_{2j} \ , \qquad (3.5.29a)$$

and

$$u_{2j+2} = Bu_{2j+1} \ , \qquad (3.5.29b)$$

respectively, where,

$$A = I-rT \text{ and } B = I+rT \ , \qquad (3.5.30)$$

with I as the identity matrix and T is given as,

$$T = \begin{bmatrix} 2 & -1 & & & O \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ O & & & -1 & 2 \end{bmatrix} . \qquad (3.5.31)$$

The two equations (3.5.29a) and (3.5.29b) can be written as,

$$u_{2j+2} = (BA^{-1})u_{2j} . \qquad (3.5.32)$$

To prove the stability of this equation it is necessary and sufficient for the *amplification matrix* (BA^{-1}) to satisfy,

$$||BA^{-1}|| \leq 1 . \qquad (3.5.33)$$

As the eigenvalues of T are given by,

$$\lambda_s(T) = -2+2\cos\frac{s\pi}{n} , \quad s=1,2,\ldots,n-1,$$
$$= -4\sin^2\frac{s\pi}{2n} .$$

Therefore,
$$\lambda_s(A) = 1+4r\sin^2\frac{s\pi}{2n} , \qquad (3.5.34)$$

and
$$\lambda_s(B) = 1-4r\sin^2\frac{s\pi}{2n} . \qquad (3.5.35)$$

Thus the stability condition for the equation (3.5.32) is,

$$\frac{1-4r\sin^2\frac{s\pi}{n}}{1+4r\sin^2\frac{s\pi}{n}} \leq 1 , \qquad (3.5.36)$$

which holds for all r>0.

We shall mention that the order of use of (3.5.28a) and (3.5.28b) does not make any change in the Theorem (3.5.2). Another way of using the explicit and implicit formulae alternatively is nodewise along the same line instead of linewise. In this method the implicit equation is

used explicitly, thus it is called *Implicitly-Explicit*. It was proposed by Saul'yev (1964, p.66), Gordon (1965) and was called Hopscotch by Gourlay (1970) later. In a simple form it can be written as,

$$u_{2i,j+1} = ru_{2i-1,j}+(1-2r)u_{2i,j}+ru_{2i+1,j}, \quad 1\leq i\leq\frac{n}{2} \tag{3.5.37a}$$

and

$$u_{2i-1,j+1} = \frac{1}{1+2r}\{u_{2i-1,j}+ru_{2i,j+1}+ru_{2i-2,j+1}\}, \quad 1\leq i\leq\frac{n+1}{2}. \tag{3.5.37b}$$

Figure 3.5.2 shows how equation (3.5.37b) can be applied explicitly, where equation (3.5.37a) is applied at the nodes denoted by the letter E and (3.5.37b) is applied at the nodes denoted by I.

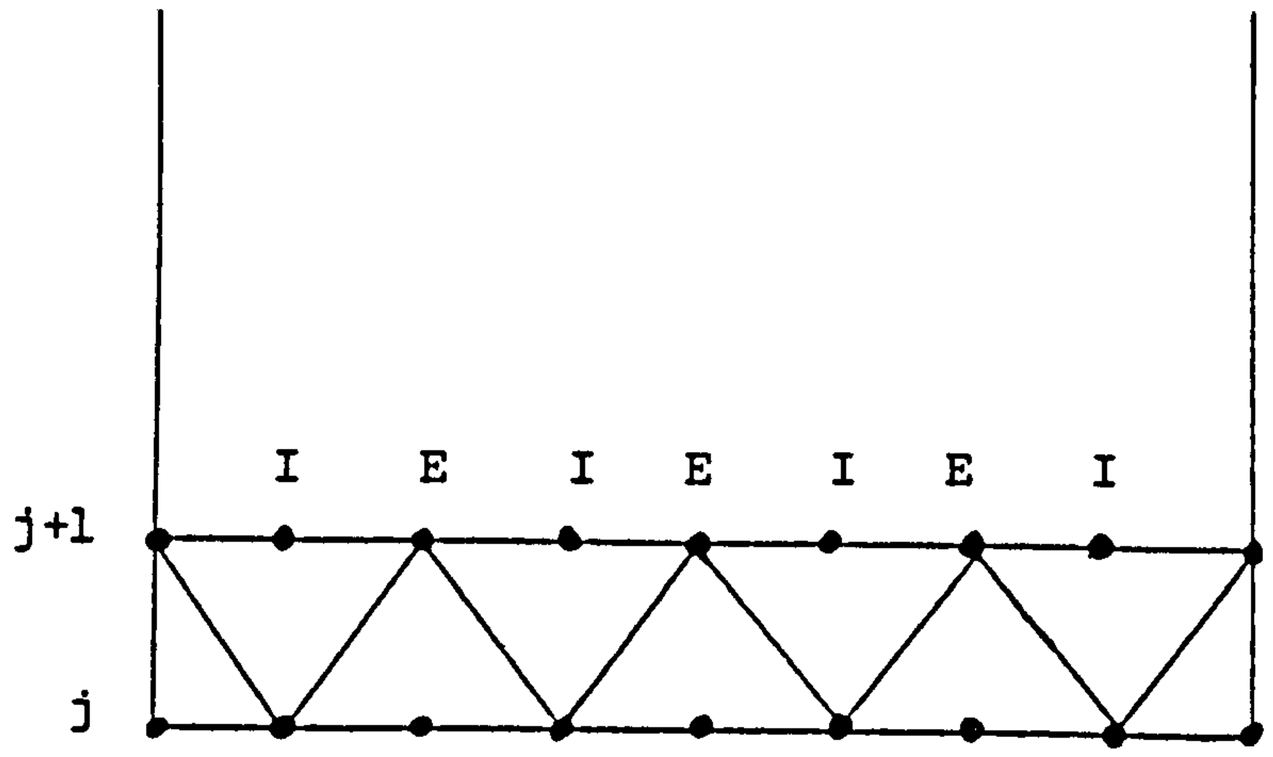


FIGURE 3.5.2

In this method we have to keep the order of the equations that are used, i.e. to apply the explicit equation first as it provides a known value at the $j+1^{th}$ time level which is then used in the implicit equation.

Other ways of increasing the accuracy of the equations (3.2.6) and (3.2.7) is to take the average, i.e.,

$$\frac{u_{i,j+1}-u_{i,j}}{k} = \tfrac{1}{2}\left(\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}+\frac{u_{i+1,j+1}-2u_{i,j+1}+u_{i-1,j+1}}{h^2}\right) \tag{3.5.38}$$

which is the well known Crank-Nicolson formula of accuracy $O(h^2+k^2)$.

The summation of (3.5.28a) and (3.5.28b) gives the formula,

$$\frac{u_{i,2j+2}-u_{i,2j}}{2k} = \frac{u_{i+1,2j+1}-2u_{i,2j+1}+u_{i-1,2j+1}}{h^2}, \qquad (3.5.39)$$

which is the unstable Richardson formula of accuracy $O(h^2+k^2)$.

Formally it is written as,

$$\frac{u_{i,\ell+1}-u_{i,\ell-1}}{2k} = \frac{u_{i-1,\ell}-2u_{i,\ell}+u_{i+1,\ell}}{h^2}, \quad \ell=2j+1. \qquad (3.5.40)$$

iv) Multi-Level Difference Schemes

A higher order accuracy is produced if more time levels than the minimum number required by the differential equation are used in constructing a difference scheme. As an example of improving the accuracy of the classical implicit scheme for the heat equation (3.2.2) we have,

$$\frac{u_{i,j+1}-u_{i,j}}{k} = \frac{u_{i-1,j+1}-2u_{i,j+1}+u_{i+1,j+1}}{h^2} + O(k)+O(h^2), \qquad (3.5.41)$$

which uses the three time levels, of Richtmyer and Morton (1967, p.86) given below,

$$\frac{3/2\, u_{i,j+1}-2u_{i,j}+\frac{1}{2}u_{i,j-1}}{k} = \frac{u_{i+1,j+1}-2u_{i,j+1}+u_{i-1,j+1}}{h^2} + O(k^2)+O(h^2). \qquad (3.5.42)$$

It is clearly seen that in a three time level scheme such as (3.5.42), initial conditions are required on two time levels j-1 and j to obtain the solution on the $(j+1)^{th}$ time level. At the beginning of the process, where only the initial conditions at the zero time level is available, a simple two-time level scheme needs to be used. When the

extra initial data on time level 1 then becomes available, the scheme (3.5.42) will be applicable. It is essential that the data calculated at level 1 should be of an accuracy comparable with that of the three-level scheme. [MITCHELL, 1969].

Equation (3.5.42) can be written as,

$$-2ru_{i-1,j+1}+(3+4r)u_{i,j+1}-2ru_{i+1,j+1} = 4u_{i,j}-u_{i,j-1} \ . \qquad (3.5.43)$$

In matrix form, this becomes,

$$u_{j+1} = 4A^{-1}u_j - A^{-1}u_{j-1} + A^{-1}c_{j+1} \ , \qquad (3.5.44)$$

where c_{j+1} is the vector of the associated boundary conditions and,

$$A = \begin{bmatrix} (3+4r) & -2r & & & \\ -2r & 3+4r & -2r & & O \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -2r \\ O & & & -2r & (3+4r) \end{bmatrix} \ .$$

If we put,

$$w_{j+1} = \begin{bmatrix} u_{j+1} \\ u_j \end{bmatrix} \ , \qquad (3.5.45)$$

then we can write equation (3.5.44) in the partitioned form as,

$$\begin{bmatrix} u_{j+1} \\ \hline u_j \end{bmatrix} = \left[\begin{array}{c|c} 4A^{-1} & -A^{-1} \\ \hline I & O \end{array}\right] \begin{bmatrix} u_j \\ \hline u_{j-1} \end{bmatrix} + \begin{bmatrix} c_j \\ \hline O \end{bmatrix} \ , \qquad (3.5.46)$$

or $$w_{j+1} = Pw_j + c_j \ , \qquad (3.5.47)$$

where P is the coefficient matrix in (3.5.46).

This is now reduced to a two level formula which will be stable

if the modulus of the eigenvalues of P are less than or equal to unity. Assuming that the matrix A, of order n-1, has n-1 distinct eigenvalues, λ_s, s=1,2,...,n-1. Then it has n-1 linearly independent eigenvectors. The eigenvalues μ of P are the eigenvalues of the matrix,

$$\begin{bmatrix} 4/\lambda_s & -1/\lambda_s \\ 1 & 0 \end{bmatrix} ,$$

where λ_s, s=1,2,...,n-1 are described above, and are given by,

$$\lambda_s = 3 + 8r \sin^2 \frac{s\pi}{2n} \quad , s=1,2,...,n-1 . \tag{3.5.48}$$

From,

$$\det(P-\mu I) = \det \begin{bmatrix} 4/\lambda_s-\mu & -1/\lambda_s \\ 1 & -\mu \end{bmatrix} = 0 ,$$

we get,

$$\mu^2 - (\frac{4}{\lambda_s})\mu + \frac{1}{\lambda_s} = 0 ,$$

or,

$$\mu = \frac{2\pm\sqrt{4-\lambda_s}}{\lambda_s}$$

$$= [2\pm\sqrt{1-8r\sin^2 \frac{s\pi}{2n}}]/(3+8r\sin^2 \frac{s\pi}{2n}) ,$$

if the roots are real, $|\mu| \leq (2\pm1)/(3+\sigma)$, $\sigma>0$, hence $|\mu|<1$.

When the roots are complex, $|\mu| = \frac{1}{(3+8r\sin^2 \frac{s\pi}{2n})^{\frac{1}{2}}} < 1$. Therefore, the scheme (3.5.42) is unconditionally stable.

As the above example is an implicit scheme, now we demonstrate an explicit scheme. Richardson (1910) has proposed the scheme,

$$\frac{u_{i,j+1}-u_{i,j-1}}{2k} = \frac{u_{i-1,j}-2u_{i,j}+u_{i+1,j}}{h^2} + O(k^2)+O(h^2) . \tag{3.5.49}$$

As mentioned earlier, although it has an order of accuracy of $O(k^2)+O(h^2)$, it is unconditionally unstable. To show this we will use the same partitioning technique to reduce it to two time levels.

Equation (3.5.49) can be written as,

$$u_{i,j+1} = 2r\delta_x^2 u_{i,j} + u_{i,j-1} \quad . \tag{3.5.50}$$

As in (3.5.45) let,

$$w_{j+1} = \begin{bmatrix} u_{i,j+1} \\ u_{i,j} \end{bmatrix} \quad ,$$

equation (3.5.50) can then be put in matrix form as,

$$\begin{bmatrix} u_{j+1} \\ u_j \end{bmatrix} = \begin{bmatrix} 2r\delta_x^2 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_j \\ u_{j-1} \end{bmatrix} \quad , \tag{3.5.51}$$

or

$$w_{j+1} = Pw_j \quad , \tag{3.5.52}$$

where,

$$P = \begin{bmatrix} 2r\delta_x^2 & 1 \\ 1 & 0 \end{bmatrix} \quad .$$

As before using a Fourier series to represent the error terms,

$$w_{i,j} = w_{0,j} e^{i\beta ih} \quad ,$$

where $w_{0,j}$ is a constant vector, and substituting in (3.5.52) gives,

$$w_{j+1} = \begin{bmatrix} -8r\sin^2 \frac{\beta h}{2} & 1 \\ 1 & 0 \end{bmatrix} w_j \quad , \tag{3.5.53}$$

The amplification matrix above has the eigenvalues,

$$\lambda_i = -4r\sin^2 \frac{\beta h}{2} \pm (1+16r^2\sin^4 \frac{\beta h}{2})^{\frac{1}{2}} \quad , \; i=1,2.$$

For the stability of (3.5.52) we need that $|\lambda_i| \leq 1$, $i=1,2$.

This condition is violated by λ_2, i.e. when the negative sign is used, for all r, [MITCHELL, A.R., 1969, p.88]. If however, the initial boundary value problem considered is reduced to a pure boundary value problem, then the Richardson scheme becomes stable [SAUL'YEV, 1964,p.90].

The stability problem of Richardson's formula can be overcome and explicitly maintained if DuFort-Frankel's suggestion is used. DuFort-Frankel (1953) replaced the term $2u_{i,j}$ by $(u_{i,j-1}+u_{i,j+1})$, therefore (3.5.49) becomes,

$$\frac{u_{i,j+1}-u_{i,j-1}}{2k} = \frac{u_{i+1,j}-(u_{i,j-1}+u_{i,j+1})+u_{i-1,j}}{h^2} , \qquad (3.5.54)$$

which can be written as,

$$(1+2r)u_{i,j+1} = 2r(u_{i+1,j}+u_{i-1,j})+(1-2r)u_{i,j-1} . \qquad (3.5.55)$$

In matrix form this becomes,

$$Au_{j+1} = Bu_j + Cu_{j-1} , \qquad (3.5.56)$$

where,

$$A = (1+2r)I, \quad C = (1-2r)I ,$$

and,

$$B = 2r \begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 0 & 1 \\ & & & 1 & 0 \end{bmatrix} .$$

Reducing the formula to two time levels, from equation (3.5.56) we get,

$$w_{j+1} = Pw_j , \quad P = \begin{bmatrix} A^{-1}B & A^{-1}C \\ I & 0 \end{bmatrix} . \qquad (3.5.57)$$

For stability, we require that the eigenvalues μ, of P to be less than or equal unity in modulus.

For P we have,

$$|\mu^2 - \mu A^{-1}B - A^{-1}C| = 0 , \qquad (3.5.58)$$

or

$$|\mu^2 A - \mu B - C| = 0 . \qquad (3.5.59)$$

This in fact can be written directly as,

$$(1+2r)\mu^2-4r\mu\cos\frac{s\pi}{n}-(1-2r)=0\ . \qquad (3.5.60)$$

which shows that $|\mu|<1$ for all values of r, therefore (3.5.54) is stable.

However, although this scheme is stable, it suffers from the inconsistency with the heat equation as discussed in Section 2.6.

A general three-time level explicit difference scheme for the approximation of the heat equation (3.2.2) is given by Jain (1979, p.214). It involves seven points and may be written as,

$$(1+\tau)u_{i,j+1}=[1+2\tau+r(1-\gamma)\delta_x^2]u_{i,j}-(\tau-r\gamma\delta_x^2]u_{i,j-1}\ , \qquad (3.5.61)$$

where τ and γ are arbitrary parameters. The truncation error of (3.5.61) is of order,

1. $(\Delta t+(\Delta x)^2)$ if γ and τ are arbitrary.
2. $(\Delta t)^2+(\Delta x)^2$ if $\tau+\gamma+\frac{1}{2}=0$ and either γ or τ is arbitrary.
3. $(\Delta x)^4$ if $\tau+\gamma+\frac{1}{2}-\frac{1}{12r}=0$ and either γ or τ is arbitrary.

(3.5.62)

The stability condition for (3.5.61) is:

1. $1+2\tau-2(1-2\gamma)r\geq 0$,
2. $1-4\gamma r\geq 0$.

(3.5.63)

We find that for:

1. $\gamma\leq 0$, the conditions (3.5.63) are satisfied if,

$$0<r<\frac{1+2\tau}{2(1-2\gamma)} \quad \text{and } 1+2r>0\ ,$$

2. $\gamma<\frac{1}{2}$, the stability condition is obtained as $0<r<r_{min}$, where $1+2\tau>0$ and,

$$r_{min} = \min\left[\frac{1}{4\gamma}, \frac{1+2\tau}{2(1-2\gamma)}\right] \quad . \tag{3.5.64}$$

The above conditions are shown in Fig.(3.5.3).

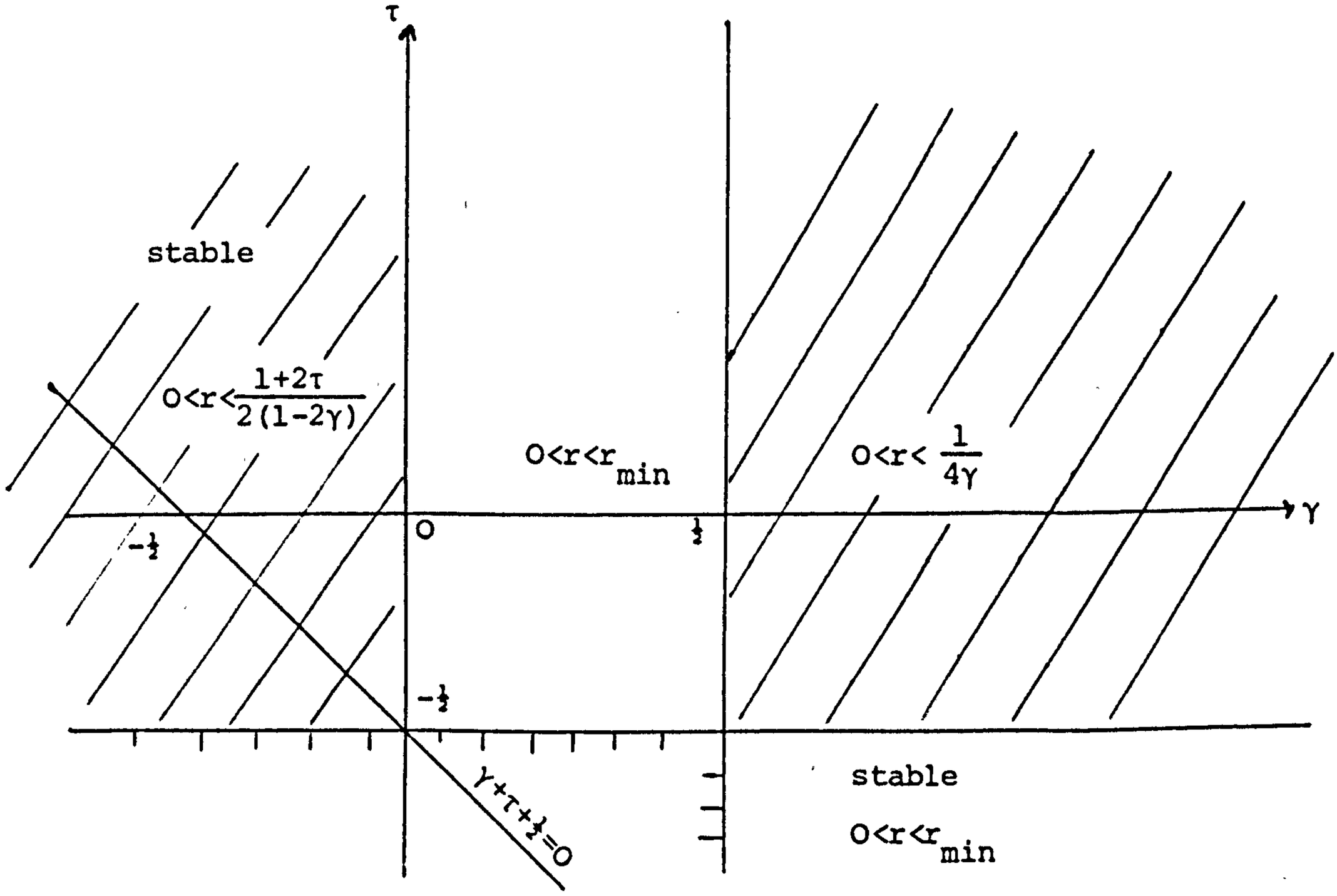


FIGURE 3.5.3

Having seen some advantages of the multi-level formulae over some two-level formulae as far as the accuracy is concerned, they, in fact, have some disadvantages, where the stability in Richardson's method and the consistency in DuFort-Frankel's method are serious problems.

3.6 DERIVATIVE BOUNDARY CONDITIONS

Boundary conditions of derivative type are not less frequent, in practice, than other types of boundary conditions.

Consider the heat equation (3.2.2) subject to the boundary conditions,

$$\frac{\partial U}{\partial x} = p(U-w_1) \text{ at } x=0,\ 0<t\leq T\ , \qquad (3.6.1a)$$

$$\frac{\partial U}{\partial x} = -q(U-w_2) \text{ at } x=1,\ 0<t\leq T\ , \qquad (3.6.1b)$$

where p,q,w_1,w_2 are constants and $p,q\geq 0$.

These conditions do not give the temperatures at the end points, i.e. the temperature is still unknown at the end points. Therefore we need two more equations, one for each end. These equations are the difference approximations of the derivatives on the boundary conditions (3.6.1a) and (3.6.1b). A forward difference for the left end and backward difference for the right end can be used, see Figure(3.6.1), and equations (3.6.1a) and (3.6.1b) will be,

$$\frac{u_{1,j}-u_{0,j}}{\Delta x} \cong p(u_{0,j}-w_1)\ , \qquad (3.6.2a)$$

$$\frac{u_{n,j}-u_{n-1,j}}{\Delta x} \cong -q(u_{n,j}-w_2)\ , \qquad (3.6.2b)$$

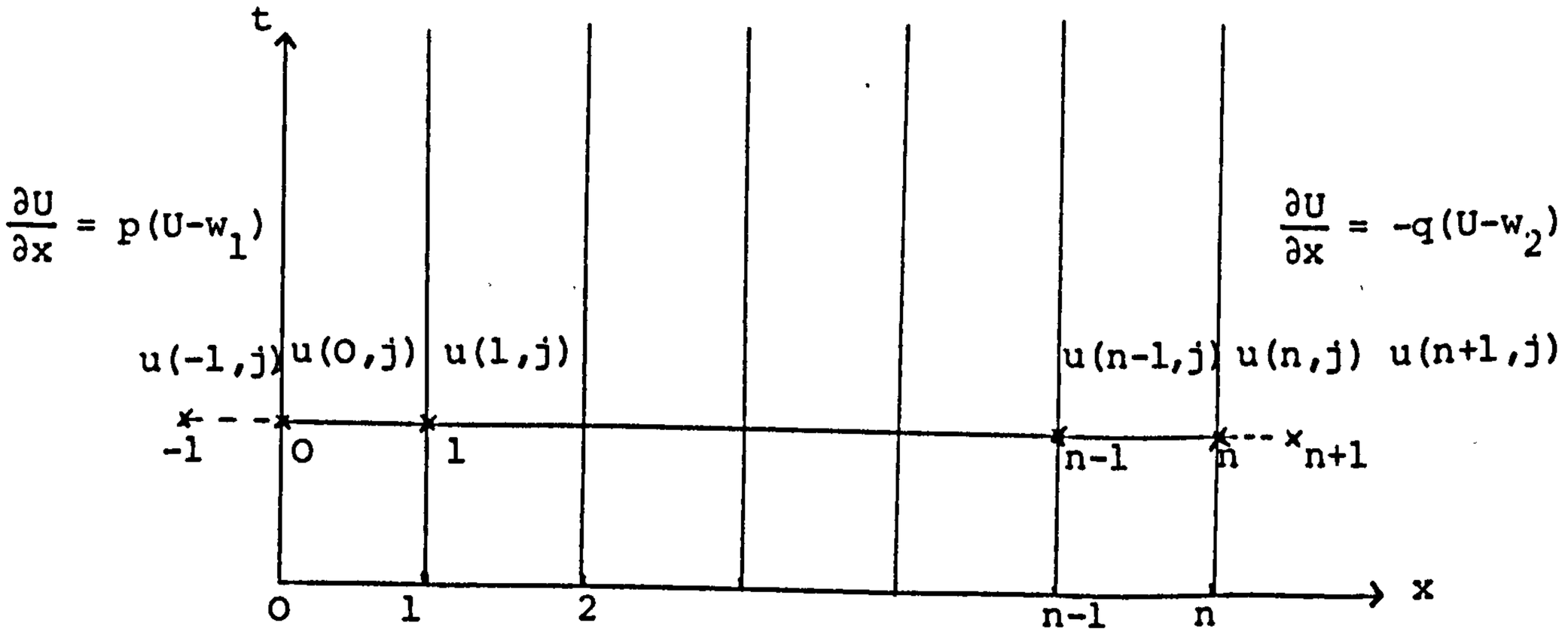


FIGURE 3.6.1

respectively. For a more accurate approximation, the central difference is used, and is written as,

$$\frac{u_{1,j}-u_{-1,j}}{2\Delta x} \cong p(u_{0,j}-w_1), \text{ for the left end,} \qquad (3.6.3a)$$

and

$$\frac{u_{n+1}-u_{n-1}}{2\Delta x} \cong -q(u_{n,j}-w_2), \text{ for the right end,} \qquad (3.6.3b)$$

which can be written as,

$$u_{i-1,j} = -2\Delta x p(u_{0,j}-w_1)+u_{1,j}\ , \qquad (3.6.4a)$$

and,

$$u_{n+1,j} = -2\Delta x q(u_{n,j}-w_2)+u_{n-1,j}\ . \qquad (3.6.4b)$$

This, as we can see, creates two extra nodes, which are not on the domain, namely the points $u_{-1,j}$ and $u_{n+1,j}$, where the temperature is not known. To eliminate them we apply the formula used for the interior points, on these boundary points, assuming that the heat equation is satisfied at the end points. The unknown $u_{-1,j}$ and $u_{n+1,j}$ are obtained as below.

To solve this problem using the Crank-Nicolson formula, we apply equation (2.4.15) for i=0 to get,

$$-ru_{-1,j+1}+(2+2r)u_{0,j+1}-ru_{1,j+1} = ru_{-1,j}+(2-2r)u_{0,j}+ru_{1,j}. \qquad (3.6.5)$$

Substituting $u_{-1,j}$ of equation (3.6.4a), in equation (3.6.5) results in, for all time levels,

$$-2ru_{1,j+1}+(2+2r+2rhp)u_{0,j+1} = 2ru_{1,j}+(2-2r-2rhp)u_{0,j}+4rhpw_1. \qquad (3.6.6)$$

Analogously for the points (n,j) and (n,j+1) then by applying equation (2.4.15) for i=n, and substituting $u_{n+1,j}$ and $u_{n+1,j+1}$ in (3.6.4b) we get,

$$-2ru_{n-1,j+1}+(2+2r+2rhq)u_{n,j+1} = 2ru_{n-1,j}+(2-2r-2rhq)u_{n,j}+4rhqw_2 \tag{3.6.7}$$

Thus, the system of equations can be expressed in the form,

$$Au_{j+1} = Bu_j + b \ , \tag{3.6.8}$$

where $u_j=(u_{0,j},u_{1,j},\ldots,u_{n,j})^T$,

$$A = 2I-rQ, \quad B = 2I+rQ \ ,$$

where Q is an $(n+1)^{th}$ order matrix and is given as,

$$Q = \begin{bmatrix} -2(1+rhp) & 2 & & & \\ 1 & -2 & 1 & & O \\ & \ddots & \ddots & \ddots & \\ & O & 1 & -2 & 1 \\ & & & 2 & -2(1+rhq) \end{bmatrix} ,$$

and b is the $(n+1)^{th}$ component vector involving the boundary conditions and given as $b=(4rhpw_1,0,0,\ldots,4rhqw_2)$.

To prove the stability of (3.6.8) we put it in the form,

$$u_{j+1} = A^{-1}Bu_j + A^{-1}b \ , \tag{3.6.9}$$

where it is assumed that A is non-singular.

We note that the matrix Q is not symmetric and so

$$A^{-1}B = (2I-rQ)^{-1}(2I+rQ) \ .$$

Therefore, we introduce the diagonal matrix,

$$D = \begin{bmatrix} \sqrt{2} & & & \\ & 1 & & O \\ & & \ddots & \\ & O & & 1 \\ & & & & \sqrt{2} \end{bmatrix}$$

so that Q is similar to the symmetric matrix,

$$\tilde{Q} = D^{-1}QD \ ,$$

i.e. they have the same set of eigenvalues. Then,

$$\begin{aligned}
\widetilde{(A^{-1}B)} &= D^{-1}(A^{-1}B)D \\
&= D^{-1}(2I-rQ)^{-1}(2I+rQ)D \\
&= [D^{-1}(2I-rQ)^{-1}D][D^{-1}(2I+rQ)D] \\
&= [D(2I-rQ)D^{-1}]^{-1}[D^{-1}(2I+rQ)D] \\
&= (2I-r\tilde{Q})^{-1}(2I+r\tilde{Q}) \ .
\end{aligned}$$

The two matrices $(2I-r\tilde{Q})^{-1}$ and $(2I+r\tilde{Q})$ are symmetric and commute, and so $\widetilde{(A^{-1}B)}$ is symmetric. Therefore $A^{-1}B$ is similar to the symmetric matrix $\widetilde{(A^{-1}B)}$. Then the necessary and sufficient condition for stability is,

$$\rho(A^{-1}B) \equiv \rho(\widetilde{A^{-1}B}) \leq 1 \ , \qquad (3.6.10)$$

where $\rho(A^{-1}B)$ is the spectral radius of $A^{-1}B$.

Let μ_i (i=0,1,...,n) be the eigenvalues of $A^{-1}B$ given by

$$\mu_i = (2-r\lambda_i)^{-1}(2+r\lambda_i) \ , \qquad (3.6.11)$$

where λ_i (i=0,1,...,n) are the eigenvalues of the matrix Q.

To satisfy the condition (3.6.10) we need λ_i (i=0,1,...,n)$\leq$0, which can be seen to be quite easily satisfied by using the Gerschgorin's Theorem, where we find that,

$$\lambda_i = \left\{ \begin{array}{l} -4-2rhp \\ \text{or } -4 \\ \text{or } -4-4rhq \end{array} \right\} \ , \ i=0,1,...,n,$$

which are negative in all cases. Therefore, equation (3.6.8) is unconditionally stable.

3.7 NON-LINEAR PARABOLIC EQUATIONS

Many of the numerical methods and techniques of proof for linear equations with constant coefficients carry over to nonlinear equations. However conditions of stability and convergence are more complicated to investigate. Direct methods, in general, are difficult to apply, so iterative methods are usually used to solve them.

There are many physical problems of nonlinear behaviour, for example, the heat conduction problem given by,

$$PC\,\frac{\partial U}{\partial t} = \frac{\partial}{\partial x}\left(K\,\frac{\partial U}{\partial x}\right) , \tag{3.7.1}$$

where P,C and K may depend on U.

A more general case of nonlinear parabolic equation is written as,

$$F\left(x,t,U,\frac{\partial U}{\partial t},\frac{\partial U}{\partial x},\frac{\partial^2 U}{\partial x^2}\right) = 0 , \quad \begin{matrix} 0<x<1, \\ 0<t\leq T, \end{matrix} \tag{3.7.2}$$

with the conditions,

$$\left.\begin{aligned} U(x,0) &= f(x) \quad , 0\leq x\leq 1 \\ U(0,t) &= g_1(t) \\ U(1,t) &= g_2(t) \end{aligned}\right\} \quad 0<t\leq T \tag{3.7.3}$$

For special cases of this we mention the work of Douglas (1956), for the quasi-linear parabolic equation,

$$\frac{\partial^2 U}{\partial x^2} = F(x,t,U)\,\frac{\partial U}{\partial t} + G(x,t,U), \quad F\geq a>0. \tag{3.7.4}$$

Also Richtmyer and Morton (1967, p.201) considered the non-linear problem of the form,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U^n}{\partial x^2} , \tag{3.7.5}$$

with n=5.

Stability of the finite difference scheme, for nonlinear equations, depends not only on the form of the scheme but also upon the solution

itself (i.e. whether we have local stability), where the system may be stable for some values t (time) and not for others. In practice, when there is conditional stability, it is necessary, sometimes, to alter Δt in order to restore stability.

Equation (3.7.1) can be written in the form,

$$\frac{\partial^2 U}{\partial x^2} = F(x,t,U, \frac{\partial U}{\partial t}, \frac{\partial^2 U}{\partial x^2}) , \qquad (3.7.6)$$

or

$$\frac{\partial U}{\partial t} = F(x,t,U, \frac{\partial U}{\partial x}, \frac{\partial^2 U}{\partial x^2}) , \qquad (3.7.7)$$

where $\frac{\partial F}{\partial(\partial^2 U/\partial x^2)} \geq a > 0$ for real a.

In analogy, as the weighted average formula,

$$\frac{(u_{i,j+1} - u_{i,j})}{\Delta t} = \theta \frac{1}{(\Delta x)^2} \delta_x^2 (u_{i,j}) + (1-\theta) \frac{1}{(\Delta x)^2} \delta_x^2 (u_{i,j+1}) , \qquad (3.7.8)$$

is used for approximating the linear heat equation (3.2.2), it can also be used to approximate equation (3.7.7), to give,

$$\frac{(u_{i,j+1} - u_{i,j})}{\Delta t} = F(i\Delta x, j\Delta t, [\theta u_{i,j} + (1-\theta) u_{i,j+1}] ,$$

$$[\theta \frac{(u_{i+1,j} - u_{i-1,j})}{2\Delta x}, \quad + \quad , (1-\theta) [\frac{(u_{i+1,j+1} - u_{i-1,j+1})}{2\Delta x}],$$

$$[\theta \frac{\delta_x^2 U_{i,j}}{(\Delta x)^2} + (1-\theta) \frac{\delta_x^2 u_{i,j+1}}{(\Delta x)^2}]) . \qquad (3.7.9)$$

As in the linear case, for $\theta = 1, \frac{1}{2}, 0$, equation (3.7.9) becomes the explicit, Crank-Nicolson and implicit formula respectively.

In the case of equation (3.7.1), a different technique is used. The simplest difference approximation to (3.7.1) is,

$$P(u_{i,j}).C(u_{i,j}) \frac{(u_{i,j+1} - u_{i,j})}{\Delta t} = \frac{1}{(\Delta x)^2} \delta_x [K(u_{i,j}) \delta_x u_{i,j}], \qquad (3.7.10)$$

or

$$P(u_{i,j}).C(u_{i,i})\ \frac{u_{i,j+1}-u_{i,j}}{\Delta t} = \frac{1}{(\Delta x)^2}[K(u_{i+\frac{1}{2},j})(u_{i+1,j}-u_{i,j})-$$

$$K(u_{i-\frac{1}{2},j})(u_{i,j}-u_{i-1,j})] \ . \qquad (3.7.11)$$

Putting,

$$K(u_{i+\frac{1}{2},j}) = K(\frac{u_{i+1,j}+u_{i,j}}{2}) \text{ and}$$

$$K(u_{i-\frac{1}{2},j}) = K(\frac{u_{i,j}+u_{i-1,j}}{2}) \ ,$$

leads to,

$$P(u_{i,j})C(u_{i,j})\ \frac{(u_{i,j+1}-u_{i,j})}{\Delta t} = \frac{1}{h^2}[K(\frac{u_{i+1,j}+u_{i,j}}{2})(u_{i+1,j}-u_{i,j})$$

$$-K(\frac{u_{i,j}+u_{i-1,j}}{2})(u_{i,j}-u_{i-1,j})] \ , \qquad (3.7.12)$$

which is an explicit formula.

Also, an implicit replacement for the nonlinear equation gives rise to a system of nonlinear difference equations. For example, let us consider the equation,

$$L(U) = F(x,t,U) \ , \qquad (3.7.13)$$

where L is a linear partial differential operator. Denoting $L \equiv (\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2})$, the weighted average form of (3.7.13) becomes,

$$\frac{(u_{i,j+1}-u_{i,j})}{\Delta t} = \theta[\frac{\delta_x^2}{(\Delta x)^2}u_{i,j}+F(i\Delta x,j\Delta t,u_{i,j})]+(1+\theta)[\frac{\delta_x^2}{(\Delta x)^2}u_{i,j+1}+$$

$$F(i\Delta x,j\Delta t,u_{i,j+1})] \ , \qquad (3.7.14)$$

or,

$$-\theta r u_{i-1,j+1}+(1+2r\theta)u_{i,j+1}-\theta r u_{i+1,j}-\theta(\Delta t)F(u_{i,j+1}) = r(1-\theta)u_{i-1,j}+$$

$$[1-2r(1-\theta)]u_{i,j}+r(1-\theta)u_{i+1,j}+(\Delta t)(1-\theta)F(u_{i,j}) \ .$$

$$(3.7.15)$$

Let the boundary conditions be given by,

$$\frac{\partial U}{\partial x} = U \ , \ x=0, \ t>0 \ ,$$

$$\frac{\partial U}{\partial x} = -U, \ x=1, \ t>0 \ , \qquad (3.7.16)$$

and the initial conditions $U(x,0)=1$, $0<x\leq 1$, and using the central difference operator to approximate the boundary conditions, we get, after some arrangements, the matrix equation,

$$[I-r(1-\theta)B]u_{j+1}-\Delta t(1-\theta)F(u_{j+1}) = [I+r\theta B]u_j+\Delta t\theta F(u_j) \ , \qquad (3.7.17)$$

where,

$$B = \begin{bmatrix} 2(1+\Delta x) & -2 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -2 & 2(1+\Delta x) \end{bmatrix} .$$

As can be seen from equation (3.7.17), the function F on the left side, is dependent on u_{j+1} and this often accounts for the non-linearity, therefore the system cannot be solved directly, but an iterative (indirect) method needs to be employed.

3.8 ITERATIVE TECHNIQUES

Iterative methods consist of the repeated application of a simple algorithm. They yield the exact answer only as a limit of a sequence, even without consideration of round-off errors.

In any iteration one begins with an initial approximation and then successively modifies the approximation according to some rule. To be useful the iteration must converge but it is not considered to be effective unless the convergence is rapid [W.F. AMES, 1977, p.98].

i) Nonlinear Equations

Let us consider the iterative methods for determining the solutions of the equation,

$$f(x) = 0 \ , \tag{3.8.1}$$

where f and x are vectors of the same dimension n. If n=1, then (3.8.1) is a single equation, and if n>1, then it is a system of n equations. As it is impossible to solve such systems directly, iterative methods of solution are necessary.

Successive substitutions, in which (3.8.1) is first rewritten in an equivalent form,

$$x = g(x) \ , \tag{3.8.2}$$

and use is made then of the recurrence formula,

$$x_{k+1} = g(x_k) \ . \tag{3.8.3}$$

Generally there are many convenient ways of rewriting (3.8.1) in the form (3.8.2), and the convergence or divergence of the sequence of approximations to the limit α may depend upon the particular form chosen.

In order to see this, we assume that g(x) has a continuous

derivative on the closed interval bounded by α and x_k and it is clear that,

$$\alpha = g(\alpha) \ . \tag{3.8.4}$$

Equation (3.8.3) implies that,

$$\alpha - x_{k+1} = g(\alpha) - g(x_k) = (\alpha - x_k) g'(\xi_k) \ , \tag{3.8.5}$$

where the ξ_k lies between x_k and α. If the iteration converges, i.e. $x_k \to \alpha$, then also $g'(\xi_k) = g'(\alpha)$ as $k \to \infty$. If we exclude the cases where $g'(\alpha) = 0$ and $g'(\alpha) = \pm 1$, we can deduce that,

$$\alpha - x_{k+1} \cong (\alpha - x_k) g'(\alpha) \ ,$$

and therefore also that,

$$\alpha - x_k \cong c[g'(\alpha)]^k \ , \text{ as } k \to \infty \ , \tag{3.8.6}$$

where c is a certain constant. This deviation actually would grow unboundedly in magnitude as k increases if $|g'(\alpha)| > 1$. Therefore in order that the iteration converges to α, it is necessary to have $|g'(\alpha)| \lesssim 1$.

Theorem 3.8.1

Let g(x) satisfy the Lipschitz condition,

$$|g(x) - g(x')| \lesssim \lambda |x - x'| \ , \tag{3.8.7}$$

for all values x,x' in the closed interval $I = [x_0 - \varepsilon, x_0 + \varepsilon]$ where the Lipschitz constant λ is such that $0 \lesssim \lambda \lesssim 1$, and let the initial estimate x_0 be such that

$$|x_0 - g(x_0)| \lesssim (1 - \lambda)\varepsilon \ . \tag{3.8.8}$$

Then:

1. all the iterates x_k, defined by (3.8.3), lie within the interval I, i.e.,

$$x_0-\varepsilon \leq x_k \leq x_0+\varepsilon \ ,$$

2. (Existence) the iterates converge to some point, say,

$$\lim_{k\to\infty} x_k = \alpha$$

which is a root of (3.8.2), and

3. (Uniqueness) α is the only root in I.

For proof see[ISAACSON, E. & KELLER, H.B., 1966, p.86]

Corollary

If $|g'(x)| \leq \lambda < 1$ for $|x-x_0| \leq \varepsilon$ and (3.8.8) is satisfied, then the conclusion of Theorem (3.8.1) is valid.

Proof

From the mean value theorem we have,

$$g(x_1)-g(x_2) = g'(\xi)(x_1-x_2), \text{ where } x_1<\xi<x_2 \ ,$$

the Lipschitz constant λ in (3.8.7) is replaced here by $g'(\xi)$.

Definition 3.8.1

Let $x_0, x_1, \ldots,$ be a sequence that converges to α, and $e_k = x_k - \alpha$. If there exists a number P and a constant $c \neq 0$ such that,

$$\lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k|^p} = c \ , \qquad (3.8.9)$$

then P is called the order of convergence of the sequence and c is the asymptotic error constant.

Newton-Raphson Method

Let us consider the nonlinear equation (3.8.1). To derive the method from Taylor's formula we expand about x_k to get,

$$f(x) = f(x_k)+(x-x_k)f'(x_k)+\tfrac{1}{2}(x-x_k)^2 f''(\xi) \quad , \tag{3.8.10}$$

where $\xi(x_k,x)$. By neglecting the quadratic term, and rewriting the equation in iterative form, i.e. $x=x_{k+1}$, we get,

$$f'(x_k)(x_{k+1}-x_k)+f(x_k)\cong f(x)=0, \quad k=1,2,... \tag{3.8.11}$$

The Newton Raphson formula is defined as,

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad . \tag{3.8.12}$$

Recalling equation (3.8.3) gives,

$$g(x_k) = x_k - \frac{f(x_k)}{f'(x_k)} \quad , \tag{3.8.13}$$

and $g(x_k)$ is called the iteration function.

The convergence of the method is found by rewriting equation (3.8.10) as,

$$\frac{f(x_k)}{f'(x_k)} + (x-x_k) = -\tfrac{1}{2}(x-x_k)^2 \frac{f''(\xi)}{f'(x_k)} \quad ,$$

and from (3.8.12),

$$x-x_{k+1} = -\tfrac{1}{2}(x-x_k)^2 \frac{f''(\xi)}{f'(x_k)} \quad . \tag{3.8.14}$$

Let $e_k=x_k-x$, we get,

$$e_{k+1} = \tfrac{1}{2} e_k^2 \frac{f''(\xi)}{f'(x_k)} \quad , \tag{3.8.15}$$

or,

$$\frac{e_{k+1}}{e_k^2} \to \tfrac{1}{2} \frac{f''(x)}{f'(x)} \quad \text{as } x_k \to x.$$

By definition (3.8.1), this method is second order, or quadratically convergent.

This method actually must be provided with a good initial estimate to converge. This is shown by the following theorem.

Theorem 3.8.2

Assume that $f'(x) \neq 0$ and $f''(x)$ does not change sign in the interval $[a,b]$ and that $f(a).f(b)<0$. Then if,

$$\left|\frac{f(a)}{f'(a)}\right| < b - a \text{ and } \left|\frac{f(b)}{f'(b)}\right| < b - a ,$$

then the Newton Raphson method converges from an arbitrary initial point x_0 in the interval $[a,b]$.

This is illustrated in Figure (3.8.1), where the iteration process converges from any point $x_0 \in [a,b]$, it may as well diverge from some points $x_0' \in [a_1,b_1]$.

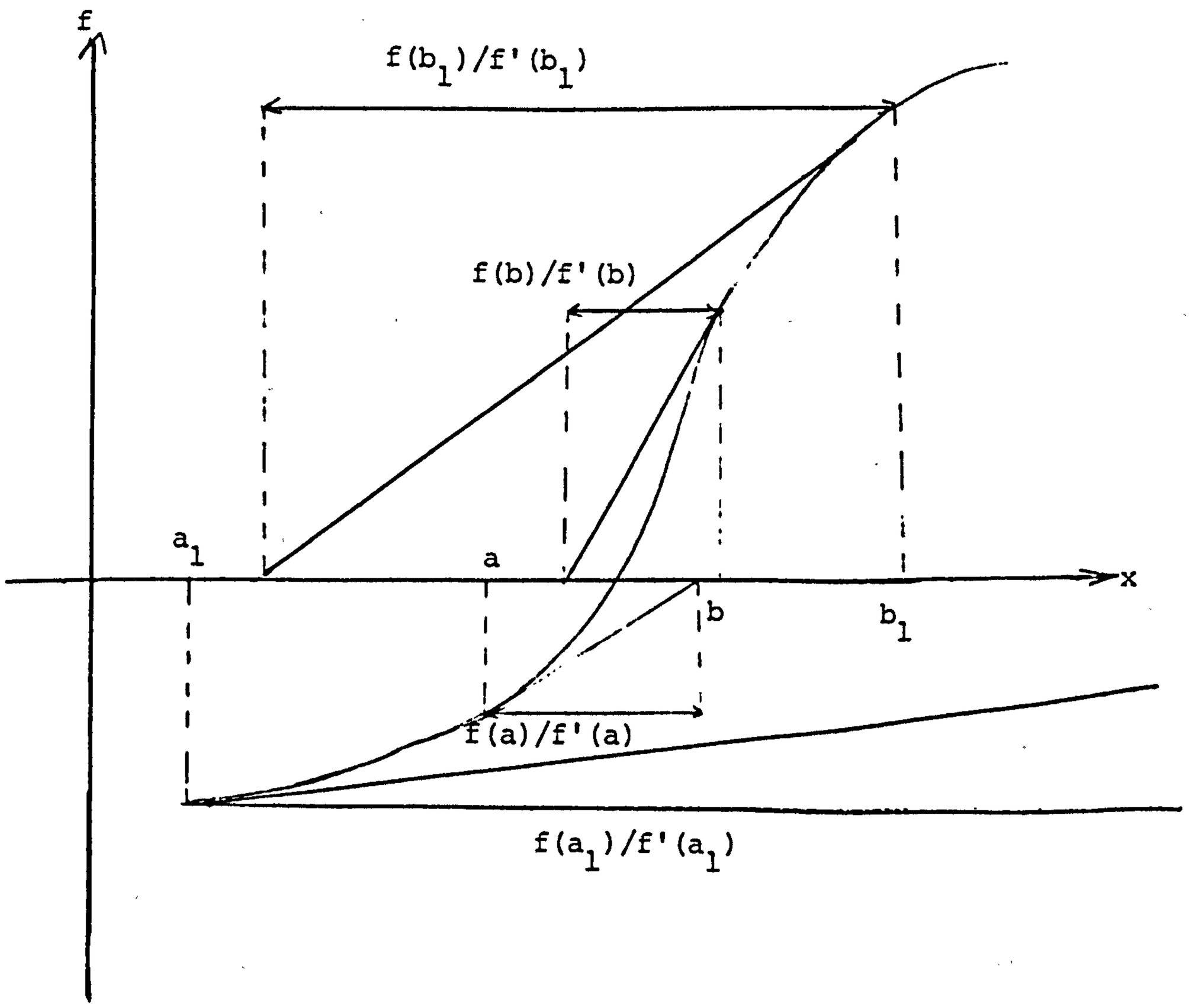


FIGURE 3.8.1

The Secant Method

This method is derived from the Newton Raphson method (3.8.12) by approximating the derivative $f'(x_k)$ by the difference quotient,

$$f'(x_k) \cong \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} \qquad (3.8.16)$$

which leads to the following analysis.

Given the initial approximations x_0 and x_1, the sequence $x_2, x_3, \ldots$ is computed by,

$$x_{k+1} = x_k + \delta_k \ ,$$

where $$\delta_k = -f(x_k) \frac{(x_k - x_{k-1})}{(f(x_k) - f(x_{k-1}))} \ , \ f(x_k) \neq f(x_{k-1}). \qquad (3.8.17)$$

The iteration function $g(x_k)$ for this method is,

$$g(x_k) = \delta_k + x_k \ .$$

As the evaluation of x_{k+1} requires the information of two points i.e. x_k and x_{k-1}, this type of method is called a two-points iterative formula.

The preference between this method and Newton Raphson's method depends on the amount of work required to compute f'(x). If the work required to compute f'(x) is more than 0.44 of the work required to compute a value of f(x), then the Secant method is recommended, otherwise Newton Raphson method is to be used. [DAHLQUIST, 1974, p.228].

For a system of non-linear equations, neglecting the second order term, equation (3.8.10) can be written in system notation as,

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + O(|x - x_k|)^2 \ , \qquad (3.8.18)$$

where $\underline{x}$ and f are vectors of dimension n and $f'(x_k)$ is the Jacobian

matrix denoted by J with elements

$$f'_{i,j}(x) = \frac{\partial f_i(x)}{\partial x_j} , \quad i,j=1,2,\ldots,n. \tag{3.8.19}$$

This leads to Newton Raphson's n dimensional method,

$$J(x_k)(x_{k+1}-x_k) + f(x_k) = 0 , \tag{3.8.20}$$

which is a linear system of equations for x_{k+1} and can be solved directly, if $J(x_k)$ is non-singular.

Although Newton's method is theoretically attractive it may be difficult to use in practice. In fact, each step requires the solution of the linear system (3.8.20) [the inverse of $J(x_k)$ is rarely computed explicitly] and especially for problems arising from partial differential equations in which the dimension of the system may be several thousand, this may be a difficult task. Moreover, at each step, not only n components of $J(x_k)$ but also the n^2 entries of $J'(x_k)$ are needed, and, unless the partial derivatives $\partial f_i/\partial x_j$ have a simple functional form, it may be desirable to avoid their computation altogether. [ORTEGA, J.M. & RHEINBOLDT, W.C., 1970, p.185].

Algorithmic Considerations of the Method

The algorithmic steps which can be applied to the system (3.8.20) are as follows:

1. Calculate $f(x_k)$ and $J(x_k)$
2. Evaluate x_{k+1} from the system (3.8.20)
3. Calculate $f(x_{k+1})$.

Now the following cases can arise:

a) If $||f(x_{k+1})|| << ||f(x_k)||$, we continue with the same Jacobian.

b) if $||f(x_{k+1})|| < ||f(x_k)||$ go to step 2.

c) if $||f(x_{k+1})|| > ||f(x_k)||$ take $x^*_{k+1} = x_k - \lambda_k w_k$, where $w_k = x_{k+1} - x_k$ is found by solving the system $J(x_k)(x_{k+1} - x_k) + f(x_k) = 0$, and $\lambda_k = \frac{1}{2^k}$ $k=1,...$ until a reduction in $||f(x_{k+1})||$ is obtained [WALSH, J., 1976].

As the calculation of the Jacobian matrix is too expensive for some non-linear systems, a functional iteration method which does not use the Jacobian matrix at all, or a method which replaces the Jacobian matrix by some approximation to it is recommended. This latter method is the Secant method for linear systems. In this method let,

$$\gamma_k = f(x_{k+1}) - f(x_k) \quad , \tag{3.8.21}$$

then from (3.8.18) and neglecting the error term, we get,

$$J(x_k)\delta_k = \gamma_k \quad , \tag{3.8.21}$$

where $\delta_k = x_{k+1} - x_k$.

If the Jacobian matrix $J(x_k)$ is replaced by some approximation B_k, we obtain,

$$B_{k+1}\delta_k = \gamma_k \quad , \tag{3.8.22}$$

where,

$$B_{k+1} = B_k - \frac{(B_k\delta_k - \gamma_k)q_k^T}{q_k^T\delta_k} \quad , \; q_k^T\delta_k \neq 0 \; .$$

with q_k is arbitrarily chosen (say, $q_k = \delta_k$).

ii) Linear Equations

Iterative methods are frequently used for large sparse linear systems as they make good use of this property. Also in using iterative methods, errors due to round-off may be damped out as the procedure continues. In fact, special iterative methods are frequently used to improve solutions obtained by direct methods.

Let the linear system be,

$$Ax = b \ , \tag{3.8.23}$$

where A is the coefficient matrix, x is the unkoxwn vector and b is the r.h.s. known vector.

For our purposes it will be found convenient to express the matrix A as the sum of its main diagonal, its strictly lower triangular and upper triangular elements, as in the matrix notation,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = D-L-U \ , \text{ where}$$

$$D = \begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & a_{33} & \\ & & & a_{44} \end{bmatrix} \ , \ -L = \begin{bmatrix} 0 & & & \\ a_{21} & 0 & & \\ a_{31} & a_{32} & 0 & \\ a_{41} & a_{42} & a_{43} & 0 \end{bmatrix}$$

and

$$-U = \begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} \\ & 0 & a_{23} & a_{24} \\ & & 0 & a_{34} \\ & & & 0 \end{bmatrix} \ .$$

Equation (3.8.23) can now be written as,

$$(D-L-U)x = b \ . \tag{3.8.24}$$

From this form in fact the Jacobi iteration formula is derived as,

$$Dx^{(k+1)} = (L+U)x^{(k)} + b \ , \tag{3.8.25}$$

giving,

$$x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b \ . \tag{3.8.26}$$

The matrix $D^{-1}(L+U)$ is called the Jacobi iteration matrix. Inspite

of its simplicity, it is seldom used because of its very slow convergence.

In the Jacobi method, one does not use the new values of $x_i^{(k+1)}$, $i=0,1,\ldots,n$ until all the components of the vectors $x^{(k+1)}$ have been evaluated. In the Gauss-Seidel method the values of $x_j^{(k+1)}$, $j=0,1,\ldots,i-1$ that have already been calculated are to be used in evaluating $x_i^{(k+1)}$. This formula is written as,

$$Dx^{(k+1)} = Lx^{(k+1)} + Ux^{(k)} + b \ , \tag{3.8.27}$$

leading to,

$$x^{(k+1)} = (D-L)^{-1}Ux^{(k)} + (D-L)^{-1}b \ , \tag{3.8.28}$$

and the matrix $(D-L)^{-1}U$ is called the Gauss-Seidel iteration matrix.

The Gauss-Seidel method can be modified by introducing the parameter ω such that,

$$(x_i^{(k+1)} - x_i^{(k)}) = \omega(\tilde{x}_i^{(k+1)} - x_i^{(k)}) \ , \tag{3.8.29}$$

where $\tilde{x}_i^{(k+1)}$ is the new value obtained from (3.8.28), and ω is called the over (or under) relaxation factor if > (or <) 1. From equation (3.8.27) we have that,

$$D(\tilde{x}^{(k+1)} - x^{(k)}) = Lx^{(k+1)} + Ux^{(k)} + b - Dx^{(k)} \ ,$$

therefore,

$$D(x^{(k+1)} - x^{(k)}) = \omega(Lx^{(k+1)} + Ux^{(k)} + b - Dx^{(k)}) \ ,$$

or

$$x^{(k+1)} - x^{(k)} = \omega D^{-1}(Lx^{(k+1)} + Ux^{(k)} + b - Dx^{(k)}) \ ,$$

hence,

$$(I-\omega D^{-1}L)x^{(k+1)} = \{(1-\omega)I + \omega D^{-1}U\}x^{(k)} + \omega D^{-1}b, \tag{3.8.30}$$

which gives,

$$x^{(k+1)} = (I-\omega D^{-1}L)^{-1}\{(1-\omega)I + \omega D^{-1}U\}x^{(k)} + (I-\omega D^{-1}L)^{-1}\omega D^{-1}b. \tag{3.8.31}$$

This method is known as the Successive Over-relaxation (Under relaxation) method corresponding to the range of $1<\omega<2$ ($\omega<1$), abbreviated as S.O.R. and the matrix $(I-\omega D^{-1}L)^{-1}\{(1-\omega)I+\omega D^{-1}U\}$ is called the S.O.R. iteration matrix.

This modification on the Gauss-Seidel method will allow us to choose ω so that the solution converges faster, and an optimum convergence rate can be achieved if the optimal ω is chosen. It is noticeable that when $\omega=1$, the method reduces to the Gauss-Seidel method.

Each of the three methods described above can be written as,

$$x^{(k+1)} = Gx^{(k)} + c \ , \tag{3.8.32}$$

where G is the iteration matrix and c is the vector of known values. This equation was derived from the original equation,

$$x = Gx + c \ , \tag{3.8.33}$$

where x is the unique solution of the linear system Ax=b.

Theorem 3.8.2

A necessary and sufficient condition for a stationary iterative method, (3.8.32) to converge from an arbitrary initial approximation x_0 is that

$$\rho(G) = \max_{0\leq i\leq n} |\lambda_i(G)| < 1 \ , \tag{3.8.34}$$

where $\rho(G)$ is the spectral radius of G.

For proof see Section 2.5.

Theorem 3.8.3

Let A be a symmetric matrix with positive diagonal elements. Then the S.O.R. method converges if and only if A is positive definite and $0<\omega<2$.

Proof: See Young, D.M., 1971, p.113.

Remarks:

1. Often positive definite matrices are produced in the finite difference approximation of partial differential equations.
2. In the practical use of S.O.R. ω is usually chosen to be $1 \leq \omega < 2$.
3. The optimal (best value of) ω, denoted as ω_{opt}, for maximum rate of convergence is given by [YOUNG, D.M., 1971, p.169],

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\mu^2}} \quad , \tag{3.8.35}$$

where μ is the spectral radius of the Jacobi iteration matrix $D^{-1}(L+U)$.

3.9 NUMERICAL METHODS FOR THE SOLUTION OF TWO DIMENSIONAL PROBLEMS

Numerical methods for solving problems in one space dimension, can usually be extended to solve two dimensional problems. This includes explicit, implicit, direct and iterative methods.

For the heat flow problem in two dimensions, let us consider the equation,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} \quad , \tag{3.9.1}$$

in the (x,y) plane where $0 \leq x \leq n$, $0 \leq y \leq m$ and $t > 0$, with the initial condition,

$$U(x,y,0) = f(x,y) \quad , \tag{3.9.1a}$$

and boundary conditions,

$$\begin{aligned} U(x,0,t) &= g_1(x,t) \quad , \\ U(x,m,t) &= g_2(x,t) \quad , \\ U(0,y,t) &= g_3(y,t) \quad , \end{aligned} \tag{3.9.1b}$$

and $$U(n,y,t) = g_4(y,t) \quad .$$

One way of approximating equation (3.9.1) explicitly is by the extension of the classical explicit equation (3.2.6), which leads to the formula,

$$u_{i,j,k+1} = u_{i,j,k} + \frac{\Delta t}{(\Delta x)^2} \delta_x^2(u_{i,j,k}) + \frac{\Delta t}{(\Delta y)^2} \delta_y^2(u_{i,j,k}), \tag{3.9.2}$$

where the central difference operator δ^2 is defined as,

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

To determine the stability of (3.9.2) we use the Von Neumann analysis by assuming an error of the form,

$$\varepsilon_{i,j,k} = \xi^k e^{\hat{i}\beta_1 i\Delta x} \, e^{\hat{i}\beta_2 j\Delta y} \quad . \tag{3.9.3}$$

By substitution of (3.9.3) into (3.9.2) gives that,

$$\gamma = \frac{\xi^{k+1}}{\xi^{k}}$$

$$= 1-4(r_1 \sin^2(\beta_1 \frac{\Delta x}{2}) - r_2 \sin^2(\beta_2 \frac{\Delta y}{2})) , \quad (3.9.4)$$

where $r_1 = \frac{\Delta t}{(\Delta x)^2}$, $r_2 = \frac{\Delta t}{(\Delta y)^2}$ and γ is the amplification factor.

For the stability of (3.9.2) we require that $|\gamma| \leq 1$, i.e.

$-1 \leq 1-4(r_1 \sin^2(\beta_1 \frac{\Delta x}{2}) - r_2 \sin^2(\beta_2 \frac{\Delta y}{2})) \leq 1$.

The right handside inequality is neglected as trivial, so we are left with $-2 \leq -4(r_1 \sin^2(\beta_1 \frac{\Delta x}{2}) - r_2 \sin^2(\beta_2 \frac{\Delta y}{2}))$, which results in the relation,

$$r_1 + r_2 \leq \tfrac{1}{2} . \quad (3.9.5)$$

Thus, for a square mesh $\Delta x = \Delta y$ and $r_1 = r_2$ (=r say) we have,

$$r \leq 1/4 , \quad (3.9.6)$$

which is more restrictive than that for the one dimensional problem. The local truncation error for the formula (3.9.2) can be derived by using Taylor's series expansion and is given as follows:

$$T_{i,j,k} = u_{i,j,k+1} - u_{i,j,k} - \frac{\Delta t}{\Delta x^2}\delta_x^2(u_{i,j,k}) - \frac{\Delta t}{\Delta y^2}\delta_y^2(u_{i,j,k})$$

$$= (\frac{\partial U}{\partial t})_{i,j,k} - (\frac{\partial^2 U}{\partial x^2})_{i,j,k} - (\frac{\partial^2 U}{\partial y^2})_{i,j,k}$$

$$+ \tfrac{1}{2}\Delta t\,(\frac{\partial^2 U}{\partial t^2})_{i,j,k} - \frac{(\Delta x)^2}{12}(\frac{\partial^4 U}{\partial x^4})_{i,j,k} - \frac{(\Delta y)^2}{12}(\frac{\partial^4 U}{\partial y^4})_{i,j,k} + \cdots$$

(3.9.7)

Hence,

$$T = O(\Delta t) + O((\Delta x)^2 + (\Delta y)^2) . \quad (3.9.8)$$

For unconditional stability then logically we turn to the implicit formula (3.2.7) which when generalized to two dimensional gives,

$$\frac{u_{i,j,k+1}-u_{i,j,k}}{\Delta t} = \frac{\delta_x^2}{(\Delta x)^2}(u_{i,j,k+1}) + \frac{\delta_y^2}{(\Delta y)^2}(u_{i,j,k+1}). \quad (3.9.9)$$

Using the same stability analysis as above we find that,

$$\gamma = \frac{1}{1+4r_1\sin^2(\beta_1 \frac{\Delta x}{2})+4r_2\sin^2(\beta_2 \frac{\Delta y}{2})} , \quad (3.9.10)$$

which is less than unity for any r_1, r_2. The local truncation error is (not surprisingly) again of the form,

$$T = O(\Delta t + (\Delta x)^2 + (\Delta y)^2) . \quad (3.9.11)$$

The Crank-Nicolson two dimensional version of equation (3.2.8) is given as,

$$\frac{u_{i,j,k+1}-u_{i,j,k}}{\Delta t} = \tfrac{1}{2}[\frac{\delta_x^2}{(\Delta x)^2}(u_{i,j,k+1}) + \frac{\delta_x^2}{(\Delta x)^2}(u_{i,j,k}) + \frac{\delta_y^2}{(\Delta y)^2}(u_{i,j,k+1}) + \frac{\delta_y^2}{(\Delta y)^2}(u_{i,j,k})] . \quad (3.9.12)$$

For the stability of this formula using the same analysis we have,

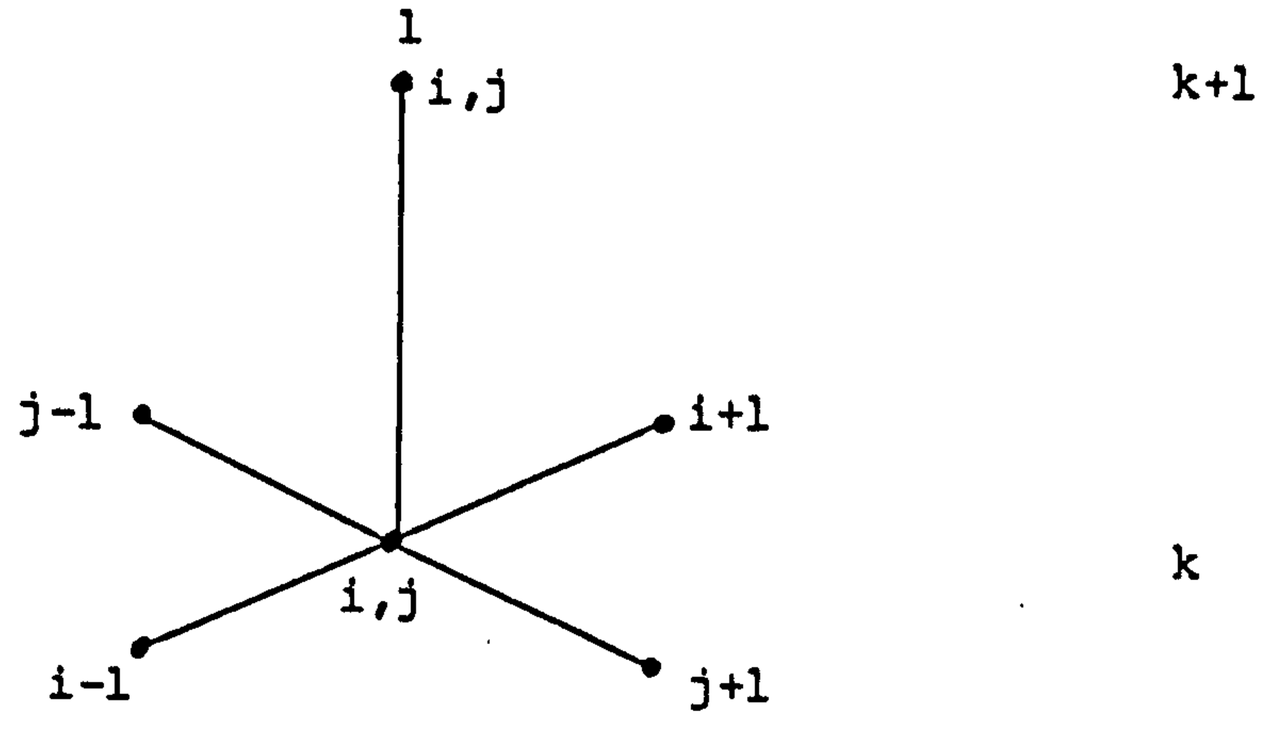
$$\gamma = \frac{1-2r_1\sin^2(\beta_1 \frac{\Delta x}{2})-2r_2\sin^2(\beta_2 \frac{\Delta y}{2})}{1+2r_1\sin^2(\beta_1 \frac{\Delta x}{2})+2r_2\sin^2(\beta_2 \frac{\Delta y}{2})} , \quad (3.9.13)$$

which is always less than unity in magnitude. Hence the Crank Nicolson formula is unconditionally stable for $r_1, r_2 > 0$. However its local truncation error is of the form,

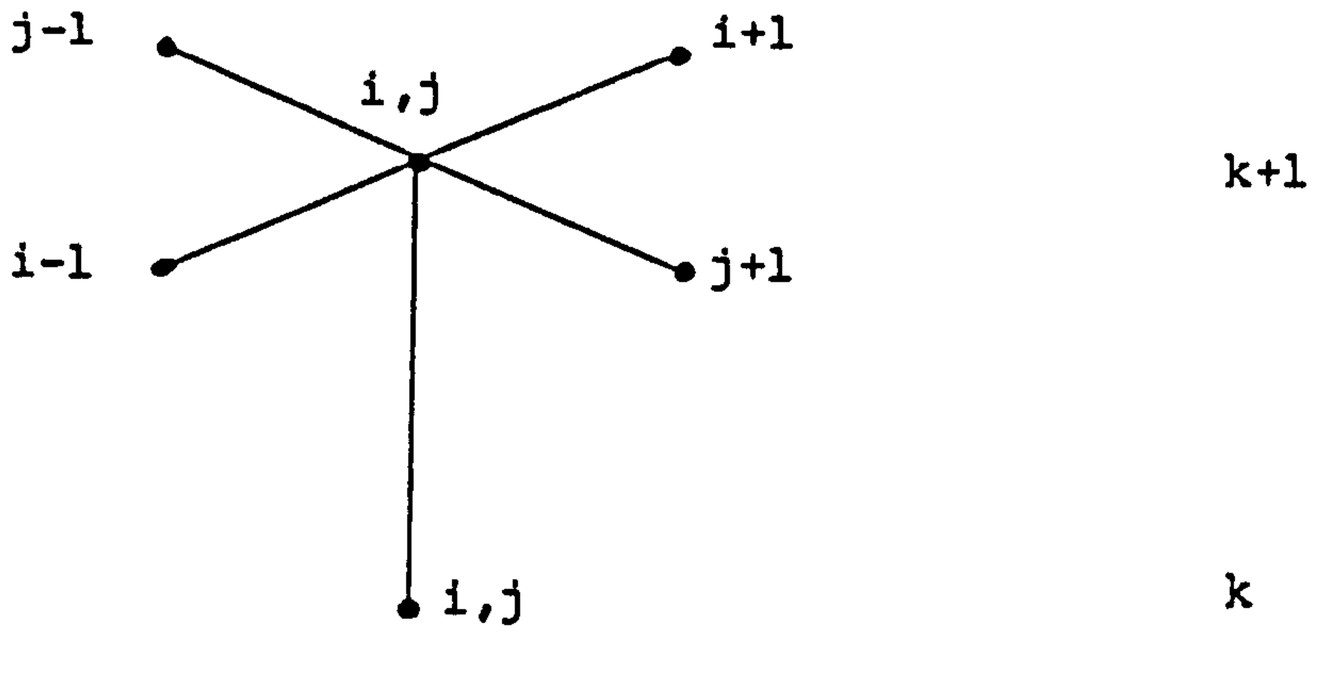
$$T = O((\Delta t)^2+(\Delta x)^2+(\Delta y)^2) . \quad (3.9.14)$$

All of the schemes presented above are shown in molecular form by Figure 3.9.1.

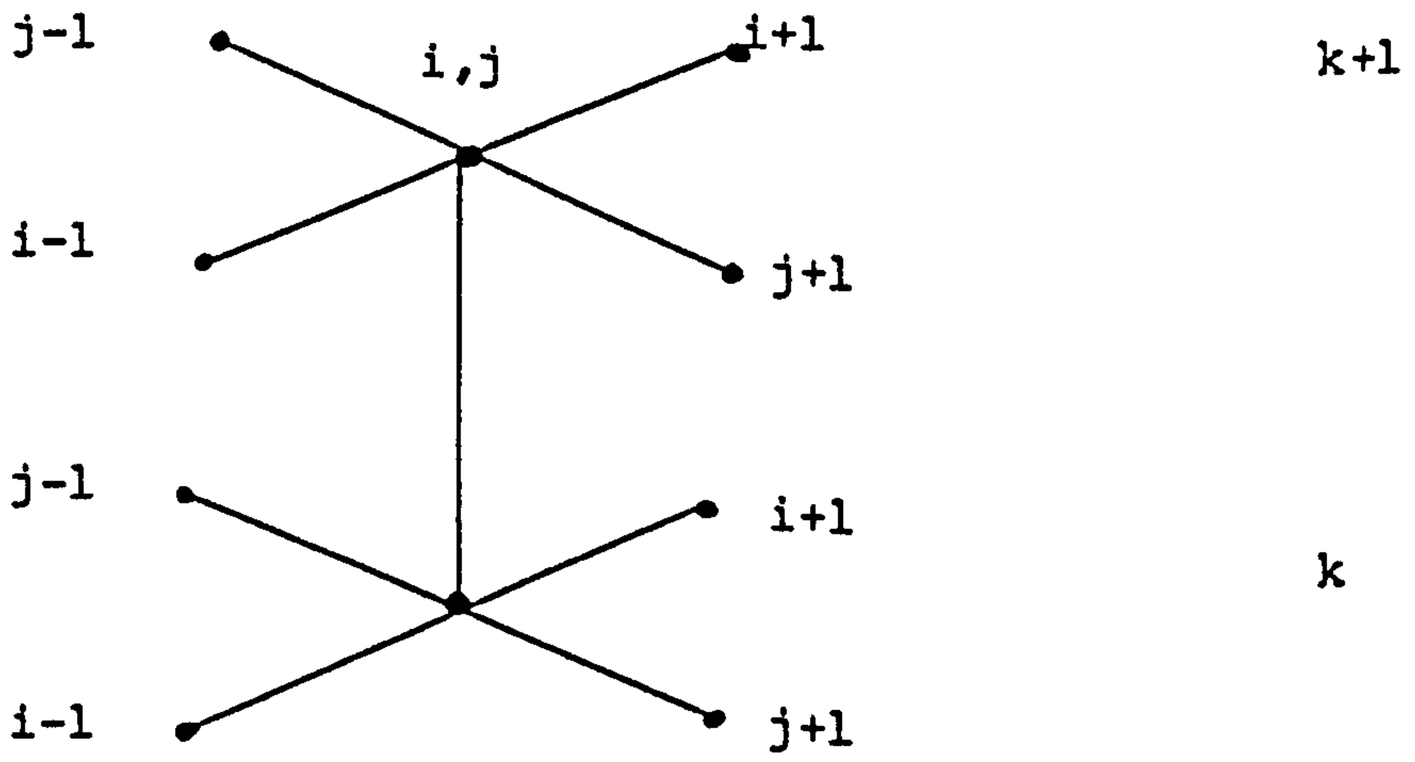
Although the implicit equations (3.9.9) and (3.9.12) are unconditionally stable, they are much more difficult to solve than



(a) Explicit



(b) Implicit



(c) Crank-Nicolson

FIGURE 3.9.1

their one dimensional equivalents. They require, for each time step, the solution of ($nm^{\times}nm$) linear systems, and as each equation of (3.9.9) or (3.9.12) involves five unknowns the system is no longer tridiagonal. This requires a great amount of computational work. Therefore a method which maintains simplicity, reduces computational work and possesses a good stability property was needed. This was the Alternating Direction Implicit (A.D.I.) class of methods.

3.10 THE ALTERNATING DIRECTION IMPLICIT METHOD (A.D.I.)

This method is especially designed to solve problems in more than one spatial dimension. There is no one dimensional analogue scheme which has been developed yet.

In this method the solution is obtained after two successive sweeps each time step. These two sweeps are in different directions, if the first sweep is in the x direction, then the second one is in the y direction.

This is done by approximating only one of the second derivatives (say $\frac{\partial^2 U}{\partial x^2}$) of equation (3.9.1) in terms of the unknown (new) values of U, while the other second derivative ($\frac{\partial^2 U}{\partial y^2}$) is approximated in terms of the known (old) values of U, that is for the first sweep. For the second sweep, the opposite thing happens i.e., ($\frac{\partial^2 U}{\partial y^2}$) is to be approximated in terms of the unknown values of U, while the other second derivative ($\frac{\partial^2 U}{\partial x^2}$) is approximated in terms of the known values of U. So for the 1st sweep, we have,

$$\frac{u_{i,j,k+\frac{1}{2}} - u_{i,j,k}}{\Delta t/2} = \frac{\delta_x^2}{(\Delta x)^2} u_{i,j,k+\frac{1}{2}} + \frac{\delta_y^2}{(\Delta y)^2} u_{i,j,k}, \qquad (3.10.1a)$$

and for the 2nd sweep,

$$\frac{u_{i,j,k+1} - u_{i,j,k+\frac{1}{2}}}{\Delta t/2} = \frac{\delta_x^2}{(\Delta x)^2} u_{i,j,k+\frac{1}{2}} + \frac{\delta_y^2}{(\Delta y)^2} u_{i,j,k+1}. \qquad (3.10.1b)$$

This results in tridiagonal systems which can be solved directly using (say) the Thomas algorithm.

The method above was introduced by Peaceman and Rachford in 1955. They show that using only one single equation of (3.10.1) leads to an unstable scheme and that (3.10.1a) and (3.10.1b) have to be used

alternatively for stability. Also it is necessary in the 2 sweeps that Δt is the same for each particular step. This can be shown by using the Von Neumann analysis for equation (3.10.1) which gives for each sweep,

$$\gamma = \frac{\xi^{k+\frac{1}{2}}}{\xi^{k}} = \frac{1-4r\sin^2(\beta_2\ \Delta y/2)}{1+4r\sin^2(\beta_1\ \Delta x/2)} \quad , \tag{3.10.2a}$$

and

$$\gamma = \frac{\xi^{k+1}}{\xi^{k+1/2}} = \frac{1-4r\sin^2(\beta_1\ \Delta x/2)}{1+4r\sin^2(\beta_2\ \Delta y/2)} \quad , \tag{3.10.2b}$$

respectively, where $r = \frac{\Delta t}{2(\Delta x)^2} = \frac{\Delta t}{2(\Delta y)^2}$ for simplicity. For some values of β_1, β_2 and r any of the above two ratios has an absolute value considerably greater than unity. However the overall stability ratio for the two sweeps is given by,

$$\gamma = \frac{\xi^{k+1}}{\xi^{k}} = \frac{1-4r\sin^2(\beta_1\ \Delta x/2)}{1+4r\sin^2(\beta_1\ \Delta x/2)} \quad \frac{1-4r\sin^2(\beta_2\ \Delta y/2)}{1+4r\sin^2(\beta_2\ \Delta y/2)} \quad , \tag{3.10.3}$$

which has an absolute value less than unity for all β_1, β_2 and r.

Equation (3.10.1) can be written in implicit form as,

$$(1-r/2\ \delta_x^2)u_{i,j,k+\frac{1}{2}} = (1+r/2\ \delta_y^2)u_{i,j,k} \quad , \tag{3.10.4a}$$

and

$$(1-r/2\ \delta_y^2)u_{i,j,k+1} = (1+r/2\ \delta_x^2)u_{i,j,k+\frac{1}{2}} \quad . \tag{3.10.4b}$$

Eliminating the intermediate values of $u_{i,j,k+\frac{1}{2}}$ in these 2 equations gives,

$$(1-r/2\ \delta_y^2)u_{i,j,k+1} = (1+r/2\ \delta_x^2)(1-r/2\ \delta_x^2)^{-1}(1+r/2\ \delta_y^2)u_{i,j,k} \quad . \tag{3.10.5}$$

This formula can be shown to have a local truncation error of order,

$$T = O((\Delta t)^2 + (\Delta x)^2 + (\Delta y)^2) \quad , \tag{3.10.6}$$

which is similar to Crank-Nicolson's [DOUGLAS, J., 1961].

3.11 ITERATION BY A.D.I. METHOD

We have seen in the previous section that the A.D.I. method enables us to solve directly the resulting tridiagonal systems of the heat flow linear equations.

The A.D.I. method may also be used to iterate to the solution of Laplace's equation. In this case, each stage of iteration may be regarded as a time step of an unsteady-state problem, while the starting values used for the first iteration correspond to the initial condition.

Consider the Laplace's equation,

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \ , \qquad (3.11.1)$$

defined on a square domain with Dirichlet boundary conditions. This can be represented in terms of the five point finite difference formula,

$$(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + (u_{i,j-1} - 2u_{i,j} + u_{i,j+1}) = 0 \ ,$$

$$i,j = 1,2,\ldots,n, \qquad (3.11.2)$$

where it is assumed that $\Delta x = \Delta y$. In matrix form this is written,

$$Au = b \ , \qquad (3.11.3)$$

where b is the vector associated with the boundary conditions, A is a $n^2 \times n^2$ matrix of coefficient and u is the vector of unknowns. The matrix A can be split into component form as $A = H + V + \Sigma$, where H and V are tridiagonal symmetric and positive definite matrices, and the matrix Σ is a non-negative diagonal matrix. Thus, equation (3.11.3) can be written as,

$$(H + V + \Sigma)u = b \ . \qquad (3.11.4)$$

By ordering the mesh points by rows, one can make H tridiagonal whilst by ordering them by columns, one can make V tridiagonal. However one cannot in general make them both tridiagonal simultaneously.

Equation (3.11.4) is equivalent to the forms,

$$(H_1+\rho I)u = b-(V_1-\rho I)u \ , \qquad (3.11.5)$$

or $$(V_1+\rho I)u = b-(H_1-\rho I)u \ ,$$

where $H_1=H+\frac{1}{2}\Sigma$, $V_1=V+\frac{1}{2}\Sigma$, ρ is scalar and I is the identity matrix. This suggests the iterative method,

$$(H_1+\rho_k I)u^{(k+\frac{1}{2})} = b-(V_1-\rho_k I)u^{(k)}, \qquad (3.11.6a)$$

$$(V_1+\rho_k I)u^{(k+1)} = b-(H_1-\rho_k I)u^{(k+\frac{1}{2})}, \ k=0,1,\ldots, \qquad (3.11.6b)$$

where k is an iteration index. The scalar ρ is called the acceleration parameter, and its value is chosen to maximise the rate of convergence. It is obvious that $u^{(0)}$ represents a starting approximation. We shall bear in mind that in equation (3.11.6a), H_1 is a tridiagonal matrix and in equation (3.11.6b), V_1 is also tridiagonal. In fact this alteration must occur in each cycle of double iterations (two sweeps).

For the convergence of the method of A.D.I., from equation (3.11.6) we have,

$$u^{(k+\frac{1}{2})} = (H_1+\rho_k I)^{-1}b-(H_1+\rho_k I)^{-1}(V_1-\rho_k I)u^{(k)}, \qquad (3.11.7)$$

$$u^{(k+1)} = (V_1+\rho_k I)^{-1}b-(V_1+\rho_k I)^{-1}(H_1-\rho_k I)u^{(k+\frac{1}{2})}, \qquad (3.11.8)$$

and this leads to,

$$u^{(k+1)} = (V_1+\rho_k I)^{-1}(H_1-\rho_k I)(H_1+\rho_k I)^{-1}(V_1-\rho_k I)u^{(k)}+S, \qquad (3.11.9)$$

or $$u^{(k+1)} = G_\rho u^{(k)}+S \ ,$$

where,

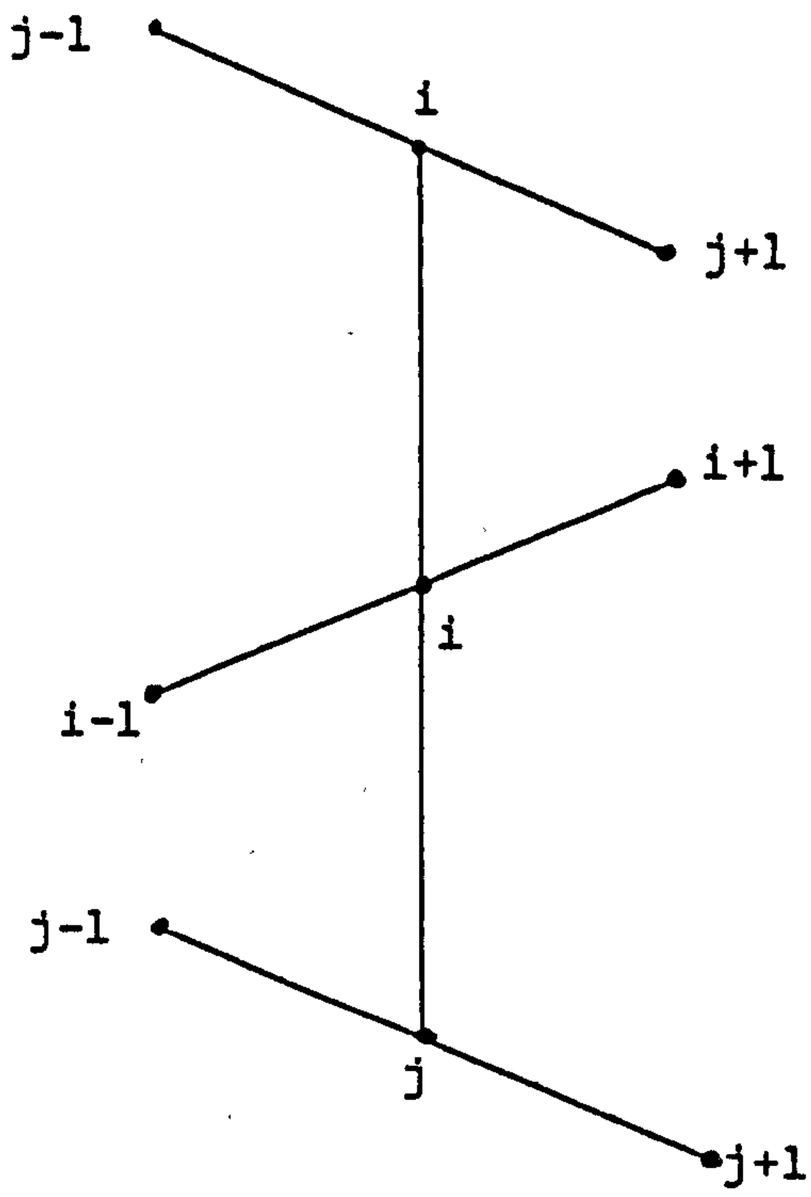


FIGURE 3.11.2: The A.D.I. method molecular diagram for one time step

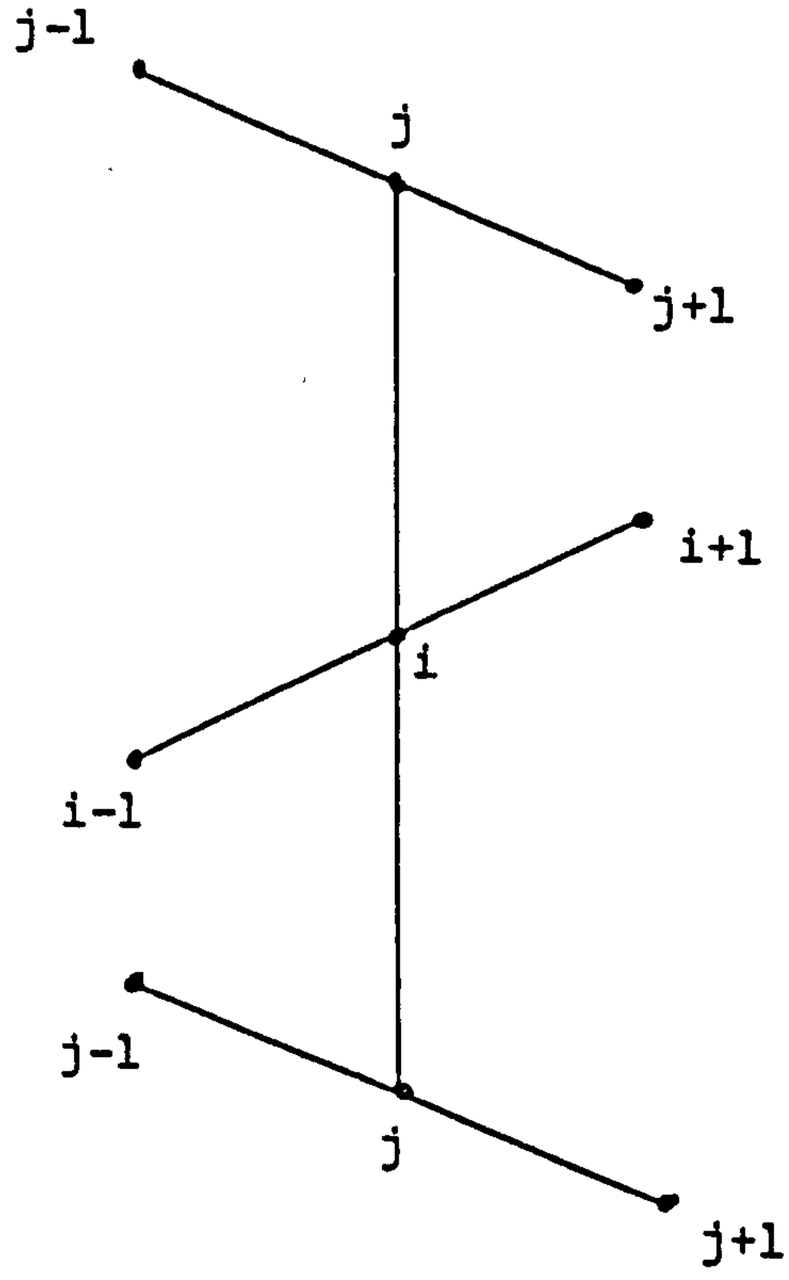


FIGURE 3.11.3: The A.D.I. iterative method molecular diagram for one iteration

$$G_\rho = (V_1+\rho_k I)^{-1}(H_1-\rho_k I)(H_1+\rho_k I)^{-1}(V_1-\rho_k I) , \qquad (3.11.10)$$

is called the iteration matrix, and,

$$S = (V_1+\rho_k I)^{-1}\{I+(H_1-\rho_k I)(H_1+\rho_k I)^{-1}\}\underline{b} . \qquad (3.11.11)$$

If we denote the error vector by e then $e^{(k)}=u^{(k)}-u$ and $e^{(k+1)}=G_{\rho_k} e^{(k)}$, in general after k iterations,

$$\underline{e}^{(k)} = (\prod_{j=1}^{k} G_{\rho_j})e^{(0)}, \quad K\geq 1 , \qquad (3.12.11)$$

where,

$$\prod_{j=1}^{k} G_{\rho_j} \equiv G_{\rho_k}.G_{\rho_{k-1}} \ldots G_{\rho_1} . \qquad (3.11.13)$$

Now for the stationary case with $\rho_j=\rho$, the matrix $\bar{G}_\rho$ defined as,

$$\bar{G}_\rho = (V_1+\rho I)G_\rho(V_1+\rho I)^{-1} , \qquad (3.11.14)$$

is similar to G_ρ, and thus has the same eigenvalues as G_ρ. With (3.11.10) it follows that,

$$\bar{G}_\rho = (H_1-\rho I)(H_1+\rho I)^{-1}(V_1-\rho I)(V_1+\rho I)^{-1} . \qquad (3.11.15)$$

Therefore,

$$\rho(G_\rho) = \rho(\bar{G}_\rho) \leq ||\bar{G}_\rho|| \leq ||(H_1-\rho I)(H_1+\rho I)^{-1}||.||(V_1-\rho I)(V_1+\rho I)^{-1}|| ,$$

where $\rho(G_\rho)$ is the spectral radius of G_ρ. Since H_1 and V_1 are symmetric and positive definite, in the L_2 norm, we get that,

$$||(H_1-\rho I)(H_1+\rho I)^{-1}|| = \max_{1\leq i\leq n} \left|\frac{\lambda_i-\rho}{\lambda_i+\rho}\right| < 1 , \qquad (3.11.16)$$

where λ_i, i=1,...,n are the eigenvalues of H_1 and they are positive.

Similarly for the corresponding matrix product with V_1 shows that

$$||(V_1-\rho I)(V_1+\rho I)^{-1}|| = \max_{1\leq i\leq n} \left|\frac{\mu_i-\rho}{\mu_i+\rho}\right| < 1 , \qquad (3.11.17)$$

where μ_i, i=1,...,n are the eigenvalues of V_1 (positive). Hence $\rho(G_\rho)<1$ for all $\rho>0$, which proves that the Peaceman Rachford (P.R.) iterative method converges.

It is shown in Varga (1962) that,

$$\min_{\rho>0} \rho(G_\rho) = \min \rho(G_{\omega_{opt}}) , \qquad (3.11.18)$$

where $G_{\omega_{opt}}$ is the iteration matrix of the point S.O.R. method, this concludes that the P.R. method and point S.O.R. method have the same rate of convergence for the model problem. However, the A.D.I. iterative method involves a good deal more arithmetic work than the point successive overrelaxation iterative method, in fact using a sequence of acceleration parameters $\{\rho_i\}$ is essential to recover this weakness in convergence rate, and this is known as the Non-Stationary case.

For our model problem (3.11.1) it can be shown that H_1 and V_1 commute, i.e. $H_1V_1=V_1H_1$, this implies, as the theorem of Frobenius states below, that H_1 and V_1 possessa common basis of orthogonal vectors x_i, i=1,...,n.

Theorem 3.11.1

Let H_1 and V_1 be Hermitian n×n matrices. Then, there exists an orthogonal basis of eigenvectors $\{x_i\}_{i=1}^n$ with $H_1x_i=\lambda_ix_i$ and $V_1x_i=\mu_ix_i$ for $1\leq i\leq n$ if and only if $H_1V_1=V_1H_1$. For proof see [VARGA, S. 1962, p.220].

Now consider the A.D.I. iteration matrix G_ρ. For κ iterations of the method, it is clear that,

$$\left(\prod_{j=1}^{k} G_{\rho_j}\right)x_i = \left\{\prod_{j=1}^{k} \left(\frac{\mu_i-\rho_j}{\mu_i+\rho_j}\right)\left(\frac{\lambda_i-\rho_j}{\lambda_i+\rho_j}\right)\right\}x_i \quad , \quad 1\leq i\leq n.$$

Using the $n\times n$ unitary matrix U generated by the vectors x_i, we see that $U\left(\prod_{j=1}^{k} G_{\rho_j}\right)U^*$ is a real diagonal matrix. Therefore,

$$\left\|\prod_{j=1}^{k} G_{\rho_j}\right\| = \rho\left(\prod_{j=1}^{k} G_{\rho_j}\right) = \max_{1\leq i\leq n} \prod_{j=1}^{k} \left|\frac{\mu_i-\rho_j}{\mu_i+\rho_j}\right|\left|\frac{\lambda_i-\rho_j}{\lambda_i+\rho_j}\right| < 1. \tag{3.11.19}$$

We notice that if all the eigenvalues of either of the matrices H_1 or V_1 are known a priori, then we could choose a sequence of positive real numbers $\{\xi_j\}_{j=1}^{k}$, so that,

$$\left\|\prod_{j=1}^{k} G_{\xi_j}\right\| = 0 \ ,$$

and we could have a direct method instead of iterative method. Since this is not an easy task, we assume that we can estimate a lower bound α and an upper bound β of the eigenvalues λ_i and μ_i of H_1 and V_1, i.e.,

$$0 < \alpha \leq \lambda_i, \ \mu_i \leq \beta \quad , \quad 1\leq i\leq n \ . \tag{3.11.20}$$

Then clearly,

$$\max_{1\leq i\leq n} \prod_{j=1}^{k} \left|\frac{\lambda_i-\rho_j}{\lambda_i+\rho_j}\right|\left|\frac{\mu_i-\rho_j}{\mu_i+\rho_j}\right| \leq \left\{\max_{1\leq i\leq n} \prod_{j=1}^{k}\left|\frac{\lambda_i-\rho_j}{\lambda_i+\rho_j}\right|\right\}\left\{\max_{1\leq i\leq n} \prod_{j=1}^{k}\left|\frac{\mu_i-\rho_j}{\mu_i+\rho_j}\right|\right\}$$

$$\leq \max_{\alpha\leq\gamma\leq\beta} \prod_{j=1}^{k} \left|\frac{\gamma-\rho_j}{\gamma+\rho_j}\right| \Bigg\}^2 \quad , \tag{3.11.21}$$

which leads to,

$$\left\|\prod_{j=1}^{k} G_{\rho_j}\right\| \leq \{\max_{\alpha\leq\gamma\leq\beta} |\phi(\gamma;\rho_j)|\}^2 \ , \tag{3.11.22}$$

where

$$\phi(\gamma,\rho_j) \equiv \prod_{j=1}^{k} \left(\frac{\gamma-\rho_j}{\gamma+\rho_j}\right) . \qquad (3.11.23)$$

The problem of minimizing $\phi(\gamma,\rho_j)$ has been solved analytically in terms of elliptic functions by Wachspress (1966) and is given by,

$$\rho_j^w = \beta\left(\frac{\alpha}{\beta}\right)^{(j-1)/(k-1)} , \quad k\geq 2 , \; j=1,2,\ldots,k. \qquad (3.11.24)$$

Peaceman and Rachford (1955) used a different set of parameters given by the expression,

$$\rho_j^p = \beta\left(\frac{\alpha}{\beta}\right)^{(2j-1)/(2k)} , \; j=1,2,\ldots,k . \qquad (3.11.25)$$

With this set of parameters it can be shown that,

$$\left|\left|\prod_{j=1}^{k} G_{\rho_j}\right|\right| \leq \left\{\frac{1-\left(\frac{\alpha}{\beta}\right)^{1/(2k)}}{1+\left(\frac{\alpha}{\beta}\right)^{1/(2k)}}\right\}^2 . \qquad (3.11.26)$$

3.12 VARIANTS OF THE A.D.I. AND OTHER FORMS OF SCHEMES

Many variants of the P.R. scheme have been proposed. For example, a modification on equation (3.11.6b), where from (3.11.6a) we get,

$$H_1 u^{(k+\frac{1}{2})} = b-(V_1-\rho_k I)u^{(k)}-\rho_k u^{(k+\frac{1}{2})} , \qquad (3.12.1)$$

writing (3.11.6b) as,

$$(V_1+\rho_k I)u^{(k+1)} = b-H_1 u^{(k+\frac{1}{2})}+\rho_k u^{(k+\frac{1}{2})} , \qquad (3.12.2)$$

and substituting $H_1 u^{(k+\frac{1}{2})}$ by (3.12.1) to give,

$$(V_1+\rho_k I)u^{(k+1)} = (V_1-\rho I)u^{(k)}+2\rho u^{(k+\frac{1}{2})} , \qquad (3.12.3)$$

and by introducing a weighting parameter ω we get,

$$(V_1+\rho_k I)u^{(k+1)} = (V_1-(1-\omega)\rho I)u^{(k)}+(2-\omega)\rho u^{(k+\frac{1}{2})} , \qquad (3.12.4)$$

which gives various schemes as ω varies, where for $\omega=0$, it is obviously the basic P.R. formula (3.11.6) and for $\omega=1$, it produces a formula due to Douglas and Rachford (1956). For H_1 and V_1 having the same properties as the P.R. scheme and for fixed acceleration parameter $\rho>0$, the above general A.D.I. scheme is convergent for $0\leq\omega\leq2$.

Apart from these variant schemes, several other alternating direction schemes have been considered for parabolic equations. We mention here the Locally One Dimensional method (L.O.D.) which was developed by the Russian numerical analysts, in particular D'Yakonov, Marchuk, Samarskii and Yanenko. They considered the equation,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} , \qquad (3.12.5)$$

which is to be written as the pair,

$$\tfrac{1}{2}\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} , \qquad (3.12.6)$$

and $$\tfrac{1}{2}\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial y^2} \quad . \tag{3.12.7}$$

This assumption is valid when the partial differential equation is termed separable.

For the solution to advance from the k time level to the (k+1) time level it is assumed that (3.12.6) holds from k to k+½ and (3.12.7) holds from k+½ to k+1. The discretisation of (3.12.6) and (3.12.7) are,

$$u_{i,j}^{k+\frac{1}{2}} = ru_{i-1,j}^{k} + (1-2r)u_{i,j}^{k} + ru_{i+1,j}^{k} \tag{3.12.8}$$

and,

$$u_{i,j}^{k+1} = ru_{i,j-1}^{k+\frac{1}{2}} + (1-2r)u_{i,j}^{k+\frac{1}{2}} + ru_{i,j+1}^{k+\frac{1}{2}} \quad . \tag{3.12.9}$$

The elimination of $u^{k+\frac{1}{2}}$ leads to the nine point formula,

$$u^{k+1} = (1+r\delta_x^2)(1+r\delta_y^2)u^{K} \quad , \tag{3.12.10}$$

which has a truncation error of $O([\Delta t]^2 + t[\Delta x]^2 + t[\Delta y]^2)$.

Although both the L.O.D. method and formula (3.12.10) have the same accuracy and stability properties, it is more economical to use the L.O.D. method, because in using the formula (3.12.10) nine function evaluations are required for the calculation of u^{k+1} at each grid point, while in using L.O.D. method, only six function evaluations are required.

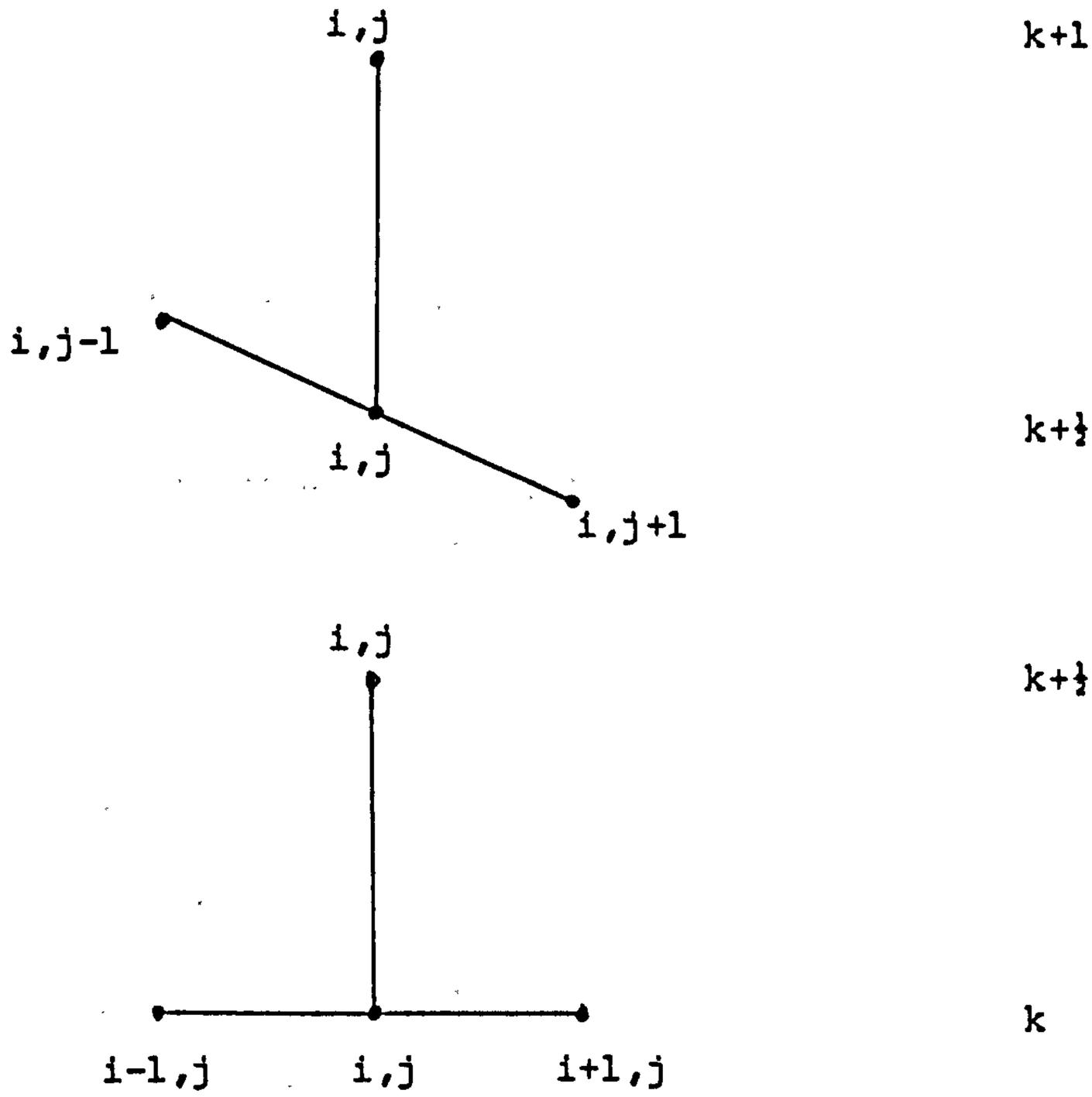


FIGURE 3.12.1: The L.O.D. molecule

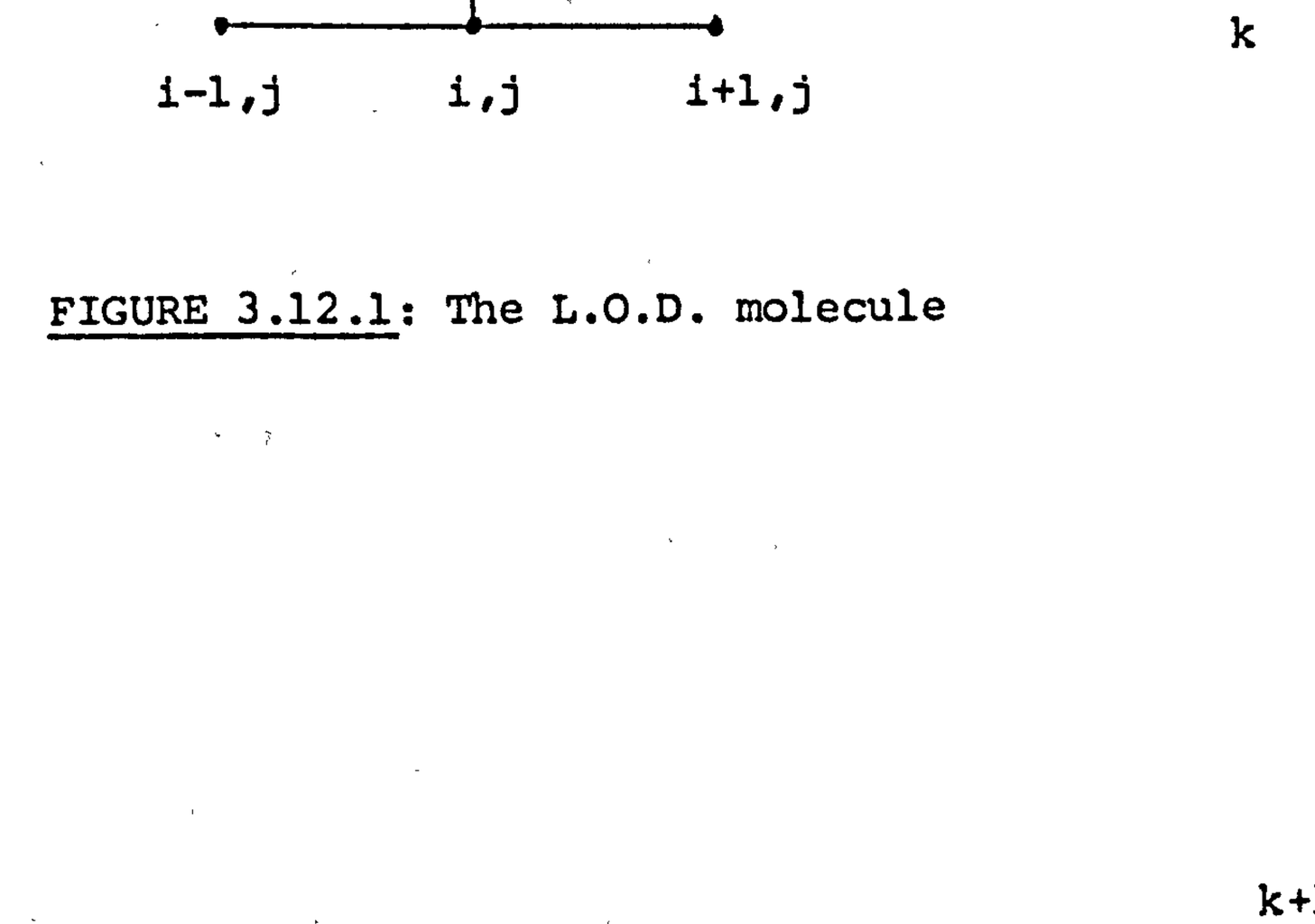


FIGURE 3.12.2: The nine points formula of Eq.(3.12.10)

CHAPTER FOUR

THE SPLINE ALTERNATING GROUP EXPLICIT METHOD

4.1 INTRODUCTION

In this chapter we introduce a new formula for solving parabolic partial differential equations explicitly. It is derived from the cubic splines properties and its function and derivative relationships. The analysis of the stability and an estimate of the truncation errors, together with some numerical results are also included.

The derivation of the method is based on the simplest heat conduction problem given by the equation,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \quad , \; 0 \leq x \leq 1, \; t \geq 0 \; , \tag{4.1.1}$$

under various boundary conditions.

4.2 THE SPLINE ALTERNATING GROUP EXPLICIT METHOD; (SPAGE)

In Section 1.4, some important relations between the spline function, its first derivative and its second derivative are as stated. Now consider the function U in the x-t plane and let us assume the cubic spline polynomial S is interpolating the function U at the points $U_i=S_i$, i=0,1,...,n. The function S satisfies the relations (1.4.9) and (1.4.10) for any time level, since it is independent of time. This leads us to write,

$$m_{i,j}(+) = -\frac{h}{6} M_{i+1,j} - \frac{h}{3} M_{i,j} + \frac{1}{h}(u_{i+1,j}-u_{i,j}) \qquad (4.2.1)$$

and

$$m_{i,j}(-) = \frac{h}{6} M_{i-1,j} + \frac{h}{3} M_{i,j} + \frac{1}{h}(u_{i,j}-u_{i-1,j}) \ , \qquad (4.2.1a)$$

or

$$m_{i,j+1}(-) = \frac{h}{6} M_{i-1,j+1} + \frac{h}{3} M_{i,j+1} + \frac{1}{h}(u_{i,j+1}-u_{i-1,j+1}) \qquad (4.2.2)$$

and

$$m_{i,j+1}(+) = -\frac{h}{6} M_{i+1,j+1} - \frac{h}{3} M_{i,j+1} + \frac{1}{h}(u_{i+1,j+1}-u_{i,j+1}) \qquad (4.2.2a)$$

From the continuity properties of the cubic splines we have that,

$$m_{i,j}(+) = m_{i,j}(-) \ , \qquad (4.2.3)$$

and

$$m_{i,j+1}(+) = m_{i,j+1}(-) \ . \qquad (4.2.4)$$

Letting $\Delta t \to 0$ we get,

$$m_{i,j}(+) = m_{i,j+1}(-) \ , \qquad (4.2.5)$$

and

$$m_{i,j+1}(+) = m_{i,j}(-) \ . \qquad (4.2.6)$$

Substituting equation (4.2.5) in equations (4.2.1) and (4.2.2) gives,

$$\frac{h}{6} M_{i-1,j+1} + \frac{h}{3} M_{i,j+1} + \frac{1}{h}(u_{i,j+1}-u_{i-1,j+1}) = -\frac{h}{6} M_{i+1,j} - \frac{h}{3} M_{i,j}$$

$$+ \frac{1}{h}(u_{i+1,j}-u_{i,j}) \ . \qquad (4.2.7)$$

Similarly from equations (4.2.6), (4.2.1a) and (4.2.2a) we get,

$$-\frac{h}{6}M_{i+1,j+1}-\frac{h}{3}M_{i,j+1}+\frac{1}{h}(u_{i+1,j+1}-u_{i,j+1})=\frac{h}{6}M_{i-1,j}+\frac{h}{3}M_{i,j}$$

$$+\frac{1}{h}(u_{i,j}-u_{i-1,j}) \quad . \qquad (4.2.8)$$

Multiplying by $\frac{6}{h}$ equations (4.2.7) and (4.2.8) become,

$$M_{i-1,j+1}+2M_{i,j+1}+\frac{6}{h^2}(u_{i,j+1}-u_{i-1,j+1})=-M_{i+1,j}-2M_{i,j}$$

$$+\frac{6}{h^2}(u_{i+1,j}-u_{i,j}) \quad , \qquad (4.2.9)$$

and,

$$-M_{i+1,j+1}-2M_{i,j+1}+\frac{6}{h^2}(u_{i+1,j+1}-u_{i,j+1})=M_{i-1,j}+2M_{i,j}$$

$$+\frac{6}{h^2}(u_{i,j}-u_{i-1,j}) \quad , \qquad (4.2.10)$$

respectively.

Assuming that the function U is such that $U \in C^n[a,b]$, $n\geq 4$, it is well known that the best order of uniform convergence that can be achieved by S and its derivatives is,

$$||S^{(r)}-U^{(r)}|| = O(h^{4-r}) \; ; \; r=0,1,2 \quad , \qquad (4.2.11)$$

where $||.||$ denotes the uniform norm on [0,1], [PARAMICHAEL, N. AND WORSEY, A.J.], [DANIEL, J.W. and SWARTZ, B.K.].

Approximating the second derivative of U in equation (4.1.1) by M with the forward difference approximation of $\frac{\partial U}{\partial t}$ on the j^{th} time level once and the backward difference approximation of $\frac{\partial U}{\partial t}$ on the $(j+1)^{th}$ time level for another, we get,

$$M_{i,j} = \frac{u_{i,j+1}-u_{i,j}}{\Delta t} \quad , \qquad (4.2.12)$$

$$M_{i,j+1} = \frac{u_{i,j+1}-u_{i,j}}{\Delta t} \quad , \qquad (4.2.13)$$

respectively, with the approximation errors being ignored. Adding equation (4.2.12) to (4.2.13) gives,

$$M_{i,j}+M_{i,j+1} = 2\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}\right) ,$$

or

$$2(M_{i,j}+M_{i,j+1}) = 4\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}\right) . \qquad (4.2.14)$$

At the points (i-1,j+1) and (i+1,j) we approximate $\frac{\partial U}{\partial t}$ by a weighted backward difference formula at the points (i-1,j+1) and (i,j+1) and a forward difference formula at the points (i-1,j) and (i,j) respectively, to result in rewriting equation (4.1.1) as,

$$M_{i-1,j+1} = \alpha\left(\frac{u_{i-1,j+1}-u_{i-1,j}}{\Delta t}\right) + (1-\alpha)\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}\right) \qquad (4.2.15)$$

and,

$$M_{i+1,j} = \beta\left(\frac{u_{i-1,j+1}-u_{i-1,j}}{\Delta t}\right) + (1-\beta)\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}\right) , \qquad (4.2.16)$$

where $0\leq\alpha,\beta\leq 1$. Adding (4.2.15) to (4.2.16) gives,

$$M_{i-1,j+1}+M_{i+1,j} = 2\left\{\theta\left(\frac{u_{i-1,j+1}-u_{i-1,j}}{\Delta t}\right)+(1-\theta)\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}\right)\right\},$$

(4.2.17)

where $0\leq\theta\leq 1$.

Rearranging equation (4.2.9) to give,

$$(M_{i-1,j+1}+M_{i+1,j})+2(M_{i,j+1}+M_{i,j}) + \frac{6}{h^2}(u_{i,j+1}-u_{i-1,j+1}) = \frac{6}{h^2}(u_{i+1,j}-u_{i,j}).$$

(4.2.18)

Substituting equations (4.2.14) and (4.2.17) in (4.2.18) results in,

$$2\left\{\theta\left(\frac{u_{i-1,j+1}-u_{i-1,j}}{\Delta t}\right)+(1-\theta)\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}\right)\right\}+4\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}\right) + \frac{6}{h^2}(u_{i,j+1}-$$

$$u_{i-1,j+1}) = \frac{6}{h^2}(u_{i+1,j}-u_{i,j}) .$$

(4.2.19)

Multiplying by $\frac{\Delta t}{6}$ equation (4.2.19) leads to the asymmetric formula,

$$(\frac{3-\theta}{3}+r)u_{i,j+1}+(\frac{\theta}{3}-r)u_{i-1,j+1} = ru_{i+1,j}+(\frac{3-\theta}{3}-r)u_{i,j} + \frac{\theta}{3}u_{i-1,j}, \tag{4.2.20}$$

where $r=\frac{\Delta t}{h^2}$ and its associated diagram is shown in Figure (4.2.1).

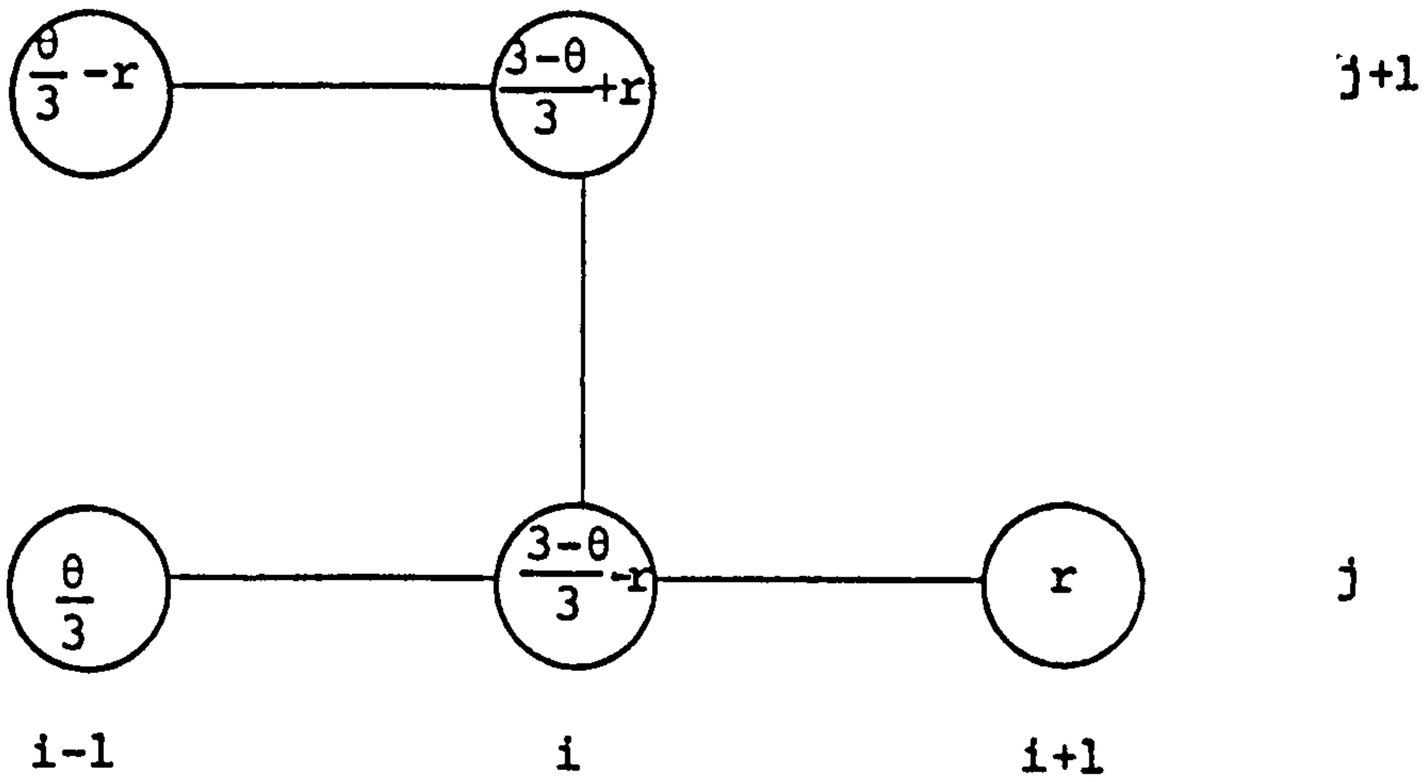


FIGURE 4.2.1

In a similar manner and by approximating $\frac{\partial U}{\partial t}$ at the points (i+1,j+1) and (i-1,j) by weighted backward differences at the points (i+1,j+1) and (i,j+1), and forward differences at the points (i+1,j+1) and (i,j+1) respectively, results in rewriting equation (4.1.1) as,

$$M_{i+1,j+1} = \alpha(\frac{u_{i+1,j+1}-u_{i+1,j}}{\Delta t}) + (1-\alpha)(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}), \tag{4.2.21}$$

and,

$$M_{i-1,j} = \beta(\frac{u_{i+1,j+1}-u_{i+1,j}}{\Delta t}) + (1-\beta)(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}), \tag{4.2.22}$$

and from equations (4.2.10) and (4.2.14) we obtain the other asymmetric formula,

$$(\frac{3-\theta}{3}+r)u_{i,j+1}+(\frac{\theta}{3}-r)u_{i+1,j+1} = \frac{\theta}{3}u_{i+1,j}+(\frac{3-\theta}{3}-r)u_{i,j}+ru_{i-1,j}, \tag{4.2.23}$$

which is represented diagramaticially in Figure (4.2.2).

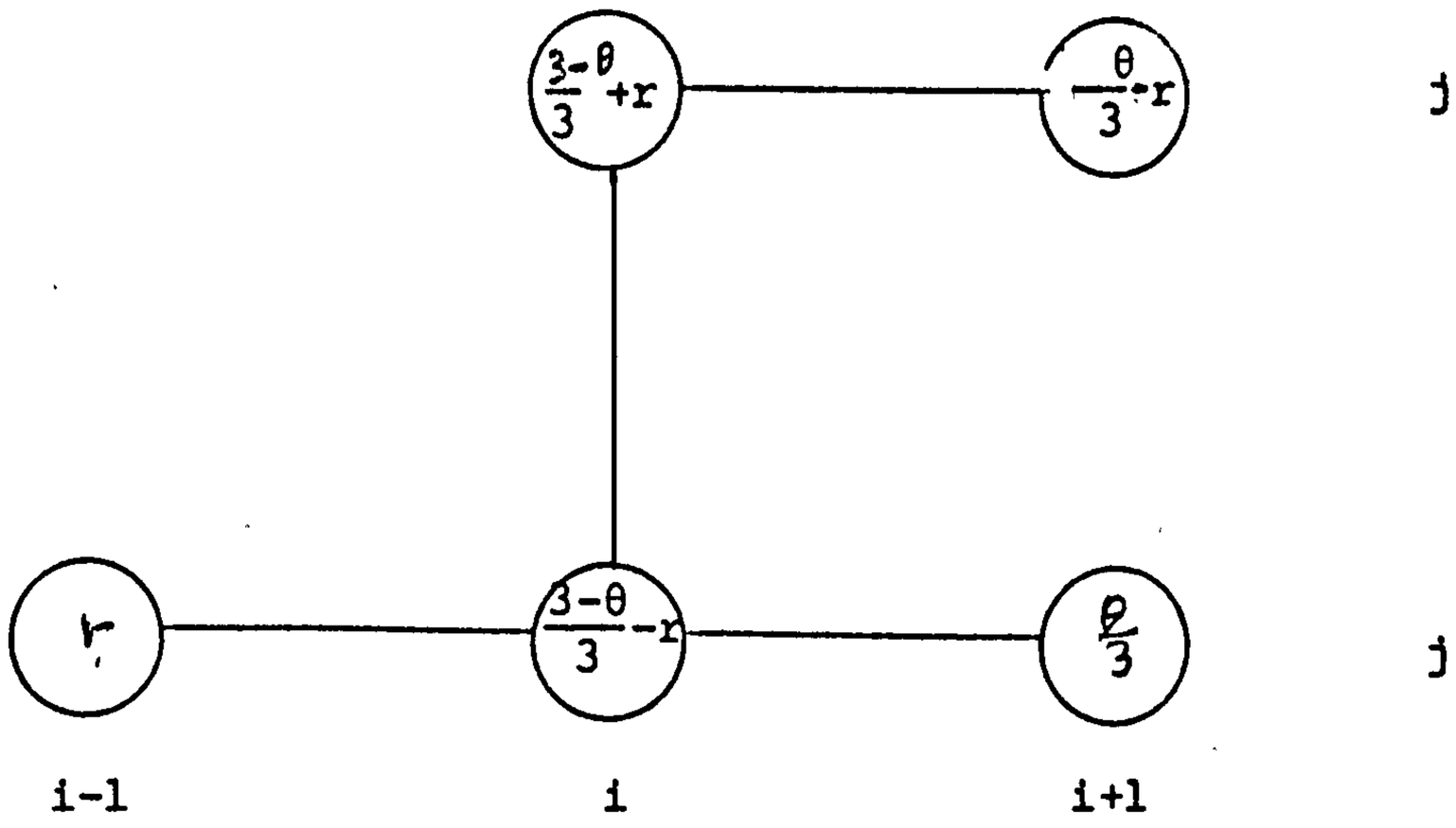


FIGURE 4.2.2

Equations (4.2.20) and (4.2.23) are the spline equivalence to the formulae proposed by Saul'yev (1964), p.29.

It is obvious that these two formulae can be used as was suggested by Saul'yev (1964), i.e. to use equation (4.2.20) from left to right, starting from the node (1,j+1), which is adjacent to the left boundary, and to use equation (4.2.23) from right to left, starting from the node (n-1,j+1) which is adjacent to the right boundary.

Here we do not follow the above procedure, but we combine the two equations (4.2.20) and (4.2.23) to produce a (2x2) coefficient matrix for a group of two unknowns at the points i,j+1 and i+1,j+1. This matrix is then inverted to produce an explicit formula for each of the two unknowns. This is done as follows:

Consider the two points $(i,j+\frac{1}{2})$ and $(i+1,j+\frac{1}{2})$ at which we apply

the equations (4.2.23) and (4.2.20), respectively simultaneously to find the solution u at the points (i,j+1) and (i+1,j). Therefore, at the point (i,j+½) the p.d.e. is approximated by the equation,

$$(\frac{(3-\theta)}{3}+r)u_{i,j+1}+(\frac{\theta}{3}-r)u_{i+1,j+1}=\frac{\theta}{3}u_{i+1,j}+(\frac{(3-\theta)}{3}-r)u_{i,j}+ru_{i-1,j}, \tag{4.2.24}$$

and at the point (i+1,j+½) $u_{i+1,j+1}$ is approximated by the equation,

$$(\frac{(3-\theta)}{3}+r)u_{i+1,j+1}+(\frac{\theta}{3}-r)u_{i,j+1}=\frac{\theta}{3}u_{i,j}+(\frac{(3-\theta)}{3}-r)u_{i+1,j}+ru_{i+2,j}. \tag{4.2.25}$$

In matrix form, equations (4.2.24) and (4.2.25) can be written as,

$$\begin{bmatrix}\frac{(3-\theta)}{3}+r & \frac{\theta}{3}-r\\ \frac{\theta}{3}-r & \frac{(3-\theta)}{3}+r\end{bmatrix}\begin{bmatrix}u_{i,j+1}\\ u_{i+1,j+1}\end{bmatrix}=\begin{bmatrix}\frac{(3-\theta)}{3}-r & \frac{\theta}{3}\\ \frac{\theta}{3} & \frac{(3-\theta)}{3}-r\end{bmatrix}\begin{bmatrix}u_{i,j}\\ u_{i+1,j}\end{bmatrix}+r\begin{bmatrix}u_{i-1,j}\\ \\ u_{i+2,j}\end{bmatrix}. \tag{4.2.26}$$

The 2×2 coefficient matrix can be easily inverted and equation (4.2.26) can be written as,

$$\begin{bmatrix}u_{i,j+1}\\ u_{i+1,j+1}\end{bmatrix}=\frac{1}{\det}\begin{bmatrix}\frac{(3-\theta)}{3}+r & r-\frac{\theta}{3}\\ r-\frac{\theta}{3} & \frac{(3-\theta)}{3}+r\end{bmatrix}\left\{\begin{bmatrix}\frac{(3-\theta)}{3}-r & \frac{\theta}{3}\\ \frac{\theta}{3} & \frac{(3-\theta)}{3}-r\end{bmatrix}\begin{bmatrix}u_{i,j}\\ u_{i+1,j}\end{bmatrix}+r\begin{bmatrix}u_{i-1,j}\\ \\ u_{i+2,j}\end{bmatrix}\right\}, \tag{4.2.27}$$

where $\det=1+\frac{2(3r-\theta)}{3}$. This simplifies to,

$$\begin{bmatrix}u_{i,j+1}\\ \\ u_{i+1,j+1}\end{bmatrix}=\frac{1}{\det}\left\{\begin{bmatrix}\frac{(3-2\theta-3r^2+\theta r)}{3} & \frac{(r\theta-3r^2+3r)}{3}\\ \frac{(r\theta-3r^2+3r)}{3} & \frac{(3-2\theta-3r^2+r\theta)}{3}\end{bmatrix}\begin{bmatrix}u_{i,j}\\ \\ u_{i+1,j}\end{bmatrix}+\right.$$

$$\left.\left[\begin{array}{l}\frac{(3r-r\theta+3r^2)}{3}u_{i-1,j} + \frac{(3r^2-r\theta)}{3}u_{i+2,j} \\ \frac{(3r^2-r\theta)}{3}u_{i-1,j} + \frac{(3r-r\theta+3r^2)}{3}u_{i+2,j}\end{array}\right]\right\}, \qquad (4.2.28)$$

which is obviously two single explicit equations represented diagrammatically by Figure (4.2.3), in a more simplified form.

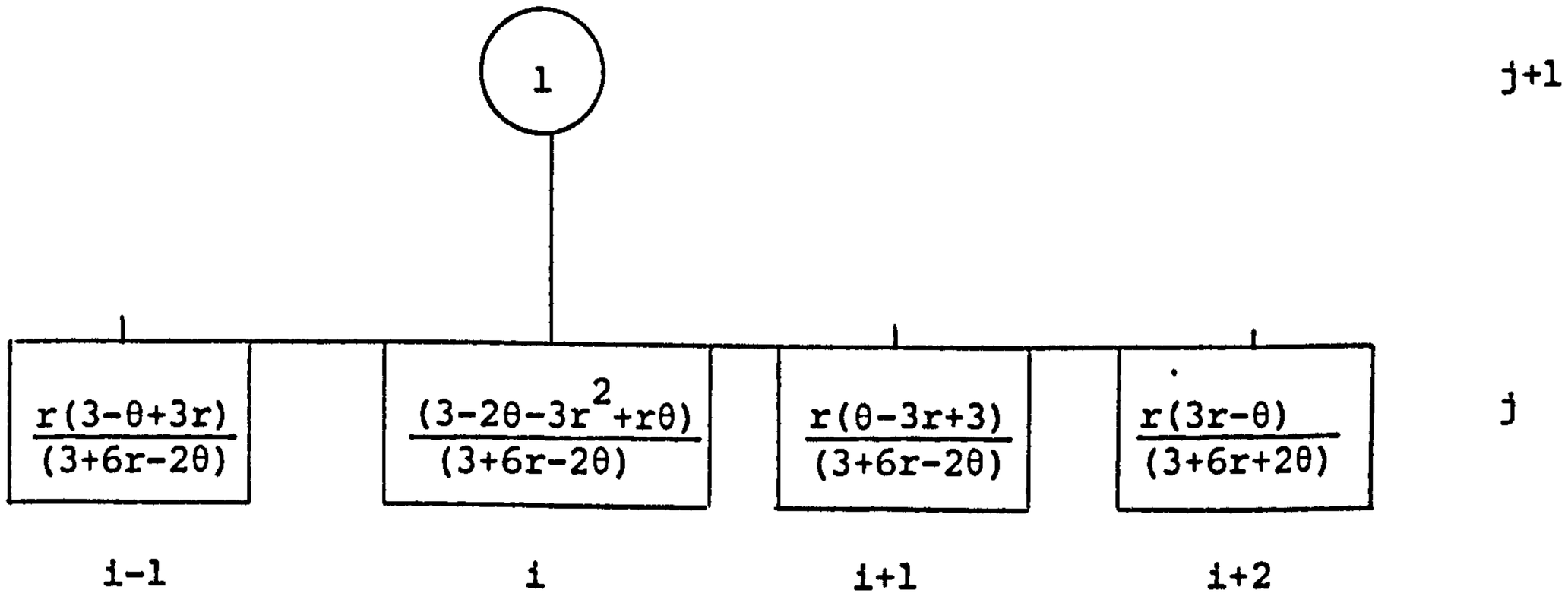


FIGURE 4.2.3

In the case of single points near the left or right boundaries we use equation (4.2.20) or equation (4.2.23) respectively, which are written in the form,

$$u_{1,j+1} = [ru_{2,j}+(\frac{3-\theta}{3}-r)u_{1,j}+\frac{\theta}{3}u_{0,j}-(\frac{\theta}{3}-r)u_{0,j+1}]/(\frac{3-\theta}{3}+r), \tag{4.2.29}$$

for the left boundary and,

$$u_{n-1,j+1} = [ru_{n-2,j}+(\frac{3-\theta}{3}r)u_{n-1,j} + \frac{\theta}{3}u_{n,j}-(\frac{\theta}{3}-r)u_{n,j+1}]/(\frac{3-\theta}{3}+r), \tag{4.2.30}$$

for the right boundary.

4.3 VARIETIES OF THE SPAGE METHOD

The domain line $0 \leq x \leq 1$ is divided into n equal intervals. We show now the varieties that can be produced when n is an even or odd integer.

a) Even Number of Intervals: This means an odd number of internal nodes (n-1) at every time level, which results in one single (ungrouped) node at either end of the interval.

a1) SPAGER (with Right single points)

This scheme is obtained by using equation (4.2.28) for $\frac{(n-2)}{2}$ times for the first (n-2) points (u_i , i=1,2,...,n-2) grouped as two points at a time, and equation (4.2.30) for the single point (u_{n-1}). This is called the Spline Group Alternating Explicit with Right single point. Its diagram is shown in Figure (4.3.1), and its implicit matrix form is written as,

$$
\begin{bmatrix}
3-\theta+3r & \theta-3r & & & & & \\
\theta-3r & 3-\theta+3r & & & & & \\
 & & 3-\theta+3r & \theta-3r & & & \\
 & & \theta-3r & 3-\theta+3r & & & \\
 & & & & \ddots & & \\
 & & & & & 3-\theta+3r & \theta-3r \\
 & & & & & \theta-3r & 3-\theta+3r \\
 & & & & & & 3-\theta+3r
\end{bmatrix}
\begin{bmatrix}
u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ u_{4,j+1} \\ \vdots \\ u_{n-3,j+1} \\ u_{n-2,j+1} \\ u_{n-1,j+1}
\end{bmatrix} =
$$

$$
\begin{bmatrix}
3-\theta-3r & \theta & & & & & & \\
\theta & 3-\theta-3r & 3r & & & & & \\
 & 3r & 3-\theta-3r & \theta & & & & \\
 & & & O & & & & \\
 & & O & & \theta & 3-\theta-3r & 3r & \\
 & & & & 3r & 3-\theta-3r & \theta & \\
 & & & & & \theta & 3-\theta-3r & 3r \\
 & & & & & & 3r & 3-\theta-3r
\end{bmatrix}
\begin{bmatrix}
u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ \vdots \\ u_{n-4,j} \\ u_{n-3,j} \\ u_{n-2,j} \\ u_{n-1,j}
\end{bmatrix}
+ b_1 , \qquad (4.3.1)
$$

where $b_1^T = [3ru_{0,j}, 0, 0, \ldots, 0, \theta u_{n,j} - (\theta-3r)u_{n,j+1}]$,

which includes the known boundary values. The system (4.3.1) can be written as,

$$(3I+(3r-\theta)G_1)\underline{u}_{j+1} = (3I-3rG_2-\theta G_1)u_j + b_1 , \qquad (4.3.2)$$

where,

$$G_1 = \begin{bmatrix} 1 & -1 & & & & \\ -1 & 1 & & & & \\ & & 1 & -1 & & \\ & & -1 & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix} = \begin{bmatrix} G^{(1)} & & & & \\ & G^{(2)} & & & \\ & & \ddots & & \\ & & & G^{(\frac{(n-2)}{2}-1)} & \\ & & & & G^{(\frac{(n-2)}{2})} \end{bmatrix}$$

and

$$G_2 = \begin{bmatrix} 1 & & & & & \\ & 1 & -1 & & & \\ & -1 & 1 & & & \\ & & & \ddots & & \\ & & & & 1 & -1 \\ & & & & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ & G^{(1)} & & & \\ & & G^{(2)} & & \\ & & & \ddots & \\ & & & & G^{(\frac{(n-2)}{2}-1)} \\ & & & & & G^{(\frac{(n-2)}{2})} \end{bmatrix}$$

where,

$$G^{(i)} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad i=1,2,\ldots,(n-2)/2 .$$

a2) <u>SPAGEL</u> (with Left single point)

For this scheme we use equation (4.2.29) for the single point (u_1) and equation (4.2.28) $(\frac{n-2}{2})$ times for the remaining (n-2) points $(u_i,$ $i=2,3,\ldots,n-1)$ grouped two at a time. The scheme is called the Spline Alternating Group Explicit with Left single point, its diagram is shown by Figure (4.3.2) and it is written in matrix form as,

$$(3I+(3r-\theta)G_2)u_{j+1} = (3I-3rG_1-\theta G_2)u_j + b_2 , \qquad (4.3.3)$$

where $b_2 = [\theta u_{0,j} - (\theta-3r)u_{0,j+1}, 0, 0, \ldots, 0, 3ru_{n,j}]$.

a3) SSPAGE

In this scheme the two above mentioned schemes are used at alternative time levels. It is called Single Spline Alternating Group Explicit, and its diagram is shown by Figure (4.3.3). The scheme is expressed in matrix form as,

$$\left.\begin{aligned} (3I+(3r-\theta)G_1)u_{j+1} &= (3I-3rG_2-\theta G_1)u_j + b_1 \\ (3I+(3r-\theta)G_2)u_{j+2} &= (3I-3rG_1-G_2)u_{j+1} + b_2 \end{aligned}\right\}, \qquad (4.3.4)$$

or

$$\left.\begin{aligned} (3I+(3r-\theta)G_2)u_{j+1} &= (3I-3rG_1-\theta G_2)u_j + b_2 \\ (3I+(3r-\theta)G_1)u_{j+2} &= (3I-3rG_2-\theta G_1)u_{j+1} + b_1 \end{aligned}\right\}. \qquad (4.3.5)$$

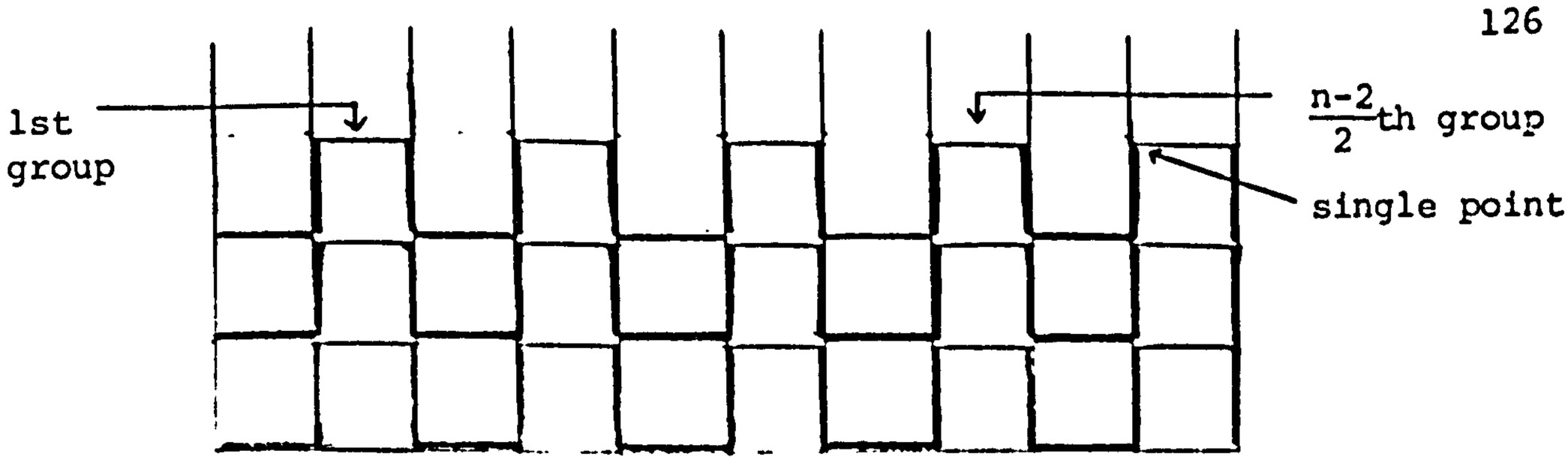
a4) DSPAGE

The final scheme of this group is obtained by using (4.3.4) and (4.3.5) at every other time level alternatively. The Double Spline Alternating Group Explicit requires four time level steps to complete a single cycle. Its diagram is given in Figure (4.3.4), and its matrix form is as follows:-

$$\left.\begin{aligned} (3I+(3r-\theta)G_1)u_{j+1} &= (3I-3rG_2-\theta G_1)u_j + b_1 \\ (3I+(3r-\theta)G_2)u_{j+2} &= (3I-3rG_1-\theta G_2)u_{j+1} + b_2 \\ (3I+(3r-\theta)G_2)u_{j+3} &= (3I-3rG_1-\theta G_2)u_{j+2} + b_2 \\ (3I+(3r-\theta)G_1)u_{j+4} &= (3I-3rG_2-\theta G_1)u_{j+3} + b_1 \end{aligned}\right\}, \qquad (4.3.6)$$

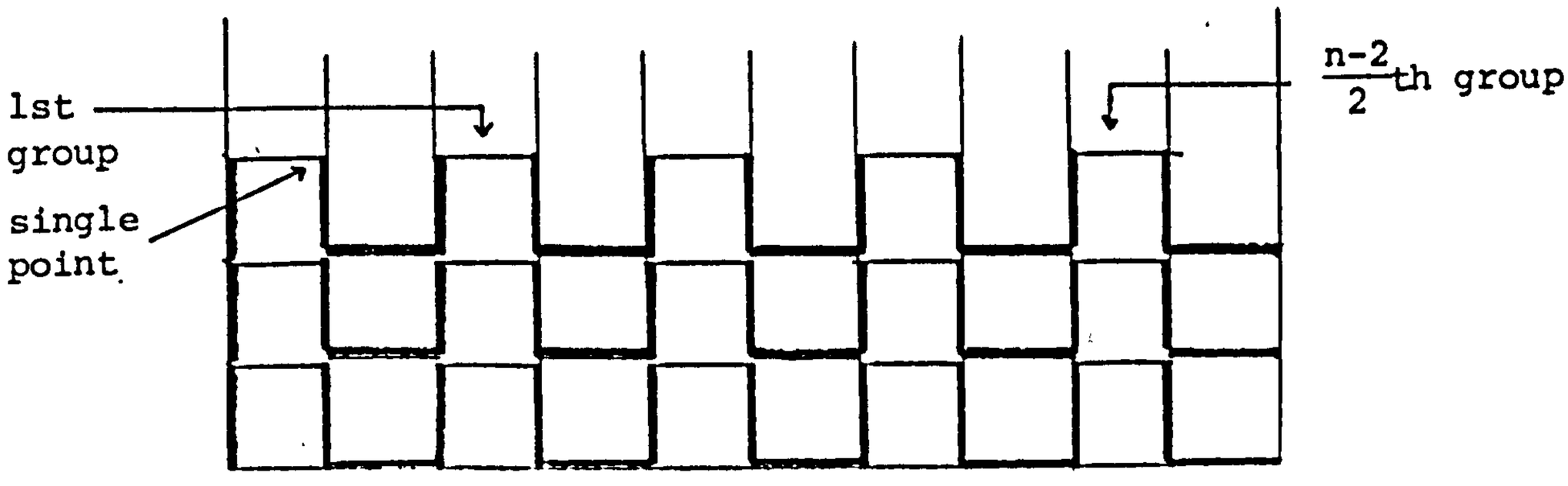
or

$$\left.\begin{aligned} (3I+(3r-\theta)G_2)u_{j+1} &= (3I-3rG_1-\theta G_2)u_j + b_2 \\ (3I+(3r-\theta)G_1)u_{j+2} &= (3I-3rG_2-\theta G_1)u_{j+1} + b_1 \end{aligned}\right\}, \qquad (4.3.7)$$



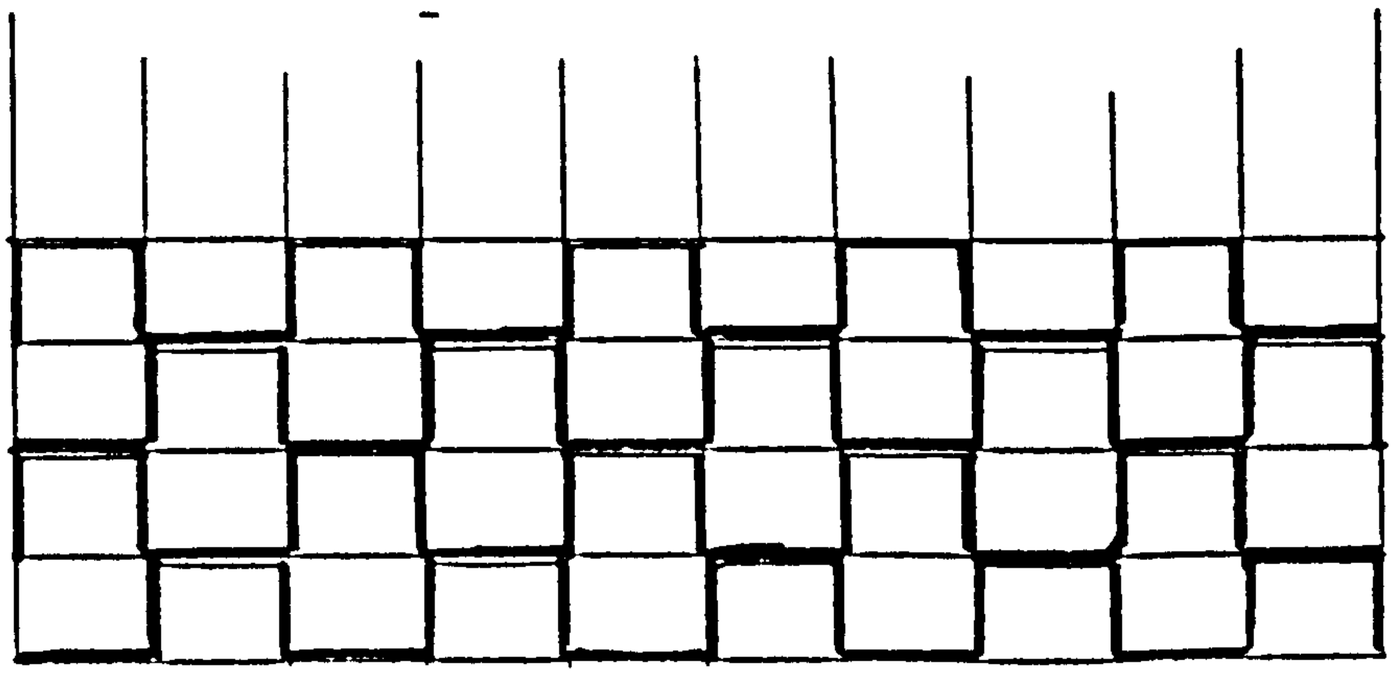
Spline Alternating Group with Right single point (SPAGER) method

FIGURE 4.3.1



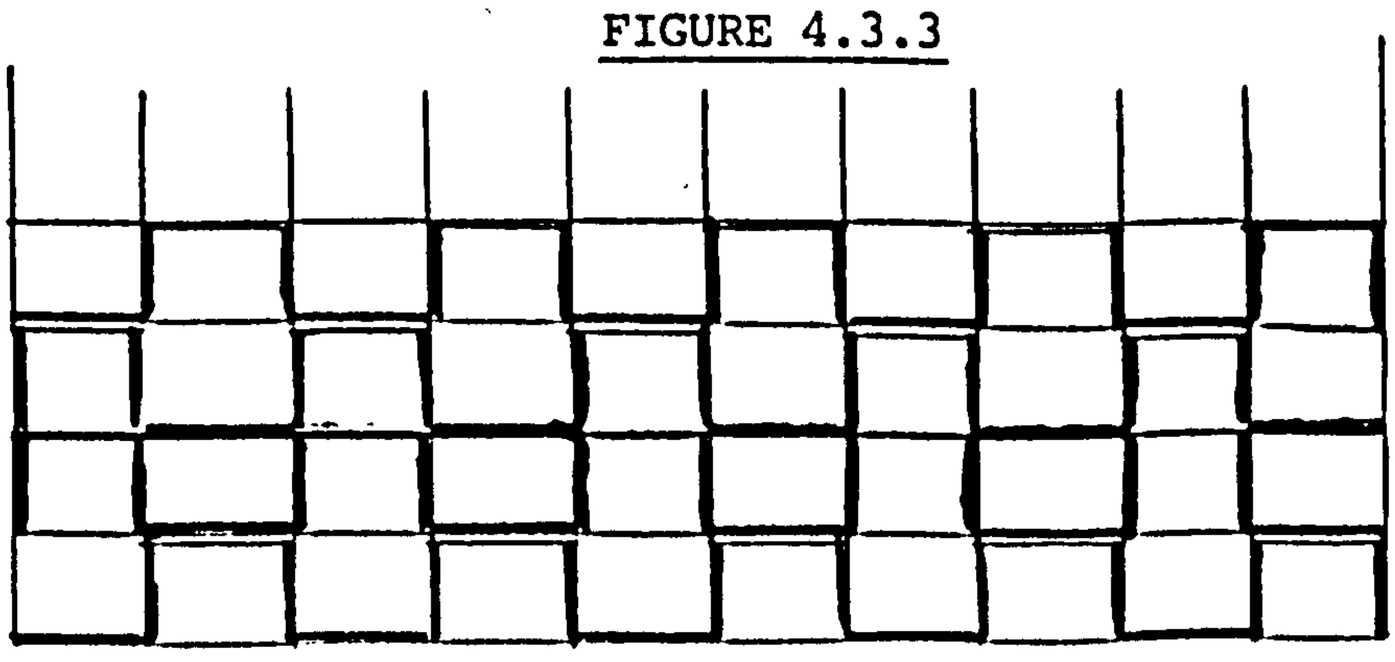
Spline Alternating Group with Left single point (SPAGEL) method

FIGURE 4.3.2



Single Spline Alternating Group Explicit (SSPAGE) method

FIGURE 4.3.3



Double Spline Alternating Group Explicit (DSPAGE) method

FIGURE 4.3.4

$$\left.\begin{aligned}(3I+(3r-\theta)G_1)u_{j+3} &= (3I-3rG_2-\theta G_1)u_{j+2} + b_1\\ (3I+(3r-\theta)G_2)u_{j+4} &= (3I-3rG_1-\theta G_2)u_{j+3} + b_2\end{aligned}\right|$$

b) Odd Number of Intervals: This grid obviously produces an even number of internal nodes (n-1). This leads to either no single point at all at each end or two single (ungrouped) points, one adjacent to each boundary.

b1) SPAGEU (with both Ungrouped ends)

This scheme is obtained by using equation (4.2.29) for the single point near the right boundary, equation (4.2.28) for the grouped points (u_i, i=2,...,n-2) two at a time, i.e., using it ($\frac{n-3}{2}$) times and equation (4.2.30) for the single left point. This is shown in Figure (4.3.5) and its matrix form is written as,

$$(3I+(3r-\theta)\tilde{G}_1)u_{j+1} = (3I-3r\tilde{G}_2-\theta\tilde{G}_1)u_j + b_3\ , \qquad (4.3.8)$$

where $b_3^T = [(\theta u_{0,j}-(\theta-3r)u_{0,j+1}),0,0,\ldots,(\theta u_{n,j}-(\theta-3r)u_{n,j+1}]$,

$$\tilde{G}_1 = \begin{bmatrix} 1 & & & & \\ & G^{(1)} & & & \\ & & G^{(2)} & & O \\ & & & \ddots & \\ & O & & & G^{(\frac{n-3}{2}-1)} \\ & & & & & G^{(\frac{n-3}{2})} \\ & & & & & & 1 \end{bmatrix} ,$$

$$\tilde{G}_2 = \begin{bmatrix} G^{(1)} & & & \\ & G^{(2)} & & O \\ & & \ddots & \\ O & & & G^{\frac{n-1}{2}} \end{bmatrix} ,$$

with $G^{(i)}$, $i=1,\ldots,\frac{n-1}{2}$, as previously defined.

This scheme is called Spline Alternating Group Explicit with both Ungrouped ends.

b2) SPAGEC (with Complete ends)

In this scheme there is no single (ungrouped) point. All the internal points are combined into groups of two points each. It is obtained by using equation (4.2.28) only for the (n-1) points, i.e. to be used $(\frac{n-1}{2})$ times and is called Spline Alternating Group Explicit with Complete ends. The mesh diagram is shown by Figure (4.3.6) and its matrix form is given as,

$$(3I+(3r-\theta)\tilde{G}_2)u_{j+1} = (3I-3r\tilde{G}_1-\theta\tilde{G}_2)u_j + b_4 , \qquad (4.3.9)$$

where $b_4 = [3ru_{0,j},0,0,\ldots,0,3ru_{n,j}]$.

b3) SSPAGE

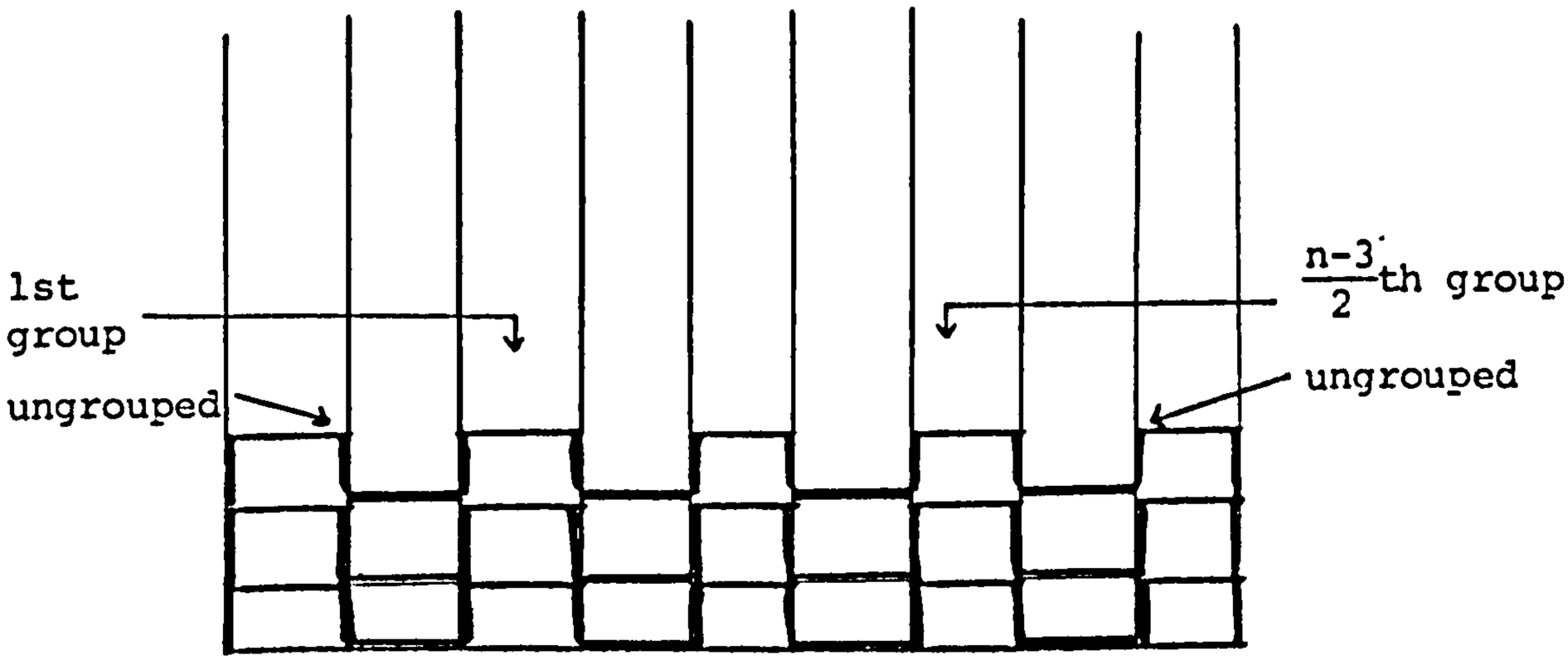
This scheme corresponds to the scheme (4.3.4) and is given by,

$$\left.\begin{aligned} (3I+(3r-\theta)\tilde{G}_1)u_{j+1} &= (3I-3r\tilde{G}_2-\theta\tilde{G}_1)u_j + b_3 \\ (3I+(3r-\theta)\tilde{G}_2)u_{j+2} &= (3I-3r\tilde{G}_1-\theta\tilde{G}_2)u_{j+1} + b_4 \end{aligned}\right\} . \qquad (4.3.10)$$

Its diagram is shown in Figure (4.3.7).

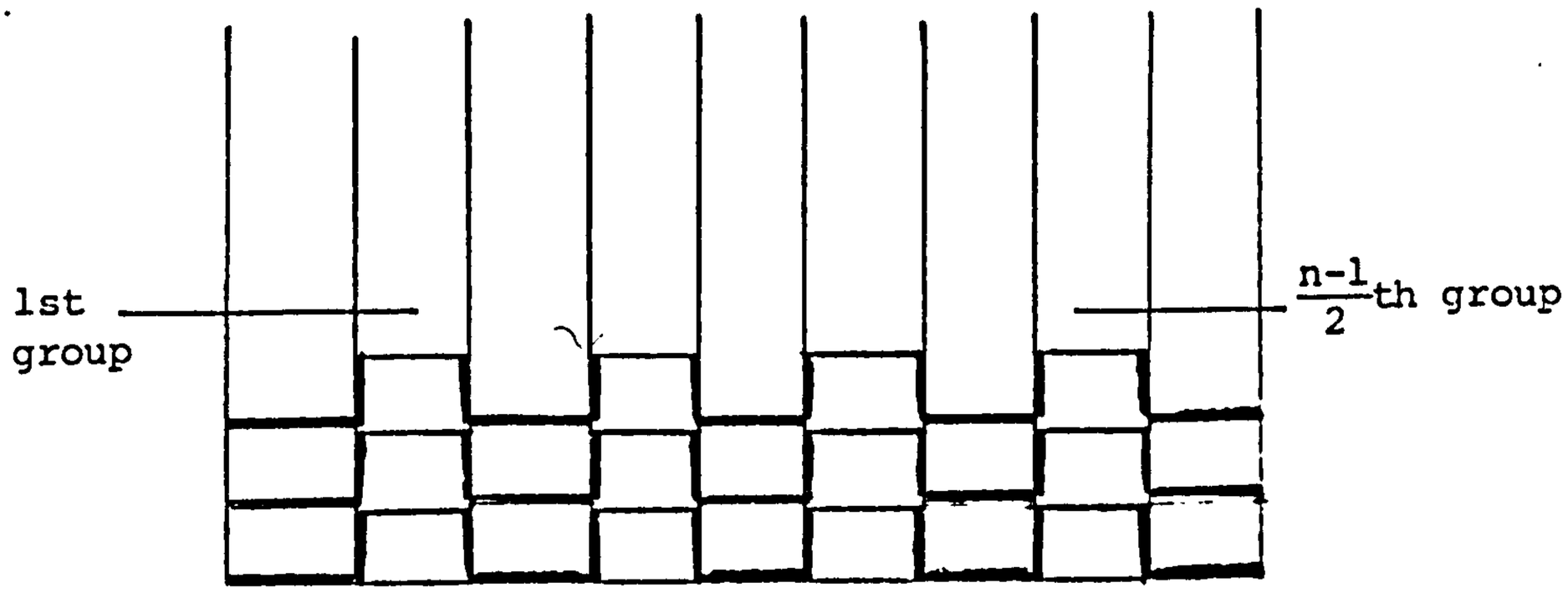
b4) DSPAGE

This is the corresponding scheme to (4.3.6). Its diagram is



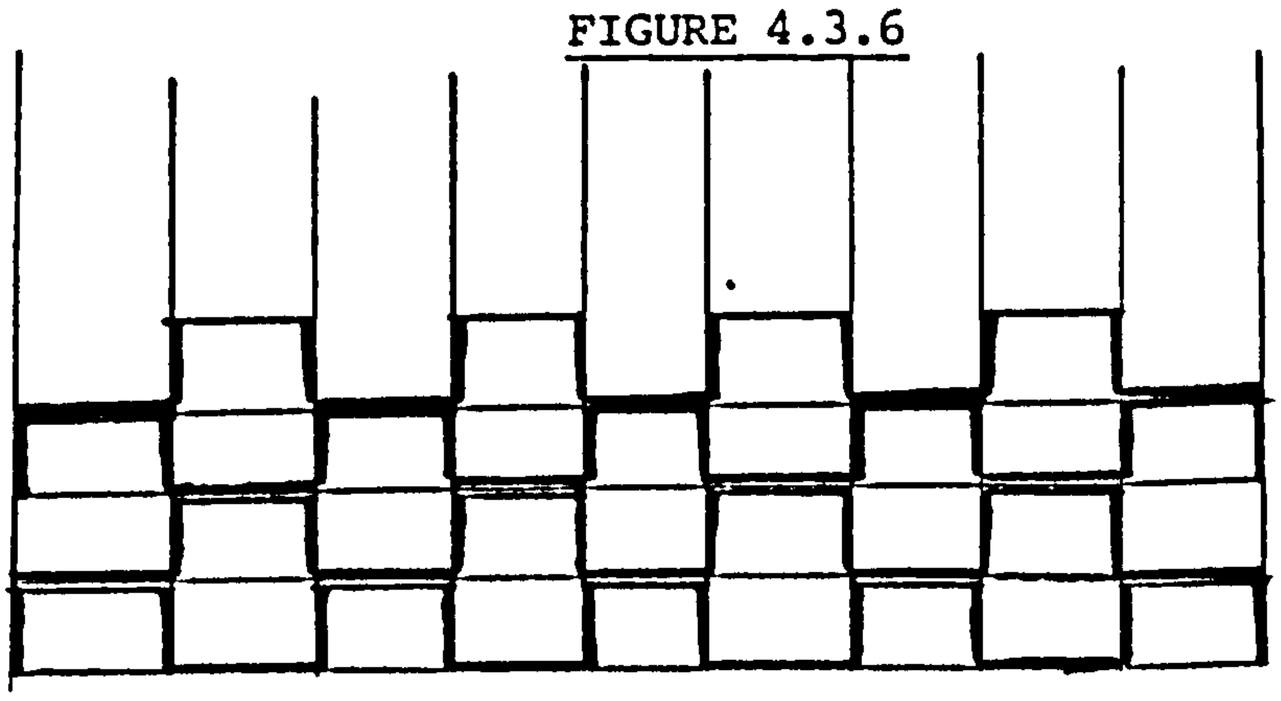
Spline Alternating Group with Ungrouped ends (SPAGEU) method

FIGURE 4.3.5



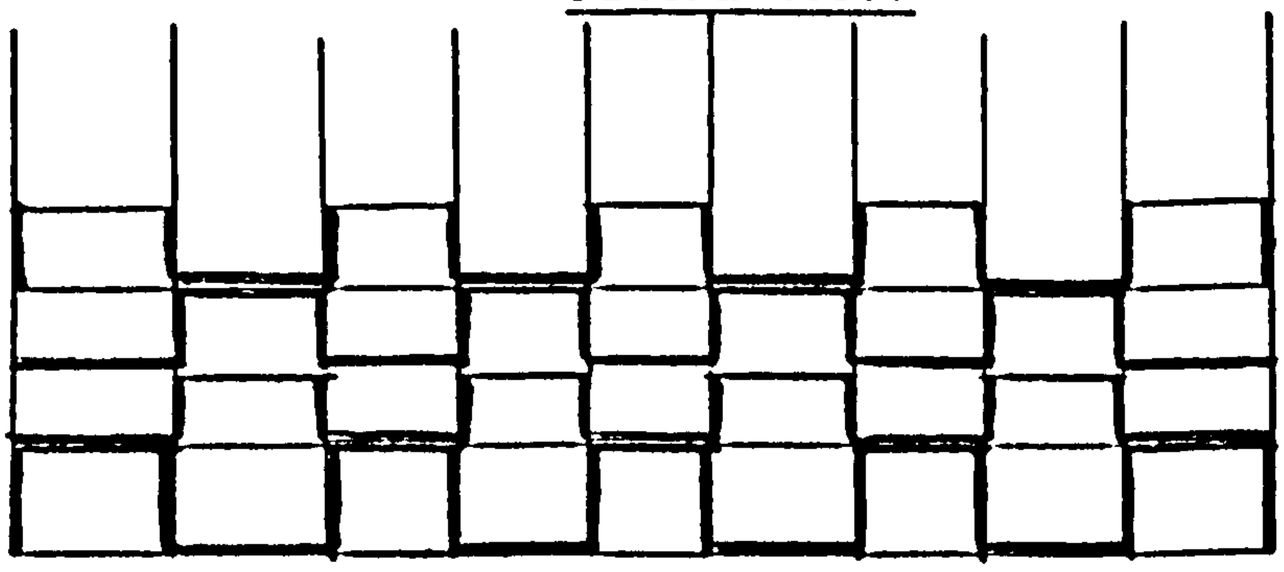
Spline Alternating Group with Complete groups (SPAGEC) method

FIGURE 4.3.6



Single Spline Alternating Group Explicit (SSPAGE) method

FIGURE 4.3.7



Double Spline Alternating Group Explicit (DSPAGE) method

FIGURE 4.3.8

shown in Figure (4.3.8) and its matrix form is given as,

$$\left.\begin{aligned}
(3I+(3r-\theta)\tilde{G}_1)u_{j+1} &= (3I-3r\tilde{G}_2-\theta\tilde{G}_1)u_j + b_3 \\
(3I+(3r-\theta)\tilde{G}_2)u_{j+2} &= (3I-3r\tilde{G}_1-\theta\tilde{G}_2)u_{j+1} + b_4 \\
(3I+(3r-\theta)\tilde{G}_2)u_{j+3} &= (3I-3r\tilde{G}_1-\theta\tilde{G}_2)u_{j+2} + b_4 \\
(3I+(3r-\theta)\tilde{G}_1)u_{j+4} &= (3I-3r\tilde{G}_2-\theta\tilde{G}_1)u_{j+3} + b_3
\end{aligned}\right\} . \qquad (4.3.11)$$

4.4 THE SPLINE ALTERNATING GROUP EXPLICIT METHOD FOR PERIODIC BOUNDARY CONDITIONS

In Section 4.3 we described schemes that are only applicable for the Dirichlet boundary condition. In the case of periodic boundary conditions, the coefficient matrix will be slightly different. We now present the varieties of the case:

c) Even Number of Intervals: This grid produces an even number of unknown points, since $u_{0,j}=u_{n,j}$, for j=0,1,..., and both are unknown.

c1) SPAGEC

This scheme, where we have no single (ungrouped) points, is written in matrix form as below (see Fig.(4.4.1)).

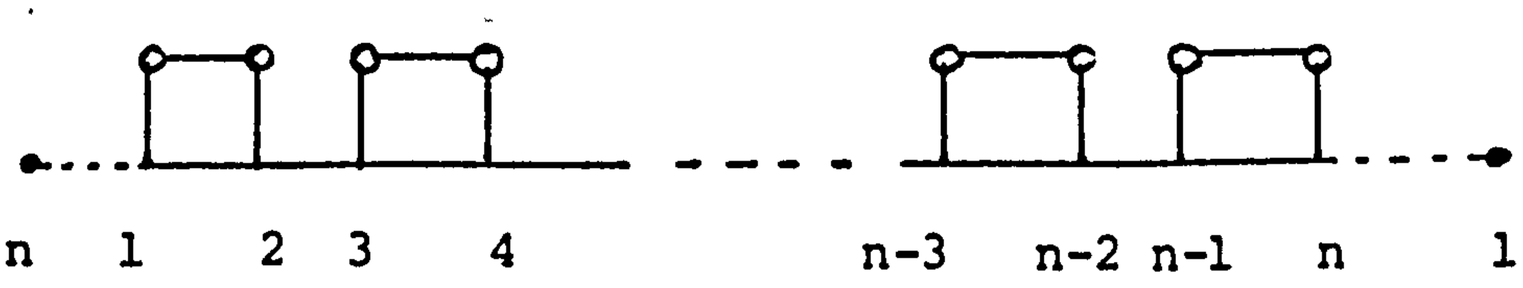


FIGURE 4.4.1

$$\begin{bmatrix} 3-\theta+3r & \theta-3r & & & & & & \\ \theta-3r & 3-\theta+3r & & & & & & \\ & & 3-\theta+3r & \theta-3r & & & & \\ & & \theta-3r & 3-\theta+3r & & & & \\ & & & & \ddots & & & \\ & & & & & 3-\theta+3r & \theta-3r & \\ & & & & & \theta-3r & 3-\theta+3r & \end{bmatrix} \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ u_{4,j+1} \\ \vdots \\ u_{n-1,j+1} \\ u_{n,j+1} \end{bmatrix} =$$

$$\begin{bmatrix} 3-\theta-3r & \theta & & & 3r \\ \theta & 3-\theta-3r & 3r & & \\ & 3r & 3-\theta-3r & \theta & \\ & & & & \\ & & \theta & 3-\theta-3r & 3r \\ & & & 3r & 3-\theta-3r \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ \\ u_{n-2,j} \\ u_{n-1,j} \end{bmatrix}$$

$$\begin{bmatrix} & & & \theta & 3-\theta-3r \\ 3r & & & & \theta \end{bmatrix} \begin{bmatrix} \\ u_{n,j} \end{bmatrix} \qquad (4.4.1)$$

or

$$(3I+(3r-\theta)\tilde{G}_2)u_{j+1} = (3I-3r\tilde{\tilde{G}}_1-\theta\tilde{G}_2)u_j \ , \qquad (4.4.2)$$

where,

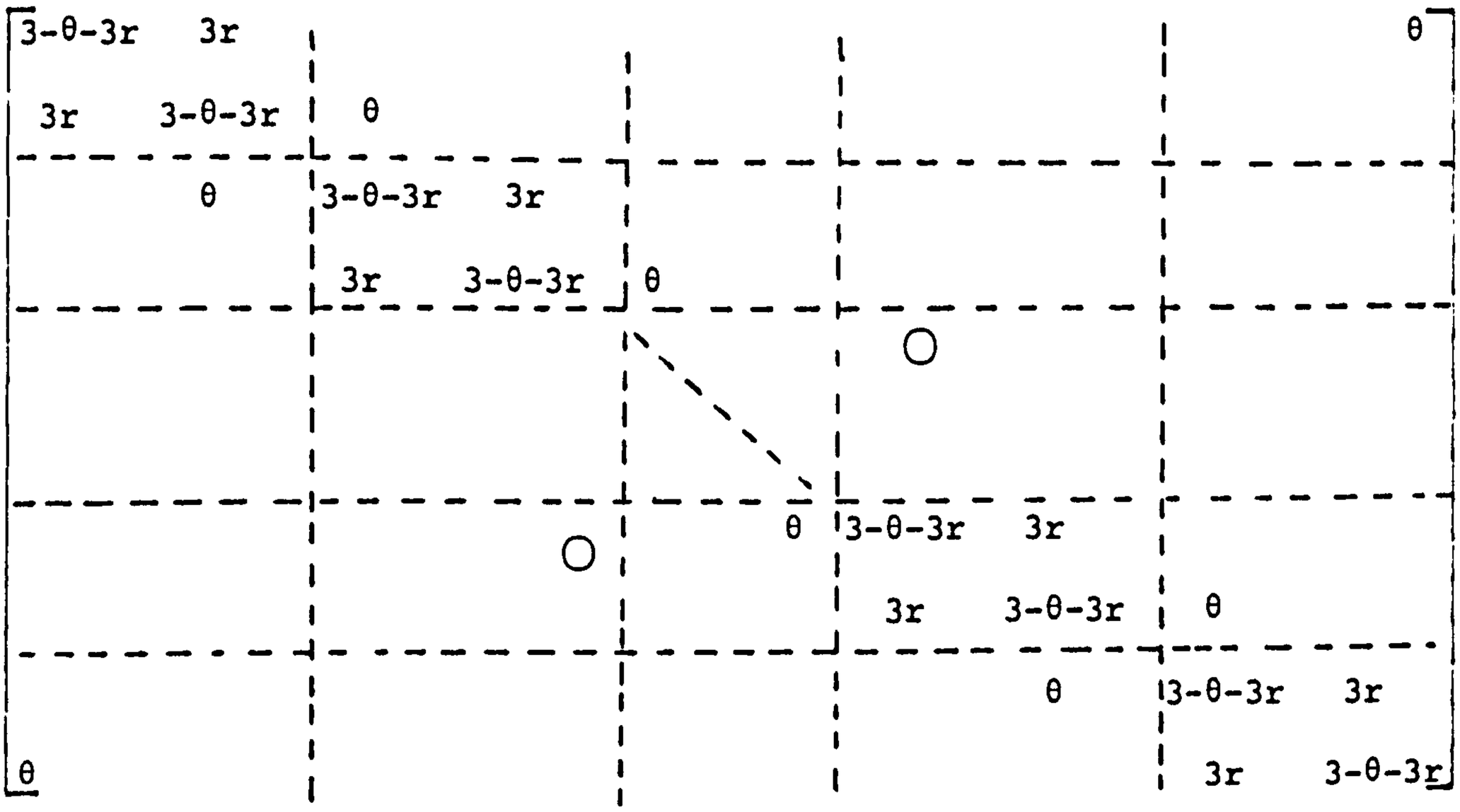
$$\tilde{\tilde{G}}_1 = \begin{bmatrix} 1 & & & & -1 \\ & G^{(1)} & & & \\ & & G^{(2)} & & \\ & & & \ddots & \\ & & & G^{((n-2/2))} & \\ & & C & & \\ -1 & & & & 1 \end{bmatrix} \ , \qquad (4.4.3)$$

with $G^{(i)}$, $i=1,2,\ldots,(\frac{n-2}{2})$ as defined previously.

c2) SPAGEU

i) In this scheme we have two ungrouped end points and is given by,

$$\begin{bmatrix} 3-\theta+3r & & & & \theta-3r \\ & 3-\theta+3r & \theta-3r & & \\ & \theta-3r & 3-\theta+3r & & \\ & & & \ddots & \\ & & & 3-\theta+3r & \theta-3r \\ & & & \theta-3r & 3-\theta+3r \\ \theta-3r & & & & 3-\theta+3r \end{bmatrix} \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ \\ u_{n-2,j+1} \\ u_{n-1,j+1} \\ u_{n,j+1} \end{bmatrix} =$$

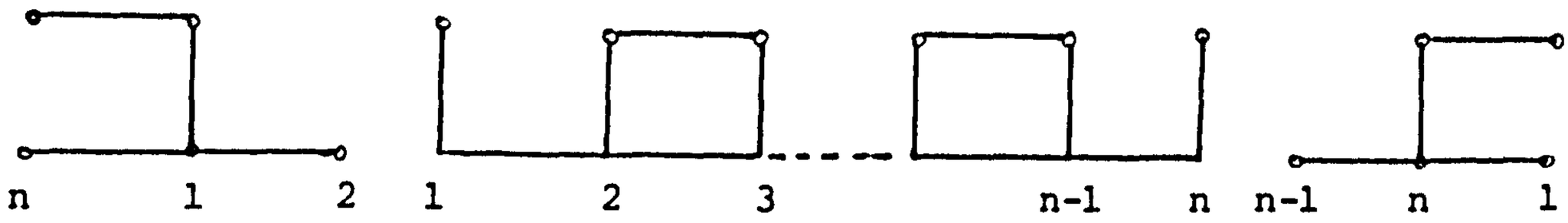


$$\begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ u_{4,j} \\ \vdots \\ u_{n-3,j} \\ u_{n-2,j} \\ u_{n-1,j} \\ u_{n,j} \end{bmatrix}, \qquad (4.4.4)$$

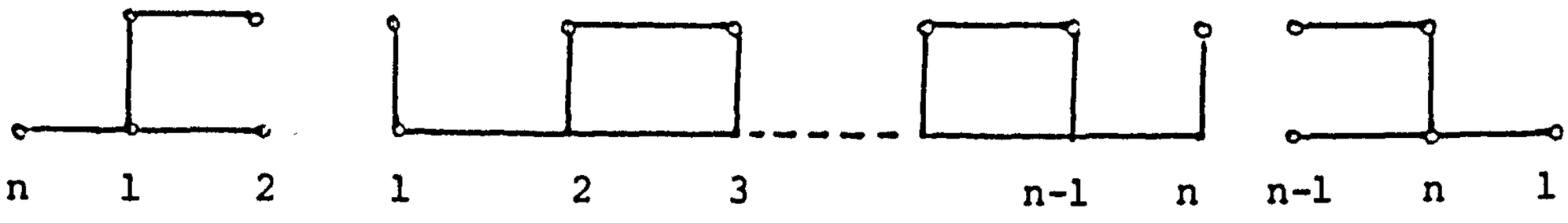
or $$(3I+(3r-\theta)\tilde{\tilde{G}}_1)u_{j+1} = (3I-3r\tilde{G}_2-\theta\tilde{\tilde{G}}_1)u_j \,, \qquad (4.4.5)$$

(see Fig.(4.4.2a)).

ii) Instead of using equation (4.2.20) and (4.2.23) at the points i=1 and i=n respectively, we do the reverse and use equation (4.2.20) for i=n and (4.2.23) for i=1, (see Fig.(4.4.2b)). The system will be,



(a)



(b)

FIGURE 4.4.2: The two ways of treating the single boundary points

$$
\begin{bmatrix}
3-\theta+3r & \theta-3r & & & & \\
3-\theta+3r & \theta-3r & & & & \\
\theta-3r & 3-\theta+3r & & & & \\
 & & \ddots & & & \\
 & & & 3-\theta+3r & \theta-3r & \\
 & & & \theta-3r & 3-\theta+3r & \\
 & & & & \theta-3r & 3-\theta+3r
\end{bmatrix}
\begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \\ \\ \\ u_{n-1,j+1} \\ u_{n,j+1} \end{bmatrix} =
$$

$$
\begin{bmatrix}
3-\theta-3r & \theta & & & & & 3r \\
\theta & 3-\theta-3r & & & & & \\
 & 3-\theta-3r & 3r & & & & \\
 & 3r & 3-\theta-3r & \theta & & & \\
 & & & \theta & 3-\theta-3r & 3r & \\
 & & & & 3r & 3-\theta-3r & \theta \\
3r & & & & & \theta & 3-\theta-3r
\end{bmatrix}
\begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \\ \\ \\ u_{n-1,j} \\ u_{n,j} \end{bmatrix}
$$

or

$$(3I+(3r-\theta)\hat{G}_1^{\vee})u_{j+1} = (3I-3r\hat{G}_2^{\vee}-\theta\hat{G}_3^{\vee})u_j \ ,$$

where,

$$\hat{G}_1^{\vee} = \begin{bmatrix} 1 & -1 & & & & \\ & G^{(1)} & & & & \\ & & G^{(2)} & & O & \\ & & & \ddots & & \\ & & & & G^{(n-2)/2} & \\ & O & & & & -1 & 1 \end{bmatrix},$$

$$\hat{G}_2^{\vee} = \begin{bmatrix} 1 & & & & & & -1 \\ & 1 & & & & & \\ & & G^{(1)} & & & & \\ & & & G^{(2)} & O & & \\ & & & & \ddots & & \\ & O & & & & G^{(n-4)/2} & \\ & & & & & & 1 \\ -1 & & & & & & 1 \end{bmatrix},$$

and,

$$\hat{G}_3^{\vee} = \begin{bmatrix} G^{(1)} & & & & & & \\ & -1 & 1 & & & & \\ & & & G^{(2)} & & & \\ & & & & G^{(3)} & & O \\ & & & & & \ddots & \\ & O & & & & G^{n} & \\ & & & & & 1 & -1 \\ & & & & & & G^{(n-2)/2} \end{bmatrix}.$$

c3) SSPAGE

This scheme is composed by using, alternatively, the schemes SPAGEC and SPAGEU and is written in matrix form, as,

$$\left.\begin{aligned}(3I+(3r-\theta)\tilde{G}_2)u_{j+1} &= (3I-3r\tilde{\tilde{G}}_1-\theta\tilde{G}_2)u_j\\ (3I+(3r-\theta)\tilde{\tilde{G}}_1)u_{j+2} &= (3I-3r\tilde{G}_2-\theta\tilde{\tilde{G}}_1)u_{j+1}\end{aligned}\right\}, \qquad (4.4.6)$$

c4) DSPAGE

As given previously in a4) and b4), this scheme is obtained by applying (4.4.6) followed by a rearranged form of (4.4.6) to make a cycle of four equations written in matrix form as,

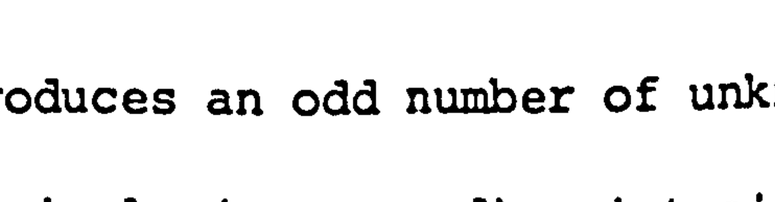
$$\begin{aligned}(3I+(3r-\theta)\tilde{G}_2)u_{j+1} &= (3I-3r\tilde{\tilde{G}}_1-\theta\tilde{G}_2)u_j\ ,\\ (3I+(3r-\theta)\tilde{\tilde{G}}_1)u_{j+2} &= (3I-3r\tilde{G}_2-\theta\tilde{\tilde{G}}_1)u_{j+1}\ ,\\ (3I+(3r-\theta)\tilde{\tilde{G}}_1)u_{j+3} &= (3I-3r\tilde{G}_2-\theta\tilde{\tilde{G}}_1)u_{j+2}\ ,\\ (3I+(3r-\theta)\tilde{G}_2)u_{j+4} &= (3I-3r\tilde{\tilde{G}}_1-\theta\tilde{G}_2)u_{j+3}\ .\end{aligned} \qquad (4.4.7)$$

d) Odd Number of Intervals: This produces an odd number of unknown points at every time level. At the single (ungrouped) point either of the two equations (4.2.24) or (4.2.25) can be used. This leads to produce different forms of systems.

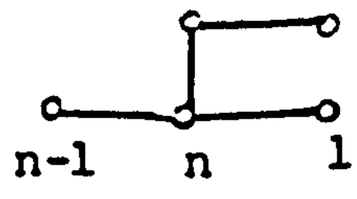
d1) SPAGER

The scheme with right single (ungrouped) point.

i) If equation (4.2.24) is used, for i=n we get,

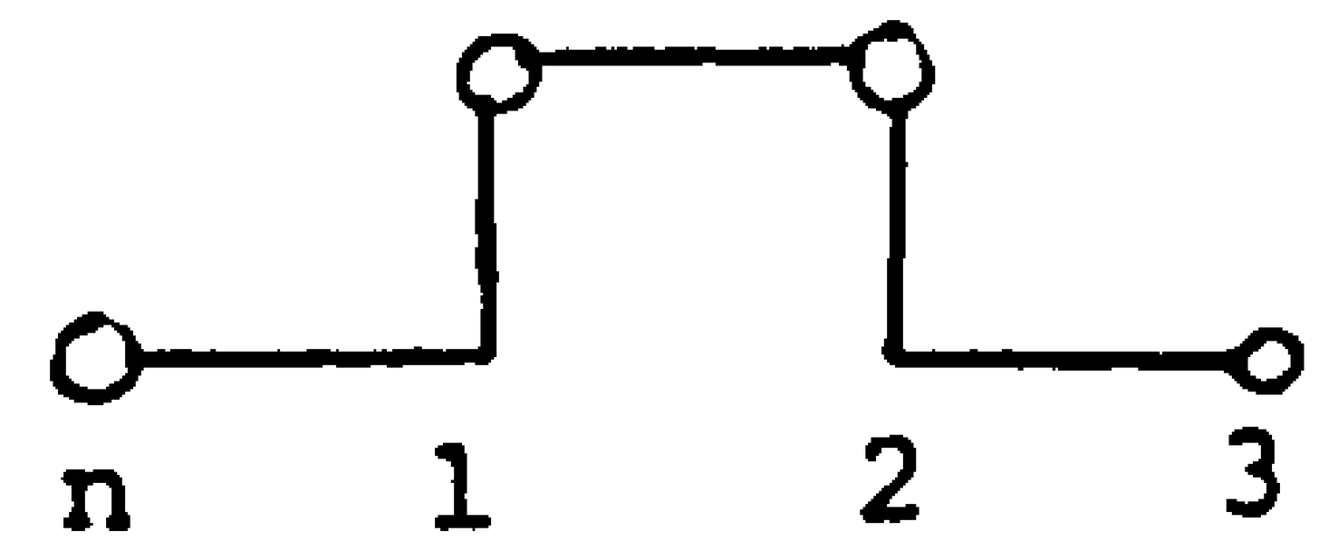


$$(3-\theta+3r)u_{n,j+1}+(\theta-3)u_{1,j+1} = \theta u_{1,j}+(3-\theta-3r)u_{n,j}+3ru_{n-1,j}. \qquad (4.4.8)$$



The first group will be of the form,

$$\begin{bmatrix} 3-\theta+3r & \theta-3r \\ \theta-3r & 3-\theta+3r \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_{j+1} = \begin{bmatrix} 3-\theta-3r & \theta \\ \theta & 3-\theta-3r \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_j + 3r \begin{bmatrix} u_n \\ u_3 \end{bmatrix}_j$$



(4.4.9)

thus the system is written as,

$$[3I+(3r-\theta)\hat{G}_1]u_{j+1} = [3I-3r\hat{G}_2-\theta\hat{G}_1]u_j \ , \qquad (4.4.10)$$

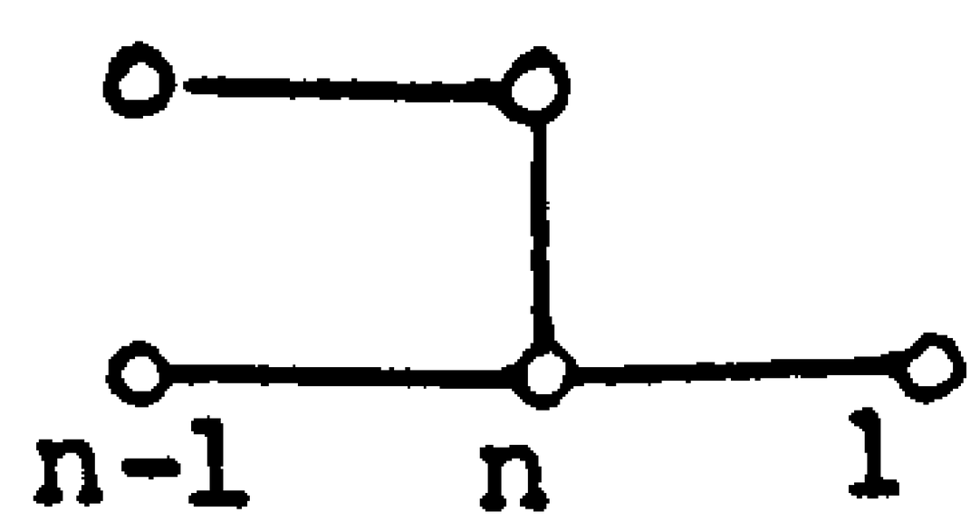
where,

$$\hat{G}_1 = \begin{bmatrix} G^{(1)} & & & & \\ & G^{(2)} & & O & \\ & & G^{(3)} & & \\ & O & & \ddots & \\ & & & & G^{(n-1)/2} \\ -1 & & & & & 1 \end{bmatrix} ,$$

$$\hat{G}_2 = \begin{bmatrix} 1 & & & & -1 \\ & G^{(1)} & & & \\ & & G^{(2)} & O & \\ & O & & \ddots & \\ & & & & G^{(n-1)/2} \end{bmatrix} .$$

ii) If equation (4.2.25) is used, for $i=n-1$, we get,

$$(3-\theta+3r)u_{n,j+1}+(\theta-3r)u_{n-1,j+1} = \theta u_{n-1,j}+(3-\theta-3r)u_{n,j}+3ru_{1,j}$$



(4.4.11)

while for the first group it is the same as in i). Thus the

system will be,

$$[3I+(3r-\theta)\hat{\hat{G}}_1]u_{j+1} = [3I-3r\hat{\hat{G}}_2-\theta\hat{\hat{G}}_1]u_j \ , \qquad (4.4.12)$$

where,

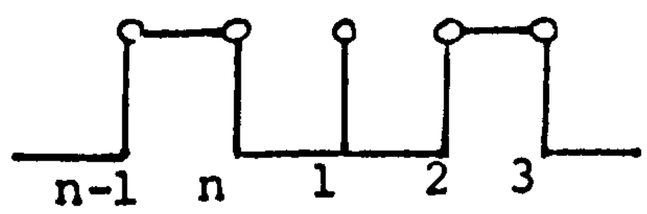
$$\hat{\hat{G}}_1 = \begin{bmatrix} G^{(1)} & & & \\ & G^{(2)} & & O \\ & & \ddots & \\ & O & & G^{(n-1)/2} \\ & & & & -1 & 1 \end{bmatrix}$$

$$\hat{\hat{G}}_2 = \begin{bmatrix} 1 & & & & & -1 \\ & G^{(1)} & & & & \\ & & G^{(2)} & & O & \\ & & & \ddots & & \\ & & O & & G^{(n-3)/2} & \\ & & & & 1 & -1 \\ -1 & & & & & 1 \end{bmatrix}$$

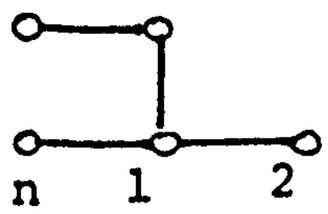
d2) SPAGEL

In this scheme two cases also arise:

i) When using equation (4.2.25), for i+1=1 we get,



$$(3-\theta+3r)u_{1,j+1}+(\theta-3r)u_{n,j+1} = \theta u_{n,j}+(3-\theta-3r)u_{1,j}+3ru_{2,j}$$



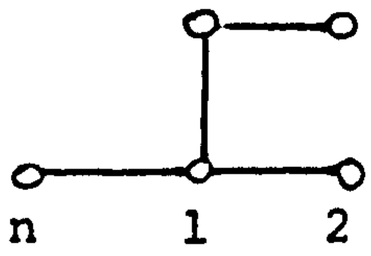
(4.4.13)

and the system will be,

$$[3I+(3r-\theta)\hat{G}_2]u_{j+1} = [3I-3r\hat{G}_1-\theta\hat{G}_2]u_j \ . \qquad (4.4.14)$$

ii) When equation (4.2.24) is used, for i=1 we get,

$$(3-\theta+3r)u_{1,j+1}+(\theta-3r)u_{2,j+1} = \theta u_{2,j}+(3-\theta-3r)u_{1,j}+3ru_{n,j}, \qquad (4.4.15)$$



and the system will be,

$$[3I+(3r-\theta)\overset{\ll}{G}_1]u_{j+1} = [3I-3r\overset{\ll}{G}_2-\theta\overset{\ll}{G}_1]u_j , \qquad (4.4.16)$$

where,

$$\overset{\ll}{G}_1 = \begin{bmatrix} 1 & -1 & & & & \\ & G^{(1)} & & & & \\ & & G^{(2)} & & O & \\ & & & \ddots & & \\ & O & & & G^{(n-1)/2} & \\ 1 & & & & & -1 \end{bmatrix} ,$$

$$\overset{\ll}{G}_2 = \begin{bmatrix} -1 & 1 & & & & \\ & & G^{(1)} & & & \\ & & & G^{(2)} & O & \\ & & & & \ddots & \\ & O & & & G^{(n-3)/2} & \\ -1 & & & & & 1 \end{bmatrix} .$$

d3) SSPAGE

This scheme is obtainable by the use of d1) and d2) alternatively from one time level to another, with either (i) or (ii) always.

d4) DSPAGE

In a similar way to c4) a scheme corresponding to (4.4.7) is obtainable, using either (i) or (ii) in d1) and d2).

4.5 THE SPLINE ALTERNATING GROUP EXPLICIT METHOD FOR DERIVATIVE BOUNDARY CONDITIONS

In the case of derivative boundary conditions, we approximate the derivatives at the end points by the central difference formula.

Let the boundary conditions, defined on $[0\leq x\leq 1,\ t>0]$ be of the form,

$$\frac{\partial U}{\partial x}(x,t) = f_0, \text{ at } x=0 , \tag{4.5.1}$$

and

$$\frac{\partial U}{\partial x}(x_n,t) = f_n, \text{ at } x=1 . \tag{4.5.2}$$

Using the central difference operator on $\frac{\partial U}{\partial x}$, equation (4.5.1) and (4.5.2) give,

$$\frac{u_1-u_{-1}}{2h} = f_0 , \tag{4.5.3}$$

$$\frac{u_{n+1}-u_{n-1}}{2h} = f_n , \tag{4.5.4}$$

respectively. Recalling equation (4.2.20) which for i=0 is written as,

$$(3-\theta+3r)u_{0,j+1}+(\theta-3r)u_{-1,j+1} = 3ru_{1,j}+(3-\theta-3r)u_{0,j}+\theta u_{-1,j} . \tag{4.5.5}$$

Now rearranging equation (4.5.3) as,

$$u_{-1} = u_1-2hf_0 , \tag{4.5.6}$$

and substituting in equation (4.5.5), for both time levels, leads to,

$$(3-\theta+3r)u_{0,j+1}+(\theta-3r)u_{1,j+1} = 3ru_{1,j}+(3-\theta-3r)u_{0,j}+\theta u_{1,j}+2h(\theta-3r)f_{0,j+1} -2h\theta f_{0,j} . \tag{4.5.7}$$

For the other boundary point, we use equation (4.2.23) which for i=n is,

$$(3-\theta+3r)u_{n,j+1}+(\theta-3r)u_{n+1,j+1} = \theta u_{n+1,j}+(3-\theta-3r)u_{n,j}+3ru_{n-1,j} \cdot \tag{4.5.8}$$

Again, rearranging equation (4.5.4) to give,

$$u_{n+1} = u_{n-1}+2hf_n \ , \tag{4.5.9}$$

and substituting in equation (4.5.8), for both time levels, leads to,

$$(3-\theta+3r)u_{n,j+1}+(\theta-3r)u_{n-1,j+1} = \theta u_{n-1,j}+(3-\theta-3r)u_{n,j}+3ru_{n-1,j}$$

$$-2h(\theta-3r)f_{n,j+1}+2h\theta f_{n,j} \ . \tag{4.5.10}$$

Thus, for a single point at either boundary we use; 1) equation (4.5.7) for the lefthand boundary point, which will be written as,

$$(3-\theta+3r)u_{0,j+1} = (3r+\theta)u_{1,j}+(3-\theta-3r)u_{0,j}-(\theta-3r)u_{1,j+1}+$$

$$+2h(\theta-3r)f_{0,j+1}-2h\theta f_{0,j} \ , \tag{4.5.11}$$

where $u_{1,j+1}$ would be known from the first group, 2) equation (4.5.10) for the righthand boundary point which is,

$$(3-\theta+3r)u_{n,j+1} = (\theta+3r)u_{n-1,j}+(3-\theta-3r)u_{n,j}-(\theta-3r)u_{n-1,j+1}-$$

$$-2h(\theta-3r)f_{n,j+1}+2h\theta f_{n,j} \ , \tag{4.5.12}$$

where $u_{n-1,j+1}$ would be known from the previous group.

For the groups which include the end points we proceed as follows: At the lefthand boundary point we have the group of functions u_0 and u_1, so we use the system (4.2.26), with the function u_{-1} being replaced by equation (4.5.6). Therefore, the system will be,

$$\begin{bmatrix} 3-\theta+3r & \theta-3r \\ \theta-3r & 3-\theta+3r \end{bmatrix}\begin{bmatrix} u_{0,j+1} \\ u_{1,j+1} \end{bmatrix} = \begin{bmatrix} 3-\theta-3r & \theta \\ \theta & 3-\theta-3r \end{bmatrix}\begin{bmatrix} u_{0,j} \\ u_{1,j} \end{bmatrix} +3r\begin{bmatrix} u_{1,j}-2hf_{0,j} \\ u_{2,j} \end{bmatrix}. \tag{4.5.13}$$

Similarly at the righthand boundary for the points u_{n-1} and u_n and using the same system but replacing $u_{n+1,j}$ by equation (4.5.9) results in,

$$\begin{bmatrix} 3-\theta+3r & \theta-3r \\ \theta-3r & 3-\theta+3r \end{bmatrix} \begin{bmatrix} u_{n-1,j+1} \\ u_{n,j+1} \end{bmatrix} = \begin{bmatrix} 3-\theta-3r & \theta \\ \theta & 3-\theta-3r \end{bmatrix} \begin{bmatrix} u_{n-1,j} \\ u_{n,j} \end{bmatrix} +$$

$$+3r \begin{bmatrix} u_{n-2,j} \\ u_{n-1,j}+2hf_{n,j} \end{bmatrix} . \qquad (4.5.14)$$

This shows that the coefficient matrix on the righthand side of the system that results from the occurrence of derivative boundary conditions is the same as that resulting from the case of Dirichlet boundary conditions. But the matrices on the lefthand side are slightly different, thus we have the following varieties:

e) Odd Number of Intervals: This produces an even number of points and two cases.

e1) SPAGEC

In this scheme there is no single point and the system is written as follows,

$$[3I+(3r-\theta)\tilde{G}_2]u_{j+1} = [3I-3rG_1^*-\theta\tilde{G}_2]u_j + b_5 , \qquad (4.5.15)$$

where $b_5^T = [-6rhf_0,0,\ldots,0,6rhf_n]$,

$$G_1^* = \begin{bmatrix} 1 & -1 & & & & \\ & G^{(1)} & & & & \\ & & G^{(2)} & & O & \\ & & & \ddots & & \\ & O & & & G^{(n-1)/2} & \\ & & & & -1 & 1 \end{bmatrix} .$$

e2) SPAGEU

This scheme has two single points, one at each boundary. It is written as,

$$[3I+(3r-\theta)\tilde{G}_1]u_{j+1} = [3I-3r\tilde{G}_2-\theta G_1^*]u_j + b_6 , \qquad (4.5.16)$$

where,

$$b_6^T = [-(\theta-3r)(u_{1,j+1}-2hf_{0,j+1})-2h\theta f_{0,j},0,\ldots,0,-(\theta-3r)$$
$$(u_{n-1,j+1}+2hf_{n,j+1})+2h\theta f_{n,j}] .$$

For the schemes SSPAGE and DSPAGE, we follow the same arrangement as in the cases of the Dirichlet and the periodic boundary conditions.

f) Even Number of Intervals: This produces an odd number of points, thus two cases:

f1) SPAGER

A single point at the right boundary is obtained and the system produced is written as,

$$[3I+(3r-\theta)G_1]u_{j+1} = [3I-3r\check{\check{G}}_1-\theta\hat{\hat{G}}_1]u_j + b_7 , \qquad (4.5.17)$$

where,

$$b_7^T = [-6rhf_{0,j},0,\ldots,0,-(\theta-3r)(u_{n-1,j+1}+2hf_{n,j+1})+2h\theta f_{n,j}] .$$

f2) SPAGEL

A single point at the left boundary is obtained and the system produced is written as,

$$[3I+(3r-\theta)G_2]u_{j+1} = [3I-3r\hat{\hat{G}}_1-\theta\check{\check{G}}_1]u_j + b_8 \quad , \tag{4.5.18}$$

where,

$$b_8^T = [-(\theta-3r)(u_{1,j+1}-2hf_{0,j+1})-2h\theta f_{0,j},0,\ldots,0,6rhf_{n,j}] \ .$$

Similarly as in (a), (b), (c) and (d) the schemes SSPAGE and DSPAGE can be produced.

4.6 TRUNCATION ERROR FOR THE SPAGE

In order to estimate the truncation error for the difference schemes of the SPAGE method, we need to estimate the truncation error for the three main equations in the method. The main equations (4.2.20), (4.2.23) and (4.2.26) can be written in simpler forms, respectively, as

$$(3-\theta+3r)u_{i,j+1}+(\theta-3r)u_{i+1,j+1} = \theta u_{i+1,j}+(3-\theta-3r)u_{i,j}+3ru_{i-1,j}\ , \tag{4.6.1}$$

$$(3-\theta+3r)u_{i,j+1}+(\theta-3r)u_{i-1,j+1} = \theta u_{i-1,j}+(3-\theta-3r)u_{i,j}+3ru_{i+1,j}\ , \tag{4.6.2}$$

and

$$\begin{bmatrix} 3-\theta+3r & \theta-3r \\ \theta-3r & 3-\theta+3r \end{bmatrix} \begin{bmatrix} u_{i,j+1} \\ u_{i+1,j+1} \end{bmatrix} = \begin{bmatrix} 3-\theta-3r & \theta \\ \theta & 3-\theta-3r \end{bmatrix} \begin{bmatrix} u_{i,j} \\ u_{i+1,j} \end{bmatrix} + 3r \begin{bmatrix} u_{i-1,j} \\ u_{i+2,j} \end{bmatrix} \tag{4.6.3}$$

where all the schemes we described earlier are combinations of these equations.

Rewriting equation (4.6.1) as,

$$(3-\theta+3r)u_{i,j+1}+(\theta-3r)u_{i+1,j+1}-\theta u_{i+1,j}-(3-\theta-3r)u_{i,j}-3ru_{i-1,j} = 0 \tag{4.6.4}$$

and using the Taylor's expansion about the point $(i,j+\frac{1}{2})$, we get for the following terms,

$$(3-\theta+3r)u_{i,j+1} = (3-\theta+3r)\{u_{i,j+\frac{1}{2}}+\frac{k}{2}u_t+(\frac{k}{2})^2/2\ u_{tt}+(\frac{k}{2})^3/6$$

$$u_{ttt}+(\frac{k}{2})^4/24\ u_{tttt}+O(k^5)\}, \tag{4.6.5a}$$

$$(\theta-3r)u_{i+1,j+1} = (\theta-3r)\ \{u_{i,j+\frac{1}{2}}+hu_x+(\frac{k}{2})u_t+\tfrac{1}{2}(h^2u_{xx}+2h(\frac{k}{2})u_{xt}+(\frac{k}{2})^2u_{tt})$$

$$+\frac{1}{6}(h^3u_{xxx}+3h^2(\frac{k}{2})u_{xxt}+3h(\frac{k}{2})^2u_{xtt}+(\frac{k}{2})^3u_{ttt})+$$

$$\frac{1}{24}(h^4 u_{xxxx}+4h^3(\tfrac{k}{2})u_{xxxt}+6h^2(\tfrac{k}{2})^2 u_{xxtt}+4h(\tfrac{k}{2})^3 u_{xttt}$$

$$+(\tfrac{k}{2})^4 u_{tttt} + O(h^\alpha,k^\beta)\} \quad . \tag{4.6.5b}$$

$$-\theta u_{i+1,j} = -\theta\{u_{i,j+\frac{1}{2}}+hu_x-(\tfrac{k}{2})u_t+\tfrac{1}{2}(h^2u_{xx}-2h(\tfrac{k}{2})u_{xt}+(\tfrac{k}{2})^2 u_{tt})$$

$$+\frac{1}{6}(h^3u_{xxx}-3h^2(\tfrac{k}{2})u_{xxt}+3h(\tfrac{k}{2})^2u_{xtt}-(\tfrac{k}{2})^3u_{ttt})$$

$$+\frac{1}{24}(h^4u_{xxxx}-4h^3(\tfrac{k}{2})u_{xxxt}+6h^2(\tfrac{k}{2})^2u_{xxtt}-4h(\tfrac{k}{2})^3u_{xttt}$$

$$+(\tfrac{k}{2})^4u_{tttt})+O(h^\alpha,k^\beta)\} \tag{4.5.6c}$$

$$-(3-\theta-3r)u_{i,j} = -(3-\theta-3r)\{u_{i,j+\frac{1}{2}}-(\tfrac{k}{2})u_t+(\tfrac{k}{2})^2/2\,u_{tt}-(\tfrac{k}{2})^3/6$$

$$u_{ttt}+(\tfrac{k}{2})^4/24\,u_{tttt}+O(k^5)\} \tag{4.6.5d}$$

and,

$$-3r\,u_{i-1,j} = -3r\{u_{i,j+\frac{1}{2}}-hu_x-(\tfrac{k}{2})u_t+\tfrac{1}{2}(h^2u_{xx}+2h(\tfrac{k}{2})u_{xt}+(\tfrac{k}{2})^2u_{tt})$$

$$-\frac{1}{6}(h^3u_{xxx}+3h^2(\tfrac{k}{2})u_{xxt}+3h(\tfrac{k}{2})^2u_{xtt}+(\tfrac{k}{2})^3u_{ttt})$$

$$+\frac{1}{24}(h^4u_{xxxx}+4h^3(\tfrac{k}{2})u_{xxxt}+6h^2(\tfrac{k}{2})^2u_{xxtt}+4h(\tfrac{k}{2})^3u_{xttt}$$

$$+(\tfrac{k}{2})^4u_{tttt})+O(h^\alpha,k^\beta)\}, \tag{4.6.5e}$$

where $\alpha+\beta=5$. Adding the terms on the righthand side gives,

$$\begin{aligned}
u_{i,j+\frac{1}{2}}\{3+3r-\theta+\theta-3r-3+3r+\theta-\theta-3r\} &= 0\\
hu_x\{\theta-3r-\theta+3r\} &= 0\\
\tfrac{k}{2}u_t\{3+3r-\theta+\theta-3r+3-3r-\theta+\theta+3r\} &= 6\tfrac{k}{2}u_t\\
\frac{h^2}{2}u_{xx}\{\theta-3r-\theta-3r\} &= -6r\,\frac{h^2}{2}u_{xx}\\
h\tfrac{k}{2}u_{xt}\{\theta-3r+\theta-3r\} &= (2\theta-6r)\frac{hk}{2}u_{xt}
\end{aligned}$$

$$\left.\begin{aligned}
\frac{k^2}{8}\,u_{tt}\{3+3r-\theta+\theta-3r-3+3r+\theta-\theta-3r\} &= 0\\
\frac{h^3}{6}\,u_{xxx}\{\theta-3r-\theta+3r\} &= 0\\
h^2\,\frac{k}{4}\,u_{xxt}\{\theta-3r+\theta+3r\} &= 2\theta\frac{h^2k}{4}u_{xxt}\\
\frac{h^2k}{8}u_{xtt}\{\theta-3r-\theta+3r\} &= 0\\
\frac{k^3}{48}\,u_{ttt}\{3+3r-\theta+\theta-3r+3-3r-\theta+\theta+3r\} &= 6\,\frac{k^3}{48}\,u_{ttt}\\
\frac{h^4}{24}\,u_{xxxx}\{\theta-3r-\theta-3r\} &= -6r\,\frac{h^4}{24}\,u_{xxxx}\\
\frac{kh^3}{12}\,u_{xxxt}\{\theta-3r+\theta-3r\} &= (2\theta-6r)\,\frac{kh^3}{12}\,u_{xxxt}\\
\frac{k^2h^2}{16}\,u_{xxtt}\{\theta-3r-\theta-3r\} &= -6r\,\frac{k^2h^2}{16}u_{xxtt}\\
\frac{k^3h}{48}\,u_{xttt}\{\theta-3r+\theta-3r\} &= (2\theta-6r)\,\frac{k^3h}{48}\,u_{xttt}\\
\frac{k^4}{24\times16}\{3+3r-\theta+\theta-3r-3+3r+\theta-\theta-3r\} &= 0
\end{aligned}\right\} \qquad (4.6.6)$$

and

$$O(h^{\alpha},k^{\beta})\ .$$

Adding, again, the terms of the r.h.s. of (4.6.6) and dividing by 3k gives,

$$T_{4.6.1} = (\theta-3r)\frac{h}{3}\,u_{xt} + \frac{\theta h^2}{6}\,u_{xxt} + \frac{k^2}{24}\,u_{ttt} - \frac{h^2}{12}\,u_{xxxx} + (\theta-3r)\,\frac{h^3}{18}\,u_{xxxt} - \frac{k^2}{8}\,u_{xxtt} + (\theta-3r)\,\frac{k^2h}{72}\,u_{xttt} + O(h^{\alpha},k^{\beta})\ , \qquad (4.6.7)$$

where $u_t-u_{xx}=0$, from equation (4.1.1).

Similarly for equation (4.6.2), using Taylor expansion about the point $(i,j+\frac{1}{2})$ we get,

$$T_{4.6.2} = -(\theta-3r)\frac{h}{3}u_{xt} + \frac{\theta h^2}{6}u_{xxt} + \frac{k^2}{24}u_{ttt} - \frac{h^2}{12}u_{xxxx} -$$

$$-(\theta-3r)\frac{h^3}{18}u_{xxxt} - \frac{k^2}{8}u_{xxtt} - (\theta-3r)\frac{hk^2}{72}u_{xttt} + O(h^{\alpha},k^{\beta}). \qquad (4.6.8)$$

For equation (4.6.3), we use Taylor expansion about the point $(i,j+\frac{1}{2})$ for the 1st equation of (4.6.3) and about the point $(i+1,j+\frac{1}{2})$ for the 2nd equation of (4.6.3). This results in, respectively,

$$T_{(4.6.3)I} = (-h(3r-\theta)/(3+6r-2\theta))u_{xxx} + (\frac{k}{2})u_{xxt} + \frac{k^2}{24}u_{ttt} +$$

$$(\frac{h^2(8\theta-3-24r)}{12(3+6r-2\theta)})u_{xxxx} + (\frac{kh(3r-\theta)}{2(3+6r-2\theta)})u_{xxxt} - \frac{k^2}{8}u_{xxtt}$$

$$+ O(h^{\alpha},k^{\beta}) , \qquad (4.6.9)$$

and,

$$T_{(4.6.3)II} = (h(3r-\theta)/(3+6r-2\theta))u_{xxx} + \frac{k}{2}u_{xxt} + \frac{k^2}{24}u_{ttt} +$$

$$(\frac{h^2(8\theta-3-24r)}{12(3+6r-2\theta)})u_{xxxx} - (\frac{kh(3r-\theta)}{2(3+6r-2\theta)})u_{xxxt} - \frac{k^2}{8}u_{xxtt}$$

$$+ O(h^{\alpha},k^{\beta}) . \qquad (4.6.10)$$

Thus having found the truncation errors for the equations (4.6.1), (4.6.2) and (4.6.3), we can express the truncation error for the difference schemes of the Spline Alternating Group Explicit Method in terms of equations (4.6.7), (4.6.8), (4.6.9) and (4.6.10).

i) SPAGER

For any group of two points, $(i,j+\frac{1}{2})$ and $(i+1,j+\frac{1}{2})$, that are computationally combined, the truncation error is given by the two equations (4.6.9) and (4.6.10), respectively, for i=1,3,...,n-3. For the last single (ungrouped) point, the truncation error is given by equation (4.6.7). (See Figure (4.6.1a)).

ii) SPAGEL

For the first single point of this scheme, the truncation error is given by (4.6.8), and for the grouped points $(i,j+\frac{1}{2})$ and $(i+1,j+\frac{1}{2})$, $i=2,4,\ldots,n-2$, the truncation error is given by equations (4.6.9) and (4.6.10) respectively. (See Figure (4.6.1b)).

iii) SPAGEC

In this scheme, which occurs in the case of an odd number of intervals, there is no single (ungrouped) point, therefore the truncation error is given by the equations (4.6.9) and (4.6.10) only. (See Figure (4.6.2a)).

iv) SPAGEU

Since this scheme has a single point at each end, then the truncation error for the first (left end) single point is given by equation (4.6.8), for the last (right end) single points is given by equation (4.6.7) and for the point $(i,j+\frac{1}{2})$ and $(i+1,j+\frac{1}{2})$, $i=2,4,\ldots,n-3$, is given by the equations (4.6.9) and (4.6.10), respectively. (See Figure (4.6.2b)).

v) SSPAGE

For this scheme, in the both cases of even or odd number of intervals, the truncation error on the (j)th and (j+1)th time level is given by the truncation errors of SPAGER and SPAGEL, respectively, for the even case, or by the truncation error of SPAGEC and SPAGEV, respectively for the odd case. In any case, some terms of the truncation error, on most of the interior points, tend to cancel each other due to the difference in their signs. These terms are the

[h(3r-θ)/(3+6r-2θ)] and [kh(3r-θ)/2(3+6r-2θ)]. This leaves the truncation error to be approximately $O(k,h^2)$. (See Figure (4.6.1c) for even case and Figure (4.6.2c) for odd case).

vi) DSPAGE

Similar to the previous scheme, this has a truncation error of $O(k,h^2)$, due to the same reason as given in (v). (See Figure (4.6.1d) for even case and Figure (4.6.2d) for odd case).

FIGURE 4.6.1: The distribution of errors at every time level (even case)

(4.6.9)	(4.6.10)	(4.6.9)	(4.6.10)		(4.6.9)	(4.6.10)	(4.6.9)	(4.6.10)	j

(a)

(4.6.8)	(4.6.9)	(4.6.10)	(4.6.9)		(4.6.10)	(4.6.9)	(4.6.10)	(4.6.7)	j

(b)

(4.6.8)	(4.6.9)	(4.6.10)	(4.6.9)		(4.6.10)	(4.6.9)	(4.6.10)	(4.6.7)	j+1
(4.6.9)	(4.6.10)	(4.6.9)	(4.6.10)		(4.6.9)	(4.6.10)	(4.6.9)	(4.6.10)	j

(c)

(4.6.9)	(4.6.10)	(4.6.9)	(4.6.10)		(4.6.9)	(4.6.10)	(4.6.9)	(4.6.10)	j+3
(4.6.8)	(4.6.9)	(4.6.10)	(4.6.9)		(4.6.9)	(4.6.9)	(4.6.10)	(4.6.7)	j+2
(4.6.8)	(4.6.7)	(4.6.10)	(4.6.9)		(4.6.10)	(4.6.9)	(4.6.10)	(4.6.7)	j+1
(4.6.9)	(4.6.10)	(4.6.9)	(4.6.10)		(4.6.9)	(4.6.10)	(4.6.9)	(4.6.10)	j

(d)

FIGURE 4.6.2: The distribution of errors at every time level (odd case)

4.7 STABILITY OF THE SPLINE ALTERNATING GROUP EXPLICIT METHOD

To establish the stability conditions of this group of schemes, we will deal first with the case when Dirichlet boundary conditions are imposed on all the boundaries. Any other case will be easy to consider with, may be, a little modification.

The scheme a1) can be written as,

$$u_{j+1} = T_r u_j + \underline{b} \ , \tag{4.7.1}$$

where,

$$T_r = \begin{bmatrix} t_1 & t_2 & t_3 & & & & & \\ t_2 & t_1 & t_4 & & & & & \\ & t_4 & t_1 & t_2 & t_3 & & & \\ & t_3 & t_2 & t_1 & t_4 & & & \\ & & & & \ddots & & & \\ & & & & t_4 & t_1 & t_2 & t_3 \\ & & & & t_3 & t_2 & t_1 & t_4 \\ & & & & & & t_4' & t_1' \end{bmatrix} , \tag{4.7.2}$$

where,

$$t_1 = (3-2\theta-3r^2+r\theta)/det \ ,$$

$$t_2 = r(\theta-3r+3)/det \ ,$$

$$t_3 = r(3r-\theta)/det \ ,$$

$$t_4 = r(3-\theta+3r)/det \ ,$$

$$t_4' = r/(3-\theta+3r) \ ,$$

$$t_1' = (3-\theta-3r)/(3-\theta+3r) \ ,$$

and

$$\underline{b} = [t_4 u_{0,j}, t_3 u_{0,j}, 0, \ldots, 0, (\theta u_{n,j} - (\theta-3r) u_{n,j-1})/(3-\theta+3r)] \ ,$$

where,

$$det = (3+6r-2\theta) \ .$$

The system (4.7.1) is stable if,

$$||T_r||_\infty = \max_{\text{row}} \left(\sum_{i=1}^{4} |t_i| \right) \leq 1 \ . \qquad (4.7.3)$$

From examining the elements of the matrix T_r we find that the rows from the third to the (n-1)th have the maximum sum of modulus.

Therefore,

$$||T_r||_\infty = |t_1|+|t_2|+|t_3|+|t_4|$$

$$= \frac{|3-2\theta-3r^2+r\theta|+|r\theta-3r^2+3r|+|3r^2-r\theta|+|3r-r\theta+3r^2|}{|3+6r-2\theta|}$$

From the condition $0 \leq \theta \leq 1$ and $r>0$, we have:

1) For $\theta=0$

$$||T_r||_\infty = \frac{3|1-r^2|+3r|1-r|+3r^2+3r|1+r|}{|3+6r|}$$

1a) If $r \leq 1$,

$$||T_r||_\infty = \frac{3-3r^2+3r-3r^2+3r^2+3r+3r^2}{3+6r} = 1$$

1b) If $r>1$,

$$||T_r||_\infty = \frac{3r^2-3+3r^2-3r+3r^2+3r+3r^2}{3+6r} = \frac{12r^2-3}{3+6r} > 1 \ .$$

Thus $r>1$ is invalid for stability.

2) For $\theta=1$

$$||T_r||_\infty = \frac{|1+r-3r^2|+r|4-3r|+r|3r-1|+|2r+3r^2|}{|1+6r|}$$

For the first term $(1+r-3r^2)$ to be positive we need r to be <0.767.

2a) If $r<1/3$,

$$||T_r||_\infty = \frac{1+r-3r^2+4r-3r^2+r-3r^2+2r+3r^2}{1+6r} = \frac{1+8r-6r^2}{1+6r}$$

This is >1 as soon as $r>0$.

Therefore $||T_r||_\infty = 1+\varepsilon$, where $0<\varepsilon<0.09$ and $(1+0.09)^n$ does not grow very rapidly.

2b) If $1/3 \leq r \leq 0.767$,

$$||T_r||_\infty = \frac{1+r-3r^2+4r-3r^2+3r^2-r+2r+3r^2}{1+6r} = 1.$$

2c) If $0.767 < r < 4/3$,

$$||T_r||_\infty = \frac{3r^2-r-1+4r-3r^2+3r^2-r+2r+3r^2}{1+6r} = \frac{6r^2+4r-1}{1+6r}$$

$$> 1.$$

In this instance, we have evidence from the numerical experiments carried out that the scheme is stable for $r \leq 1$.

2d) If $r \geq 4/3$,

$$||T_r||_\infty = \frac{3r^2-r-1+3r^2-4r+3r^2-r+2r+3r^2}{1+6r} = \frac{12r^2-4r-1}{1+6r}$$

$$> 1.$$

We have proved that the method is stable when $\theta=0$ and $r \leq 1$ and when $\theta=1$ and $1/3 \leq r \leq 0.767$.

By direct evaluation of the eigenvalues of T_r when $n=31$ and $r=1$, for $\theta=0.5, 0.7, 0.8, 0.9, 0.95, 0.99$ and from the apparent stability of the numerical solutions obtained in Chapter 5 and 6 it appears that

$$\rho(T_r) \leq 1 \text{ for } r \leq 1. \qquad (4.7.4)$$

Hence we believe that the SPAGER scheme is stable for $r \leq 1$.

Similarly for the a2) scheme, where the amplification matrix T_ℓ is given as,

$$T_\ell = \begin{bmatrix} t_1' & t_4 & & & & & \\ t_4 & t_1 & t_2 & t_3 & & & \\ t_1 & t_2 & t_1 & t_4 & & O & \\ & & & \ddots & \ddots & & \\ & O & & & t_4 & t_1 & t_2 \\ & & & & t_3 & t_2 & t_1 \end{bmatrix},$$

which is also stable for $r \leq 1$, i.e. SPAGEL is stable for $r \leq 1$.

We can also prove the stability of the equations (4.3.2) and (4.3.3) in their implicit form by using this Lemma of Kellog [1964, p.848].

Lemma 4.7.1

If $\rho > 0$ and B is non-negative definite, then $(\rho I+B)$ has a bounded inverse and $||(\rho I+B)^{-1}||_2 \leq \rho^{-1}$.

In our case, for equations (4.3.2) and (4.3.3) we need to show that $(3r-\theta)G_1$ and $(3r-\theta)G_2$ are non-negative. This is easily shown by observing that the eigenvalues of G_1 or G_2 are, 0 and 2 of multiplicity of the number of the groups and a single eigenvalue of 1, which are all non-negative. Thus $(3r-\theta)G_1$ and $(3r-\theta)G_2$ are non-negative definite. It is important to draw attention to the fact that equations (4.3.2), (4.3.3) and all similar equations representing other schemes are multiplied by a factor of 3 of the original equations. Thus according to Lemma (4.7.1), from equation (4.3.2) $(I+(r-\theta/3)G_1)^{-1}$ is bounded and $||(I+(r-\theta/3)G_1)^{-1}||_2 \leq 1$. Similarly from equation (4.3.3) we get $||(I+(r-\theta/3)G_2)^{-1}||_2 \leq 1$.

For the stability of these two equations we need that

$$||T_r|| = ||(I+(r-\theta/3)G_1)^{-1}(I-rG_2-\theta/3G_1)|| \leq 1$$

and

$$||T_\ell|| = ||(I+(r-\theta/3)G_2)^{-1}(I-rG_1-\theta/3G_2)|| \leq 1.$$

We have,

$$||(I+(r-\theta/3)G_1)^{-1}(I-rG_2-\theta/3G_1)||_2 \leq ||(I+(r-\theta/3)G_1)^{-1}||_2$$

$$||(I-rG_2-\theta/3G_1)||_2 \leq ||(I-rG_2-\theta/3G_1)||_2$$

= max. modulus of eigenvalues,

where the eigenvalues of $(I-rG_2-\theta/3G_1)$ are

$(1), (1-2\theta/3), (1-\theta/3), (1-2r), (1-2r-2\theta/3), (1-2r-\theta/3), (1-r),$

$(1-r-2\theta/3)$ and $(1-r-\theta/3)$.

Therefore $||1-rG_2-\theta/3G_1||_2 \leq 1$ for $r \leq 1$.

Thus, stability is fulfilled for both equations (4.3.2) and (4.3.3).

For the stability of the single and double sweep processes in equations (4.3.4) and (4.3.6) respectively we need to state the second Lemma of Kellogg [1964, p.849].

Lemma 4.7.2

If $\rho > 0$ and B is non-negative definite, the operator $T(B) = (\rho I - B) \cdot (\rho I + B)^{-1}$ is a bounded operator with $||T(B)||_2 \leq 1$.

Now we write equation (4.3.4) as,

$$(I+(r-\theta/3)G_1)u_{j+1} = (I-rG_2-\theta/3G_1)u_j + b_1 ,$$

$$(I+(r-\theta/3)G_2)u_{j+2} = (I-rG_1-\theta/3G_2)u_{j+1} + b_2 .$$

Eliminating u_{j+1} leads to,

$$u_{j+2} = Tu_j + b_{12} , \qquad (4.7.5)$$

where b_{12} is independent of u and,

$$T = (I+(r-\theta/3)G_2)^{-1}(I-rG_1-\theta/3G_2)(I+(r-\theta/3)G_1)^{-1}(I-rG_2-\theta/3G_1). \tag{4.7.6}$$

Let,

$$\tilde{T} = (I+(r-\theta/3)G_2)T(I+(r-\theta/3)G_2)^{-1} . \tag{4.7.7}$$

The matrix $\tilde{T}$ is similar to T, thus they have the same eigenvalues.

From (4.7.6),

$$\tilde{T} = (I-rG_1-\theta/3G_2)(I+(r-\theta/3)G_1)^{-1}(I-rG_2-\theta/3G_1)(I+(r-\theta/3)G_2)^{-1} , \tag{4.7.8}$$

and

$$||\tilde{T}||_2 = ||(I-rG_1-\theta/3G_2)(I+(r-\theta/3)G_1)^{-1}(I-rG_2-\theta/3G_1)(I+(r-\theta/3)G_2)^{-1}||_2$$

$$\leq ||(I-rG_1-\theta/3G_2)(I+(r-\theta/3)G_1)^{-1}||_2||(I-rG_2-\theta/3G_1)$$

$$(I+(r-\theta/3)G_2)^{-1}||_2 . \tag{4.7.9}$$

As we have shown G_1 and G_2 are non-negative definite and consequently $(r-\theta/3)G_1$, $(r-\theta/3)G_2$, $(rG_1+\theta/3G_2)$ and $(rG_2+\theta/3G_1)$. Thus from Lemma (4.7.2) we obtain the stability of this scheme for $r>0$.

In a closely analogous way, the stability of the double sweep scheme, equation (4.3.6) can be established. We can now write equation (4.3.6) as,

$$\left.\begin{aligned}
(I+(r-\theta/3)G_1)u_{j+1} &= (I-rG_2-\theta/3G_1)u_j + b_1 ,\\
(I+(r-\theta/3)G_2)u_{j+2} &= (I-rG_1-\theta/3G_2)u_{j+1} + b_2 ,\\
(I+(r-\theta/3)G_2)u_{j+3} &= (I-rG_1-\theta/3G_2)u_{j+2} + b_2 ,\\
(I+(r-\theta/3)G_1)u_{j+4} &= (I-rG_2-\theta/3G_1)u_{j+3} + b_1 ,
\end{aligned}\right\} \tag{4.7.10}$$

which can be treated as two single sweeps working in a reverse order of each other. Thus for the first two equations of (4.7.10) we have the previous case of the single sweep scheme. For the last two equations

at (4.7.10), eliminating u_{j+3} gives,

$$u_{j+4} = T'u_{j+2} + b_{34} \ , \tag{4.7.11}$$

where b_{34} is independent of u and,

$$T' = (I+(r-\theta/3)G_1)^{-1}(I-rG_2-\theta/3G_1)(I+(r-\theta/3)G_2)^{-1}$$
$$(I-rG_1-\theta/3G_2) \ . \tag{4.7.12}$$

Let,

$$\tilde{T}' = (I+(r-\theta/3)G_1)T'(I+(r-\theta/3)G_1)^{-1} \ . \tag{4.7.13}$$

The matrix $\tilde{T}'$ is similar to T' and thus has the same eigenvalues as T'.

$$\tilde{T}' = (I-rG_2-\theta/3G_1)(I+(r-\theta/3)G_2)^{-1}(I-rG_1-\theta/3G_2)(I+(r-\theta/3)G_1)^{-1},$$

$$||\tilde{T}'||_2 = ||(I-rG_2-\theta/3G_1)(I+(r-\theta/3)G_2)^{-1}(I-rG_1-\theta/3G_2)(I+(r-\theta/3)G_1)^{-1}$$

$$\leq ||(I-rG_2-\theta/3G_1)(I+(r-\theta/3)G_2)^{-1}||_2||(I-rG_1-\theta/3G_2)$$

$$(I+(r-\theta/3)G_1)^{-1}|| \tag{4.7.14}$$

which is similar to $\tilde{T}$ in equation (4.7.9), thus,

$$||T'||_2 = ||\tilde{T}'||_2 \leq 1 \ .$$

Now eliminating u_{j+1}, u_{j+2} and u_{j+3} in equation (4.7.10) we get,

$$u_{j+4} = T''u_j + b_{11} \ , \tag{4.7.15}$$

where b_{11} is independent of u and,

$$T'' = (I+(r-\theta/3)G_1)^{-1}(I-rG_2-\theta/3G_1)(I+(r-\theta/3)G_2)^{-1}(I-rG_1-\theta/3G_2)$$

$$(I+(r-\theta/3)G_2)^{-1}(I-rG_1-\theta/3G_2)(I+(r-\theta/3)G_1)^{-1}(I-rG_2-\theta/3G_1) \ . \tag{4.7.16}$$

From (4.7.6) and (4.7.12) we find that,

$$T'' = T'.T \ ,$$

or
$$||T''||_2 = ||T'T||_2 \ ,$$
$$||T'||_2||T||_2 \ .$$

Since the eigenvalues of T' and T are the same eigenvalues as $\tilde{T}'$ and $\tilde{T}$ respectively, we get,

$$||T''||_2 \leq ||\tilde{T}'||_2||\tilde{T}||_2 \ ,$$
$$\leq 1 \ .$$

Therefore stability of the double sweep scheme is unconditional for r>0.

In the case of an even number of internal points, i.e. as in (b1) and (b2), we notice that the eigenvalues of $\tilde{G}_1$ are 1 of multiplicity 2, 0 and 2 of multiplicity of the number of the groups involved. Whereas, the eigenvalues of $\tilde{G}_2$ are 0 and 2 of multiplicity of the number of the groups. Therefore, the stability of the schemes SPAGEU and SPAGEC is obtained in the same way as was used in the case of an odd number of points. Thus, these two schemes are unconditionally stable for $r \leq 1$. The single and double sweep schemes of this case are treated in the same way as in the previous case, thus they enjoy unconditional stability for r>0.

4.8 STABILITY OF THE SPLINE ALTERNATING GROUP EXPLICIT METHOD: PERIODIC CASE

In this case although the coefficient matrices are different from those in the previous section, they can be proved to have the same eigenvalues.

Let us take the scheme c1) SPAGEC. Equation (4.4.2) can be written as,

$$u_{j+1} = T_3 \, u_j \ , \tag{4.8.1}$$

where,

$$T_3 = (I+(r-\theta/3)\tilde{G}_2)^{-1}(I-r\tilde{\tilde{G}}_1-\theta/3\tilde{G}_2) \ ,$$

where we have divided by 3 to put into original form.

The eigenvalues of $\tilde{G}_2$ are discussed in the previous section. For the matrix $\tilde{\tilde{G}}_1$, it can be transformed to $\tilde{G}_2$ by using the permutation matrix P and its inverse P^{-1},

$$P \, \tilde{\tilde{G}}_1 P^{-1} = \tilde{G}_2 \ , \tag{4.8.2}$$

where $P^{-1}=P^T$ and,

$$P = \begin{bmatrix} 0 & 1 & & & \\ 0 & 0 & 1 & & \\ \vdots & & \ddots & \ddots & \\ 0 & & & & 1 \\ 1 & 0 & \cdots & \cdots & 0 \end{bmatrix} \ .$$

Thus $\tilde{\tilde{G}}_1$ is similar to $\tilde{G}_2$ and has the same eigenvalues. Therefore, following the same analysis as in the previous section leads to the same condition of stability, that is unconditional stability for $r \leq 1$. This concludes that the stability of the single and double sweep schemes in the periodic case is also unconditional for $r>0$.

For the case of the odd number of intervals, where we have the matrices $\hat{G}_1$, $\hat{G}_2$, $\hat{\hat{G}}_1$, $\hat{\hat{G}}_2$, $\check{\check{G}}_1$ and $\check{\check{G}}_2$, apart from the 2×2 matrices $G^{(i)}$ that are in these matrices, a 3×3 group is produced and is of the form,

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

or can be reformed so. This group can be shown to have the eigenvalues 0,1 and 2. This leads us to follow the same stability analysis as for the previous cases.

4.9 STABILITY OF THE SPLINE ALTERNATING GROUP EXPLICIT METHOD FOR DERIVATIVE BOUNDARY CONDITIONS CASE

In this case the matrices arising are not different from the cases discussed earlier except for G_1^* in e1) and e2), where,

$$G_1^* = \begin{bmatrix} 1 & -1 & & & & \\ & G^{(1)} & & & & \\ & & G^{(2)} & & \text{O} & \\ & & & \ddots & & \\ & \text{O} & & & G^{(n-1)/2} & \\ & & & & -1 & 1 \end{bmatrix}$$

It can be easily shown that a similarity transformation using a permutation matrix can transform G_1^* to $\tilde{G}_2$. Thus, the analysis is the same as in the last two sections and the conclusion is that the schemes SPAGEL, SPAGER, SPAGEC and SPAGEU are stable for $r \leq 1$ and the schemes SSPAGE and DSPAGE are stable for $r > 0$.

4.10 COMPUTATIONAL COMPLEXITY OF THE SPLINE ALTERNATING GROUP EXPLICIT METHOD

The simplicity of this method is shown through its easy form and by its inexpensive cost. The number of arithmetic operations involved to evaluate one point of the solution at one time step is as follows,

	addition	multiplication	division
(grouped point)	3	4	1 .

For an ungrouped point the number of arithmetic operations depends on the type of boundary conditions, but for a general form of a single ungrouped point it is the same as for a grouped point.

4.11 THE RELATIONSHIP BETWEEN THE SPLINE ALTERNATING GROUP EXPLICIT METHOD AND SOME OTHER METHODS

If we recall our original SPAGE equations,

$$(3-\theta+3r)u_{i,j+1}+(\theta-3r)u_{i+1,j+1} = \theta u_{i+1,j}+(3-\theta-3r)u_{i,j}+3ru_{i-1,j}\ , \tag{4.11.1}$$

and,

$$(3-\theta+3r)u_{i,j+1}+(\theta-3r)u_{i-1,j+1} = 3ru_{i+1,j}+(3-\theta-3r)u_{i,j}+\theta u_{i-1,j}\ , \tag{4.11.2}$$

then by putting $\theta=0$, produces Saulyev's formulae (3.2.13) and (3.2.14) respectively. Therefore, our SPAGE formulae reduces to the Group Explicit formulae of Abdullah (1983).

Also, by putting $\theta=3r$ both equations (4.11.1) and (4.11.2) reduce to the classical explicit formula. In fact, adding the two equations above produces the three point two time level formula,

$$(\theta-3r)u_{i-1,j+1}+2(3-\theta+3r)u_{i,j+1}+(\theta-3r)u_{i+1,j+1} = (\theta+3r)u_{i-1,j} + 2(3-\theta-3r)u_{i,j}+(\theta+3r)u_{i+1,j}\ , \tag{4.11.3}$$

which reduces to the Crank-Nicolson formula when $\theta=0$. It reduces to the well known explicit formula for $\theta=3r$ and to the fully implicit formula for $\theta=-3r$. Putting $\theta=1$, this formula produces a particular form of the cubic splines formula (due to Papamichael and Whiteman (1973)) with optimum truncation error, see equation (3.2.12). These schemes are shown in Figure (4.11.1).

Another way of looking at the relationship between the SPAGE scheme and the other schemes mentioned above is through the mechanical transposition of some of the elements of the coefficient matrices. We

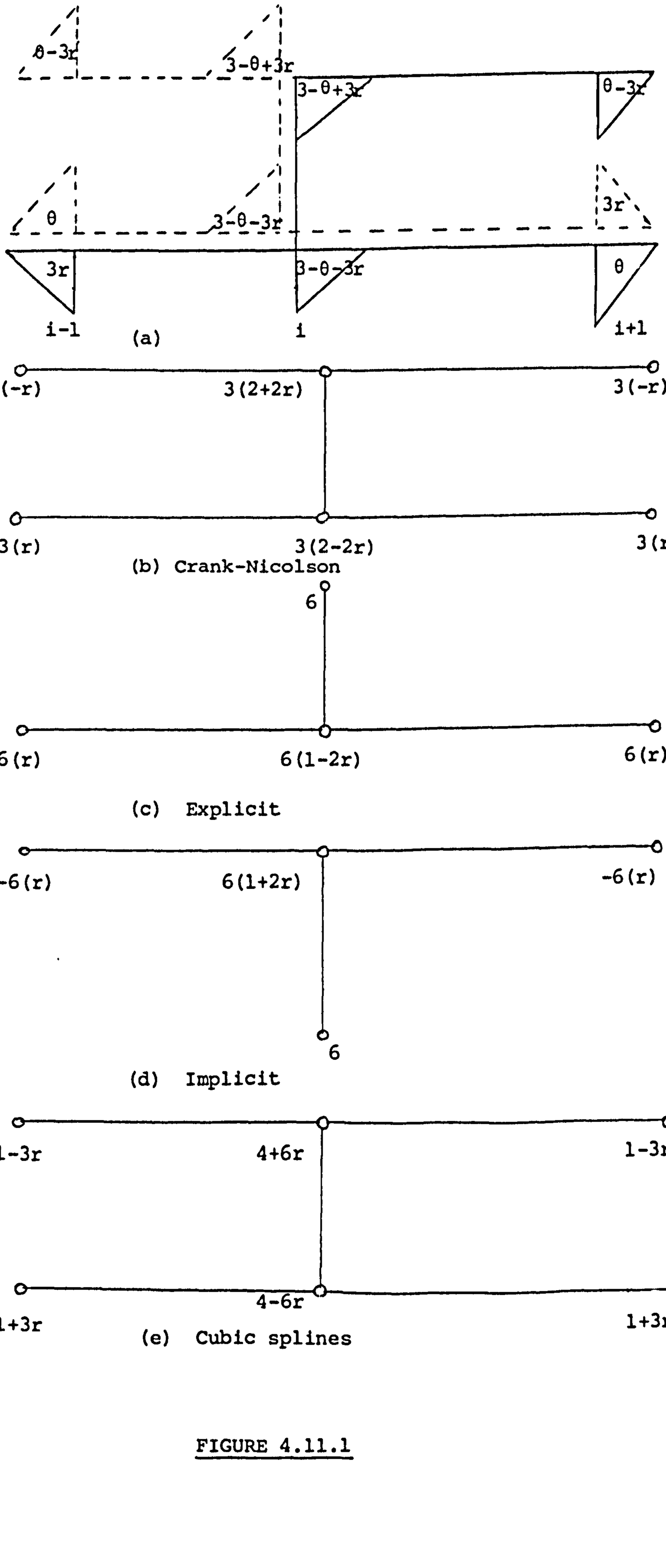

θ-3r
3-θ+3r
3-θ+3r
θ-3r
j+1
θ
3-θ-3r
3r
j
3r
3-θ-3r
θ
i-1
i
i+1
(a)
3(-r)
3(2+2r)
3(-r)
3(r)
3(2-2r)
3(r)
(b) Crank-Nicolson
6
6(r)
6(1-2r)
6(r)
(c) Explicit
-6(r)
6(1+2r)
-6(r)
6
(d) Implicit
1-3r
4+6r
1-3r
4-6r
1+3r
1+3r
(e) Cubic splines

FIGURE 4.11.1

take, as an example, the Crank-Nicolson case, i.e. when $\theta=0$ in the SPAGE schemes. Consider the SPAGER scheme, this is written when $\theta=0$, as,

$$(3I+3rG_1)u_{j+1} = (3I-3rG_2)u_j + b_1 , \qquad (4.11.4)$$

with,

$$G_1 = \begin{bmatrix} 1 & -1 & & & & \\ -1 & 1 & & & & \\ & & 0 & & & \\ & & & 1 & -1 & \\ & & & -1 & 1 & \\ & & & & & 1 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 1 & & & & & \\ & 1 & -1 & & & \\ & -1 & 1 & & & \\ & & & 0 & & \\ & & & & 1 & -1 \\ & & & & -1 & 1 \end{bmatrix}$$

and $b_1^T = [3ru_{0,j}, 0, 0, \ldots, 0, \theta u_{n,j} - (\theta-3r)u_{n,j+1}]$.

The matrix form of the Crank-Nicolson method is as follows,

$$\frac{3}{2}\begin{bmatrix} 2+2r & -r & & & & \\ -r & 2+2r & -r & & & \\ & -r & 2+2r & -r & & \\ & & & 0 & & \\ & & & -r & 2+2r & -r \\ & & & & -r & 2+2r \end{bmatrix} \begin{bmatrix} u_1 \\ \\ \\ \\ \\ u_{n-1} \end{bmatrix}_{j+1} =$$

$$\frac{3}{2}\begin{bmatrix} 2-2r & r & & & \\ r & 2-2r & r & & \\ & r & 2-2r & r & \\ & & r & 2-2r & r \\ & & & r & 2-2r \end{bmatrix}\begin{bmatrix} u_1 \\ \\ \\ \\ u_{n-1} \end{bmatrix}_j + b_2. \qquad (4.11.5)$$

where, $b_2^T = [\frac{3r}{2}(u_{0,j+1}+u_{0,j}), 0, 0, \ldots, 0, \frac{3r}{2}(u_{n,j+1}+u_{n,j})]$.

The process is to transfer the elements outside the diagonal boxes (in circles) from the matrix of j+1th time level to the matrix of the jth time level. The resulting system is similar to the system of SPAGER and is written as,

$$(3I+3r\hat{G}_1)u_{j+1}^{(k+1)} = (3I-3r\hat{G}_1)u_j+(3rF_1)u_j+(3rF_1)u_{j+1}^{(k)} , \qquad (4.11.6)$$

where,

$$\hat{G}_1 = \begin{bmatrix} 1 & -\frac{1}{2} & & & \\ -\frac{1}{2} & 1 & & & \\ & & 1 & -\frac{1}{2} & \\ & & -\frac{1}{2} & 1 & \\ & & & & 1 \end{bmatrix}, \quad F_1 = \begin{bmatrix} 0 & & & & \\ & 0 & \frac{1}{2} & & \\ & \frac{1}{2} & 0 & & \\ & & & 0 & \frac{1}{2} \\ & & & \frac{1}{2} & 0 \end{bmatrix}$$

Analogously, the Crank-Nicolson scheme can be converted to a SPAGEL-like system and can be written as,

$$(3I+3r\hat{G}_2)u_{j+1}^{(k+1)} = (3I-3r\hat{G}_2)u_j+(3rF_2)u_j+(3rF_2)u_{j+1}^{(k)} , \qquad (4.11.7)$$

where,

$$\hat{G}_2 = \begin{bmatrix} 1 & & & & & \\ & 1 & -\frac{1}{2} & & & \\ & -\frac{1}{2} & 1 & & & \\ & & & \ddots & & \\ & & & & 1 & -\frac{1}{2} \\ & & & & -\frac{1}{2} & 1 \end{bmatrix}, \quad F_2 = \begin{bmatrix} 0 & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & & & \\ & & \ddots & & \\ & & & 0 & \frac{1}{2} \\ & & & \frac{1}{2} & 0 \end{bmatrix}$$

This suggests using equation (4.11.6) and (4.11.7) alternatively as a single iteration.

The same strategy can be applied to the cubic splines formula.

In the SPAGER scheme let $\theta=1$, this leads to the equation,

$$[3I+(3r-1)G_1]u_{j+1} = [3I-G_1-3rG_2]u_j \ , \qquad (4.11.8)$$

where G_1 and G_2 are as described earlier. The cubic spline system is given as follows,

$$(2+3r)u_{i,j+1}+(\frac{1-3r}{2})(u_{i-1,j+1}+u_{i+1,j+1}) = (2-3r)u_{i,j} + (\frac{1+3r}{2})(u_{i-1,j}+u_{i+1,j}) \ , \qquad (4.11.9)$$

$$i=1,\ldots,n-1,$$

or

$$\begin{bmatrix} 2+3r & \frac{1-3r}{2} & & & & \\ \frac{1-3r}{2} & 2+3r & \frac{1-3r}{2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \frac{1-3r}{2} & 2+3r & \frac{1-3r}{2} & \\ & & & \frac{1-3r}{2} & 2+3r & \end{bmatrix} \begin{bmatrix} u_1 \\ \\ \\ \\ u_{n-1} \end{bmatrix}_{j+1} =$$

$$\begin{bmatrix} 2-3r & \frac{1+3r}{2} & & & & \\ \frac{1+3r}{2} & 2-3r & \frac{1+3r}{2} & & & \\ & \frac{1+3r}{2} & 2-3r & \frac{1+3r}{2} & & \\ & & \frac{1+3r}{2} & 2-3r & \frac{1+3r}{2} & \\ & & & \frac{1+3r}{2} & 2-3r & \frac{1+3r}{2} \\ & & & & \frac{1+3r}{2} & 2-3r \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_{n-1} \end{bmatrix}_j + b_3 ,$$

(4.11.10)

where, $b_3^T = [\frac{1+3r}{2}u_{0,j} + \frac{1-3r}{2}u_{0,j+1}, 0, 0, \ldots, 0, \frac{1+3r}{2}u_{n,j} + \frac{1-3r}{2}u_{n,j+1}]$.

Following the same process of transferring the elements from the lefthand side coefficient matrix to the righthand side matrix to produce,

$$(3I+(3r-1)\hat{G}_1)u_{j+1}^{(k+1)} = (3I-(3r-1)\hat{G}_1)u_j + (1+3r)F_1u_j + (1-3r)F_1u_{j+1}^{(k)} .$$

(4.11.11)

Different partitioning to the above matrices produces again a system similar to a SPAGEL system which is written as,

$$(3I+(3r-1)\hat{G}_2)u_{j+1}^{(k+1)} = (3I-(3r-1)\hat{G}_2)u_j + (1+3r)F_2u_j + (1-3r)F_2u_{j+1}^{(k)} .$$

(4.11.12)

Again equations (4.11.11) and (4.11.12) form a single iteration of a sequence of iterations that can be carried on until convergence is achieved. For the convergence of these techniques, we need, in equations (4.11.6) and (4.11.7), that the eigenvalues of,

$$(3I+3r\hat{G}_i^{-1})3rF_i < 1 , \quad i=1,2,$$

and in equation (4.11.11) and (4.11.12), that the eigenvalues of

$$(3I+(3r-1)\hat{G}_i^{-1})(1-3r)F_i < 1, \quad i=1,2, .$$

This is guaranteed since the eigenvalues of F_i, $i=1,2$ are always < 1 .

In the next chapter, this strategy is further developed.

4.12 NUMERICAL RESULTS

On three different problems, the schemes described in this chapter are tested and their results compared with the exact solutions in terms of the absolute error, $|e_i|$, and the percentage error, $\%e_i$, which are calculated as follows.

$$|e_{i,j}| = |u_{i,j} - U_{i,j}|, \tag{4.12.1}$$

and

$$\%e_i = \frac{|e_{i,j}|}{U_{i,j}} \times 100 , \tag{4.12.2}$$

where $u_{i,j}$ represents the numerical solution and $U_{i,j}$ represents the exact solution.

In the three problems, numerical results were obtained by dividing the domain into both even and odd number of intervals. Also different values of r were used (0.1), (0.5), (1.0), (1.5), (0.121), (0.605), (1.21) and (1.815). This was done for a different number of time steps 10, 50 and 100. Also because of the four steps of the double sweep schemes we used 12 and 52 time steps (instead of 10 and 50) for these schemes only, assuming that two extra steps would not show a big difference in error and percentage error comparison.

Because of the large number of tables that can be involved, we show only some of them.

Example 1

The model problem below is considered,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \quad , 0 \leq x \leq 1$$

with the initial conditions,

$$U(x,0) = 4x(1-x) \ , \ 0 \leq x \leq 1$$

and the boundary conditions,

$$U(0,t) = U(1,t) = 0 \ , \quad t \geq 0 \ . \qquad (4.12.3)$$

The exact solution is given by,

$$U(x,t) = \frac{32}{\pi^3} \sum_{k=1,3,5,\ldots}^{\infty} \frac{1}{k^3} e^{-k^2\pi^2 t} \sin(k\pi x), \qquad (4.12.4)$$

[SAUL'YEV, 1964, p.34].

For this example the results are shown in Tables [4.12.1-4.12.6].

Example 2

In this example a problem with derivative boundary conditions is solved. The problem is given by,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \ , \ 0 \leq x \leq 1 \ ,$$

with the initial condition,

$$U(x,0) = \sin x + \cos x, \ x \geq 0$$

and the boundary conditions,

$$\frac{\partial U}{\partial x}(0,t) = e^{-t} \ , \ t \geq 0,$$

$$\frac{\partial U}{\partial x}(1,t) = e^{-t}(\cos(1.0) - \sin(1.0)), \ t \geq 0 \ , \qquad (4.12.5)$$

where the analytical solution is given by [MITCHELL and WAIT, 1977,p.179].

$$U(x,t) = e^{-t}(\sin x + \cos x) \ , \quad 0 \leq x \leq 1, \quad t \geq 0 \ . \qquad (4.12.6)$$

In this case, the derivatives at the boundaries x=0.0 and x=1.0 are approximated by the first order central difference formula.

The numerical results are shown in Tables [4.12.7-4.12.12].

Example 3

The problem considered here has periodic boundary conditions.

The equation is,

$$\left.\begin{aligned} &\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + 10(1-x)xt \ , \ 0 \leq x \leq 1 \\ &\text{with the initial condition,} \\ &\qquad U(x,0) = x(1-x) \\ &\text{and the boundary conditions,} \\ &\qquad U(0,t) = U(1,t), \ \frac{\partial u}{\partial x}(0,t) = \frac{\partial U}{\partial x}(1,t) \ . \end{aligned}\right\} \qquad (4.12.7)$$

The exact solution of this problem is,

$$U(x,t) = \frac{1+5t^2}{6} - \frac{5}{8} \sum_{n=1}^{\infty} \frac{\cos 2n\pi x}{(n\pi)^6} \{4n^2\pi^2 t - 1 + e^{-4n^2\pi^2 t}\}$$

$$- \sum_{n=1}^{\infty} \frac{e^{-4n^2\pi^2 t}}{n^2\pi^2} \cos 2n\pi x .$$

In this example we used the second treatment (ii) for the single boundary points, as it almost always gives more accurate results than the first treatment (i) in Section 4.4.

The numerical results are shown in Tables [4.12.13-4.12.16].

In Table (4.12.7) we compare the SPAGE method with the C.N. and the G.E. method with r=0.1 and for 100 time steps. Also in Table (4.12.18) a comparison made with r=1.0 and for 100 time steps. All comparisons are made in terms of the percentage errors.

$\Delta x = 1/10$ $r = 0.1$

$\Delta t = 0.001$ No. of steps=10 time=0.01

Scheme \ Δx x		0	1	2	3	4	5	6	7	8	9
SPAGEL	e		1703×10^{-6}	198×10^{-6}	433×10^{-6}	932×10^{-6}	1151×10^{-6}	932×10^{-6}	760×10^{-6}	198×10^{-6}	1645×10^{-6}
$\theta=0.6$	e%		0.5636	0.0351	0.0570	0.1061	0.1252	0.1061	0.1000	0.0351	0.5447
SPAGER	e		1645×10^{-6}	198×10^{-6}	760×10^{-6}	932×10^{-6}	1151×10^{-6}	932×10^{-6}	433×10^{-6}	198×10^{-6}	1703×10^{-6}
$\theta=0.6$	e%		0.5447	0.0351	0.1000	0.1061	0.1252	0.1061	0.0570	0.0351	0.5636
SSPAGEV	e		535×10^{-6}	13×10^{-6}	238×10^{-6}	865×10^{-6}	1111×10^{-6}	793×10^{-6}	343×10^{-6}	562×10^{-6}	4×10^{-6}
$\theta=0.7$	e%		0.1771	0.0024	0.0313	0.0984	0.1209	0.0903	0.0452	0.0996	0.0014
12 steps DSPAGEV	e		95×10^{-6}	10×10^{-6}	519×10^{-6}	837×10^{-6}	1098×10^{-6}	866×10^{-6}	399×10^{-6}	174×10^{-6}	62×10^{-6}
$\theta=0.6$	e%		0.0324	0.0018	0.0697	0.0970	0.1216	0.1004	0.0536	0.0315	0.0212
Exact solution			0.302108	0.563997	0.759858	0.879069	0.918840	0.879069	0.759858	0.563997	0.302108

TABLE 1: Example 1 - Dirichlet Boundary Condition

$\Delta x = 1/10$ $r = 0.5$
$\Delta t = 0.005$ No. of steps=50 time=0.25

Scheme \ Δx x		0	1	2	3	4	5	6	7	8	9
SPAGEL	e		357×10^{-6}	69×10^{-6}	153×10^{-6}	97×10^{-6}	89×10^{-6}	97×10^{-6}	336×10^{-6}	69×10^{-6}	433×10^{-6}
θ=0.6	%e		1.3254	0.1353	0.2169	0.1170	0.1018	0.1170	0.4753	0.1353	1.6064
SPAGER	e		433×10^{-6}	69×10^{-6}	336×10^{-6}	97×10^{-6}	89×10^{-6}	97×10^{-6}	153×10^{-6}	69×10^{-6}	357×10^{-6}
θ=0.6	%e		1.6064	0.1353	0.4753	0.1170	0.1018	0.1170	0.2169	0.1353	1.3254
SSPAGEV	e		126×10^{-6}	1436×10^{-6}	1547×10^{-6}	2215×10^{-6}	2228×10^{-6}	2009×10^{-6}	1940×10^{-6}	894×10^{-6}	763×10^{-6}
θ=0.0	%e		0.4659	2.7986	2.1908	2.6695	2.5540	2.4205	2.7476	1.7430	2.8273
52 steps DSPAGEV	e		239×10^{-6}	98×10^{-6}	480×10^{-6}	259×10^{-6}	362×10^{-6}	443×10^{-6}	119×10^{-6}	491×10^{-6}	269×10^{-6}
θ=0.0	%e		0.9773	0.2103	0.7503	0.3439	0.4578	0.5890	0.1866	1.0559	1.1001
Exact solution			0.026971	0.051299	0.070602	0.082989	0.087245	0.082989	0.070602	0.051299	0.026971

TABLE 2: Example 1 - Dirichlet Boundary Conditions

$\Delta x=1/10$ $r=1.0$

$\Delta t=0.01$ No. of steps=100 time=1.0

Scheme \ Δx x		0	1	2	3	4	5	6	7	8	9
SPAGEL	e		$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$
θ=0.6	%e		3.7799	0.1125	0.8085	0.0942	0.0859	0.0942	1.0351	0.1125	4.0291
SPAGER	e		$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$
θ=0.6	%e		4.0291	0.1125	1.0351	0.0942	0.0859	0.0942	0.8085	0.1125	3.7799
SSPAGE	e										
θ=	%e										
DSPAGEV	e		$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$
θ=0.0	%e		8.7631	4.0819	6.9425	5.8146	6.3823	6.6521	5.7754	9.0855	1.0181
Exact solution			0.000016	0.000031	0.000043	0.000050	0.000053	0.000050	0.000043	0.000031	0.000016

TABLE 3: Example 1 - Dirichlet Boundary Conditions

$\Delta x = 1/11$ $r = 0.121$
$\Delta t = 0.001$ No. of steps=100 time=0.1

Scheme \ Δx x		0	1	2	3	4	5	6	7	8	9	10
SPAGEU	e		351×10^{-6}	7×10^{-6}	241×10^{-6}	3×10^{-6}	36×10^{-6}	36×10^{-6}	3×10^{-6}	241×10^{-6}	7×10^{-6}	351×10^{-6}
$\theta=0.6$	%e		0.3246	0.0033	0.0831	0.0008	0.0094	0.0094	0.0008	0.0831	0.0033	0.3246
SPAGEC	e		417×10^{-6}	129×10^{-6}	191×10^{-6}	109×10^{-6}	53×10^{-6}	53×10^{-6}	109×10^{-6}	191×10^{-6}	129×10^{-6}	417×10^{-6}
$\theta=0.7$	%e		0.3854	0.0619	0.0658	0.0313	0.0140	0.0140	0.0313	0.0658	0.0619	0.3854
SSPAGEOD	e		51×10^{-6}	107×10^{-6}	31×10^{-6}	103×10^{-6}	29×10^{-6}	29×10^{-6}	103×10^{-6}	31×10^{-6}	107×10^{-6}	51×10^{-6}
$\theta=0.6$	%e		0.0467	0.0515	0.0107	0.0296	0.0076	0.0076	0.0296	0.0107	0.0515	0.0467
DSPAGEOD	e		27×10^{-6}	72×10^{-6}	23×10^{-6}	63×10^{-6}	3×10^{-6}	3×10^{-6}	63×10^{-6}	23×10^{-6}	72×10^{-6}	27×10^{-6}
$\theta=0.6$	%e		0.0246	0.0349	0.0078	0.0180	0.0008	0.0008	0.0180	0.0078	0.0349	0.0246
Exact solution			0.108200	0.207624	0.290212	0.349268	0.380007	0.380007	0.349268	0.290212	0.207624	0.108200

TABLE 4: Example 1 - Dirichlet Boundary Conditions

$\Delta x = 1/11$ $r = 0.605$
$\Delta t = 0.005$ No. of steps = 100 time = 0.5

Scheme	Δx x	0	1	2	3	4	5	6	7	8	9	10
SPAGEU	e		20×10^{-6}	16×10^{-6}	2×10^{-6}	23×10^{-6}	17×10^{-6}	17×10^{-6}	23×10^{-6}	2×10^{-6}	16×10^{-6}	20×10^{-6}
θ=0.8	%e		0.9421	0.4041	0.0338	0.3429	0.2326	0.2326	0.3429	0.0338	0.4041	0.9421
SPAGEC	e		39×10^{-6}	8×10^{-6}	20×10^{-6}	9×10^{-6}	$<10^{-6}$	$<10^{-6}$	9×10^{-6}	20×10^{-6}	8×10^{-6}	39×10^{-6}
θ=0.3	%e		1.8568	0.1889	0.3577	0.1329	0.0025	0.0025	0.1329	0.3577	0.1889	1.8568
SSPAGEOD	e		174×10^{-6}	282×10^{-6}	463×10^{-6}	528×10^{-6}	594×10^{-6}	594×10^{-6}	528×10^{-6}	463×10^{-6}	282×10^{-6}	174×10^{-6}
θ=0.0	%e		8.3541	7.0732	8.2866	7.8534	8.1290	8.1290	7.8534	8.2866	7.0732	8.3541
DSPAGEOD	e		51×10^{-6}	35×10^{-6}	115×10^{-6}	101×10^{-6}	128×10^{-6}	128×10^{-6}	101×10^{-6}	115×10^{-6}	35×10^{-6}	51×10^{-6}
θ=0.0	%e		2.4340	0.8776	2.0531	1.5023	1.7561	1.7561	1.5023	2.0531	0.8776	2.4340
Exact solution			0.002081	0.003994	0.005582	0.006718	0.007310	0.007310	0.006718	0.005582	0.003994	0.002081

TABLE 5: Example 1 - Dirichlet Boundary Conditions

$\Delta x = 1/11$ $r = 1.21$

$\Delta t = 0.01$ No. of steps=100 time=1.0

Scheme / Δx x		0	1	2	3	4	5	6	7	8	9	10
SPAGE	e											
θ=	%e											
SPAGE	e											
θ=	%e											
SSPAGE	e											
θ=	%e											
DSPAGEOD	e		1×10^{-6}	$<10^{-6}$	1×10^{-6}	1×10^{-6}	2×10^{-6}	2×10^{-6}	1×10^{-6}	1×10^{-6}	$< 10^{-6}$	1×10^{-6}
θ=0.2	%e		4.4799	0.4006	3.1211	1.9549	4.1581	4.1581	1.9549	3.1211	0.4006	4.4799
Exact solution			15×10^{-6}	29×10^{-6}	40×10^{-6}	48×10^{-6}	52×10^{-6}	52×10^{-6}	48×10^{-6}	40×10^{-6}	29×10^{-6}	15×10^{-6}

TABLE 6: Example 1 - Dirichlet Boundary Conditions

$\Delta x = 1/10$ $r = 0.1$

$\Delta t = 0.001$ No. of steps=100 time=0.1

Schemes	Δx x	0	1	2	3	4	5	6	7	8	9	10
SPAGER	e	496×10^{-6}	312×10^{-6}	234×10^{-6}	112×10^{-6}	73×10^{-6}	9×10^{-6}	$<10^{-6}$	14×10^{-6}	2×10^{-6}	26×10^{-6}	51×10^{-6}
θ=0.0	%e	0.0548	0.0315	0.0220	0.0099	0.0062	0.0007	0.0000	0.0011	0.0002	0.0020	0.0041
SPAGEL	e	431×10^{-6}	338×10^{-6}	188×10^{-6}	135×10^{-6}	46×10^{-6}	27×10^{-6}	8×10^{-6}	1×10^{-6}	11×10^{-6}	35×10^{-6}	88×10^{-6}
θ=0.0	%e	0.0477	0.0341	0.0177	0.0119	0.0039	0.0022	0.0006	0.0001	0.0008	0.0028	0.0070
SSPAGEV	e	468×10^{-6}	339×10^{-6}	218×10^{-6}	137×10^{-6}	68×10^{-6}	31×10^{-6}	7×10^{-6}	4×10^{-6}	16×10^{-6}	42×10^{-6}	81×10^{-6}
θ=0.0	%e	0.0518	0.0342	0.0205	0.0121	0.0058	0.0025	0.0005	0.0004	0.0012	0.0032	0.0065
DSPAGEV	e	474×10^{-6}	333×10^{-6}	223×10^{-6}	133×10^{-6}	71×10^{-6}	28×10^{-6}	8×10^{-6}	3×10^{-6}	15×10^{-6}	41×10^{-6}	79×10^{-6}
θ=0.0	%e	0.0524	0.0336	0.0209	0.0117	0.0060	0.0023	0.0006	0.0003	0.0012	0.0032	0.0063
Exact solution		0.904837	0.990650	1.066564	1.131822	1.185771	1.227872	1.257704	1.274970	1.2794971	1.271239	1.250280

TABLE 7: Example 2 - Neumann Boundary Conditions

$\Delta x = 1/10$ $r = 0.5$

$\Delta t = 0.005$ No. of steps = 100 time = 0.5

Scheme \ Δx \ x		0	1	2	3	4	5	6	7	8	9	10
SPAGER	e	852×10^{-6}	603×10^{-6}	635×10^{-6}	411×10^{-6}	460×10^{-6}	329×10^{-6}	344×10^{-6}	274×10^{-6}	264×10^{-6}	276×10^{-6}	254×10^{-6}
$\theta=0.0$	%e	0.1405	0.0908	0.0889	0.0582	0.0578	0.0400	0.0396	0.0320	0.0308	0.0324	0.0303
SPAGEL	e	638×10^{-6}	693×10^{-6}	488×10^{-6}	531×10^{-6}	387×10^{-6}	419×10^{-6}	344×10^{-6}	363×10^{-6}	360×10^{-6}	365×10^{-6}	434×10^{-6}
$\theta=0.0$	%e	0.1051	0.1044	0.0682	0.0700	0.0487	0.0509	0.0408	0.0424	0.0419	0.0428	0.0518
SSPAGEV	e	1781×10^{-6}	1837×10^{-6}	1634×10^{-6}	1679×10^{-6}	1536×10^{-6}	1565×10^{-6}	1485×10^{-6}	1495×10^{-6}	1479×10^{-6}	1468×10^{-6}	1518×10^{-6}
$\theta=0.0$	%e	0.2936	0.2767	0.2286	0.2213	0.1932	0.1901	0.1761	0.1749	0.1724	0.1723	0.1811
DSPAVEV	e	1407×10^{-6}	1173×10^{-6}	1221×10^{-6}	1040×10^{-6}	1073×10^{-6}	957×10^{-6}	975×10^{-6}	927×10^{-6}	930×10^{-6}	954×10^{-6}	935×10^{-6}
$\theta=0.0$	%e	0.2320	0.1767	0.1707	0.1371	0.1350	0.1163	0.1156	0.1084	0.1084	0.1119	0.1116
Exact solution		0.606531	0.664053	0.714939	0.758683	0.794846	0.823067	0.843064	0.854638	0.857672	0.852137	0.838088

TABLE 8: Example 2 - Neumann Boundary Conditions

$\Delta x = 1/10$ $r = 1.0$

$\Delta t = 0.01$ No. of steps = 100 time = 1.0

Scheme \ Δx x		0	1	2	3	4	5	6	7	8	9	10
SPAGER	e	990×10^{-6}	747×10^{-6}	839×10^{-6}	648×10^{-6}	708×10^{-6}	579×10^{-6}	606×10^{-6}	544×10^{-6}	537×10^{-6}	546×10^{-6}	504×10^{-6}
$\theta=0.0$	%e	0.2692	0.1854	0.1934	0.1408	0.1469	0.1161	0.1185	0.1050	0.1032	0.1056	0.0992
SPAGEL	e	800×10^{-6}	926×10^{-6}	729×10^{-6}	827×10^{-6}	691×10^{-6}	758×10^{-6}	688×10^{-6}	724×10^{-6}	723×10^{-6}	725×10^{-6}	794×10^{-6}
$\theta=0.0$	%e	0.2174	0.2299	0.1680	0.1798	0.1432	0.1519	0.1346	0.1396	0.1391	0.1402	0.1563
SSPAGEV	e	8014×10^{-6}	8330×10^{-6}	7992×10^{-6}	8240×10^{-6}	8005×10^{-6}	8167×10^{-6}	8035×10^{-6}	8098×10^{-6}	8071×10^{-6}	8029×10^{-6}	8107×10^{-6}
$\theta=0.0$	%e	2.1785	2.0681	1.8430	1.7907	1.6605	1.6359	1.5714	1.5621	1.5514	1.5534	1.5948
DSPAGEV	e	2556×10^{-6}	2235×10^{-6}	2429×10^{-6}	2184×10^{-6}	2321×10^{-6}	2162×10^{-6}	2233×10^{-6}	2168×10^{-6}	2190×10^{-6}	2229×10^{-6}	2157×10^{-6}
$\theta=0.4$	%e	0.6948	0.5550	0.5600	0.4747	0.4815	0.4330	0.4367	0.4182	0.4211	0.4312	0.4242
Exact solution		0.367879	0.402768	0.433633	0.460164	0.482098	0.499215	0.511344	0.518364	0.520205	0.516847	0.508326

TABLE 9: Example 2 - Neumann Boundary Conditions

$\Delta x = 1/11$ r=0.121
$\Delta t = 0.001$ No. of steps=100 time=0.1

Scheme	Δx x	0	1	2	3	4	5	6	7	8	9	10	11
SPAGEU	e	355×10^{-6}	291×10^{-6}	169×10^{-6}	131×10^{-6}	52×10^{-6}	36×10^{-6}	4×10^{-6}	4×10^{-6}	10×10^{-6}	$< 10^{-6}$	24×10^{-6}	42×10^{-6}
θ=0.0	%e	0.0392	0.0296	0.0161	0.0117	0.0044	0.0029	0.004	0.003	0.0008	0.0000	0.0019	0.0033
SPAGEC	e	414×10^{-6}	268×10^{-6}	214×10^{-6}	112×10^{-6}	84×10^{-6}	24×10^{-6}	17×10^{-6}	4×10^{-6}	4×10^{-6}	16×10^{-6}	34×10^{-6}	76×10^{-6}
θ=0.0	%e	0.0458	0.0273	0.0203	0.0100	0.0072	0.0020	0.0014	0.0003	0.0003	0.0012	0.0027	0.0061
SSPAGEOD	e	391×10^{-6}	296×10^{-6}	200×10^{-6}	138×10^{-6}	79×10^{-6}	46×10^{-6}	19×10^{-6}	11×10^{-6}	11×10^{-6}	22×10^{-6}	44×10^{-6}	71×10^{-6}
θ=0.0	%e	0.0432	0.0301	0.0190	0.0124	0.0067	0.0038	0.0015	0.0009	0.0008	0.0017	0.0034	0.0057
DSPAGEOD	e	398×10^{-6}	289×10^{-6}	206×10^{-6}	132×10^{-6}	82×10^{-6}	42×10^{-6}	21×10^{-6}	9×10^{-6}	10×10^{-6}	22×10^{-6}	42×10^{-6}	72×10^{-6}
θ=0.0	%e	0.0440	0.0294	0.0195	0.0119	0.0070	0.0035	0.0017	0.0007	0.0008	0.0017	0.0033	0.0058
Exact solution		0.94837	0.983246	1.053534	1.115121	1.167498	1.210233	1.242974	1.265448	1.277472	1.278940	1.269857	1.250280

TABLE 10: Example 2 - Neumann Boundary Conditions

$\Delta x = 1/11$ $r = 1.21$
$\Delta t = 0.01$

DSPAGEOD Scheme

No. of steps \ Δx x		0	1	2	3	4	5	6	7	8	9	10	11
12	e	994×10^{-6}	275×10^{-6}	711×10^{-6}	215×10^{-6}	598×10^{-6}	215×10^{-6}	456×10^{-6}	273×10^{-6}	349×10^{-6}	342×10^{-6}	270×10^{-6}	441×10^{-6}
$\theta=0.4$	%e	0.1121	0.0285	0.746	0.0197	0.0523	0.0181	0.0374	0.0220	0.0279	0.0273	0.0217	0.0360
Exact solution		0.886920	0.963776	0.1032672	1.093040	1.144380	1.186269	1.218361	1.240391	1.252177	1.253621	1.244712	1.225523
52	e	1693×10^{-6}	1208×10^{-6}	1536×10^{-6}	1154×10^{-6}	1401×10^{-6}	1132×10^{-6}	1284×10^{-6}	1150×10^{-6}	1191×10^{-6}	1179×10^{-6}	1125×10^{-6}	1238×10^{-6}
$\theta=0.4$	%e	0.2847	0.1871	0.2219	0.1575	0.1827	0.1423	0.1573	0.1383	0.1419	0.1403	0.1348	0.1508
Exact solution		0.594521	0.646038	0.692221	0.732686	0.767101	0.795180	0.816692	0.831459	0.839359	0.840327	0.834355	0.821493
100	e	2186×10^{-6}	1886×10^{-6}	2088×10^{-6}	1852×10^{-6}	2005×10^{-6}	1838×10^{-6}	1932×10^{-6}	1849×10^{-6}	1874×10^{-6}	1866×10^{-6}	1833×10^{-6}	1903×10^{-6}
$\theta=0.4$	%e	0.5941	0.4718	0.4876	0.4085	0.4224	0.3735	0.3823	0.3593	0.3609	0.3589	0.3550	0.3744
Exact solution		0.367879	0.399758	0.428335	0.453374	0.474669	0.492044	0.505355	0.514493	0.519381	0.519981	0.516285	0.508326

TABLE 11: Example 2 - Neumann Boundary Conditions

$\Delta x = 1/10$ $r = 1.50$

$\Delta t = 0.0150$

DSPAGEV Scheme

No. of steps \ Δx x		0	1	2	3	4	5	6	7	8	9	10	11
12	e	$1498\ 10^{-6}$	$393\ 10^{-6}$	$1325\ 10^{-6}$	$521\ 10^{-6}$	$1079\ 10^{-6}$	$534\ 10^{-6}$	$671\ 10^{-6}$	$451\ 10^{-6}$	$730\ 10^{-6}$	$837\ 10^{-6}$	$57\ 10^{-6}$	
$\theta=0.6$	%e	0.1793	0.0429	0.1346	0.0498	0.0985	0.0471	0.0578	0.0383	0.0618	0.0713	0.0495	
Exact solution		0.835270	0.914485	0.984563	1.044803	1.094604	1.133468	1.161007	1.176946	1.181125	1.173502	1.154154	
52	e	$2080\ 10^{-6}$	$1344\ 10^{-6}$	$2381\ 10^{-6}$	$1986\ 10^{-6}$	$2108\ 10^{-6}$	$1849\ 10^{-6}$	$1231\ 10^{-6}$	$1154\ 10^{-6}$	$1707\ 10^{-6}$	$1664\ 10^{-6}$	$1739\ 10^{-6}$	
$\theta=0.6$	%e	0.4538	0.2677	0.4407	0.3463	0.3510	0.2973	0.1932	0.1787	0.2634	0.2584	0.2746	
Exact solution		0.458406	0.501880	0.540340	0.573400	0.600732	0.622061	0.637174	0.645922	0.648215	0.644032	0.633413	
100	e	$1700\ 10^{-6}$	$1141\ 10^{-6}$	$2510\ 10^{-6}$	$2406\ 10^{-6}$	$2224\ 10^{-6}$	$2189\ 10^{-6}$	$734\ 10^{-6}$	$791\ 10^{-6}$	$1697\ 10^{-6}$	$1531\ 10^{-6}$	$1946\ 10^{-6}$	
$\theta=0.6$	%e	0.7617	0.4671	0.9545	0.8622	0.7605	0.7230	0.2367	0.2515	0.5378	0.4884	0.6313	
Exact solution		0.223130	0.244291	0.263012	0.279104	0.292407	0.302789	0.310146	0.314404	0.315520	0.313484	0.308315	

TABLE 12: Example 2 - Neumann Boundary Conditions

$\Delta x = 1/10$ $r = 0.1$
$\Delta t = 0.001$ No. of steps = 100 time = 0.1

Scheme \ Δx x		0	1	2	3	4	5	6	7	8	9
SPAGEU	e	86×10^{-6}	51×10^{-6}	72×10^{-6}	153×10^{-6}	208×10^{-6}	267×10^{-6}	294×10^{-6}	261×10^{-6}	228×10^{-6}	146×10^{-6}
$\theta=0.6$	%e	0.0502	0.0294	0.0415	0.0867	0.1167	0.1491	0.1651	0.1481	0.1310	0.0851
SPAGEC	e	1728×10^{-6}	1715×10^{-6}	1737×10^{-6}	1727×10^{-6}	1765×10^{-6}	1773×10^{-6}	1771×10^{-6}	1786×10^{-6}	1748×10^{-6}	1749×10^{-6}
$\theta=0.7$	%e	1.0110	0.9978	0.9989	0.9797	0.9913	0.9917	0.9942	1.0130	1.0049	1.0178
SSPAGEV	e	479×10^{-6}	472×10^{-6}	580×10^{-6}	730×10^{-6}	886×10^{-6}	968×10^{-6}	953×10^{-6}	860×10^{-6}	702×10^{-6}	567×10^{-6}
$\theta=0.8$	%e	0.2801	0.2744	0.3335	0.4141	0.4976	0.5416	0.5354	0.4879	0.4037	0.3297
DSPAGEV	e	630×10^{-6}	616×10^{-6}	715×10^{-6}	861×10^{-6}	1010×10^{-6}	1094×10^{-6}	1087×10^{-6}	1000×10^{-6}	854×10^{-6}	719×10^{-6}
$\theta=0.8$	%e	0.3686	0.3587	0.4109	0.4882	0.5672	0.6122	0.6106	0.5673	0.4907	0.4185
Exact solution		0.170916	0.171834	0.173940	0.176297	0.178091	0.178758	0.178091	0.176297	0.173940	0.171834

TABLE 13: Example 3 - Periodic Boundary Conditions

$\Delta x = 1/10$ $r = 0.5$
$\Delta t = 0.005$ No. of steps=1 time=0.5

Scheme \ Δx x		0	1	2	3	4	5	6	7	8	9
SPAGEU	e	6183×10^{-6}	6164×10^{-6}	6159×10^{-6}	6178×10^{-6}	6170×10^{-6}	6177×10^{-6}	6186×10^{-6}	6182×10^{-6}	6202×10^{-6}	6186×10^{-6}
θ=1.0	%e	1.7090	1.6888	1.6558	1.6289	1.6043	1.5982	1.6086	1.6300	1.6673	1.6947
SPAGEC	e	3725×10^{-6}	3711×10^{-6}	3763×10^{-6}	3736×10^{-6}	3776×10^{-6}	3769×10^{-6}	3756×10^{-6}	3776×10^{-6}	3723×10^{-6}	3751×10^{-6}
θ=0.0	%e	1.0295	1.0166	1.0117	0.9851	0.9818	0.9752	0.9766	0.9957	1.0009	1.0275
SSPAGEV	e	5953×10^{-6}	5935×10^{-6}	5948×10^{-6}	5905×10^{-6}	5919×10^{-6}	5907×10^{-6}	5904×10^{-6}	5945×10^{-6}	5926×10^{-6}	5972×10^{-6}
θ=1.0	%e	1.6454	1.6259	1.5992	1.5568	1.5391	1.5283	1.5351	1.5674	1.5932	1.6361
DSPAGEV	e	4624×10^{-6}	4607×10^{-6}	4620×10^{-6}	4620×10^{-6}	4589×10^{-6}	4605×10^{-6}	4598×10^{-6}	4624×10^{-6}	4608×10^{-6}	4640×10^{-6}
θ=1.0	%e	1.2782	1.2622	1.2422	1.2100	1.1973	1.1896	1.1951	1.2191	1.2389	1.2712
Exact solution		0.361773	0.356015	0.371970	0.379278	0.384588	0.386512	0.384988	0.379278	0.371970	0.365015

TABLE 14: Example 3 - Periodic Boundary Conditions

$\Delta x = 1/10$ $r = 1.0$

$\Delta t = 0.01$ No. of steps = 100 time = 1.0

Scheme \ Δx x		0	1	2	3	4	5	6	7	8	9
SPAGEU	e	0.018967	0.018916	0.018925	0.019029	0.019012	0.019046	0.019084	0.019042	0.019074	0.018992
θ=1.0	%e	1.9496	1.9312	1.9044	1.8864	1.8645	1.8605	1.8715	1.8877	1.9194	1.9390
SPAGEC	e	0.009974	0.009938	0.010031	0.009966	0.010035	0.010018	0.009995	0.010044	0.009951	0.010020
θ=0.0	%e	1.0252	1.0146	1.0094	0.9879	0.9841	0.9786	0.9802	0.9957	1.0014	1.0230
SSPAGEV	e	0.022086	0.022035	0.022117	0.021780	0.021857	0.021744	0.021708	0.022009	0.021880	0.022205
θ=1.0	%e	2.2702	2.2496	2.2256	2.1590	2.1435	2.1241	2.1289	2.1818	2.2017	2.2669
DSPAGEV	e	0.013035	0.013003	0.013090	0.012943	0.013039	0.013041	0.013023	0.013099	0.012994	0.013110
θ=1.0	%e	1.3398	1.3275	1.3172	1.2830	1.2787	1.2740	1.2771	1.2985	1.3076	1.3384
Exact solution		0.972884	0.979501	0.993748	1.008764	1.019700	1.023665	1.019700	1.008764	0.993748	0.979501

TABLE 15: Example 3 - Periodic Boundary Conditions

$\Delta x = 1/10$ $r = 1.5$
$\Delta t = 0.015$ No. of steps = 100 time = 1.5

Scheme \ Δx x		0	1	2	3	4	5	6	7	8	9
SPAGE	e										
θ=	%e										
SPAGE	e										
θ=	%e										
SSPAGEV	e	0.042504	0.042624	0.042724	0.042031	0.042088	0.041353	0.041788	0.042304	0.042129	0.042726
θ=0.5	%e	2.1245	2.1199	2.1024	2.0454	2.0318	2.0146	2.0173	2.0587	2.0731	2.1250
DSPAGEV	e	0.030937	0.030914	0.031137	0.030753	0.030965	0.031214	0.031200	0.031174	0.030862	0.031047
θ=0.0	%e	1.5463	1.5375	1.5322	1.4966	1.4948	1.5025	1.5062	1.5170	1.5186	1.5441
Exact solution		2.000662	2.010654	2.032192	2.054917	2.071477	2.077484	2.071477	2.054917	2.032192	2.010654

TABLE 16: Example 3 - Periodic Boundary Conditions

$\Delta x = 1/10$ $r = 0.1$
$\Delta t = 0.001$ No. of steps=100 time=0.1

Δx	x	1	2	3	4	5	6	7	8	9	
C.N.	%e	0.811	0.809	0.808	0.807	0.806	0.807	0.808	0.809	0.811	
Explicit	%e	0.324	0.324	0.323	0.323	0.323	0.323	0.323	0.324	0.324	
GER	%e	0.317	0.809	0.691	0.807	0.806	0.807	0.924	0.809	1.304	
SPAGER $\theta=0.6$	%e	0.4859	0.0007	0.1234	0.0192	0.0355	0.0192	0.0175	0.0007	0.4931	
Exact solution		0.118678	0.225727	0.310659	0.365156	0.383883	0.365156	0.310659	0.225727	0.118678	

TABLE 17: Example 1 - Dirichlet Boundary Conditions

$\Delta x = 1/10$ $r = 1.0$
$\Delta t = 0.01$ No. of steps=100 time=1.0

Δx	x	1	2	3	4	5	6	7	8	9	
C.N.	%e	7.577	7.577	7.577	7.577	7.577	7.577	7.577	7.577	7.577	
GER	%e	4.534	9.913	8.633	9.913	9.900	9.913	11.166	9.913	15.265	
SPAGER $\theta=0.6$	%e	4.0291	0.1125	1.0351	0.0942	0.0859	0.0942	0.8085	0.1125	3.7799	
DAGE	%e	9.530	4.874	7.808	6.670	7.255	7.508	6.614	9.934	1.856	
Exact solution		16×10^{-6}	31×10^{-6}	43×10^{-6}	50×10^{-6}	53×10^{-6}	50×10^{-6}	43×10^{-6}	31×10^{-6}	16×10^{-6}	

TABLE 18: Example 1 - Dirichlet Boundary Conditions

4.13 REMARKS

A large number of numerical tests were carried out to check the validity and accuracy of the schemes derived in this chapter. From these experiments the following remarks can be made.

a) In the tables of Section 4.12 the effect of θ on the first and third problems can be clearly seen.

In fact θ may not increase the accuracy in some cases but it usually smooths the curve of the solution to match with the curve of the exact solution.

b) In the first example, with Dirichlet boundary conditions, we see that for an even number of intervals the value of θ=0.6 for the SPAGEL and SPAGER schemes gives the most accurate solution for the different values of r=0.1, 0.5 and 1.0.

c) In most cases of the Dirichlet boundary problem we notice that the two schemes SSPAGE and DSPAGE give their most accurate solution when θ=0 except for the cases of odd number of intervals and $r<0.5$.

d) For the Neumann boundary problem, the schemes SPAGEL, SPAGER, SPAGEC, SPAGEU and SSPAGE for both even and odd numbers of intervals, give their most accurate solution when θ=0 for all values of r, while the DSPAGE scheme, for both even and odd numbers of intervals, gives the most accurate solutions when θ=0 for $r<1.0$ and when $\theta\cong 0.5$ for $r\gtrsim 1.0$.

e) For the periodic problem it is recommended for all schemes to use $\theta\cong 0.7$ when r=0.1 and θ=1.0 when r=0.5, 1.0, except for SPAGEC where θ=0.0 appears to be best for r=0.5, 1.0.

When $r<1$, the SSPAGE and DSPAGE appear to be at their optimum accuracy with $0.0 \leq \theta \leq 0.5$.

f) In almost all cases for our numerical examples we do not recommend the SSPAGE scheme when $r>0.5$. This scheme appears to be as good as the other schemes for $r \leq 0.5$.

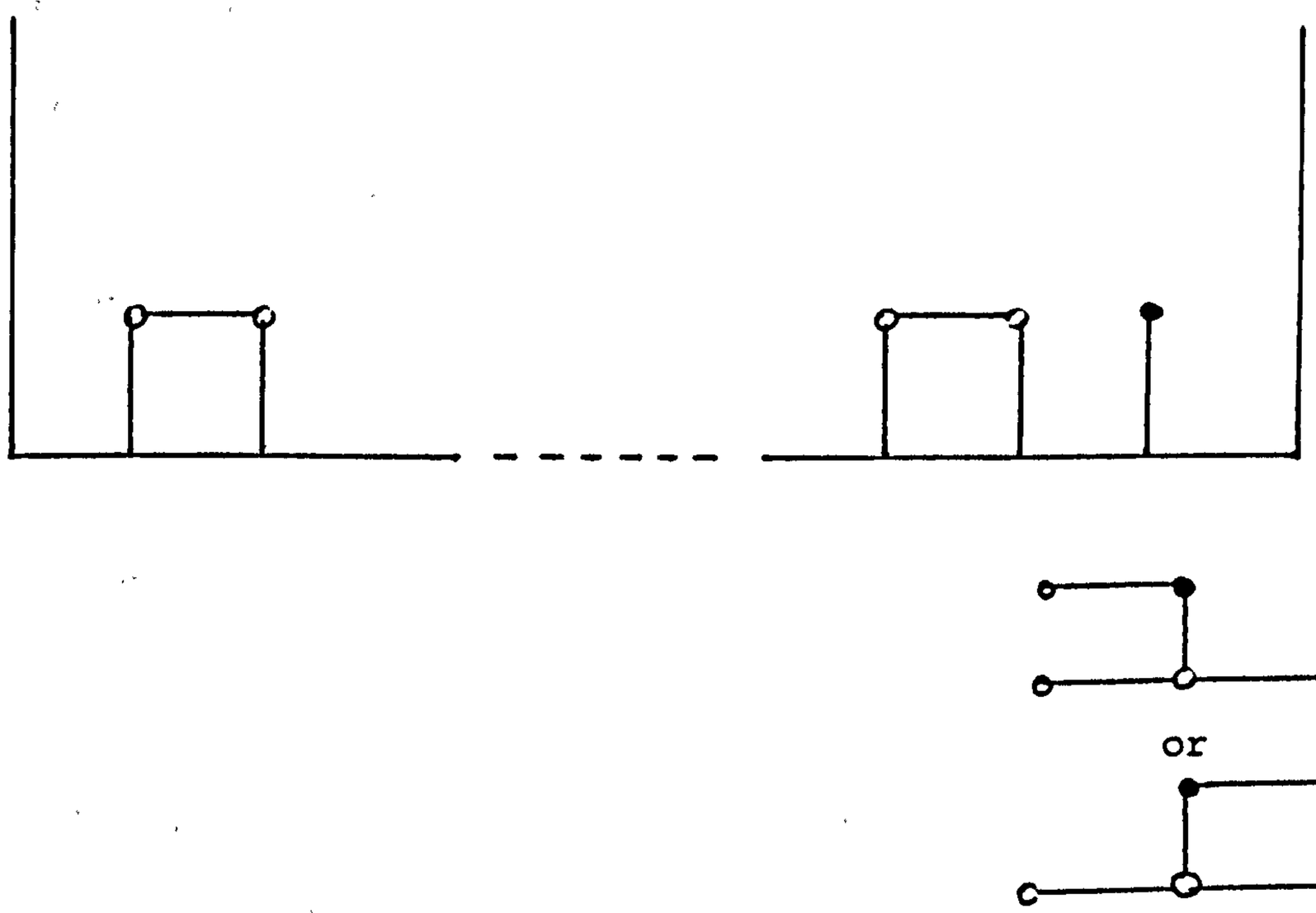
g) From the comparison results in Tables (4.12.17) and (4.12.18) we can clearly see the superiority of the results from SPAGE over other results.

h) The number of intervals (odd or even) does not appear to make any large difference in the numerical results.

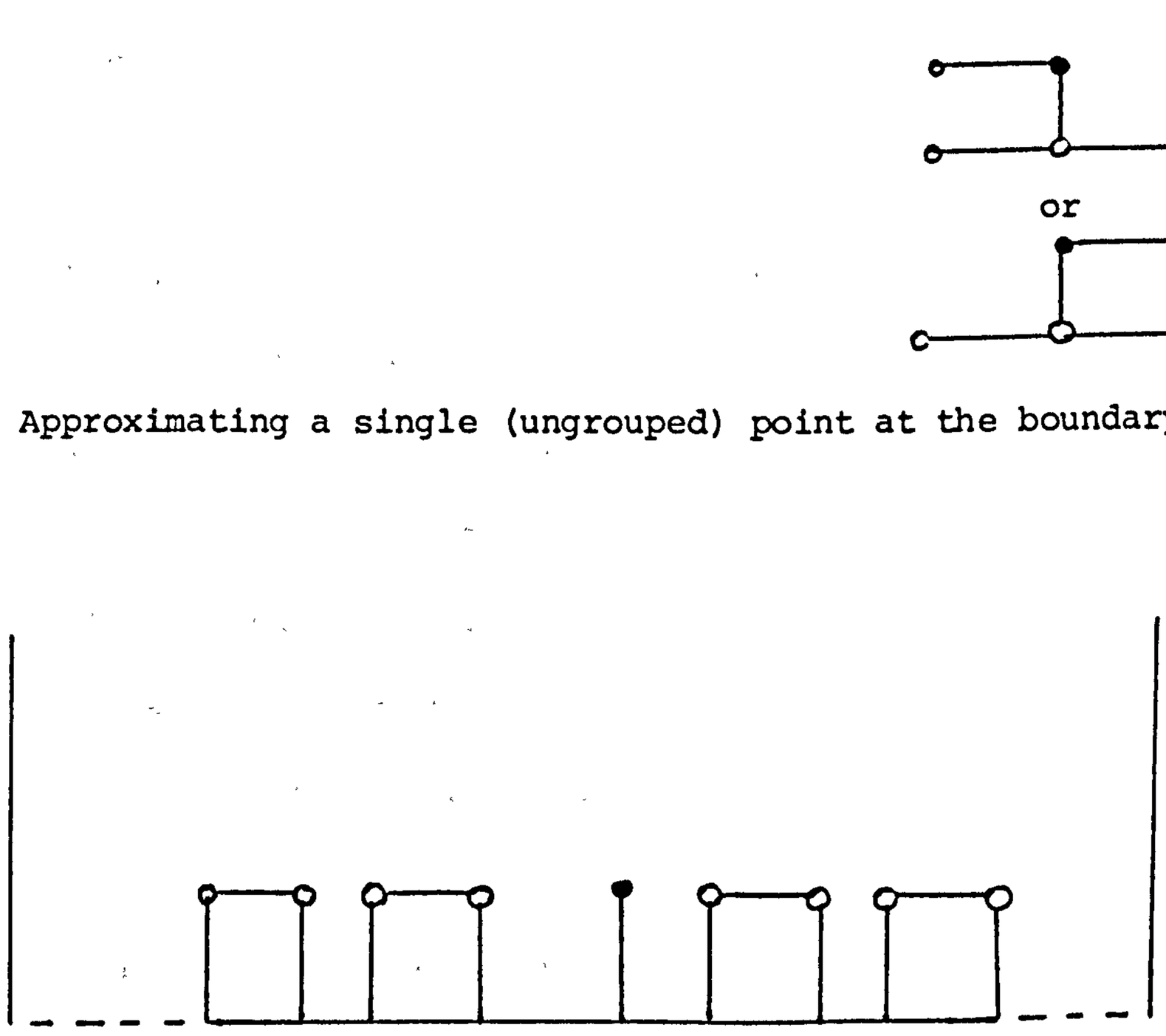
i) The results confirm the theoretical stability conditions that the simple schemes (SPAGER, SPAGEL, SPAGEC and SPAGEU) are stable for $r \leq 1$ and the composed schemes (SSPAGE and DSPAGE) are stable for $r>0$. However, this does not imply that we can use higher values of r than that used already, simply because of lack of accuracy due to the rounding errors and the limited arithmetic capacity of the machine.

j) From Tables (4.12.13,14 and 15) we notice that as r increases the accuracy of the SPAGEC scheme is almost unchanged, while the scheme SPAGEU gets less accurate.

k) Although, in our numerical results, the effect of having a single solitary point at one or both boundary points is not clear due to the use of θ, we think that, in the case of one single boundary point, it can be moved to the middle of the domain. Then use



Approximating a single (ungrouped) point at the boundary



Approximating a single point at the centre

FIGURE 4.13.1

complete groups at both boundaries and for the single middle points an accurate formula can be used. By this technique we appear to eliminate the high truncation error in approximating a single (ungrouped) point, (see Figure (4.13.1)). The formula to be used for the single point can be determined according to r.

For $r<0.5$, the classical explicit formula can be used, while for $r\geq 0.5$, an implicit formula, like the classical implicit of Crank-Nicolson (C.N.) can be used.

ℓ) Using a very similar technique to that used in deriving the SPAGE method, a three time level group explicit scheme can be derived. This is done in the same manner as in Section (4.2) up to equation (4.2.14), where then at the point (i-1,j+1) we approximate $\frac{\partial U}{\partial t}$ by a weighted backward difference formula, as in equation (4.2.15), at the two points (i-1,j+1) and (i,j+1), while at (i+1,j) we use again a weighted backward difference formula instead of the forward formula, as in (4.2.16) and the weighted points are (i+1,j) and (i,j) instead of (i-1,j) and (i,j), as in (4.2.16). This produces the following equations in correspondence to equations (4.2.15) and (4.2.16),

$$M_{i-1,j+1} = \alpha\left(\frac{u_{i-1,j+1}-u_{i-1,j}}{\Delta t}\right)+(1-\alpha)\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}\right) \qquad (4.13.1)$$

and

$$M_{i+1,j} = \alpha\left(\frac{u_{i,j}-u_{i,j-1}}{\Delta t}\right)+(1-\alpha)\left(\frac{u_{i+1,j}-u_{i+1,j-1}}{\Delta t}\right), \qquad (4.13.2)$$

where $0\leq\alpha\leq 1$.

Recalling equations (4.2.14) and (4.2.18), we have,

$$2(M_{i,j}+M_{i,j+1}) = 4\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}\right), \qquad (4.13.3)$$

and

$$M_{i-1,j}+M_{i+1,j}+2(M_{i,j+1}+M_{i,j})+\frac{6}{h^2}(u_{i,j+1}-u_{i-1,j+1}) = \frac{6}{h^2}(u_{i,j+1}-u_{i-1,j}) \qquad (4.13.4)$$

respectively.

Now substituting equations (4.13.1), (4.13.2) and (4.13.3) in (4.13.4) gives,

$$\alpha\left(\frac{u_{i-1,j+1}-u_{i-1,j}}{\Delta t}\right)+(1-\alpha)\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}\right)$$

$$+\alpha\left(\frac{u_{i,j}-u_{i,j-1}}{\Delta t}\right)+(1-\alpha)\left(\frac{u_{i+1,j}-u_{i+1,j-1}}{\Delta t}\right)$$

$$+4\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}\right)+\frac{6}{h^2}(u_{i,j+1}-u_{i-1,j+1}) = \frac{6}{h^2}(u_{i,j}-u_{i-1,j}). \qquad (4.13.5)$$

Rearranging equation (4.13.5) leads to,

$$\left(\frac{\alpha}{\Delta t}-\frac{6}{h^2}\right)u_{i-1,j+1}+\left(\frac{(1-\alpha)}{\Delta t}+\frac{4}{\Delta t}+\frac{6}{h^2}\right)u_{i,j+1} = \frac{\alpha}{\Delta t}u_{i-1,j}$$

$$+\left(\frac{(1-\alpha)}{\Delta t}-\frac{\alpha}{\Delta t}+\frac{4}{\Delta t}-\frac{6}{h^2}\right)u_{i,j}+\left(\frac{6}{h^2}-\frac{(1-\alpha)}{\Delta t}\right)u_{i+1,j}+\frac{\alpha}{\Delta t}u_{i,j-1}$$

$$+\frac{1-\alpha}{\Delta t}u_{i+1,j-1}. \qquad (4.13.6)$$

As in Section (4.2) but in the reverse direction to the above, we approximate $\frac{\partial U}{\partial t}$ at the points (i+1,j+1) and (i-1,j) by a weighted backward difference formula at the groups of points (i+1,j+1),(i,j+1) and (i,j),(i-1,j) respectively. This results in rewriting equation (4.1.1) as,

$$M_{(i+1,j+1)} = \alpha\left(\frac{u_{i+1,j+1}-u_{i+1,j}}{\Delta t}\right)+(1-\alpha)\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta t}\right) \qquad (4.13.7)$$

and,

$$M_{(i-1,j)} = \alpha\left(\frac{u_{i,j}-u_{i,j-1}}{\Delta t}\right)+(1-\alpha)\left(\frac{u_{i-1,j}-u_{i-1,j-1}}{\Delta t}\right), \qquad (4.13.8)$$

respectively.

Substituting equation (4.2.14), (4.13.7) and (4.13.8) in equation (4.2.10) gives,

$$(\frac{\alpha}{\Delta t} - \frac{6}{h^2})u_{(i+1,j+1)} + (\frac{(1-\alpha)}{\Delta t} + \frac{4}{\Delta t} + \frac{6}{h^2})u_{i,j+1} = \frac{\alpha}{\Delta t}u_{i+1,j} + (\frac{(1-\alpha)}{\Delta t} - \frac{\alpha}{\Delta t}$$

$$+ \frac{4}{\Delta t} - \frac{6}{h^2})u_{i,j} + (\frac{6}{h^2} - \frac{(1-\alpha)}{\Delta t})u_{i-1,j} + \frac{\alpha}{\Delta t}u_{i,j-1} + \frac{(1-\alpha)}{\Delta t}u_{i-1,j-1}.$$

(4.13.9)

Multiply equations (4.13.6) and (4.13.9) by Δt to give,

$$(\alpha-6r)u_{i-1,j+1} + (5-\alpha+6r)u_{i,j+1} = \alpha u_{i-1,j} + (5-2\alpha-6r)u_{i,j} + (6r-(1-\alpha))u_{i+1,j}$$

$$+\alpha u_{i,j-1} + (1-\alpha)u_{i+1,j-1}, \qquad (4.13.10)$$

and

$$(\alpha-6r)u_{i+1,j+1} + (5-\alpha+6r)u_{i,j+1} = \alpha u_{i+1,j} + (5-2\alpha-6r)u_{i,j} + (6r-(1-\alpha))u_{i-1,j}$$

$$+ \alpha u_{i,j-1} + (1-\alpha)u_{i-1,j-1}, \qquad (4.13.11)$$

respectively, where $r = \frac{\Delta t}{h^2}$. See Figures (4.13.2a and b).

Using the same idea of producing the SPAGE scheme, then by shifting equation (4.13.10) one step towards the positive direction of the x axis and combining this with equation (4.13.11) results in the equation,

$$\begin{bmatrix} 5-\alpha+6r & \alpha-6r \\ \alpha-6r & 5-\alpha+6r \end{bmatrix} \begin{bmatrix} u_{i,j+1} \\ u_{i+1,j+1} \end{bmatrix} = \begin{bmatrix} 5-2\alpha-6r & \alpha \\ \alpha & 5-2\alpha-6r \end{bmatrix} \begin{bmatrix} u_{i,j} \\ u_{i+1,j} \end{bmatrix} +$$

$$\begin{bmatrix} (6r-(1-\alpha))\, u_{i-1,j} \\ (6r-(1-\alpha))\, u_{i+2,j} \end{bmatrix} + \begin{bmatrix} (1-\alpha)u_{i-1,j-1} + \alpha u_{i,j-1} \\ \alpha u_{i+1,j-1} + (1-\alpha)u_{i+2,j-1} \end{bmatrix}$$

(4.13.12)

(see Figure (4.13.3).

Equation (4.13.12) can then be easily solved.

This scheme still requires some theoretical analysis and the associated numerical experiment to test its applicability. As a three time level scheme, this could increase the accuracy of the SPAGE schemes even more so and improve the stability bounds. This will require further experimentation and will be left for future study.

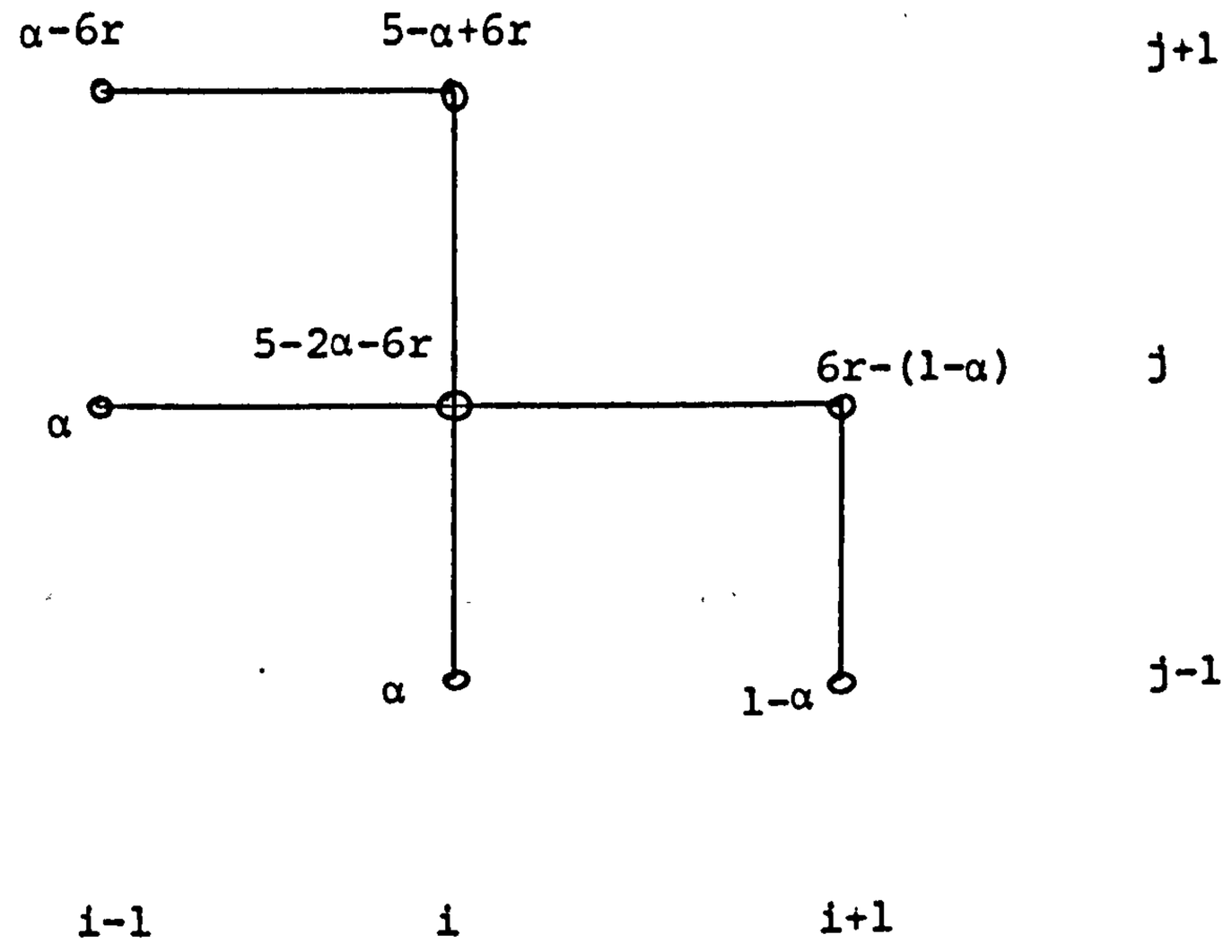


FIGURE 4.13.2a

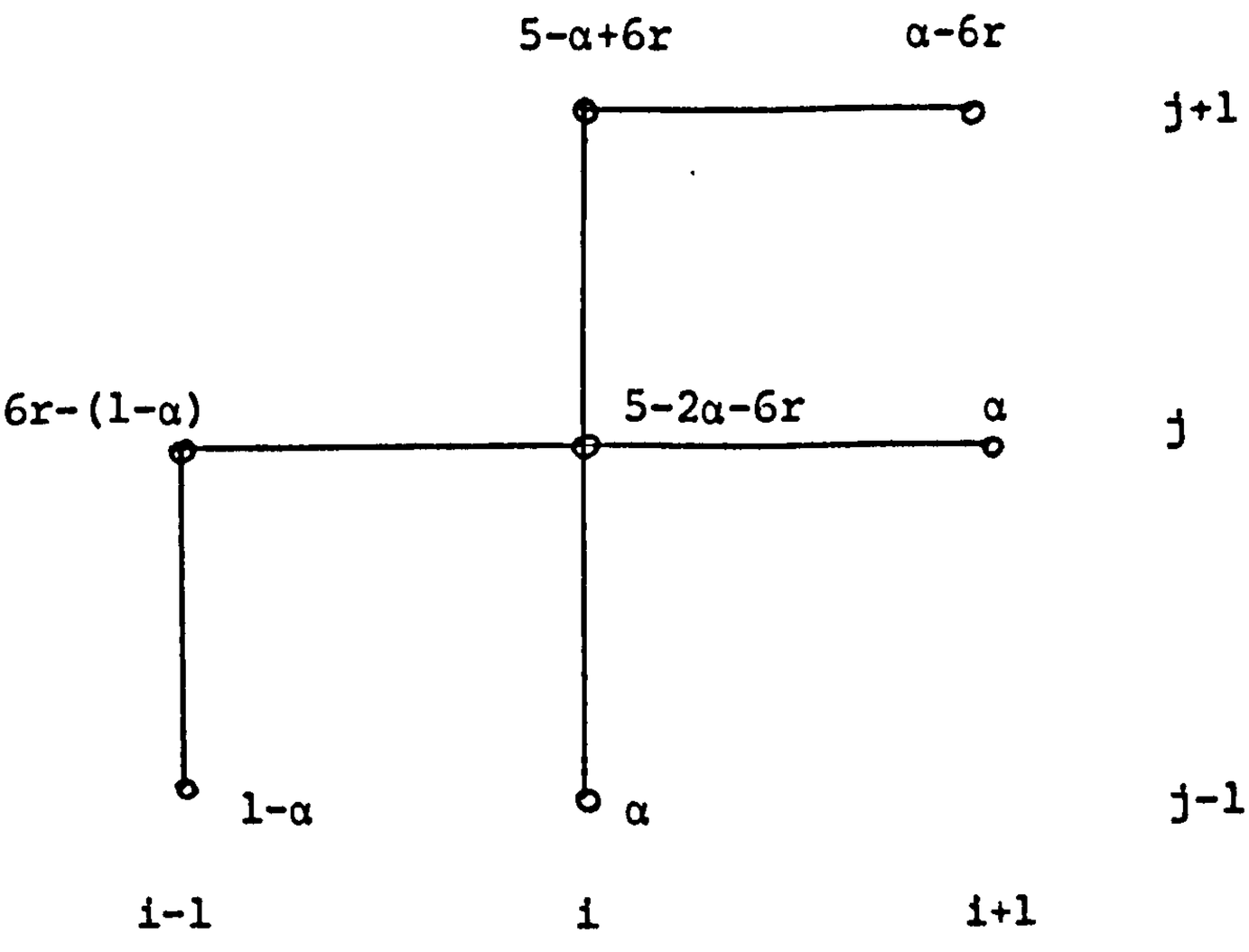
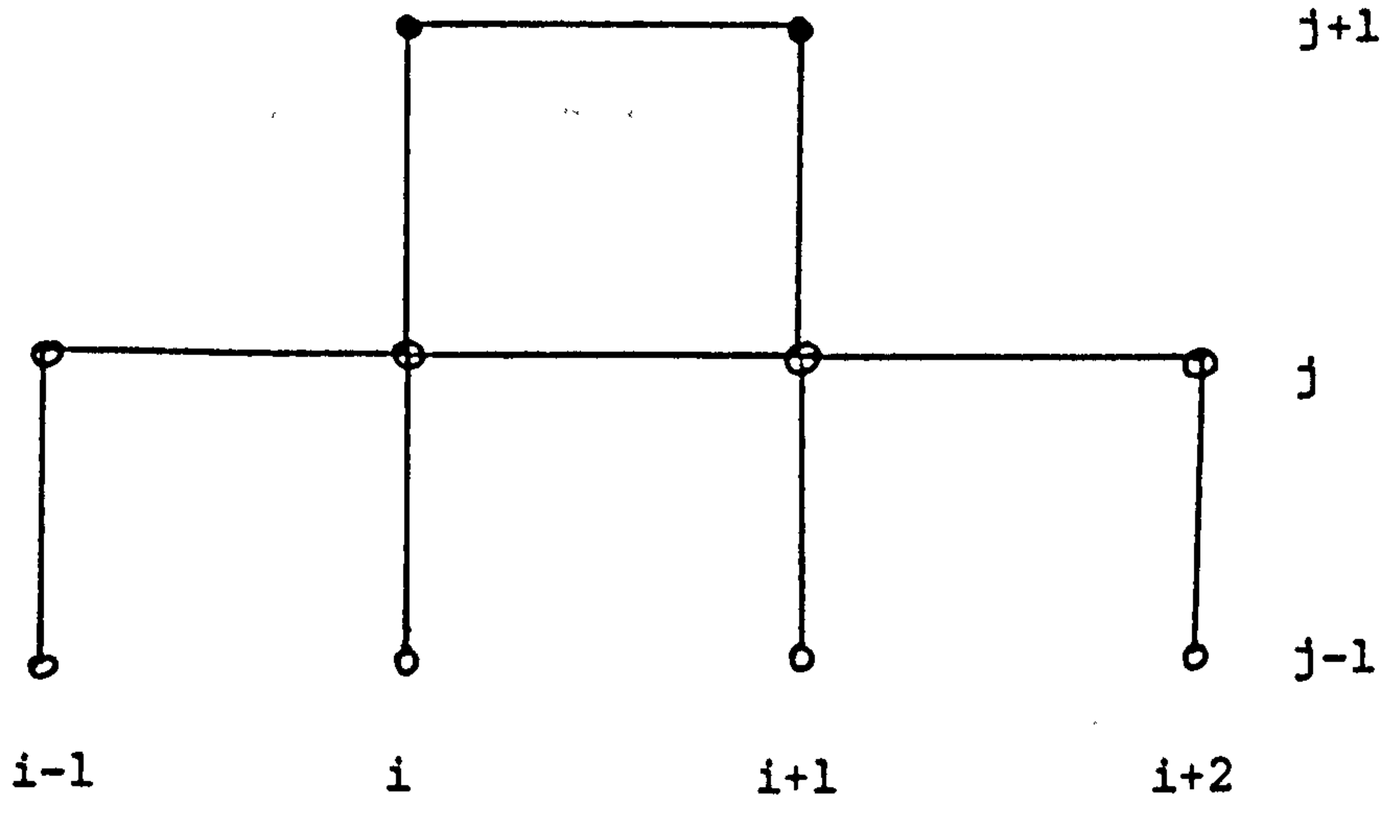


FIGURE 4.13.2b



Molecular Diagram of Equation (4.13.12)

FIGURE 4.13.3

CHAPTER FIVE

THE SPLINE ALTERNATING GROUP EXPLICIT ITERATIVE (SPAGEI) METHOD

5.1 INTRODUCTION

In the previous chapter we introduced a direct explicit method for solving parabolic partial differential equations, and examined its efficiency.

In this chapter, we will introduce an iterative explicit method for solving a similar problem which is closely related to the direct method. The method is derived from the cubic splines formula of Papamichael and Whiteman (1973). A splitting technique is used to produce a (2×2) block diagonal system which is them solved iteratively.

The method is tested using the Peaceman-Rachford and Wachspress ADI parameters. Also, the eigenvalues of the (2×2) matrix are used as parameters to accelerate convergence. A constant single parameter is determined which gives the minimum number of iterations possible. Some numerical results are presented and compared.

The problem is again the heat conduction equation,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \quad , \quad 0 \leq x \leq 1, \; t \geq 0 \quad , \tag{5.1.1}$$

for different boundary conditions.

5.2 THE ALTERNATING DIRECTION IMPLICIT (ADI) METHOD

This method was introduced by Peaceman, D.W. and Rachford, H.H., Jr. in 1955. They considered the two dimensional unsteady-state heat flow equation

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} \qquad (5.2.1)$$

The ADI method suggests that only one of the second derivatives, say $\frac{\partial^2 U}{\partial x^2}$, is replaced by a central difference evaluated in terms of the unknown values of U at 3 unknown points, while the other derivative, $\frac{\partial^2 U}{\partial y^2}$, is replaced by a central difference evaluated in terms of known values of U. Then, sets of simultaneous equations are formed that can be solved implicitly in the x-direction. The procedure is then repeated for a second time step of equal size, with the difference equations implicit in the y-direction. The overall procedure of the two time steps is stable for any size time step Δt.

Thus, two difference equations are used

$$\frac{u_{i,j}^{k+1} - u_{i,j}^{k}}{\Delta t} = \frac{u_{i-1,j}^{k+1} - 2u_{i,j}^{k+1} + u_{i+1,j}^{k+1}}{(\Delta x)^2} + \frac{u_{i,j-1}^{k} - 2u_{i,j}^{k} + u_{i,j+1}^{k}}{(\Delta y)^2} \qquad (5.2.2)$$

for the first time step and

$$\frac{u_{i,j}^{k+2} - u_{i,j}^{k+1}}{\Delta t} = \frac{u_{i-1,j}^{k+1} - 2u_{i,j}^{k+1} + u_{i+1,j}^{k+1}}{(\Delta x)^2} + \frac{u_{i,j-1}^{k+2} - 2u_{i,j}^{k+2} + u_{i,j+1}^{k+2}}{(\Delta y)^2} \qquad (5.2.3)$$

for the second time step, where i,j,k represents the spaces x,y and time directions respectively.

These two equations can be arranged to give the following sets of equations,

$$-ru_{i-1,j}^{k+1} + (1+2r)u_{i,j}^{k+1} - ru_{i+1,j}^{k+1} = ru_{i,j-1}^{k} + (1-2r)u_{i,j}^{k} + ru_{i,j+1}^{k} \qquad (5.2.4)$$

and

$$-ru^{k+2}_{i,j-1}+(1+2r)u^{k+2}_{i,j}-ru^{k+2}_{i,j+1} = ru^{k+1}_{i-1,j}+(1-2r)u^{k+1}_{i,j}+ru^{k+1}_{i+1,j} \qquad (5.2.5)$$

Each of them produces a tridiagonal system which can be solved using the direct tridiagonal system solver.

The ADI method is also used to iterate to the solution of Laplace's equation

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \ . \qquad (5.2.6)$$

In this case each stage of iteration is regarded as time step of an unsteady state problem.

In both cases, the method is stable for r>0 provided that the two steps (5.2.4) and (5.2.5) are applied alternatively, or the corresponding form of the Laplace equation. The convergence proof for the iterative procedure follows from the stability analysis of the unsteady state case, see [Peaceman, D.W. and Rachford, H.H. Jr., 1955].

The two difference equations that approximate equation (5.2.6) can be written in the matrix form as

$$(H+\rho_k I)u^{(k+\frac{1}{2})} = c + (V-\rho_k I)u^{(k)} \qquad (5.2.7)$$

$$(V+\tilde{\rho}_k I)u^{(k+1)} = c + (H-\tilde{\rho}_k I)u^{(k+\frac{1}{2})} \ , \qquad (5.2.8)$$

where H and V are the coefficient matrices, $\rho_k, \tilde{\rho}_k$ are acceleration parameters, I is the identity matrix, c is the vector associated with the boundary conditions and k represents the number of iterations [Birkhoff, G., Varga, R.S. and Young, D., 1962].

Equations (5.2.7) and (5.2.8) can be written as

$$u^{(k+\frac{1}{2})} = (H+\rho_k I)^{-1}[c+(V-\rho_k I)u^{(k)}] \ , \qquad (5.2.9)$$

$$u^{(k+1)} = (V+\tilde{\rho}_k I)^{-1}[c+(H-\tilde{\rho}_k I)u^{(k+\frac{1}{2})}] \ . \qquad (5.2.10)$$

Douglas, J. Jr. and Rachford, H. [1956, p.422] have suggested a variant of the Peaceman and Rachford method. They used the vector $u^{(k)}$ in the second equation so that equations (5.2.9) and (5.2.10) can be written as

$$u^{(k+\frac{1}{2})} = (H+\rho_k I)^{-1}[c+(V-\rho_k I)u^{(k)}] \quad , \tag{5.2.11}$$

$$u^{(k+1)} = (V+\rho_k I)^{-1}[Vu^{(k)}+\rho_k u^{(k+\frac{1}{2})}] \quad , \tag{5.2.12}$$

which is also convergent for all $r>0$.

5.3 THE SPLINE ALTERNATING GROUP EXPLICIT ITERATIVE METHOD

Now we recall the cubic splines equation due to Papamichael and Whiteman (1973), i.e.,

$$(1-3r)(u_{i-1,j+1}+u_{i+1,j+1})+(4+6r)u_{i,j+1} = (1+3r)(u_{i-1,j}+u_{i+1,j}) + (4-6r)u_{i,j} \ . \qquad (5.3.1)$$

This is used to give an approximate solution to the equation (5.1.1).

Let us define equation (5.1.1) in the domain $0 \leq x \leq 1$ and $t \geq 0$.

For i=1,...,n, equation (5.3.1) produces the system

$$Au_{j+1} = Bu_j + c \ , \qquad (5.3.2)$$

where,

$$A = \begin{bmatrix} 4+6r & 1-3r & & & & \\ 1-3r & 4+6r & 1-3r & & O & \\ & \ddots & \ddots & \ddots & & \\ & & & 1-3r & 4+6r & 1-3r \\ & O & & & 1-3r & 4+6r \end{bmatrix} ,$$

$$B = \begin{bmatrix} 4-6r & 1+3r & & & & \\ 1+3r & 4-6r & 1+3r & & O & \\ & \ddots & \ddots & \ddots & & \\ & & & 1+3r & 4-6r & 1+3r \\ & O & & & 1+3r & 4-6r \end{bmatrix} ,$$

and c is the vector associated with the boundary conditions.

The procedure described earlier in the previous chapter, section 11 is to split the left hand side coefficient matrix A into A_1 and A_2

so that A_1 and A_2 are block diagonal matrices and each block is a (2×2) matrix. Then transferring each of A_1 and A_2 in turn to the righthand side of the equation results in a single block diagonal matrix on the left side which is easily solvable.

Let A be of order n where n (without loss of generality) is an odd number. Then we have

$$A = A_1 + A_2 \,, \tag{5.3.3}$$

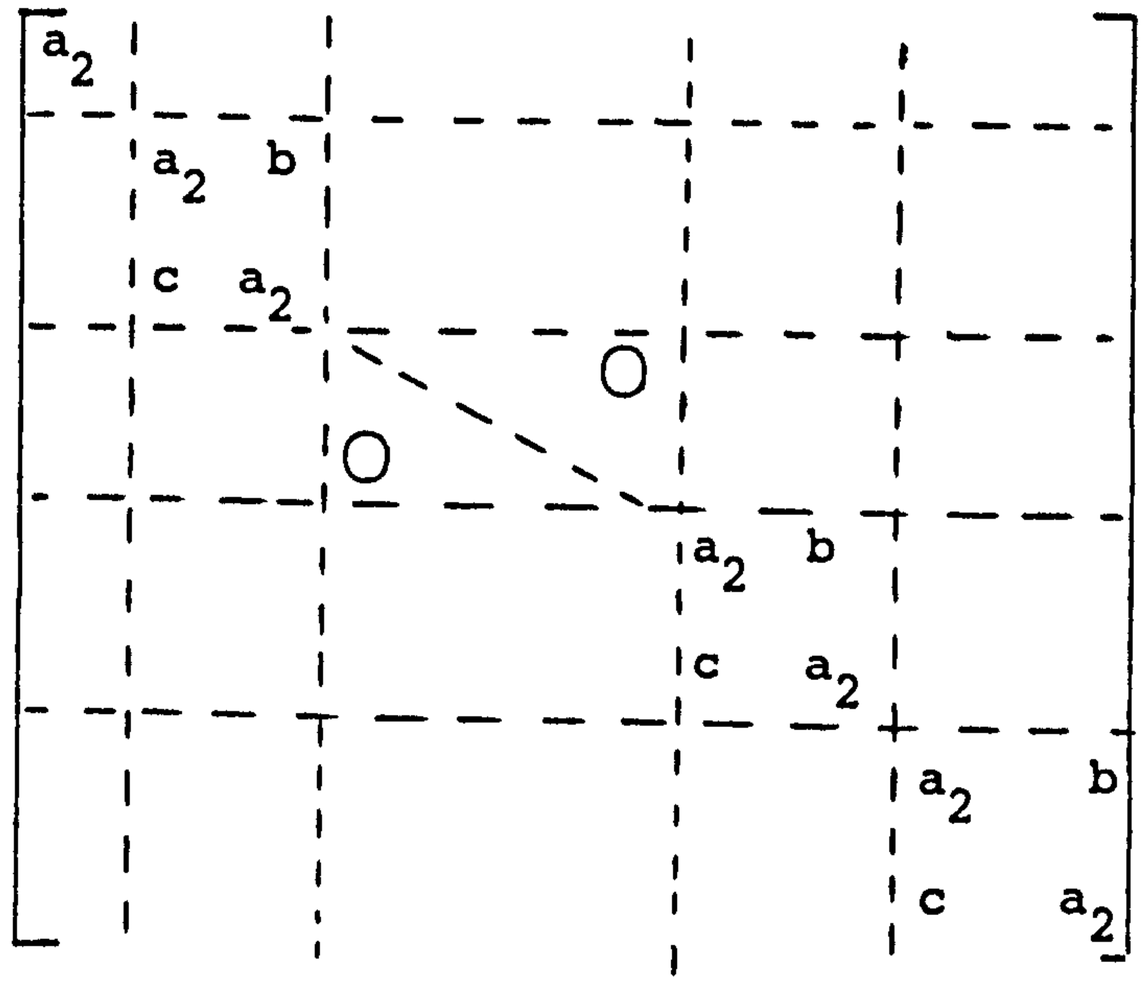
where,

$$A = \begin{bmatrix} a & b & & & & & \\ c & a & b & & & & \\ & c & a & b & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & c & a & b & \\ & & & & c & a & b \\ & & & & & c & a \end{bmatrix},$$

$$A_1 = \begin{bmatrix} a_1 & b & & & & & \\ c & a_1 & & & & & \\ & & a_1 & b & & & \\ & & c & a_1 & & & \\ & & & & \ddots & & \\ & & & & & a_1 & b \\ & & & & & c & a_1 \end{bmatrix},$$

and

$A_2 =$



where $a_1 + a_2 = a$. (5.3.4)

We are going to consider the case when $a_1=a_2=a/2$. Thus from the system above we have

$$a_1 = a_2 = 2+3r ,$$

and $$b = c = 1-3r .$$

This leads us to write equation (5.2.2) as

$$(A_1+\rho I)u_{j+1}^{(k+\frac{1}{2})} = Bu_j + c - (A_2-\rho I)u_{j+1}^{(k)} , \qquad (5.3.5a)$$

$$(A_2+\rho I)u_{j+1}^{(k+1)} = Bu_j + c - (A_1-\rho I)u_{j+1}^{(k+\frac{1}{2})} , \qquad (5.3.5b)$$

where k is the iteration counter and ρ is a positive scalar.

The application of these two equations alternatively gives an approximate solution to the equation (5.1.1) after Δt time steps. It is necessary that Δt be the same for both equations (5.3.5a) and (5.3.5b) for stability.

As the order of the equations (5.3.5a) and (5.3.5b) does not make any difference as long as they alternate successively, we write (without loss of generality) equation (5.3.5a) in a systematic form to get,

$$
\begin{bmatrix}
a_1+\rho & b & & & & \\
b & a_1+\rho & & & & \\
 & & a_1+\rho & b & & \\
 & & b & a_1+\rho & & \\
 & & & & \ddots & \\
 & & & & a_1+\rho & b \\
 & & & & b & a_1+\rho
\end{bmatrix}
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \vdots \\ u_{n-3} \\ u_{n-2} \\ u_{n-1} \end{bmatrix}^{(k+\frac{1}{2})}_{j+1}
=
$$

$$
\begin{bmatrix}
d & f & & & & \\
f & d & f & & & \\
 & f & d & f & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & f & d & f \\
 & & & & f & d
\end{bmatrix}
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-3} \\ u_{n-2} \\ u_{n-1} \end{bmatrix}_j
+
\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{bmatrix}
$$

$$
-
\begin{bmatrix}
a_1-\rho & & & & & \\
 & a_1-\rho & b & & & \\
 & b & a_1-\rho & & & \\
 & & & \ddots & & \\
 & & & & a_1-\rho & b \\
 & & & & b & a_1-\rho
\end{bmatrix}
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-4} \\ u_{n-3} \\ u_{n-2} \\ u_{n-1} \end{bmatrix}^{(k)}_{j+1},
$$

(5.3.6)

where $a_1 = 2+3r, \quad b = 1-3r, \quad d = 4-6r$,

$f = 1+3r.$

To solve this system for $u_{j+1}^{(k+\frac{1}{2})}$, we follow the normal way of substituting the initial and boundary conditions into the vectors u_j and c respectively.

For the vector $u_{j+1}^{(k)}$ we first substitute the initial conditions and then after having found $u_{j+1}^{(k+\frac{1}{2})}$, we use equation (5.3.5b) to find the values of $u_{j+1}^{(k+1)}$. At this stage a convergence test will be carried out for all the elements of the vector u_{j+1}. If the test fails another iteration will be needed and the vector $u_{j+1}^{(k)}$ in equation (5.3.5a) will take the value of the vector $u_{j+1}^{(k+1)}$, while the vector u_j will keep the same initial values. If the convergence test is successful then either we terminate the process (if that was the required solution) or carry on for another time step where in this case the last value of $u_{j+1}^{(k+1)}$ will be fed into the vectors u_j and $u_{j+1}^{(k)}$ and the above described process is repeated.

In equations (5.3.5a), (5.3.5b) and (5.3.6) we have shown the acceleration parameter ρ as a fixed number for the simplicity of representation. It, in fact, will be of great advantage to use more than one parameter to reduce the number of iterations to achieve convergence and this will become clear later.

If we write equation (5.3.2) as

$$Au = g \ , \tag{5.3.7}$$

where

$$g = Bu_j + c \ , \tag{5.3.8}$$

then from the system (5.3.6) a (2×2) block of interior points will look like this,

$$\begin{bmatrix} a_1+\rho & b \\ b & a_1+\rho \end{bmatrix} \begin{bmatrix} u_{i,j+1} \\ u_{i+1,j+1} \end{bmatrix}^{(k+\frac{1}{2})} = \begin{bmatrix} g_i \\ g_{i+1} \end{bmatrix} - \begin{bmatrix} a_1-\rho & 0 \\ 0 & a_1-\rho \end{bmatrix} \begin{bmatrix} u_{i,j+1} \\ u_{i+1,j+1} \end{bmatrix}^{(k)} - \begin{bmatrix} bu_{i-1,j+1} \\ bu_{i+2,j+1} \end{bmatrix}^{(k)} \qquad (5.3.9)$$

where $g_i = f(u_{i-1,j}+u_{i+1,j})+du_{i,j}+c_i$.

Inverting the lefthand side matrix leads to,

$$\begin{bmatrix} u_{i,j+1} \\ u_{i+1,j+1} \end{bmatrix}^{(k+\frac{1}{2})} = \frac{1}{\det} \begin{bmatrix} a_1+\rho & -b \\ -b & a_1+\rho \end{bmatrix} \left\{ \begin{bmatrix} g_i \\ g_{i+1} \end{bmatrix} - \begin{bmatrix} a_1-\rho & 0 \\ 0 & a_1-\rho \end{bmatrix} \begin{bmatrix} u_{i,j+1} \\ u_{i+1,j+1} \end{bmatrix}^{(k)} - \begin{bmatrix} b & u_{i-1,j+1} \\ b & u_{i+2,j+1} \end{bmatrix}^{(k)} \right\} \qquad (5.3.10)$$

where det is the determinant of the inverted matrix and is equal to $(a_1+\rho)^2-b^2$.

Equation (5.3.10) is now written as two equations,

$$u_{i,j+1}^{(k+\frac{1}{2})} = (pu_{i-1,j+1}^{(k)}+qu_{i,j+1}^{(k)}+tu_{i+1,j+1}^{(k)}+su_{i+2,j+1}^{(k)}+w_i)/\det \ , \qquad (5.3.11)$$

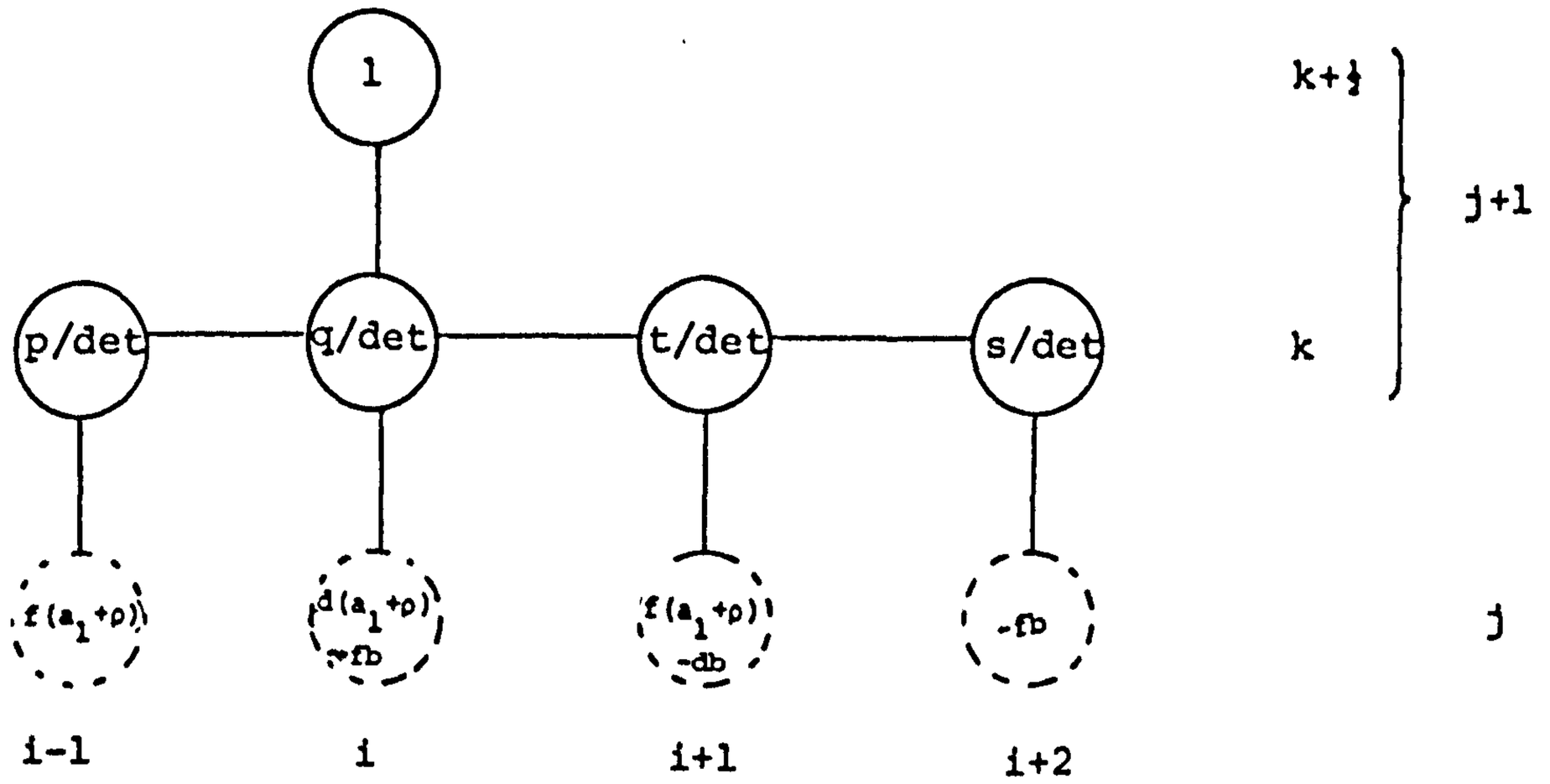
$$u_{i+1,j+1}^{(k+\frac{1}{2})} = (su_{i-1,j+1}^{(k)}+tu_{i,j+1}^{(k)}+qu_{i+1,j+1}^{(k)}+pu_{i+2,j+1}^{(k)}+w_{i+1})/\det, \qquad (5.3.12)$$

where,

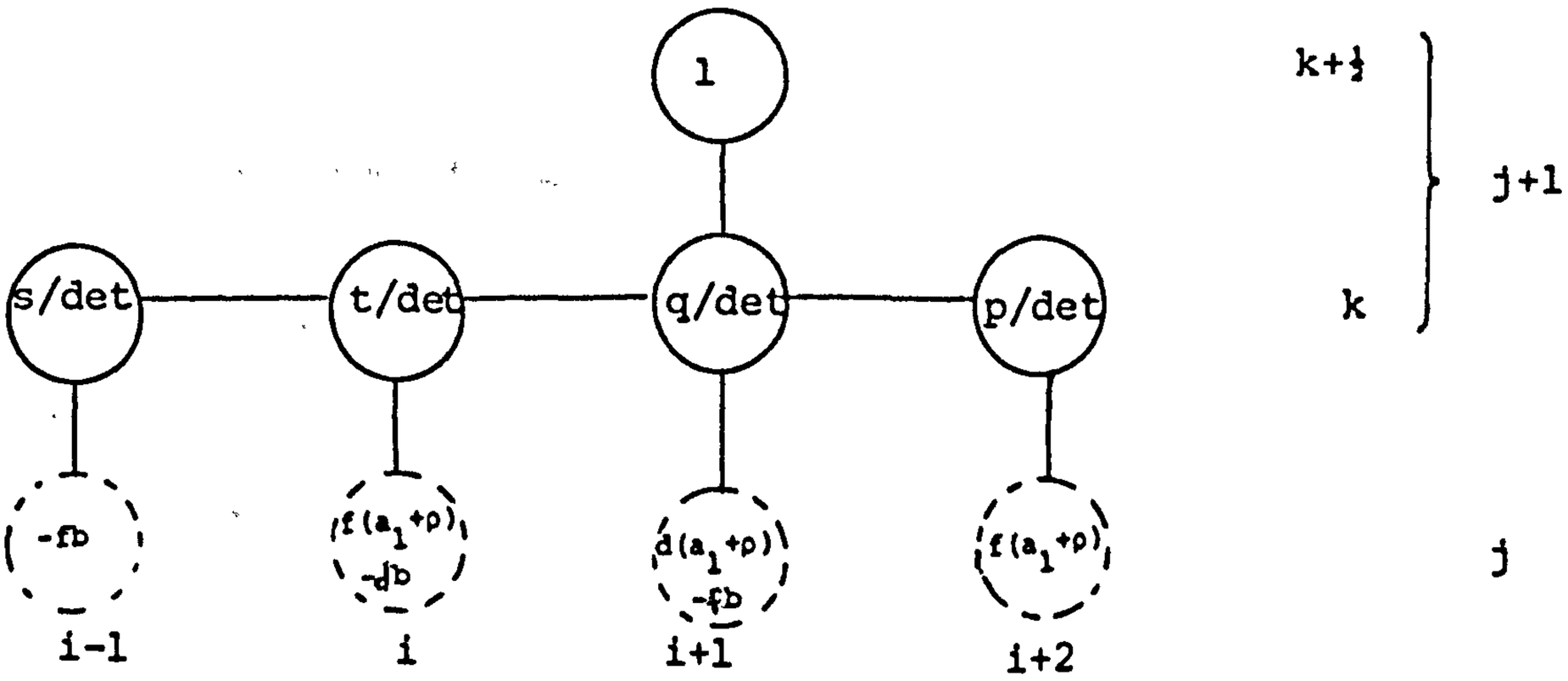
$$p = -(a_1+\rho)b, \quad q = -(a_1+\rho)(a_1-\rho) \ , \ t = b(a_1-\rho) \ ,$$

$$s = b^2, \ w_i = (a_1+\rho)g_i-bg_{i+1} \text{ and } w_{i+1} = -bg_i+(a_1+\rho)g_{i+1} \ .$$

The computational molecule diagrams of equation (5.3.11) and (5.3.12) are shown in Figs. (5.3.1a) and (5.3.1b) respectively.



a.



b.

FIGURE 5.3.1: Computational diagram of equation a.(5.3.11) and b. (5.3.12)

For a single point near the boundary, i.e. for the n-1th point in (5.3.6) we have,

$$(a_1+\rho)u^{(k+\frac{1}{2})}_{n-1,j+1} = g_{n-1} - bu^{(k)}_{n-2,j+1} -(a_1-\rho)u^{(k)}_{n-1,j+1} , \qquad (5.3.13)$$

or

$$u^{(k+\frac{1}{2})}_{n-1,j+1} = \frac{1}{(a_1+\rho)} [g_{n-1} -bu^{(k)}_{n-2,j+1} -(a_1-\rho)u^{(k)}_{n-1,j+1}] . \qquad (5.3.14)$$

This equation involves the boundary condition on the right end and on both time levels, thus its molecular diagram will look like Fig.(5.3.2).

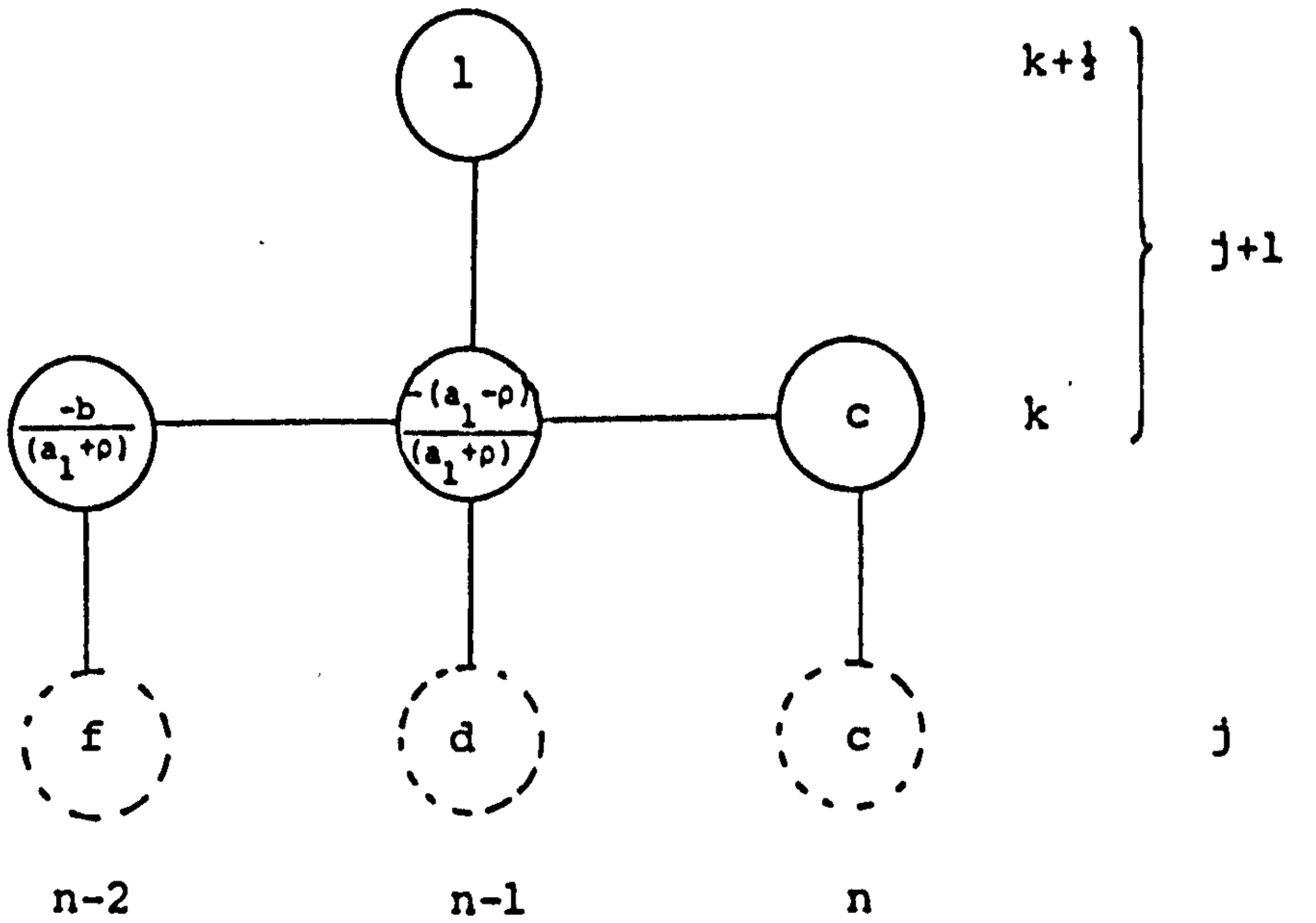


FIGURE 5.3.2: Computational diagram of equation (5.3.14)

In the case of having a single point near the left boundary, i.e. when using the second sweep

$$(A_2+\rho I)u_{j+1}^{(k+1)} = Bu_j + c - (A_1-\rho I)u_{j+1}^{(k+\frac{1}{2})} ,$$

we will have the single equation

$$(a_1+\rho)u_{1,j+1}^{(k+1)} = g_1-(a_1-\rho)u_{1,j+1}^{(k+\frac{1}{2})} -bu_{2,j+1}^{(k+\frac{1}{2})} , \qquad (5.3.15)$$

or

$$u_{1,j+1}^{(k+1)} = \frac{1}{(a_1-\rho)} [g_1-(a_1-\rho)u_{1,j+1}^{(k+\frac{1}{2})} - bu_{2,j+1}^{(k+\frac{1}{2})}] . \qquad (5.3.16)$$

which its molecular diagram is shown in Fig.(5.3.3).

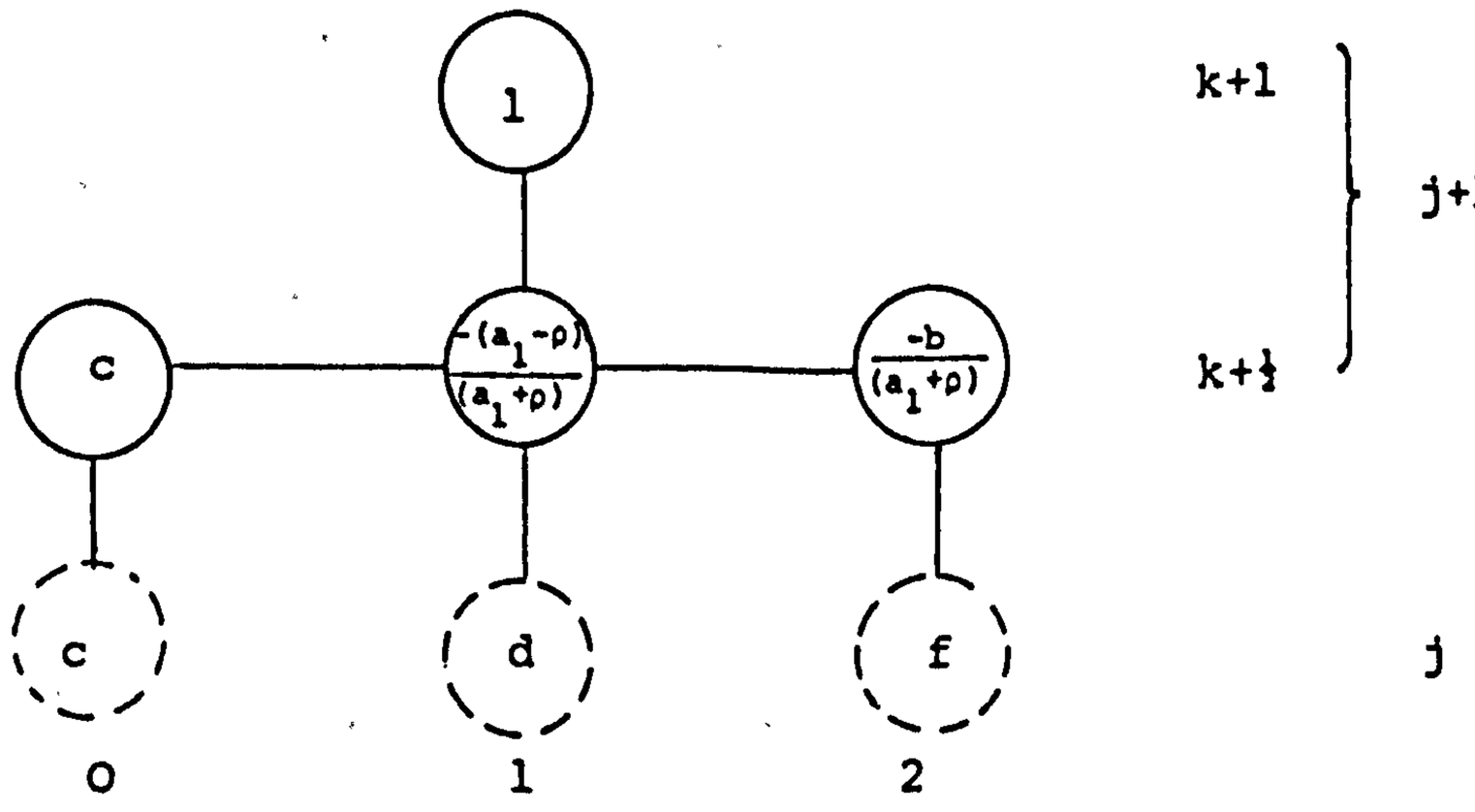


FIGURE 5.3.3: Computational diagram of equation (5.3.16)

5.4 CONVERGENCE OF THE (SPAGEI) METHOD

It can be easily seen that $(A_1+\rho I)$ and $(A_2+\rho I)$ are symmetric and positive definite. In fact, the eigenvalues of A_1 and A_2 are easily found where we have for the same pair of eigenvalues for all the 2x2 group matrices and they are (1+6r) and 3, while the eigenvalue of the single element is the element itself (2+3r). Thus, the eigenvalues of A_1 or A_2 are (1+6r) and 3 each of multiplicity of (numbered groups) plus the single element (2+3r).

Let U be the true solution of (5.1.1) then

$$(A_1+A_2)U = g \ , \tag{5.4.1}$$

and

$$(A_1+\rho I)U = g-(A_2-\rho I)U \ . \tag{5.4.2}$$

Therefore by (5.3.5a) we have

$$(A_1+\rho I)e^{(k+\frac{1}{2})} = (A_2-\rho I)e^{(k)} \ , \tag{5.4.3}$$

similarly,

$$(A_2+\rho I)e^{(k+1)} = (A_1-\rho I)e^{(k+\frac{1}{2})} \ , \tag{5.4.4}$$

where $e^{(k)} = U-u^{(k)}$ represents the error vector.

Hence

$$e^{(k+1)} = T_\rho e^{(k)} \ , \tag{5.4.5}$$

where

$$T_\rho = (A_2+\rho I)^{-1}(A_1-\rho I)(A_1+\rho I)^{-1}(A_2-\rho I) \ . \tag{5.4.6}$$

We need the spectral radius of T_ρ to be less than unity for the method to be convergent.

Let

$$\tilde{T}_\rho = (A_2+\rho I)T\,(A_2+\rho I)^{-1} \ , \tag{5.4.7}$$

then $\tilde{T}_\rho$ is similar to T_ρ, and

$$\tilde{T}_\rho = (A_1-\rho I)(A_1+\rho I)^{-1}(A_2-\rho I)(A_2+\rho I)^{-1} \ . \tag{5.4.8}$$

The spectral norm of (5.4.9) leads to

$$||\tilde{T}_\rho||_2 = ||(A_1-\rho I)(A_1+\rho I)^{-1}(A_2-\rho I)(A_2+\rho I)^{-1}||_2 ,$$

and

$$||\tilde{T}_\rho||_2 \leq ||(A_1-\rho I)(A_1+\rho I)^{-1}||_2 ||(A_2-\rho I)(A_2+\rho I)^{-1}||_2 . \quad (5.4.9)$$

But we have

$$||(A_1-\rho I)(A_1+\rho I)^{-1}||_2 = \max_\lambda \left|\frac{\lambda-\rho}{\lambda+\rho}\right| , \quad (5.4.10)$$

where λ ranges over all the eigenvalues of A_1.

As we have found that λ is positive, therefore

$$||(A_1-\rho I)(A_1+\rho I)^{-1}||_2 \leq 1 . \quad (5.4.11)$$

Similarly for A_2 we have,

$$||(A_2-\rho I)(A_2+\rho I)^{-1}|| \leq 1 , \quad (5.4.12)$$

and thus,

$$||\tilde{T}_\rho||_2 \leq 1 ,$$

or

$$||T_\rho||_2 \leq 1 , \quad (5.4.13)$$

hence convergence immediately follows.

5.5 EIGENVALUE AND ITERATION PARAMETERS

There is extensive research underway into what kind of iteration parameters, how to choose them and how many to use in order to have an optimum number of iterations to achieve convergence.

a) Stationary Case; ρ is constant

In the case of one acceleration parameter, it is found from Young (1971, p.504) that,

$$\rho = \sqrt{ab} \ , \tag{5.5.1}$$

is the optimum value where a and b are such that,

$$0 < a \leq \mu, \nu \leq b \ , \tag{5.5.2}$$

and μ and ν are the eigenvalues of the matrices A_1 and A_2 respectively.

From the previous section, the spectral radius of the matrix T_ρ is given by,

$$S(T_\rho) \leq S((G_1-\rho I)(G_1+\rho I)^{-1}) S((G_2-\rho I)(G_2+\rho I)^{-1})$$

$$= \left(\max_{a\leq\mu\leq b} \left|\frac{\mu-\rho}{\mu+\rho}\right|\right)\left(\max_{a\leq\nu\leq b} \left|\frac{\nu-\rho}{\nu+\rho}\right|\right) \tag{5.5.3}$$

$$= \left(\max_{a\leq\gamma\leq b} \left|\frac{\gamma-\rho}{\gamma+\rho}\right|\right)^2 = \phi(a,b,\rho) \ . \tag{5.5.4}$$

Since $(\gamma-\rho)/(\gamma+\rho)$ is an increasing function of γ we have,

$$\max_{a\leq\gamma\leq b} \left|\frac{\gamma-\rho}{\gamma+\rho}\right| = \max\left(\left|\frac{a-\rho}{a+\rho}\right|, \left|\frac{b-\rho}{b+\rho}\right|\right) . \tag{5.5.5}$$

Now when $\rho=\sqrt{ab}$, we get,

$$\left|\frac{a-\rho}{a+\rho}\right| = \left|\frac{b-\rho}{b+\rho}\right| = \frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}} \ . \tag{5.5.6}$$

If $0<\rho<\sqrt{ab}$, then,

$$\left|\frac{b-\rho}{b+\rho}\right| - \frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}} = \frac{2\sqrt{b}(\sqrt{ab}-\rho)}{(b+\rho)(\sqrt{b}+\sqrt{a})} > 0 \ . \tag{5.5.7}$$

Also, if $\sqrt{ab}<\rho$, then,

$$\left|\frac{a-\rho}{a+\rho}\right| - \frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}} = \frac{2\sqrt{b}(\rho-\sqrt{ab})}{(\rho+a)(\sqrt{b}+\sqrt{a})} > 0 . \qquad (5.5.8)$$

Thus $\rho=\sqrt{ab}$ is an optimum value that minimizes $\phi(a,b;\rho)$ and,

$$S(T_{\sqrt{ab}}) \leq \phi(a,b;\rho) = \left(\frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}\right)^2 .$$

Since in the case of A_1 and A_2 where the eigenvalues are known, we can determine the optimum parameter quite easily. Evidently the three eigenvalues coincide when r=1/3 to give the eigenvalue 3 of multiplicity n, where n is the order of either of the matrices.

b) Non-stationary case

In this case where we allow ρ_n to vary from one iteration to another in a cyclic manner, we aim to reduce the number of iterations.

For ρ_i, i=1,...,n, we shall need that,

$$\left\| \prod_{i=1}^{n} T_{\rho_i} \right\| = S\left(\prod_{i=1}^{n} T_{\rho_i} \right) = \max_{1\leq j\leq m} \prod_{i=1}^{n} \left(\left|\frac{\rho_i-\mu_j}{\rho_i+\mu_j}\right| . \left|\frac{\rho_i-\nu_j}{\rho_i+\nu_j}\right| \right) < 1. \qquad (5.5.9)$$

and the requirement that A_1 and A_2 are commutative.

Therefore if we know all the eigenvalues of A_1 or A_2 a priori we could choose a sequence of positive numbers ρ_i so that,

$$\left\| \prod_{i=1}^{n} T_{\rho_i} \right\| = 0 , \qquad (5.5.10)$$

and we would then have a direct method, [VARGA, R.S. 1962].

Although this is easily done in our case, we should also state what Peaceman and Rachford presented as a choice of parameters [PEACEMAN, D.W. and RACHFORD, H.H. Jr., 1955].

They supposed that the eigenvalues μ of A_1 and ν of A_2 are such that,

$$a \leq \mu,\nu \leq b \ , \tag{5.5.11}$$

where a and b are the least and greatest eigenvalues respectively.

The Peaceman and Rachford parameters are given by,

$$\rho_i(p) = b(\frac{a}{b})^{(2i-1)/2m} \ , \ i=1,2,\ldots,m. \tag{5.5.12}$$

Another choice of parameters is given by Wachspress [1960] which is,

$$\rho_i(w) = b(\frac{a}{b})^{(i-1)/(m-1)} \ , \ m\geq 2, \ i=1,2,\ldots,m, \tag{5.5.13}$$

where a and b are as defined above.

In Table (5.5.1) we present the eigenvalues of A_1 or A_2, which can be used as acceleration parameters, also the Peaceman Rachford (P.R.) and Wachspress (W) parameters for some values of r. In the table, m represents the number of parameters, e.v. is the eigenvalues or the theoretical parameters which should be used.

In Table (5.5.1) for m=1, column e.v. contains the mean value of the eigenvalues and for m=2 it contains the largest and the smallest eigenvalues.

Unfortunately the rate of convergence for the non-stationary case improves by an appropriate choice of the iteration parameters if and only if the matrices A_1 and A_2 commute, i.e.,

$$A_1A_2 = A_2A_1 \ , \tag{5.5.14}$$

and this is not satisfied for our matrices A_1 and A_2. Even in the case of periodic boundary conditions the commutivity condition is only satisfied when A_1 and A_2 are of order 4.

Now we see how the eigenvalues λ of the iteration matrix

$$T = \frac{(A_1-\rho I)(A_2-\rho I)}{(A_1+\rho I)(A_2+\rho I)} \ ,$$

r	m	e.v.	P.R.	W.
0.1	1	2.3	2.191	-
	2	3.0	2.564	3.0
		1.6	1.872	1.6
	3	3.0	2.702	3.0
		2.3	2.191	2.191
		1.6	1.777	1.6
	4	3.0	2.773	3.0
		2.3	2.370	2.433
		1.6	2.025	1.973
		-	1.731	1.6
	5	3.0	2.817	3.0
		2.3	2.484	2.564
		1.6	2.191	2.191
			1.932	1.872
			1.704	1.6
0.333	any	3	3	3
0.5	1	3.5	3.464	-
	2	4.0	3.722	4.0
		3.0	3.224	3.0
	3	4.0	3.813	4.0
		3.5	3.464	3.464
		3.0	3.147	3.0
	4	4.0	3.859	4.0
		3.5	3.591	3.634
		3.0	3.342	3.302
		-	3.110	3.0
	5	4.0	3.887	4.0
		3.5	3.669	3.722
		3.0	3.464	3.464
		-	3.270	3.224
		-	3.088	3.0
1	1	5.0	4.583	-
	2	7.0	5.664	7.0
		3.0	3.708	3.0
	3	7.0	6.078	7.0
		5.0	4.583	4.583
		3.0	3.455	3.0
	4	7.0	6.297	7.0
		5.0	5.095	5.278
		3.0	4.122	3.979
		-	3.335	3.0
	5	7.0	6.431	7.0
		5.0	5.429	5.664
		3.0	4.583	4.583
		-	3.868	3.708
		-	3.265	3.0

TABLE 5.5.1: continued...

r	m	e.v.	P.R.	W.
2	1	8.0	6.245	-
	2	13.0	9.010	13.0
		3.0	4.328	3.0
	3	13.0	10.181	13.0
		8.0	6.245	6.245
		3.0	3.831	3.0
	4	13.0	10.823	13.0
		8.0	7.501	7.974
		3.0	5.199	4.891
		-	3.603	3.0
	5	13.0	11.227	13.0
		8.0	8.373	9.010
		3.0	6.245	6.245
		-	4.658	4.328
		-	3.474	3.0

TABLE 5.5.1: continued

behave towards the acceleration parameter ρ for different values of r.

Figure (5.5.1), (5.5.2) and (5.5.3) show the behaviour of the eigenvalues of T as ρ changes values. This is done for r with the values 0.1, 0.5 and 1.0 respectively.

In our numerical experiments here we let A_1 and A_2 be (without loss of generality) of order 9.

Let μ_1, μ_2 and μ_3 represent the eigenvalues of A_1 or A_2, where μ_1 and μ_3 are of multiplicity of the number of the groups (which is 4 in this case). We have seen in Section (5.4) that $\mu_1=1+6r$, $\mu_3=3$ and $\mu_2=2+3r$. It is noticeable that,

$$\mu_2 = \frac{\mu_1 + \mu_3}{2} \quad . \tag{5.5.15}$$

Since μ_1 is a function of r (it is less than 3 for $r<1/3$ and greater than 3 for $r>1/3$), we let $\underline{\mu}$ and $\bar{\mu}$ be the smallest and the largest eigenvalues of A_1 respectively.

Initially let ρ be less than $\underline{\mu}$, which makes all the eigenvalues λ_i of T real, and as ρ gets closer to $\underline{\mu}$, the eigenvalues λ_i will get closer to zero. When ρ becomes equal to $\underline{\mu}$ we find that (n-1)/2 of the λ_i (4 in this case) becomes zero and the remaining λ_i are real. Once ρ passes this value of $\underline{\mu}$ to lie within the range between $\underline{\mu}$ and $\bar{\mu}$, some of λ_i will commence to have complex values and will be arranged in such a way that when $\rho=\sqrt{\underline{\mu}\bar{\mu}}$, the $\lambda_i\big|_{i=1}^{9}$ will be on a circle in the complex plane.

Then as ρ increases further, the λ_i will start to diminish the circle and will gradually attain real values again in the opposite order as previously.

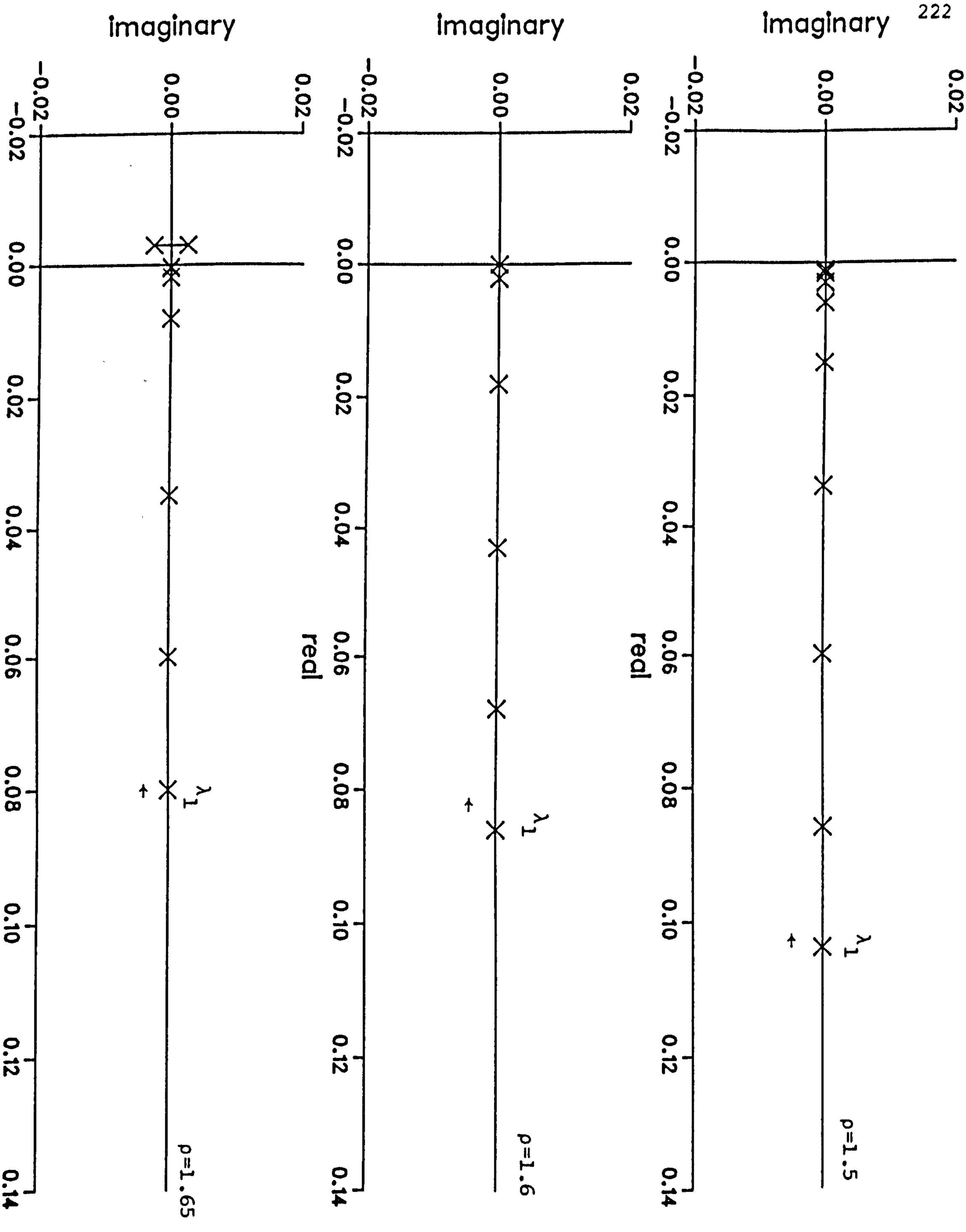


FIGURE 5.5.1: The behaviour of λ_i, i=1,...,9 towards the acceleration parameter ρ. r=0.1

continued...

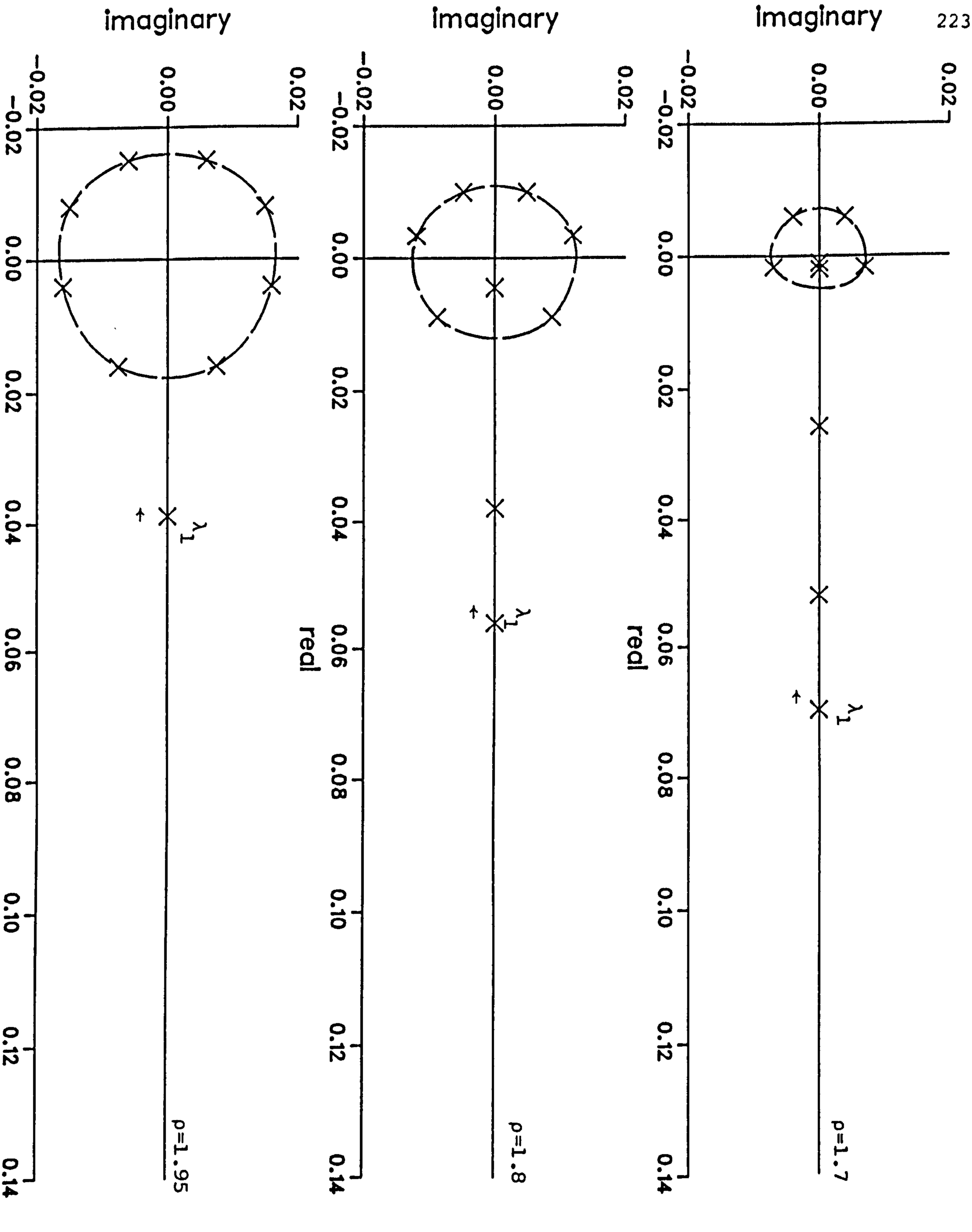


FIGURE 5.5.1: continued

FIGURE 5.5.1: continued

FIGURE 5.5.1: continued

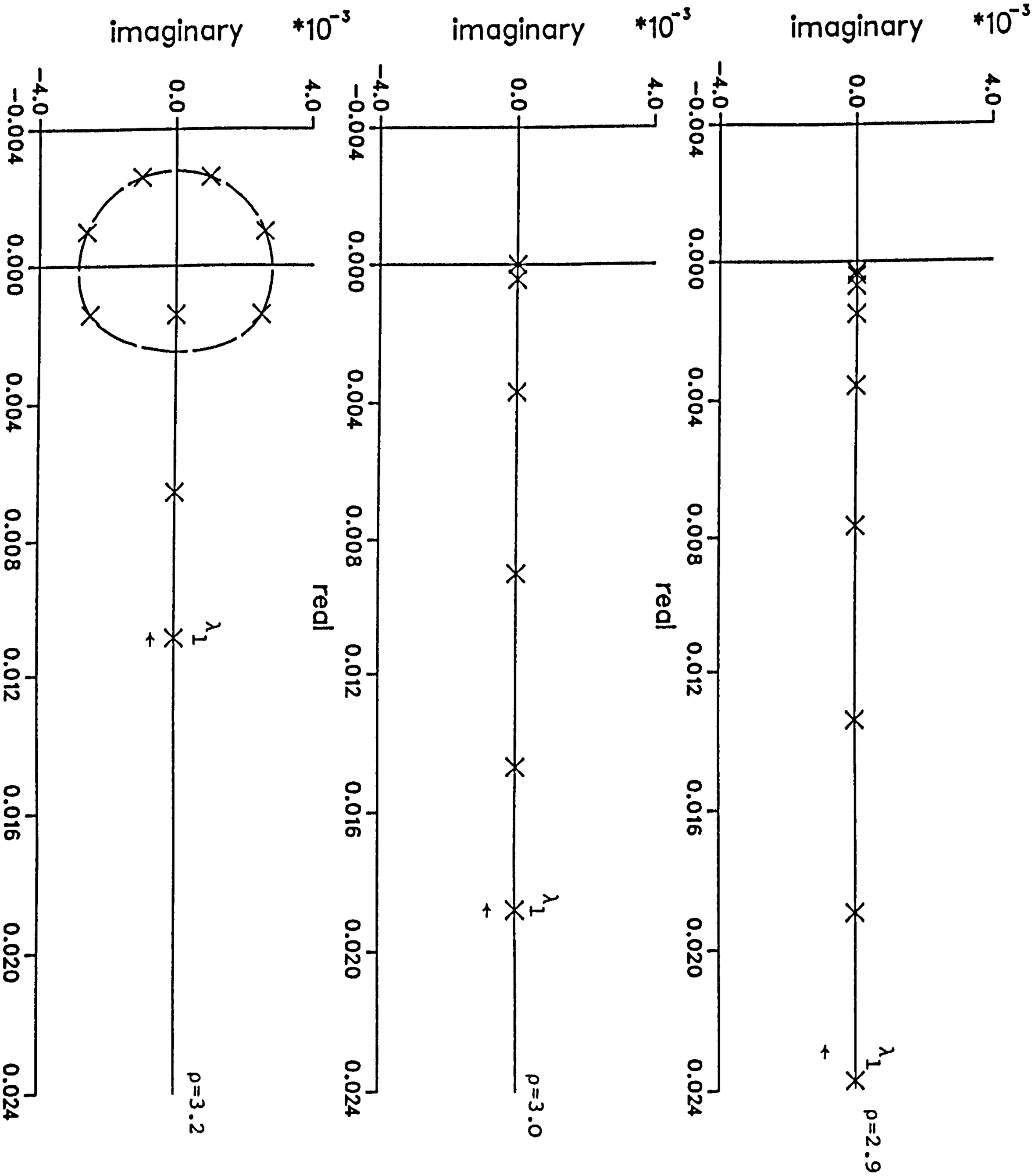


FIGURE 5.5.2: The behaviour of λ_i, i=1,...,9 towards the acceleration parameter ρ. r=0.5

continued...

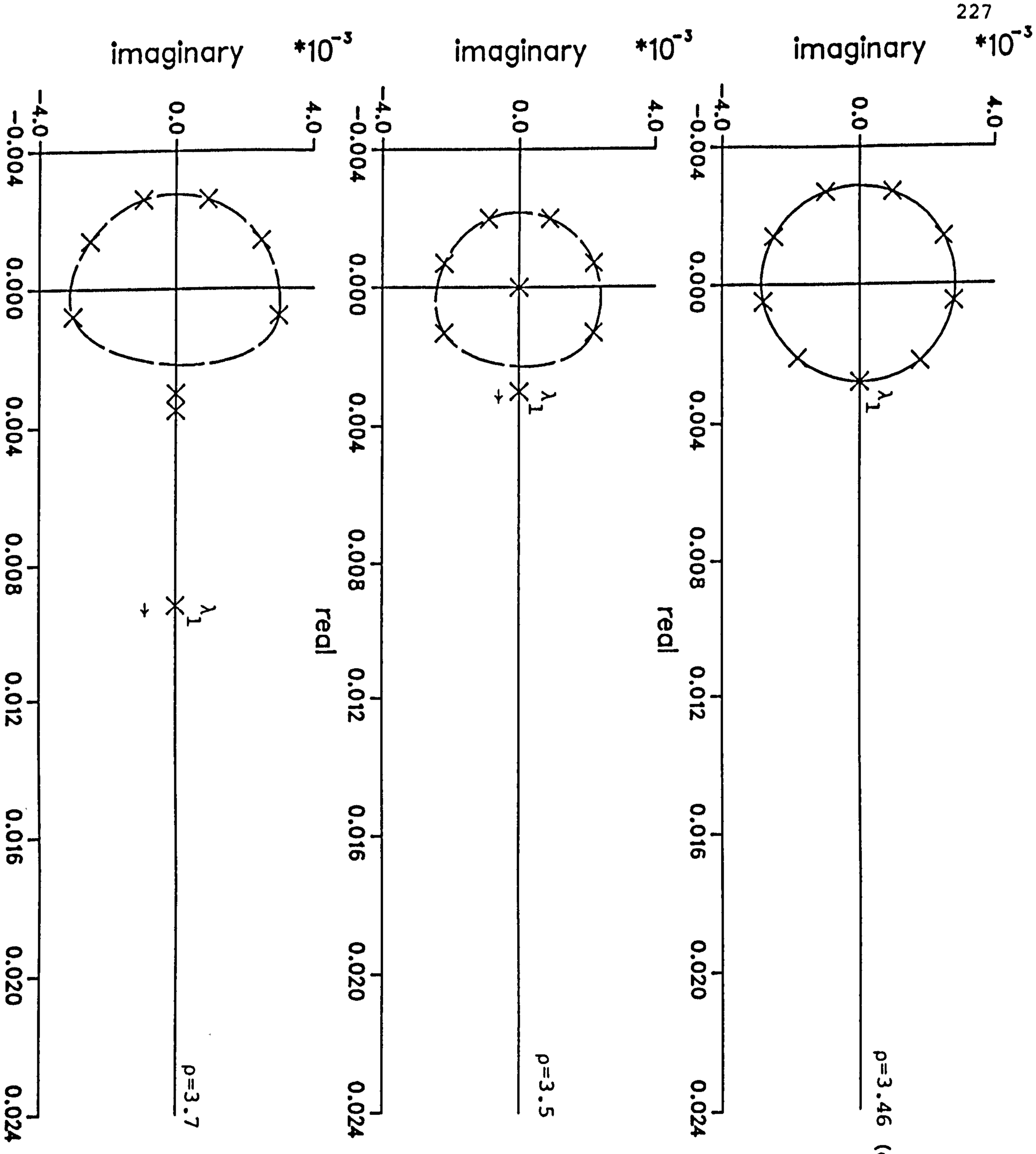


FIGURE 5.5.2: continued

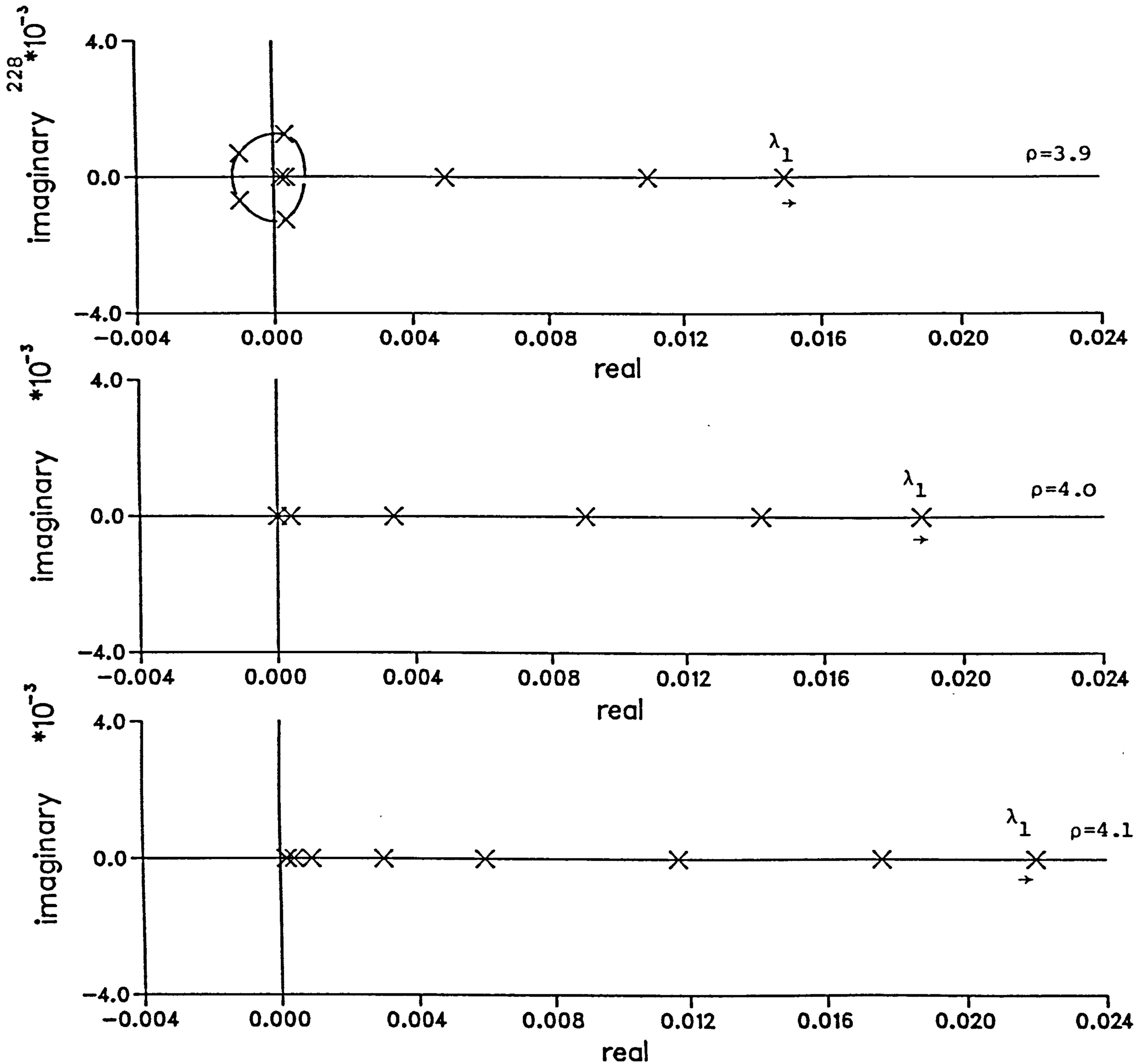


FIGURE 5.5.2: continued

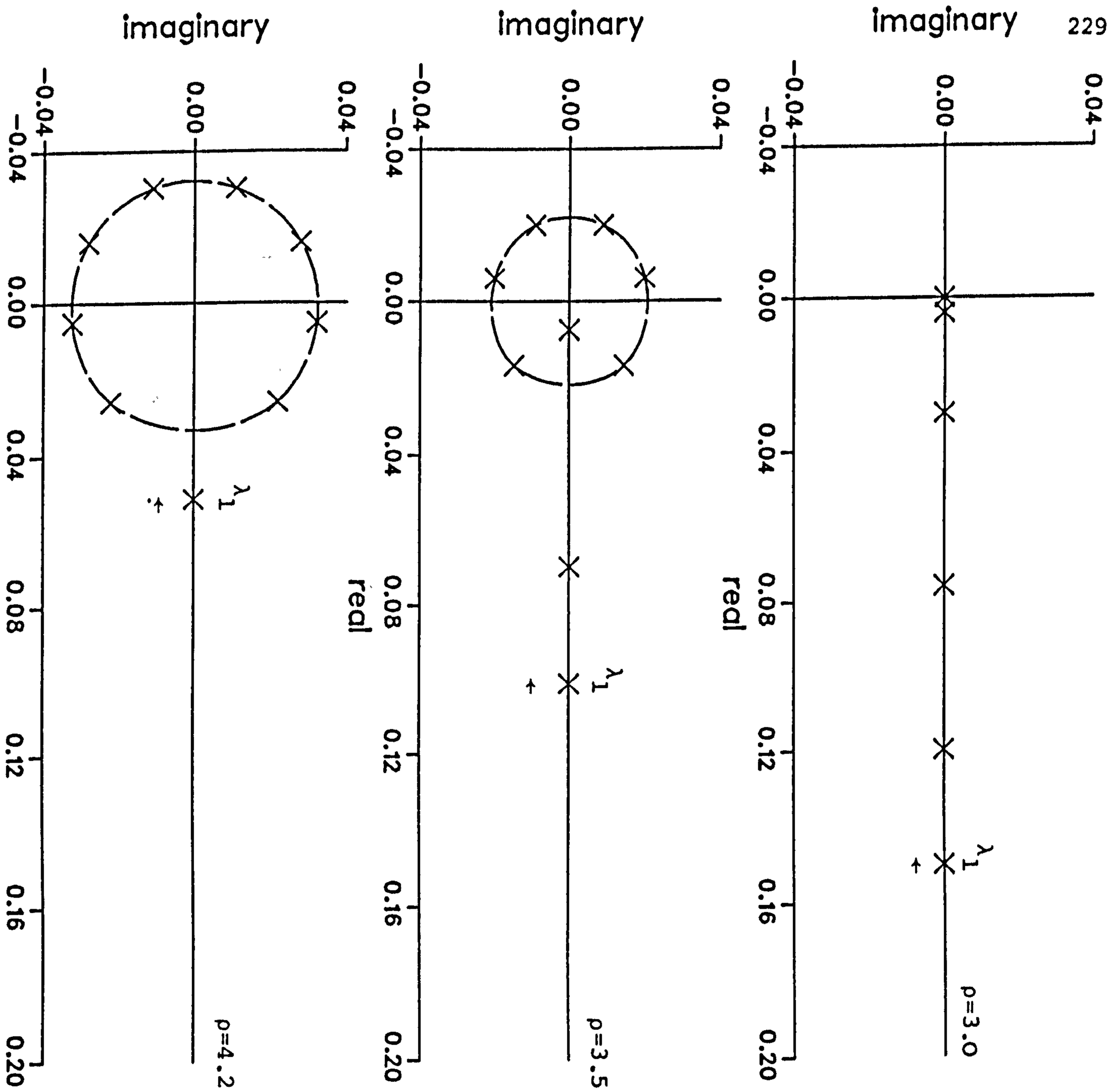


FIGURE 5.5.3: The behaviour of λ_i, i=1,...,9 towards the acceleration parameter ρ.
r=1.0

continued...

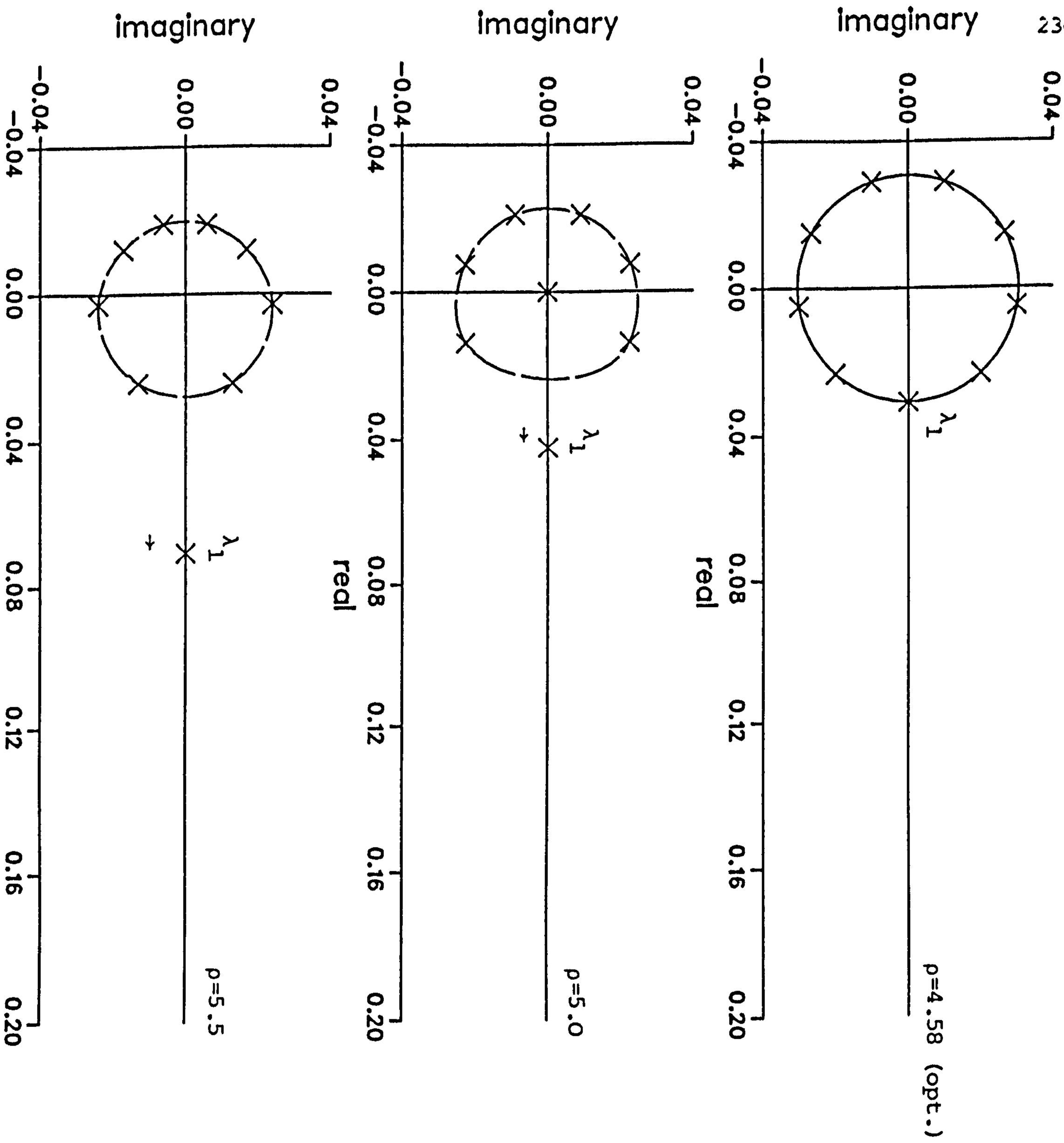


FIGURE 5.5.3: continued

FIGURE 5.5.3: continued

This experiment justifies the optimization of $\rho=\sqrt{ab}$. In addition we notice from Figure (5.5.1) that the eigenvalues λ_i have their minimum modulus when ρ ranges between $\sqrt{\underline{\mu}\bar{\mu}}$ and μ_2, i.e.,

$$\sqrt{\underline{\mu}\bar{\mu}} \leq \rho < \mu_2 \tag{5.5.16}$$

or

$$\sqrt{\underline{\mu}\bar{\mu}} \leq \rho < \frac{\underline{\mu}+\bar{\mu}}{2} \quad . \tag{5.5.17}$$

From the rate of convergence formula,

$$R = -\log_{10}\rho(T) \ , \tag{5.5.18}$$

and in order to have a high rate of convergence for our method we need $\max_{1\leq i\leq n} |\lambda_i|$ to be small. Thus, using $\rho=\sqrt{\underline{\mu}\bar{\mu}}$ to produce n eigenvalues of T equal in modulus enables us to attain the optimum rate of convergence for the stationary case.

But as the λ_i have lower modulus for

$$\sqrt{\underline{\mu}\bar{\mu}} < \rho < \frac{\underline{\mu}+\mu}{2} \ , \tag{5.5.19}$$

which give a higher rate of convergence, this suggests the use of more than one parameter ρ with values within the range (5.5.19) in order to reduce the number of iterations.

This result has been established by Wachspress [Wachspress, E.L., 1962] analytically, where he suggested that the optimum sequence of parameters are in the interval,

$$\sqrt{ab} \leq \rho_i \leq (a+b)/2 \ , \tag{5.5.20}$$

where a and b are the minimum and maximum bounds of the eigenvalues respectively.

The only difference from the numerical experiment is that we have found numerically and geometrically in Figures (5.5.1), (5.5.2) and (5.5.3) that when $\rho=(\underline{\mu}+\bar{\mu})/2$, the iteration matrix T will have at

least one eigenvalue with modulus greater than $\sqrt{\underline{\mu}\bar{\mu}}$ which is not recommended.

In Figure (5.5.1), for r=0.1, $\underline{\mu}$=1.6, μ_2=2.3 and $\bar{\mu}$=3. We notice that when ρ=2.3, the eigenvalue λ_1 is greater than $\sqrt{(1.6)(3)}$.

In Figure (5.5.2), for r=0.5, $\underline{\mu}$=3, μ_2=3.5 and $\bar{\mu}$=4. We notice that λ_1 is greater than $\sqrt{(3)(4)}$ when $\rho=\mu_2$.

Similarly for r=1 in Figure (5.5.3). Therefore we omit the equal sign on the righthand side of (5.5.20) and write it as,

$$\sqrt{ab} \leq \rho_i < (a+b)/2 \ . \qquad (5.5.21)$$

In Figure (5.5.4) the eigenvalues (1+6r), (2+3r) and 3 are shown as E_1, $(E_1+E_3)/2$ and E_3 respectively. The curve sqrt(E1E3) represents the optimum parameter ρ, and the shaded area represents the possible range of optimum acceleration parameters ρ_i as in (5.5.21) which confirms the earlier hypothesis of equation (5.5.16).

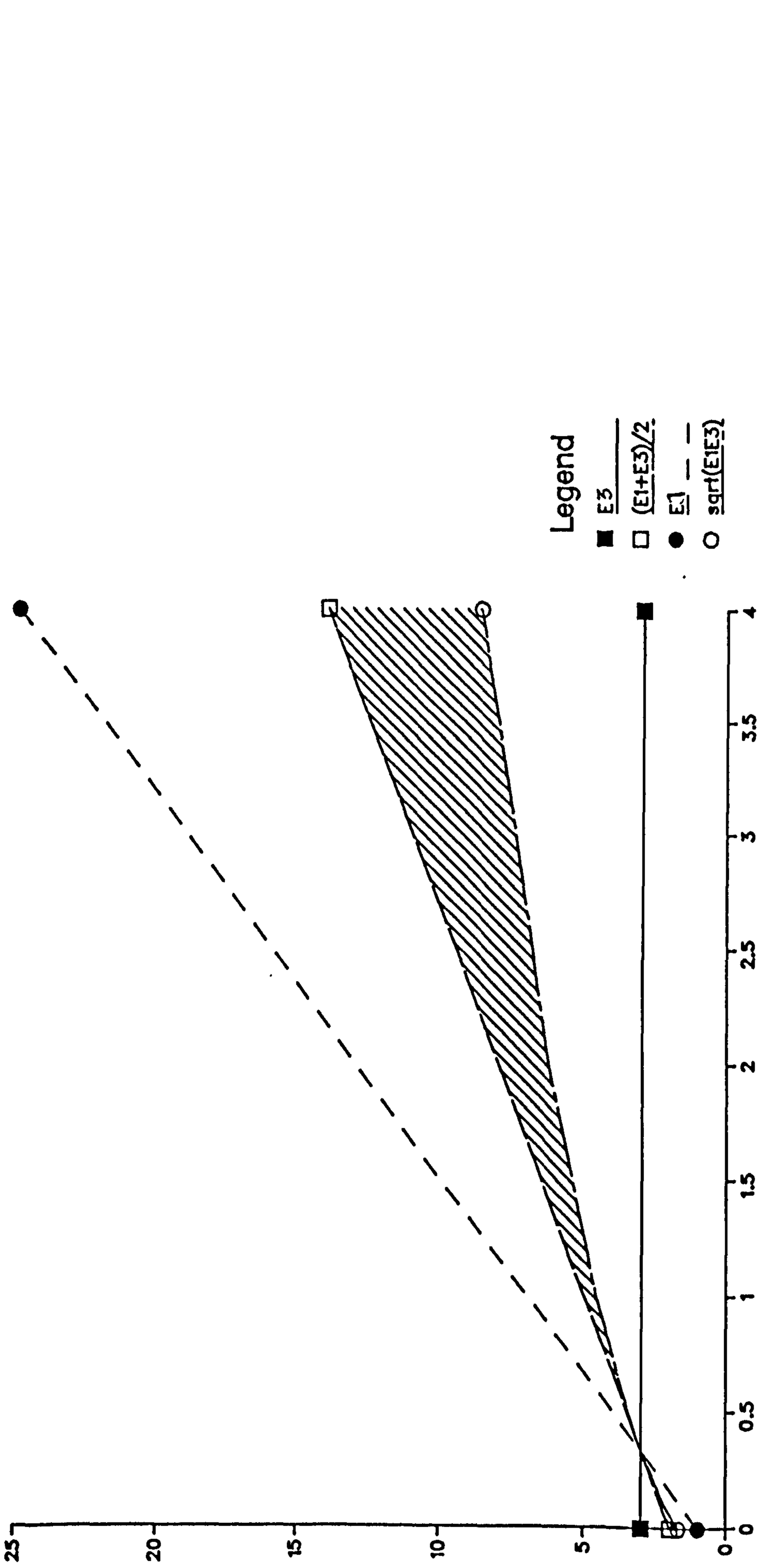


FIGURE 5.5.4: The range of optimum acceleration parameters

$$\sqrt{E_1 E_3} \leq \rho_i < \frac{(E_1+E_3)}{2}$$

5.6 THE SPAGEI METHOD FOR DIFFERENT BOUNDARY CONDITIONS

a) Dirichlet Boundary Condition Case

We have presented in Section 5.3 the case when the domain is divided into an even number of intervals. We now discuss the case when the domain is divided into an odd number of intervals. By splitting the coefficient matrix A into A_1 and A_2 (as before) we get either A_1 (or A_2) with no single element at either end, while A_2 (or A_1) has two single elements, one at each end. This is written as,

$$A = A_1 + A_2 \tag{5.6.1}$$

Thus if A_1 has no single element then,

$$A_1 = \begin{bmatrix} a_1 & b & & & & & \\ c & a_1 & & & & & \\ & & a_1 & b & & & \\ & & c & a_1 & & & \\ & & & & O & & \\ & & & O & & a_1 & b \\ & & & & & c & a_1 \end{bmatrix},$$

and

$$A_2 = \begin{bmatrix} a_1 & & & & & & \\ & a_1 & b & & & & \\ & c & a_1 & & & & \\ & & & O & & & \\ & & O & & & & \\ & & & & a_1 & b & \\ & & & & c & a_1 & \\ & & & & & & a_1 \end{bmatrix}$$

Although the matrix A_1 will now have only two eigenvalues of the same multiplicity, our choice of acceleration parameters will not change. It is obvious that the two eigenvalues are (1+6r) and (3), the missing eigenvalue (2+3r) can still be used as an acceleration parameter since its value is in-between the other two eigenvalues.

For the matrix A_2 the situation has not changed in using the acceleration parameters, although it has the eigenvalue (2+3r) of multiplicity 2.

For the two single points at the ends of the range equation (5.3.14) is used for the point $u_{n-1,j+1}$ and for the point $u_{(1,j+1)}$ equation (5.3.16) is used.

b) Periodic Boundary Condition Case

In this case the boundary conditions are expressed as follows,

$$u(0,t) = u(n,t) \ , \tag{5.6.2}$$

and

$$\frac{\partial u}{\partial x}(0,t) = \frac{\partial u}{\partial x}(n,t) \ . \tag{5.6.3}$$

It is sufficient to use the information of equation (5.6.2) only. Thus, our system of equations will be,

$$(1-3r)(u_{i+1,j+1}+u_{i-1,j+1})+(4+6r)u_{i,j+1} = (1+3r)(u_{i+1,j}+u_{i-1,j}) +(4-r)u_{i,j} \ , \ i=1,\ldots,n, \tag{5.6.4}$$

or since $u_{(i,t)}=u_{(n+i,t)}$ we get,

$$\left.\begin{aligned} (1-3r)(u_{2,j+1}+u_{n,j+1})+(4+6r)u_{1,j+1} &= (1+3r)(u_{2,j}+u_{n,j})+(4-6r)u_{1,j}, \\ (1-3r)(u_{i+1,j+1}+u_{i-1,j+1})+(4+6r)u_{i,j+1} &= (1+3r)(u_{i+1,j}+u_{i-1,j}) \\ &\quad +(4-6r)u_{i,j}, \ i=2,n-1. \\ (1-3r)(u_{1,j+1}+u_{n-1,j+1})+(4+6r)u_{n,j+1} &= (1+3r)(u_{1,j}+u_{n-1,j}) \\ &\quad +(4-6r)u_{n,j} \ . \end{aligned}\right\} \tag{5.6.5}$$

In matrix form the system (5.6.5) will look like this,

$$\begin{bmatrix} 4+6r & 1-3r & & & 1-3r \\ 1-3r & 4+6r & 1-3r & & \\ & \ddots & \ddots & \ddots & \\ & & 1-3r & 4+6r & 1-3r \\ 1-3r & & & 1-3r & 4+6r \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}_{j+1} =$$

$$\begin{bmatrix} 4-6r & 1+3r & & & 1-3r \\ 1+3r & 4-6r & 1+3r & & \\ & \ddots & \ddots & \ddots & \\ & & 1+3r & 4-6r & 1+3r \\ 1-3r & & & 1+3r & 4-6r \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}_{j} , \qquad (5.6.6)$$

or

$$Au_{j+1} = Bu_j \quad . \qquad (5.6.7)$$

Again by splitting A we have,

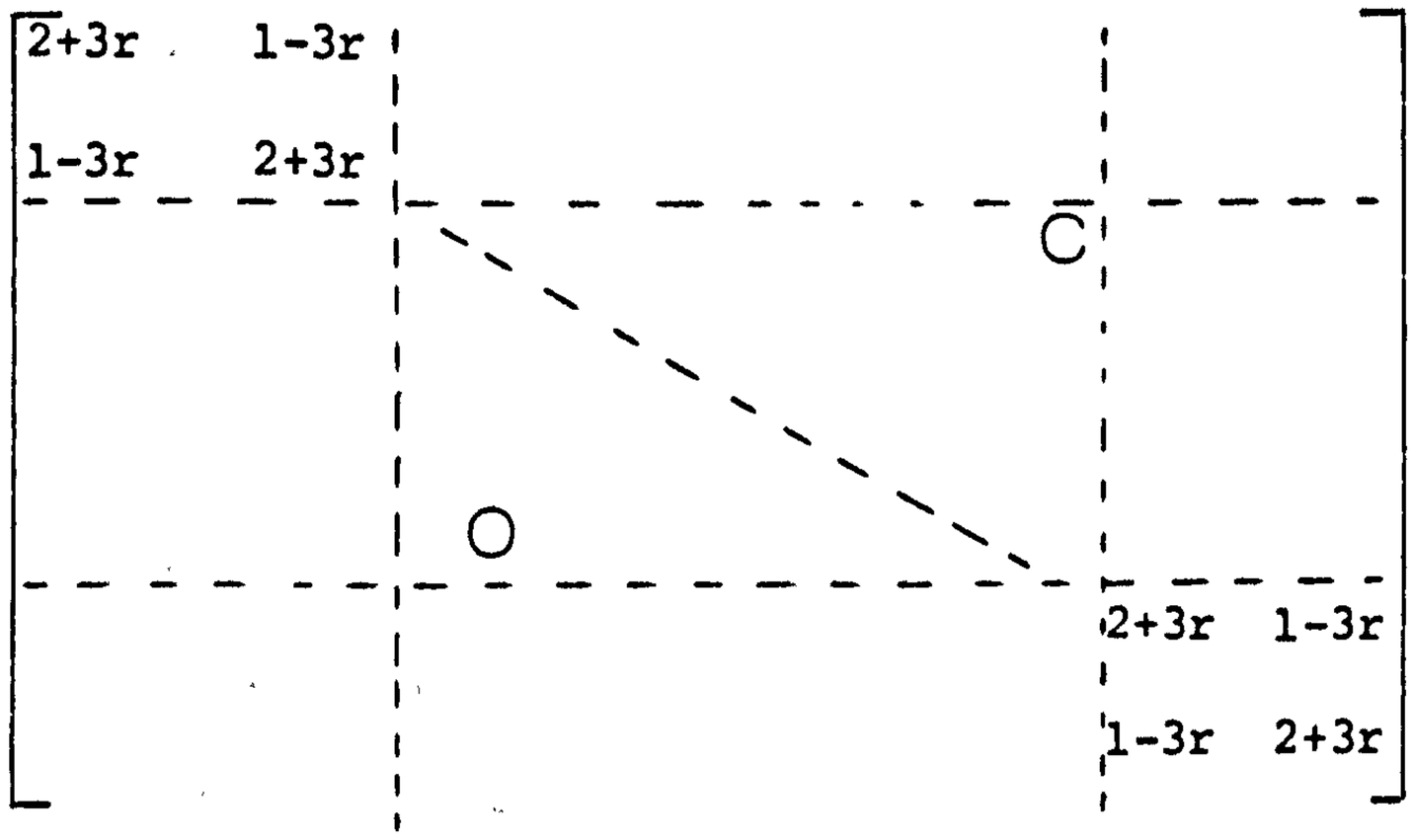
$$A = A_1 + A_2 \ .$$

Now two cases arise by putting the value of n to be even or odd.

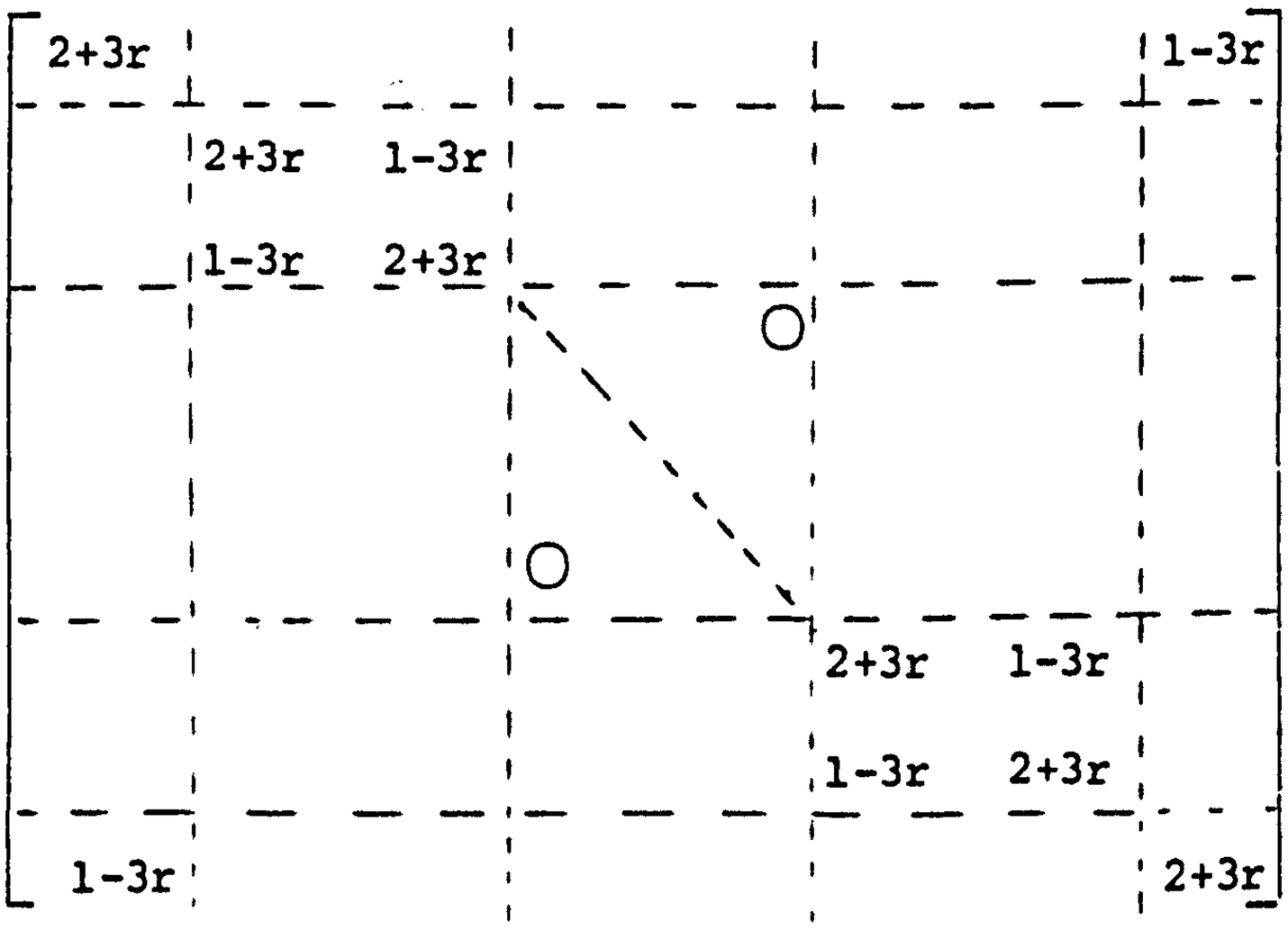
b1) n is an even integer:

This leads to an even number of unknown points. Thus the matrices A_1 and A_2 will look like this,

$A_1 =$



$A_2 =$



In solving the system all 2×2 groups will be treated as in the case of Dirichlet boundary conditions, but in the sweep,

$$(A_2+\rho I)u_{j+1}^{(k+1)} = Bu_j - (A_1-\rho I)u_{j+1}^{(k+\frac{1}{2})} \quad . \tag{5.6.8}$$

two different (2×2) sets of points will appear, one set at each end.

i) For i=1, we will have the equation,

$$(a_1+\rho)u_{1,j+1}^{(k+1)}+bu_{n,j+1}^{(k+1)} = g_1-(a_1-\rho)u_{1,j+1}^{(k+\frac{1}{2})}-bu_{2,j+1}^{(k+\frac{1}{2})} \quad , \tag{5.6.9}$$

where $(a_1+\rho)$, $(a_1-\rho)$ and b are as before, and

$$g_1 = f(u_{n,j}+u_{2,j}) + du_{1,j} \quad , \tag{5.6.10}$$

where f and d are as previously defined.

ii) For i=n, we will have the equation,

$$(a_1+\rho)u_{n,j+1}^{(k+1)} + bu_{1,j+1}^{(k+1)} = g_n-(a_1-\rho)u_{n,j+1}^{(k+\frac{1}{2})}-bu_{n-1,j+1}^{(k+\frac{1}{2})} \quad , \quad (5.6.11)$$

where

$$g_n = f(u_{n-1,j}+u_{1,j}) + du_{n,j} \quad . \quad (5.6.12)$$

We can easily notice that by adding equation (5.6.9) to (5.6.12) gives a (2×2) group which can be solved as any other (2×2) group of our system. This will be

$$\begin{bmatrix} a_1+\rho & b \\ b & a_1+\rho \end{bmatrix} \begin{bmatrix} u_1 \\ u_n \end{bmatrix}_{j+1}^{(k+1)} = \begin{bmatrix} g_1 \\ g_n \end{bmatrix} - \begin{bmatrix} a_1-\rho & b \\ b & a_1-\rho \end{bmatrix} \begin{bmatrix} u_1 \\ u_n \end{bmatrix}_{j+1}^{(k+\frac{1}{2})} - \begin{bmatrix} bu_{2,j+1}^{(k+\frac{1}{2})} \\ bu_{n-1,j+1}^{(k+\frac{1}{2})} \end{bmatrix} \quad . \quad (5.6.13)$$

Inverting the lefthand side coefficient matrix leads to

$$\begin{bmatrix} u_1 \\ u_n \end{bmatrix}_{j+1}^{(k+1)} = \frac{1}{\det} \begin{bmatrix} a_1+\rho & -b \\ -b & a_1+\rho \end{bmatrix} \left\{ \begin{bmatrix} g_1 \\ g_n \end{bmatrix} - \begin{bmatrix} a_1-\rho & 0 \\ 0 & a_1-\rho \end{bmatrix} \begin{bmatrix} u_1 \\ u_n \end{bmatrix}_{j+1}^{(k+\frac{1}{2})} - \begin{bmatrix} bu_{2,j+1}^{(k+\frac{1}{2})} \\ bu_{n-1,j+1}^{(k+\frac{1}{2})} \end{bmatrix} \right\} , \quad (5.6.14)$$

which gives the two equations

$$u_{1,j+1}^{(k+1)} = \frac{1}{\det}[w_1 + su_{n-1,j+1}^{(k+\frac{1}{2})} + tu_{n,j+1}^{(k+\frac{1}{2})} + qu_{1,j+1}^{(k+\frac{1}{2})} + pu_{2,j+1}^{(k+\frac{1}{2})}] \quad (5.6.15)$$

and

$$u_{n,j+1}^{(k+1)} = \frac{1}{\det}[w_n + pu_{n-1,j+1}^{(k+\frac{1}{2})} + qu_{n,j+1}^{(k+\frac{1}{2})} + tu_{1,j+1}^{(k+\frac{1}{2})} + su_{2,j+1}^{(k+\frac{1}{2})}] \quad , \quad (5.6.16)$$

where

$$w_1 = (a_1+\rho)g_1-bg_n, \quad w_n = (a_1-\rho)g_n-bg_1,$$

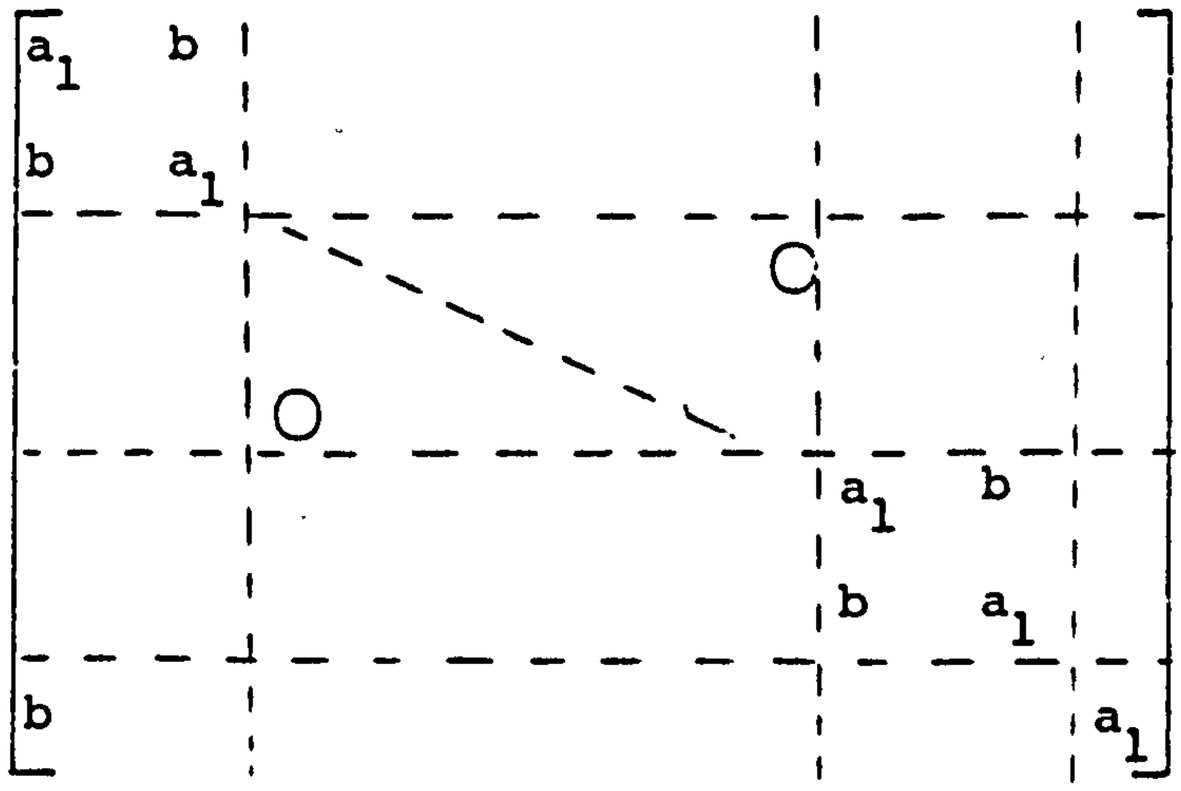
p,q,s and t are as defined in Section 5.3.

b2) n is an odd integer:

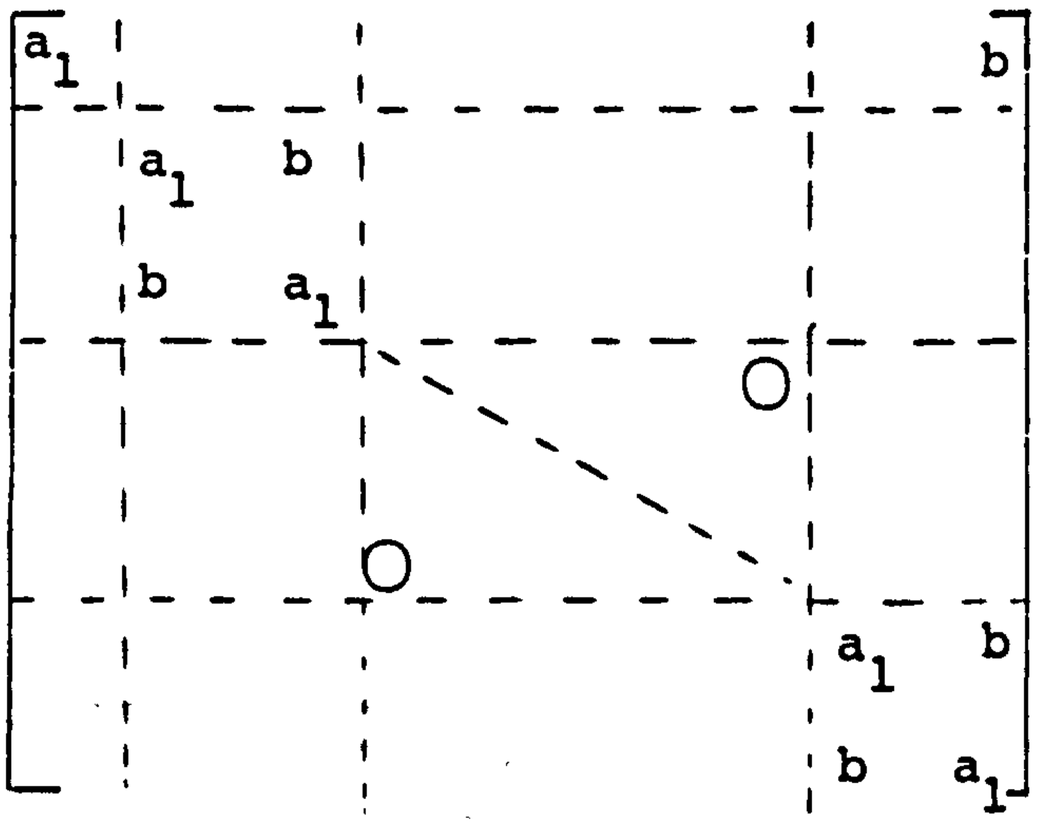
This obviously leads to an odd number of unknown points, and we have two ways of splitting the matrix A.

i) When the matrix A is evenly split, then A_1 and A_2 will look like,

$A_1 =$



$A_2 =$



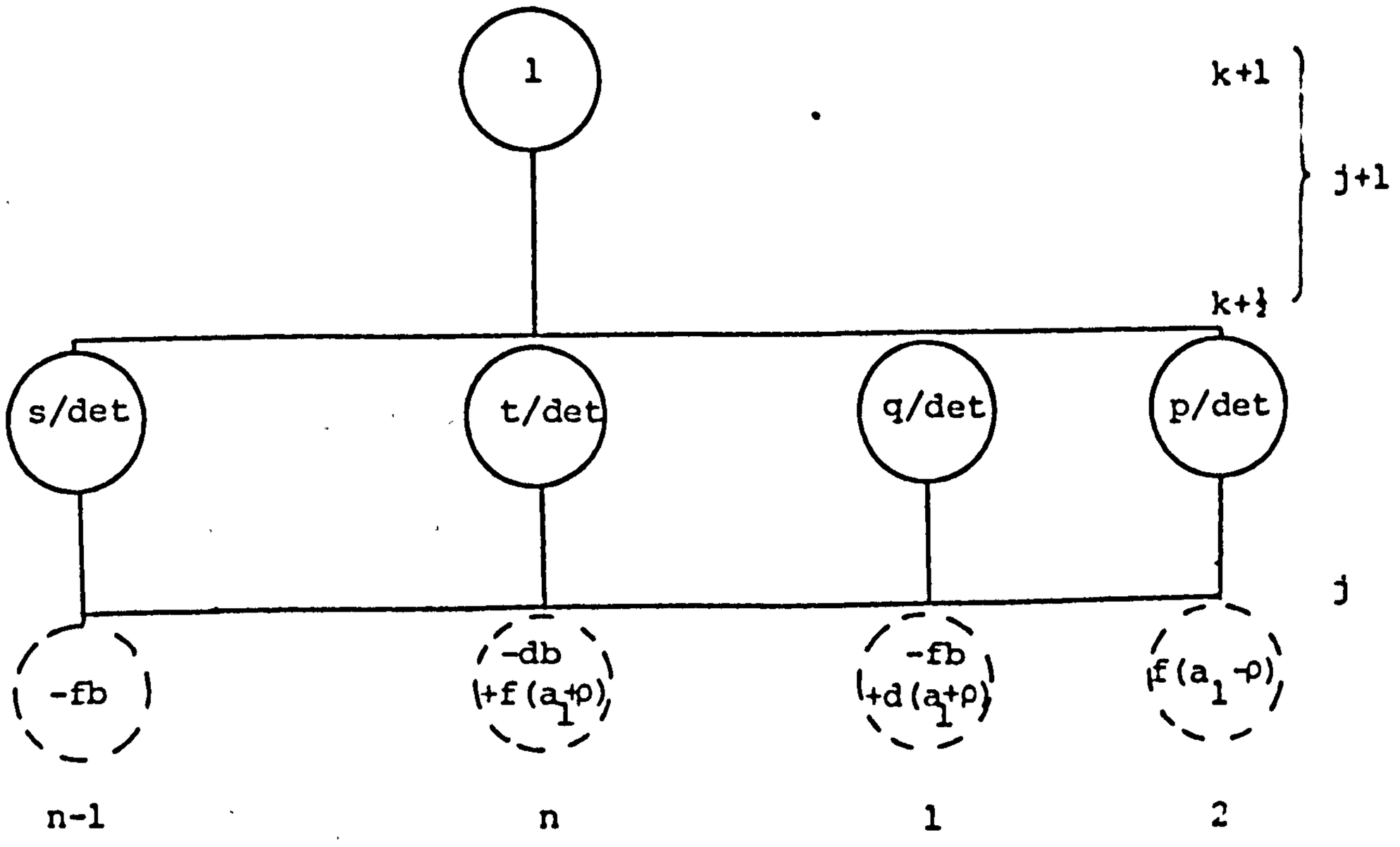
Solving the system for both sweeps will be as previously done, except the end points must be left last to be evaluated. The reason is that each of them is presented as a pair of end points in one sweep. For the sweep,

$$(A_1+\rho I)u_{j+1}^{(k+\frac{1}{2})} = Bu_j-(A_2-\rho I)u_{j+1}^{(k)} ,$$

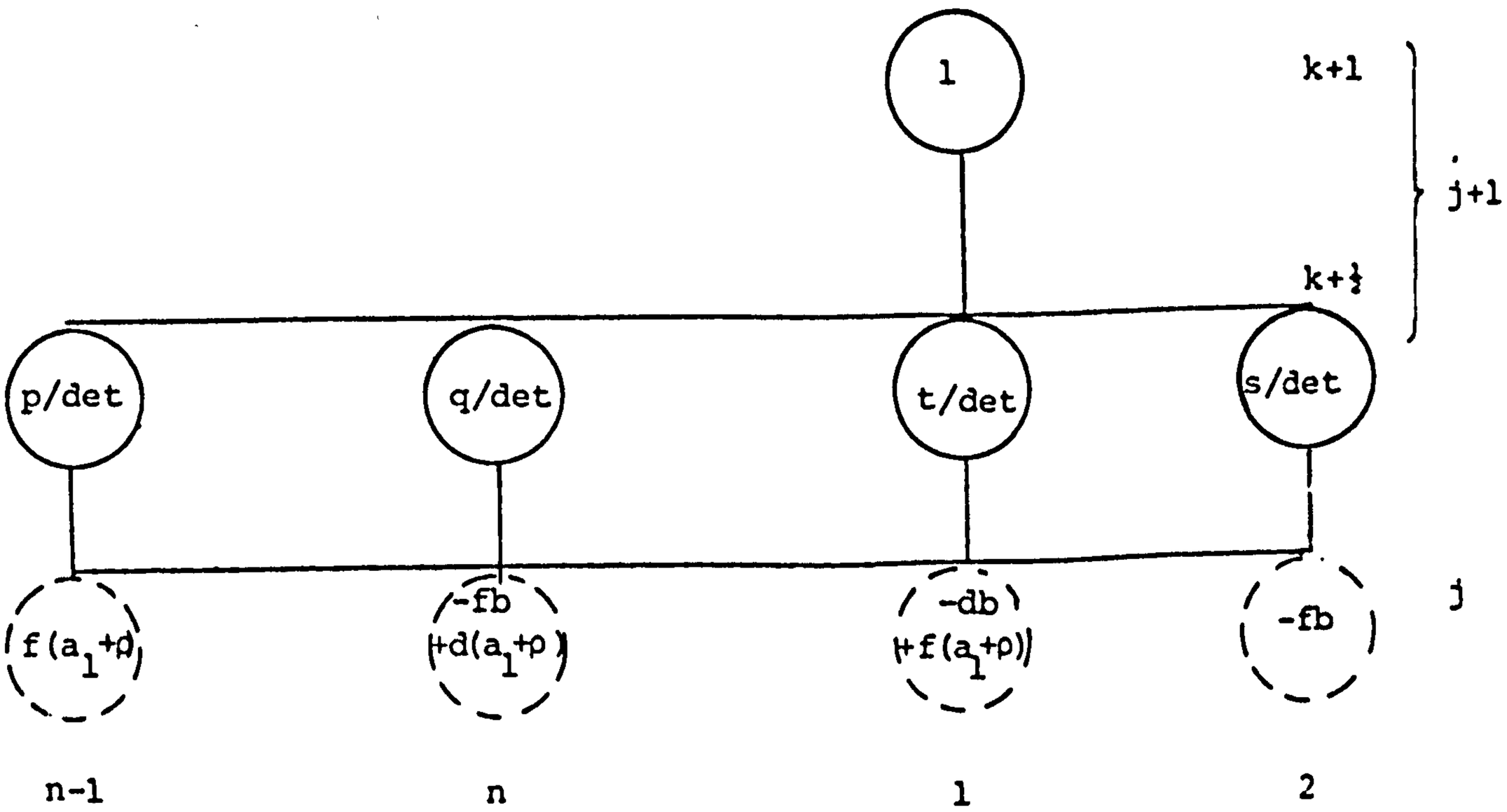
the right end point is presented in the equation,

$$(a_1+\rho)u_{n,j+1}^{(k+\frac{1}{2})}+bu_{1,j+1}^{(k+\frac{1}{2})} = g_n-bu_{n-1,j+1}^{(k)}-(a_1-\rho)u_{n,j+1}^{(k)} , \quad (5.6.17)$$

The computational molecule diagram of equations (5.6.15) and (5.6.16) is shown in Fig.(5.6.1).



(a)



(b)

FIGURE 5.6.1: Computational diagram of equations a) (5.6.15) and b) (5.6.16)

where the function $u_{1,j+1}^{(k+\frac{1}{2})}$ should be known in order to find $u_{n,j+1}^{(k+\frac{1}{2})}$ by

$$u_{n,j+1}^{(k+\frac{1}{2})} = \frac{1}{(a_1+\rho)}\left[g_n - bu_{n-1,j+1}^{(k)} - (a_1-\rho)u_{n,j+1}^{(k)} - bu_{1,j+1}^{(k+\frac{1}{2})}\right]. \tag{5.6.18}$$

While for the sweep

$$(A_2+\rho I)u_{j+1}^{(k+1)} = Bu_j - (A_1-\rho I)u_{j+1}^{(k+\frac{1}{2})}$$

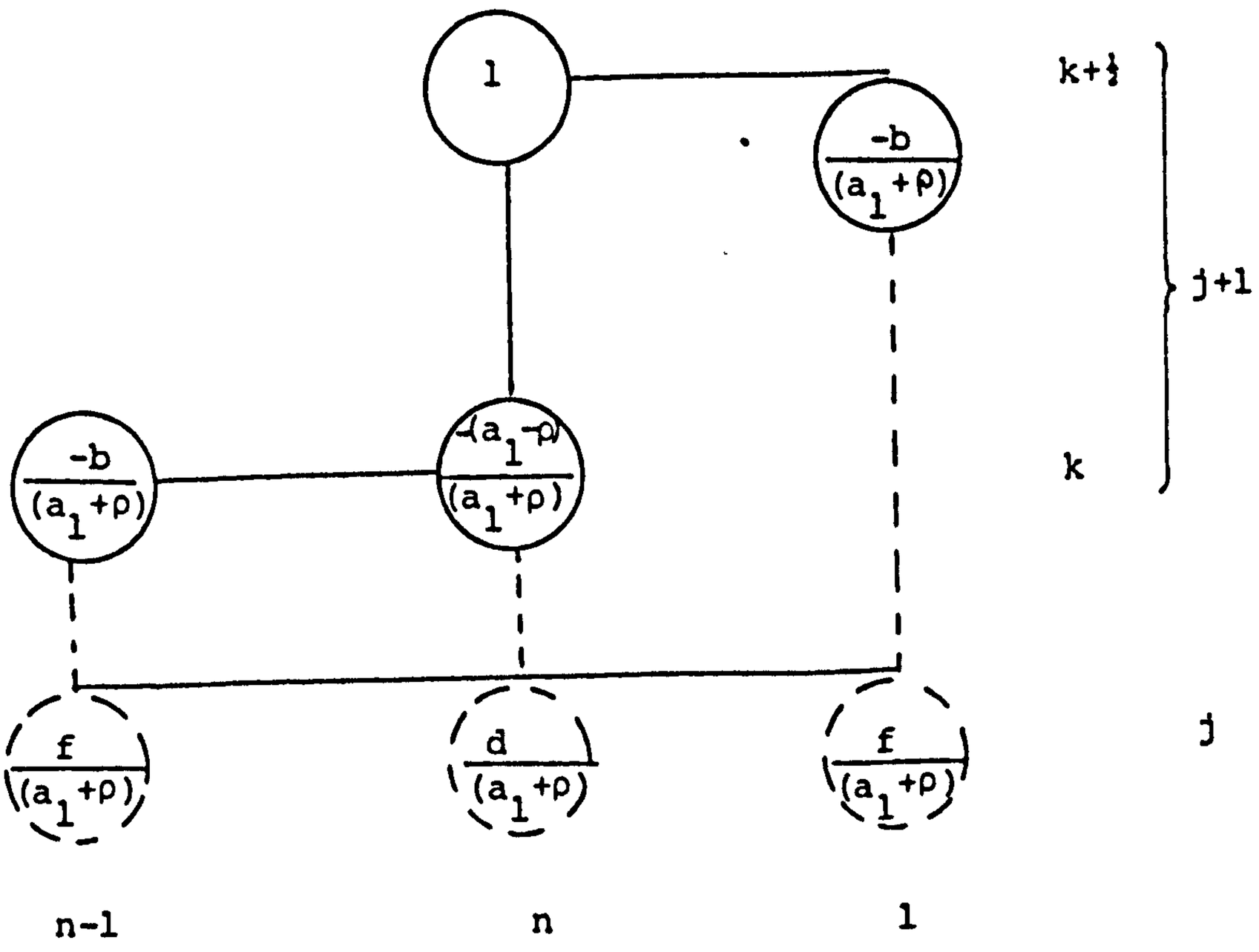
the left end point is presented in the equation,

$$(a_1+\rho)u_{1,j+1}^{(k+1)} + bu_{n,j+1}^{(k+1)} = g_1 - (a_1-\rho)u_{1,j+1}^{(k+\frac{1}{2})} - bu_{2,j+1}^{(k+\frac{1}{2})}, \tag{5.6.19}$$

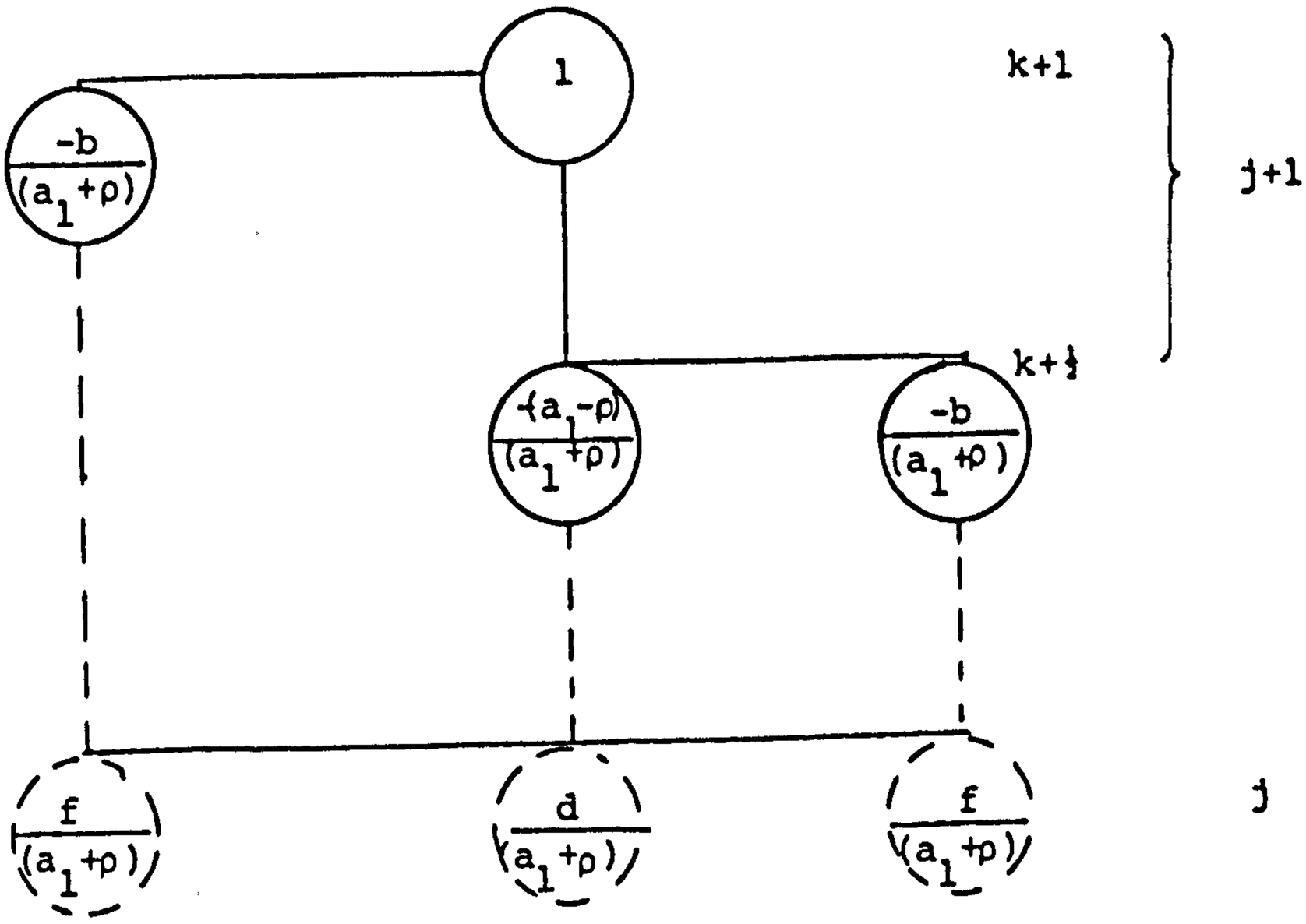
in which the function $u_{n,j+1}^{(k+1)}$ should be known from the (2×2) group at the right end boundary. Therefore equation (5.6.19) is written as,

$$u_{1,j+1}^{(k+1)} = \frac{1}{(a_1+\rho)}\left[g_1 - (a_1-\rho)u_{1,j+1}^{(k+\frac{1}{2})} - bu_{2,j+1}^{(k+\frac{1}{2})} - bu_{n,j+1}^{(k+1)}\right] \tag{5.6.20}$$

The molecular diagram of equations (5.6.18) and (5.6.20) are shown in Fig.(5.6.2).



(a)

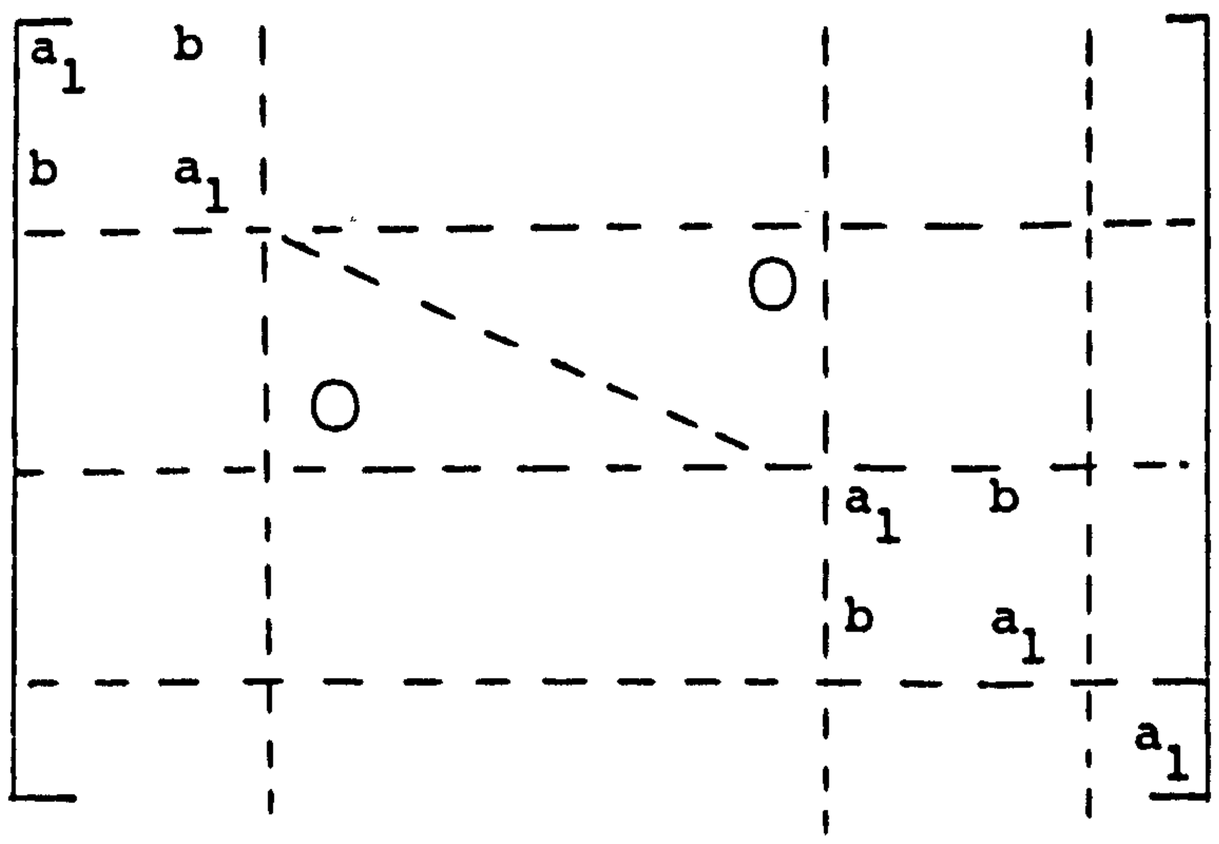


(b)

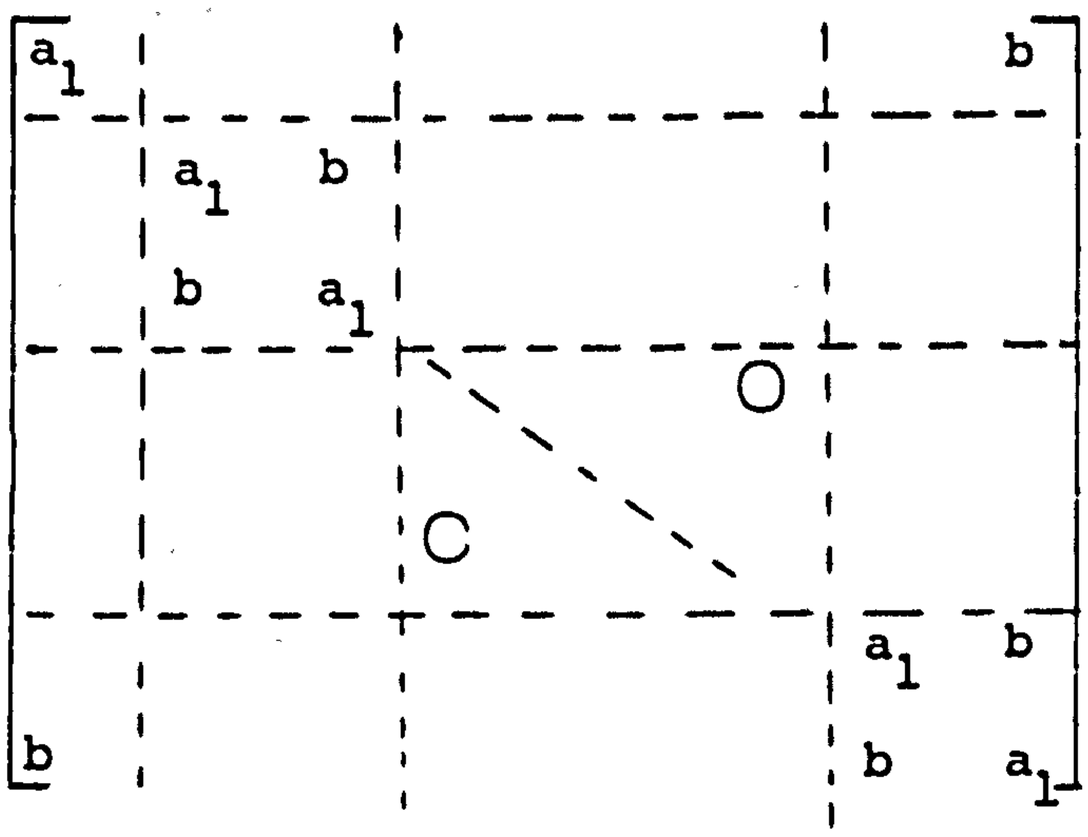
FIGURE 5.6.2: Computational molecules of equation a) (5.6.8) and b) (5.6.20)

ii) When the matrix A is split unevenly, then A_1 and A_2 will look like

$A_1 =$



$A_2 =$



.

This means in the direction

$$(A_1+\rho I)u_{j+1}^{(k+\frac{1}{2})} = BU_j-(A_2-\rho I)u_{j+1}^{(k)}$$

we have one single (ungrouped) point which is at the right end boundary and is presented in the equation

$$u_{n,j+1}^{(k+\frac{1}{2})} = \frac{1}{(a_1+\rho)}\left[g_n-(a_1-\rho)u_{n,j+1}^{(k)}-bu_{1,j+1}^{(k)}\right] . \qquad (5.6.21)$$

While for the direction,

$$(A_2+\rho I)u_{j+1}^{(k+1)} = BU_j-(A_1-\rho I)u_{j+1}^{(k+\frac{1}{2})} .$$

We have at the left end boundary one equation with two unknowns, namely,

$$(a_1+\rho)u_{1,j+1}^{(k+1)}+bu_{n,j+1}^{(k+1)} = g_1-(a_1-\rho)u_{1,j+1}^{(k+\frac{1}{2})}-bu_{2,j+1}^{(k+\frac{1}{2})} , \qquad (5.6.22)$$

and at the right end boundary we have two equations with three unknowns. These are,

$$(a_1+\rho)u_{n-1,j+1}^{(k+1)}+bu_{n,j+1}^{(k+1)} = g_{n-1}-bu_{n-2,j+1}^{(k+\frac{1}{2})}-(a_1-\rho)u_{n-1,j+1}^{(k+\frac{1}{2})} ,$$

and (5.6.23)

$$(a_1+\rho)u_{n,j+1}^{(k+1)}+b(u_{1,j+1}^{(k+1)}+u_{n-1,j+1}^{(k+1)}) = g_n-(a_1-\rho)u_{n,j+1}^{(k+\frac{1}{2})} . \qquad (5.6.24)$$

Neither of the two boundary cases above can be solved independently, since the number of unknowns is more than the number

of equations, but adding them together provides the right information needed to solve them simultaneously. This leads to the 3×3 system,

$$\begin{bmatrix} a_1+\rho & 0 & b \\ 0 & a_1+\rho & b \\ b & b & a_1+\rho \end{bmatrix} \begin{bmatrix} u_1 \\ u_{n-1} \\ u_n \end{bmatrix}^{(k+\frac{1}{2})}_{j+1} = \begin{bmatrix} g_1 \\ g_{n-1} \\ g_n \end{bmatrix} - \begin{bmatrix} a_1-\rho & 0 & 0 \\ 0 & a_1-\rho & 0 \\ 0 & 0 & a_1-\rho \end{bmatrix} \begin{bmatrix} u_1 \\ u_{n-1} \\ u_n \end{bmatrix}^{(k+\frac{1}{2})}_{j+1}$$

$$- \begin{bmatrix} bu^{(k+\frac{1}{2})}_{2,j+1} \\ bu^{(k+\frac{1}{2})}_{n-2,j+1} \\ 0 \end{bmatrix} \qquad (5.6.25)$$

Inverting the l.h.s. coefficient matrix gives,

$$\begin{bmatrix} u_1 \\ u_{n-1} \\ u_n \end{bmatrix}^{(k+1)}_{j+1} = \begin{bmatrix} (a_1+\rho)^2-b^2 & b^2 & -(a_1+\rho)b \\ b^2 & (a_1+\rho)^2-b^2 & -(a_1+\rho)b \\ -(a_1+\rho)b & -(a_1+\rho)b & (a_1+\rho)^2 \end{bmatrix} \left\{ \begin{bmatrix} g_1 \\ g_{n-1} \\ g_n \end{bmatrix} \right.$$

$$\left. -(a_1-\rho) \begin{bmatrix} u_1 \\ u_{n-1} \\ u_n \end{bmatrix}^{(k+\frac{1}{2})}_{j+1} - b \begin{bmatrix} u_2 \\ u_{n-2} \\ 0 \end{bmatrix}^{(k+\frac{1}{2})}_{j+1} \right\} , \qquad (5.6.26)$$

which leads to the three equations for $u_{1,j+1}$, $u_{n-1,j+1}$ and $u_{n,j+1}$,

$$u^{(k+1)}_{1,j+1} = \frac{1}{\det}[Pg_1+b^2g_{n-1}+Qg_n-P(a_1-\rho)u^{(k+\frac{1}{2})}_{1,j+1}+Ru^{(k+\frac{1}{2})}_{n-1,j+1}$$

$$+Sbu^{(k+\frac{1}{2})}_{n,j+1}-bPu^{(k+\frac{1}{2})}_{2,j+1}-b^3u^{(k+\frac{1}{2})}_{n-2,j+1}] , \qquad (5.6.27)$$

$$u^{(k+1)}_{n-1,j+1} = \frac{1}{\det}[b^2g_1+Pg_{n-1}+Qg_n+Ru^{(k+\frac{1}{2})}_{1,j+1}-(a_1-\rho)Pu^{(k+\frac{1}{2})}_{n-1,j+1}$$

$$+Sbu^{(k+\frac{1}{2})}_{n,j+1}-b^3u^{(k+\frac{1}{2})}_{2,j+1}-Pbu^{(k+\frac{1}{2})}_{n-2,j+1}] , \qquad (5.6.28)$$

and

$$u^{(k+1)}_{n,j+1} = \frac{1}{\det}[Qg_1+Qg_{n-1}+(a_1+\rho)^2g_n+Sbu^{(k+\frac{1}{2})}_{1,j+1}+Sbu^{(k+\frac{1}{2})}_{n-1,j+1}$$

$$-(a_1+\rho)Su^{(k+\frac{1}{2})}_{n,j+1}-Qbu^{(k+\frac{1}{2})}_{2,j+1}-Qbu^{(k+\frac{1}{2})}_{n-2,j+1}] , \qquad (5.6.29)$$

where,

$$\det = (a_1+\rho)^3-2(a_1+\rho)b^2 \ ,$$

$$P = (a_1+\rho)^2-b^2 \ ,$$

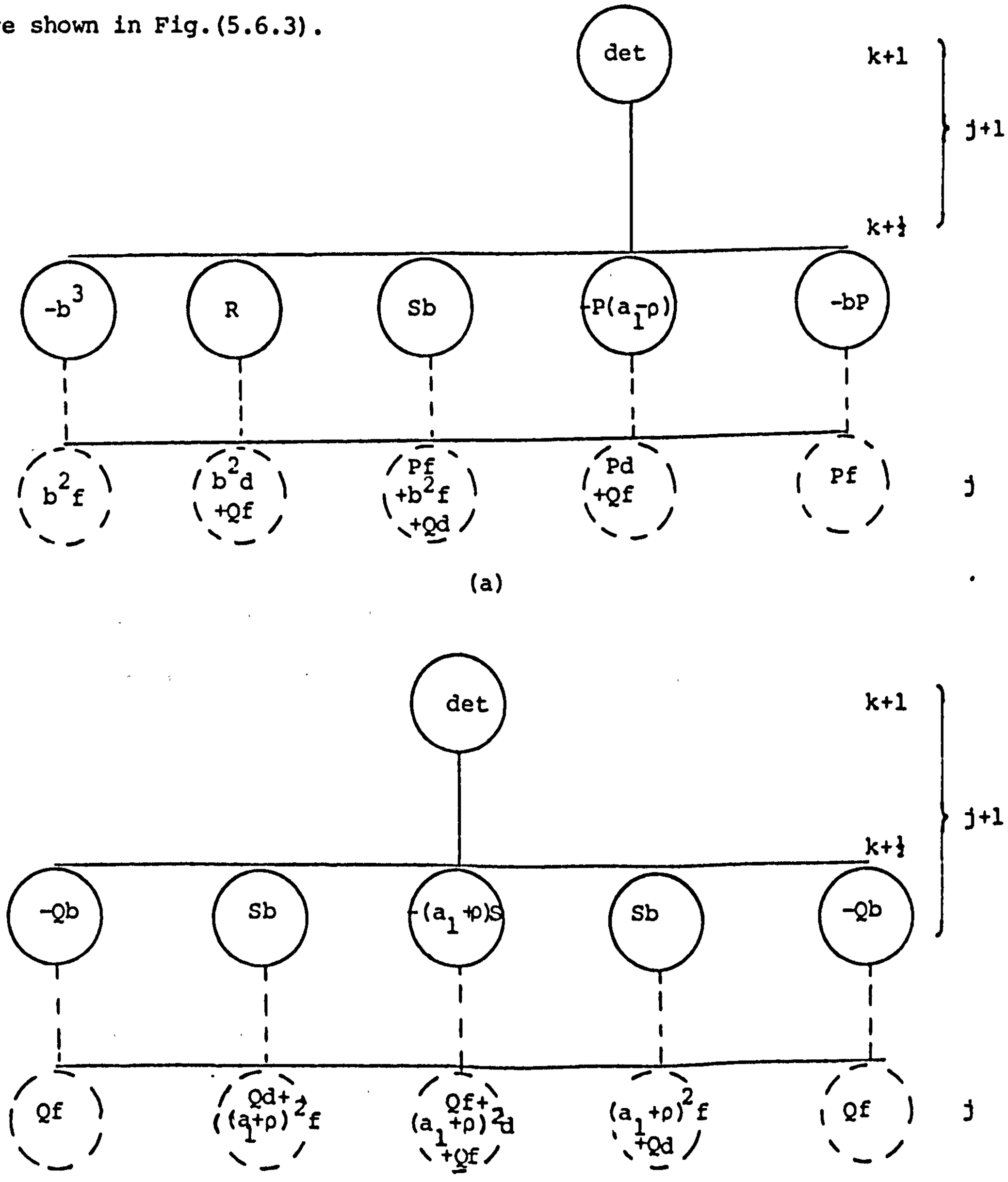
$$Q = -(a_1+\rho)b \ ,$$

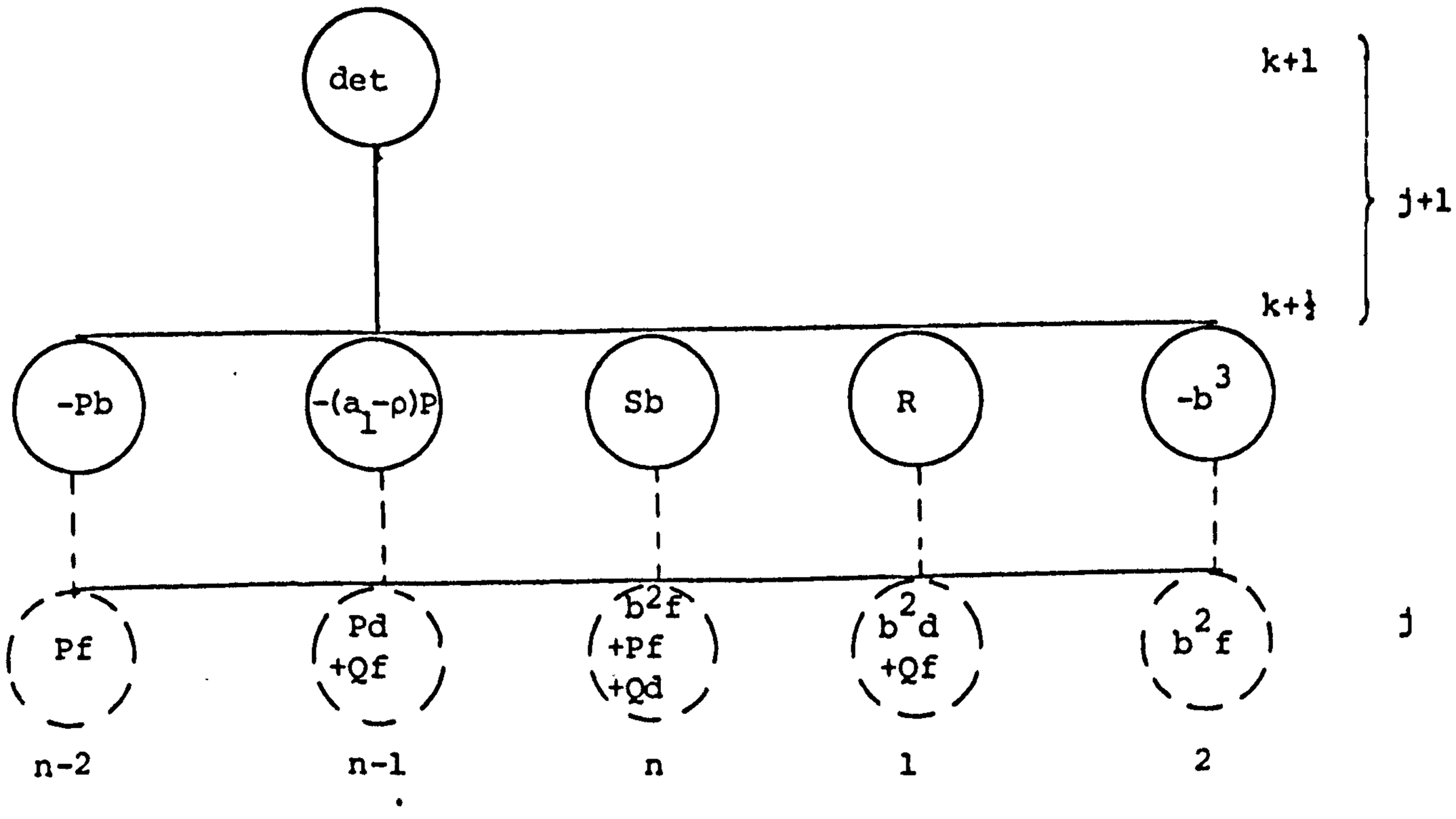
$$R = -b^2(a_1-\rho) \ ,$$

and

$$S = (a_1-\rho)(a_1+\rho) \ .$$

The molecular diagram of equations (5.6.27), (5.6.28) and (5.6.29) are shown in Fig.(5.6.3).





(c)

FIGURE 5.6.3: Molecular representation of equations a) (5.6.27), b) (5.6.29) and c) (5.6.28)

c) Derivative Boundary Condition Case

For this case we have the boundary conditions expressed in terms of the derivative of the unknown function. For the problem given by equation (5.1.1), the derivative boundary conditions are as follows:-

$$\left.\frac{\partial U}{\partial x}\right|_{x=0} = f_1 \,, \tag{5.6.30}$$

$$\left.\frac{\partial U}{\partial x}\right|_{x=1} = f_2 \,. \tag{5.6.31}$$

For this type of boundary condition we use either the central difference or the forward difference approximation to the derivatives involved.

c1) The central difference:

By using this approximation (5.6.30) and (5.6.31) are written as,

$$\frac{u_{1,j}-u_{-1,j}}{2\Delta x} \cong f_1 \quad , \tag{5.6.32}$$

and

$$\frac{u_{n+1,j}-u_{n-1,j}}{2\Delta x} \cong f_2 \quad , \tag{5.6.33}$$

respectively, for any time level j. This approximation is of order (h^2) accurate.

The points (-1,j) and (n+1,j) are not within the given domain and hence are virtual points. To eliminate them we use equation (5.3.1). To determine the function at the points (-1,j) and (-1,j+1), we let i=0 in equation (5.3.1) to give,

$$(1-3r)(u_{-1,j+1}+u_{1,j+1})+(4+6r)u_{0,j+1} = (1+3r)(u_{-1,j}+u_{1,j}) +(4-6r)u_{0,j} \ . \tag{5.6.34}$$

From equation (5.6.32) we have,

$$u_{-1,j} \cong u_{1,j}-2\Delta x f_1 \ , \tag{5.6.35}$$

while for the (j+1)th level we have,

$$u_{-1,j+1} \cong u_{1,j+1}-2\Delta x f_1 \ . \tag{5.6.36}$$

Substituting the values of $u_{-1,j}$ and $u_{-1,j+1}$ in equation (5.6.34) we get,

$$(1-3r)(2u_{1,j+1}-2\Delta x f_1)+(4+6r)u_{0,j+1} = (1+3r)(2u_{1,j}-2\Delta x f_1) +(4-6r)u_{0,j} \ , \tag{5.6.37}$$

which becomes,

$$(4+6r)u_{0,j+1}+2(1-3r)u_{1,j+1} = (4-6r)u_{0,j}+2(1+3r)u_{1,j}-12\Delta x f_1 r. \tag{5.6.38}$$

Similarly for the remaining end point, when i=n, equation

(5.3.1) becomes,

$$(1-3r)(u_{n-1,j+1}+u_{n+1,j+1})+(4+6r)u_{n,j+1} = (1+3r)(u_{n-1,j}+u_{n+1,j}) +(4+6r)u_{n,j} . \quad (5.6.39)$$

Rewriting equation (5.6.33) as,

$$u_{n+1,j} \approx u_{n-1,j} + 2\Delta x f_2 , \quad (5.6.40)$$

and similarly for the (j+1)th level we have,

$$u_{n+1,j+1} \approx u_{n-1,j+1} + 2\Delta x f_2 . \quad (5.6.41)$$

Substituting $u_{n+1,j+1}$ and $u_{n+1,j}$ into equation (5.6.39) we get,

$$(1-3r)(2u_{n-1,j+1}+2\Delta x f_2)+(4+6r)u_{n,j+1} = (1+3r)(2u_{n-1,j}+2\Delta x f_2)+ (4-6r)u_{n,j} , \quad (5.6.42)$$

which becomes,

$$2(1-3r)u_{n-1,j+1}+(4+6r)u_{n,j+1} = 2(1+3r)u_{n-1,j}+(4-6r)u_{n,j} +12\Delta x f_2 r . \quad (5.6.43)$$

Thus, the system of equations will become,

$$\left.\begin{aligned} &(4+6r)u_{0,j+1}+2(1-3r)u_{1,j+1} = (4-6r)u_{0,j}+2(1+3r)u_{1,j}-12\Delta x\, f_1 r , \\ &(1-3r)(u_{i-1,j+1}+u_{i+1,j+1})+(4+6r)u_{i,j+1} = (1+3r)(u_{i-1,j}+u_{i+1,j}) \\ &\qquad +(4-6r)u_{i,j},\ i=1,\ldots,n-1 \\ &2(1-3r)u_{n-1,j+1}+(4+6r)u_{n,j+1} = 2(1+3r)u_{n-1,j}+(4-6r)u_{n,j}+12\Delta x f_n r . \end{aligned}\right\} \quad (5.6.44)$$

In matrix form this is written as,

$$Au_{j+1} = Bu_j + c , \quad (5.6.45)$$

where

$$A = \begin{bmatrix} a & 2b & & & & \\ b & a & b & & & \\ & b & a & b & & \\ & & \ddots & \ddots & \ddots & \\ & & & b & a & b \\ & & & & 2b & a \end{bmatrix},$$

$$B = \begin{bmatrix} d & 2f & & & & \\ f & d & f & & & \\ & f & d & f & & \\ & & \ddots & \ddots & \ddots & \\ & & & f & d & f \\ & & & & 2f & d \end{bmatrix}, \quad u_j = \begin{bmatrix} u_{0,j} \\ u_{1,j} \\ \vdots \\ u_{n-1,j} \\ u_{n,j} \end{bmatrix}$$

and

$$c = [c_0, 0, 0, \ldots, 0, c_n]^T ,$$

where a,b,d and f are as previously defined,

$$c_0 = -12\Delta x f_1 r \qquad \text{and} \qquad c_n = 12\Delta x f_n r$$

By following the same procedure in splitting the matrix A we get,

$$(A_1+\rho I)u_{j+1}^{(k+\frac{1}{2})} = g-(A_2-\rho I)u_{j+1}^{(k)} , \tag{5.6.46}$$

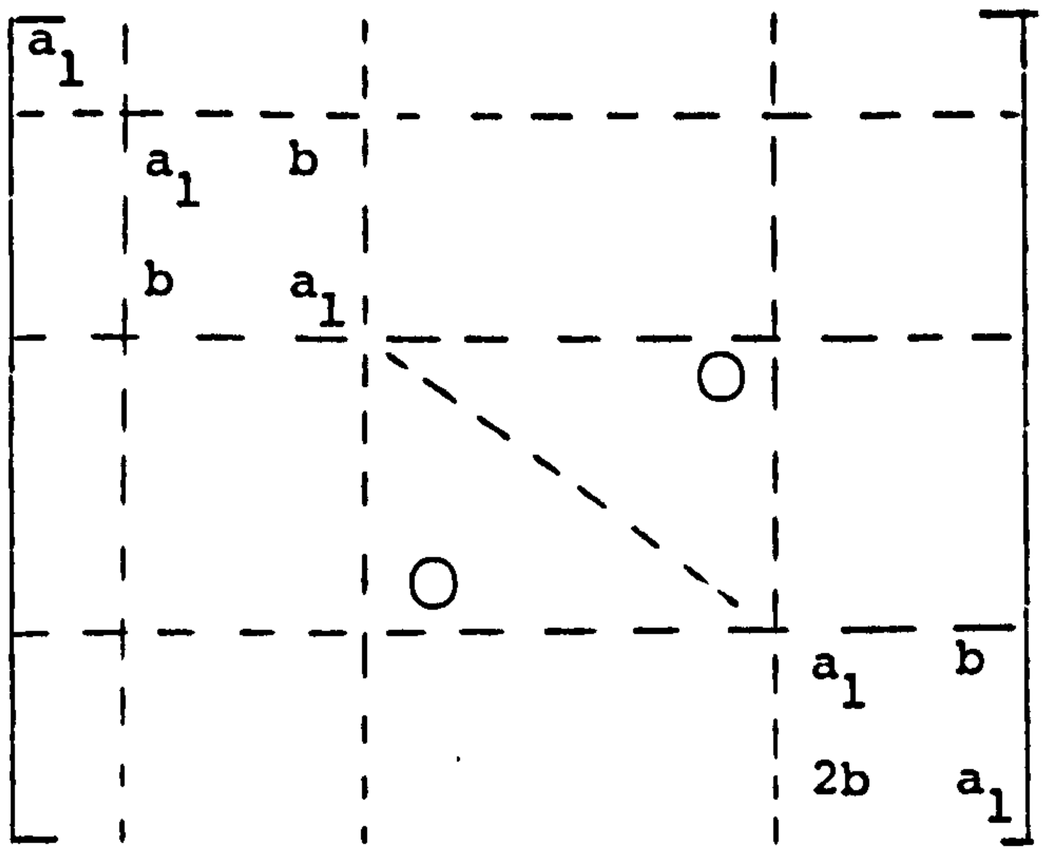
$$(A_2+\rho I)u_{j+1}^{(k+1)} = g-(A_1-\rho I)u_{j+1}^{(k+\frac{1}{2})} . \tag{5.6.47}$$

In the case of n being an even or an odd number we will always have the general 2×2 group, the equations for the end points will have slight differences.

i) When n is an even number:

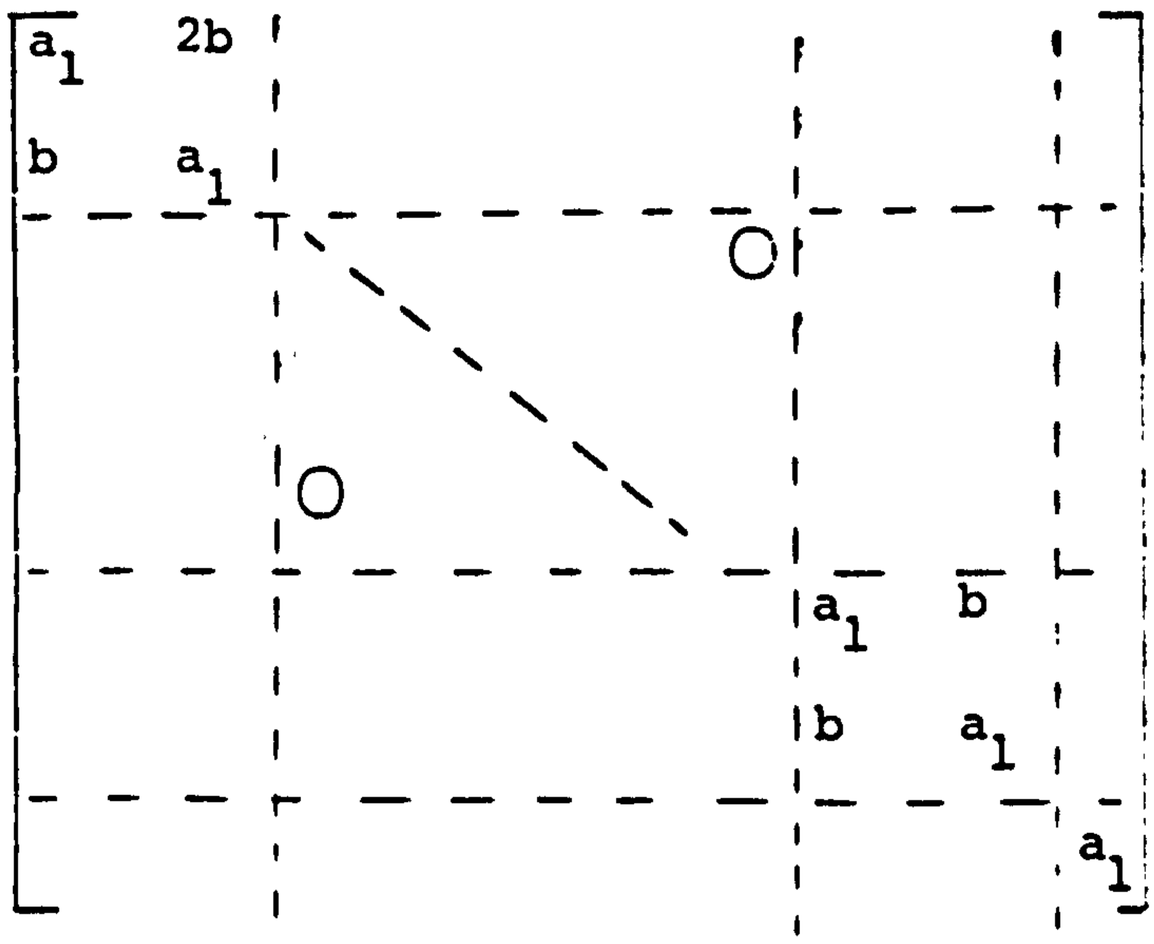
This will lead to A_1 and A_2 as follows,

$A_1 =$



and

$A_2 =$



Thus, from equation (5.6.46) the end point equations are,

$$(a_1+\rho)u_{0,j+1}^{(k+\frac{1}{2})} = g_0-(a_1-\rho)u_{0,j+1}^{(k)}-2bu_{1,j+1}^{(k)}$$

to give,

$$u_{0,j+1}^{(k+\frac{1}{2})} = \frac{1}{(a_1+\rho)}\left[g_0-(a_1-\rho)u_{0,j+1}^{(k)}-2bu_{1,j+1}^{(k)}\right], \qquad (5.6.48)$$

and in submatrix block form the equations at the other end are,

$$\begin{bmatrix} a_1+\rho & b \\ 2b & a_1+\rho \end{bmatrix} \begin{bmatrix} u_{n-1} \\ u_n \end{bmatrix}_{j+1}^{(k+\frac{1}{2})} = \begin{bmatrix} g_{n-1} \\ g_n \end{bmatrix} - \begin{bmatrix} a_1-\rho & 0 \\ 0 & a_1-\rho \end{bmatrix} \begin{bmatrix} u_{n-1} \\ u_n \end{bmatrix}_{j+1}^{(k)} - \begin{bmatrix} bu_{n-2} \\ 0 \end{bmatrix}_{j+1}^{(k)}$$

which in explicit form becomes,

$$\begin{bmatrix} u_{n-1} \\ u_n \end{bmatrix}_{j+1}^{(k+\frac{1}{2})} = \frac{1}{\det} \begin{bmatrix} a_1+\rho & -b \\ -2b & a_1+\rho \end{bmatrix} \left\{ \begin{bmatrix} g_{n-1} \\ g_n \end{bmatrix} - \begin{bmatrix} a_1-\rho & \\ & a_1-\rho \end{bmatrix} \begin{bmatrix} u_{n-1} \\ u_n \end{bmatrix}_{j+1}^{(k)} \right.$$

$$\left. \begin{bmatrix} bu_{n-2} \\ o \end{bmatrix}_{j+1}^{(k)} \right\} ,$$

or as two equations,

$$u_{n-1,j+1}^{(k+\frac{1}{2})} = \frac{1}{\det}[w_{n-1}+qu_{n-1,j+1}^{(k)}+tu_{n,j+1}^{(k)}+pu_{n-2,j+1}^{(k)}] \qquad (5.6.49)$$

and,

$$u_{n,j+1}^{(k+\frac{1}{2})} = \frac{1}{\det}[w_n+2tu_{n-1,j+1}^{(k)}+qu_{n,j+1}^{(k)}+2su_{n-2,j+1}^{(k)}] , \qquad (5.6.50)$$

where,

$$\det = (a_1+\rho)^2-2b^2, \quad w_{n-1} = (a_1+\rho)g_{n-1}-bg_n ,$$

$$w_n = -2bg_{n-1}+(a_1+\rho)g_n, \quad q = -(a_1+\rho)(a_1-\rho), \; t = b(a_1-\rho) ,$$

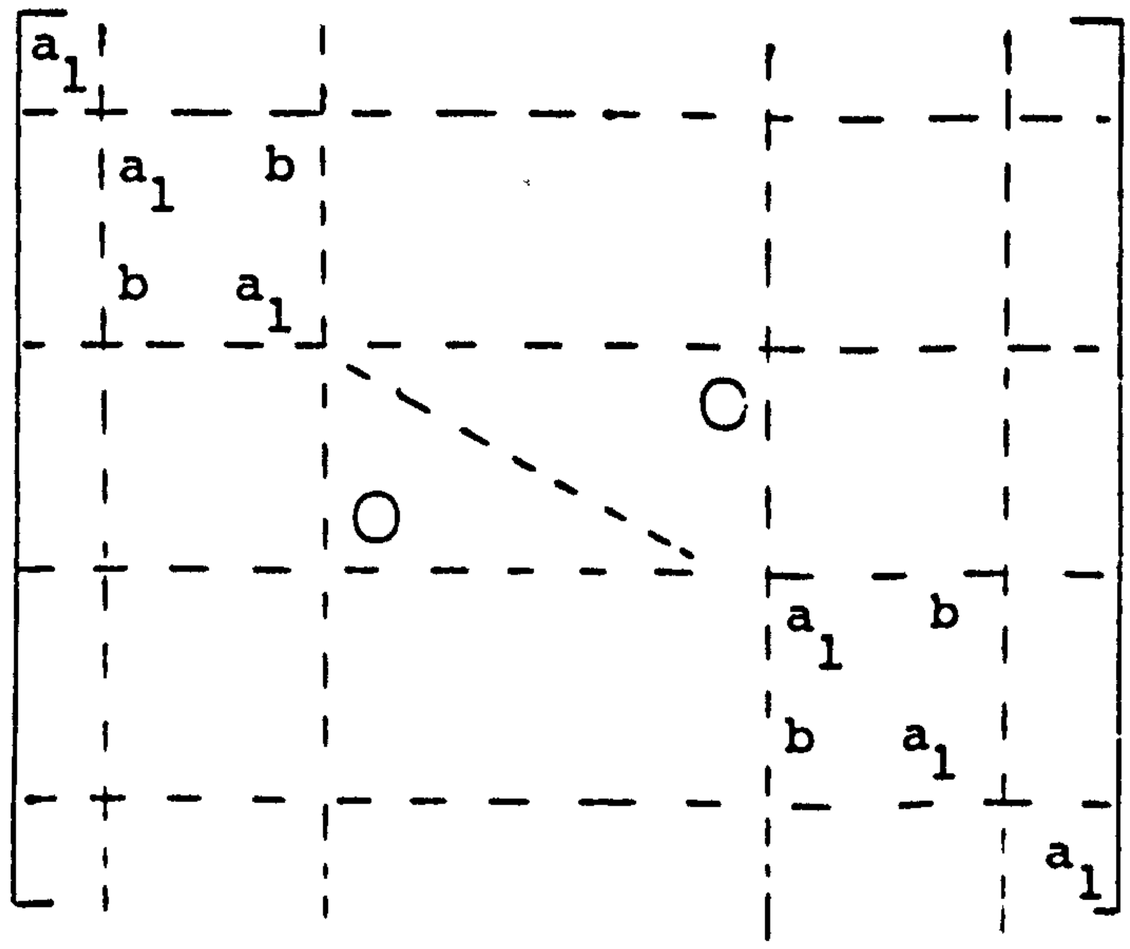
$$p = -(a_1+\rho)b \text{ and } s=b^2 .$$

From equation (5.6.47) the end points equation will have a similar form to (5.6.48), (5.6.49) and (5.6.50) with a revised spacial position.

ii) When n is an odd number:

This will make A_1 and A_2 of an even order with the form,

$A_1 =$



and

$$A_2 = \begin{bmatrix} a_1 & 2b & & & & & \\ b & a_1 & & & & & \\ & & a_1 & b & & & \\ & & b & a_1 & & & \\ & & & & a_1 & b & \\ & & & & 2b & a_1 & \end{bmatrix}$$

where the cases of the end points are treated as in i) above.

c2) The forward and backward difference method

In this case we use the forward difference for the left end point and the backward difference for the right end point.

Therefore equation (3.6.30) will be

$$\frac{u_{1,j} - u_{0,j}}{\Delta x} \approx f_1 , \qquad (5.6.51)$$

and equation (3.6.31) will be,

$$\frac{u_{n,j} - u_{n-1,j}}{\Delta x} \approx f_2 , \qquad (5.6.52)$$

for any time level j. This approximation is of order (h) accurate.

Rewriting equations (5.6.51) and (5.6.52) to get

$$u_{0,j} \approx u_{1,j} - \Delta x f_1 , \qquad (5.6.53)$$

and

$$u_{n,j} \approx u_{n-1,j} + \Delta x f_2 , \qquad (5.6.54)$$

respectively.

Using equation (5.3.1) for i=1, and substituting $u_{0,j}$ in (5.6.53) we get,

$$(5+3r)u_{1,j+1}+(1-3r)u_{2,j+1} = (5-3r)u_{1,j}+(1+3r)u_{2,j}+6r\Delta x f_1 \ . \tag{5.6.55}$$

Similarly for i=n-1 and equation (5.6.54) we get,

$$(1-3r)u_{n-2,j+1}+(5+3r)u_{n-1,j+1} = (1+3r)u_{n-2,j}+(5-3r)u_{n-1,j} + 6r\Delta x f_2 \ . \tag{5.6.56}$$

Hence the system of equations will be,

$$\left.\begin{aligned}
&(5+3r)u_{1,j+1}+(1-3r)u_{2,j+1} = (5-3r)u_{1,j}+(1+3r)u_{2,j}+6r\Delta x f_1 \ , \\
&(1-3r)u_{i-1,j+1}+(4+6r)u_{i,j+1}+(1-3r)u_{i+1,j+1} = (1+3r)u_{i-1,j} \\
&\qquad +(4-6r)u_{i,j}+(1+3r)u_{i+1,j}, \quad i=2,\ldots,n-2 \\
&(5+3r)u_{n-1,j+1}+(1-3r)u_{n-2,j+1} = (5-3r)u_{n-1,j}+(1+3r)u_{n-2,j}+6r\Delta x f_2 \ ,
\end{aligned}\right\} \tag{5.6.57}$$

which is solvable by the SPAGEI method. For the evaluation of $u_{0,j+1}$ and $u_{n,j+1}$ we use equations (5.6.53) and (5.6.54) respectively.

5.7 NUMERICAL RESULTS FOR STATIONARY CASES

In this section we present some numerical results obtained by solving the heat conduction equation (5.1.1) associated with different boundary conditions. They are the Dirichlet, periodic and the derivative (Neumann) boundary conditions.

The purpose of these results is to show the accuracy of the method and the effect of the use of different kinds of parameters. Also, to demonstrate the high stability of the schemes.

The parameters we used here are the Peaceman-Rachford, the Wachpress and the eigenvalues of the constituent matrices of the iteration matrix, where we take the small and the large eigenvalue of the 2×2 group matrix as the lower and upper limits of the sequence of parameters that are chosen. In the case of periodic boundary conditions we include the range of parameters that we obtained in Section 5.5 as another range of eigenvalues, from which we considered (see equation (5.5.21)) $\sqrt{ab}$ and (a+b)/2 as the lower and upper limits of the sequence of parameters.

In these experiments, we have used only the minimum number of parameters. Thus, only one parameter is used for the Peaceman-Rachford method (5.5.12) and the mid eigenvalue or (a+b)/2 in the case of only two eigenvalues available. The Wachpress parameters (equation (5.5.13)) chosen are the minimum number of 2.

Example 1

The heat conduction equation

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \quad , \; 0 \leq x \leq 1, \; t \geq 0 \; . \tag{5.7.1}$$

is considered with the initial conditions,

$$U(x,0) = 4x(1-x),\ 0\leq x\leq 1,\ t=0, \tag{5.7.2}$$

and the Dirichlet boundary conditions,

$$\left.\begin{aligned} U(0,t) &= 0.0, \quad x=0 \\ \text{and} \quad U(1,t) &= 0.0, \quad x=1 \end{aligned}\right\},\ t\geq 0\ . \tag{5.7.3}$$

The numerical solution is tested against the exact solution which is given by,

$$U(x,t) = \frac{32}{\pi^3} \sum_{k=1,3,\ldots,} \frac{1}{k^3} \sin(k\pi x) e^{-k^2\pi^2 t}\ . \tag{5.7.4}$$

The numerical results are shown in Tables (5.7.1,...,5). In Tables (5.7.1,2 and 3) we see the results of an even number of intervals with r having the values 0.1, 1.0 and 2.0, while the Tables (5.7.4 and 5) show the results of an odd number of intervals for r equal to 0.606 and 1.815.

Example 2

In this example a problem with Neumann boundary condition is considered. The heat equation is,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}\ ,\ 0\leq x\leq 1,\quad t\geq 0\ , \tag{5.7.5}$$

with the initial conditions

$$U(x,0) = \sin x + \cos x\ ,\ x\geq 0\ , \tag{5.7.6}$$

and boundary conditions,

$$\left.\begin{aligned} U(0,t) &= e^{-t} \\ U_x(1,t) &= e^{-t}(\cos(1)+\sin(1)) \end{aligned}\right\}\ t\geq 0\ . \tag{5.7.7}$$

and the exact solution is given by,

$$U(x,t) = e^{-t}(\sin x + \cos x)\ ,\ 0\leq x\leq 1,\quad t\geq 0\ , \tag{5.7.8}$$

The numerical results are shown in Tables (5.7,6,7,...,10), compared with the exact solution. Tables (5.7.6,7 and 8) are for an even number of intervals with r having the values 0.1, 1.0 and 2.0 with Tables (5.7.9 and 10) for an odd number of intervals with r equal to 0.605 and 1.815.

Example 3

In this final example the method is tested for periodic boundary conditions. The problem is given by

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + 10(1-x)xt, \quad 0 \leq x \leq 1, \quad t \geq 0 , \tag{5.7.9}$$

with the initial conditions,

$$U(x,0) = x(1-x) , \tag{5.7.10}$$

and boundary conditions,

$$U(0,t) = U(1,t), \quad \frac{\partial U}{\partial x}(x_0,t) = \frac{\partial U}{\partial x}(x_n,t) . \tag{5.7.11}$$

The exact solution of this problem is given by

$$U(x,t) = \left(\frac{1+5t^2}{6}\right) - \frac{5}{8} \sum_{n=1}^{\infty} \frac{\cos 2n\pi x}{(n\pi)^6} \{4n^2\pi^2 t - 1 + e^{-4n^2\pi^2 t}\}$$

$$- \sum_{n=1}^{\infty} \frac{e^{-4n^2\pi^2 t}}{n^2\pi^2} \cos 2n\pi x . \tag{5.7.12}$$

For this example we tested the method with an even number of intervals with values of r 0.1 , 1.0 and 4.0, which are shown in Tables (5.7.11, 12 and 13), and for an odd number of intervals with values of r 0.605 and 2.42. The results are shown in Tables (5.7.14 and 15).

Experiment 1

This experiment is to compare between using 1) the range of parameters $a \leq \rho \leq b$ (where a and b represent the smallest and largest eigenvalues respectively) or 2) the range $(a+b)/2 \leq \rho \leq \sqrt{ab}$, in the sense of the number of iterations required to obtain the same results. The Peaceman Rachford (P.R.) formula (5.5.12) and the Wachspress (W.) formula (5.5.13) are used. Different values of r are considered. The results are shown in Table (5.7.16). The periodic boundary condition problem in Example 3 above is used for this purpose.

Experiment 2

In this experiment we test the use of the optimum parameter ($\rho=3$). This means that we will always have $r=1/3$. The aim is to see the order of accuracy that this parameter will provide. The problems considered here are the Dirichlet, Neumann and Periodic boundary condition problems. The number of iterations is 2 iterations per a time step. All the results are shown in Table (5.7.17).

$\Delta t=0.001$ $r=0.1$
$\Delta x=1/10$ $t=0.1$

Even number of points

kind of prmtr. \ Δx \ x		0	1	2	3	4	5	6	7	8	9	10	No.of iters.
P.R. 1	E.	0.000000	0.000781	0.001475	0.002005	0.002315	0.002370	0.002315	0.002005	0.001475	0.000781	0.000000	328
	P.E.	0.0000	0.6579	0.6534	0.6455	0.6339	0.6174	0.6339	0.6455	0.6534	0.6581	0.0000	
W. 2	E.	0.000000	0.000782	0.001476	0.002006	0.002315	0.002371	0.002315	0.002006	0.001476	0.000781	0.000000	300
	P.E.	0.0000	0.6585	0.6537	0.6458	0.6340	0.6175	0.6341	0.6457	0.6538	0.6585	0.0000	
E.V. 1	E.	0.000000	0.000781	0.001476	0.002007	0.002316	0.002372	0.002316	0.002007	0.001476	0.000782	0.000000	300
	P.E.	0.0000	0.6584	0.6538	0.6459	0.6343	0.6178	0.6343	0.6459	0.6539	0.6586	0.0000	
Exact solution		0.000000	0.118678	0.225727	0.310659	0.365156	0.383883	0.365156	0.310659	0.225727	0.118678	0.000000	

TABLE 5.7.1: Dirichlet Boundary Conditions - Example 1

$\Delta t = 0.01$ $r = 1.0$
$\Delta x = 1/10$ $t = 1.0$

Even number of points

kind or prmtr. \ Δx \ x		0	1	2	3	4	5	6	7	8	9	10	No.of iters.
P.R. 1	E.	0.000000	0.000001	0.000002	0.000003	0.000004	0.000004	0.000004	0.000003	0.000002	0.000001	0.000000	300
	P.E.	0.0000	6.6385	7.4832	7.1373	7.3017	7.2353	7.1968	7.3764	7.0047	7.4885	0.0000	
W. 2	E.	0.000000	0.000001	0.000003	0.000004	0.000004	0.000004	0.000004	0.000004	0.000003	0.000001	0.000000	306
	P.E.	0.0000	8.2351	8.2455	8.2590	8.2558	8.2431	8.2560	8.2577	8.2582	8.2513	0.0000	
E.V. 1	E.	0.000000	0.000001	0.000002	0.000003	0.000003	0.000003	0.000003	0.000003	0.000002	0.000001	0.000000	315
	P.E.	0.0000	6.0559	6.8245	6.5079	6.6579	6.5958	6.5618	6.7268	6.3882	6.8297	0.0000	
Exact solution		0.000000	0.000016	0.000031	0.000043	0.000050	0.000053	0.000050	0.000043	0.000031	0.000016	0.000000	

TABLE 5.7.2: Dirichlet Boundary Conditions - Example 1

$\Delta t=0.02$ $r=2.0$
$\Delta x=1/10$ $t=0.2$

Even number of points

kind of prmtr. \ Δx x		0	1	2	3	4	5	6	7	8	9	10	No.of iters.
P.R. 1	E.	0.000000	0.000753	0.001893	0.002221	0.002763	0.002868	0.002763	0.002221	0.001893	0.000753	0.000000	56
	P.E.	0.0000	1.7043	2.2520	1.9200	2.0316	2.0061	2.0316	1.9200	2.2520	1.7041	0.0000	
W. 2	E.	0.000000	0.000753	0.001893	0.002221	0.002763	0.002868	0.002763	0.002221	0.001893	0.000753	0.000000	66
	P.E.	0.0000	1.7032	2.2515	1.9198	2.0315	2.0063	2.0316	1.9199	2.2517	1.7035	0.0000	
E.V. 1	E.	0.000000	0.000752	0.001891	0.002219	0.002760	0.002865	0.002760	0.002219	0.001891	0.00752	0.000000	66
	P.E.	0.0000	1.7016	2.2496	1.9176	2.0292	2.0037	2.0293	1.9176	2.2496	1.7017	0.0000	
Exact solution		0.000000	0.044196	0.084062	0.115693	0.135990	0.142965	0.135990	0.115693	0.084062	0.044196	0.000000	

TABLE 5.7.3: Dirichlet Boundary Conditions - Example 1

$\Delta t=0.005$ $r=0.605$
$\Delta x=1/11$ $t=0.5$

Odd number of points

kind of prmtr. \ Δx \ x		0	1	2	3	4	5	6	7	8	9	10	11	No.of iters.
P.R. 1	E.	0.000000	0.000061	0.000117	0.000164	0.000197	0.000213	0.000213	0.000197	0.000164	0.000117	0.000061	0.000000	294
	P.E.	0.0000	2.9434	2.9421	2.9364	2.9286	2.9162	2.9162	2.9286	2.9364	2.9421	2.9434	0.0000	
W. 2	E.	0.000000	0.000061	0.000118	0.000164	0.000197	0.000214	0.000214	0.000197	0.000164	0.000118	0.000061	0.000000	285
	P.E.	0.0000	2.9512	2.9489	2.9444	2.9362	2.9242	2.9242	2.9362	2.9444	2.9489	2.9512	0.0000	
E.V. 1	E.	0.000000	0.000061	0.000118	0.000164	0.000197	0.000213	0.000213	0.000197	0.000164	0.000118	0.000061	0.000000	307
	P.E.	0.0000	2.9442	2.9426	2.9369	2.9288	2.9164	2.9164	2.9288	2.9369	2.9426	2.9442	0.0000	
Exact solution		0.000000	0.002081	0.003994	0.005582	0.006718	0.007310	0.007310	0.006718	0.005582	0.003944	0.002081	0.000000	

TABLE 5.7.4: Dirichlet Boundary Conditions - Example 1

$\Delta t = 0.015$ $r = 1.815$
$\Delta x = 1/11$ $t = 0.15$

Odd number of points

Kind of prmt. \ Δx \ x		0	1	2	3	4	5	6	7	8	9	10	11	No.of iters.
P.R. 1	E.	0.000000	0.000626	0.001486	0.001823	0.002302	0.002439	0.002439	0.002302	0.001823	0.001486	0.000626	0.000000	52
	P.E.	0.0000	0.9483	1.1731	1.0291	1.0798	1.0517	1.0517	1.0798	1.0291	1.1731	0.9483	0.0000	
W. 2	E.	0.000000	0.000626	0.001486	0.001822	0.002301	0.002439	0.002439	0.002301	0.001822	0.001486	0.000626	0.000000	60
	P.E.	0.0000	0.9478	1.1726	1.0288	1.0796	1.0515	1.0515	1.0796	1.0288	1.1726	0.9478	0.0000	
E.V. 1	E.	0.000000	0.000625	0.001484	0.001819	0.002298	0.002435	0.002435	0.002298	0.001819	0.001484	0.000625	0.000000	63
	P.E.	0.0000	0.9465	1.1714	1.0273	1.0781	1.0500	1.0500	1.0781	1.0273	1.1714	0.9465	0.0000	
Exact solution		0.000000	0.066027	0.126701	0.177102	0.213144	0.231905	0.231905	0.213144	0.177102	0.126701	0.066027	0.000000	

TABLE 5.7.5: Dirichlet Boundary Conditions - Example 1

$\Delta t = 0.001$ $r = 0.1$
$\Delta x = 1/10$ $t = 0.1$

Even number of points

kind of prmtr. \ Δx x		0	1	2	3	4	5	6	7	8	9	10	No.of iters.
P.R. 1	E.	0.000074	0.000032	0.000003	0.000032	0.000053	0.000068	0.000076	0.000078	0.000075	0.000067	0.000056	300
	P.E.	0.0082	0.0032	0.0003	0.0028	0.0045	0.0055	0.0060	0.0061	0.0059	0.0053	0.0045	
W. 2	E.	0.000074	0.000032	0.000004	0.000032	0.000054	0.000068	0.000076	0.000078	0.000076	0.000068	0.000057	400
	P.E.	0.0081	0.0032	0.0004	0.0029	0.0045	0.0055	0.0060	0.0061	0.0059	0.0054	0.0045	
E.V. 1	E.	0.000073	0.000031	0.000004	0.000033	0.000054	0.000068	0.000076	0.000079	0.000076	0.000068	0.000056	300
	P.E.	0.0081	0.0032	0.0004	0.0029	0.0046	0.0056	0.0061	0.0062	0.0059	0.0054	0.0045	
Exact solution		0.904837	0.990650	1.066564	1.131822	1.185771	1.227872	1.257704	1.274970	1.279497	1.271239	1.250280	

TABLE 5.7.6: Neumann Boundary Conditions - Example 2

$\Delta t = 0.01$ $r = 1.0$
$\Delta x = 1/10$ $t = 1.0$

Even number of points

kind of prmtr. \ Δx x		0	1	2	3	4	5	6	7	8	9	10	No.of iters.
P.R. 1	E.	0.003978	0.003803	0.003648	0.003513	0.003402	0.003315	0.003253	0.003216	0.003206	0.003222	0.003265	400
	P.E.	1.0813	0.9442	0.8412	0.7635	0.7057	0.6640	0.6361	0.6205	0.6163	0.6235	0.6423	
W. 2	E.	0.003981	0.003806	0.003650	0.003516	0.003404	0.003317	0.003255	0.003219	0.003209	0.003225	0.003267	441
	P.E.	1.0821	0.9450	0.8418	0.7641	0.7062	0.6645	0.6366	0.6210	0.6169	0.6240	0.6428	
E.V. 1	E.	0.003985	0.003810	0.003655	0.003520	0.003409	0.003322	0.003260	0.003223	0.003213	0.003229	0.003272	400
	P.E.	1.0832	0.9460	0.8428	0.7650	0.7071	0.6654	0.6374	0.6218	0.6177	0.6248	0.6437	
Exact solution		0.367879	0.402768	0.433633	0.460164	0.482098	0.499215	0.511344	0.518364	0.520205	0.516847	0.508326	

TABLE 5.7.7: Neumann Boundary Conditions - Example 2

$\Delta t=0.02$ $r=2.0$
$\Delta x=1/10$ $t=2.0$

Even number of points

kind of prmtr. \ Δx x		0	1	2	3	4	5	6	7	8	9	10	No.of iters.
P.R. 1	E.	0.010675	0.010546	0.010432	0.010334	0.010253	0.010189	0.010144	0.010118	0.010111	0.010123	0.010154	508
	P.E.	7.8875	7.1176	6.5395	6.1045	5.7809	5.5481	5.3924	5.2056	5.2832	5.3239	5.4299	
W. 2	E.	0.010677	0.010549	0.010435	0.010337	0.010256	0.01093	0.010147	0.010120	0.010113	0.010125	0.010156	746
	P.E.	7.8894	7.1194	6.5414	6.1065	5.7829	5.5500	5.3942	5.3071	5.2845	5.3251	5.4310	
E.V. 1	E.	0.010722	0.010594	0.010480	0.010382	0.010301	0.010237	0.010192	0.010165	0.010158	0.010170	0.010202	609
	P.E.	7.9225	7.1496	6.5694	6.1327	5.8079	5.5741	5.4179	5.3307	5.3081	5.3489	5.4554	
Exact solution		0.135335	0.148170	0.159525	0.169285	0.177354	0.183651	0.188113	0.190696	0.191373	0.190138	0.187003	

TABLE 5.7.8: Neumann Boundary Conditions - Example 2

Δt=0.005 r=0.605
Δx=1/11 t=0.5

Odd number of points

Kind of prmt. \ Δx x		0	1	2	3	4	5	6	7	8	9	10	11	No.of iters.
P.R. 1	E.	0.001387	0.001256	0.001137	0.001032	0.000943	0.000870	0.000814	0.000775	0.000754	0.000752	0.000767	0.000799	300
	P.E.	0.2287	0.1905	0.1610	0.1381	0.1205	0.1073	0.0977	0.0914	0.0881	0.0877	0.0900	0.0954	
W. 2	E.	0.001384	0.001252	0.001134	0.001029	0.000940	0.000867	0.000811	0.000772	0.000751	0.000748	0.000763	0.000796	309
	P.E.	0.2282	0.1900	0.1605	0.1377	0.1201	0.1069	0.0973	0.0910	0.0877	0.0872	0.0896	0.0949	
E.V. 1	E.	0.001388	0.001257	0.001138	0.001033	0.000944	0.000871	0.000815	0.000776	0.000755	0.000752	0.000767	0.000800	300
	P.E.	0.2289	0.1906	0.1611	0.1382	0.1206	0.1074	0.0978	0.0915	0.0882	0.0878	0.0902	0.0955	
Exact solution		0.606531	0.659089	0.706205	0.747488	0.782597	0.811244	0.833190	0.848255	0.856315	0.857303	0.851210	0.838088	

TABLE 5.7.9: Neumann Boundary Conditions - Example 2

$\Delta t = 0.015$ $r = 1.815$
$\Delta x = 1/11$ $t = 1.5$

Odd number of points

Kind of prmt. \ Δx \ x		0	1	2	3	4	5	6	7	8	9	10	11	No.of iters.
P.R. 1	E.	0.007369	0.007224	0.007093	0.006978	0.006880	0.006800	0.006738	0.006696	0.006674	0.006671	0.006688	0.006723	490
	P.E.	3.3025	2.9793	2.7302	2.5377	2.3897	2.2785	2.1984	2.1459	2.1186	2.1152	2.1357	2.1808	
W. 2	E.	0.007359	0.007214	0.007084	0.006969	0.006872	0.006792	0.006731	0.006688	0.006665	0.006662	0.006678	0.006714	700
	P.E.	3.2982	2.9753	2.7266	2.5345	2.3869	2.2759	2.1959	2.1433	2.1158	2.1123	2.1327	2.1778	
E.V. 1	E.	0.007408	0.007263	0.007132	0.007018	0.006920	0.006841	0.006779	0.006737	0.006714	0.006711	0.006727	0.006763	569
	P.E.	3.3199	2.9954	2.7453	2.5521	2.4037	2.2921	2.117	2.1589	2.1313	2.1278	2.1483	2.1936	
Exact solution		0.223130	0.242465	0.259798	0.274985	0.287901	0.298440	0.306514	0.312056	0.315021	0.315384	0.313143	0.308315	

TABLE 5.7.10: Neumann Boundary Conditions - Example 2

$\Delta t = 0.001$ $r = 0.1$
$\Delta x = 1/10$ $t = 0.1$

Even number of points

kind of prmtr. \ Δx x		0	1	2	3	4	5	6	7	8	9	10	No.of iters.
P.R. 1	E.	0.001748	0.001635	0.001616	0.001653	0.001699	0.001720	0.001699	0.001653	0.001616	0.001635	0.001748	304
	P.E.	1.0226	0.9516	0.9292	0.9375	0.9542	0.9619	0.9542	0.9375	0.9292	0.9516	1.0226	
W. 2	E.	0.001747	0.001634	0.001616	0.001653	0.001700	0.001720	0.001700	0.001653	0.001616	0.001634	0.001757	331
	P.E.	1.0220	0.9512	0.9289	0.9375	0.9544	0.9622	0.9544	0.9374	0.9289	0.9511	1.0220	
E.V. 1	E.	0.001748	0.001635	0.001616	0.001653	0.001700	0.001720	0.001700	0.001653	0.001616	0.001635	0.001748	296
	P.E.	1.0226	0.9517	0.9293	0.9377	0.9549	0.9622	0.9545	0.9377	0.9292	0.9518	1.0226	
Exact solution		0.170916	0.171834	0.173940	0.176297	0.178091	0.178758	0.178091	0.176297	0.173940	0.171834	0.170916	

TABLE 5.7.11: Periodic Boundary Conditions - Example 3

$\Delta t = 0.01$ $r = 1.0$
$\Delta x = 1/10$ $t = 1.0$

Even number of points

kind of prmtr. \ Δx x		0	1	2	3	4	5	6	7	8	9	10	No.of iters.
P.R. 1	E.	0.004630	0.003094	0.001861	0.000960	0.000411	0.000226	0.000411	0.000960	0.001861	0.003094	0.004630	397
	P.E.	0.4759	0.3159	0.1873	0.0952	0.0403	0.0221	0.0403	0.0952	0.1873	0.3159	0.4759	
W. 2	E.	0.004626	0.003090	0.001858	0.000956	0.000407	0.000223	0.000407	0.000957	0.001858	0.003091	0.004626	404
	P.E.	0.4755	0.3155	0.1869	0.0948	0.0399	0.0218	0.0399	0.0948	0.1870	0.3156	0.4755	
E.V. 1	E.	0.004638	0.003103	0.001870	0.000968	0.000419	0.000235	0.000419	0.000968	0.001870	0.003103	0.004638	401
	P.E.	0.4767	0.3168	0.1882	0.0960	0.0411	0.0229	0.0411	0.0960	0.1882	0.3168	0.4767	
Exact solution		0.972884	0.979501	0.993748	1.008764	1.019700	1.023665	1.019700	1.008764	0.993748	0.979501	0.972884	

TABLE 5.7.12: Periodic Boundary Conditions - Example 3

$\Delta t = 0.04$ $r = 4.0$
$\Delta x = 1/10$ $t = 4.0$

Even number of points

kind of prmtr. \ Δx x		0	1	2	3	4	5	6	7	8	9	10	No.of iters.
P.R. 1	E.	0.014619	0.008456	0.003490	0.000158	0.002381	0.003132	0.002381	0.000158	0.003490	0.008456	0.014619	854
	P.E.	0.1092	0.0630	0.0259	0.0012	0.0175	0.0230	0.0175	0.0012	0.0259	0.0630	0.1092	
W. 2	E.	0.014577	0.008414	0.003447	0.000200	0.002423	0.003172	0.002422	0.000198	0.003450	0.008415	0.014577	3383
	P.E.	0.1089	0.0627	0.0256	0.0015	0.0178	0.0233	0.0178	0.0015	0.0256	0.0627	0.1089	
E.V. 1	E.	0.014804	0.008642	0.003676	0.000028	0.002195	0.002946	0.002195	0.000028	0.003675	0.008642	0.014804	1218
	P.E.	0.1106	0.0644	0.0273	0.0002	0.0162	0.0217	0.0162	0.0002	0.0273	0.0644	0.1106	
Exact solution		13.389551	13.41641	13.47441	13.53568	13.58036	13.59658	13.58036	13.53568	13.47441	13.41641	13.389551	

TABLE 5.7.13: Periodic Boundary Conditions - Example 3

Δt=0.005 r=0.605
Δx=1/11 t=0.5

Odd number of points

Kind of prmt. \ Δx x		0	1	2	3	4	5	6	7	8	9	10	11	No.of iters.
P.R. 1	E.	0.002226	0.001644	0.001164	0.000795	0.000545	0.000418	0.000418	0.000545	0.000795	0.001164	0.001644	0.002226	301
	P.E.	0.6154	0.4511	0.3140	0.2106	0.1422	0.1083	0.1083	0.1422	0.2106	0.3140	0.4511	0.6154	
W. 2	E.	0.002226	0.001644	0.001164	0.000795	0.000545	0.000418	0.000418	0.000545	0.000795	0.001164	0.001644	0.002226	305
	P.E.	0.6154	0.4511	0.3140	0.2106	0.1422	0.1083	0.1083	0.1422	0.2106	0.3140	0.4511	0.6154	
E.V. 1	E.	0.002226	0.001644	0.001164	0.000795	0.000545	0.000419	0.000419	0.000545	0.000795	0.001164	0.001644	0.002226	301
	P.E.	0.6154	0.4511	0.3141	0.2106	0.1423	0.1084	0.1084	0.1423	0.2106	0.3141	0.4511	0.6154	
Exact solution		0.361773	0.364508	0.370594	0.377402	0.382996	0.386108	0.386108	0.382996	0.377402	0.370594	0.364508	0.361773	

TABLE 5.7.14: Periodic Boundary Conditions - Example 3

$\Delta t = 0.02$ $r = 2.42$
$\Delta x = 1/11$ $t = 2.0$

Odd number of points

Kind of prmt. \ Δx x		0	1	2	3	4	5	6	7	8	9	10	11	No.of iters.
P.R. 1	E.	0.000741	0.001600	0.003549	0.005055	0.006081	0.006600	0.006600	0.006081	0.005055	0.003549	0.001600	0.000741	598
	P.E.	0.0215	0.0463	0.1019	0.1440	0.1721	0.1861	0.1861	0.1721	0.1440	0.1019	0.0463	0.0215	
W. 2	E.	0.000707	0.001633	0.003582	0.005087	0.006112	0.00663	0.006631	0.006112	0.005087	0.003582	0.001633	0.000707	516
	P.E.	0.0205	0.0473	0.1029	0.1449	0.1730	0.1870	0.1870	0.1730	0.1449	0.1029	0.0473	0.0205	
E.V. 1	E.	0.00792	0.001549	0.003498	0.005005	0.006030	0.006549	0.006549	0.006030	0.005005	0.003498	0.001549	0.000792	766
	P.E.	0.0230	0.0448	0.1005	0.1426	0.1707	0.1847	0.1847	0.1707	0.1426	0.1005	0.0448	0.0230	
Exact solution		3.445106	3.456379	3.481590	3.509912	3.533265	3.546281	3.546281	3.533265	3.509912	3.481590	3.456379	3.445106	

TABLE 5.7.15: Periodic Boundary Conditions - Example 3

NO. OF ITERATIONS				
range of parameters	$a \leq \rho \leq b$		$\frac{a+b}{2} \leq \rho \leq \sqrt{ab}$	
r	P.R.	W.	P.R.	W.
0.1	304	331	306	304
0.5	300	300	239	215
1.0	397	404	401	401
2.0	573	765	616	624
4.0	854	3383	1002	994
0.121	288	319	320	306
0.605	301	305	251	221
1.21	402	317	402	402
2.42	598	516	671	675
4.84	900	929	1111	1067

TABLE 5.7.16: Experiment 1

$\Delta x = 1/10$ $r = 0.333$
$\Delta t = 0.00333$ $t = 0.333$

Boundary conditions \ Δx x		0	1	2	3	4	.5	6	7	8	9	10
Dirichlet	E.	0.000000	0.000282	0.000535	0.000735	0.000860	0.000898	0.000860	0.000735	0.000535	0.000282	0.000000
	P.E.	0.0000	2.3732	2.3692	2.3622	2.3514	2.3354	2.3514	2.3622	2.3692	2.3732	0.0000
	Exact sol.	0.000000	0.011881	0.022598	0.031101	0.036557	0.038432	0.036557	0.031101	0.022598	0.011881	0.000000
Neumann	E.	0.000666	0.000552	0.000451	0.000365	0.000293	0.000237	0.000198	0.000176	0.000170	0.000181	0.000209
	P.E.	0.0929	0.0704	0.0534	0.0407	0.0312	0.0244	0.0199	0.0174	0.0168	0.0180	0.0211
	Exact sol.	0.716770	0.784747	0.844883	0.896577	0.939313	0.972663	0.996295	1.009972	1.013558	1.007017	0.990414
Periodic	E.	0.002617	0.002110	0.001708	0.001417	0.001240	0.001181	0.001240	0.001417	0.001708	0.002110	0.002617
	P.E.	1.0450	0.8354	0.6643	0.5411	0.4675	0.4432	0.4675	0.5411	0.6643	0.8354	1.0450
	Exact sol.	0.250485	0.252600	0.257120	0.261854	0.265286	0.266527	0.265286	0.261854	0.257120	0.252600	0.250485

TABLE 5.7.17: Experiment 2 - Optimum no. of iterations (200)

5.8 MORE THAN ONE PARAMETER

Although it was stated in Section 5.5 that in the SPAGEI method, the matrices A_1 and A_2 are not commutative and therefore there is no theoretical justification for using more than one acceleration parameter, we actually have used more than one parameter and found that in many cases it reduces the number of iterations significantly.

From equation (2.8.3) and by the definition of the average rate of convergence we have,

$$R(\prod_{i=1}^{m} T_{\rho_i}) = -\frac{\ell n}{m} \left\| \prod_{i=1}^{m} T_{\rho_i} \right\|$$

$$= -\frac{1}{m} \log \rho(\prod_{i=1}^{m} T_{\rho_i}) \ , \qquad (5.8.1)$$

where m is the number of parameters and T is the iteration matrix.

It is proved in [Young, D.M., 1971] and [Varga, R.S., 1962] that the number of iterations increases as $|\log \Delta x|$ or $|\log h|$.

In Birkhoff et al, 1962, it is shown that the average rate of convergence is asymptotically proportional to $h^{1/m}$ for small h. Consequently, one would expect that, asymptotically for small h, log(N) would be a linear function of $\log(h^{-1})$ with slope 1/(m-1) for the Wachspress parameters, where (N) is the number of iterations. [Birkhoff, et al, 1962]. In our experimental results, we find that the number of iterations are reduced but they do not support the theoretical predictions.

Experiment 1

In this experiment, in solving the periodic boundary condition problem of the previous section (Example 2), we use the Wachspress

parameters (5.5.13) as the acceleration parameters. This is done for various values of h or (Δx), but Δt is kept constant in each case. For Δt=0.00005 we used $h=170^{-1},180^{-1},\ldots,240^{-1}$, for Δt=0.0001 we used $h=100^{-1},110^{-1},\ldots,180^{-1}$ and for Δt=0.0005 we used $h=30^{-1},40^{-1},\ldots,90^{-1}$. The results are shown in Tables (5.8.1), (5.8.2) and (5.8.3). To illustrate the results we display, with logarithmic scales, the number of iterations versus h^{-1}. In Figures (5.8.1), (5.8.2) and (5.8.3) we show the regression plots to compare with the theory proposed by Birkhoff et al [1962] and in Figures (5.8.4), (5.8.5) and (5.8.6) the plots show the real behaviour of the development of the number of iterations.

$\Delta t=0.00005$
$t=0.005$

No. of Intervals	r	No. of Parameters		
		2	3	4
170	1.445	320	315	313
180	1.62	347	323	322
190	1.805	392	357	355
200	2.0	436	404	402
210	2.205	496	422	413
220	2.42	525	431	420
230	2.645	581	453	432
240	2.88	693	495	443
250	3.125	775	533	474

TABLE 5.8.1: The number of iterations

$\Delta t=0.0001$
$t=0.01$

No. of intervals	r	No. of Parameters		
		2	3	4
100	1.0	305	305	305
110	1.21	313	309	309
120	1.44	335	320	316
130	1.69	386	357	355
140	1.96	449	417	411
150	2.25	519	429	421
160	2.50	581	465	435
170	2.89	727	529	470
180	3.24	909	576	527

TABLE 5.8.2: The number of iterations

$\Delta t=0.0005$
$t=0.05$

No. of intervals	r	No. of Parameters		
		2	3	4
30	0.45	208	208	208
40	0.8	262	260	260
50	1.25	332	320	319
60	1.8	437	396	391
70	2.45	582	482	448
80	3.2	920	602	544
90	4.05	1980	718	662

TABLE 5.8.3: The number of iterations

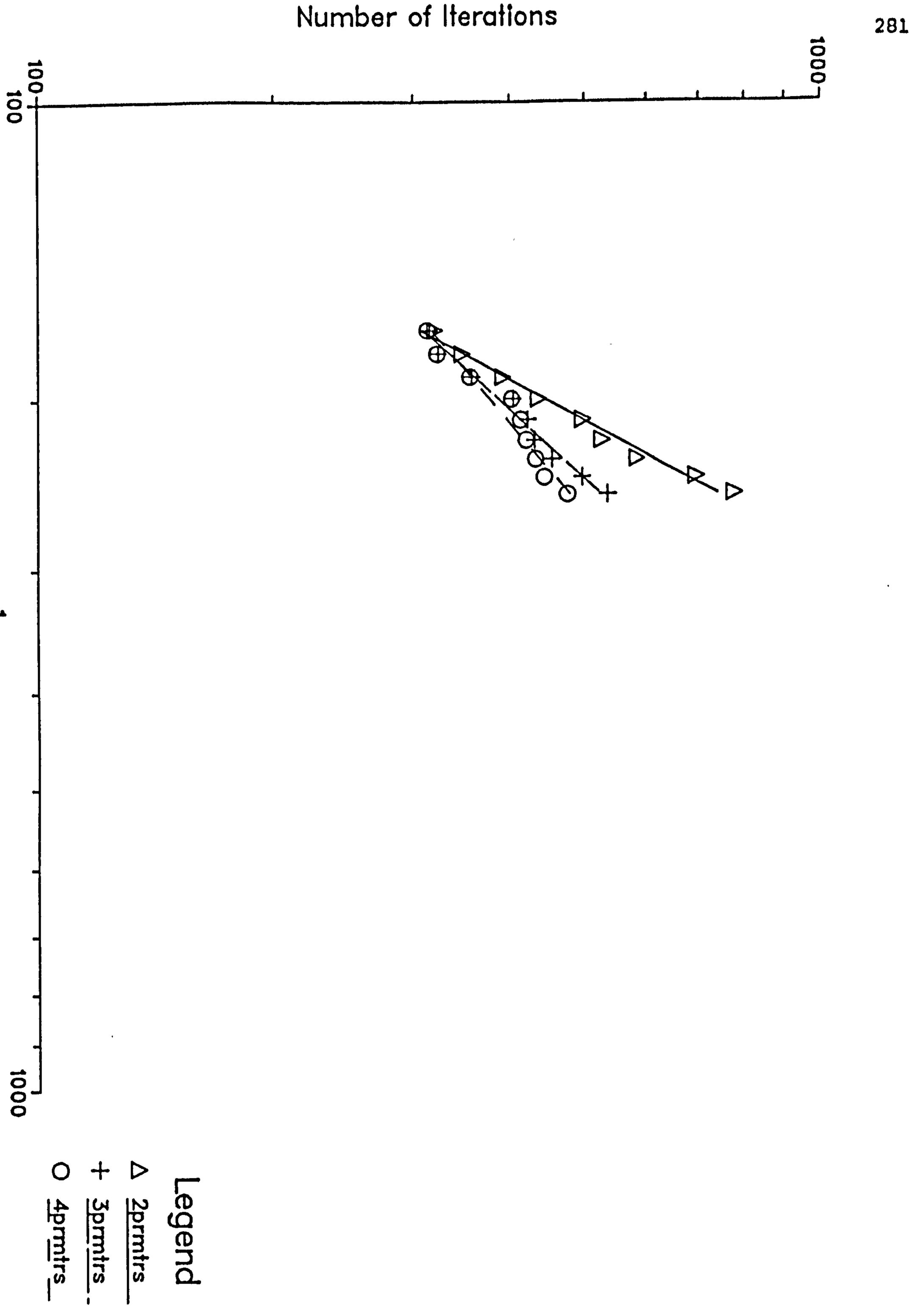


FIGURE 5.8.1: A regression plot of the number of iterations
$\Delta t=0.00005$

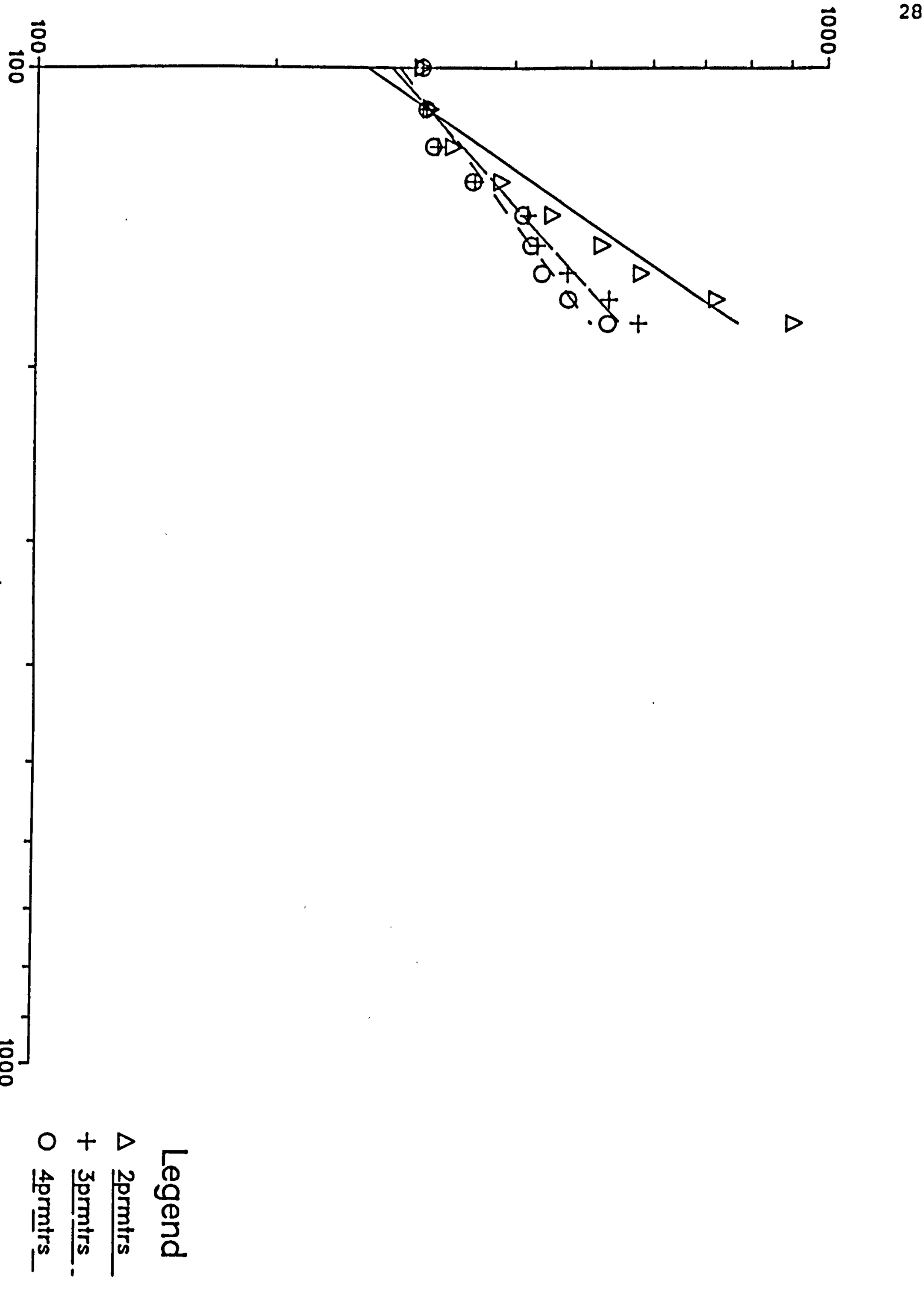


FIGURE 5.8.2: A regression plot of the number of iterations
$\Delta t=0.0001$

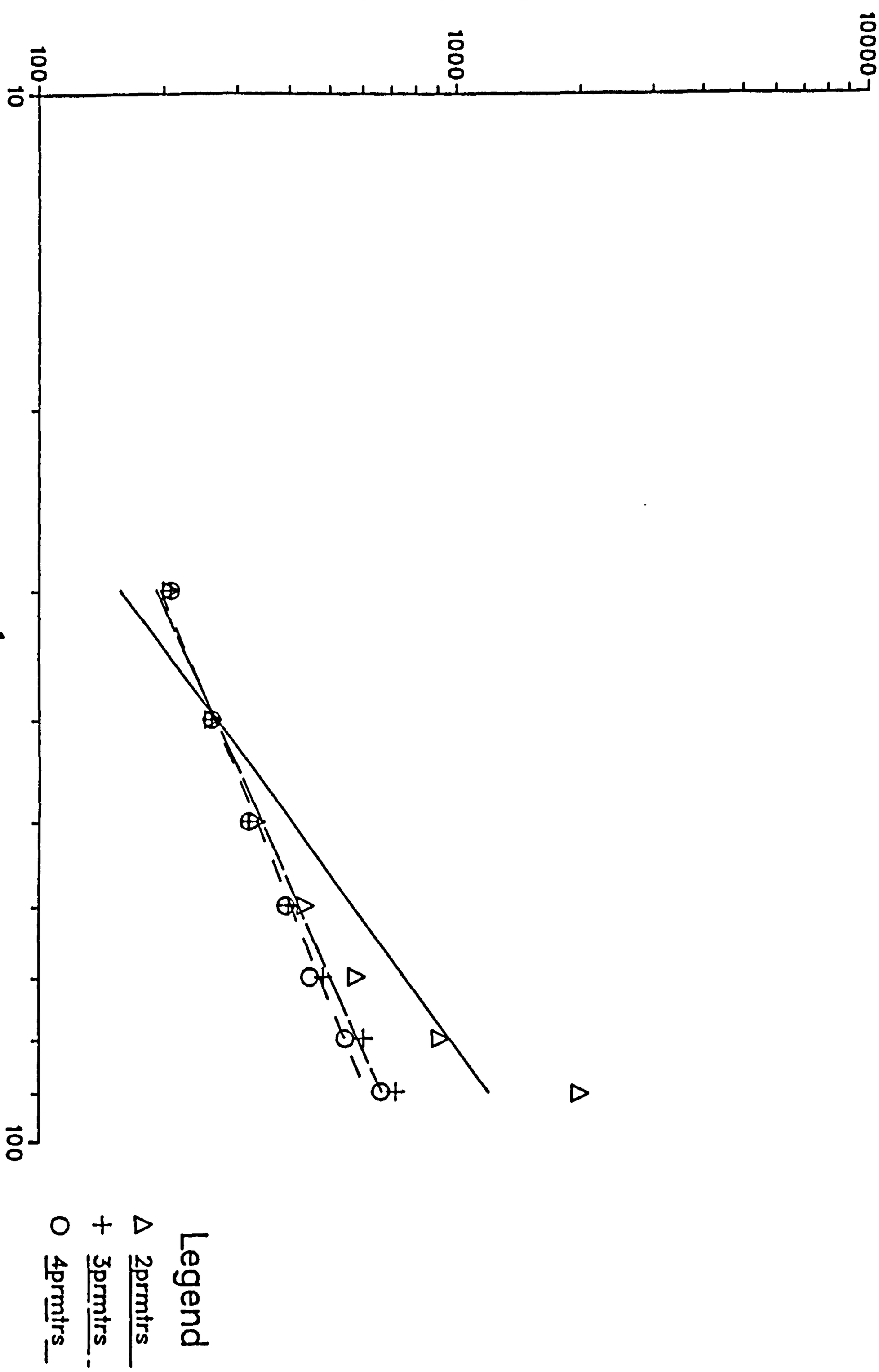


FIGURE 5.8.3: A regression plot of the number of iterations
$\Delta t=0.0005$

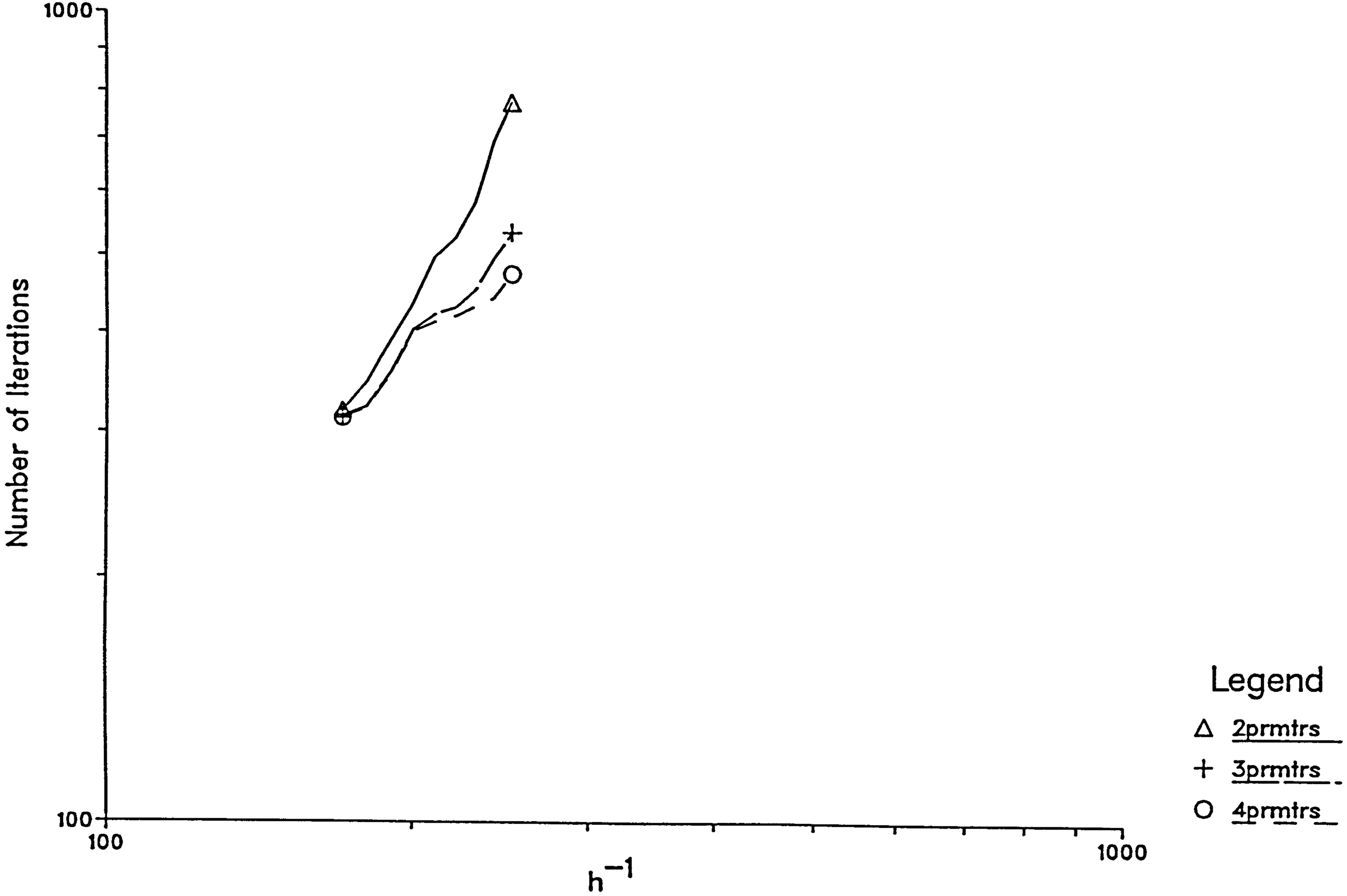


FIGURE 5.8.4: A plot of the number of iterations
Δt=0.00005

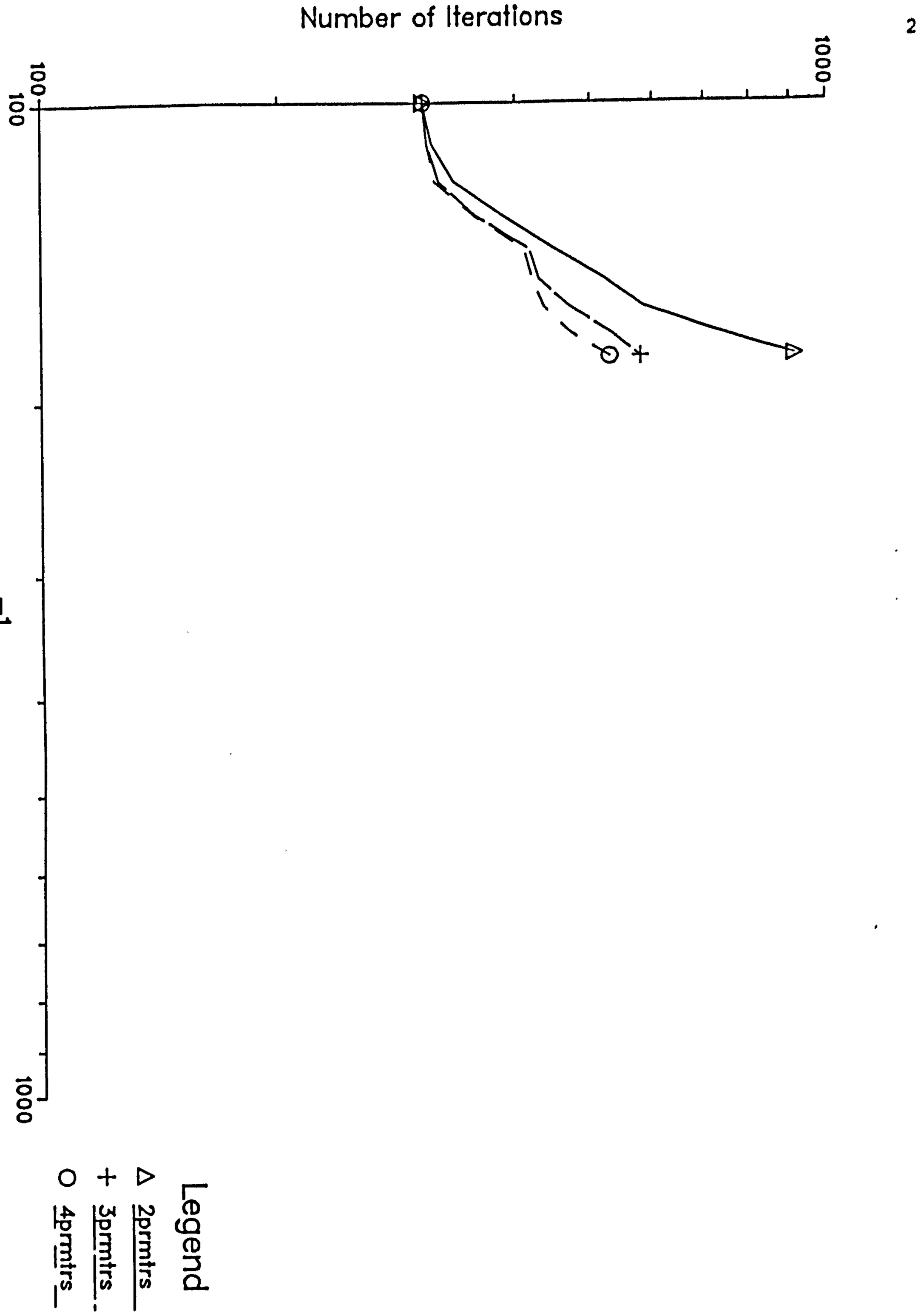


FIGURE 5.8.5: A plot of the number of iterations
$\Delta t=0.0001$

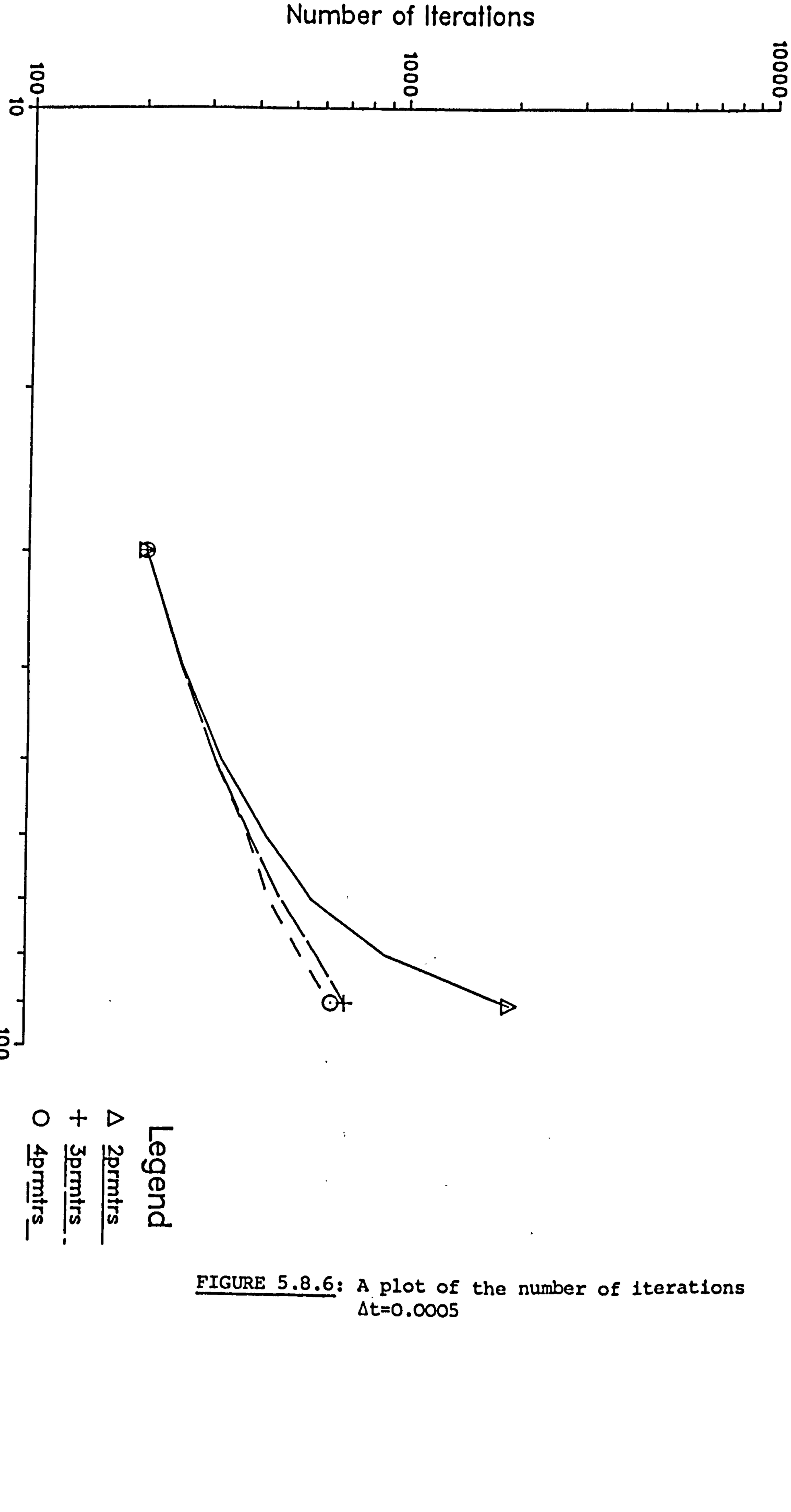


FIGURE 5.8.6: A plot of the number of iterations
$\Delta t=0.0005$

5.9 THE SUCCESSIVE OVER-RELAXATION (SOR) METHOD

We have mentioned briefly the SOR method in Section 3.8. Here we apply this method to our cubic splines equation,

$$(1-3r)(u_{i-1,j+1}+u_{i+1,j+1})+(4+6r)u_{i,j+1} = (1+3r)(u_{i-1,j}+u_{i+1,j}) +(4-6r)u_{i,j} \quad , \; i=1,\ldots,n-1. \tag{5.9.1}$$

By considering the totality of equation (5.9.1) written in matrix form as,

$$Au_{j+1} = Bu_j \quad , \tag{5.9.2}$$

or

$$\begin{bmatrix} d & a & & & & \\ a & d & a & & O & \\ & \ddots & \ddots & \ddots & & \\ & & & a & d & a \\ & & O & & a & d \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix} \quad ,$$

where,

$$d = 4+6r, \; a = 1-3r,$$

and $$b_i = (1+3r)(u_{i-1,j}+u_{i+1,j})+(4-6r)u_{i,j} \quad , \; i=1,\ldots,n-1.$$

The SOR iterative method is defined by,

$$u_{i,j+1}^{(k+1)} = u_{i,j+1}^{(k)} + \frac{\omega}{d}\{b_i - du_{i,j+1}^{(k)} - au_{i+1,j+1}^{(k)} - au_{i-1,j+1}^{(k+1)}\}, \tag{5.9.3}$$

where ω, as previously defined, is the relaxation parameter.

To determine the optimum value of ω to attain the maximum rate of convergence we consider the linear system (5.9.2). Let,

$$A = D + S + R \quad ,$$

where,

$$D = \begin{bmatrix} 4+6r & & O \\ & \ddots & \\ O & & 4+6r \end{bmatrix}, \quad S = \begin{bmatrix} 0 & 1-3r & & O \\ & \ddots & \ddots & \\ O & & & 1-3r \\ & & & 0 \end{bmatrix}$$

and

$$R = \begin{bmatrix} 0 & & & O \\ 1-3r & \ddots & & \\ & \ddots & & \\ O & & 1-3r & 0 \end{bmatrix}$$

From Section 3.8 we recall the Jacobi iteration matrix,

$$J = -D^{-1}(S+R) ,$$

which, for equation (5.9.1), is written as,

$$J = \frac{-1}{4+6r} \begin{bmatrix} 0 & 1-3r & & \\ 1-3r & \ddots & \ddots & \\ & \ddots & \ddots & 1-3r \\ O & & 1-3r & 0 \end{bmatrix} .$$

For a tridiagonal matrix of the special form,

$$\begin{bmatrix} a & b & & & & \\ c & a & b & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & c & a & b \\ & & & & c & a \end{bmatrix}$$

we use the following formula to evaluate the eigenvalues of the matrix,

$$\lambda_s = a + 2\sqrt{bc}\,\cos\frac{s\pi}{N}\,, \quad s=1,2,\ldots,N-1\,,$$

where (N-1) is the order of the matrix, [Smith, G.D., 1978].

Therefore,

$$\lambda_1 = 0 + 2\sqrt{\left(\frac{(1-3r)}{(4+6r)}\right)^2}\,\cos\left(\frac{\pi}{N}\right)$$

$$\lambda_{N-1} = 0 + 2\sqrt{\left(\frac{(1-3r)}{(4+6r)}\right)^2}\,\cos\left(\frac{(N-1)\pi}{N}\right)$$

For sufficiently large N we get,

$$\lambda_1 = 2\left(\frac{1-3r}{4+6r}\right)\cos(0) = \frac{1-3r}{2+3r}$$

and $$\lambda_{N-1} = 2\left(\frac{1-3r}{4+6r}\right)\cos(\pi) = -\frac{1-3r}{2+3r}\,.$$

Thus the spectral radius $\rho(J)$ of the matrix J is $|\lambda_1|$ or $|\lambda_{N-1}|$.

The relaxation parameter ω is then found by,

$$\omega = \frac{2}{1+\sqrt{1-\rho^2(J)}}\,. \qquad (5.9.3)$$

(For details see [Young, D.M., 1954].

Substituting $\rho(J)$ in (5.9.3) gives,

$$\omega = \frac{2}{1+\sqrt{1-\left(\frac{1-3r}{2+3r}\right)^2}}\,, \qquad (5.9.4)$$

$$= \frac{2}{1+\sqrt{\frac{(3+18r)}{(2+3r)^2}}} \quad ,$$

$$= \frac{4+6r}{2+3r+\sqrt{3+18r}} \quad . \qquad (5.9.5)$$

This shows that ω is a function of r and it is interesting to find that when r=1/3 the method becomes the Gauss-Seidel's, where ω=1. For any other positive value of r we have,

$$1 < \omega < 2 \; . \qquad (5.9.6)$$

Experiment 1

This experiment is to compute the number of iterations that are required by the SOR and the SPAGEI method to achieve convergence to the same solution. We have chosen for this purpose the Dirichlet boundary condition problem of Example 1, Section 4.12. We have solved the problem for several number of subintervals of the domain and for several values of Δt. The results are obtained after 10 time steps from time =0. In the SPAGEI method we used the Peaceman-Rachford single parameter. The results are presented in Tables (5.9.2) and are illustrated in Figure (5.9.1). Table (5.9.1) shows the values of r that are used in this experiment.

Δt \ n	10	20	40	80
0.001	0.1	0.4	1.8	6.4
0.0025	0.25	1	4	16
0.005	0.5	2	8	32
0.01	1.0	4	16	64
0.02	2.0	8	32	128
0.04	4.0	16	64	256

TABLE 5.9.1: The value of r used in Experiment 1

Δt \ n	10	20	40	80
0.001	60 40	40 30	70 50	160 80
0.0025	40 30	70 40	120 70	217 120
0.005	50 30	99 55	175 91	314 164
0.01	72 41	135 72	245 124	582 222
0.02	95 56	174 101	323 184	593 340
0.04	103 71	199 132	379 257	720 480

TABLE 5.9.2: A comparison of the number of iterations between the SOR method (1st line) and the SPAGEI method (2nd line)

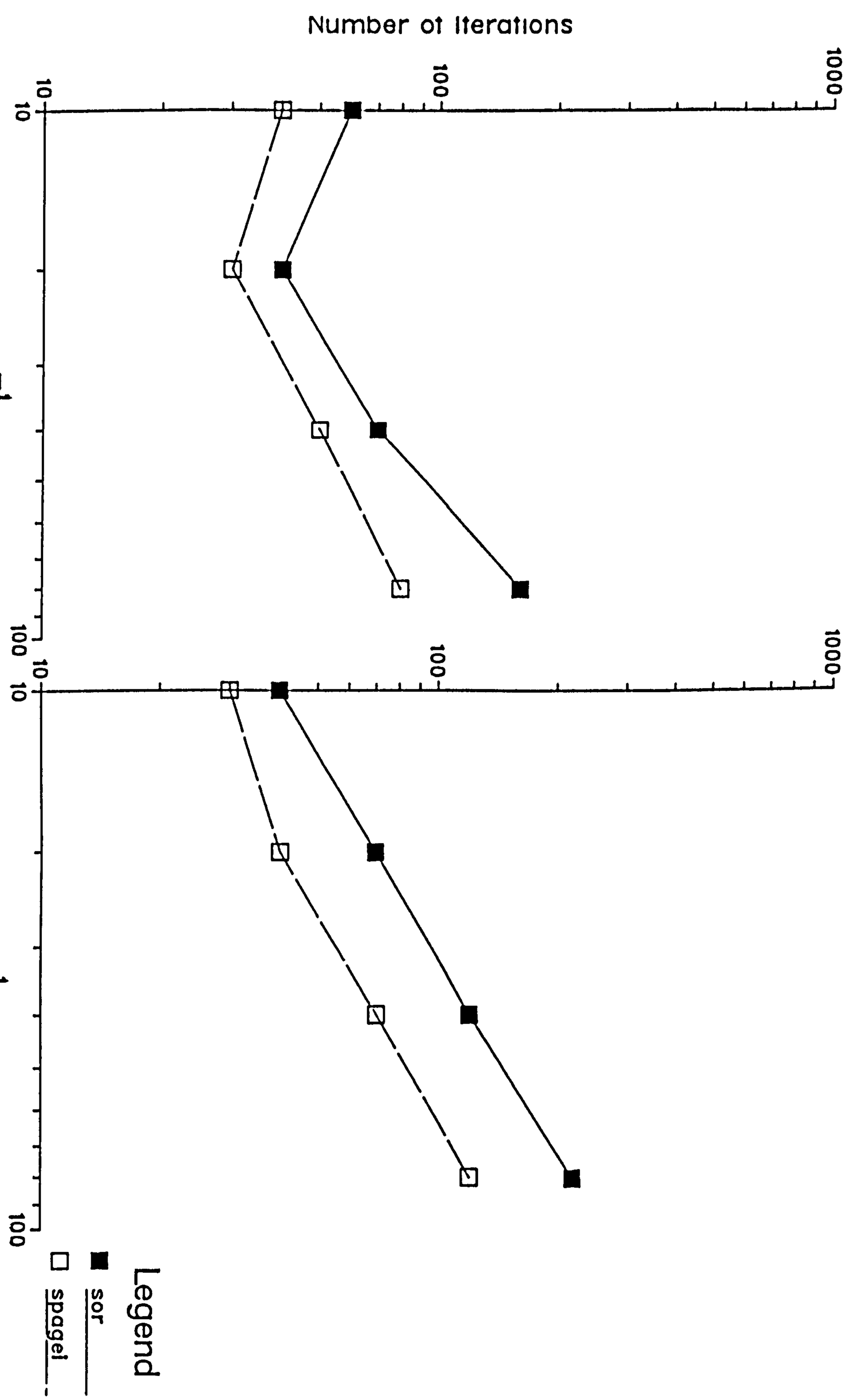


FIGURE 5.9.1: A comparison between the SOR and the SPAGEI methods

continued...

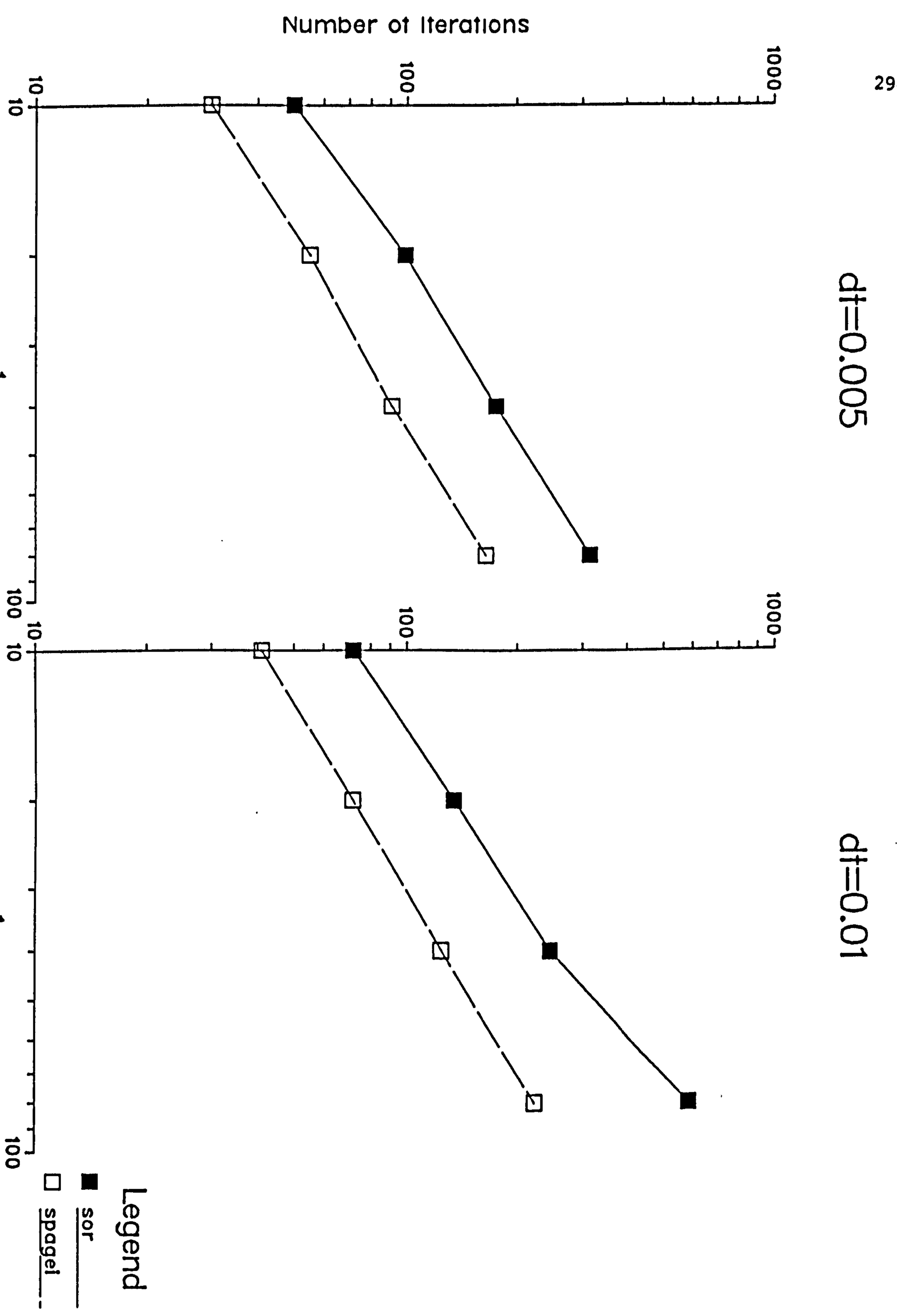


FIGURE 5.9.1: continued

continued...

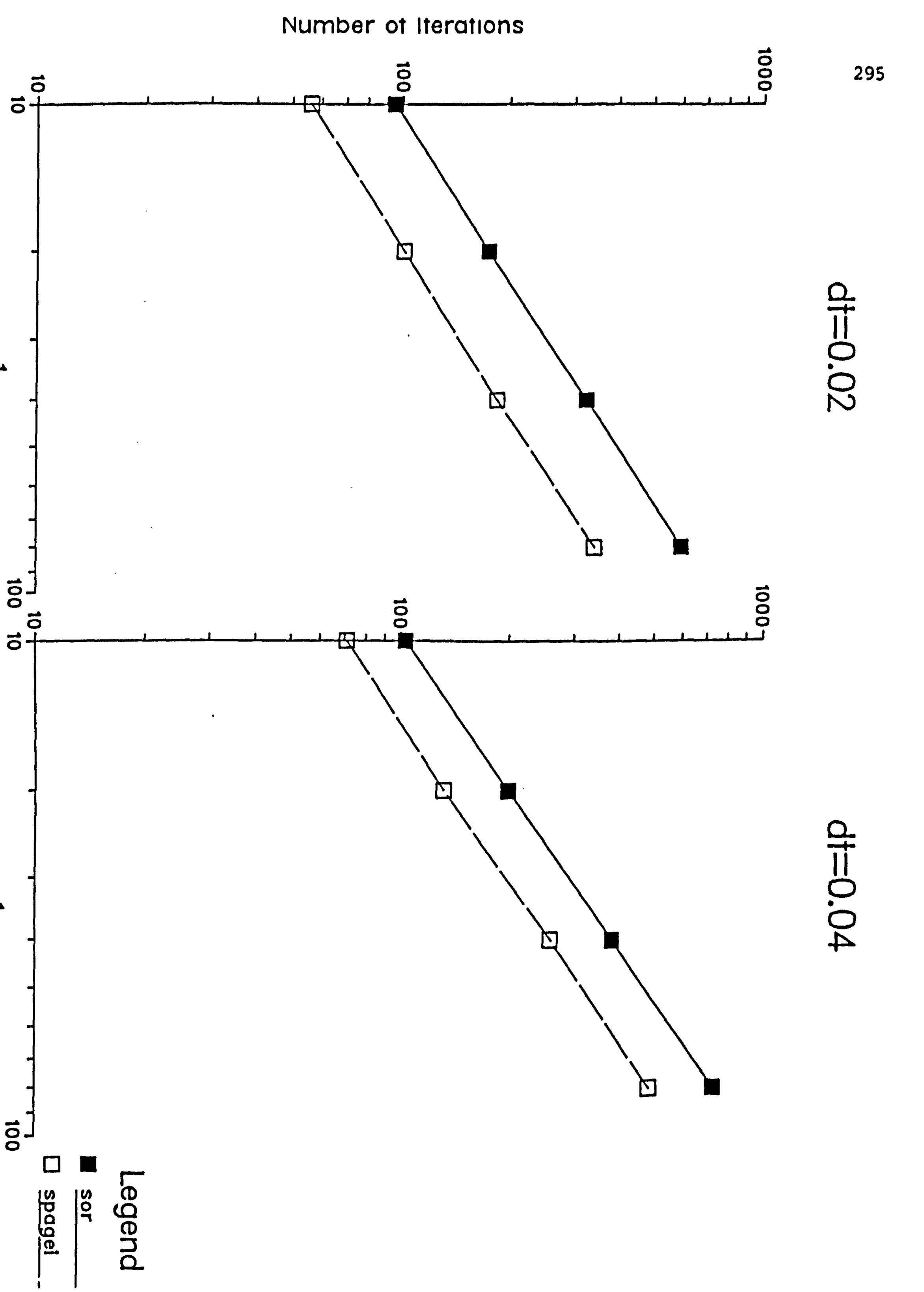


FIGURE 5.9.1: Continued

5.10 COMPUTATIONAL COMPLEXITY

Initially we may state that the truncation error for the cubic splines equation and thus the SPAGEI scheme is of order $(\Delta x)^2+(\Delta t)^2$, i.e. it is the same as that of the Crank-Nicolson equation.

From the molecular diagram of SPAGEI Figure (5.6.1) we find that for a half cycle of the iteration the number of multiplications is 4 and the number of additions is 3, with the addition of the right hand side element to make 4 multiplications and 4 additions as the total for half a cycle, i.e. for one cycle it is 8 multiplications and 8 additions.

5.11 REMARKS

a) Although the SPAGEI method was not compared directly with the SPAGE method, it is clearly seen that the SPAGEI method is superior in the sense of stability.

b) From the results in Tables (5.7.1),...,(5.7.15) we see that there is no significant difference in the number of iterations when $r<1.5$. The difference however becomes more marked in Tables 8,10, 13 and 15 where we notice that the Peaceman-Rachford parameter is better in the first three tables, while the Wachspress parameters are better in the latter table.

c) In Table 16, Sec.(5.7) the change in the parameter range does not generally affect the number of iterations except for certain values of r. For $r=0.5$ and 0.605, the parameters in the optimum range $(a+b)/2 \lesssim \rho \lesssim \sqrt{ab}$ reduced the number of iterations, while for $r=2.42$ and 4.84 they increased the number of iterations.

d) In Sec.(5.8) we clearly see the effect of increasing the number of parameters particularly for large values of h^{-1}. It is found by experiment that increasing the number of parameters to more than 3 or 4 does not have any significant effects and sometimes it even starts to increase the number of iterations.

e) In the comparison between the SPAGEI and the SOR methods we notice that the SPAGEI is faster (even though we have only used one parameter) as far as the number of iterations is concerned. From Table (5.9.2) we notice that in some cases the SOR method requires twice the number of iterations as SPAGEI, in particular, when $n=80$.

f) In the iteration process it could be of some advantage to use the new values $u^{(k+1)}$ instead of the old values $u^{(k)}$. Recalling

equations (5.3.11) and (5.3.12), by using the new values the two equations can be written as (supposing we evaluate the functions from left to right)

$$u_{i,j+1}^{(k+\frac{1}{2})} = (pu_{i-1,j+1}^{(k+\frac{1}{2})} + qu_{i,j+1}^{(k)} + tu_{i+1,j+1}^{(k)} + su_{i+2,j+1}^{(k)} + w_i)/det \ , \quad (5.11.1)$$

$$u_{i+1,j+1}^{(k+\frac{1}{2})} = (su_{i-1,j+1}^{(k+\frac{1}{2})} + tu_{i,j+1}^{(k+\frac{1}{2})} + qu_{i+1,j+1}^{(k)} + pu_{i+2,j+1}^{(k)} + w_{i+1})/det. \quad (5.11.2)$$

This would be applied again when doing the other half of the iteration as we start from right to left to give,

$$u_{i+1,j+1}^{(k+1)} = (su_{i-1,j+1}^{(k+\frac{1}{2})} + tu_{i,j+1}^{(k+\frac{1}{2})} + qu_{i+1,j+1}^{(k+\frac{1}{2})} + pu_{i+2,j+1}^{(k+1)} + w_{i+1})/det \quad (5.11.3)$$

$$u_{i,j+1}^{(k+1)} = (pu_{i-1,j+1}^{(k+\frac{1}{2})} + qu_{i,j+1}^{(k+\frac{1}{2})} + tu_{i+1,j+1}^{(k+1)} + su_{i+2,j+1}^{(k+1)} + w_i)/det \ . \quad (5.11.4)$$

This will give a new shape to the molecular diagram of the method which is shown in Fig.(5.11.1).

g) In actual fact, the number of iterations required for convergence is only 1 for the case of using the optimum value of the parameter, i.e. 3. This is not evident unless the results are printed out after each iteration, then we see that the results after the first iteration and those after the second iteration are identical. Whereas by using the criterion technique we have nothing to compare the results after the first iteration until the second iteration is completed. After the test is made and convergence fulfilled, the number of iterations becomes 2 instead of 1.

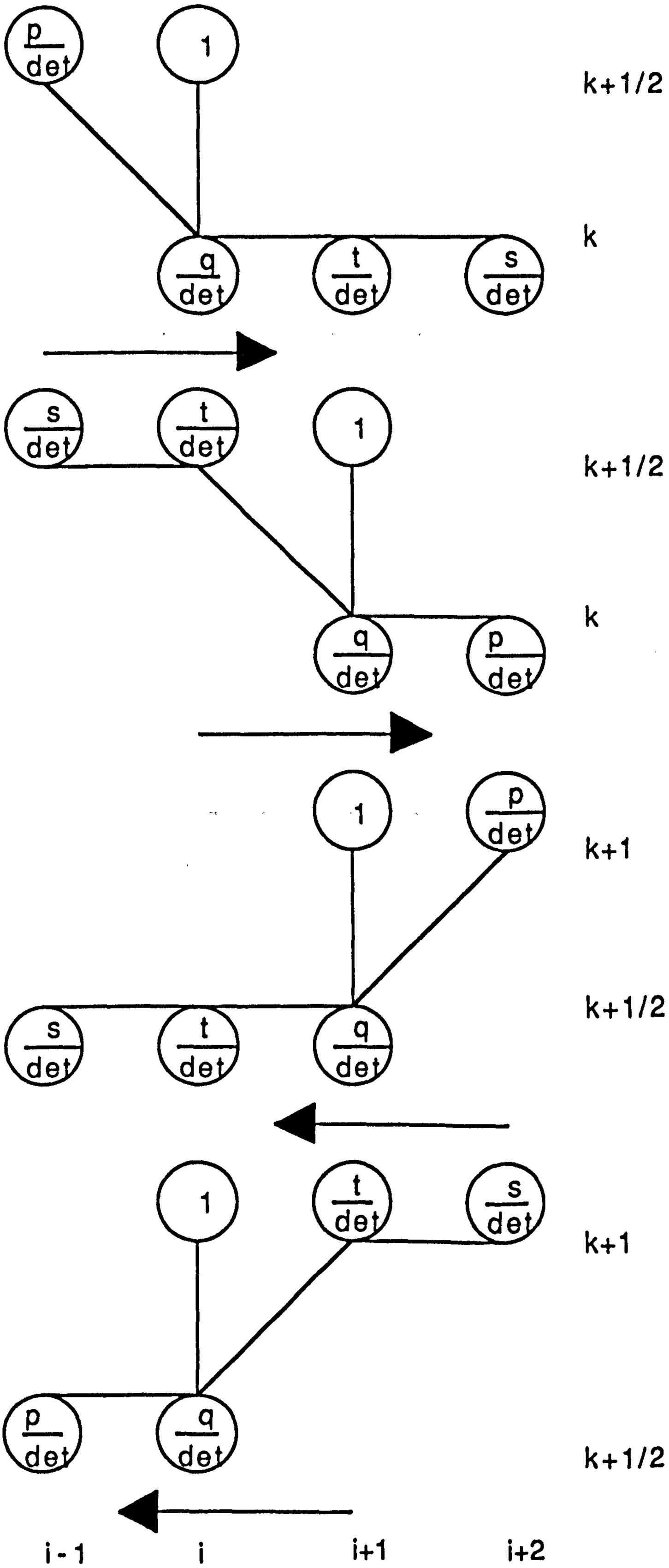


FIGURE 5.11.1: The molecular diagram of the suggested scheme of equation (5.11.1)-(5.11.4)

CHAPTER SIX

TWO DIMENSIONAL PROBLEMS

6.1 INTRODUCTION

In Sections (3.9)-(3.12) we introduced some numerical methods for solving two dimensional problems.

In this chapter we introduce an iterative explicit scheme which is an extension to the material presented in Chapter 5. The scheme is derived by splitting the quindiagonal coefficient matrix into two component matrices. In exactly the same manner as in the one dimensional case an iterative scheme is produced by transfering one of the two components (each in turn) to the right-hand side. The aim of this split process is to produce a (4×4) block diagonal system. The (4×4) blocks will not directly show by splitting. Reformation of the left-hand side part of the coefficient matrix will be needed, which is done by pre- and post-multiplication operations by the appropriate elementary matrices.

Also included in this chapter are some numerical results and analysis to confirm the validity of the proposed schemes.

6.2 THE SPLINE ALTERNATING EXPLICIT ITERATIVE (SPAGEI) METHOD FOR TWO DIMENSIONAL PROBLEMS

The problem that we want to solve is the two dimensional heat conduction equation,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} \qquad (6.2.1)$$

defined on the domain $0 \leq x,y \leq 1$ and $t \geq 0$.

Using the forward difference formula on $\frac{\partial U}{\partial t}$, and a weighted central difference formula on $\frac{\partial^2 U}{\partial x^2}$ and $\frac{\partial^2 U}{\partial y^2}$, will lead to the generalised equation

$$\frac{u_{i,j,k+1} - u_{i,j,k}}{\Delta t} \cong \frac{1}{(\Delta x)^2}\{\theta_1 \delta_x^2(u_{i,j,k+1}) + \theta_1' \delta_x^2(u_{i,j,k})\}$$

$$+ \frac{1}{(\Delta y)^2}\{\theta_2 \delta_y^2(u_{i,j,k+1}) + \theta_2' \delta_y^2(u_{i,j,k})\}. \qquad (6.2.2)$$

Let
$$\theta_i = \frac{1}{2} - \frac{1}{6r}, \quad i=1,2 \ ,$$

$$\theta_i' = \frac{1}{2} + \frac{1}{6r}, \quad i=1,2 \ ,$$

where
$$r = \frac{\Delta t}{(\Delta x)^2} = \frac{\Delta t}{(\Delta y)^2} \ .$$

This choice of θ_i and θ_i', $i=1,2$, is the one which leads to the cubic splines fromula for the one dimensional equation. Thus, analogously equation (6.2.2) will lead to the cubic splines equation for the two dimensional partial differential equations.

This is written as

$$(2+12r)u_{i,j,k+1} + (1-3r)(u_{i-1,j,k+1} + u_{i+1,j,k+1} + u_{i,j-1,k+1} + u_{i,j+1,k+1})$$

$$\cong (2-12r)u_{i,j,k} + (1+3r)(u_{i-1,j,k} + u_{i+1,j,k} + u_{i,j-1,k} + u_{i,j+1,k}) \qquad (6.2.3)$$

where the truncation error is $O(\Delta t^3+\Delta t\Delta x^2)$. The molecular diagrams of equation (6.2.3) is shown in Figure (6.2.1).

Considering equation (6.2.3) at the points (i,j), (i+1,j), (i,j+1) and (i+1,j+1) produces the system,

$$(2+12r)u_{i,j,k+1}+(1-3r)(u_{i-1,j,k+1}+u_{i+1,j,k+1}+u_{i,j-1,k+1}+u_{i,j+1,k+1})$$

$$= (2-12r)u_{i,j,k}+(1+3r)(u_{i-1,j,k}+u_{i+1,j,k}+u_{i,j-1,k}+u_{i,j+1,k}) , \qquad (6.2.3a)$$

$$(2+12r)u_{i+1,j,k+1}+(1-3r)(u_{i,j,k+1}+u_{i+2,j,k+1}+u_{i+1,j-1,k+1}+$$

$$u_{i+1,j+1,k+1}) = (2-12r)u_{i+1,j,k}+(1+3r)(u_{i,j,k}+u_{i+2,j,k}+$$

$$u_{i+1,j-1,k}+u_{i+1,j+1,k}) , \qquad (6.2.3b)$$

$$(2+12r)u_{i,j+1,k+1}+(1-3r)(u_{i-1,j+1,k+1}+u_{i+1,j+1,k+1}+u_{i,j,k+1}+$$

$$u_{i,j+2,k+1}) = (2-12r)u_{i,j+1,k}+(1+3r)(u_{i-1,j+1,k}+u_{i,j,k}+$$

$$u_{i,j+2,k}) , \qquad (6.2.3c)$$

$$(2+12r)u_{i+1,j+1,k+1}+(1-3r)(u_{i,j+1,k+1}+u_{i+2,j+1,k+1}+u_{i+1,j,k+1}+$$

$$u_{i+1,j+2,k+1}) = (2-12r)u_{i+1,j+1,k}+(1-3r)(u_{i,j+1,k}+u_{i+2,j+1,k}+$$

$$u_{i+1,j,k}+u_{i+1,j+2,k}) , \qquad (6.2.3d)$$

where the molecular diagram of this system is shown in Figure (6.2.2).

Our aim is to leave only the 4 points (i,j,k+1), (i+1,j,k+1), (i,j+1,k+1) and (i+1,j+1,k+1), which make the square in the middle of Figure (6.2.2), with half the coefficient of the diagonal points on the left-hand side. Then transfer all the remaining points plus the diagonal points with their half coefficients to the right-hand side. This will

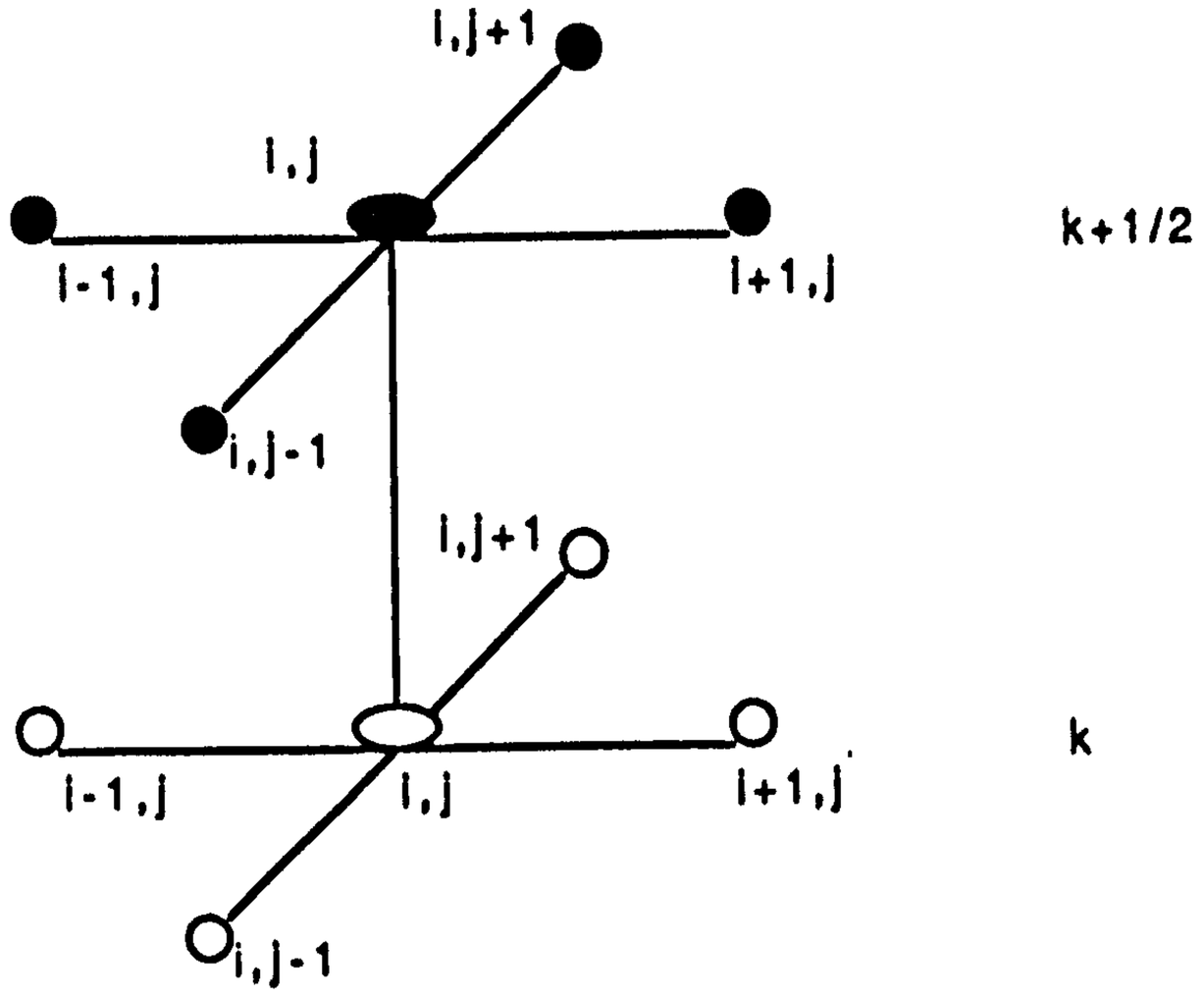


FIGURE 6.2.1

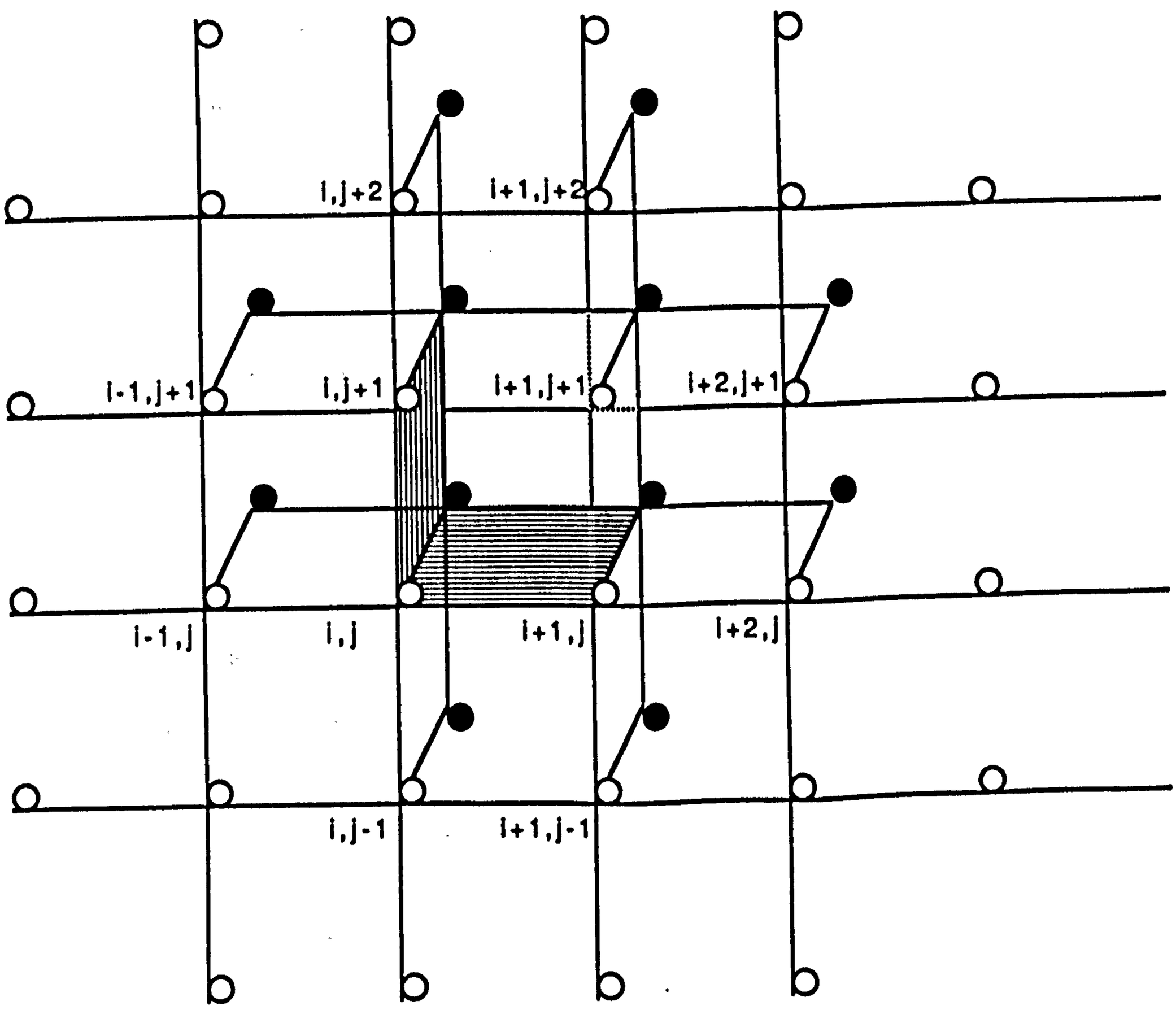


FIGURE 6.2.2

take the form of a (4×4) system which is written as,

$$\begin{bmatrix} a & f & f & 0 \\ f & a & 0 & f \\ f & 0 & a & f \\ 0 & f & f & a \end{bmatrix} \begin{bmatrix} u_{i,j} \\ u_{i+1,j} \\ u_{i,j+1} \\ u_{i+1,j+1} \end{bmatrix}_{k+1}^{(n+\frac{1}{2})} = \begin{bmatrix} b_{i,j} \\ b_{i+1,j} \\ b_{i,j+1} \\ b_{i+1,j+1} \end{bmatrix} - \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & a \end{bmatrix} \begin{bmatrix} u_{i,j} \\ u_{i+1,j} \\ u_{i,j+1} \\ u_{i+1,j+1} \end{bmatrix}_{k+1}^{(n)}$$

$$- f \begin{bmatrix} u_{i-1,j}+u_{i,j-1} \\ u_{i+1,j-1}+u_{i+2,j} \\ u_{i+1,j+1}+u_{i,j+2} \\ u_{i+1,j+2}+u_{i+2,j+1} \end{bmatrix}_{k+1}^{(n)} , \qquad (6.2.4)$$

where

$$a = \frac{(2+12r)}{2} = 1+6r, \; f = 1-3r,$$

the subscript (k+1) is the time level, the superscript (n) is the iteration counter and

$$b_{i,j} = (2-12r)u_{i,j,k}+(1+3r)(u_{i-1,j,k}+u_{i+1,j,k}+u_{i,j-1,k}+u_{i,j+1,k})$$

Now let the system (6.2.4) be written in matrix form as

$$A_1 u^{(n+\frac{1}{2})} = b-A_2 u^{(n)} , \qquad (6.2.5)$$

where $A_1 u^{(n+\frac{1}{2})}$ is the l.h.s. of (6.24), $b=[b_{i,j}, b_{i+1,j}, b_{i,j+1}, b_{i+1,j+1}]$ and $A_2 u^{(n)}$ is the r.h.s. of (6.2.4). For convenience we delete the subscript (k) since we are iterating within the same time level. Adding an acceleration parameter ρ to (6.2.5) results in the equation.

$$(A_1+\rho I)u^{(n+\frac{1}{2})} = b-(A_2-\rho I)u^{(n)} . \qquad (6.2.6)$$

Thus inverting the coefficient matrix on the l.h.s. gives the equation,

$$\begin{bmatrix} u_{i,j} \\ u_{i+1,j} \\ u_{i,j+1} \\ u_{i+1,j+1} \end{bmatrix}^{(n+\frac{1}{2})} = \frac{1}{\det} \begin{bmatrix} a_p^2-2f^2 & -a_pf & -a_pf & 2f^2 \\ -a_pf & a_p^2-2f^2 & 2f^2 & -a_pf \\ -a_pf & 2f^2 & a_p^2-2f^2 & -a_pf \\ 2f^2 & -a_pf & -a_pf & a_p^2-2f^2 \end{bmatrix} \left\{ \begin{bmatrix} b_{i,j} \\ b_{i+1,j} \\ b_{i,j+1} \\ b_{i+1,j+1} \end{bmatrix} \right.$$

$$\left. -a_m \begin{bmatrix} u_{i,j} \\ u_{i+1,j} \\ u_{i,j+1} \\ u_{i+1,j+1} \end{bmatrix}^{(n)} -f \begin{bmatrix} u_{i-1,j}+u_{i,j-1} \\ u_{i+1,j-1}+u_{i+2,j} \\ u_{i-1,j+1}+u_{i,j+2} \\ u_{i+1,j+2}+u_{i+2,j+1} \end{bmatrix}^{(n)} \right\} \quad (6.2.7)$$

where $a_p=a+\rho$, $a_m=a-\rho$ and $\det=a_p(a_p^2-4f^2)$.

The equation (6.2.7) can be written in 4 single equations of the form,

$$u_{i,j}^{(n+\frac{1}{2})} = \frac{1}{\det}[w_{i,j}-a_m\{(a_p^2-2f^2)u_{i,j,k}^{(n)}-a_pf(u_{i+1,j}^{(n)}+u_{i,j+1}^{(n)})+2f^2u_{i+1,j+1}^{(n)}\}$$

$$-f\{(a_p^2-2f^2)(u_{i-1,j}^{(n)}+u_{i,j-1}^{(n)})-a_pf(u_{i+1,j-1}^{(n)}+u_{i+2,j}^{(n)}+u_{i-1,j+1}^{(n)}$$

$$+u_{i,j+2}^{(n)})+2f^2(u_{i+1,j+2}^{(n)}+u_{i+2,j+1}^{(n)})\}] \ , \quad (6.2.8a)$$

$$u_{i+1,j}^{(n+\frac{1}{2})} = \frac{1}{\det}[w_{i+1,j}-a_m\{-a_pf(u_{i,j}^{(n)}+u_{i+1,j+1}^{(n)})+(a_p^2-2f^2)u_{i+1,j}^{(n)}+$$

$$2f^2u_{i,j+1}^{(n)}\}-f\{-a_pf(u_{i-1,j}^{(n)}+u_{i,j-1}^{(n)}+u_{i+1,j+2}^{(n)}+u_{i+2,j+1}^{(n)})$$

$$+(a_p^2-2f^2)(u_{i+1,j-1}^{(n)}+u_{i+2,j}^{(n)})-2f^2(u_{i-1,j+1}^{(n)}+u_{i,j+2}^{(n)})\}] \quad (6.2.8b)$$

$$u_{i,j+1}^{(n+\frac{1}{2})} = \frac{1}{\det}[w_{i,j+1}-a_m\{-a_pf(u_{i,j}^{(n)}+u_{i+1,j+1}^{(n)})+2f^2u_{i+1,j}^{(n)}+(a_p^2-2f^2)$$

$$u_{i,j+1}^{(n)}\}-f\{-a_pf(u_{i-1,j}^{(n)}+u_{i,j-1}^{(n)}+u_{i+1,j+2}^{(n)}+u_{i+2,j+1}^{(n)})$$

$$+2f^2(u_{i+1,j-1}^{(n)}+u_{i+2,j}^{(n)})+(a_p^2-2f^2)(u_{i-1,j+1}^{(n)}+u_{i,j+2}^{(n)})\}], \quad (6.2.8c)$$

$$u_{i+1,j+1}^{(n+\frac{1}{2})} = \frac{1}{\det}[w_{i+1,j+1} - a_m\{2f^2 u_{i,j}^{(n)} - a_p f(u_{i+1,j}^{(n)} + u_{i,j+1}^{(n)})$$

$$+(a_p^2-2f^2)u_{i+1,j+1}^{(n)}\} - f\{2f^2(u_{i-1,j}^{(n)} + u_{i,j-1}^{(n)}) - a_p f(u_{i+1,j-1}^{(n)}$$

$$+u_{i+2,j}^{(n)} + u_{i-1,j+1}^{(n)} + u_{i,j+2}^{(n)}) + (a_p^2-2f^2)(u_{i+1,j+2}^{(n)} + u_{i+2,j+1}^{(n)})\}] \qquad (6.2.8d)$$

where,

$$w_{i,j} = (a_p^2-2f^2)b_{i,j} - a_p f(b_{i+1,j} + b_{i,j+1}) + 2f^2 b_{i+1,j+1} ,$$

$$w_{i+1,j} = (a_p^2-2f^2)b_{i+1,j} - a_p f(b_{i,j} + b_{i+1,j+1}) + 2f^2 b_{i,j+1} ,$$

$$w_{i,j+1} = (a_p^2-2f^2)b_{i,j+1} - a_p f(b_{i,j} + b_{i+1,j+1}) + 2f^2 b_{i,j+1} ,$$

and $$w_{i+1,j+1} = (a_p^2-2f^2)b_{i+1,j+1} - a_p f(b_{i+1,j} + b_{i,j+1}) + 2f^2 b_{i,j} .$$

The iterative process is continued until convergence is achieved.

The molecular diagram of (6.2.4) is shown in Figure (6.2.3) and the diagrams of (6.2.8a,b,c and d) are shown in Figures (6.2.4a,b,c and d) respectively.

Therefore in order to get the re-arranged system we first write the system that represents all points in the region. Let the region to be examined in the (x,y,t) space be covered by a rectilinear grid with sides parallel to the axes, with h and k being the grid spacing in the distance and time directions change respectively. Thus, there are $(\ell-1)$ $(\hat{m}-1)$ interior points in the region. For simplicity let ℓ and $\hat{m}$ (without loss of generality) be equal 6 (Figure 6.2.5), then for the usual column ordering of the grid points the linear system will be,

$$Au_{k+1} = Cu_k + g = b , \qquad (6.2.9)$$

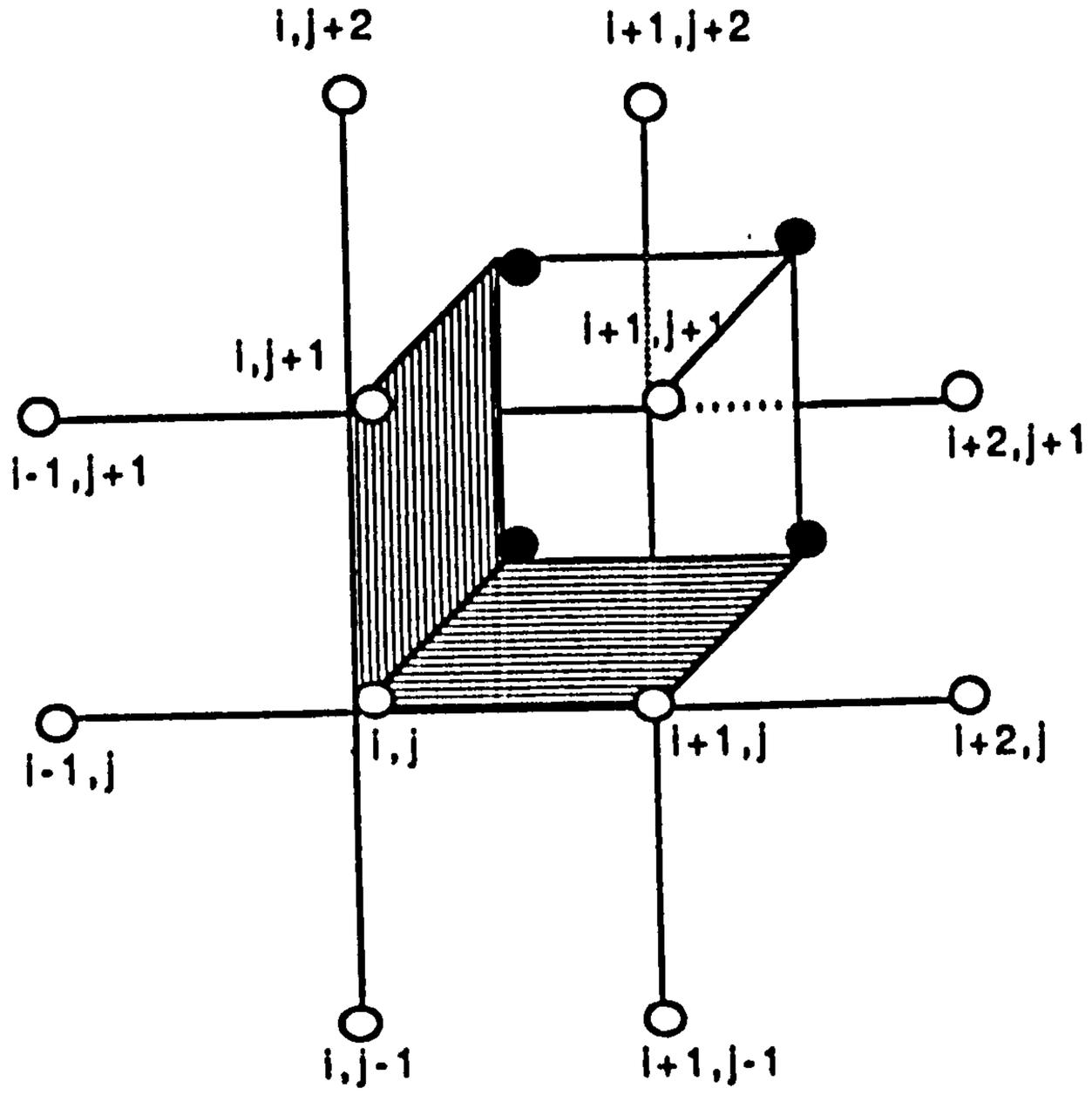


FIGURE 6.2.3

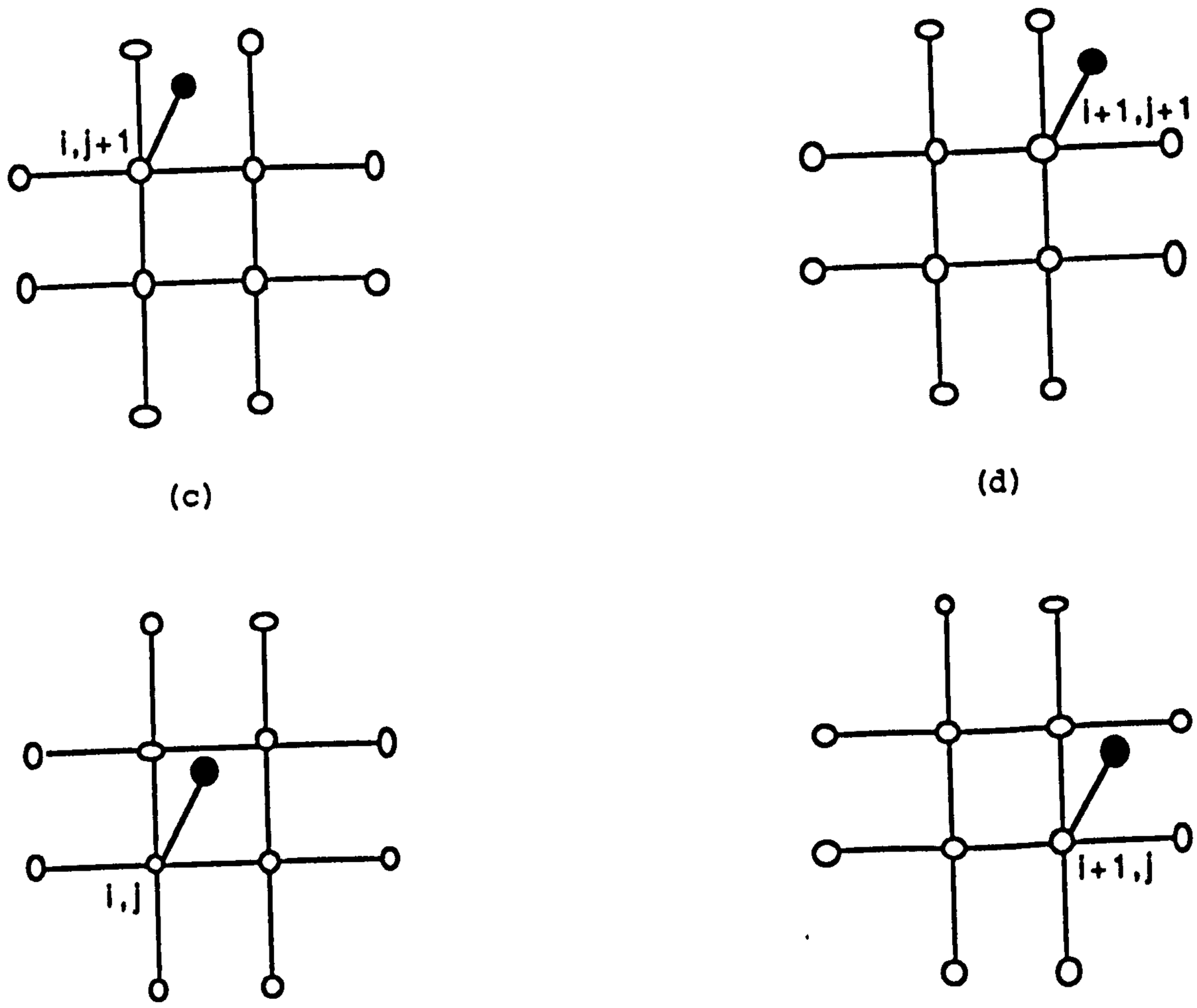


FIGURE 6.2.4

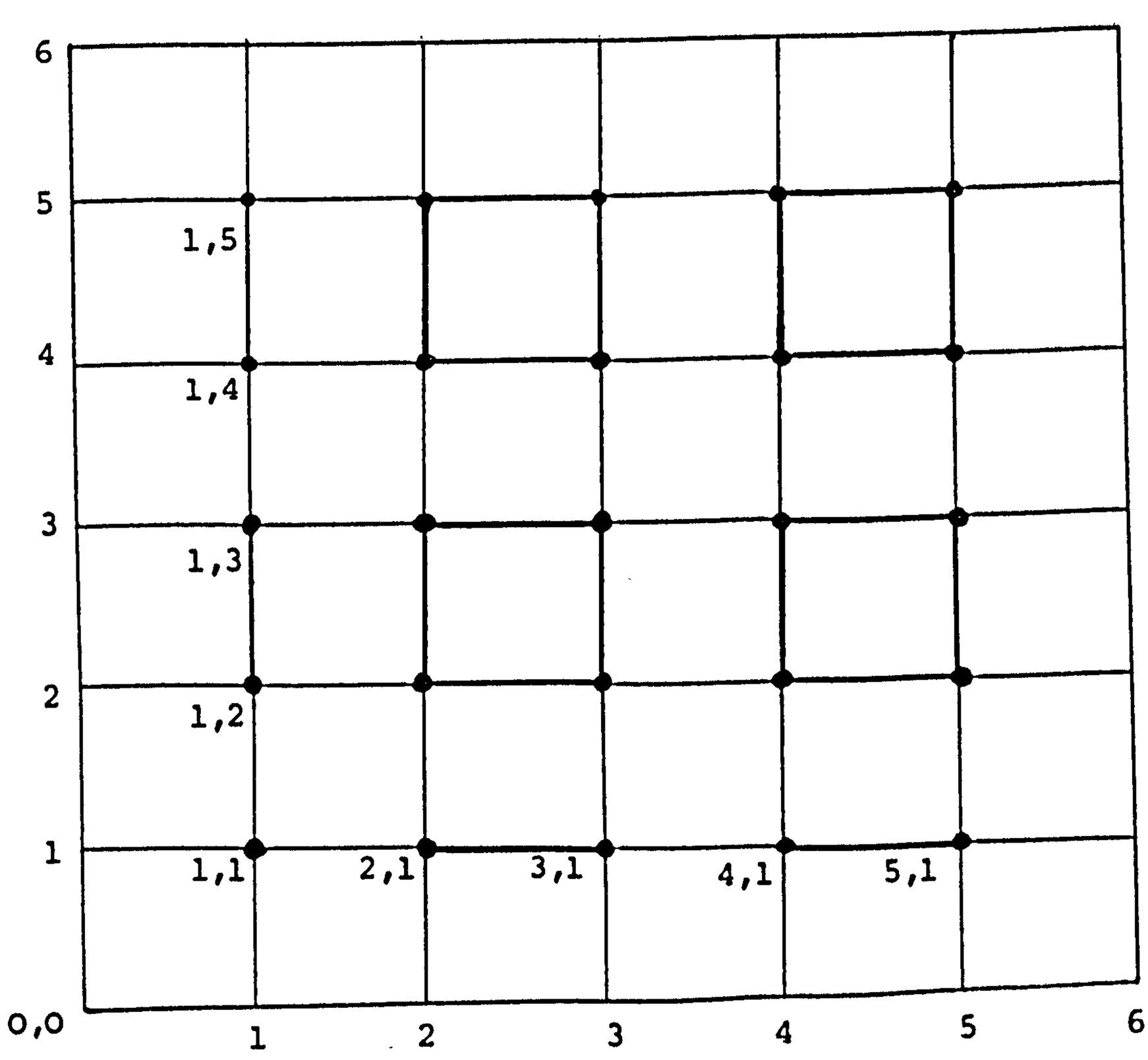


FIGURE 6.2.5

where

$$
A = \left[\begin{array}{ccccc|ccccc|ccccc|ccccc|ccccc}
2a&f&&& & f&&&& & &&&& & &&&& & &&&& \\
f&2a&f&& & &f&&& & &&&& & &&&& & &&&& \\
&f&2a&f& & &&f&& & &&&& & &&&& & &&&& \\
&&f&2a&f & &&&f& & &&&& & &&&& & &&&& \\
&&&f&2a & &&&&f & &&&& & &&&& & &&&& \\
\hline
f&&&& & 2a&f&&& & f&&&& & &&&& & &&&& \\
&f&&& & f&2a&f&& & &f&&& & &&O&& & &&&& \\
&&f&& & &f&2a&f& & &&f&& & &&&& & &&&& \\
&&&f& & &&f&2a&f & &&&f& & &&&& & &&&& \\
&&&&f & &&&f&2a & &&&&f & &&&& & &&&& \\
\hline
&&&& & f&&&& & 2a&f&&& & f&&&& & &&&& \\
&&&& & &f&&& & f&2a&f&& & &f&&& & &&&& \\
&&&& & &&f&& & &f&2a&f& & &&f&& & &&&& \\
&&&& & &&&f& & &&f&2a&f & &&&f& & &&&& \\
&&&& & &&&&f & &&&f&2a & &&&&f & &&&& \\
\hline
&&&& & &&&& & f&&&& & 2a&f&&& & f&&&& \\
&&&& & &&&& & &f&&& & f&2a&f&& & &f&&& \\
&&&& & &&O&& & &&f&& & &f&2a&f& & &&f&& \\
&&&& & &&&& & &&&f& & &&f&2a&f & &&&f& \\
&&&& & &&&& & &&&&f & &&&f&2a & &&&&f \\
\hline
&&&& & &&&& & &&&& & f&&&& & 2a&f&&& \\
&&&& & &&&& & &&&& & &f&&& & f&2a&f&& \\
&&&& & &&&& & &&&& & &&f&& & &f&2a&f& \\
&&&& & &&&& & &&&& & &&&f& & &&f&2a&f \\
&&&& & &&&& & &&&& & &&&&f & &&&f&2a
\end{array}\right],
$$

$$
C = \left[\begin{array}{ccccc|ccccc|ccccc|ccccc|ccccc}
a' & f' & & & & f' & & & & & & & & & & & & & & & & & & & \\
f' & a' & f' & & & & f' & & & & & & & & & & & & & & & & & & \\
& f' & a' & f' & & & & f' & & & & & & & & & & & & & & & & & \\
& & f' & a' & f' & & & & f' & & & & & & & & & & & & & & & & \\
& & & f' & a' & & & & & f' & & & & & & & & & & & & & & & \\
\hline
f' & & & & & a' & f' & & & & f' & & & & & & & & & & & & & & \\
& f' & & & & f' & a' & f' & & & & f' & & & & & & & & & & & & & \\
& & f' & & & & f' & a' & f' & & & & f' & & & & & C & & & & & & & \\
& & & f' & & & & f' & a' & f' & & & & f' & & & & & & & & & & & \\
& & & & f' & & & & f' & a' & & & & & f' & & & & & & & & & & \\
\hline
& & & & & f' & & & & & a' & f' & & & & f' & & & & & & & & & \\
& & & & & & f' & & & & f' & a' & f' & & & & f' & & & & & & & & \\
& & & & & & & f' & & & & f' & a' & f' & & & & f' & & & & & & & \\
& & & & & & & & f' & & & & f' & a' & f' & & & & f' & & & & & & \\
& & & & & & & & & f' & & & & f' & a' & & & & & f' & & & & & \\
\hline
& & & & & & & & & & f' & & & & & a' & f' & & & & f' & & & & \\
& & & & & & & & & & & f' & & & & f' & a' & f' & & & & f' & & & \\
& & & & & & & & & & & & f' & & & & f' & a' & f' & & & & f' & & \\
& & & & & & & & & & & & & f' & & & & f' & a' & f' & & & & f' & \\
& & & & & & & & & & & & & & f' & & & & f' & a' & & & & & f' \\
\hline
& & & & & & & & & & & & & & & f' & & & & & a' & f' & & & \\
& & & & & & & & & & & & & & & & f' & & & & f' & a' & f' & & \\
& & & & & & & O & & & & & & & & & & f' & & & & f' & a' & f' & \\
& & & & & & & & & & & & & & & & & & f' & & & & f' & a' & f' \\
& & & & & & & & & & & & & & & & & & & f' & & & & f' & a'
\end{array}\right]
\mu_k = \left[\begin{array}{c}
u_{1,1} \\ u_{1,2} \\ u_{1,3} \\ u_{1,4} \\ u_{1,5} \\ \hline
u_{2,1} \\ u_{2,2} \\ u_{2,3} \\ u_{2,4} \\ u_{2,5} \\ \hline
u_{3,1} \\ u_{3,2} \\ u_{3,3} \\ u_{3,4} \\ u_{3,5} \\ \hline
u_{4,1} \\ u_{4,2} \\ u_{4,3} \\ u_{4,4} \\ u_{4,5} \\ \hline
u_{5,1} \\ u_{5,2} \\ u_{5,3} \\ u_{5,4} \\ u_{5,5}
\end{array}\right],
$$

g is the vector associated with the boundary conditions, $a'=(2-12r)$ and $f'=(1+3r)$.

Following the same procedure as in the previous chapter, we split the matrix A in (6.2.9) into A_1 and A_2 to give the iterative system,

$$A_1 u^{(n+\frac{1}{2})} = b-A_2 u^{(n)} \quad , \tag{6.2.10a}$$

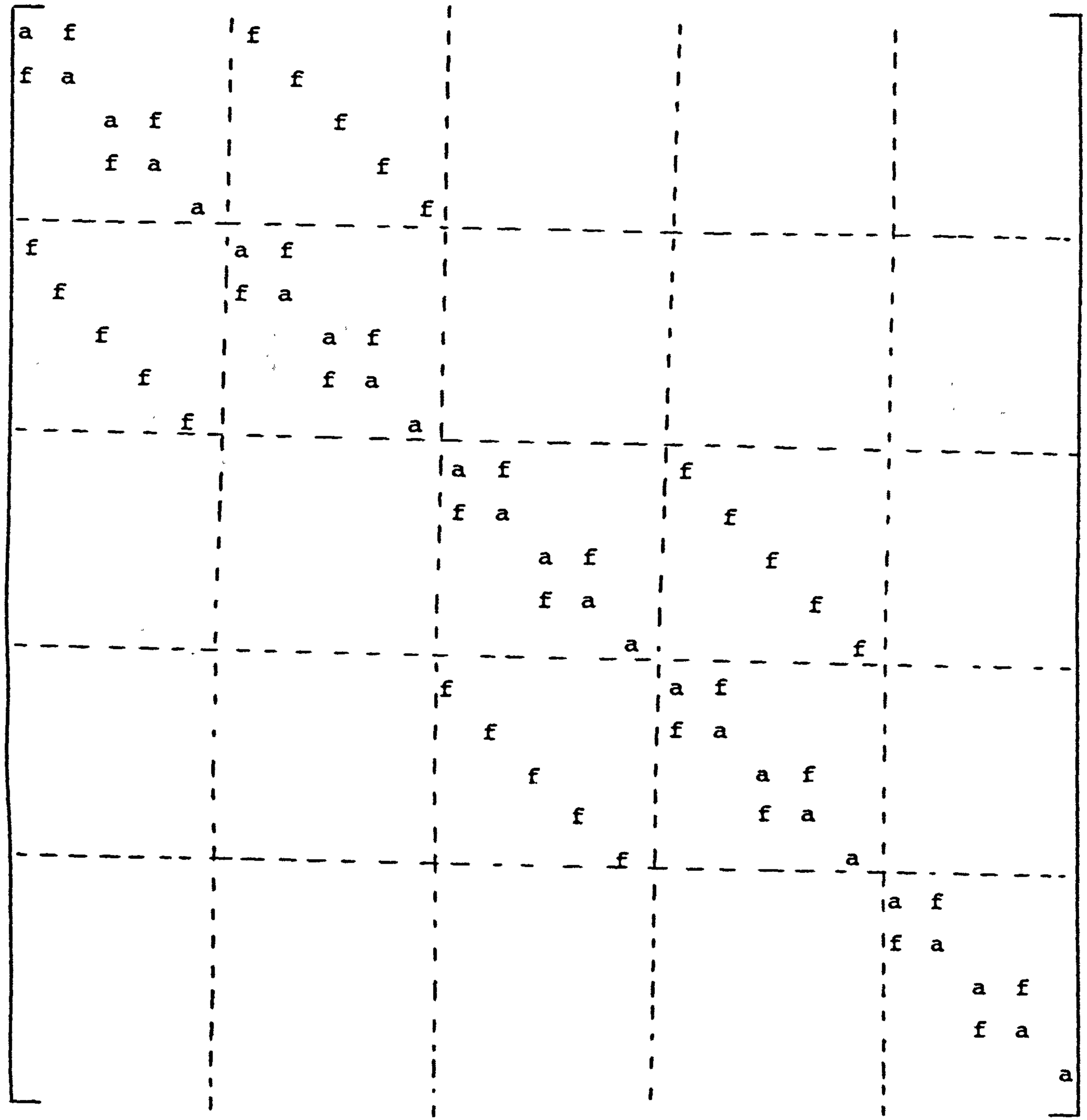
$$A_2 u^{(n+1)} = b-A_1 u^{(n+\frac{1}{2})} \quad , \tag{6.2.10b}$$

where,

$$
A_1 = \left[\begin{array}{ccccc:ccccc:ccccc:ccccc:ccccc}
a&&&& & &&&& & &&&& & &&&& & &&&& \\
&a&f&& & &&&& & &&&& & &&&& & &&&& \\
&f&a&& & &&&& & &&&& & &&&& & &&&& \\
&&&a&f & &&&& & &&&& & &&&& & &&&& \\
&&&f&a & &&&& & &&&& & &&&& & &&&& \\
\hdashline
&&&& & a&&&& & f&&&& & &&&& & &&&& \\
&&&& & &a&f&& & &f&&& & &&&& & &&&& \\
&&&& & &f&a&& & &&f&& & &&&& & &&&& \\
&&&& & &&&a&f & &&&f& & &&&& & &&&& \\
&&&& & &&&f&a & &&&&f & &&&& & &&&& \\
\hdashline
&&&& & f&&&& & a&&&& & &&&& & &&&& \\
&&&& & &f&&& & &a&f&& & &&&& & &&&& \\
&&&& & &&f&& & &f&a&& & &&&& & &&&& \\
&&&& & &&&f& & &&&a&f & &&&& & &&&& \\
&&&& & &&&&f & &&&f&a & &&&& & &&&& \\
\hdashline
&&&& & &&&& & &&&& & a&&&& & f&&&& \\
&&&& & &&&& & &&&& & &a&f&& & &f&&& \\
&&&& & &&&& & &&&& & &f&a&& & &&f&& \\
&&&& & &&&& & &&&& & &&&a&f & &&&f& \\
&&&& & &&&& & &&&& & &&&f&a & &&&&f \\
\hdashline
&&&& & &&&& & &&&& & f&&&& & a&&&& \\
&&&& & &&&& & &&&& & &f&&& & &a&f&& \\
&&&& & &&&& & &&&& & &&f&& & &f&a&& \\
&&&& & &&&& & &&&& & &&&f& & &&&a&f \\
&&&& & &&&& & &&&& & &&&&f & &&&f&a
\end{array}\right]
$$

and

$A_2 =$



Let us consider equation (6.2.10a). In order to have a group of 4 points [say the points (2,2)-(2,3)-(3,2)-(3,3), see Fig. (6.2.5)] ordered in a square form on the grid region and retain the same systematic representation as in equation (6.2.4) we need to rearrange the linear system as follows.

To do this we need, as mentioned in Section(6.1), a similarity transformation.

Let us consider the 4 points above, which are included in the first 10×10 block submatrix of A_1. Thus, the l.h.s. of equation (6.2.10a) (for the 10×10 block) will be

$$\begin{bmatrix} a & f & & & & & & & & \\ f & a & f & & & & & & & \\ & f & a & & & f & & & & \\ & & & a & f & & & f & & \\ & & & f & a & & & & f & \\ & & & & & a & f & & & \\ & & & & & f & a & f & & \\ & & & & & & f & a & & \\ & & & & & & & & a & f \\ & & & & & & & & f & a \end{bmatrix} \begin{bmatrix} u_{2,1} \\ u_{2,2} \\ u_{2,3} \\ u_{2,4} \\ u_{2,5} \\ u_{3,1} \\ u_{3,2} \\ u_{3,3} \\ u_{3,4} \\ u_{3,5} \end{bmatrix} , \qquad (6.2.11)$$

or in matrix notation

$$A_{2,3}{}^{u} . \qquad (6.2.12)$$

The elements that are involved in our target 4×4 matrix are all inside the dotted strip in (6.2.11).

Post multiplying $A_{2,3}$ by $I_{4,7} \cdot I_{5,8}$, gives, where $I_{4,7} \cdot I_{5,8}$ is as defined overleaf,

$$
\overset{A_{2,3}}{\left[\begin{array}{ccccc|ccccc}
a & & & & & f & & & & \\
 & a & f & & & & f & & & \\
 & f & a & & & & & f & & \\
 & & & a & f & & & & f & \\
 & & & f & a & & & & & f \\
\hline
f & & & & & a & & & & \\
 & f & & & & & a & f & & \\
 & & f & & & & f & a & & \\
 & & & f & & & & & a & f \\
 & & & & f & & & & f & a
\end{array}\right]}
\times
\overset{I_{4,7}\cdot I_{5,8}}{\left[\begin{array}{ccccc|ccccc}
1 & & & & & & & & & \\
 & 1 & & & & & & & & \\
 & & 1 & & & & & & & \\
 & & & 0 & & 0 & 1 & 0 & 0 & 0 \\
 & & & & 0 & 0 & 0 & 1 & 0 & 0 \\
\hline
 & & & 0 & 0 & 1 & & & & \\
 & & & 1 & 0 & & 0 & & & \\
 & & & 0 & 1 & & & 0 & & \\
 & & & 0 & 0 & & & & 1 & \\
 & & & 0 & 0 & & & & & 1
\end{array}\right]}
$$

$$
= \left[\begin{array}{ccccc|ccccc}
a & & & & & f & & & & \\
 & a & f & f & 0 & & & & & \\
 & f & a & 0 & f & & & & & \\
 & & & 0 & & 0 & a & f & f & 0 \\
 & & & & 0 & 0 & f & a & 0 & f \\
\hline
f & & & & & 0 & & & & \\
 & f & 0 & a & f & & 0 & & & \\
 & & f & f & a & & & 0 & & \\
 & & & & & & f & 0 & a & f \\
 & & & & & & 0 & f & f & a
\end{array}\right], \qquad (6.2.13)
$$

and premultiplying the resulting matrix by $I_{4,7} \cdot I_{5,8}$ gives,

$$\left[\begin{array}{cccc|cccc}
a & f & f & O & f & & & \\
f & a & O & f & & & & \\
f & O & a & f & & & & \\
O & f & f & a & & & & \\
\hline
f & & & & a & f & f & O \\
 & & & & f & a & O & f \\
 & & & & f & O & a & f \\
 & & & & O & f & f & a
\end{array}\right] .$$

Applying the same similarity transformation on the vector u of (6.2.11) gives,

$$[u_{2,1}\ u_{2,2}\ u_{2,3}\ u_{3,2}\ u_{3,3}\ u_{3,1}\ u_{2,4}\ u_{2,5}\ u_{3,4}\ u_{3,5}]^T .$$

Therefore the resulting 10×10 block will be

$$\left[\begin{array}{cccc|cccc}
a & f & f & O & f & & & \\
f & a & O & f & & & & \\
f & O & a & f & & & & \\
O & f & f & a & & & & \\
\hline
f & & & & a & f & f & O \\
 & & & & f & a & O & f \\
 & & & & f & O & a & f \\
 & & & & O & f & f & a
\end{array}\right]
\left[\begin{array}{c} u_{2,1} \\ u_{2,2} \\ u_{2,3} \\ u_{3,2} \\ u_{3,3} \\ u_{3,1} \\ u_{2,4} \\ u_{2,5} \\ u_{3,4} \\ u_{3,5} \end{array}\right] , \qquad (6.2.14)$$

which shows clearly the 4×4 block corresponding to the points 2,2-2,3-3,2-3,3 is isolated and can be treated separately. Thus with the suitable sequence of similarity transformations A_1 could be formed into

a block diagonal matrix with 4×4 sub-blocks, which can be inverted separately as previously done with the system (6.2.4).

For points near the boundary there are other groups of 2 which appear as line segments on the grid region (Fig. 6.2.5) and a block of order 2×2 in the linear system, and finally a single point representing the corner points as the point (1,1) in Figure (6.2.5).

For the groups of 2 points and their related equations we have:

1. Left side:

$$\begin{bmatrix} a_p & f \\ f & a_p \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{1,j+1} \end{bmatrix}^{(n+\frac{1}{2})} = \begin{bmatrix} b_{1,j} \\ b_{1,j+1} \end{bmatrix} - a_m \begin{bmatrix} u_{1,j} \\ u_{1,j+1} \end{bmatrix}^{(n)} - \begin{bmatrix} u_{1,j-1}+u_{2,j} \\ u_{1,j+2}+u_{2,j+1} \end{bmatrix}^{(n)} \tag{6.2.15}$$

to give,

$$u_{1,j}^{(n+\frac{1}{2})} = \frac{1}{\det}[w_{1,j}-a_m(a_p u_{1,j}^{(n)}-f u_{1,j+1}^{(n)})-f\{a_p(u_{1,j-1}^{(n)}+u_{2,j}^{(n)}) -f(u_{1,j+2}^{(n)}+u_{2,j+1}^{(n)})\}] , \tag{6.2.16a}$$

$$u_{1,j+1}^{(n+\frac{1}{2})} = \frac{1}{\det}[w_{1,j+1}-a_m(-f u_{1,j}^{(n)}+a_p u_{1,j+1}^{(n)})-f\{-f(u_{1,j-1}^{(n)}+u_{2,j}^{(n)}) +a_p(u_{1,j+2}^{(n)}+u_{2,j+1}^{(n)})\}] . \tag{6.2.16b}$$

2. Right side:

$$\begin{bmatrix} a_p & f \\ f & a_p \end{bmatrix} \begin{bmatrix} u_{\ell-1,j} \\ u_{\ell-1,j+1} \end{bmatrix}^{(n+\frac{1}{2})} = \begin{bmatrix} b_{\ell-1,j} \\ b_{\ell-1,j+1} \end{bmatrix} - a_m \begin{bmatrix} u_{\ell-1,j} \\ u_{\ell-1,j+1} \end{bmatrix}^{(n)} - f \begin{bmatrix} u_{\ell-2,j}^{(n)}+u_{\ell-1,j-1}^{(n)} \\ u_{\ell-2,j+1}^{(n)}+u_{\ell-1,j+2}^{(n)} \end{bmatrix} \tag{6.2.17}$$

gives, $$u_{\ell-1,j}^{(n+\frac{1}{2})} = \frac{1}{\det}[w_{\ell-1,j}-a_m(a_p u_{\ell-1,j}^{(n)}-f u_{\ell-1,j+1}^{(n)})-f\{a_p(u_{\ell-2,j}^{(n)}+ u_{\ell-1,j-1}^{(n)})-f(u_{\ell-2,j+1}^{(n)}+u_{\ell-1,j+2}^{(n)})\}] , \tag{6.2.18a}$$

$$u^{(n+\frac{1}{2})}_{\ell-1,j+1} = \frac{1}{\det}[w_{\ell-1,j+1} - a_m(-fu^{(n)}_{\ell-1,j} + a_p u^{(n)}_{\ell-1,j+1}) - f\{-f(u^{(n)}_{\ell-2,j} + u^{(n)}_{\ell-1,j-1}) + a_p(u^{(n)}_{\ell-2,j+1} + u^{(n)}_{\ell-1,j+2})\}] \quad . \tag{6.2.18b}$$

3. Bottom side:

$$\begin{bmatrix} a_p & f \\ f & a_p \end{bmatrix} \begin{bmatrix} u_{i,1} \\ u_{i+1,1} \end{bmatrix}^{(n+\frac{1}{2})} = \begin{bmatrix} b_{i,1} \\ b_{i+1,1} \end{bmatrix} - a_m \begin{bmatrix} u_{i,1} \\ u_{i+1,1} \end{bmatrix}^{(n)} - f \begin{bmatrix} u^{(n)}_{i-1,1} + u^{(n)}_{i,2} \\ u^{(n)}_{i+1,2} + u^{(n)}_{i+2,1} \end{bmatrix} \tag{6.2.19}$$

gives,

$$u^{(n+\frac{1}{2})}_{i,1} = \frac{1}{\det}[w_{i,1} - a_m(a_p u^{(n)}_{i,1} - fu^{(n)}_{i+1,j}) - f\{a_p(u^{(n)}_{i-1,1} + u^{(n)}_{i,2}) - f(u^{(n)}_{i+1,2} + u^{(n)}_{i+2,1})\}] \quad , \tag{6.2.20a}$$

$$u^{(n+\frac{1}{2})}_{i+1,1} = \frac{1}{\det}[w_{i+1,1} - a_m(-fu^{(n)}_{i,1} + a_p u^{(n)}_{i+1,1}) - f\{-f(u^{(n)}_{i-1,1} + u^{(n)}_{i,2}) + a_p(u^{(n)}_{i+1,2} + u^{(n)}_{i+2,1})\}] \quad . \tag{6.2.20b}$$

4. Top side:

$$\begin{bmatrix} a_p & f \\ f & a_p \end{bmatrix} \begin{bmatrix} u_{i,\hat{m}-1} \\ u_{i+1,\hat{m}-1} \end{bmatrix}^{(n+\frac{1}{2})} = \begin{bmatrix} b_{i,\hat{m}-1} \\ b_{i-1,\hat{m}-1} \end{bmatrix} - a_m \begin{bmatrix} u_{i,\hat{m}-1} \\ u_{i+1,\hat{m}-1} \end{bmatrix}^{(n)} - f \begin{bmatrix} u^{(n)}_{i,\hat{m}-2} + u^{(n)}_{i-1,\hat{m}-1} \\ u^{(n)}_{i+1,\hat{m}-2} + u^{(n)}_{i+2,\hat{m}-1} \end{bmatrix} \tag{6.2.21}$$

gives,

$$u^{(n+\frac{1}{2})}_{i,\hat{m}-1} = \frac{1}{\det}[w_{i,\hat{m}-1} - a_m(a_p u^{(n)}_{i,\hat{m}-1} - fu^{(n)}_{i+1,\hat{m}-1}) - f\{a_p(u^{(n)}_{i,\hat{m}-2} + u^{(n)}_{i-1,\hat{m}-1}) - f(u^{(n)}_{i+1,\hat{m}-2} + u^{(n)}_{i+2,\hat{m}-1})\}] \quad , \tag{6.2.22a}$$

$$u^{(n+\frac{1}{2})}_{i+1,\hat{m}-1} = \frac{1}{\det}[w_{i+1,\hat{m}-1} - a_m(-fu^{(n)}_{i,\hat{m}-1} + a_p u^{(n)}_{i+1,\hat{m}-1}) - f\{-f(u^{(n)}_{i,\hat{m}-2} + u^{(n)}_{i-1,\hat{m}-1}) + a_p(u^{(n)}_{i+1,\hat{m}-2} + u^{(n)}_{i+2,\hat{m}-1})\}] \quad , \tag{6.2.22b}$$

where,

$$w_{1,j} = a_p b_{1,j} - f b_{1,j+1}, \quad w_{\ell-1,j} = a_p b_{\ell-1,j} - f b_{\ell-1,j+1},$$

$$w_{i,1} = a_p b_{i,1} - f b_{i+1,1}, \quad w_{i,\hat{m}-1} = a_p b_{i,\hat{m}-1} - f b_{i+1,\hat{m}-1},$$

and $$\det = a_p^2 - f^2.$$

The molecular diagrams of (6.2.18, 19, 20 and 21) are shown in Figure (6.2.6a.b.c and d) respectively.

For the single corner points we have:

1. Bottom left:

$$u_{1,1}^{(n+\frac{1}{2})} = [b_{1,1} - a_m u_{11}^{(n)} - f(u_{1,2}^{(n)} + u_{2,1}^{(n)})]/a_p , \tag{6.2.23}$$

2. Bottom right:

$$u_{\ell-1,1}^{(n+\frac{1}{2})} = [b_{\ell-1,1} - a_m u_{\ell-1,1}^{(n)} - f(u_{\ell-2,1}^{(n)} + u_{\ell-1,2}^{(n)})]/a_p , \tag{6.2.24}$$

3. Top left:

$$u_{1,\hat{m}-1}^{(n+\frac{1}{2})} = [b_{1,\hat{m}-1} - a_m u_{1,\hat{m}-1}^{(n)} - f(u_{1,\hat{m}-2}^{(n)} + u_{2,\hat{m}-1}^{(n)})]/a_p , \tag{6.2.25}$$

and

4. Top right:

$$u_{\ell-1,\hat{m}-1}^{(n+\frac{1}{2})} = [b_{\ell-1,\hat{m}-1} - a_m u_{\ell-1,\hat{m}-1}^{(n)} - f(u_{\ell-2,\hat{m}-1}^{(n)} + u_{\ell-1,\hat{m}-2}^{(n)})]/a_p \tag{6.2.26}$$

The molecular diagrams of (6.2.23, 24, 25 and 26) are shown in Figure (6.2.7a,b,c and d) respectively.

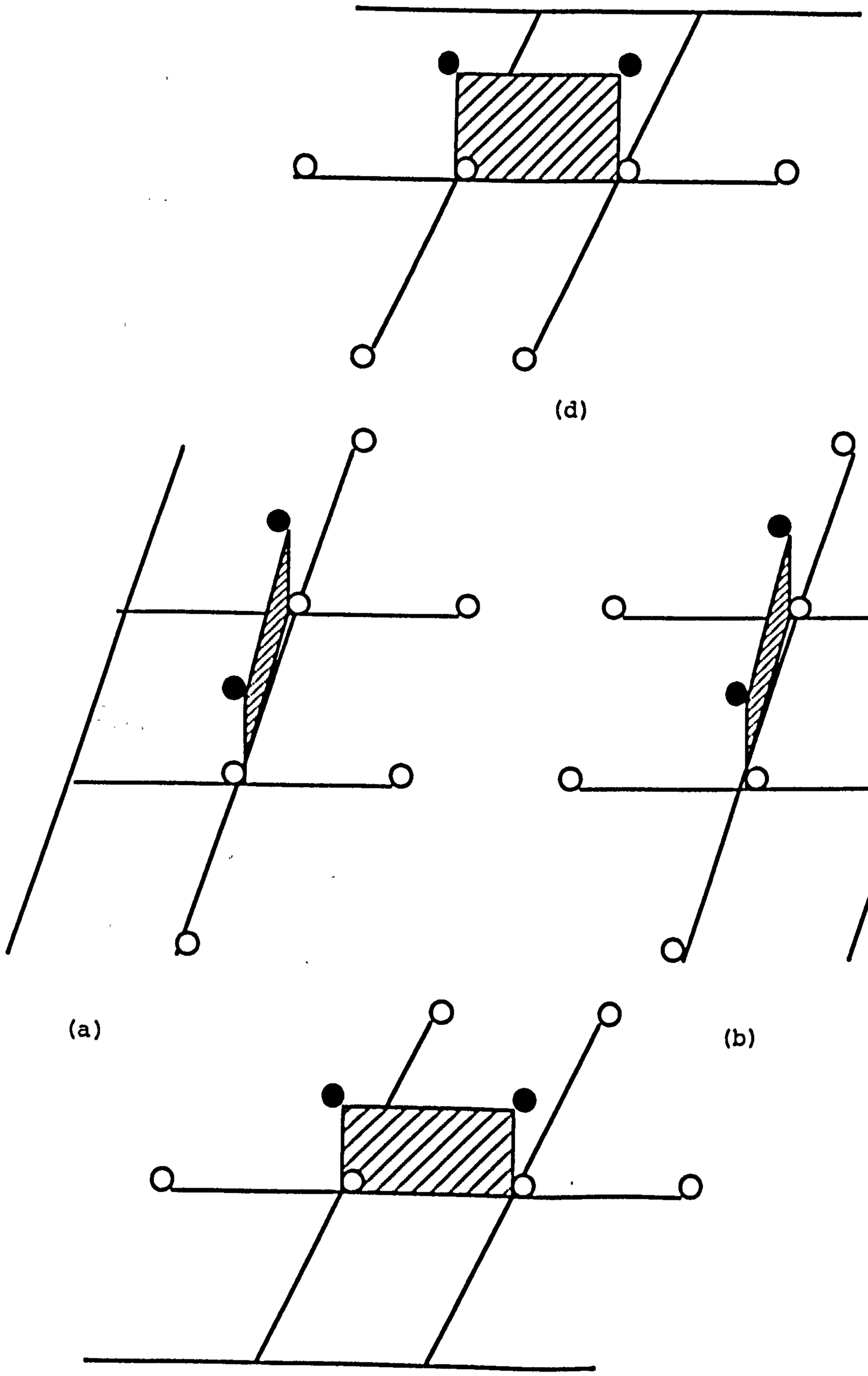
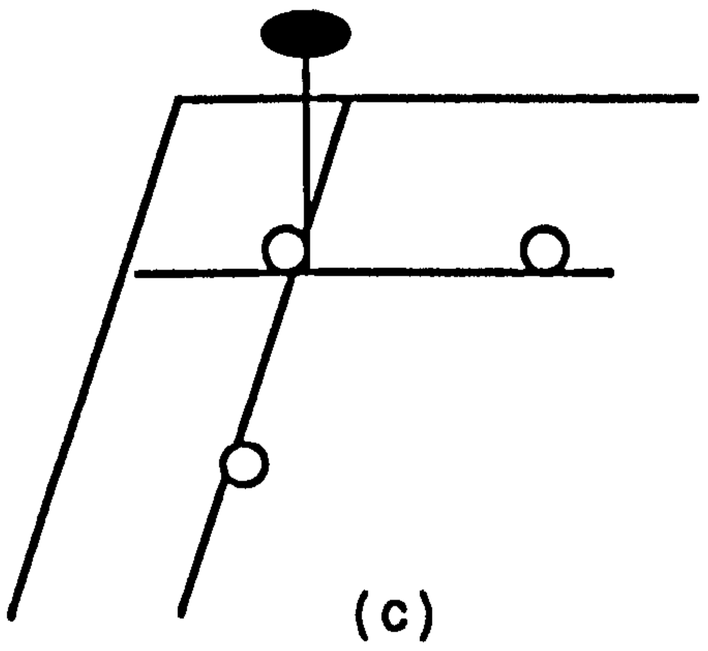
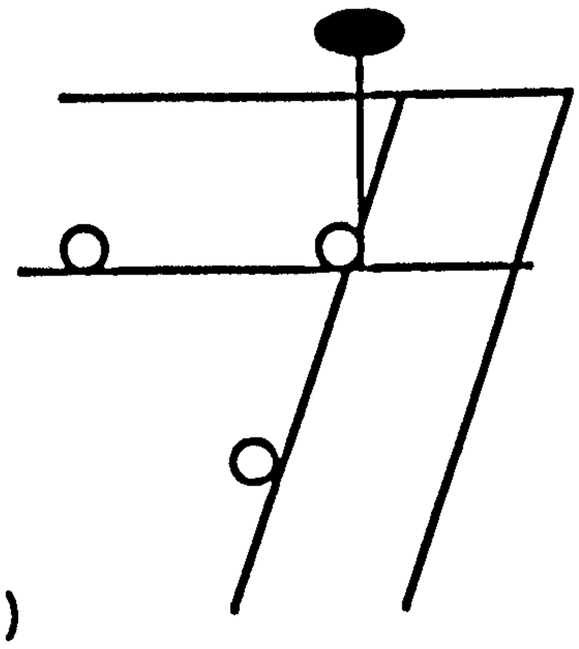


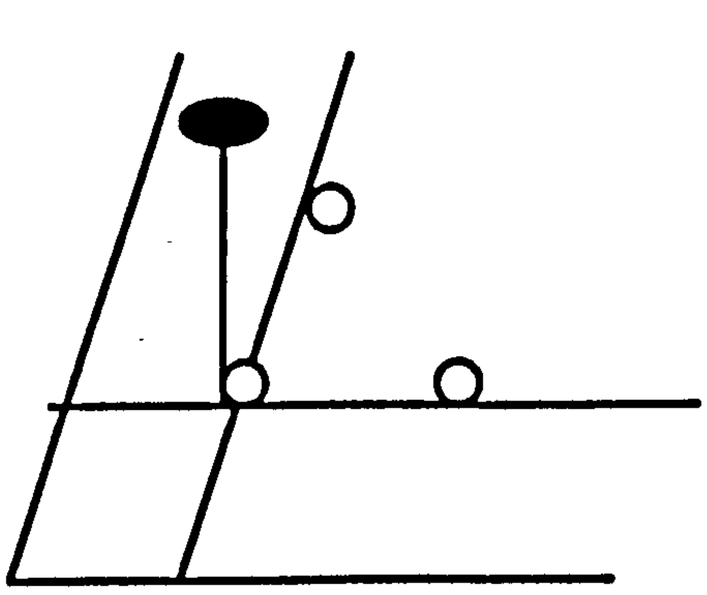
FIGURE 6.2.6



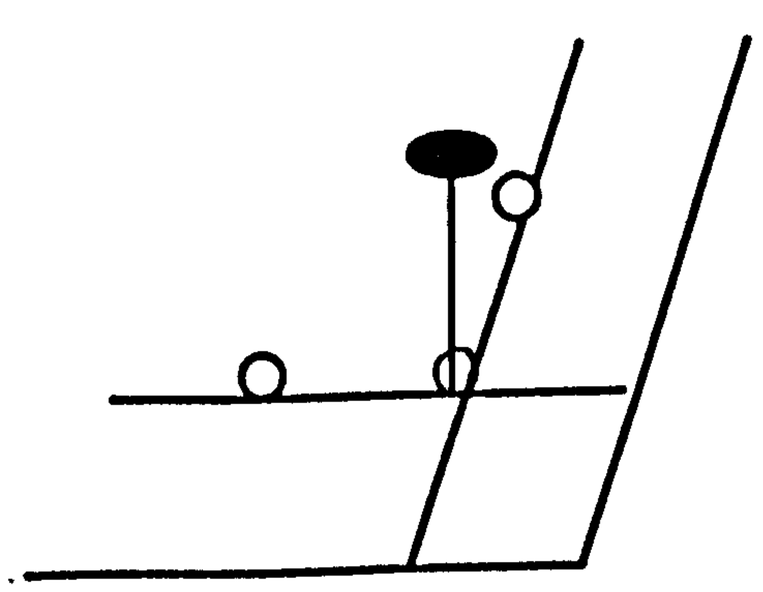
(c)



(d)



(a)



(b)

FIGURE 6.2.7

6.3 THE EIGENVALUES OF THE ITERATIVE MATRIX OF SYSTEM (6.2.10)

Since we rearranged the linear system in the way we wanted, we can now easily obtain the eigenvalues of the l.h.s. coefficient matrix where we have blocks of 4×4, 2×2 and single elements.

For the 4×4 blocks we easily obtain the eigenvalues

12r-1, 1+6r, 1+6r and 3.

For the 2×2 blocks we get

9r and 2+3r,

and the single element which is 1+6r.

Ignoring the multiplicity of the eigenvalues, we record only 5 eigenvalues

$$\lambda_1 = 12r-1,\ \lambda_2 = 9r,\ \lambda_3 = 1+6r,\ \lambda_4 = 2+3r \text{ and } \lambda_5 = 3.$$

It is not surprising to find that all these eigenvalues coincide when r=1/3 to give the value 3, which as in the previous chapter leads us to the optimum value of the parameter which gives the minimum number of iterations.

We also notice that the eigenvalue 12r-1 has a critical value when r=1/12 which leads to the value λ_1=0 and thus a singular system is produced. In fact, this affects the solution when r is nearly 1/12, where it leads to a weakly stable system.

In Table 6.3.1 we present some numerical values of the eigenvalues for different values of r and these are shown in Figure (6.3.1).

	λ_1	λ_2	λ_3	λ_4	λ_5
r	12r-1	9r	1+6r	2+3r	3
0	-1	0	1	2	3
1/20	-2/5	0.45	1.3	2.15	"
1/12	0	0.75	1.5	2.25	"
1/10	0.2	0.9	1.6	2.3	"
1/6	1.0	1.5	2.0	2.5	"
1/3	3.0	3.0	3.0	3.0	"
1/2	5.0	4.5	4.0	3.5	"
1.0	11.0	9.0	7.0	5.0	"

TABLE 6.3.1

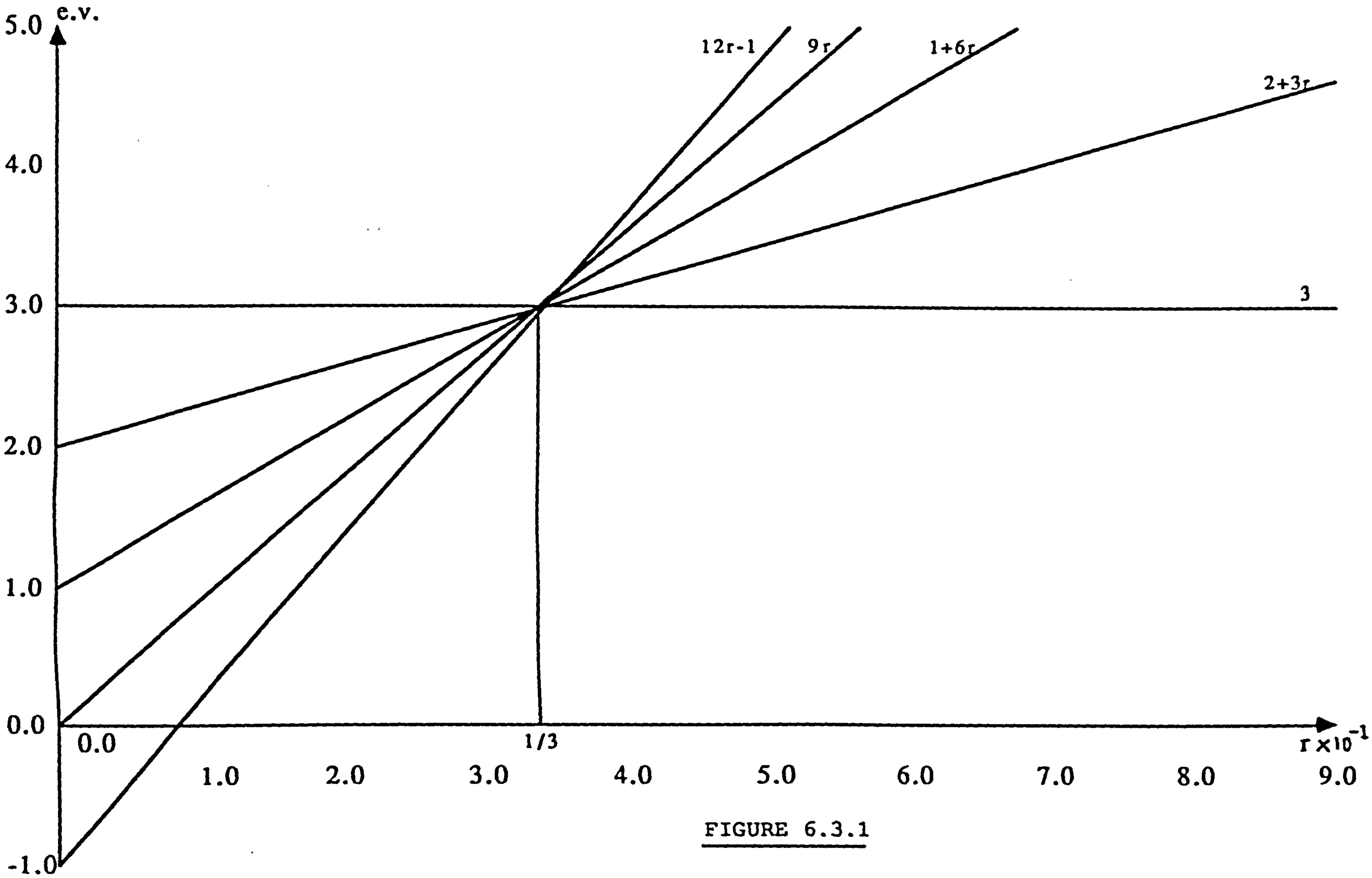

5.0
e.v.
4.0
3.0
2.0
1.0
0.0
-1.0
12r-1
9r
1+6r
2+3r
3
0.0
1/3
r x10^-1
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0

FIGURE 6.3.1

6.4 TRUNCATION ERROR, CONVERGENCE AND STABILITY CONSIDERATIONS

To determine the truncation error which was estimated earlier we rewrite equation (6.2.3) as,

$$(2+12r)u_{i,j,k+1}+(1-3r)\{u_{i-1,j,k+1}+u_{i+1,j,k+1}+u_{i,j-1,k+1}+u_{i,j+1,k+1}\}-$$
$$(2-12r)u_{i,j,k}+(1+3r)\{u_{i-1,j,k}+u_{i+1,j,k}+u_{i,j-1,k}+u_{i,j+1,k}\} = 0 \qquad (6.4.1)$$

Using Taylor's series expansion about the point $(i,j,k+\frac{1}{2})$, we get the term,

$$T.E. = 6\Delta t\, u_t - 6\Delta t(u_{xx}+u_{yy}) + \frac{(\Delta t)^3}{4}u_{ttt} + \Delta t(\Delta x)^2(u_{xxt}+u_{yyt})$$
$$-\frac{3}{4}(\Delta t)^3(u_{xxtt}+u_{yytt}) - \frac{\Delta t}{2}(\Delta x)^2(u_{xxxx}+u_{yyyy}) +$$
$$O((\Delta t)^{\alpha_1} + (\Delta x)^{\alpha_2}) , \qquad (6.4.2)$$

where $\alpha_1+\alpha_2=5$.

The first two terms cancel out since they satisfy the partial differential equation (6.1.1), the principal part of the truncation error is given by $O((\Delta t)^3+\Delta t(\Delta x)^2)$.

To establish the convergence of the method we proceed as previously in an earlier chapter. Thus we have,

$$(A_1+\rho I)u^{(n+\frac{1}{2})} = (A_2-\rho I)u^{(n)} + b , \qquad (6.4.3a)$$

$$(A_2+\rho I)u^{(n+1)} = (A_1-\rho I)u^{(n+\frac{1}{2})} + b . \qquad (6.4.3b)$$

From (6.4.3a) we have,

$$u^{(n+\frac{1}{2})} = (A_1+\rho I)^{-1}(A_2-\rho I)u^{(n)} + c \qquad (6.4.4)$$

where $c = (A_1+\rho I)^{-1}b$, and from (6.4.3b), by substituting $u^{(n+\frac{1}{2})}$ in (6.4.4) we get,

$$u^{(n+1)} = Tu^{(n)} + f + d \qquad (6.4.5)$$

where T is the iterative matrix and is written as,

$$(A_2+\rho I)^{-1}(A_1-\rho I)(A_1+\rho)^{-1}(A_2-\rho I) ,$$

$$f = (A_2+\rho I)^{-1}(A_1+\rho I)^{-1}b ,$$

and $$d = (A_2+\rho I)^{-1}b .$$

For convergence we need that,

$$||T||_2 \leq 1 \qquad (6.4.6)$$

i.e.,

$$||(A_2+\rho I)^{-1}(A_1-\rho I)(A_1+\rho I)^{-1}(A_2-\rho I)||_2 \leq 1.$$

By using the similarity transformation we can write,

$$\tilde{T} = (A_2-\rho I)T(A_2-\rho I)^{-1} , \qquad (6.4.7)$$

i.e.,

$$\tilde{T} = (A_2-\rho I)(A_2+\rho I)^{-1}(A_1-\rho I)(A_1+\rho I)^{-1} , \qquad (6.4.8)$$

which has the same eigenvalues as T. Thus,

$$||\tilde{T}||_2 = ||(A_2-\rho I)(A_2+\rho I)^{-1}(A_1-\rho I)(A_1+\rho I)^{-1}||_2 \qquad (6.4.9)$$

i.e.,

$$\rho(\tilde{T}) \leq \max_{\lambda_1,\lambda_2} ||\frac{(\lambda_2-\rho)}{(\lambda_2+\rho)}\frac{(\lambda_1-\rho)}{(\lambda_1+\rho)}|| , \qquad (6.4.10)$$

where λ_1 and λ_2 represent the eigenvalues of A_1 and A_2 respectively.

Since A_1 and A_2 have the same eigenvalues, therefore, we have,

$$\rho(\tilde{T}) = \max_{\lambda}||\frac{(\lambda-\rho)}{(\lambda+\rho)}||^2 \qquad (6.4.11)$$

or

$$\rho(\tilde{T}) \leq 1 \text{ for all } \lambda>0 . \qquad (6.4.12)$$

Thus convergence immediately follows.

This result establishes the stability of the method for r>1/12.

6.5 NUMERICAL EXPERIMENTS

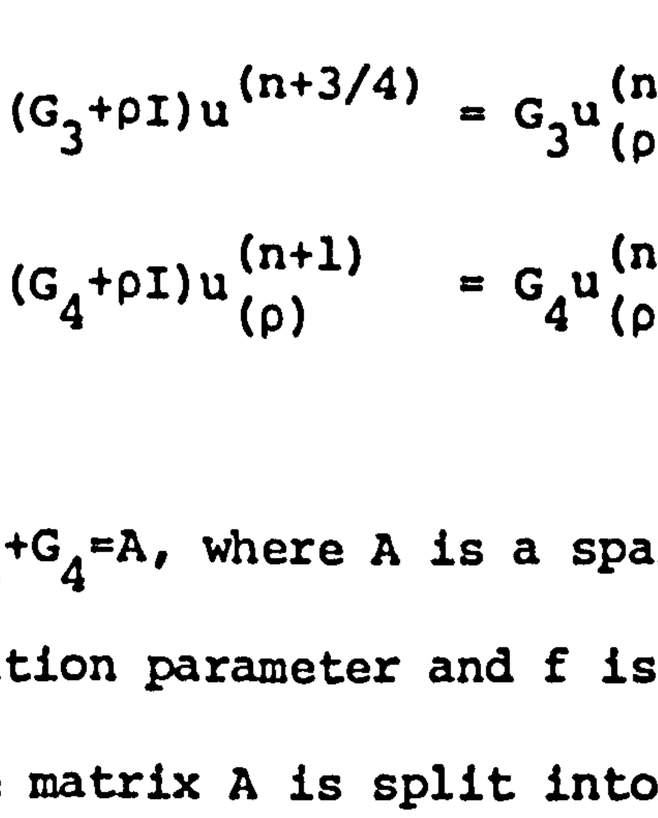
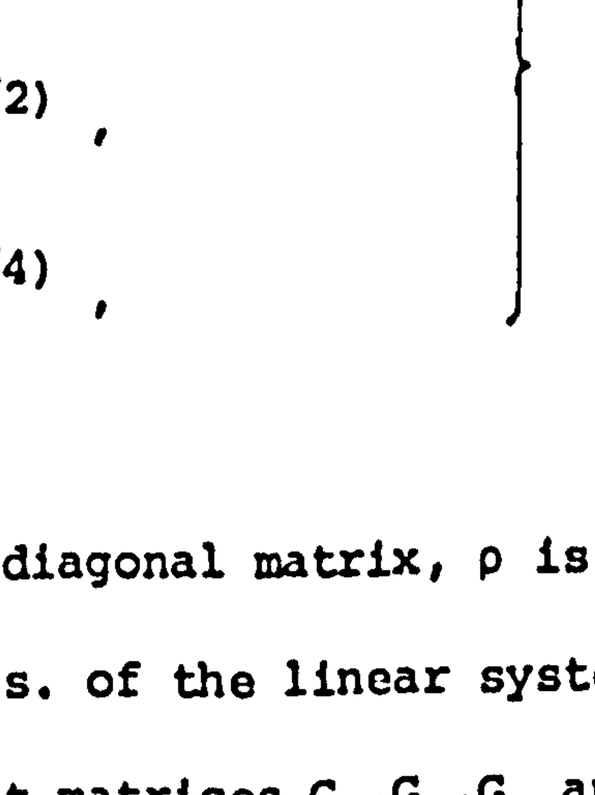
Before we test the SPAGIE method for two dimensional problems, we mention here the Alternating Group Explicit (AGE) Iterative Method for solving multi-dimensional parabolic problems which was introduced by SAHIMI, M.S. (1987). This method, which is the Douglas-Rachford formula for the AGE scheme, is written as,

$$\left.\begin{aligned}
(G_1+\rho I)u^{(n+1/4)} &= (\rho I-G_1-2G_2-2G_3-2G_4)u^{(n)}_{(\rho)}+2f\ ,\\
(G_2+\rho I)u^{(n+1/2)} &= G_2u^{(n)}_{(\rho)}+\rho u^{(n+1/4)}_{(\rho)}\ ,\\
(G_3+\rho I)u^{(n+3/4)} &= G_3u^{(n)}_{(\rho)}+\rho u^{(n+1/2)}_{(\rho)}\ ,\\
(G_4+\rho I)u^{(n+1)}_{(\rho)} &= G_4u^{(n)}_{(\rho)}+\rho u^{(n+3/4)}_{(\rho)}\ ,
\end{aligned}\right\} \qquad (6.5.1)$$

with,

$G_1+G_2+G_3+G_4=A$, where A is a sparse quin-diagonal matrix, ρ is an acceleration parameter and f is the r.h.s. of the linear system.

The matrix A is split into component matrices G_1, G_2, G_3 and G_4 such that

$G_1+G_2 =$  , $G_3+G_4 =$ 

with $\mathrm{diag}(G_1+G_2)=\mathrm{diag}(G_3+G_4) = \frac{1}{2}\mathrm{diag}(A)$.

The AGE scheme corresponds to sweeping through the mesh parallel to the coordinate x and y axes involving at each stage the solution of

2×2 block systems. The iterative procedure is continued until convergence is reached [Sahimi, M.S. 1987].

Example 1

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + g(x,y,t) \ , \qquad (6.5.2)$$

where,

$$g(x,y,t) = \sin(x)\sin(y)e^{-t} - 4.0, \ 0 \leq x,y \leq 1.$$

The initial and boundary conditions satisfy the exact solution which is given as,

$$U(x,y,t) = \sin(x)\sin(y)e^{-t} + x^2 + y^2 \ . \qquad (6.5.3)$$

Example 2

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + g(x,y,t) \ , \qquad (6.5.4)$$

where,

$$g(x,y,t) = \sin(x)\ \frac{e^{-t}}{1+y^2} - 2x - 6xy, \ 0 \leq x,y \leq 1,$$

with the initial conditions

$$U(x,y,0) = \sin(x)\log(1+y) + x^3 y + xy^2 \ , \ 0 \leq x,y \leq 1,$$

and the boundary conditions satisfy the exact solution which is,

$$U(x,y,t) = \sin(x)\log(1+y)e^{-t} + x^3 y + xy^2 \ . \qquad (6.5.5)$$

In all cases we used the optimum acceleration parameter of Peaceman and Rachford which is $\sqrt{3\times(12r-1)}$, and the convergence criterion is $\varepsilon = 5\times10^{-6}$.

Experiment 1

In this experiment we tested the method on Example (1) for different values of r. In Table (2) we present the results for r=1/3,

1 and 2, and in Table (3) r=4,6 and 10. In Table (4) we compare the SPAGEI method with AGE iterative method of SAHIMI (1987) and the direct GE method of ABDULLAH, A.B. (1983).

Experiment 2

For the same number of time steps we solve Example (2) for values of r=1,2,6,10 and 20. The results are shown in Table (5).

$x=0.5$
$\Delta y=0.1$

$\varepsilon=5\times10^{-6}$

Δt	r	t	Δy \ x	1	2	3	4	5	6	7	8	9	No.of Iters
0.003	0.33	≅0.1	P.E.	0.0183	0.0387	0.0463	0.0628	0.0615	0.0651	0.0504	0.0383	0.0170	60
			E.S.	0.303312	0.376192	0.468210	0.578947	0.707997	0.854968	1.019491	1.201222	1.399843	
0.01	1.0	0.1	P.E.	0.0017	0.0027	0.0031	0.0032	0.0031	0.0027	0.0022	0.0016	0.0009	40
			E.S.	0.303308	0.376183	0.468197	0.578931	0.707976	0.854943	1.019463	1.201191	1.399809	
0.02	2.0	1.2	P.E.	0.0006	0.0013	0.0012	0.0016	0.0013	0.0013	0.0009	0.0007	0.0003	294
			E.S.	0.274416	0.318688	0.382673	0.466232	0.569229	0.691534	0.833025	0.993586	1.173113	

TABLE 2: The Percentage Error (P.E.) of the numerical solutions and the Exact Solution (E.S.) of Example 1

x=0.5
Δ y=0.1

$\varepsilon=5\times10^{-6}$

Δt	r	t	Δy \ x	1	2	3	4	5	6	7	8	9	No.of Iters
0.04	4.0	4.0	P.E.	85×10^{-6}	322×10^{-6}	154×10^{-6}	298×10^{-6}	87×10^{-6}	164×10^{-6}	68×10^{-6}	112×10^{-6}	24×10^{-6}	554
			E.S.	0.260877	0.291745	0.342595	0.413419	0.504210	0.614958	0.745657	0.89699	1.066878	
0.06	6.0	3.0	P.E.	132×10^{-6}	316×10^{-6}	393×10^{-6}	420×10^{-6}	451×10^{-6}	374×10^{-6}	265×10^{-6}	183×10^{-6}	122×10^{-6}	374
			E.S.	0.262383	0.294742	0.347054	0.419295	0.511443	0.623478	0.755377	0.907123	1.078697	
0.1	10.0	2.0	P.E.	300×10^{-6}	1405×10^{-6}	1666×10^{-6}	1748×10^{-6}	1658×10^{-6}	1431×10^{-6}	1116×10^{-6}	811×10^{-6}	388×10^{-6}	220
			E.S.	0.266478	0.302890	0.359174	0.435267	0.531107	0.646636	0.781799	0.936544	1.110825	

TABLE 3: The Percentage Error (P.E.) of the numerical solution and the Exact Solution (E.S.) of Example (1)

$x=0.5$ $\Delta t=0.01$ $t=0.5$
$r=1.0$ $\Delta y=0.1$

Method \ Δy \ x	1	2	3	4	5	6	7	8	9
Double Alternating GE	1.16×10^{-4}	4.4×10^{-5}	7.3×10^{-5}	7.6×10^{-5}	2.6×10^{-5}	1.16×10^{-4}	1.1×10^{-5}	1.28×10^{-4}	3.2×10^{-5}
AGE-CN $\varepsilon=10^{-4}$	1.56×10^{-5}	3.37×10^{-5}	4.33×10^{-5}	5.66×10^{-5}	5.88×10^{-5}	6.24×10^{-5}	5.45×10^{-5}	4.21×10^{-5}	2.61×10^{-5}
AGE-CN $\varepsilon=10^{-8}$	4.26×10^{-6}	8.33×10^{-6}	1.2×10^{-5}	1.5×10^{-5}	1.71×10^{-5}	1.8×10^{-5}	1.72×10^{-5}	1.44×10^{-5}	8.86×10^{-6}
SPAGEI $\varepsilon=10^{-6}$	3.6×10^{-6}	9.6×10^{-6}	1.0×10^{-5}	1.7×10^{-5}	1.5×10^{-5}	2.0×10^{-5}	1.5×10^{-5}	1.6×10^{-5}	7.9×10^{-6}
Exact Sol.	0.289030	0.347770	0.425933	0.523238	0.639410	0.774190	0.927330	1.098597	1.287781

TABLE 4: The absolute error of the numerical solution of Example (1)

x=0.5
Δy=0.1

No. of time steps = 50

$\epsilon = 5 \times 10^{-6}$

r	t	Δy / x	1	2	3	4	5	6	7	8	9	No.of Iters
1	0.5	P.E.	0.108534	0.098623	0.057923	0.050093	0.031482	0.024509	0.015068	0.010052	0.004244	200
		E.S.	0.45215	0.098017	0.158792	0.227842	0.305404	0.391671	0.486799	0.590920	0.704142	
2	1.0	P.E.	0.143295	0.101497	0.071256	0.050710	0.035527	0.024652	0.016277	0.009850	0.004498	246
		E.S.	0.034310	0.077156	0.128773	0.189344	0.259012	0.337895	0.426087	0.523668	0.630704	
6	3.0	P.E.	0.078806	0.049899	0.033193	0.022783	0.014835	0.009967	0.006722	0.003944	0.001686	365
		E.S.	0.019775	0.049352	0.088762	0.138031	0.197178	0.266219	0.345166	0.434030	0.532821	

continued...

r	t	Δy \ x	1	2	3	4	5	6	7	8	9	No.of Iters
10	5.0	P.E.	0.015728	0.011312	0.007777	0.005816	0.002787	0.001873	0.001799	0.000976	0.000285	444
		E.S.	0.017808	0.045589	0.083348	0.131087	0.188810	0.256518	0.334214	0.421899	0.519573	
20	10.0	P.E.	0.065697	0.004224	0.000501	0.000798	0.001382	0.001331	0.001500	0.000700	0.000055	518
		E.S.	0.017502	0.045004	0.082506	0.130007	0.187509	0.255010	0.332512	0.420013	0.517514	

TABLE 5: The Percentage Error (P.E.) of the numerical solution and the Exact Solution (E.S.) of Example (2)

r	t	Method	Average No.of Iter. Per Step
0.33	0.1	SPAGEI	2
1.0	0.1	SPAGEI	4
	0.5	SPAGEI	$\varepsilon=10^{-6}$ 4
	0.5	AGE-CN	$\varepsilon=10^{-4}$ 4 $\varepsilon=10^{-8}$ 11
	1.2	AGE-CN	$\varepsilon=10^{-4}$ 3 $\varepsilon=10^{-8}$ 11
2.0	1.0	SPAGEI	5
	1.2	SPAGEI	5
4.0	4.0	SPAGEI	5.5
6.0	3.0	SPAGEI	7.4
10.0	2.0	SPAGEI	11
	5.0	SPAGEI	9
20.0	10.0	SPAGEI	10.4

TABLE 6

6.6 REMARKS

a) From Table (2) we notice that the minimum number of iterations can still be obtained by using the optimum acceleration parameter (3). However, it is observed that we get better results by using a greater value of r. In the second line where r=1.0, the percentage errors are reduced by a factor of more than (10^{-1}) as well as a reduction in the overall number of iterations for the same solution.

b) In Tables (3) and (5) we observe the stability of the method for even greater values of r. Although the number of iterations per time step increases with r, it is easily compensated by applying a smaller number of time steps. This is noticeable in Table (5) where for r=4 it takes (5.5×100) iterations to get the solution at time (4.0) while for r=10 it takes (9×50) iterations to get the solution at time (5.0).

c) The superiority of the SPAGEI method over the AGE iterative method and the direct GE method is clearly seen in Table (4) where in the results there is a factor of about 10^{-1} difference in accuracy between the GE and the SPAGEI. For the AGE method with criterion $\varepsilon=10^{-4}$, the absolute error is about (3) times as the absolute error of the SPAGEI method, while with $\varepsilon=10^{-8}$ it is almost the same as for the SPAGEI.

From Table (6) we see that for these particular values of r,t and ε the number of iterations required for the AGE method with $\varepsilon=10^{-8}$ is (11) per time step while it is only (4) for the SPAGEI to get the same solution with similar, if not better, accuracy.

d) To exhibit the efficiency of the SPAGEI method we compare in Table (6) the number of iterations required for the solution at time (1.2). For the AGE method and a less accurate solution with $\epsilon=10^{-4}$ it takes (120×3) iterations to obtain a solution comparable with that of the SPAGEI method and when $\epsilon=10^{-8}$, it takes (120×11) iterations, while it takes only (60×5) iterations for the SPAGEI method to reach the required solution with r=2.

CHAPTER SEVEN

CONCLUSION AND SUGGESTIONS FOR FURTHER RESEARCH

In the first three chapters some introductory mathematical background was presented, as well as some definitions and advanced theories in numerical analysis and finite differences. Then an introduction to some approximations to differential operators by cubic splines was introduced and from the cubic splines formula (3.2.12) some new finite difference schemes were derived.

In Chapter 4 the direct SPAGE method proved to be compatible with other methods of the same order of truncation error like the Group Explicit (G.E.) method. Also it was found that some schemes of the SPAGE method are superior to their corresponding counterparts in the G.E. class of method, and in general the SPAGE method gives its best results when $0.4 \leq \theta \leq 0.7$.

In Chapter 5 the iterative method SPAGEI was introduced and was shown to be stable and highly accurate solutions to the problems considered were obtained. Test for validity of the Peaceman Rachford and Wachspress parameters proved positive. In addition, the eigenvalues of the constituent matrices were used as acceleration parameters. The new method was shown to have an easy form such that an optimum parameter was easily introduced. Further, the amount of computational work saved by using more than one parameter, especially in problems with periodic boundary conditions, is so encouraging that further research is required to establish guidance for the optimum number of parameters required with the appropriate value of r.

In Chapter 6 the SPAGEI method was extended to two dimensional problems and proved to be extremely stable and accurate for relatively very large values of r. The optimum parameter used in the one-dimensional problem is valid too for the two-dimensional problems.

For further work, we mentioned in Chapter 4 a way of treating the single (ungrouped) points. Numerical results are required to show the advantage of this technique. Also in Chapter 4 a three time level scheme was suggested which needs further analysis and numerical experiments.

An extension of the SPAGE method to two-dimensional problems could also be considered for further research too. The suggestion made at the end of Chapter 5 for the SPAGEI method would be equivalent to applying the Gauss-Seidel method instead of the Jacobi method. In the case of two-dimensional problems, the use of more than one parameter did not appear to show any advantage in the experiments. Whether this would be different by increasing the number of mesh points or not is also left for further study.

In general, the choice of the cubic splines equation as a base for the schemes derived in this thesis proved to be an important advantage for time dependent problems. It provides the optimum parameter with probably the highest value of r, and consequently Δt, to be used amongst all the schemes considered.

It also allows an integer value of ρ to be chosen for steady state problems.

REFERENCES

ABDULLAH, A.R.B., Ph.D., 1983, *The Study of Some Numerical Methods for Solving Parabolic Partial Differential Equations.*

AHLBERG, J.H., NILSON, E.N. and WALSH, J.L., 1967, *The Theory of Splines and Their Applications*, Academic Press.

AMES, W.F., 1977, *Numerical Methods for Partial Differential Equations*, Academic Press.

BIRKHOFF, G., VARGA, R.S. and YOUNG, D., 1962, *Alternating Direction Implicit Methods*, Advances in Computers, Vol. 3, Academic Press, New York, London.

CRANDALL, S.H., 1955, *An Optimum Implicit Recurrence Formula for the Heat Conduction Equation*, Quart.Applc.Math., 13, No. 3, 318-320.

CHOW, Chuen-Yen, 1979, *An Introduction to Computational Fluid Mechanics*, John Wiley, Toronto.

CONTE, S.D. and DE BOOR, C., 1980, *Elementary Numerical Analysis, An Algorithmic Approach*, McGraw-Hill, Kogakusha, Ltd.

DAHLQUIST, G. and BJORCK, A., 1974, *Numerical Methods*, Prentice Hall, Inc.

DANAEE, A., Ph.D., 1980, *A Study of Hopscotch Methods for Solving Parabolic Partial Differential Equations*, Loughborough University.

DANIEL, J.W. and SWARTZ, B.K., 1975, *Extrapolated Collocation for Two-Point Boundary-Value Problems using Cubic Splines*, J.Inst. Maths.Applics., 16, 161-174.

DOUGLAS, J. and RACHFORD, H.H., 1956, *On the Numerical Solution of Heat Conduction Problems in Two or Three Space Variables*, Trans.Amer.Math.Soc., 82, 421-439.

DOUGLAS, J., 1961, *A Survey of Numerical Methods for Parabolic Differential Equations*, Advances in Computers, Vol. 2, 1-54.

DU-FORT, E.C. and FRANKEL, S.P., 1953, *Stability Condition in the Numerical Treatment of Parabolic Differential Equations*, Math. Tables and Other Aids to Computation, Vol. 7, p.135.

FADEEV, D.F. and FADEEVA, V.N., *Computational Methods of Linear Algebra*, W.H. Freeman and Company.

FORSYTHE, G.E. and WASOW, W.R., 1960, *Finite Difference Methods for Partial Differential Equations*, Wiley, New York.

FOX, L., 1962, *Numerical Solution Of Ordinary and Partial Differential Equations*, Pergamon Press, Oxford.

FROBERG, C.-E., 1979, *Introduction to Numerical Analysis*, Addison-Wesley.

GERALD, C.F., 1978, *Applied Numerical Analysis*, Addison-Wesley Publishing Company.

GORDON, P., 1965, *Non-Symmetric Difference Equations*, J.Soc.Indust. Appl.Math., 13, No. 3, 667-672.

GOURLAY, A.R., 1970, *Hopscotch: A Fast Second-Order Partial Differential Equation Solver*, J.Inst.Maths.Applics., 6, 375-390.

HILDBRAND, F.B., 1968, *Finite Difference Equations and Simulations*, Prentice-Hall, Inc.

HILDEBRAND, F.B., 1974, *Introduction to Numerical Analysis*, Tata McGraw-Hill, New Delhi.

ISAACSON, E. and KELLER, H.B., 1966, *Analysis of Numerical Methods*, John Wiley & Sons.

JAIN, M.K., 1979, *Numerical Solution of Differential Equations*, Wiley Eastern Ltd., 1979.

KELLOGG, R.B., 1964, *An Alternating Direction Method for Operator Equations*, J.Soc.Indust.Appl.Math., Vol. 12, No. 4.

NIELSON, K.L., 1964, *Methods in Numerical Analysis*, MacMillan, London.

MIKHLIN, S.G., 1967, *Linear Equations of Mathematical Physics*, Holt, Rinehart and Winston, Inc.

MILNE, W.E., 1953, *Numerical Solution of Differential Equations*, J. Wiley.

MITCHELL, A.R., 1969, *Computational Methods in Partial Differential Equations*, John Wiley & Sons.

MITCHELL, A.R. and FAIRWEATHER, G., 1964, *Improved Forms of the Alternating Direction Method of Douglas, Peaceman and Rachford for Solving Parabolic and Elliptic Equations*, Numerische Mathematik, 6, 285-292.

MITCHELL, A.R. and WAIT, R., 1977, *The Finite Element Method in Partial Differential Equations*, John Wiley, N.Y.

ORTEGA, J.M. and RHEINBOLDT, W.C., 1970, *Iterative Solution of Non-linear Equations in Several Variables*, Academic Press.

OSTROWSKI, A.M., 1966, *Solution of Equations and Systems of Equations*, Academic Press, N.Y.

PANOW, D.Y., 1955, *Manual for the Numerical Solution of Differential Equations with Partial Derivatives*, Akademie-Verlag, Berlin.

PAPAMICHAEL, N. and J.R. WHITEMAN, 1973, *A Cubic Spline Technique for the One Dimensional Heat Conduction Equation*, J.Inst.Maths. Applics. 11, 111-113.

PAPAMICHAEL, N. and WORSEY, A.J., 1981, *End Conditions for Improved Cubic Spline Derivative Approximations*, J.Com.Appl.Math., Vol.7, No. 2.

PEACEMAN, D.W. and RACHFORD, Jr. H.H., 1955, *The Numerical Solution of Parabolic and Elliptic Differential Equations*, J.Soc.Indust. Appl.Math., Vol. 3, No. 1.

RICHARDSON, L.F., 1910, *The Approximate Arithmetical Solution by Finite Difference of Physical Problems Involving Differential Equations, with an Application to the Stresses in Masonry Dam*, Philos.Trans.Roy.Soc. London, Ser. A210, 307-357.

RICHMYER, R.D. and MORTON, K.W., 1967, *Difference Methods for Initial Value Problems*, Interscience Publishers.

SAHIMI, M.S., Ph.D., 1986, *Numerical Methods for Solving Hyperbolic and Parabolic Partial Differential Equations*, Loughborough Univ.

SAUL'YEV, V.K., 1964, *Integration of Equations of Parabolic Type by the Method of Nots*, G.J. Tee, Transl., Pergamon, New York.

SCHULTZ, M.H,, *Spline Analysis*, Prentice-Hall, Inc., N.J.

SMITH, D.G., 1978, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Clarendon Press, Oxford.

STANLY, G. RUBIN and RANDOLPH. A. GRAVES, Jr., 1975, *A Cubic Spline Approximation for Problems in Fluid Mechanics*, NASA TR R-436.

VARGA, R.S., 1962, *Matrix Iterative Analysis*, Prentice-Hall, Inc.

WACHSPRESS, E.L., 1962, *Optimum Alternating-Direction Implicit Iteration Parameters for a Model Problem*, J.Soc.Indust.Appl. Math., 10, 339-350.

WACHSPRESS, E.L., 1966, *Iterative Solution of Elliptic Systems*, Prentice-Hall, N.J.

WACHSPRESS, E.L. and HABETLER, G.J., 1960, *An Alternating-Direction-Implicit Iterative Technique*, J.Soc.Indust.Appl.Math., Vol. 18, No.2.

WALSH, J., *Elliptic Equations, Origine and Connection with Parabolics*, a paper Presented in summer school held In Manchester in 1976.

WILKINSON, J.H., 1978, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford.

YOUNG, D.M., 1954, *Iterative Method for Solving Partial Differential Equations of Elliptic Type*, Trans.Amer.Math.Soc., 76, 92-111, 218, 242-272, 355, 394.

YOUNG, D.M., 1971, *Iterative Solution of Large Linear Systems*, Academic Press, New York.

YOUSIF, W.S., Ph.D., 1984, *New Block Iterative Methods for the Numerical Solution of Boundary Value Problems*, Loughborough Univ.

APPENDIX

SOME SELECTED PROGRAMS

A1: dvspage.fortran

For One-Dimensional p.d.e. using the SPAGE Method

A2: agepirnnew.fortran

For One-Dimensional p.d.e. using the SPAGEI Method

A3: twodagesp.fortran

For Two-Dimensional p.d.e. using the SPAGEI Method

A1: dvspage.fortran

```
      double precision u(0:20,0:5),v(0:20),x(0:20)
     1 ,e(0:20),per(0:20)
      nd=20
c ----------------------
c  Splines A.G.E. method for solving diffusion equations with
c  Derivative B.C.
c
c  u; numerical solution
c  v; exact solution
c  x; x axis
c  e; absolute error
c  per; percentage error
c  n; number of intervals
c  dt;time step
c  m;number of time steps
c  h;interval length
c ---------------------------
c  input
c ---------------------------
      write(6,10000)
      read(5,*) n,dt,m
      h=1.0/n
      t=dt*m
      r=dt/(h*h)
      write(6,20000) r,dt,m,t
c ---------------------------
c  call for the exact solution
c ---------------------------
      call exdiv(nd,n,t,h,v)
c ---------------------------
c  initial conditions
c ---------------------------
      do 500 ip=0,10
             p=ip*0.1
      do 100 i=0,n
      x(i)=h*i
100   u(i,0)=sin(x(i))+cos(x(i))
c ---------------------------
c ---------------------------
c  EVEN OR ODD
c ---------------------------
      y=n/2.0
      iy=int(y)
      if(iy.lt.y)go to 400
c ---------------------------
c  for even number of intervals
c ---------------------------
      write(6,40000)
      read(5,*)nskm
      if(nskm.eq.1)go to 1
      if(nskm.eq.2)go to 2
      if(nskm.eq.3)go to 3
      if(nskm.eq.4)go to 4
1     call dvspr(nd,n,r,p,u,m,dt)
```

```
      go to 300
2     call dvspl(nd,n,r,p,u,m,dt)
      go to 300
3     call dvsspev(nd,n,r,p,u,m,dt)
      go to 300
4     call dvdspev(nd,n,r,p,u,m,dt)
      go to 300
c ------------------------------------
c  for odd number of intervals
c ------------------------------------
400   write(6,50000)
      read(5,*)nskm
      if(nskm.eq.1)go to 11
      if(nskm.eq.2)go to 22
      if(nskm.eq.3)go to 33
      if(nskm.eq.4)go to 44
11    call dvspu(nd,n,r,p,u,m,dt)
      go to 300
22    call dvspc(nd,n,r,p,u,m,dt)
      go to 300
33    call dvsspod(nd,n,r,p,u,m,dt)
      go to 300
44    call dvdspod(nd,n,r,p,u,m,dt)
300   call error(nd,n,j,v,u,e,per)
      write(6,60000)(e(i),i=0,n)
      write(6,70000)(per(i),i=0,n)
500   continue
      write(6,30000)(v(i),i=0,n)
10000 format(' THE SPAGE METHOD FOR HEAT EQs WITH DRVTV B.C.'/
     & ' input No of intevals,time step,No of steps,theta .')
20000 format(' r=',f6.3,2x,' dt=',f7.4,2x,' No of steps='i3,2x,/
     &' time=',f6.3,2x,' theta=',f7.4)
30000 format(' the exact sol'/6(f11.6))
40000 format(' which scheme to use,1 = ryt,2 = lft,3 = s,4 = d')
50000 format(' which scheme to use,1 = u,2 = c,3 = s,4 = d')
60000 format (' the absolute error'/6(f11 .6))
70000 format(' the percentage error'/6(f8.4,3x))
      stop
      end
```

```
      subroutine dvdspod(nd,n,r,p,u,m,dt)
      double precision u(0:nd,0:5)
      h=1.0/n
c ---------------------------------
c  Double Sweeps Scheme,Derivative B.C.,Odd No. of Intervals .
c ---------------------------------
      ap=(3.0-p)/3.0+r
      am=(3.0-p)/3.0-r
      f=p/3.0-r
      g=p/3.0
      det=ap*ap-f*f
      q=ap*am-f*g
      s=ap*g-f*am
      w=r*ap
      z=-f*r
      b=-2.0*h
      cs=cos(1.0)-sin(1.0)
c ---------------------------------
      k=0
700   j=0
      f0=exp(-dt*k)*b
      f0j=exp(-dt*(k+1))*b
      fn=-f0*cs
      fnj=-f0j*cs
      u(0,j+1)=(q*u(0,j)+(s+w)*u(1,j)+z*u(2,j)+w*f0)/det
      u(1,j+1)=(s*u(0,j)+(q+z)*u(1,j)+w*u(2,j)+z*f0)/det
      do 100 i=2,n-3,2
      u(i,j+1)=(w*u(i-1,j)+q*u(i,j)+s*u(i+1,j)+z*u(i+2,j))/det
      u(i+1,j+1)=(z*u(i-1,j)+s*u(i,j)+q*u(i+1,j)+w*u(i+2,j))/det
100   continue
      u(n-1,j+1)=((q+z)*u(n-1,j)+s*u(n,j)+w*u(n-2,j)+z*fn)/det
      u(n,j+1)=((s+w)*u(n-1,j)+q*u(n,j)+z*u(n-2,j)+w*fn)/det
c_____________________________________
      do 200 l=1,2
      j=j+1
      f0=exp(-dt*(k+l))*b
      f0j=exp(-dt*(k+l+1))*b
      fn=-f0*cs
      fnj=-f0j*cs
      do 300 i=1,n-2,2
      u(i,j+1)=(w*u(i-1,j)+q*u(i,j)+s*u(i+1,j)+z*u(i+2,j))/det
300   u(i+1,j+1)=(z*u(i-1,j)+s*u(i,j)+q*u(i+1,j)+w*u(i+2,j))/det
      u(0,j+1)=((r+g)*u(1,j)+am*u(0,j)-f*u(1,j+1)-f0j*f
     1 +f0*g)/ap
      u(n,j+1)=((g+r)*u(n-1,j)+am*u(n,j)-f*u(n-1,j+1)
     1 -f*fnj+g*fn)/ap
200   continue
c ____________________________________
      j=j+1
      f0=exp(-dt*(k+3))*b
      f0j=exp(-dt*(k+4))*b
      fn=-f0*cs
      fnj=-f0j*cs
      u(0,j+1)=(q*u(0,j)+(s+w)*u(1,j)+z*u(2,j)+w*f0)/det
```

```
      u(1,j+1)=(s*u(0,j)+(q+z)*u(1,j)+w*u(2,j)+z*f0)/det
      do 800 i=2,n-3,2
      u(i,j+1)=(w*u(i-1,j)+q*u(i,j)+s*u(i+1,j)+z*u(i+2,j))/det
      u(i+1,j+1)=(z*u(i-1,j)+s*u(i,j)+q*u(i+1,j)+w*u(i+2,j))/det
800   continue
      u(n-1,j+1)=((q+z)*u(n-1,j)+s*u(n,j)+w*u(n-2,j)+z*fn)/det
      u(n,j+1)=((s+w)*u(n-1,j)+q*u(n,j)+z*u(n-2,j)+w*fn)/det
      do 500 i=0,n
500   u(i,0)=u(i,j+1)
      if(k+4.eq.m) go to 600
      k=k+4
      go to 700
600   write(6,10000)p,(u(i,0),i=0,n)
10000 format(' The Double Sweeps Scheme Solution,Odd Intervals.',
     1 'theta =',f7.4,/6(f11.6))
      return
      end
```

```
      subroutine exdiv(nd,m,t,h,v)
      double precision v(0:nd)
c  exact solution of head conduction problem p.d.e.
c  with derivative boundary conditions (Neumann B.C.)
c
c  u(x,t)=exp(-t)*(sin(x)+cos(x))
c
      do 1 i=0,m
      z=h*i
      v(i)=exp(-t)*(sin(z)+cos(z))
1     continue
      return
      end
```

A2: agepirnnew.fortran

```
      double precision u(0:500,0:5),v(0:500),rho(20),x(0:500)
     1,per(0:500),e(0:500)
      common u,v,x
      nd=500
c  Heat conduction problem with periodic B.C.
c  Du/Dt=D2u/D2x + 10(1-x)xt
c  initial cond.    u(x,0)=x(1-x)
c  boundary cond.   u(0,t)=u(1,t)
c
c  The exact sol.:
c     u(x,t)=(1+5tt)/6 -(5/8)*sum(n=1_m)[cos(2n(pi)x)*(4nn(pi)t-1
c               +exp(-4nn(pi)(pi)t)]/(n(pi))**6
c               -sum(1_m)[exp(-4nnpipit)*cos(2npix)]/(nnpipi)    .
c
c
c  i=spatial axis counter
c  j=time step counter
c  dt=time step
c  t=total time
c  h=length of intervals
c  jm=max No. of time steps
c  k=iteration counter
c  km=total No. of iterations for t
c  np=No. of parameters
c  a=diagonal element of splitted matrix
c  f=off diagonal element of matrix
c  u=numerical solution
c  v=exact solution
c  b=r.h.s. of linear system
c  ro=acceleration parameter
      write(6,1)
      read(5,*) m,dt,jm,np
      do 18 j=0,5
      do 18 i=0,nd
18    u(i,j)=0.0
      h=1.0/m
      r=dt/(h*h)
      t=dt*jm
      write(6,6) r,t
      l1=1.0+6.0*r
      l2=2.0+3.0*r
      l3=3.0
      p2=l2
      p1=sqrt(l1*l3)
c
c   initial conditions
c
      do 8 i=0,m
      x(i)=h*i
8     u(i,0)=x(i)*(1.0-x(i))
c
c   boundary conditions
c
      do 9 j=0,5
```

```
9       u(0,j)=u(m,j)
c
c   to choose kind of parameters
c
        write(6,55555)
        read(5,*)krho
        if(krho.eq.1) go to 11111
        if(krho.eq.2) go to 22222
        if(np-2) 33,44,55
33      rho(1)=(p1+p2)/2
        go to 44444
44      rho(1)=p1
        rho(2)=p2
        go to 44444
55      rho(1)=p1
        rho(2)=(p1+p2)/2
        rho(3)=p2
        go to 44444
c   peaceman rachford parameters
11111 do 20000 ip=1,np
20000 rho(ip)=p2*((p1/p2)**((2.0*ip-1.0)/(2.0*np)))
        go to 44444
c   wachspress parameters
22222 do 30000 ip=1,np
30000 rho(ip)=p2*((p1/p2)**((ip-1.0)/(np-1.0)))
44444 y=m/2.0
        iy=int(y)
        if(iy.ge.y) go to 10
c   for odd number of intervals
40      call agepewo(nd,m,jm,r,h,dt,np,rho,u,x)
        go to 20
c
c   for even number of intervals
10      call agepewn(nd,m,jm,r,h,dt,np,rho,u,x)
20      write(6,2)(u(i,0),i=0,m)
c   for the exact solution
        call exper(nd,m,t,h,v)
        write(6,3)(v(i),i=0,m)
c   for the absolute error and percentage error
        call error(nd,m,j,v,u,e,per)
        write(6,4)(e(i),i=0,m)
        write(6,5)(per(i),i=0,m)
c       write(6,10000)time
30      format(' input number of end points')
c10000 format( ' time consumed =' f8.5)
1       format(' input no. of intervals,time step,no. of steps,'/
       1' no. of parameters')
2        format(' the numerical sol.'/6(f12.6))
3       format(' the exact sol.'/6(f12.6))
4       format(' the absolute error'/6(f12.6))
5       format(' the percentage error'/6(f12.4))
6       format(' solution of heat conduction problem'/
       1' with periodic b.c. for even number of intervals.'//
       2' r=',f6.3,2x,' time=',f6.3)
```

```
55555 format(' what kind of parameters; 1 = P.R.,2 = W.,3 = K.')
      stop
      end
```

```
      subroutine agepewn(nd,m,jm,r,h,dt,np,rho,u,x)
      double precision u(0:500,0:5),x(0:500)
     1,c(0:500),b(0:500),rho(20),w(0:500)
c  splines alternating group explicite iterative method with
c  periodic b.c. int even no. of intervals
c
c  jcount;time step counter
c  kcount;iteration counter per time step
c  km;overall iteration counter
c  esp;convergance criterion
      eps=0.000005
      a=2.0+3.0*r
      f=1.0-3.0*r
      p2=3.0
      p1=1.0+6.0*r
      j=0
      km=0
      jcount=0
12     kcount=0
      do 1 i=0,m
1     c(i)=60.0*(1.0-x(i))*x(i)*dt*dt*(jcount+1)
      do 2 i=1,m-1
2     b(i)=(1.0+3.0*r)*(u(i-1,j)+u(i+1,j))
     1 +(4.0-6.0*r)*u(i,j)+c(i)
      b(m)=(1.0+3.0*r)*(u(m-1,j)+u(1,j))+(4.0-6.0*r)*u(m,j)
14    do 13 ip=1,np
      ro=rho(ip)
      u(0,k)=u(m,k)
      ap=a+ro
      am=a-ro
      det=ap*ap-f*f
      p=-ap*f
      q=-ap*am
      v=f*am
      s=f*f
c  first sweep
      do 3 i=1,m-1,2
      w(i)=ap*b(i)-f*b(i+1)
3     w(i+1)=ap*b(i+1)-f*b(i)
c
      k=0
      do 5 i=1,m-3,2
      u(i,k+1)=(p*u(i-1,k)+q*u(i,k)+v*u(i+1,k)
     1 +s*u(i+2,k)+w(i))/det
5     u(i+1,k+1)=(s*u(i-1,k)+v*u(i,k)+q*u(i+1,k)
     1 +p*u(i+2,k)+w(i+1))/det
      u(m-1,k+1)=(p*u(m-2,k)+q*u(m-1,k)+v*u(m,k)
     1 +s*u(1,k)+w(m-1))/det
      u(m,k+1)=(s*u(m-2,k+1)+v*u(m-1,k)+q*u(m,k)
     1 +p*u(1,k)+w(m))/det
c
c  second sweep
      do 17 i=2,m-2,2
      w(i)=ap*b(i)-f*b(i+1)
```

```
      w(i+1)=ap*b(i+1)-f*b(i)
17    continue
      k=k+1
      u(0,k)=u(m,k)
      do 6 i=2,m-2,2
      u(i,k+1)=(p*u(i-1,k)+q*u(i,k)+v*u(i+1,k)
     1 +s*u(i+2,k)+w(i))/det
6     u(i+1,k+1)=(s*u(i-1,k)+v*u(i,k)+q*u(i+1,k)
     1 +p*u(i+2,k)+w(i+1))/det
      u(1,k+1)=(ap*b(1)-f*b(m)+s*u(m-1,k)+v*u(m,k)
     1 +q*u(1,k)+p*u(2,k))/det
      u(m,k+1)=(ap*b(m)-f*b(1)+p*u(m-1,k)+q*u(m,k)
     1 +v*u(1,k)+s*u(2,k))/det
c
c
      k=k+1
      u(0,k)=u(m,k)
      kcount=kcount+1
      do 7 i=1,m
      dif=abs(u(i,k)-u(i,k-2))
      if(dif.gt.eps) go to 16
7     continue
      go to 9
16    if(kcount.eq.100) go to 9
      do 20 i=0,m
20    u(i,k-2)=u(i,k)
      if(ip.eq.np) go to 14
13    continue
9     km=km+kcount
      jcount=jcount+1
c     write(6,*) kcount
      u(0,k)=u(m,k)
      do 11 i=0,m
11    u(i,j)=u(i,k)
      if (jcount.lt.jm) go to 12
10    write(6,15) km
      write(6,18)
18    format('   EVEN NUMBER OF INTERVALS ')
15    format(' the total no. of iterations is',2x,  i6)
      return
      end
```

```
      subroutine exper(nd,m,t,h,v)
      double precision v(0:nd)
c Periodic heat eq. i.c.; u(0,x)=x(1-x)
c B.C.; u(0,t)=u(1,t) .
      pi=3.141592654
      d=(1.0+5.0*t*t)/6.0
      do 2 i=0,m
      x=i*h
      p1=0.0
      q1=0.0
      do 1 n=1,40
      a=n*pi
      b=2.0*a*x
      c=4.0*a*a*t
      p1=p1+cos(b)*(c-1.0+exp(-c))/(a**6)
      q1=q1+exp(-c)*cos(b)/(a*a)
1     continue
      v(i)=d-5.0*p1/8.0-q1
2     continue
      return
      end
```

A3: twodagesp.fortran

```
      double precision u(0:12,0:12,0:160),v(0:12,0:12), rho(5)
     1 ,x(0:12),y(0:12),e(0:12,0:12),per(0:12,0:12)
     2 ,w(0:12,0:12),g(0:12,0:12),b(0:12,0:12)
c  solution of two dimensional parabolic problem
c  using the SPAGEI method.
c
c  eps; convergance criterion
c  n,m; number of x,y nodes
c  dt; time step
c  kmax; number of time steps
c  np; no. of parameters
c  h; interval length
c  u; numerical solution
c  v; exact solution
c
      nd=12
      eps=0.000005
c input no. of x&y nodes ,time step,no.of steps
c ,no. of parameters.
      write(6,1)
      read(5,*)n,m,dt,kmax,np
      do 10 i=0,n
10    x(i)=0.0
      do 2 j=0,m
      y(j)=0.0
      do 2 i=0,n
      v(i,j)=0.0
      b(i,j)=0.0
      g(i,j)=0.0
      w(i,j)=0.0
      do 2 k=0,kmax
2     u(i,j,k)=0.0
c initialisation
      h=1.0/n
      r=dt/(h*h)
      t=dt*kmax
      p2=12.0*r-1.0
      p1=3.0
c
c
      do 20 i=0,n
20    x(i)=i*h
      do 30 j=0,m
30    y(j)=j*h
c
c initial conditions
c
      do 40 i=0,n
      do 40 j=0,m
40    u(i,j,0)=sin(x(i))*sin(y(j))+(x(i))**2+(y(j))**2
c boundary conditions left&right
      do 50 k=1,kmax
      do 50 j=0,m
      u(0,j,k)=(y(j))**2
```

```
50    u(n,j,k)=sin(x(n))*sin(y(j))*exp(-dt*k)+(x(n))**2+(y(j))**2
c boundary conditions top&bottom
      do 60 k=1,kmax
      do 60 i=0,n
      u(i,0,k)=(x(i))**2
60    u(i,m,k)=sin(x(i))*sin(y(m))*exp(-dt*k)+(x(i))**2+(y(m))**2
c
      a=1.0+6.0*r
      f=1.0-3.0*r
      d=1.0+3.0*r
      c=2.0-12.0*r
c
      itsum=0
      k=0
c
c
      write(6,3)
      read(5,*)krho
c evaluating the  source term (g function)
c
280   do 200 i=1,n-1
      do 200 j=1,m-1
200   g(i,j)=6.0*dt*(sin(x(i))*sin(y(j))*exp(-dt*(k+0.5))-4.0)
c
c evaluating the r.h.s.
c
      do 210 i=1,n-1
      do 210 j=1,m-1
      b(i,j)=c*u(i,j,k)+d*(u(i-1,j,k)+u(i+1,j,k)
     1 +u(i,j-1,k)+u(i,j+1,k))+g(i,j)
 210   continue
      do 220 j=2,m-2
      b(1,j)=b(1,j)-f*u(0,j,k+1)
      b(n-1,j)=b(n-1,j)-f*u(n,j,k+1)
220   continue
c
      do 230 i=2,n-2
      b(i,1)=b(i,1)-f*u(i,0,k+1)
      b(i,m-1)=b(i,m-1)-f*u(i,m,k+1)
230   continue
c
      b(1,1)=b(1,1)-f*(u(1,0,k+1)+u(0,1,k+1))
      b(1,m-1)=b(1,m-1)-f*(u(0,m-1,k+1)+u(1,m,k+1))
      b(n-1,1)=b(n-1,1)-f*(u(n-1,0,k+1)+u(n,1,k+1))
      b(n-1,m-1)=b(n-1,m-1)-f*(u(n-1,m,k+1)+u(n,m-1,k+1))
c
      it=0
c
      do 260 i=1,n-1
      do 260 j=1,m-1
260   u(i,j,it)=u(i,j,k)
c
c choosing kind of parameters; 1 for p.r. ;2 for w.
270   do 70 ip=1,np
```

```
      if (krho.eq.2) go to 80
      ro=p2*((p1/p2)**((2.0*ip-1)/(2.0*np)))
      go to 140
80    ro=p2*((p1/p2)**((ip-1)/(np-1)))
c
140   ap=a+ro
      am=a-ro
90      call toprit(nd,n,m,k,it,am,ap,f,u,b,w)
c
c
      it=it+1
c
c
c
      do 240 i=1,n-1
      do 240 j=1,m-1
      dif=abs(u(i,j,it-2)-u(i,j,it))
      if(dif.gt.eps) go to 250
240   continue
c
190   itsum=itsum+(it)/2
      k=k+1
      do 110 i=1,n-1
      do 110 j=1,m-1
110   u(i,j,k)=u(i,j,it)
      if(k.eq.kmax)go to 120
      go to 280
250   if(it.eq.50) go to 190
      if(ip.eq.np) go to 270
70    continue
c
120   write(6,290) itsum
c
      call tdexact(nd,n,m,t,x,y,v)
      call tderror(nd,n,m,kmax,v,u,e,per)
      write(6,150) dt,kmax,t,r,krho,np
      write(6,130)((u(i,j,k),j=1,m-1),i=1,n-1,4)
      write(6,160)((v(i,j),j=1,m-1),i=1,n-1,4)
      write(6,170)((e(i,j),j=1,m-1),i=1,n-1,4)
      write(6,180)((per(i,j),j=1,m-1),i=1,n-1,4)
1      format(' input; x_nodes, y_nodes,time step
     1 ,no. of time steps, no. of parameters')
3      format(' input kind of parameters;1 for P.R. ;2 for W.')
150    format(' time step=',f5.3,x,' no. of steps='
     1,i3,x,'t=',f6.3,
     1 /' r=',f5.2,x,' kind of prmtrs='
     1 ,i2,x,' no. of prmtrs=',i3/)
130    format(/' the numerical sol.',/9(f8.6,x))
160    format(/' the exact sol.'/,9(f8.6,x))
170    format(/' the absolute error'/,9(1pe8.1,x))
180    format(/' the percentage error',/9(f8.6,x))
290    format(/' total iterations=',i5)
       stop
       end
```

```
      subroutine toprit(nd,n,m,k,it,am,ap,f,u,b,w)
c
c subroutine to find sol. of two-d diff eq.
c using splines in two dimensions .
c
c single points at top-right &bottom-left
c
c
      double precision u(0:nd,0:nd,0:120)
     1 ,w(0:nd,0:nd),b(0:nd,0:nd)
c the constant parameters in general
c
      c1=ap*ap-2.0*f*f
      c2=-ap*f
      c3=2.0*f*f
c
      c41=-am*c1
      c42=-am*c2
      c43=-am*c3
c
      c51=-f*c1
      c52=-f*c2
      c53=-f*c3
c
c determinant for 4*4 matrix
c
      det=ap*(ap*ap-4.0*f*f)
c
      c6=-ap*am
      c7=am*f
      c8=c2
      c9=f*f
c
c determinant for 2*2 matrix
c
      det1=ap*ap-f*f
c
c  first sweep
c
c
c
c for line near left boundary
c
      do 20 j=2,m-4,2
      w(1,j)=ap*b(1,j)-f*b(1,j+1)
      w(1,j+1)=-f*b(1,j)+ap*b(1,j+1)
      u(1,j,it+1)=(w(1,j)+c6*u(1,j,it)+c8*(u(1,j-1,it)+u(2,j,it))
     1 +c7*u(1,j+1,it)+c9*(u(1,j+2,it)+u(2,j+1,it)))/det1
      u(1,j+1,it+1)=(w(1,j+1)+c7*u(1,j,it)
     1 +c9*(u(1,j-1,it)+u(2,j,it))
     1 +c6*u(1,j+1,it)+c8*(u(1,j+2,it)+u(2,j+1,it)))/det1
20     continue
c
c
```

```
c for 2x2 block near top_left corner
c
      w(1,m-2)=ap*b(1,m-2)-f*b(1,m-1)
      w(1,m-1)=-f*b(1,m-2)+ap*b(1,m-1)
      u(1,m-2,it+1)=(w(1,m-2)+c6*u(1,m-2,it)+c7*u(1,m-1,it)
     1 +c8*(u(1,m-3,it)+u(2,m-2,it))+c9*u(2,m-1,it))/det1
      u(1,m-1,it+1)=(w(1,m-1)+c7*u(1,m-2,it)+c6*u(1,m-1,it)
     1 +c9*(u(1,m-3,it)+u(2,m-2,it))+c8*u(2,m-1,it))/det1
c
c for line near bottom boundary
c
      do 30 i=2,n-4,2
      w(i,1)=ap*b(i,1)-f*b(i+1,1)
      w(i+1,1)=-f*b(i,1)+ap*b(i+1,1)
      u(i,1,it+1)=(w(i,1)+c6*u(i,1,it)+c8*(u(i-1,1,it)+u(i,2,it))
     1 +c7*u(i+1,1,it)+c9*(u(i+1,2,it)+u(i+2,1,it)))/det1
      u(i+1,1,it+1)=(w(i+1,1)+c7*u(i,1,it)
     1 +c9*(u(i-1,1,it)+u(i,2,it))
     1 +c6*u(i+1,1,it)+c8*(u(i+1,2,it)+u(i+2,1,it)))/det1
30    continue
c
c
c for 2x2 block near bottom_right corner
c
      w(n-2,1)=ap*b(n-2,1)-f*b(n-1,1)
      w(n-1,1)=-f*b(n-2,1)+ap*b(n-1,1)
      u(n-2,1,it+1)=(w(n-2,1)+c6*u(n-2,1,it)+c7*u(n-1,1,it)
     1 +c8*(u(n-3,1,it)+u(n-2,2,it))+c9*u(n-1,2,it))/det1
      u(n-1,1,it+1)=(w(n-1,1)+c7*u(n-2,1,it)+c6*u(n-1,1,it)
     1 +c9*(u(n-3,1,it)+u(n-2,2,it))+c8*u(n-1,2,it))/det1
c
c for bottom-left corner single point
c
      u(1,1,it+1)=(b(1,1)-am*u(1,1,it)-f*(u(1,2,it)
     1 +u(2,1,it)))/ap
c
c  for inner 4x4 blocks
c
      do 10 i=2,n-2,2
      do 10 j=2,m-2,2
      w(i,j)=c1*b(i,j)+c2*(b(i,j+1)+b(i+1,j))+c3*b(i+1,j+1)
      w(i,j+1)=c1*b(i,j+1)+c2*(b(i,j)+b(i+1,j+1))+c3*b(i+1,j)
      w(i+1,j)=c1*b(i+1,j)+c2*(b(i,j)+b(i+1,j+1))+c3*b(i,j+1)
      w(i+1,j+1)=c1*b(i+1,j+1)+c2*(b(i,j+1)+b(i+1,j))+c3*b(i,j)
10    continue
c
c
      do 70 i=2,n-4,2
      do 70 j=2,m-4,2
      u(i,j,it+1)=(w(i,j)+c41*u(i,j,it)+c42*(u(i,j+1,it)
     1 +u(i+1,j,it))
     1 +c43*u(i+1,j+1,it)+c51*(u(i-1,j,it)+u(i,j-1,it))
     2 +c52*(u(i-1,j+1,it)+u(i,j+2,it)+u(i+1,j-1,it)+u(i+2,j,it))
     3 +c53*(u(i+1,j+2,it)+u(i+2,j+1,it)))/det
```

```
 u(i,j+1,it+1)=(w(i,j+1)+c42*(u(i,j,it)+u(i+1,j+1,it))
1 +c52*(u(i-1,j,it)+u(i,j-1,it)+u(i+1,j+2,it)+u(i+2,j+1,it))
2 +c41*u(i,j+1,it)+c51*(u(i-1,j+1,it)+u(i,j+2,it))
3 +c43*u(i+1,j,it)+c53*(u(i+1,j-1,it)+u(i+2,j,it)))/det
 u(i+1,j,it+1)=(w(i+1,j)+c42*(u(i,j,it)+u(i+1,j+1,it))
1 +c43*u(i,j+1,it)+c41*u(i+1,j,it)+c53*(u(i-1,j+1,it)
1 +u(i,j+2,it))
      2 +c52*(u(i-1,j,it)+u(i,j-1,it)+u(i+1,j+2,it)+u(i+2,j+1,it))
      3 +c51*(u(i+1,j-1,it)+u(i+2,j,it)))/det
       u(i+1,j+1,it+1)=(w(i+1,j+1)+c42*(u(i,j+1,it)+u(i+1,j,it))
      1 +c52*(u(i-1,j+1,it)+u(i,j+2,it)+u(i+1,j-1,it)+u(i+2,j,it))
      2 +c43*u(i,j,it)+c41*u(i+1,j+1,it)+c53*(u(i-1,j,it)
      1 +u(i,j-1,it))
      3 +c51*(u(i+1,j+2,it)+u(i+2,j+1,it)))/det
70     continue
c
c for 4x4 blocks near top boundary
c
       do 110 i=2,n-4,2
       u(i,m-2,it+1)=(w(i,m-2)+c42*(u(i,m-1,it)+u(i+1,m-2,it))
      1 +c43*u(i+1,m-1,it)+c51*(u(i-1,m-2,it)+u(i,m-3,it))
      2 +c52*(u(i-1,m-1,it)+u(i+1,m-3,it)+u(i+2,m-2,it))
      3 +c41*u(i,m-2,it)+c53*u(i+2,m-1,it))/det
       u(i,m-1,it+1)=(w(i,m-1)+c42*(u(i,m-2,it)+u(i+1,m-1,it))
      1 +c52*(u(i-1,m-2,it)+u(i,m-3,it)+u(i+2,m-1,it))
      2 +c41*u(i,m-1,it)+c43*u(i+1,m-2,it)+c51*u(i-1,m-1,it)
      3 +c53*(u(i+1,m-3,it)+u(i+2,m-2,it)))/det
       u(i+1,m-2,it+1)=(w(i+1,m-2)+c42*(u(i,m-2,it)+u(i+1,m-1,it))
      1 +c43*u(i,m-1,it)+c41*u(i+1,m-2,it)+c53*u(i-1,m-1,it)
      2 +c52*(u(i-1,m-2,it)+u(i,m-3,it)+u(i+2,m-1,it))
      3 +c51*(u(i+1,m-3,it)+u(i+2,m-2,it)))/det
       u(i+1,m-1,it+1)=(w(i+1,m-1)+c42*(u(i,m-1,it)+u(i+1,m-2,it))
      1 +c52*(u(i-1,m-1,it)+u(i+1,m-3,it)+u(i+2,m-2,it))
      2 +c41*u(i+1,m-1,it)+c53*(u(i-1,m-2,it)-u(i,m-3,it))
      3 +c43*u(i,m-2,it)+c51*u(i+2,m-1,it))/det
110     continue
c
c for 4x4 blocks near right boundary
c
       do 90 j=2,m-4,2
       u(n-2,j,it+1)=(w(n-2,j)+c42*(u(n-2,j+1,it)+u(n-1,j,it))
1 +c43*u(n-1,j+1,it)+c51*(u(n-3,j,it)+u(n-2,j-1,it))
2 +c52*(u(n-3,j+1,it)+u(n-2,j+2,it)+u(n-1,j-1,it))
3 +c41*u(n-2,j,it)+c53*u(n-1,j+2,it))/det
 u(n-2,j+1,it+1)=(w(n-2,j+1)+c42*(u(n-2,j,it)+u(n-1,j+1,it))
1 +c52*(u(n-3,j,it)+u(n-2,j-1,it)+u(n-1,j+2,it))
2 +c41*u(n-2,j+1,it)+c51*(u(n-3,j+1,it)-u(n-2,j+2,it))
3 +c43*u(n-1,j,it)+c53*u(n-1,j-1,it))/det
 u(n-1,j,it+1)=(w(n-1,j)+c42*(u(n-2,j,it)+u(n-1,j+1,it))
1 +c43*u(n-2,j+1,it)+c53*(u(n-3,j+1,it)-u(n-2,j+2,it))
2 +c52*(u(n-3,j,it)+u(n-2,j-1,it)+u(n-1,j+2,it))
3 +c41*u(n-1,j,it)+c51*u(n-1,j-1,it))/det
 u(n-1,j+1,it+1)=(w(n-1,j+1)+c42*(u(n-2,j+1,it)+u(n-1,j,it))
1 +c52*(u(n-3,j+1,it)+u(n-2,j+2,it)+u(n-1,j-1,it))
```

```
     2 +c41*u(n-1,j+1,it)+c53*(u(n-3,j,it)+u(n-2,j-1,it))
     3 +c43*u(n-2,j,it)+c51*u(n-1,j+2,it))/det
90     continue
c
c for 4x4 block near top-right boundary corner
c
      u(n-2,m-2,it+1)=(w(n-2,m-2)+c42*(u(n-2,m-1,it)
     1 +u(n-1,m-2,it))
     1 +c43*u(n-1,m-1,it)+c51*(u(n-3,m-2,it)+u(n-2,m-3,it))
     2 +c52*(u(n-3,m-1,it)+u(n-1,m-3,it))+c41*u(n-2,m-2,it))/det
      u(n-2,m-1,it+1)=(w(n-2,m-1)+c42*(u(n-2,m-2,it)
     1 +u(n-1,m-1,it))
     1 +c52*(u(n-3,m-2,it)+u(n-2,m-3,it))
     2 +c41*u(n-2,m-1,it)+c43*u(n-1,m-2,it)+c51*u(n-3,m-1,it)
     3 +c53*u(n-1,m-3,it))/det
      u(n-1,m-2,it+1)=(w(n-1,m-2)+c42*(u(n-2,m-2,it)
     1 +u(n-1,m-1,it))
     1 +c43*u(n-2,m-1,it)+c41*u(n-1,m-2,it)+c53*u(n-3,m-1,it)
     2 +c52*(u(n-3,m-2,it)+u(n-2,m-3,it))
     3 +c51*u(n-1,m-3,it))/det
      u(n-1,m-1,it+1)=(w(n-1,m-1)+c42*(u(n-2,m-1,it)
     1 +u(n-1,m-2,it))
     1 +c52*(u(n-3,m-1,it)+u(n-1,m-3,it))
     2 +c43*u(n-2,m-2,it)+c41*u(n-1,m-1,it)
     3 +c53*(u(n-3,m-2,it)+u(n-2,m-3,it)))/det
c
      it=it+1
c
c  second sweep
c
c for top-right corner single point
c
      u(n-1,m-1,it+1)=(b(n-1,m-1)-am*u(n-1,m-1,it)
     1 -f*(u(n-1,m-2,it)+u(n-2,m-1,it)))/ap
c
c for line near right boundary
c
      do 40 j=3,m-3,2
      w(n-1,j)=ap*b(n-1,j)-f*b(n-1,j+1)
      w(n-1,j+1)=-f*b(n-1,j)+ap*b(n-1,j+1)
      u(n-1,j,it+1)=(w(n-1,j)+c6*u(n-1,j,it)
     1 +c8*(u(n-2,j,it)+u(n-1,j-1,it))
     2 +c7*u(n-1,j+1,it)+c9*(u(n-2,j+1,it)+u(n-1,j+2,it)))/det1
      u(n-1,j+1,it+1)=(w(n-1,j+1)+c7*u(n-1,j,it)+c6*u(n-1,j+1,it)
     1 +c8*(u(n-2,j+1,it)+u(n-1,j+2,it))
     2 +c9*(u(n-2,j,it)+u(n-1,j-1,it)))/det1
40     continue
c
c
c for 2x2 block near right_bottom corner
c
      w(n-1,1)=ap*b(n-1,1)-f*b(n-1,2)
      w(n-1,2)=-f*b(n-1,1)+ap*b(n-1,2)
      u(n-1,1,it+1)=(w(n-1,1)+c6*u(n-1,1,it)+c7*u(n-1,2,it)
```

```
     1 +c8*u(n-2,1,it)+c9*(u(n-2,2,it)+u(n-1,3,it)))/det1
      u(n-1,2,it+1)=(w(n-1,2)+c7*u(n-1,1,it)+c6*u(n-1,2,it)
     1 +c9*u(n-2,1,it)+c8*(u(n-2,2,it)+u(n-1,3,it)))/det1
c
c for line near top boundary
c
      do 50 i=3,n-3,2
      w(i,m-1)=ap*b(i,m-1)-f*b(i+1,m-1)
      w(i+1,m-1)=-f*b(i,m-1)+ap*b(i+1,m-1)
      u(i,m-1,it+1)=(w(i,m-1)+c6*u(i,m-1,it)+c7*u(i+1,m-1,it)
     1 +c8*(u(i,m-2,it)+u(i-1,m-1,it))
     2 +c9*(u(i+1,m-2,it)+u(i+2,m-1,it)))/det1
      u(i+1,m-1,it+1)=(w(i+1,m-1)+c6*u(i+1,m-1,it)+c7*u(i,m-1,it)
     1 +c9*(u(i,m-2,it)+u(i-1,m-1,it))
     2 +c8*(u(i+1,m-2,it)+u(i+2,m-1,it)))/det1
50    continue
c
c
c for 2x2 block near top_left corner
c
      w(1,m-1)=ap*b(1,m-1)-f*b(2,m-1)
      w(2,m-1)=-f*b(1,m-1)+ap*b(2,m-1)
      u(1,m-1,it+1)=(w(1,m-1)+c6*u(1,m-1,it)+c7*u(2,m-1,it)
     1 +c8*u(1,m-2,it)+c9*(u(2,m-2,it)+u(3,m-1,it)))/det1
      u(2,m-1,it+1)=(w(2,m-1)+c7*u(1,m-1,it)+c6*u(2,m-1,it)
     1 +c9*u(1,m-2,it)+c8*(u(2,m-2,it)+u(3,m-1,it)))/det1
c
c  for inner 4x4 blocks
c
      do 60 i=1,n-3,2
      do 60 j=1,m-3,2
      w(i,j)=c1*b(i,j)+c2*(b(i,j+1)+b(i+1,j))+c3*b(i+1,j+1)
      w(i,j+1)=c1*b(i,j+1)+c2*(b(i,j)+b(i+1,j+1))+c3*b(i+1,j)
      w(i+1,j)=c1*b(i+1,j)+c2*(b(i,j)+b(i+1,j+1))+c3*b(i,j+1)
      w(i+1,j+1)=c1*b(i+1,j+1)+c2*(b(i,j+1)+b(i+1,j))+c3*b(i,j)
60    continue
c
c
      do 80 i=3,n-3,2
      do 80 j=3,m-3,2
      u(i,j,it+1)=(w(i,j)+c41*u(i,j,it)c42*(u(i,j+1,it)
     1 +u(i+1,j,it))
     1 +c43*u(i+1,j+1,it)+c51*(u(i-1,j,it)+u(i,j-1,it))
     2 +c52*(u(i-1,j+1,it)+u(i,j+2,it)+u(i+1,j-1,it)+u(i+2,j,it))
     3 +c53*(u(i+1,j+2,it)+u(i+2,j+1,it)))/det
      u(i,j+1,it+1)=(w(i,j+1)+c42*(u(i,j,it)+u(i+1,j+1,it))
     1 +c52*(u(i-1,j,it)+u(i,j-1,it)+u(i+1,j+2,it)+u(i+2,j+1,it))
     2 +c41*u(i,j+1,it)+c43*u(i+1,j,it)+c51*(u(i-1,j+1,it)
     1 +u(i,j+2,it))
     3 +c53*(u(i+1,j-1,it)+u(i+2,j,it)))/det
      u(i+1,j,it+1)=(w(i+1,j)+c42*(u(i,j,it)+u(i+1,j+1,it))
     1 +c43*u(i,j+1,it)+c41*u(i+1,j,it)+c53*(u(i-1,j+1,it)
     1 +u(i,j+1,it))
     2 +c52*(u(i-1,j,it)+u(i,j-1,it)+u(i+1,j+2,it)+u(i+2,j+1,it))
```

```
     3 +c51*(u(i+1,j-1,it)+u(i+2,j,it)))/det
      u(i+1,j+1,it+1)=(w(i+1,j+1)+c42*(u(i,j+1,it)+u(i+1,j,it))
     1 +c52*(u(i-1,j+1,it)+u(i,j+2,it)+u(i+1,j-1,it)+u(i+2,j,it))
     2 +c43*u(i,j,it)+c41*u(i+1,j+1,it)+c53*(u(i-1,j,it)
     1 +u(i,j-1,it))
     3 +c51*(u(i+1,j+2,it)+u(i+2,j+1,it)))/det
80    continue
c
c for 4x4 blocks near bottom boundary
c
      do 120 i=3,m-3,2
      u(i,1,it+1)=(w(i,1)+c41*u(i,1,it)+c42*(u(i,2,it)
     1 +u(i+1,1,it))+c43*u(i+1,2,it)+c51*(u(i-1,1,it))
     2 +c52*(u(i-1,2,it)+u(i,3,it)+u(i+2,1,it))
     3 +c53*(u(i+1,3,it)+u(i+2,2,it)))/det
      u(i,2,it+1)=(w(i,2)+c42*(u(i,1,it)+u(i+1,2,it))
     1 +c52*(u(i-1,1,it)+u(i+1,3,it)+u(i+2,2,it))
     2 +c41*u(i,2,it)+c43*u(i+1,1,it)+c51*(u(i-1,2,it)+u(i,3,it))
     3 +c53*(u(i+2,1,it)))/det
      u(i+1,1,it+1)=(w(i+1,1)+c42*(u(i,1,it)+u(i+1,2,it))
     1 +c43*u(i,2,it)+c41*u(i+1,1,it)+c53*(u(i-1,2,it)+u(i,3,it))
     2 +c52*(u(i-1,1,it)+u(i+1,3,it)+u(i+2,2,it))
     3 +c51*(u(i+2,1,it)))/det
      u(i+1,2,it+1)=(w(i+1,2)+c42*(u(i,2,it)+u(i+1,1,it))
     1 +c52*(u(i-1,2,it)+u(i,3,it)+u(i+2,1,it))
     2 +c43*u(i,1,it)+c41*u(i+1,2,it)+c53*(u(i-1,1,it))
     3 +c51*(u(i+1,3,it)+u(i+2,2,it)))/det
120    continue
c
c for 4x4 blocks near left boundary
c
      do 100 j=3,m-3,2
      u(1,j,it+1)=(w(1,j)+c41*u(1,j,it)+c42*(u(1,j+1,it)
     1 +u(2,j,it))+c43*u(2,j+1,it)+c51*(+u(1,j-1,it))
     2 +c52*(+u(1,j+2,it)+u(2,j-1,it)+u(3,j,it))
     3 +c53*(u(2,j+2,it)+u(3,j+1,it)))/det
      u(1,j+1,it+1)=(w(1,j+1)+c42*(u(1,j,it)+u(2,j+1,it))
     1 +c52*(u(1,j-1,it)+u(2,j+2,it)+u(3,j+1,it))
     2 +c41*u(1,j+1,it)+c43*u(2,j,it)+c51*(u(1,j+2,it))
     3 +c53*(u(2,j-1,it)+u(3,j,it)))/det
      u(2,j,it+1)=(w(2,j)+c42*(u(1,j,it)+u(2,j+1,it))
     1 +c43*u(1,j+1,it)+c41*u(2,j,it)+c53*u(1,j+2,it)
     2 +c52*(u(1,j-1,it)+u(2,j+2,it)+u(3,j+1,it))
     3 +c51*(u(2,j-1,it)+u(3,j,it)))/det
      u(2,j+1,it+1)=(w(2,j+1)+c42*(u(1,j+1,it)+u(2,j,it))
     1 +c52*(u(1,j+2,it)+u(2,j-1,it)+u(3,j,it))
     2 +c43*u(1,j,it)+c41*u(2,j+1,it)+c53*u(1,j-1,it)
     3 +c51*(u(2,j+2,it)+u(3,j+1,it)))/det

100    continue
c
c for 4x4 block near left-bottom corner
c
      u(1,1,it+1)=(w(1,1)+c41*u(1,1,it)+c42*(u(1,2,it)+u(2,1,it))
```

```
     1 +c43*u(2,2,it)
     2 +c52*(u(1,3,it)+u(3,1,it))
     3 +c53*(u(2,3,it)+u(3,2,it)))/det
      u(1,2,it+1)=(w(1,2)+c42*(u(1,1,it)+u(2,2,it))
     1 +c52*(u(2,3,it)+u(3,2,it))
     2 +c41*u(1,2,it)+c43*u(2,1,it)+c51*(u(1,3,it))
     3 +c53*(u(3,1,it)))/det
      u(2,1,it+1)=(w(2,1)+c42*(u(1,1,it)+u(2,2,it))
     1 +c43*u(1,2,it)+c41*u(2,1,it)+c53*u(1,3,it)
     2 +c52*(u(2,3,it)+u(3,2,it))
     3 +c51*(u(3,1,it)))/det
      u(2,2,it+1)=(w(2,2)+c42*(u(1,2,it)+u(2,1,it))
     1 +c52*(u(1,3,it)+u(3,1,it))
     2 +c43*u(1,1,it)+c41*u(2,2,it)
     3 +c51*(u(2,3,it)+u(3,2,it)))/det
      write(6,700)((u(i,j,it+1),i=1,n-1),j=1,n-1,4)
700   format(9(f8.6))
      return
      end
```