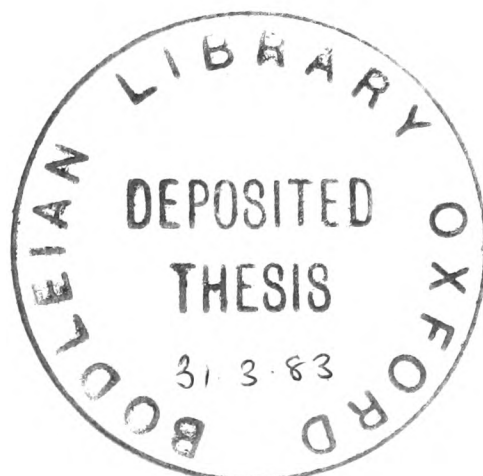


THE NUMERICAL SOLUTION OF  
ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

A thesis submitted for the degree of  
Doctor of Philosophy

Michaelmas Term 1982

T.N. Phillips,  
Merton College,  
OXFORD.



The Numerical Solution of Elliptic Partial Differential Equations  
by - Timothy N. Phillips, of Merton College, Oxford.

A thesis submitted for the degree of Doctor of Philosophy  
in Michaelmas Term 1982.

ABSTRACT

In this thesis numerical methods for solving elliptic partial differential equations are developed. These differential equations are discretized using finite differences and the resulting algebraic equations are then solved using iterative techniques. In particular, dynamic A.D.I. and multigrid methods are considered. These methods are tested on a model problem and then extended to solve more difficult problems. A coupled system of equations arising from solid state electronics is solved. Techniques are discussed to improve the accuracy of the resulting approximation near the two weak singularities the problem possesses. The dynamic A.D.I. and multigrid methods are extended to solve the biharmonic equation. A multigrid method which treats the biharmonic equation as a coupled system of two second order elliptic equations is also considered. A dynamic A.D.I. method for solving the 'double glazing' problem is presented. A non-uniform grid is used to resolve the boundary layers which develop for large values of the Rayleigh number.

## PREFACE

This thesis is an account of research undertaken in the Oxford University Computing Laboratory on the numerical solution of elliptic partial differential equations. It concerns iterative methods for solving discretizations of differential equations obtained by the finite difference method. In particular, we develop alternating direction implicit (A.D.I.) and multigrid methods for solving coupled systems of equations.

The material in Chapter 1 is a summary of known work on A.D.I. methods. We describe this method by applying it to Poisson's equation on the unit square and look at two techniques for improving convergence. The first is by choosing a priori a suitable sequence of parameters and the second is the dynamic A.D.I. (D.A.D.I.) method which uses a fully automatic strategy to change the acceleration parameter as the iteration proceeds.

Chapter 2 is mainly a summary of known work on multigrid methods. The multigrid method is an iterative procedure which uses several grids or levels of discretization. We describe the reasoning that lies behind the multigrid method, and explain how a typical algorithm is constructed and its components chosen. We describe a method that combines A.D.I. with multigrid. This is the only part of the chapter that is original.

Chapters 3, 4, 5 and 6 apply and extend the techniques developed in Chapters 1 and 2 to solve practical problems. This work is, on the whole, original except where explicit reference is made to other work.

Chapters 3 and 4 are concerned with the numerical solution of a coupled system of nonlinear partial differential equations which arise in the field of solid state electronics. The problem is defined in a rectangle and has singularities in two of the corners.

In Chapter 3 we apply the D.A.D.I. method and a multigrid method

to obtain numerical solutions to this problem.

Chapter 4 looks at techniques of improving the approximation, obtained in Chapter 3, in the neighbourhoods of the singularities. Firstly, we look into the use of non-uniform grids in finite difference calculations. The D.A.D.I. method is then applied to obtain numerical solutions to this problem on non-uniform grids which have smaller mesh spacings in the vicinities of the singularities. The second method involves a transformation of the problem to one which has no singularities.

In Chapter 5 we investigate A.D.I. and multigrid methods for solving the biharmonic equation. We review a paper which describes how an A.D.I. method may be applied to the biharmonic equation. We show how the D.A.D.I. method can be extended to solve problems of this type. Finally, we look at several multigrid methods including one which treats the biharmonic equation as a coupled system of two second order elliptic equations.

Chapter 6 is concerned with the problem of buoyancy driven flow in a vertical, rectangular cavity whose vertical sides are at different temperatures and whose horizontal sides are insulated. This is known as the 'double glazing' problem. The problem gives rise to a coupled system of nonlinear partial differential equations, one of which is fourth order. Numerical solutions to this problem are obtained using a D.A.D.I. method. For large values of the Rayleigh number the flow patterns develop strong boundary layers. These boundary layers are resolved by applying the D.A.D.I. method to the discretization of this problem on a non-uniform grid.

All of the computational work has been done on the Oxford University ICL 2980 computer.

*This thesis is dedicated to my late grandmother,  
Elsie May Phillips.*

## CONTENTS

	<u>Page</u>
<u>Chapter 1</u> ALTERNATING DIRECTION IMPLICIT METHODS	1
1.1 Introduction	1
1.2 The model problem	1
1.3 The A.D.I. method	4
1.4 A.D.I. parameter sequences	7
1.5 D.A.D.I. philosophy	9
1.6 Analysis and D.A.D.I. strategy	11
1.7 Numerical results	14
1.8 Remarks	17
<u>Chapter 2</u> MULTIGRID METHODS	18
2.1 Introduction	18
2.2 The multigrid philosophy	19
2.3 Relaxation and smoothing	20
2.4 Multigrid algorithms	23
2.5 Multigrid components	27
2.6 A.D.I. smoothing factor	30
2.7 Multigrid routines	32
2.8 Convergence	34
2.9 Numerical example	34
<u>Chapter 3</u> THE PHILIPS PROBLEM: NUMERICAL SOLUTION	37
3.1 Introduction	37
3.2 The differential equations	37
3.3 Nature of the singularities	40
3.4 Finite difference approximation	41
3.5 Dynamic A.D.I. method	47
3.6 Multigrid method	53
3.7 Numerical results	58
<u>Chapter 4</u> THE PHILIPS PROBLEM: TREATMENT OF THE SINGULARITY	66
4.1 Introduction	66
4.2 Non-uniform grids	66
4.3 D.A.D.I. method of solution	68
4.4 Transformation method	74
4.5 Conclusions	80
<u>Chapter 5</u> THE BIHARMONIC EQUATION	85
5.1 Introduction	85
5.2 Finite difference equations	85
5.3 A.D.I. method	87
5.4 Dynamic A.D.I. method	89
5.5 Multigrid using point relaxation	94
5.6 Multigrid using A.D.I.	96
5.7 Numerical results	99
5.8 The coupled equation approach	103

	<u>Page</u>
<u>Chapter 6</u> NATURAL CONVECTION IN AN ENCLOSED CAVITY	106
6.1    Introduction	106
6.2    Formulation of the problem	106
6.3    Solution of a nonlinear equation	109
6.4    Finite difference equations	110
6.5    Method of solution	111
6.6    Numerical results	117
6.7    Non-uniform grids	127
 Appendix I	 132
Appendix II	134
References	136

## ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. D.F. Mayers, for suggesting this area of work and for his continual help throughout my time of research. Thanks are also due to members of the Oxford University Computing Laboratory for many helpful discussions. I acknowledge with gratitude the financial support of the Science and Engineering Research Council over the past three years.

Finally, I would like to thank my family for their encouragement and support during the many years of my education.



## Chapter 1

### ALTERNATING DIRECTION IMPLICIT METHODS

#### 1.1 Introduction

Here we are concerned with alternating direction implicit (A.D.I) methods for solving elliptic partial differential equations. These methods arise from a certain treatment of the finite difference equations formed by the discretization of these equations. In fact A.D.I. methods for solving elliptic partial differential equations were derived from numerical methods for solving parabolic problems. The first description of an A.D.I. method was given by Peaceman and Rachford (1955). This work was extended by Douglas and Rachford (1956). The method has become important because of its guaranteed improved rate of convergence over that of the successive over-relaxation (S.O.R.) method for certain model problems. For a full treatment and analysis of the A.D.I. method for model problems the reader is referred to Varga (1962), Young (1971) and Williams (1979).

In this chapter we describe an A.D.I. method for solving Poisson's equation on the unit square and indicate how a parameter sequence can be chosen to improve convergence. We describe the dynamic A.D.I. method of Doss and Miller (1979) which uses a fully automatic strategy to change the acceleration parameter.

#### 1.2 The Model Problem

Let  $\Omega$  be the unit square  $\{(x,y): 0 < x < 1, 0 < y < 1\}$  and let  $\Gamma$  be the boundary of  $\Omega$ . Consider Poisson's equation defined in  $\Omega$ ,

$$-\nabla^2 u = f(x,y), \quad (x,y) \in \Omega \tag{1.1}$$

with Dirichlet boundary conditions

$$u(x,y) = g(x,y), \quad (x,y) \in \Gamma \tag{1.2}$$

Cover  $\Omega$  with a square mesh of grid size  $h = 1/N$  where  $N$  is a positive integer. At the interior mesh points equation (1.1) can be discretised (see Smith (1978)) by the five-point formula

$$\left\{ -v_{i+1,j} + 2v_{i,j} - v_{i-1,j} \right\} + \left\{ -v_{i,j+1} + 2v_{i,j} - v_{i,j-1} \right\} = h^2 f_{i,j} \quad (1.3)$$

$$i, j = 1, \dots, N-1,$$

and equation (1.2) by

$$v_{i,j} = g_{i,j}, \quad \begin{cases} i = 0, N, j = 1, \dots, N-1 \\ j = 0, N, i = 1, \dots, N-1. \end{cases} \quad (1.4)$$

These equations represent a system of  $(N-1)^2$  finite difference equations for the unknowns  $v_{i,j}$  at the interior mesh points, where  $v_{i,j}$  is an approximation to  $u(x_i, y_j)$  and  $x_i = ih$ ,  $y_j = jh$ . Isaacson and Keller (1966) show that (1.3) and (1.4) have a unique solution for any  $f, g$  and  $h$ . They also show that the finite difference approximations converge uniformly to the solution of (1.1) as  $h \rightarrow 0$ .

We order the unknowns along horizontal mesh lines. Let  $\underline{v}_j$  denote the unknown along the  $j$ th mesh line i.e.

$$\underline{v}_j = \left( v_{1,j}, v_{2,j}, \dots, v_{N-1,j} \right)^T, \quad j=1, \dots, N-1$$

The above finite difference equations can be written in the block tridiagonal form

$$\begin{bmatrix} D & -I_{N-1} & & & \\ -I_{N-1} & D & -I_{N-1} & & \\ & & & & \\ & & -I_{N-1} & D & -I_{N-1} \\ & & & -I_{N-1} & D \end{bmatrix} \begin{bmatrix} \underline{v}_1 \\ \underline{v}_2 \\ \\ \underline{v}_{N-2} \\ \underline{v}_{N-1} \end{bmatrix} = \begin{bmatrix} \underline{b}_1 \\ \underline{b}_2 \\ \\ \underline{b}_{N-2} \\ \underline{b}_{N-1} \end{bmatrix} \quad (1.5)$$

where the right-hand side is obtained from  $f$ ,  $g$  and  $h$ ,  $I_{N-1}$  is the

$(N-1) \times (N-1)$  unit matrix and  $D = 4 I_{N-1} - (L_{N-1} + L_{N-1}^T)$  is tridiagonal with

$$L_{N-1} = \begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ & 1 & 0 & & \\ & & 1 & 0 & \\ & & & 1 & 0 \end{bmatrix}$$

The system of equations (1.5) can be written in the form

$$A \underline{v} = \underline{b} \quad (1.6)$$

where  $A$  is a real symmetric and positive definite irreducible matrix with non-positive off-diagonal elements. The matrix  $A$  can be written as the sum of two matrices  $A = H + V$  where  $H$  and  $V$  arise from the representation of the respective bracketed terms of (1.3). The component of the vector  $H\underline{v}$  corresponding to the mesh point  $(x_i, y_j)$  is given by

$$-v_{i+1,j} + 2v_{i,j} - v_{i-1,j}.$$

From the definition of the matrix  $H$  we see that its entries are generated from discrete central difference approximations to the one-dimensional operator  $-u_{xx}$  along different horizontal mesh lines of  $\Omega$ . Similarly the components of  $V\underline{v}$  correspond to the discretisation of  $-u_{yy}$  along different vertical mesh lines of  $\Omega$ . The matrix  $H$  is block diagonal and after a suitable permutation  $V$  has this form also. We try to exploit the splitting of  $A$  to devise iterative schemes for the solution of (1.6). We may write  $H$  and  $V$  in the forms

$$H = 2I - (L + L^T), \quad V = 2I - (B + B^T),$$

where

$$L = \begin{bmatrix} L_{N-1} & & & & \\ & L_{N-1} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & L_{N-1} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & & & & \\ I_{N-1} & 0 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & I_{N-1} & 0 \end{bmatrix}$$

### 1.3 The A.D.I. Method

From (1.6) we have that

$$(H + V) \underline{v} = \underline{b} ,$$

which can be written as a pair of matrix equations

$$(H + rI) \underline{v} = (rI - V) \underline{v} + \underline{b} ,$$

$$(V + rI) \underline{v} = (rI - H) \underline{v} + \underline{b} ,$$

for any positive scalar  $r$ . This suggests the block iterative method

$$\begin{aligned} (H + rI) \underline{v}^{(n+1)} &= (rI - V) \underline{v}^{(n)} + \underline{b} , \\ (V + rI) \underline{v}^{(n+2)} &= (rI - H) \underline{v}^{(n+1)} + \underline{b} , \quad n \geq 0 , \end{aligned} \quad (1.7)$$

where  $\underline{v}^{(0)}$  is an arbitrary initial approximation and  $r$  is the acceleration parameter. The matrix  $H + rI$  is block diagonal where each block is given by  $(2+r)I_{N-1} - (L_{N-1} + L_{N-1}^T)$ , hence to obtain the new approximation  $\underline{v}_j^{(n+1)}$  on the  $j$ th mesh line we only need to solve a system of  $(N-1)$  linear equations with a symmetric diagonally dominant tridiagonal matrix.

This process can be carried out in an efficient and stable manner by a simple algorithm based on Gaussian elimination. Similarly  $V + rI$  is block diagonal after a suitable permutation.

If we replace  $r$  by  $r_m$  in (1.7) where  $m = \frac{1}{2}(n+2)$  then (1.7) defines the Peaceman-Rachford implicit alternating direction iterative method. The positive constants  $r_m$  are chosen so as to make the convergence of this process rapid. The vector  $\underline{v}^{(n+1)}$  is treated as an auxiliary vector which is not retained from one iteration to the next.

We can investigate the convergence properties of the scheme (1.7) by eliminating the auxiliary vector  $\underline{v}^{(n+1)}$  to obtain

$$\underline{v}^{(n+2)} = T(r_m) \underline{v}^{(n)} + g(r_m) , \quad n \geq 0 , \quad (1.8)$$

where  $m = \frac{1}{2}(n+2)$  and  $T(r)$  is the Peaceman-Rachford A.D.I. matrix given by

$$T(r) = (V + rI)^{-1} (rI - H) (H + rI)^{-1} (rI - V) \quad (1.9)$$

Let  $\underline{e}^{(n)} = \underline{v}^{(n)} - \underline{u}$ , then  $\underline{e}^{(n+2)} = T(r_m) \underline{e}^{(n)}$  and hence

$$\underline{e}^{(n)} = \left( \prod_{j=1}^{m-1} T(r_j) \right) \underline{e}^{(0)}, \quad n \geq 0. \quad (1.10)$$

The success of this A.D.I. scheme lies in being able to vary the acceleration parameter  $r_j$  from one iteration to the next. We first consider the simple case where all the parameters  $r_j$  are equal to the fixed constant  $r > 0$ . We define the matrix  $T^*(r)$  by

$$T^*(r) = (V+rI) T(r) (V+rI)^{-1} \quad (1.11)$$

then by similarity  $T(r)$  and  $T^*(r)$  have the same eigenvalues. Eliminating  $T(r)$  from (1.11) using (1.9) we can express  $T^*(r)$  in the form

$$T^*(r) = \left\{ (rI-H)(H+rI)^{-1} \right\} \left\{ (rI-V)(V+rI)^{-1} \right\} \quad (1.12)$$

If we denote the spectral radius of  $T(r)$  by  $\rho(T(r))$ , then using elementary norm properties we have

$$\begin{aligned} \rho(T(r)) &= \rho(T^*(r)) \leq \| T^*(r) \| \\ &\leq \| (rI-H)(H+rI)^{-1} \| \| (rI-V)(V+rI)^{-1} \| \end{aligned} \quad (1.13)$$

To bound the norms of the product matrices of (1.13), we recall that  $H$  and  $V$  are symmetric and positive definite. So using the spectral matrix norm we obtain

$$\| (rI-H)(H+rI)^{-1} \|_2 = \max_{1 \leq j \leq M} \left| \frac{r - \lambda_j}{r + \lambda_j} \right| < 1,$$

where  $\lambda_j, 1 \leq j \leq M$ , are the (positive) eigenvalues of  $H$  and  $M$  is the order of  $A$ . The same argument can be applied to the corresponding matrix product involving  $V$ , and hence  $\rho(T(r)) < 1$  for all  $r > 0$ . So we have shown that the scheme given by (1.7) is convergent for all  $r > 0$ .

Varga (1962) shows that the spectral radius,  $\rho(T(r))$ , of the A.D.I. matrix for the model problem can be minimized as a function of  $r$ . The optimum parameter is

$$r = 4 \sin (\pi/2N) \cos (\pi/2N) \quad (1.14)$$

which gives

$$\min_{r>0} \rho(T(r)) = \rho(T(\hat{r})) = \left( \frac{1 - \tan(\pi/2N)}{1 + \tan(\pi/2N)} \right)^2 \quad (1.15)$$

It turns out that  $\rho(T(\hat{r})) = \rho(L(\hat{\omega}))$ , the optimized spectral radius for the point S.O.R. method. Hence the two schemes have identical asymptotic rates of convergence for all  $h > 0$  for the model problem. However, it should be pointed out that for the model problem the A.D.I. iteration involves far more computation. Therefore, it is necessary to vary the acceleration parameter  $r_j$  for A.D.I. to be implemented efficiently.

The following theorem states under which conditions we can guarantee the convergence of the A.D.I. scheme for discretisations of more general partial differential equations.

#### Theorem 1.3.1

Suppose that  $A$  can be written out as the sum of two matrices  $A = H + V$  where  $H$  and  $V$  are symmetric, non-negative definite matrices and at least one of  $H$  and  $V$  is positive definite. Then, for any  $r > 0$ , the Peaceman-Rachford A.D.I. iterative scheme is convergent.

#### Proof

See Varga (1962)

The Peaceman-Rachford iterative method is just one of many A.D.I. schemes. Details of these are given in the texts mentioned in the introduction to this chapter. We describe the scheme due to Douglas and Rachford (1956) since it will be used later for biharmonic problems.

The second equation of the Peaceman-Rachford method is

$$(V + rI) \underline{v}^{(n+2)} = (rI - H) \underline{v}^{(n+1)} + \underline{b}, \quad n \geq 0. \quad (1.16)$$

We can write (1.16) in a form in which the matrix  $H$  does not appear explicitly. We use the first equation of (1.7) to eliminate  $H \underline{v}^{(n+1)}$  from (1.16) to obtain

$$(V+rI)\underline{v}^{(n+2)} = (V-rI)\underline{v}^{(n)} + 2r\underline{v}^{(n+1)}, \quad n \geq 0. \quad (1.17)$$

This could be derived from the identity

$$(V+rI)\underline{u} = (V-rI)\underline{u} + 2r\underline{u}. \quad (1.18)$$

So the Douglas-Rachford method is defined to be the first equation of (1.7) together with

$$(V+rI)\underline{v}^{(n+2)} = V\underline{v}^{(n)} + r\underline{v}^{(n+1)} \quad (1.19)$$

After a little manipulation we find that the Douglas-Rachford A.D.I. matrix,  $W(r)$ , is given by

$$W(r) = (V+rI)^{-1}(H+rI)^{-1}(HV+r^2I).$$

### Theorem 1.3.2

Let  $H$  and  $V$  be symmetric, non-negative definite matrices, where at least one of the matrices  $H$  and  $V$  is positive definite. Then, for any  $r > 0$ , the Douglas-Rachford A.D.I. scheme is convergent.

### Proof

See Varga (1962)

## 1.4 A.D.I. Parameter Sequences

We assume that  $H$  and  $V$  commute i.e.  $HV = VH$ . This property is equivalent to  $H$  and  $V$  having a common set of orthonormal eigenvectors, and is important in determining the best values of  $r_j$ .

### Theorem 1.4.1

Let  $H$  and  $V$  be symmetric matrices. Then, there exists an orthonormal basis of eigenvectors  $\{\underline{\alpha}_i\}_{i=1}^M$  with  $H\underline{\alpha}_i = \lambda_i \underline{\alpha}_i$  and  $V\underline{\alpha}_i = \mu_i \underline{\alpha}_i$  for  $1 \leq i \leq M$  if and only if  $HV = VH$ .

### Proof

See Varga (1962)

Let  $\{\underline{\alpha}_i\}_{i=1}^M$  be an orthonormal basis of eigenvectors of  $H$  and  $V$ .

Consider  $m$  iterations of the Peaceman-Rachford method. We can easily show that

$$\left( \prod_{j=1}^m T(r_j) \right) \underline{\alpha}_i = \left( \prod_{j=1}^m \left( \frac{r_j - \lambda_i}{r_j + \lambda_i} \right) \left( \frac{r_j - \mu_i}{r_j + \mu_i} \right) \right) \underline{\alpha}_i \quad (1.20)$$

for  $1 \leq i \leq M$ . It follows that  $\prod_{j=1}^m T(r_j)$  is symmetric and therefore

$$\left\| \prod_{j=1}^m T(r_j) \right\|_2 < 1,$$

which establishes the convergence of the iteration. If all the eigenvalues of  $H$  and  $V$  are known the problem of finding a parameter sequence reduces to the following minimax problem,

$$\text{Minimize} \quad \max_{1 \leq i \leq M} \prod_{j=1}^m \left| \frac{r_j - \lambda_i}{r_j + \lambda_i} \right| \left| \frac{r_j - \mu_i}{r_j + \mu_i} \right|$$

However, in practice the eigenvalues of  $H$  and  $V$  are not known. We now assume that we can estimate bounds for the eigenvalues  $\lambda_i$  and  $\mu_i$  of the positive definite matrices  $H$  and  $V$  i.e.

$$0 < \alpha \leq \lambda_i, \mu_i \leq \beta, \quad 1 \leq i \leq M.$$

In this case we have that

$$\begin{aligned} \max_{1 \leq i \leq M} \prod_{j=1}^m \left| \frac{r_j - \lambda_i}{r_j + \lambda_i} \right| \left| \frac{r_j - \mu_i}{r_j + \mu_i} \right| &\leq \left( \max_{1 \leq i \leq M} \prod_{j=1}^m \left| \frac{r_j - \lambda_i}{r_j + \lambda_i} \right| \right) \left( \max_{1 \leq i \leq M} \prod_{j=1}^m \left| \frac{r_j - \mu_i}{r_j + \mu_i} \right| \right) \\ &\leq \left( \max_{\alpha \leq x \leq \beta} \prod_{j=1}^m \left| \frac{r_j - x}{r_j + x} \right| \right)^2. \end{aligned}$$

If we let

$$g_m(x; r_j) = \prod_{j=1}^m \left( \frac{r_j - x}{r_j + x} \right),$$

then we obtain the following minimax problem. If  $S_m$  is the set of all functions  $g_m(x; r_j)$ , where  $r_1, \dots, r_m$  are positive or non-negative real numbers, then let



$$d_m(\alpha, \beta) = \min_{g_m \in S_m} \left\{ \max_{\alpha \leq x \leq \beta} |g_m(x; r_j)| \right\}$$

The details of the solution to this minimax problem are given in Wachspress (1962). The problem of finding an optimum parameter sequence that will give the best convergence for a particular problem is difficult. Varga (1962) obtains approximations for the parameters as well as the optimum value of  $m$ . The acceleration parameters are given by

$$r_j = \alpha(\beta/\alpha)^{(2j-1)/2m}, \quad j = 1, \dots, m, \quad (1.21)$$

where  $m \geq 1$  is the smallest integer given by

$$(\sqrt{2} - 1)^{-2m} \leq \beta/\alpha. \quad (1.22)$$

For the model problem, the bounds for the eigenvalues of the matrices  $H$  and  $V$  are

$$\alpha = 4 \sin^2 (\pi/2M)$$

and

$$\beta = 4 \cos^2 (\pi/2M).$$

The theory in this section is based on the assumption that  $H$  and  $V$  commute. In practice this is very restrictive and requires that the region  $\Omega$  be rectangular with sides parallel to the co-ordinate axes and that the differential equation be of the form

$$-\frac{\partial}{\partial x} \left( f_1(x) \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left( f_2(y) \frac{\partial u}{\partial y} \right) + \sigma u = s(x, y), \quad (1.23)$$

where  $f_1, f_2 > 0$  on  $\Omega \cup \Gamma$  and  $\sigma$  is a positive constant. However, in practice good results are obtained for some problems which do not satisfy these assumptions.

### 1.5 D.A.D.I. Philosophy

An alternative way of deriving the Peaceman-Rachford A.D.I. scheme for solving (1.1) is to first convert this equation to the parabolic equation

$$\frac{\partial u}{\partial t} = \nabla^2 u + f, \quad (1.24)$$

whose steady state solution, if one exists, solves (1.1). Equation (1.24) is discretised in time with step size  $\Delta t$ . On odd numbered time steps the right-hand side of (1.24) is discretised implicitly in the x-direction and explicitly in the y-direction,

$$\underline{v}^{(n+1)} - \underline{v}^{(n)} = \frac{\Delta t}{h^2} \left\{ -H\underline{v}^{(n+1)} - V\underline{v}^{(n)} + \underline{b} \right\} \quad (1.25)$$

On even numbered time steps the right-hand side of (1.24) is discretised implicitly in the y-direction and explicitly in the x-direction,

$$\underline{v}^{(n+2)} - \underline{v}^{(n+1)} = \frac{\Delta t}{h^2} \left\{ -H\underline{v}^{(n+1)} - V\underline{v}^{(n+2)} + \underline{b} \right\} \quad (1.26)$$

Equations (1.25) and (1.26) make one double sweep of the A.D.I. iteration. Thus we obtain the same scheme as (1.7) with  $r = h^2/\Delta t$ . A similar scheme can be derived for the more general equation (1.23).

The dynamic A.D.I. (D.A.D.I.) method (see Doss and Miller (1979)) involves a computerized strategy for the automatic change of the iteration parameter  $\Delta t$  in A.D.I. methods. Doss and Miller give the following reasons for developing an automatic step size changer:

- (i) to keep  $\Delta t$  within a region of fast convergence,
- (ii) to recognize instabilities as they start to occur and bypass them by decreasing  $\Delta t$ ,
- (iii) to avoid the necessity of choosing a priori iteration parameters.

The D.A.D.I. approach is to pass from  $\underline{v}^{(n)}$  to  $\underline{v}^{(n+4)}$  by two successive sweeps of A.D.I. with step size  $\Delta t$ ; one then backs up and, for bookkeeping purposes, passes from  $\underline{v}^{(n)}$  to  $\tilde{\underline{v}}^{(n+4)}$  by a single A.D.I. double sweep with step size  $2\Delta t$ . The test parameter is then computed

$$TP = \frac{\left\| \underline{v}^{(n+4)} - \tilde{\underline{v}}^{(n+4)} \right\|}{\left\| \underline{v}^{(n+4)} - \underline{v}^{(n)} \right\|} \quad (1.27)$$

using the  $\ell_2$ -norm. TP is an estimate of the relative local truncation error. If we are interested in solving the parabolic equation accurately then  $\Delta t$  will be small, and so will TP. We wish to accelerate convergence and attempt to push TP into an interval where convergence for A.D.I. is rapid. This will be discussed in more detail in Section 1.6. We accept the present step and move on to the next step with  $\Delta t$  increased, unchanged or decreased depending upon whether TP is too small, alright or too large. If TP is much too large then we reject the present step and start again at  $\underline{v}^{(n)}$  with  $\Delta t$  much reduced.

### 1.6 Analysis and D.A.D.I. Strategy

We assume that  $H$  and  $V$  are symmetric, positive definite and commuting matrices. These assumptions are required for the following analysis. However, this strategy appears to give good results in situations beyond the realm of these assumptions. With these assumptions we can analyse the convergence factor by which each double sweep of A.D.I. damps each eigen-component of the error. In the following analysis we assume that the error at each step is concentrated mainly in a single eigenspace. So at the  $n$ th step we assume that the error  $\underline{e}^{(n)} = \underline{v}^{(n)} - \underline{u}$  is an eigenfunction for  $H$  and  $V$  corresponding to eigenvalues  $a$  and  $b$  respectively.

The error  $\underline{e}^{(n)}$  also satisfies equations (1.25) and (1.26) but with  $\underline{b}$  set equal to zero. Thus we can easily compute the convergence factor CF and test parameter TP as functions of  $a \Delta t$  and  $b \Delta t$ . The convergence factor for a single A.D.I. double sweep is given by

$$\underline{e}^{(n+2)} = \frac{(1 - a \Delta t)(1 - b \Delta t)}{(1 + a \Delta t)(1 + b \Delta t)} \underline{e}^{(n)} \quad (1.28)$$

If we define

$$R(a \Delta t, b \Delta t) = \frac{(1 - a \Delta t)(1 - b \Delta t)}{(1 + a \Delta t)(1 + b \Delta t)} \quad (1.29)$$

then we have that

$$\underline{e}^{(n+2)} = R(a \Delta t, b \Delta t) \underline{e}^{(n)} \quad (1.30)$$

Hence the convergence factor CF for two double sweeps of A.D.I. and the test parameter TP from (1.27) become

$$CF = \|\underline{e}^{(n+4)}\| / \|\underline{e}^{(n)}\| = R^2(a \Delta t, b \Delta t), \quad (1.31)$$

and

$$TP = \left| \frac{R^2(a \Delta t, b \Delta t) - R(2a \Delta t, 2b \Delta t)}{R^2(a \Delta t, b \Delta t) - 1} \right| \quad (1.32)$$

We wish to investigate graphically whether it is possible, for all values of  $a$  and  $b$ , to make CF reasonably small by changing  $\Delta t$  to keep TP within a certain range. We can normalize  $a$  and  $b$  by assuming  $b \leq a = 1$ . We then consider the computer generated graphs of  $(CF)^{\frac{1}{2}}$  and TP as functions of  $\Delta t$  for different values of  $b$ . Two of these graphs, on a log-log scale, are shown in Figures 1.1 and 1.2, for  $b=1$  and  $b=0.1$ . Doss and Miller (1979) consider many other values of  $b$ . We notice that  $(CF)^{\frac{1}{2}}$  is quite small provided that TP lies in the range  $(0.1, 0.3]$ . These graphs allow us to investigate the effect upon CF and TP of halving, doubling, etc., of  $\Delta t$ . In each of these figures the intervals  $(0, 0.05]$ ,  $(0.05, 0.1]$ ,  $(0.1, 0.3]$ ,  $(0.3, 0.4]$ ,  $(0.4, 0.6]$ ,  $(0.6, \infty)$  imposed on the vertical TP axis lead to subsets of the TP graph which we have labelled [4], [2], [1], [ $1/2$ ], [ $1/4$ ], reject and [ $1/16$ ], corresponding to the factor by which the strategy will shift  $\Delta t$  for the next step of D.A.D.I.

With this strategy the shifted  $\Delta t$  quite often falls within a region of fast convergence. For example, suppose that  $b=1$  then when  $\Delta t$  takes the value 0.8 we find that  $TP = 0.053$ . From Figure 1.1 (see p.13) we see that the strategy will increase  $\Delta t$  by a factor of 2 for the next D.A.D.I. step. This means that  $\Delta t$  will now have the value 1.6 and so  $TP = 0.273$  which lies in the desired interval  $(0.1, 0.3]$ . This value of  $\Delta t$  yields a convergence factor of  $2.8 \times 10^{-3}$ . We can see from Figures 1.1 and 1.2 (p.13) that if TP lies in the interval  $(0, 0.05]$  it might take several D.A.D.I. steps

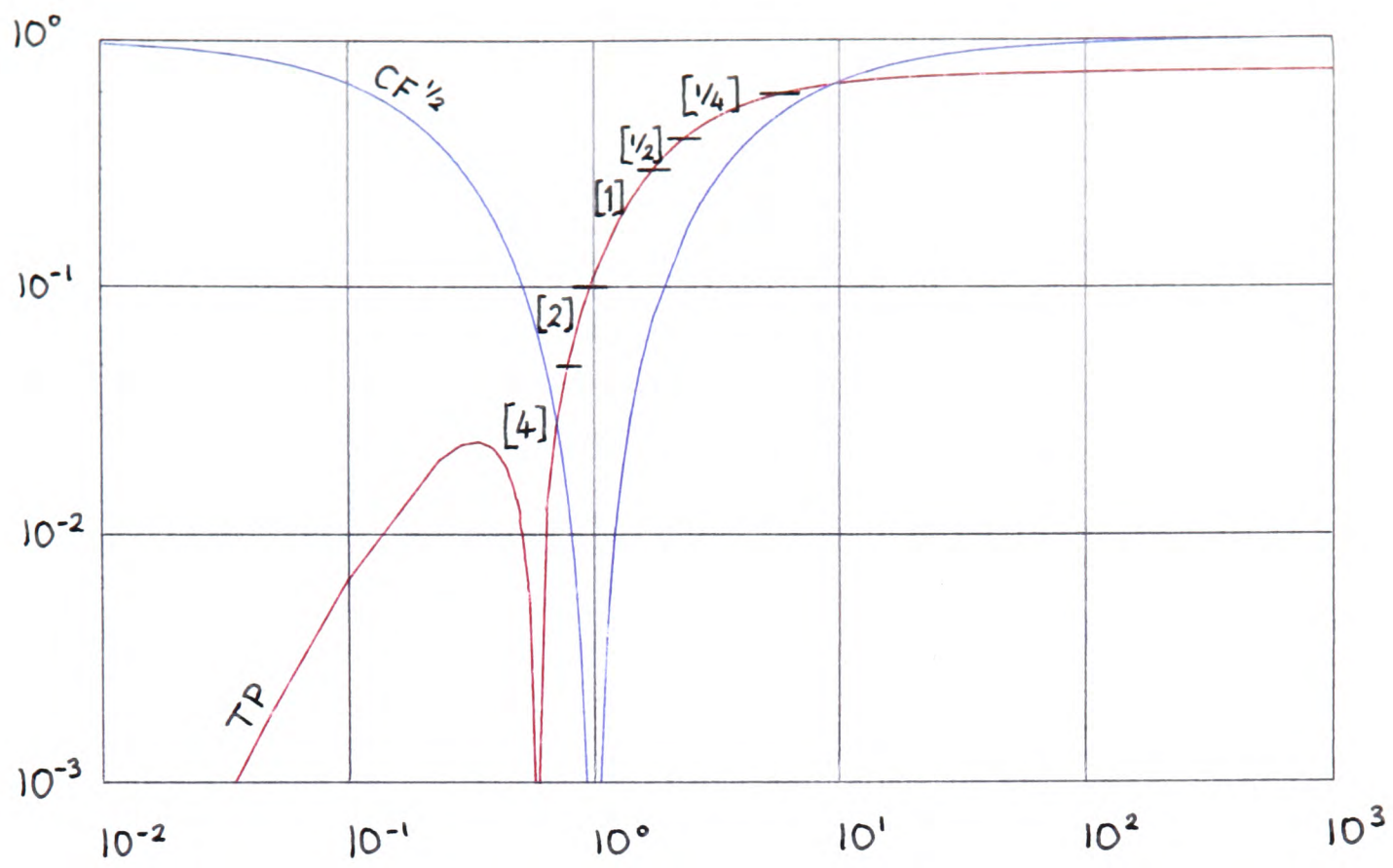


Figure 1.1. TP and  $CF^{\frac{1}{2}}$  vs.  $\Delta t$  for  $b = 1$ .

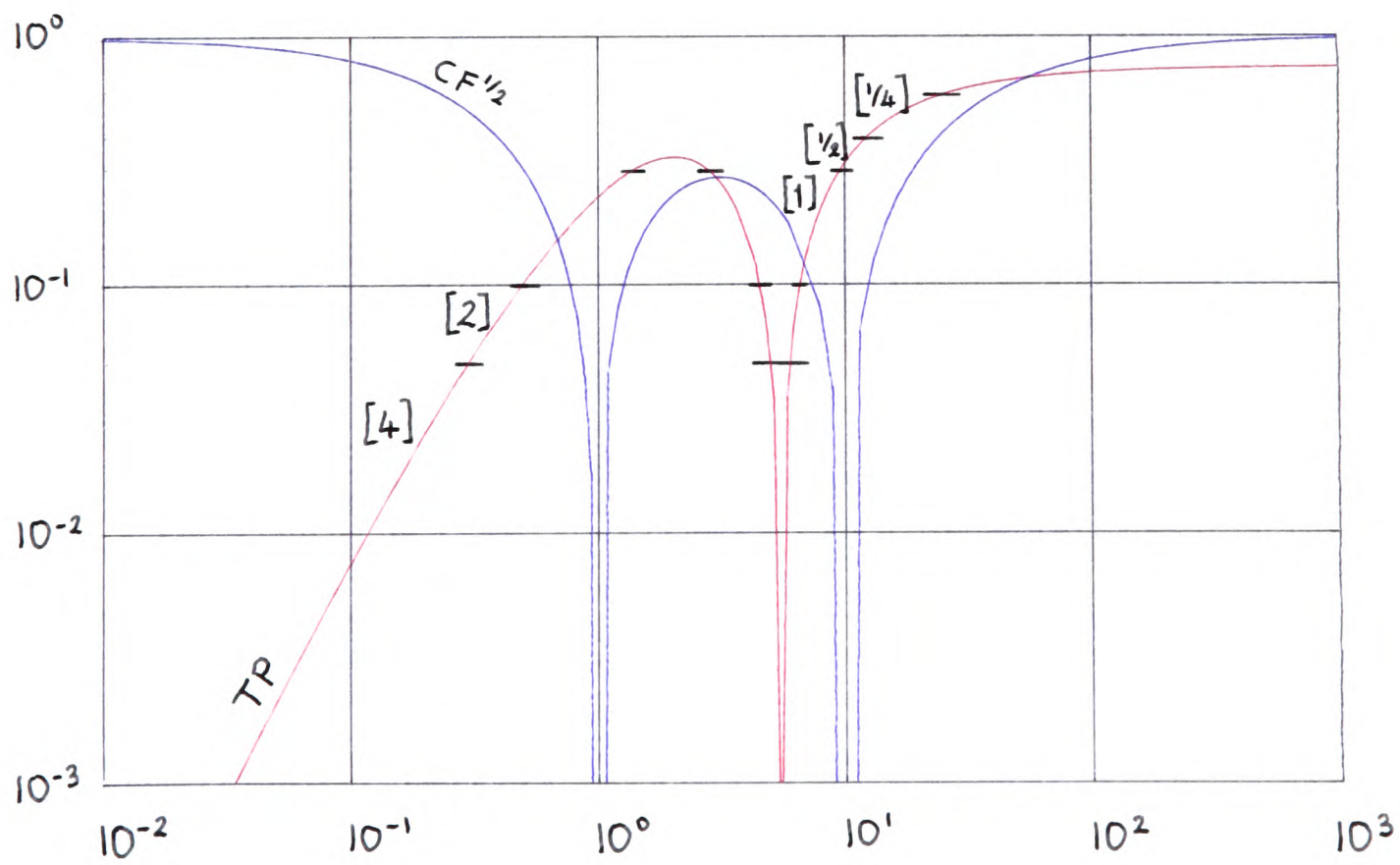


Figure 1.2. TP and  $CF^{\frac{1}{2}}$  vs.  $\Delta t$  for  $b = 0.1$ .

before  $\Delta t$  falls within a region of fast convergence.

This strategy has been designed with a bias towards smaller  $\Delta t$  in the hope of obtaining greater stability. So if TP, which is a measure of the relative local truncation error, becomes too large then the strategy will reject the present step and decrease  $\Delta t$  drastically by the factor  $1/16$ . It is also important not to over-correct in shifting  $\Delta t$ ; for example, factors of 40, 20, 1,  $1/20$ ,  $1/40$ ,  $1/160$  in the strategy, in place of our present 4, 2, 1,  $1/2$ ,  $1/4$ ,  $1/16$  would lead to wild oscillations - throwing  $\Delta t$  first too high, then rejecting and throwing  $\Delta t$  too low, then too high, etc. The aim has been to damp out the higher order components of the error using small  $\Delta t$  before increasing  $\Delta t$  to damp out the lower order components of the error.

This analysis can be extended to the Douglas-Rachford A.D.I. scheme. In this case the convergence factor for a single A.D.I. double sweep is given by

$$\underline{e}^{(n+2)} = \frac{(1 + ab \Delta t^2)}{(1 + a \Delta t)(1 + b \Delta t)} \underline{e}^{(n)}$$

If we define

$$R(a \Delta t, b \Delta t) = \frac{(1 + ab \Delta t^2)}{(1 + a \Delta t)(1 + b \Delta t)},$$

then CF and TP are defined as in (1.31) and (1.32) respectively. The corresponding graphs will be different from those obtained for the Peaceman-Rachford method but they will give rise to a similar strategy.

### 1.7 Numerical Results

We consider Laplace's equation on the unit square with Dirichlet boundary conditions such that the true solution of the problem is

$$u(x,y) = e^{-\pi y} \sin \pi x$$

The problem is solved using the A.D.I. methods described in this chapter and

also by S.O.R. The methods used are:

- (a) ADI1. The A.D.I. method with the single optimized acceleration parameter given by (1.14) i.e.

$$r = 4 \sin (\pi/2N) \cos (\pi/2N)$$

- (b) ADI2. The A.D.I. method with the parameter sequence defined by (1.21) and (1.22).
- (c) DADI. The D.A.D.I. method with the automatic strategy for changing the acceleration parameter.
- (d) SOR. The S.O.R. method with optimum relaxation parameter.

The initial approximation to the solution of the discrete equations at mesh points within the unit square was chosen to be the zero solution. For the D.A.D.I. method the initial time step was chosen to be  $(0.1)h^2$ . The algorithms were terminated when the  $\ell_2$ -norm of the residuals was less than  $10^{-6}$ .

In Table 1.1 we give the number of A.D.I. double sweeps or S.O.R. sweeps for different grid sizes. For the D.A.D.I. method we give the number of D.A.D.I. steps in brackets. In Table 1.2 we give the computational time, in seconds, to reach the convergence criterion. Table 1.3 illustrates the behaviour of the D.A.D.I. method in the case when  $h = 1/64$ . We show how  $\Delta t$  is changed after each step and the way in which the residuals are reduced. In all subsequent tables we use a.b-c as an obvious shorthand for  $a.b \times 10^{-c}$ .

From the results we see that ADI1 and SOR are comparable in terms of the number of sweeps required to attain the convergence criterion as the grid size is reduced. However, much more work is required to perform one double sweep of A.D.I. than one sweep of the S.O.R. iteration. This can be clearly seen by studying the run times for the two methods. The number of double sweeps for ADI2 and DADI to converge to the required tolerance does not increase drastically as the grid size is reduced. These methods are both superior to ADI1 and SOR for this problem.

In Table 1.3 notice the superb reduction in the  $\ell_2$ -norm of the

Method	$h = 1/4$	$h = 1/8$	$h = 1/16$	$h = 1/32$	$h = 1/64$
ADI1	9	20	41	86	182
ADI2	9	20	23	30	37
DADI	10( 4)	16(6)	17(6)	20(7)	29(10)
SOR	12	26	52	107	222

Table 1.1 Number of Sweeps

Method	$h = 1/4$	$h = 1/8$	$h = 1/16$	$h = 1/32$	$h = 1/64$
ADI1	0.4	0.3	0.7	4.0	33.5
ADI2	0.4	0.3	0.5	1.6	7.1
DADI	0.4	0.3	0.4	1.7	6.0
SOR	0.4	0.4	0.6	2.3	13.9

Table 1.2 Computational Time

Step	$\Delta t$	TP	$\ell_2$ -norm of residuals	Change in next $\Delta t$
1	0.24-5	0.033	0.188+3	4
2	0.98-5	0.099	0.651+2	2
3	0.20-4	0.051	0.322+2	2
4	0.39-4	0.039	0.175+2	4
5	0.16-3	0.072	0.686+1	2
6	0.31-3	0.040	0.319+1	4
7	0.13-2	0.050	0.808+0	4
8	0.50-2	0.013	0.119-1	4
9	0.20+0	0.348	0.596-3	$\frac{1}{2}$
10	0.10+0	0.267	0.301-7	1

Table 1.3 D.A.D.I. details for  $h = 1/64$



residuals at step 10 when TP lies in the range  $(0.1, 0.3]$ . In the course of this step the residuals are reduced by a factor of  $0.5 \times 10^{-4}$ .

### 1.8 Remarks

Doss and Miller (1979) also produce results for linear problems with strongly non-constant and non-smooth coefficients and for mesh regions with jagged boundaries. Their results show that the D.A.D.I. method performs well on these problems. They also discuss several ways of implementing D.A.D.I. for nonlinear problems and give results for a wide range of examples. In Chapters 3 and 6 we use the D.A.D.I. method to solve coupled systems of elliptic partial differential equations. Detailed descriptions of the D.A.D.I. algorithms are given in these chapters.

Fast convergence of the A.D.I. method depends on being able to choose a sequence of acceleration parameters. When we move away from model problems it is very difficult, if not impossible, to generate parameter sequences which are near to the optimum ones. For nonlinear equations it is almost impossible since to obtain them we would need to have a good idea of the true solution, which is not likely to be the case in practice. This is one of the disadvantages of standard A.D.I. as opposed to the fully automatic D.A.D.I. method.

## Chapter 2

### MULTIGRID METHODS

#### 2.1 Introduction

Here we are concerned with the multigrid method for solving partial differential equations numerically. The multigrid method is a numerical strategy to solve partial differential equations by switching between finer and coarser levels of discretisation. The characteristic feature of the multigrid method is the combination of a smoothing step and a coarse grid correction. During the smoothing step the residuals are not necessarily decreased but smoothed. In the following correction step the discrete solution is improved by means of an auxiliary equation on a coarser grid. This results in an iterative process that is usually very fast and effective.

Coarse grid acceleration techniques were first recommended and used by Southwell (1935). A multigrid method for solving Poisson's equation in a square was first described by Fedorenko (1961). Bakhvalov (1966) discusses the extension to any second order elliptic equation with variable coefficients. The first multigrid solution for finite element equations is due to Astrachancev (1971). Various multigrid techniques are described by Brandt (1977). Hackbusch (1979) discusses multigrid methods for nonlinear equations.

In this chapter we discuss the philosophy which lies behind the multigrid method and the role played by relaxation in the smoothing process. We look at different types of multigrid algorithm and describe one in detail. We consider the choice of components for the different parts of the multigrid procedure. The smoothing properties of the A.D.I. method are investigated and various multigrid methods are tested and compared on a model problem.

## 2.2 The Multigrid Philosophy

Consider the differential equation

$$Lu(x) = f(x), \quad (2.1)$$

for  $x \in \Omega$  a bounded domain, with  $u(x) = g(x)$  on the boundary  $\partial\Omega$ .

Let  $G_1, G_2, \dots, G_M$  be a sequence of grids approximating the same domain  $\Omega$  with corresponding mesh sizes  $h_1, h_2, \dots, h_M$ . Let  $h_k = 2h_{k+1}$ ,  $k = 1, \dots, M-1$ , then we have that  $G_1 \subset G_2 \subset \dots \subset G_M$ .

The above problem can be discretised on any grid  $G_k$  in the following way

$$L^k u^k(x) = f^k(x), \quad x \in G_k, \quad (2.2)$$

with  $u^k(x) = g^k(x)$  for  $x \in \partial G_k$ . We want to find the solution to this discrete problem on the finest grid  $G_M$ . We try to incorporate the fact that the solution to this problem on a coarser grid  $G_k$  can be used to approximate the solution to the problem on  $G_M$  since it approximates the same differential equation. An advantage of this type of method is that the problem on  $G_k$  is smaller and hence easier and cheaper to solve than the  $G_M$  problem.

Let  $v^M$  be an approximation to the solution of the discrete problem on  $G_M$ , then the residual is given by

$$L^M v^M - f^M = r^M, \quad (2.3)$$

where  $v^M = g^M$  on  $\partial G_M$ . Suppose that  $L$  is linear, then the exact solution of the discrete problem is  $u^M = v^M + w^M$ , where the correction  $w^M$  satisfies the residual equations

$$L^M w^M = -r^M, \quad w^M = 0 \text{ on } \partial G_M \quad (2.4)$$

For the reasons given above it would be useful if solutions to this equation on coarser grids could be used to update the first approximation  $v^M$  to  $u^M$ . This cannot always be done, however, since  $r^M$  may change rapidly on  $G_M$ . If it does, then when (2.4) is solved on a coarser grid these rapidly changing residuals are not picked up and therefore cannot be accurately approximated there. This is illustrated in Figure 2.1.

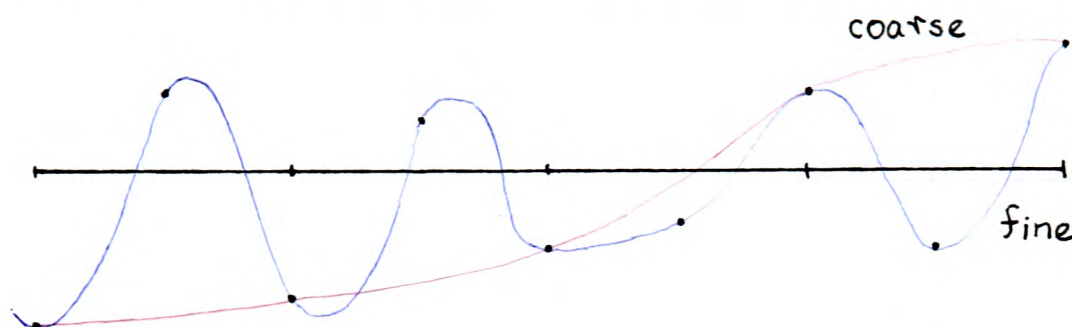


Figure 2.1

So, we attempt to ensure that the residuals are smooth before we approximate (2.3) on coarser grids. An effective means of smoothing residuals is by relaxation methods e.g. Gauss-Seidel relaxation. In practice such methods exhibit fast convergence in the first few iterations with residuals decreasing rapidly from one iteration to the next, but after that the convergence rate slows down. Closer examination reveals that the convergence is fast so long as the residuals are changing rapidly and that as soon as the residuals become smooth, convergence slows down. It is at this point that relaxation sweeps should be stopped and the approximate solution of the residual equations on coarser grids carried out.

### 2.3 Relaxation and Smoothing

The purpose of relaxation in the multigrid process is to smooth the error on the fine grids. The high frequency components of the error are reduced on the fine grids whereas the low frequency components are

reduced by relaxation sweeps on the coarser grids. We assume that the finite difference equations hold in a grid covering the whole space and so we can study the convergence properties of the relaxation procedures by expanding the error in a Fourier series. The rate of smoothing is determined by the behaviour of the high frequency components of the error.

Consider Laplace's equation as an example

$$Lu \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (2.5)$$

The standard five-point finite difference approximation to (2.5) is given by

$$\frac{u_{r+1,s} - 2u_{r,s} + u_{r-1,s}}{h^2} + \frac{u_{r,s+1} - 2u_{r,s} + u_{r,s-1}}{h^2} = 0. \quad (2.6)$$

Suppose that we wish to solve (2.6) using the Gauss-Seidel iterative method with the points ordered in the lexicographic manner. Let  $v_{r,s}$  and  $v_{r,s}^*$  be the approximations to  $u_{r,s}$  before and after a relaxation sweep, then these values satisfy the following equation

$$\frac{v_{r+1,s} - 2v_{r,s}^* + v_{r-1,s}^*}{h^2} + \frac{v_{r,s+1} - 2v_{r,s}^* + v_{r,s-1}^*}{h^2} = 0. \quad (2.7)$$

We define the convergence factor,  $\mu$ , by

$$\mu = \|w^*\|_2 / \|w\|_2,$$

where

$$w_{r,s} = u_{r,s} - v_{r,s}, \quad w_{r,s}^* = u_{r,s} - v_{r,s}^*.$$

To study the  $\theta = (\theta_1, \theta_2)$  Fourier component of the error function  $w$  and  $w^*$  before and after the relaxation sweep put

$$w_{r,s} = A_\theta e^{i(r\theta_1 + s\theta_2)}, \quad w_{r,s}^* = B_\theta e^{i(r\theta_1 + s\theta_2)} \quad (2.8)$$

Subtracting (2.7) from (2.6) we obtain

$$w_{r+1,s} - 2w_{r,s}^* + w_{r-1,s}^* + w_{r,s+1} - 2w_{r,s}^* + w_{r,s-1}^* = 0 \quad (2.9)$$

Inserting (2.8) into (2.9) we get

$$(e^{i\theta_1} + e^{i\theta_2}) A_\theta + (e^{-i\theta_1} + e^{-i\theta_2} - 4) B_\theta = 0.$$

So the convergence factor of the  $\Theta$  component is

$$\mu(\Theta) = \left| \frac{B_\Theta}{A_\Theta} \right| = \left| \frac{e^{i\Theta_1} + e^{i\Theta_2}}{4 - e^{-i\Theta_1} - e^{-i\Theta_2}} \right| \quad (2.10)$$

The smoothing factor is defined to be

$$\bar{\mu} = \max_{\frac{1}{2}\pi \leq |\Theta| \leq \pi} \mu(\Theta)$$

where  $|\Theta| = \max(|\Theta_1|, |\Theta_2|)$ . The range  $\frac{1}{2}\pi \leq |\Theta| \leq \pi$  is the set of high frequency components which are not determined by the low frequency components. We can easily show that for (2.10)

$$\bar{\mu} = \mu(\frac{1}{2}\pi, \cos^{-1}(4/5)) = 0.5$$

So by performing local mode analysis i.e. analysis of the Fourier components we are able to determine the smoothing factor for any given difference equation with any given relaxation scheme.

Suppose  $\Omega = [0,1] \times [0,1]$  and  $N = 8$  where  $Nh = 1$ . The components that are present on this grid are

$$\Theta = (p\pi/N, q\pi/N), \quad p, q = \pm 1, \dots, \pm 7.$$

The size of  $\mu(\Theta)$  for various typical values of  $\theta_1$  and  $\theta_2$  is indicated below

$$\begin{aligned} \mu(\frac{1}{2}\pi, \frac{1}{2}\pi) &= 0.45, \quad \mu(\frac{1}{4}\pi, \frac{1}{4}\pi) = 0.68, \\ \mu(\frac{1}{4}\pi, -\frac{1}{4}\pi) &= 0.55, \quad \mu(\frac{1}{8}\pi, \frac{1}{8}\pi) = 0.77, \\ \mu(\frac{1}{2}\pi, \pi) &= 0.28, \quad \mu(\frac{1}{2}\pi, \frac{3}{4}\pi) = 0.37 \end{aligned}$$

If the problem under consideration is nonlinear or if its coefficients are non-constant then  $\mu$  may well vary over the domain. In this case, we use the fact that smoothing is a local property to compute  $\bar{\mu}$  by linearizing the difference equations about some approximate solution and freezing the coefficients of the linearized equations at local values.

If S.O.R. was used to solve the finite difference equations (2.6) we could show that no  $\omega \neq 1$  yields a smoothing rate better than  $\omega = 1$ . Here  $\omega$  is the S.O.R. relaxation parameter and the case  $\omega = 1$  corresponds to the Gauss-Seidel method. For other equations  $\omega = 1$  is not necessarily optimal, but its  $\bar{\mu}$  is not significantly larger than the minimal  $\bar{\mu}$ .

## 2.4 Multigrid Algorithms

Our aim is to compute an approximation to  $u^M$ , the solution of (2.2), on the finest grid. On coarser grids ( $k < M$ ) we solve a modified version of (2.2) by changing the right-hand side. We shall denote by  $f^k$  the right-hand side of the  $k$ -th level equation. This will depend on  $v^{k+1}$ , the current approximation on the  $(k+1)$ -th level. We shall denote by  $v^k$  the computed approximation to this modified equation. Let  $I_k^\ell$  denote some form of transfer ( $k > \ell$ ) or interpolation ( $k < \ell$ ) of values between grids.

There are several types of multigrid algorithm in current use. We discuss in detail the accommodative Full Approximation Storage (F.A.S.) cycling algorithm of Brandt (1977). This algorithm can be used to solve nonlinear as well as linear problems. In accommodative algorithms the decision to switch to coarser levels or back to finer levels depends on internal checks which are related to the local mode analysis. Cycling algorithms start when a first approximation is given on the finest grid and then proceed to cycle between different levels of discretization.

In the F.A.S. scheme the residual equation on the finest grid is

$$L^M u^M - L^M v^M = -r^M, \quad (2.11)$$

where  $u^M - v^M = w^M$ . This equation differs from (2.4) in that here we do not assume that  $L$  is linear. We want to find an approximate solution to (2.11) on  $G_{M-1}$  in order to update the approximation  $v^M$  to  $u^M$  on the finest grid  $G_M$ . So on  $G_{M-1}$  we have

$$L^{M-1} u^{M-1} - L^{M-1} I_M^{M-1} v^M = -I_M^{M-1} r^M. \quad (2.12)$$

Equation (2.12) can be written in the form

$$L^{M-1} u^{M-1} = f^{M-1}, \quad (2.13)$$

where

$$f^{M-1} = L^{M-1} I_M^{M-1} v^M - I_M^{M-1} (L^M v^M - f^M). \quad (2.14)$$

If  $v^{M-1}$  is an approximation to the solution of (2.13) then the coarse grid correction to  $v^M$ , the current approximation to  $u^M$  on  $G_M$ , is

$$v^{M-1} = I_M^{M-1} v^M.$$

This procedure is easily extended to the case where we have a sequence of coarser grids. The steps of the accommodative F.A.S. cycling algorithm, which are flowcharted in Figure 2.2, are set out below.

#### Algorithm 2.4.1

(a) Starting on the finest grid. We start at the finest level of discretization and introduce  $v^M$  the given first approximation to  $u^M$ . Set  $k=M$ , our initial working level, and choose  $\epsilon_M$  to be a suitable tolerance.

(b) Starting a new working level  $k$ . Put  $\bar{e}_k = +\infty$  (or a very large number).

(c) Relaxation sweep. We improve  $v^k$  by performing one relaxation sweep.

At the same time we compute some norm  $e_k$  of the residuals. Dynamic residuals are usually computed since in schemes like the Gauss-Seidel iteration they are almost calculated as part of the relaxation and so are inexpensive to compute.

(d) Testing convergence and its rate. If relaxation at the current working level has sufficiently converged i.e.  $e_k \leq \epsilon_k$ , go to Step (f). If not, and if convergence is still fast i.e.  $e_k \leq \eta \bar{e}_k$ , where  $\eta$  is some fixed parameter, set  $\bar{e}_k = e_k$  and go to Step (c) to perform another relaxation sweep. If the convergence rate is slow i.e.  $e_k > \eta \bar{e}_k$  and our working level is not the coarsest go to Step (e). If  $k=1$  and  $e_k > \eta \bar{e}_k$  then go to Step (c). The slow convergence is an indication that the error is smooth and that the residual equations can now be solved on a coarser grid.

(e) Transfer to coarser level. Decrease  $k$  by 1. On the new level  $k$  we introduce as our first approximation the function

$$v^k = I_{k+1}^k v^{k+1} \quad (2.15)$$

The right-hand side for the new level is defined by

$$f^k = L^k v^k + I_{k+1}^k (f^{k+1} - L^{k+1} v^{k+1}). \quad (2.16)$$



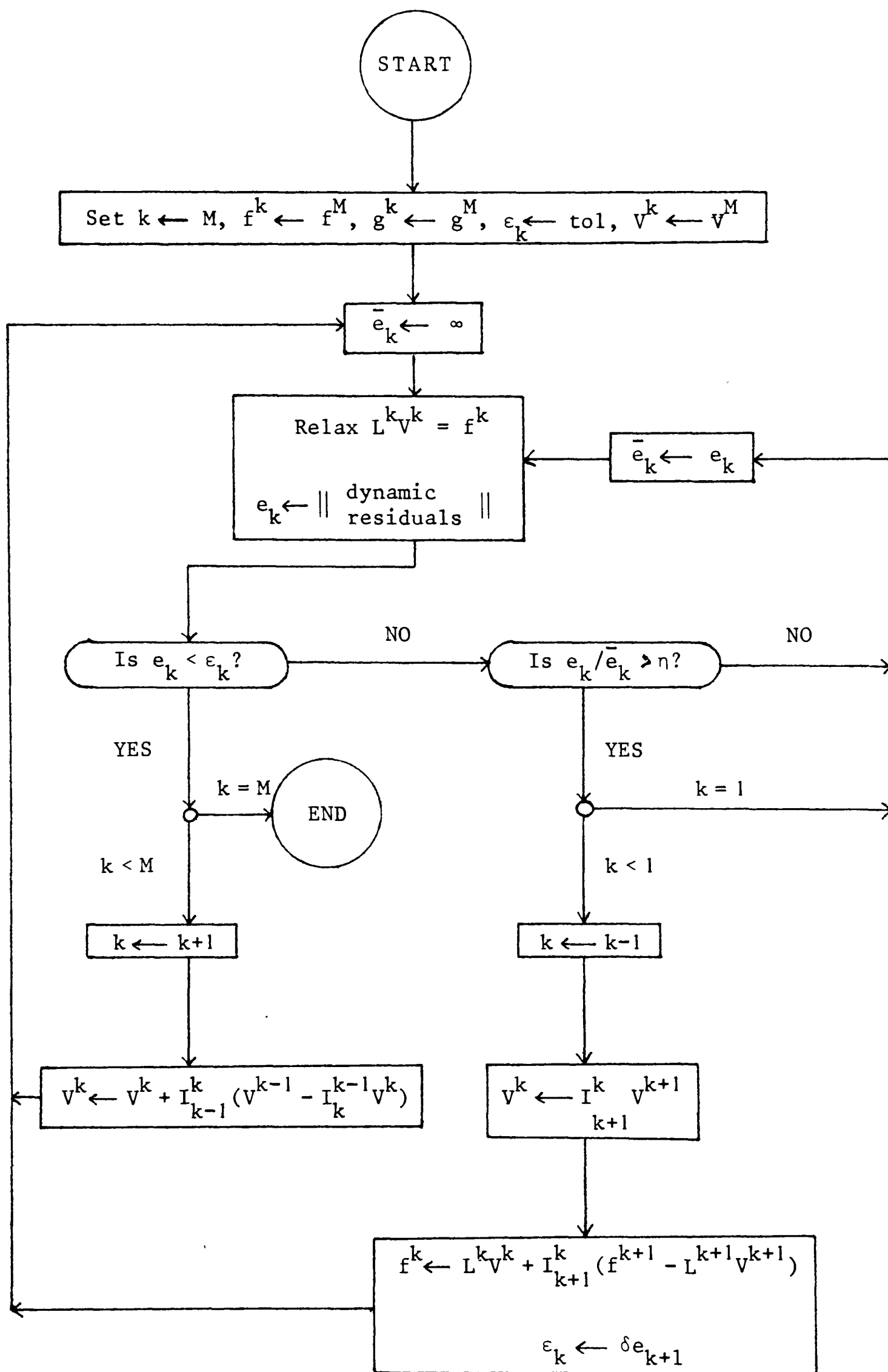


Figure 2.2

Set  $\varepsilon_k = \delta e_{k+1}$  to be the tolerance for this problem where  $\delta$  is also some parameter. Go to step (b).

(f) Correction of the solution on the finer level. If  $k = M$ , then convergence has been obtained on the finest level to the desired tolerance and so the algorithm is terminated. If  $k < M$ , then we are ready to correct  $v^{k+1}$ . Put

$$v^{k+1} = v^{k+1} + I_k^{k+1} (v^k - I_{k+1}^k v^{k+1}) \quad (2.17)$$

where the  $v_{k+1}^k$ 's on the right-hand side are the previous approximations on the level  $k+1$  and  $I_{k+1}^k$  is the same operator as the one used in Step (e). Increase  $k$  by 1 and go to Step (c).

#### Choice of parameters

The choice for the switching parameter  $\eta$  is usually  $\bar{\mu}$ , the smoothing factor per sweep. In practice we see that the choice of  $\eta$  and that of  $\delta$  is not critical in terms of efficiency of the algorithm. A safe choice for  $\eta$  to ensure efficiency of the coarse grid corrections is  $\bar{\mu}^{\frac{1}{2}}$ . Since the coarse grid solution is used to correct the fine grid solution, its residuals should be smaller than those on the fine grid. Our choice for  $\delta$  is 0.2 since there is no point having them much smaller.

For linear problems the Correction Storage (C. S.) scheme can be used instead of the F.A.S. scheme. The basic difference is that in the C.S. scheme the correction is stored on coarser grids and not the coarse grid approximation. The following changes have to be made to Algorithm 2.4.1 to convert it for the C.S. scheme:

(i) In Step (e) we replace (2.15) and (2.16) by

$$v^k = 0$$

$$f^k = I_{k+1}^k (f^{k+1} - L^{k+1} v^{k+1}).$$

So we set to zero our first approximation to the correction for  $v^{k+1}$ . The

right-hand side for the new level is some transfer of residuals from the next finer level.

(ii) In Step (f) we replace (2.17) by

$$v^{k+1} = v^{k+1} + I_k^{k+1} v^k.$$

The F.A.S. scheme can be used for general nonlinear problems. In this case we can use the same relaxation and interpolation routines at all levels. In the linear case the F.A.S. scheme is slightly less efficient than the C.S. scheme since more work is required to compute the modified right-hand sides (2.16). Brandt (1977) discusses how the F.A.S. scheme can be used to refine the solution in a subdomain of the problem. This is particularly useful if the problem has a boundary layer of a singularity.

## 2.5 Multigrid Components

The choice of components in the multigrid method depends on the problem we wish to solve. In this section we look briefly at some of the choices that are used later.

### (i) Smoothing

We can investigate the smoothing properties of a given scheme by local mode analysis. So we can determine which schemes are suitable for certain classes of problems.

(a) pointwise-relaxation. Among such methods are those of Jacobi,

Gauss-Seidel and S.O.R. with various orderings of the grid points:

lexicographic, chequer-board and four-coloured ordering. These methods are suitable for solving Poisson-like equations.

(b) block relaxation. Examples of these methods are line and alternating line relaxation schemes. These work well for anisotropic operators, i.e.

$$Lu = \varepsilon u_{xx} + u_{yy} \quad \text{where } \varepsilon \neq 1.$$

(c) collective relaxation. These methods are used to solve systems of differential equations. At each grid point we relax simultaneously the different difference schemes.

(d) incomplete LU-decomposition. For a full account of how this smoothing process is integrated into the multigrid method the reader is referred to Wesseling and Sonneveld (1979). It works particularly well for anisotropic operators.

(e) A.D.I. The smoothing properties of this method will be investigated later for Poisson's equation.

(ii) Fine-to-coarse grid transfer

(a) injection. This is a simple transfer of values from fine grid points which are also coarse grid points. The operator  $I_k^{k-1}$  is defined by

$$I_k^{k-1} V_{i,j}^k = V_{2i,2j}^k$$

(b) half-weighting. This is a non-trivial weighting of values which is necessary for nonlinear equations and for difference equations with rapidly varying coefficients. In relaxations sweeps for such equations, the slowly converging low frequencies are coupled with high frequency modes. These high frequency modes may be large in the residual and so a suitable weighting is chosen that will eliminate these troublesome components. For an analysis and fuller description of various weightings see Brandt (1977). For half-weighting the operator  $I_k^{k-1}$  is defined by

$$I_k^{k-1} V_{i,j}^k = \frac{1}{2} V_{2i,2j}^k + \frac{1}{8} (V_{2i-1,2j}^k + V_{2i+1,2j}^k + V_{2i,2j-1}^k + V_{2i,2j+1}^k).$$

This weighting is illustrated in Figure 2.3. The coarse grid is represented by the unbroken lines and the fine grid by the unbroken and the dotted lines.

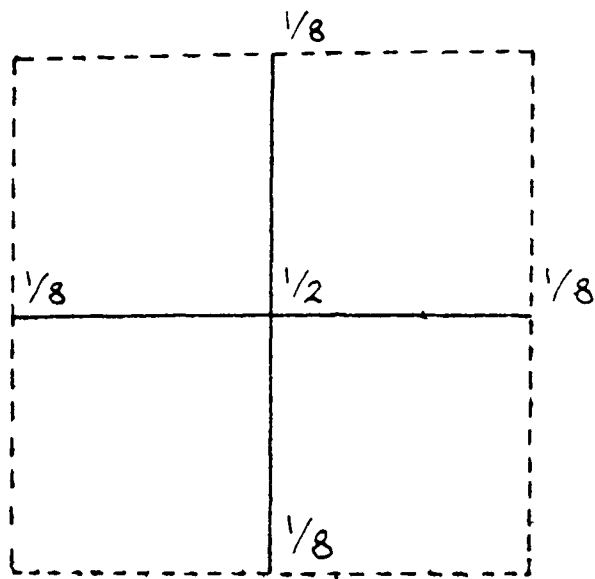


Figure 2.3

(c) full-weighting. This weighting requires more computation. The operator  $I_k^{k-1}$  is defined by

$$I_k^{k-1} V_{i,j}^k = 1/4 V_{2i,2j}^k + 1/8 (V_{2i-1,2j}^k + V_{2i+1,2j}^k + V_{2i,2j-1}^k + V_{2i,2j+1}^k) \\ + 1/16 (V_{2i-1,2j-1}^k + V_{2i-1,2j+1}^k + V_{2i+1,2j-1}^k + V_{2i+1,2j+1}^k).$$

This weighting is illustrated in Fig.2.4 for interior grid points.

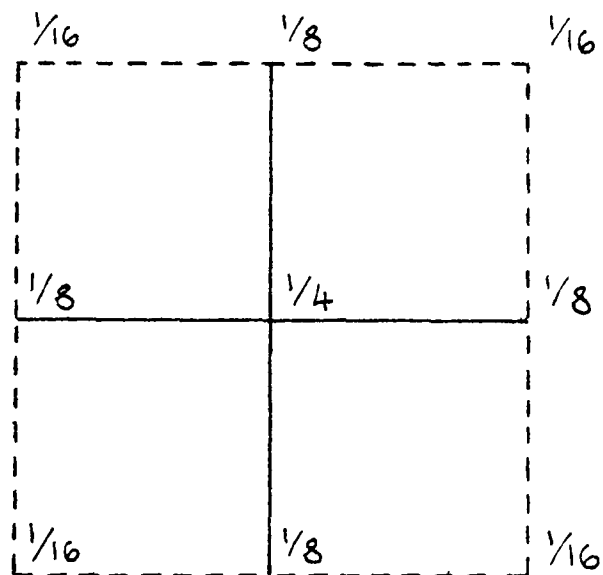


Figure 2.4

### (iii) Coarse-to-fine interpolation

The basic rule is that the order of interpolation should not be less than the order of the differential equations. This is discussed in detail by Brandt (1977); so that in the case of Poisson's equation, for example, we use bilinear interpolation.

#### (iv) Finite difference operator

For second order equations we use either the five-point or nine-point finite difference operators. In solving a given problem we use the same operator on all grids and in computing the new right-hand sides.

#### 2.6 A.D.I. Smoothing Factor

We investigate the smoothing properties of the Peaceman-Rachford A.D.I. scheme for Poisson's equation. The Peaceman-Rachford scheme for this equation is

$$\begin{aligned} (r-\delta_x^2) V_{r,s}^{(n+1)} &= (r+\delta_y^2) V_{r,s}^{(n)} + b_{r,s}, \\ (r-\delta_y^2) V_{r,s}^{(n+2)} &= (r+\delta_x^2) V_{r,s}^{(n+1)} + b_{r,s}. \end{aligned} \quad (2.18)$$

Let  $W_{r,s}^{(n)} = U_{r,s} - V_{r,s}^{(n)}$ ,  $W_{r,s}^{(n+1)} = U_{r,s} - V_{r,s}^{(n+1)}$ ,  $W_{r,s}^{(n+2)} = U_{r,s} - V_{r,s}^{(n+2)}$  where

$U$  is the true solution of the difference equations. To study the  $\theta = (\theta_1, \theta_2)$  Fourier component of the error functions  $W^{(n)}$ ,  $W^{(n+1)}$ ,  $W^{(n+2)}$  put

$$\begin{aligned} W_{r,s}^{(n)} &= A_\theta e^{i(r\theta_1+s\theta_2)}, \quad W_{r,s}^{(n+1)} = B_\theta e^{i(r\theta_1+s\theta_2)}, \\ W_{r,s}^{(n+2)} &= C_\theta e^{i(r\theta_1+s\theta_2)} \end{aligned} \quad (2.19)$$

The error functions satisfy the following equations

$$\begin{aligned} (r-\delta_x^2) W_{r,s}^{(n+1)} &= (r+\delta_y^2) W_{r,s}^{(n)}, \\ (r-\delta_y^2) W_{r,s}^{(n+2)} &= (r+\delta_x^2) W_{r,s}^{(n+1)} \end{aligned} \quad (2.20)$$

Using (2.19) we can show that

$$(r-\delta_x^2) W_{r,s}^{(n+1)} = \{r + 4\sin^2(\frac{1}{2}\theta_1)\} W_{r,s}^{(n+1)}$$

and

$$(r+\delta_y^2) W_{r,s}^{(n)} = \{r - 4\sin^2(\frac{1}{2}\theta_2)\} W_{r,s}^{(n)}$$

Substituting these expressions into the first equation in (2.20) we obtain

$$\frac{B_{\theta}}{A_{\theta}} = \frac{r - 4\sin^2(\frac{1}{2}\theta_2)}{r + 4\sin^2(\frac{1}{2}\theta_1)}$$

Similarly we obtain

$$\frac{C_{\theta}}{B_{\theta}} = \frac{r - 4\sin^2(\frac{1}{2}\theta_1)}{r + 4\sin^2(\frac{1}{2}\theta_2)}$$

So the convergence factor of the  $\theta$  component over one A.D.I. sweep is

$$\mu(\theta) = \left| \frac{C_{\theta}}{A_{\theta}} \right| = \left| \frac{(r - 4\sin^2(\frac{1}{2}\theta_1))(r - 4\sin^2(\frac{1}{2}\theta_2))}{(r + 4\sin^2(\frac{1}{2}\theta_1))(r + 4\sin^2(\frac{1}{2}\theta_2))} \right|$$

To compute the smoothing factor we need to find the maximum value of  $\mu$  over those values of  $\theta$  for which  $|\theta|$  lies in the range  $[\frac{1}{2}\pi, \pi]$  where  $|\theta| = \max(|\theta_1|, |\theta_2|)$ . This range is the set of high frequency components which are not determined by the low frequency components.

If we put

$$f(r, \theta) = \frac{r - 4\sin^2(\frac{1}{2}\theta)}{r + 4\sin^2(\frac{1}{2}\theta)},$$

then

$$\mu(\theta) = |f(r, \theta_1) f(r, \theta_2)|.$$

The maximum value of  $f(r, \theta)$  with respect to  $\theta$  in  $[-\pi, \pi]$  is 1 which occurs when  $\theta = 0$ . So our problem is reduced to the following one:

$$\bar{\mu} = \max_{\frac{1}{2}\pi \leq \theta \leq \pi} |f(r, \theta)|$$

By the change of variable  $x = \sin^2(\frac{1}{2}\theta)$  this becomes

$$\bar{\mu} = \max_{\frac{1}{2} \leq x \leq 1} \left| \frac{r - 4x}{r + 4x} \right|$$

Let

$$g(r, x) = \left( \frac{r - 4x}{r + 4x} \right),$$

then for fixed  $r$ ,  $g(r,x)$  is a monotonically decreasing function so its maximum value in the interval  $[\frac{1}{2},1]$  is attained at one of the end points.

So we have that

$$\bar{\mu} = \max \left\{ \left| \frac{r-2}{r+2} \right|, \left| \frac{r-4}{r+4} \right| \right\}.$$

It can be easily verified that

$$\bar{\mu} = \begin{cases} \mu(\Pi,0) = \frac{4-r}{4+r} & \text{for } 0 < r \leq \sqrt{8} \\ \mu(\frac{1}{2}\Pi,0) = \frac{r-2}{r+2} & \text{for } \sqrt{8} \leq r. \end{cases}$$

Thus the smoothing factor  $\bar{\mu}$  is minimized when we choose  $r = \sqrt{8}$ . In this case we have that  $\mu \simeq 0.17$ . This compares well with the smoothing factor of 0.5 for Gauss-Seidel relaxation but A.D.I. requires about two or three times as much work. Figure 2.5 shows how  $\bar{\mu}$  varies with  $r$ .

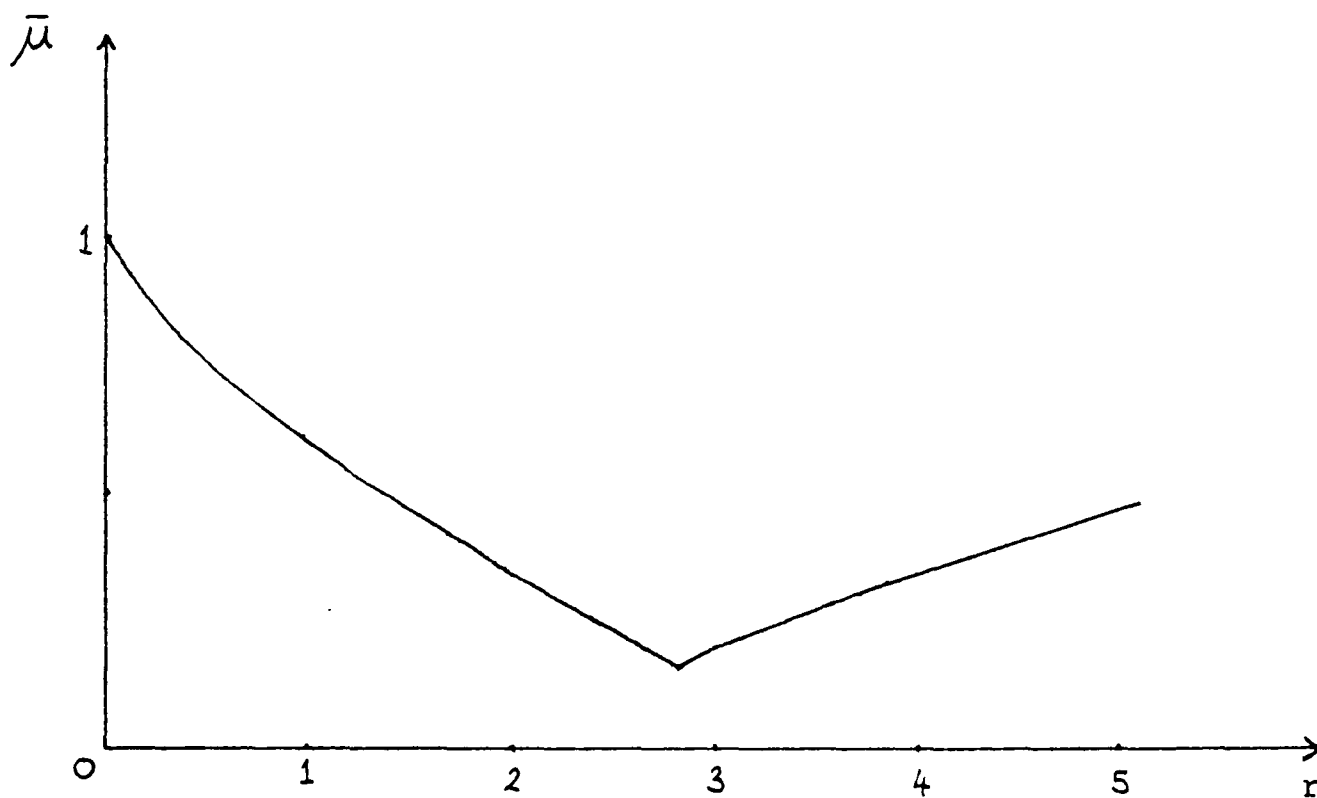


Figure 2.5

## 2.7 Multigrid Routines

Here we describe various multigrid routines that will be used later on to solve a model problem. The model problem will be Laplace's equation defined in the unit square with Dirichlet boundary conditions. The routines we use are:



(a) MGCS. Since the problem under consideration is linear we can use the Correction Storage scheme. For our relaxation procedure we use point Gauss-Seidel with the points ordered in the lexicographic manner. We use injection to transfer values from the fine grid to the coarse grid. This is the simple transfer of values which picks out values on the fine grid at points which correspond to coarse grid points. Bilinear interpolation is used for the coarse-to-fine transfer of values.

(b) MGFAS. This is the same as MGCS but with the F.A.S. scheme replacing the C.S. scheme. The F.A.S. scheme is a multigrid procedure which can be used to solve nonlinear problems. It differs from the C.S. scheme in that the coarse grid approximation is stored on the coarser grids in place of the correction.

(c) MGFASCH. The F.A.S. scheme is used in this method. For our relaxation procedure we use point Gauss-Seidel with the points ordered in the chequerboard manner i.e. we relax the finite difference equations in the following order:

- (i) at those points  $(x_i, y_j)$  for which  $i+j$  is even,
- (ii) at those points  $(x_i, y_j)$  for which  $i+j$  is odd.

Chequerboard Gauss-Seidel relaxation produces highly-oscillating residuals and so simply transferring the residuals by injection to a coarser grid is inadvisable. So half-weighting is used as the means of transferring values from the fine to the coarse grid. This multigrid procedure is described in detail by Foerster and Witsch (1981).

(d) MGADI. Again we use the F.A.S. scheme. This time A.D.I. is used as our smoothing routine. So at Step (c) in Algorithm 2.4.1 we perform one double sweep of A.D.I. with  $r = \sqrt{8}$ , which is the value that minimizes the smoothing factor  $\bar{\mu}$ . Injection is used to transfer values from the fine to the coarse grid.

## 2.8 Convergence

Convergence proofs for multigrid methods are difficult and complicated, and only exist for certain classes of problems. Fedorenko (1964) proves convergence of the multigrid method for solving Poisson's equation. Bakhvalov (1966) extends this proof to the variable coefficient case. Hackbusch (1980) gives criteria of convergence that apply to general difference schemes for boundary value problems in Lipschitzian regions using Gauss-Seidel as the smoothing routine. Wesseling (1980) proves convergence of the multigrid method for a general self-adjoint elliptic equation in the unit square with  $u=0$  on the boundary for the five-point finite difference operator.

## 2.9 Numerical Example

Suppose that the function  $u$  satisfies Laplace's equation in the unit square. We impose the following boundary conditions on  $u$ :

$$\begin{aligned} u &= 0 && \text{on } x = 0, x = 1, \\ u &= \sin \pi x, && \text{on } y = 0, \\ u &= e^{-\pi y} \sin \pi x && \text{on } y = 1. \end{aligned}$$

The true solution to this problem is  $u(x,y) = e^{-\pi y} \sin \pi x$ . We discretize Laplace's equation using the standard five-point finite difference scheme. We solve the resulting finite difference equations using the multigrid routines described in section 2.7 and also by S.O.R. with optimum relaxation parameter.

We define a work unit in the case of the multigrid methods using Gauss-Seidel for their smoothing routine as the computational work in one relaxation sweep over the finest grid  $G_M$ . For the multigrid method using A.D.I. as its smoothing routine we define a work unit to be the computational work in one A.D.I. double sweep on  $G_M$ . For the S.O.R. method we define a work unit to be the computational work in one complete S.O.R. sweep over  $G_M$ . Thus, over  $G_k$  ( $k < M$ ) the amount of computational work in one relaxation

sweep is about  $4^{(k-M)}$  work units since the number of points in  $G_k$  is approximately a quarter of those in  $G_{k+1}$ .

We denote by  $h$  the step length on the finest grid  $G_M$ . In all cases the step size on the coarsest grid is 0.25. For each of the routines we start with the initial approximation being zero at the internal mesh points. The algorithms were terminated when the  $\ell_2$ -norm of the residuals was less than  $10^{-6}$ . The choice of the switching parameter is  $\eta = 0.6$ . Table 2.1 contains the number of work units for different step sizes. Table 2.2 contains the computational time, in seconds, for different step sizes. (See page 36)

From these results we can see that the number of work units required for convergence for MGCS and MGFAS is the same which is what we would expect. As we decrease  $h$  the number of work units for the S.O.R. method increases quadratically. For MGFASCH this number remains almost constant as we decrease  $h$ . MGADI is comparable with MGCS and MGFAS, in terms of computational work and time, for this model problem. It is also worth noting that this is a rather special problem, since it is possible to predict the optimum value of  $r$ . For more difficult problems it may not be possible to do this.

The calculation of the number of work units has ignored the work expended by interpolation and transfer of values between grids. This is a significant amount for MGFASCH since we have used half-weighting to transfer the solution and residuals to the coarser grid.

Method	$h = 1/8$	$h = 1/16$	$h = 1/32$	$h = 1/64$
MGCS	28	31	35	35
MGFAS	28	31	35	35
MGFASCH	25	24	27	27
MGADI	25	23	27	28
SOR	26	52	107	222

Table 2.1 Number of work units

Method	$h = 1/8$	$h = 1/16$	$h = 1/32$	$h = 1/64$
MGCS	0.5	0.8	2.2	7.6
MGFAS	0.6	0.9	2.3	7.8
MGFASCH	0.7	0.8	2.3	7.8
MGADI	0.7	0.8	2.4	9.0
SOR	0.4	0.7	2.3	13.9

Table 2.2 Computational time (seconds)

### Chapter 3

#### THE PHILIPS PROBLEM: NUMERICAL SOLUTION

##### 3.1 Introduction

This problem was presented by Philips Research Laboratories, Redhill, at the Oxford Study Group with Industry in March 1979. In solid state electronics the designers of PIN diodes are interested in the effect on the performance of the diode of changes in various design parameters. An advantage in producing a good mathematical model is that the testing, when performed experimentally, could be a lengthy process. A description of the model and the derivation of the equations can be found in Aitchison and Berz (1981). The model gives rise to a coupled system of nonlinear partial differential equations.

In this chapter we consider numerical techniques for the solution of these equations. We consider the geometry of diode which is in two-dimensional Cartesian co-ordinates where it is assumed that the diode is very long in the third dimension. After deriving the finite difference equations we shall obtain numerical solutions by applying the D.A.D.I. method and a multigrid method described in previous chapters.

##### 3.2 The Differential Equations

The problem is formulated in terms of the carrier density  $c(x,y)$  and a stream function  $u(x,y)$ . The behaviour of diodes which are effectively two-dimensional and of rectangular cross-section can be described by the following equations:

$$\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} = c \quad (3.1)$$

$$\frac{\partial}{\partial x} \left( \frac{1}{c} \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{1}{c} \frac{\partial u}{\partial y} \right) = 0, \quad (3.2)$$

in the region  $R = \{(x,y) : 0 \leq x \leq A, 0 \leq y \leq B\}$ .

The boundary conditions are

$$\frac{\partial c}{\partial x} = \frac{b}{(1+b)} \frac{\partial u}{\partial y}, \quad \frac{\partial u}{\partial x} = (1+b) \frac{\partial c}{\partial y}, \quad \text{on } x = 0, \quad (3.3, 3.4)$$

$$\frac{\partial c}{\partial x} = \frac{-1}{(1+b)} \frac{\partial u}{\partial y}, \quad \frac{\partial u}{\partial x} = -\frac{(1+b)}{b} \frac{\partial c}{\partial y}, \quad \text{on } x = A, \quad (3.5, 3.6)$$

$$\frac{\partial c}{\partial y} = 0, \quad u = 0, \quad \text{on } y = 0, \quad (3.7, 3.8)$$

$$\frac{\partial c}{\partial y} = -sc, \quad u = -1, \quad \text{on } y = B, \quad (3.9, 3.10)$$

where  $s$  and  $b$  are positive constants.

This problem has weak singularities in the corners located at the points  $x = 0, y = B$  and  $x = A, y = B$ . This will be shown in Section 3.3. Methods of treating the singularities will be considered in Chapter 4 but for the present purposes of obtaining a numerical solution we shall ignore them.

The S.O.R. method has been tried on this problem. It worked well for each of the equations (3.1) and (3.2) separately but the coupling between them lead to serious difficulties with convergence. Aitchison (1980) has solved this problem successfully using Newton's method and a sparse matrix routine.

There are two quantities in which the designers of diodes are particularly interested. The first is the equipotential check,  $K$ , which will be defined as a function of  $y$  and shown to be constant. The equipotential check is defined by

$$K(y) = \frac{-2b}{(1+b)^2} \int_0^A \frac{1}{c} \frac{\partial u}{\partial y} dx + \left( \frac{b-1}{b+1} \right) \log \left( \frac{c(0,y)}{c(A,y)} \right) + \log (c(0,y) \cdot c(A,y)). \quad (3.11)$$

Differentiating (3.11) with respect to  $y$  we obtain

$$\begin{aligned} \frac{dK}{dy} = & \frac{-2b}{(1+b)^2} \int_0^A \frac{\partial}{\partial y} \left( \frac{1}{c} \frac{\partial u}{\partial y} \right) dx + \frac{2b}{(1+b)} \left[ \frac{1}{c} \frac{\partial c}{\partial y} \right]_{x=0} \\ & + \frac{2}{(1+b)} \left[ \frac{1}{c} \frac{\partial c}{\partial y} \right]_{x=A} \end{aligned}$$

Using equations (3.2), (3.4) and (3.6) we obtain

$$\begin{aligned} \int_0^A \frac{\partial}{\partial y} \left( \frac{1}{c} \frac{\partial u}{\partial y} \right) dx &= - \int_0^A \frac{\partial}{\partial x} \left( \frac{1}{c} \frac{\partial u}{\partial x} \right) dx \\ &= - \left[ \frac{1}{c} \frac{\partial u}{\partial x} \right]_{x=A} + \left[ \frac{1}{c} \frac{\partial u}{\partial x} \right]_{x=0} \\ &= \frac{(1+b)}{b} \left[ \frac{1}{c} \frac{\partial c}{\partial y} \right]_{x=A} + (1+b) \left[ \frac{1}{c} \frac{\partial c}{\partial y} \right]_{x=0} \end{aligned}$$

Therefore we see that  $\frac{dK}{dy} = 0$ , and so  $K$  is constant.

The second quantity of interest is the total charge,  $Q$ , defined by

$$Q = \iint_R c \, dx \, dy.$$

Using equation (3.1) we obtain

$$\begin{aligned} Q &= \iint_R \left( \frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} \right) dx \, dy \\ &= - \int_0^A \frac{\partial c}{\partial y} (x, 0) \, dx + \int_0^B \frac{\partial c}{\partial x} (A, y) \, dy \\ &\quad + \int_0^A \frac{\partial c}{\partial y} (x, B) \, dx - \int_0^B \frac{\partial c}{\partial x} (0, y) \, dy \end{aligned}$$

by Green's theorem.

We use the boundary conditions (3.3), (3.5), (3.7) and (3.9) to obtain

$$\begin{aligned}
Q &= -\frac{1}{(1+b)} \int_0^B \frac{\partial u}{\partial y}(A,y) dy - s \int_0^A c(x,B) dx \\
&\quad - \frac{b}{(1+b)} \int_0^B \frac{\partial u}{\partial y}(0,y) dy \\
&= -\frac{1}{(1+b)} [u(A,B) - u(A,0)] - s \int_0^A c(x,B) dx \\
&\quad - \frac{b}{(1+b)} [u(0,B) - u(0,0)].
\end{aligned}$$

Substituting conditions (3.8) and (3.10) gives

$$Q = 1 - s \int_0^A c(x,B) dx \quad (3.12)$$

### 3.3 Nature of the singularities

Consider the corner at the point  $x=0$ ,  $y=B$ . The region  $R$  is translated so that this point lies at the origin. This is illustrated in Figure 3.1. On  $Ox$  we have  $\partial u / \partial x = 0$  since  $u = -1$ . Therefore  $\partial u / \partial x \rightarrow 0$  at the origin. On  $OA$  we have that  $\partial u / \partial x = (1+b) \partial c / \partial y$ . Therefore, since  $\partial u / \partial x \rightarrow 0$  at  $0$ , we have  $\partial c / \partial y \rightarrow 0$  at  $0$ . On  $Ox$  we have  $\partial c / \partial y = -sc$  and therefore  $c \rightarrow 0$  at  $0$ . However, we know from the model that  $c(0,0) \neq 0$ . This means that the boundary

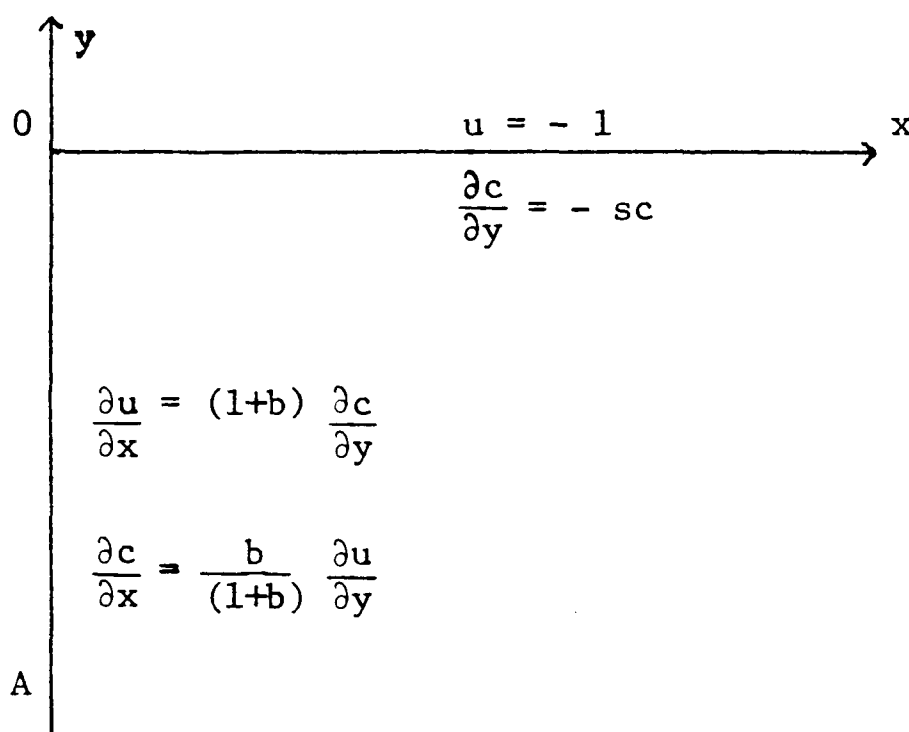


Figure 3.1



conditions are not compatible at this point and so a boundary singularity occurs. So one or both of the following hold:

- (i)  $\lim_{x \rightarrow 0_+} \frac{\partial u}{\partial x} \neq \lim_{y \rightarrow 0_-} \frac{\partial u}{\partial x},$
- (ii)  $\lim_{x \rightarrow 0_+} \frac{\partial c}{\partial y} \neq \lim_{y \rightarrow 0_-} \frac{\partial c}{\partial y}.$

If the limit of  $\partial u / \partial x$  at 0 does not exist then  $\partial^2 u / \partial x^2$  is unbounded near 0. Similarly  $\partial^2 c / \partial y^2$  is unbounded near 0 if the limit of  $\partial c / \partial y$  does not exist there. Since  $\nabla^2 c = c$  this means that  $\partial^2 c / \partial x^2$  is also unbounded near 0. Similarly  $\partial^2 u / \partial y^2$  is unbounded if  $\partial^2 u / \partial x^2$  is unbounded near 0.

We can perform a similar analysis for the corner at the point  $x=A$ ,  $y=B$  to establish that there is a singularity there of the same type. Reasonable numerical results can be obtained by ignoring the singularities. The resulting solution is reasonably accurate except near the corners.

### 3.4 Finite Difference Approximation

The finite difference equations are constructed using the integration method of Varga (1962). This technique is used because the conservative form of equation (3.2) is retained in the finite difference scheme. The resulting finite difference equations are the same as those used by Aitchison (1980).

We consider the rectangle  $R$  with  $A=2$  and  $B=4$ . Let the region  $R$  be covered with a square grid of step size  $h$  in both the  $x$  and  $y$  directions where  $Nh = 2$ . Let  $c_{i,j}$  and  $u_{i,j}$  be the values of  $c(x,y)$  and  $u(x,y)$  at the grid point  $(x_i, y_j)$  where  $x_i = ih$  and  $y_j = jh$ . Let the region  $r_{i,j}$  be defined as lying within  $R$  and being bounded by the lines  $x = x_i - \frac{1}{2}h$ ,  $x = x_i + \frac{1}{2}h$ ,  $y = y_j - \frac{1}{2}h$ ,  $y = y_j + \frac{1}{2}h$ . In our application of the technique the regions  $r_{i,j}$  are either square or rectangular. Various regions  $r_{i,j}$  are shown on Figure 3.2. Let  $s_{i,j}$  be the boundary of the region  $r_{i,j}$ .

We first consider equation (3.1). Integrating (3.1) over the region

$r_{i,j}$  gives

$$\iint_{r_{i,j}} \left( \frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} - c \right) dx dy = 0.$$

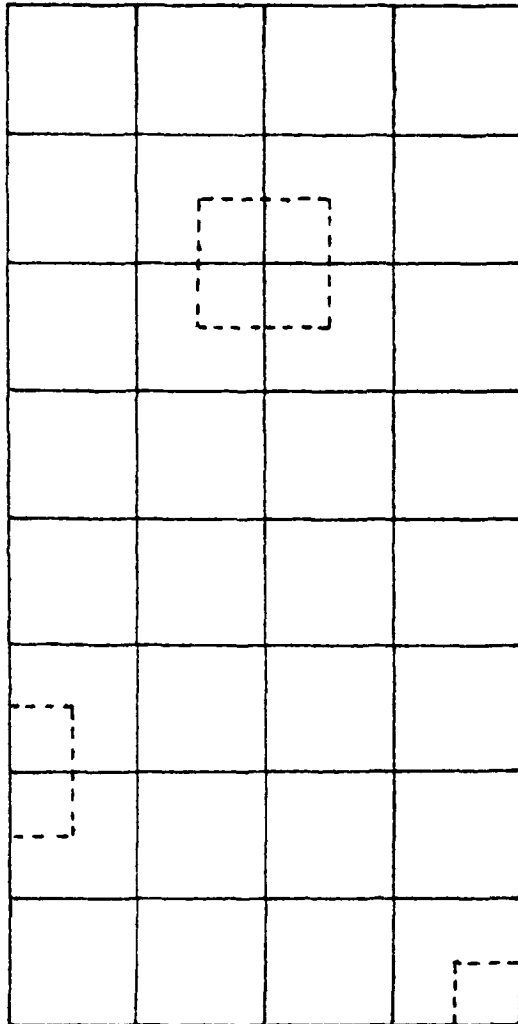


Figure 3.2

The regions enclosed by the dotted lines illustrate the three types of region  $r_{i,j}$  in our example.

We use Green's theorem on the first two terms in this integral to obtain

$$\int_{s_{i,j}} \frac{\partial c}{\partial n} ds - \iint_{r_{i,j}} c dx dy = 0 \quad (3.13)$$

where  $n$  is the unit outward drawn normal.

The finite difference approximation at internal points is therefore

$$\begin{aligned} & \left( \frac{c_{i+1,j} - c_{i,j}}{h} \right) h + \left( \frac{c_{i,j+1} - c_{i,j}}{h} \right) h + \left( \frac{c_{i-1,j} - c_{i,j}}{h} \right) h \\ & + \left( \frac{c_{i,j-1} - c_{i,j}}{h} \right) h - h^2 c_{i,j} = 0, \end{aligned}$$

where  $0 < i < N$ ,  $0 < j < 2N$ .

This equation can be simplified to give

$$c_{i+1,j} + c_{i-1,j} + c_{i,j+1} + c_{i,j-1} - 4c_{i,j} = h^2 c_{i,j} \quad (3.14)$$

which is the same as that obtained by using the standard five-point finite difference approximation to equation (3.1).

To construct the finite difference approximation along  $x = 0$  we need to consider the following integral

$$\begin{aligned} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \frac{\partial c}{\partial x} dy &= \frac{b}{(1+b)} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \frac{\partial u}{\partial y} dy \\ &= \frac{b}{(1+b)} \{ u(o, (j+\frac{1}{2})h) - u(o, (j-\frac{1}{2})h) \} \\ &\simeq \frac{b}{(1+b)} \left( \frac{u_{o,j+1} - u_{o,j-1}}{2} \right) \end{aligned}$$

where  $0 < j < 2N$ .

So the finite difference approximation to (3.13) along  $x = 0$  is

$$\begin{aligned} & \left( \frac{c_{1,j} - c_{o,j}}{h} \right) h + \left( \frac{c_{o,j+1} - c_{o,j}}{h} \right) \frac{h}{2} + \left( \frac{c_{o,j-1} - c_{o,j}}{h} \right) \frac{h}{2} \\ & - \frac{b}{(1+b)} \left( \frac{u_{o,j+1} - u_{o,j-1}}{2} \right) = \frac{h^2}{2} c_{o,j}, \end{aligned} \quad (3.15)$$

where  $0 < j < 2N$ .

This equation can be written in the form

$$\begin{aligned} & 2c_{1,j} + c_{o,j+1} + c_{o,j-1} - 4c_{o,j} \\ & - \frac{b}{(1+b)} (u_{o,j+1} - u_{o,j-1}) = h^2 c_{o,j} \end{aligned}$$

Similarly we can show that the finite difference approximation to (3.13) along  $x=2$  is given by

$$2c_{N-1,j} + c_{N,j+1} + c_{N,j-1} - 4c_{N,j} - \frac{1}{(1+b)} (u_{N,j+1} - u_{N,j-1}) = h^2 c_{N,j}, \quad (3.16)$$

where  $0 < j < 2N$

Along  $y=0$  the finite difference approximation to (3.13) is given by

$$2c_{i,1} + c_{i+1,0} + c_{i-1,0} - 4c_{i,0} = h^2 c_{i,0} \quad (3.17)$$

where  $0 < i < N$ .

Similarly the finite difference approximation along  $y=4$  is given by

$$2c_{i,2N-1} + c_{i+1,2N} + c_{i-1,2N} - 4c_{i,2N} - 2hs c_{i,2N} = h^2 c_{i,2N}, \quad (3.18)$$

where we have used boundary condition (3.9). Equation (3.18) holds for  $0 < i < N$ .

We now consider the equations for the four corner points. The finite difference representation of (3.13) in the corner  $x=0, y=0$  is given by

$$\left( \frac{c_{1,0} - c_{0,0}}{h} \right) \frac{h}{2} + \left( \frac{c_{0,1} - c_{0,0}}{h} \right) \frac{h}{2} - \frac{b}{(1+b)} \frac{(u_{0,1} - u_{0,0})}{2} = \frac{h^2}{4} c_{0,0}$$

This simplifies to

$$2c_{1,0} + 2c_{0,1} - 4c_{0,0} - \frac{2b}{(1+b)} (u_{0,1} - u_{0,0}) = h^2 c_{0,0} \quad (3.19)$$

Similarly the finite difference approximations to (3.13) in the other three corners are

$$2c_{1,2N} + 2c_{0,2N-1} - 4c_{0,2N} - \frac{2b}{(1+b)} (u_{0,2N} - u_{0,2N-1}) - 2sh c_{0,2N} = h^2 c_{0,2N}, \quad (3.20)$$

$$2 c_{N-1,0} + 2 c_{N,1} - 4 c_{N,0} - \frac{2}{(1+b)} (u_{N,1} - u_{N,0}) = h^2 c_{N,0}, \quad (3.21)$$

$$2 c_{N-1,2N} + 2 c_{N,2N-1} - 4 c_{N,2N} - \frac{2}{(1+b)} (u_{N,2N} - u_{N,2N-1}) - 2 \operatorname{sh} c_{N,2N} = h^2 c_{N,2N}. \quad (3.22)$$

Now we consider the discretisation of equation (3.2). Integrating equation (3.2) over the region  $r_{i,j}$  yields

$$\iint_{r_{i,j}} \left\{ \frac{\partial}{\partial x} \left( \frac{1}{c} \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{1}{c} \frac{\partial u}{\partial y} \right) \right\} dx dy = 0.$$

Applying Green's theorem to this equation we obtain

$$\int_{s_{i,j}} \frac{1}{c} \frac{\partial u}{\partial n} ds = 0. \quad (3.23)$$

The finite difference approximation at internal points is therefore

$$\begin{aligned} & \frac{(u_{i+1,j} - u_{i,j})}{\frac{1}{2}(c_{i+1,j} + c_{i,j})} + \frac{(u_{i,j+1} - u_{i,j})}{\frac{1}{2}(c_{i,j+1} + c_{i,j})} + \frac{(u_{i-1,j} - u_{i,j})}{\frac{1}{2}(c_{i-1,j} + c_{i,j})} \\ & + \frac{(u_{i,j-1} - u_{i,j})}{\frac{1}{2}(c_{i,j-1} + c_{i,j})} = 0, \end{aligned} \quad (3.24)$$

where  $0 < i < N$ ,  $0 < j < 2N$ .

To construct the finite difference approximation along  $x=0$  we need to consider the following integral

$$\begin{aligned} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \frac{1}{c(o,y)} \frac{\partial u}{\partial x}(o,y) dy &= (1+b) \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \frac{1}{c(o,y)} \frac{\partial c}{\partial y}(o,y) dy \\ &= (1+b) \{ \log(c(o, (j+\frac{1}{2})h)) - \log(c(o, (j-\frac{1}{2})h)) \} \\ &\simeq (1+b) \log \left( \frac{c_{o,j+1} + c_{o,j}}{c_{o,j-1} + c_{o,j}} \right) \end{aligned}$$

In this calculation we have used boundary condition (3.4). So the finite difference approximation to (3.23) along  $x=0$  is given by

$$\begin{aligned}
& \frac{(u_{1,j} - u_{0,j})}{\frac{1}{2}(c_{1,j} + c_{0,j})} + \frac{(u_{0,j+1} - u_{0,j})}{\frac{1}{2}(c_{0,j+1} + c_{0,j})} \cdot \frac{1}{2} + \frac{(u_{0,j-1} - u_{0,j})}{\frac{1}{2}(c_{0,j-1} + c_{0,j})} \cdot \frac{1}{2} \\
& - (1+b) \log \left( \frac{c_{0,j+1} + c_{0,j}}{c_{0,j-1} + c_{0,j}} \right) = 0, \quad (3.25)
\end{aligned}$$

where  $0 < j < 2N$ .

Similarly we obtain the finite difference approximation to (3.23) along  $x=2$  using boundary condition (3.6)

$$\begin{aligned}
& \frac{(u_{N-1,j} - u_{N,j})}{\frac{1}{2}(c_{N-1,j} + c_{N,j})} + \frac{(u_{N,j+1} - u_{N,j})}{\frac{1}{2}(c_{N,j+1} + c_{N,j})} \cdot \frac{1}{2} + \frac{(u_{N,j-1} - u_{N,j})}{\frac{1}{2}(c_{N,j-1} + c_{N,j})} \cdot \frac{1}{2} \\
& - \frac{(1+b)}{b} \log \left( \frac{c_{N,j+1} + c_{N,j}}{c_{N,j-1} + c_{N,j}} \right) = 0, \quad (3.26)
\end{aligned}$$

where  $0 < j < 2N$ .

Along  $y=0$  and  $y=4$  we have

$$u_{i,0} = 0 \quad (3.27)$$

$$\text{and } u_{i,2N} = -1, \quad (3.28)$$

respectively where  $0 \leq i \leq N$ .

Equations (3.14) to (3.22) and (3.24) to (3.28) represent a system of  $2(N+1)(2N+1)$  equations for the  $4N(N+1)$  unknowns  $c_{i,j}$  ( $0 \leq i \leq N$ ,  $0 \leq j \leq 2N$ ) and  $u_{i,j}$  ( $0 \leq i \leq N$ ,  $1 \leq j \leq 2N-1$ ), and  $2(N+1)$  knowns  $u_{i,0}$  ( $0 \leq i \leq N$ ) and  $u_{i,2N}$  ( $0 \leq i \leq N$ ).

Let  $K_j$  be the discrete form of  $K((j+\frac{1}{2})h)$  for  $j=0, 1, \dots, 2N-1$ .  $K((j+\frac{1}{2})h)$  is discretized using the trapezium rule. The resulting discretization,  $K_j$ , is given by

$$\begin{aligned}
K_j = & \frac{-2b}{(1+b)^2} \sum_{i=0}^N \left( \frac{u_{i,j+1} - u_{i,j}}{\frac{1}{2}(c_{i,j+1} + c_{i,j})} \right) + \left( \frac{b-1}{b+1} \right) \log \left( \frac{c_{0,j+1} + c_{0,j}}{c_{N,j+1} + c_{N,j}} \right) \\
& + \log \left\{ (c_{0,j+1} + c_{0,j}) \cdot (c_{N,j+1} + c_{N,j}) \right\}
\end{aligned}$$

where the summation notation is defined by

$$\sum_{i=0}^N a_i = 1/2 a_0 + a_1 + \dots + a_{N-1} + 1/2 a_N.$$

Aitchison (1980) shows that  $K_j$  is a constant independent of  $j$  and so the above difference scheme exactly conserves this constant.

To calculate the total charge we discretize equation (3.12), again using the trapezium rule. Let  $\hat{Q}$  be the discrete form of  $Q$ , then  $\hat{Q}$  is given by

$$\hat{Q} = 1 - \text{sh} \sum_{i=0}^N c_{i,2N} \quad (3.30)$$

### 3.5 Dynamic A.D.I. Method

We consider a dynamic A.D.I. (D.A.D.I.) method of solution to this problem. The A.D.I. approach first converts equations (3.1) and (3.2) to the parabolic equations

$$\frac{\partial c}{\partial t} = \frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} - c, \quad (3.31)$$

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( \frac{1}{c} \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{1}{c} \frac{\partial u}{\partial y} \right). \quad (3.32)$$

In the following we assume that as  $t \rightarrow \infty$  the solution of these time-dependent equations tends to the steady state solution, which satisfies equations (3.1) and (3.2).

It would also be possible to use (3.31) and the equation

$$\frac{\partial u}{\partial t} = \lambda \left( \frac{\partial}{\partial x} \left( \frac{1}{c} \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{1}{c} \frac{\partial u}{\partial y} \right) \right), \quad (3.33)$$

where  $\lambda \neq 1$  is a constant, to reach the steady state solution. When we come to discretize these equations in time it means that, effectively, we use different time steps for the two equations. In the remainder of this section we describe the D.A.D.I. method for  $\lambda = 1$ . Numerical results are obtained for the case  $\lambda \neq 1$  and compared with those for which  $\lambda = 1$ .

The introduction of operators  $A_1, A_2, B_1, B_2$  where

$$A_1 = \left( \frac{\partial^2}{\partial x^2} - \frac{1}{2} \right), \quad B_1 = \left( \frac{\partial^2}{\partial y^2} - \frac{1}{2} \right),$$

$$A_2 = \frac{\partial}{\partial x} \left( \frac{1}{c} \cdot \frac{\partial}{\partial x} \right), \quad B_2 = \frac{\partial}{\partial y} \left( \frac{1}{c} \cdot \frac{\partial}{\partial y} \right),$$

enables us to write (3.31) and (3.32) in the form

$$\frac{\partial c}{\partial t} = (A_1 + B_1) c, \quad (3.34)$$

$$\frac{\partial u}{\partial t} = (A_2 + B_2) u, \quad (3.35)$$

Starting from an initial approximation  $c^0$  we discretize (3.34) in time with step size  $\Delta t$ . The A.D.I. method requires the solution of

$$c^{n+1} - c^n = \Delta t \{ A_1 c^{n+1} + B_1 c^n \}, \quad (3.36)$$

$$c^{n+2} - c^{n+1} = \Delta t \{ A_1 c^{n+1} + B_1 c^{n+2} \}, \quad (3.37)$$

alternately on odd and even numbered time steps. Similarly, starting from an initial approximation  $u^0$ , equation (3.35) is discretized in time with step size  $\Delta t$ . In this case the A.D.I. equations are given by

$$u^{n+1} - u^n = \Delta t \{ A_2 u^{n+1} + B_2 u^n \}, \quad (3.38)$$

$$u^{n+2} - u^{n+1} = \Delta t \{ A_2 u^{n+1} + B_2 u^{n+2} \}. \quad (3.39)$$

The finite difference approximations to the right-hand sides of (3.36) (3.37), (3.38) and (3.39) are constructed using the technique described in Section 3.4. The resulting A.D.I. equations can be written in the following matrix form:

$$\underline{u}^{(n+1)} - \underline{u}^{(n)} = \Delta t \{ L_2^{(n)} \underline{u}^{(n+1)} + M_2^{(n)} \underline{u}^{(n)} + \underline{d}^{(n)} \}. \quad (3.40)$$

$$\underline{u}^{(n+2)} - \underline{u}^{(n+1)} = \Delta t \{ L_2^{(n)} \underline{u}^{(n+1)} + M_2^{(n)} \underline{u}^{(n+2)} + \underline{d}^{(n)} \}, \quad (3.41)$$

$$\underline{c}^{(n+1)} - \underline{c}^{(n)} = \Delta t \{ L_1 \underline{c}^{(n+1)} + M_1 \underline{c}^{(n)} + \underline{f}^{(n)} \}, \quad (3.42)$$

$$\underline{c}^{(n+2)} - \underline{c}^{(n+1)} = \Delta t \{ L_1 \underline{c}^{(n+1)} + M_1 \underline{c}^{(n+2)} + \underline{f}^{(n)} \}, \quad (3.43)$$

where  $\underline{u}^{(n)}$  and  $\underline{c}^{(n)}$  are the vectors of the unknown values of  $u$  and  $c$  respectively at the grid points, and  $\underline{d}^{(n)}$  and  $\underline{f}^{(n)}$  are vectors containing



the boundary conditions. The matrices  $L_2^{(n)}$  and  $M_2^{(n)}$  vary between successive double sweeps of the A.D.I. process since they contain values of  $c$  through the coefficients of the finite difference equations. Equations (3.40) to (3.43) comprise one double sweep of the A.D.I. iteration.

If we let  $r = h^2/\Delta t$  the A.D.I. equations at internal points are given by

$$\begin{aligned}
 & - \frac{2 u_{i-1,j}^{(n+1)}}{(c_{i,j}^{(n)} + c_{i-1,j}^{(n)})} + \left( \frac{2}{(c_{i,j}^{(n)} + c_{i-1,j}^{(n)})} + \frac{2}{(c_{i,j}^{(n)} + c_{i+1,j}^{(n)})} + r \right) u_{i,j}^{(n+1)} \\
 & - \frac{2 u_{i+1,j}^{(n+1)}}{(c_{i,j}^{(n)} + c_{i+1,j}^{(n)})} = \frac{2 u_{i,j-1}^{(n)}}{(c_{i,j}^{(n)} + c_{i,j-1}^{(n)})} \\
 & - \left( \frac{2}{(c_{i,j}^{(n)} + c_{i,j-1}^{(n)})} + \frac{2}{(c_{i,j}^{(n)} + c_{i,j+1}^{(n)})} - r \right) u_{i,j}^{(n)} + \frac{2 u_{i,j+1}^{(n)}}{(c_{i,j}^{(n)} + c_{i,j+1}^{(n)})} \\
 & - \frac{2 u_{i,j-1}^{(n+2)}}{(c_{i,j}^{(n)} + c_{i,j-1}^{(n)})} + \left( \frac{2}{(c_{i,j}^{(n)} + c_{i,j-1}^{(n)})} + \frac{2}{(c_{i,j}^{(n)} + c_{i,j+1}^{(n)})} + r \right) u_{i,j}^{(n+2)} \\
 & - \frac{2 u_{i,j+1}^{(n+2)}}{(c_{i,j}^{(n)} + c_{i,j+1}^{(n)})} = \frac{2 u_{i-1,j}^{(n+1)}}{(c_{i,j}^{(n)} + c_{i-1,j}^{(n)})} \\
 & - \left( \frac{2}{(c_{i,j}^{(n)} + c_{i-1,j}^{(n)})} + \frac{2}{(c_{i,j}^{(n)} + c_{i+1,j}^{(n)})} - r \right) u_{i,j}^{(n+1)} + \frac{2 u_{i+1,j}^{(n+1)}}{(c_{i,j}^{(n)} + c_{i+1,j}^{(n)})}
 \end{aligned}$$

and

$$\begin{aligned}
 & - c_{i-1,j}^{(n+1)} + (2 + r + 1/2h^2) c_{i,j}^{(n+1)} - c_{i+1,j}^{(n+1)} \\
 & = c_{i,j-1}^{(n)} - (2 - r + 1/2h^2) c_{i,j}^{(n)} + c_{i,j+1}^{(n)},
 \end{aligned}$$

$$\begin{aligned}
& - c_{i,j-1}^{(n+2)} + (2 + r + 1/2h^2) c_{i,j}^{(n+2)} - c_{i,j+1}^{(n+2)} \\
& = c_{i-1,j}^{(n+1)} - (2 - r + 1/2h^2) c_{i,j}^{(n+1)} + c_{i+1,j}^{(n+1)}
\end{aligned}$$

where  $0 < i < N$ ,  $0 < j < 2N$ .

A detailed description of the A.D.I. method is given in Chapter 1. We note that the systems of equations which we solve in this A.D.I. process are tridiagonal and diagonally dominant. This means that the necessary matrix inverses exist and the solution of the systems by Gaussian elimination and back-substitution is stable without the need for interchanges.

Each step of the D.A.D.I. method comprises two double sweeps of the A.D.I. iteration with time step  $\Delta t$  together with a bookkeeping double sweep of the A.D.I. iteration with time step  $2\Delta t$ . With the original strategy, which was described in Chapter 1, we found that after several D.A.D.I. steps the method became satisfied with a value of  $\Delta t$  and never changed it again. The effect of this was slow convergence. Doss and Miller (1979) suggest that, in cases like this, the strategy should be revised so that  $\Delta t$  increases by the factor  $\sqrt{3}$ , for example, instead of 1 when TP (test parameter) falls in the desired interval  $(0.1, 0.3]$ . This revised strategy, with an irrational factor, causes  $\Delta t$  to change more without ever returning to exactly the same value. This worked to good effect with the Philips problem and avoided the near stagnation of the original strategy.

#### Algorithm 3.5.1

The following algorithm, which is flowcharted in Figure 3.3, is used to solve the Philips problem by the D.A.D.I. method.

(a) To start the algorithm we choose an initial time step  $\Delta t = \Delta t_0 = (0.1)h^2$ , and put  $r = h^2/\Delta t$ . Let  $\epsilon$  be the required tolerance. Set  $k = 0$  where  $k$  is the total number of time steps we have advanced. Choose initial approximations  $\underline{c}^{(k)}$  and  $\underline{u}^{(k)}$ .

- (b) Start a step of the D.A.D.I. process with current approximations  $\underline{c}^{(k)}$  and  $\underline{u}^{(k)}$ .
- (c) Perform two double sweeps of the A.D.I. iteration with time step  $\Delta t$ . Equation (3.40) to (3.43) represent one such double sweep. Let  $\underline{c}^{(k+4)}$  and  $\underline{u}^{(k+4)}$  be the new approximations obtained. Compute the  $\ell_2$ -norm of the residuals of the finite difference equations (3.14) to (3.22) and (3.24) to (3.28). Let  $e$  denote  $\|\text{residuals}\|_2$ .
- (d) If  $e < \epsilon$  then the residuals are sufficiently small and the algorithm is terminated. If  $e \geq \epsilon$ , set  $\Delta t^* = 2\Delta t$  and put  $r = h^2/\Delta t^*$ .
- (e) Perform a double sweep of the A.D.I. iteration with time step  $\Delta t^*$  starting with approximations  $\underline{c}^{(k)}$  and  $\underline{u}^{(k)}$  to obtain  $\tilde{\underline{c}}^{(k+4)}$  and  $\tilde{\underline{u}}^{(k+4)}$ . This is the bookkeeping part of the D.A.D.I. process.
- (f) Compute TP, the test parameter, given by

$$TP = \sqrt{[SUM/ASUM]},$$

where

$$SUM = \|\underline{c}^{(k+4)} - \tilde{\underline{c}}^{(k+4)}\|_2^2 + \|\underline{u}^{(k+4)} - \tilde{\underline{u}}^{(k+4)}\|_2^2$$

and

$$ASUM = \|\underline{c}^{(k+4)} - \underline{c}^{(k)}\|_2^2 + \|\underline{u}^{(k+4)} - \underline{u}^{(k)}\|_2^2$$

- (g) If  $TP > 0.6$ , then we reject the present D.A.D.I. step, replace  $\Delta t$  by  $1/16 \Delta t$  and go to Step (b). If  $TP \leq 0.6$ , then we accept the present D.A.D.I. step and change  $\Delta t$  by the factor of 4, 2,  $\sqrt{3}$ ,  $1/2$ ,  $1/4$  for the next step if TP falls in the intervals  $(-\infty, 0.05]$ ,  $(0.05, 0.1]$ ,  $(0.1, 0.3]$ ,  $(0.3, 0.4]$ ,  $(0.4, 0.6]$  respectively. Go to Step (b).

The D.A.D.I. method can be easily adapted to the case  $\lambda \neq 1$  by replacing  $\Delta t$  by  $\lambda \Delta t$  in the A.D.I. equations (3.40) and (3.41) for  $u$ . In this case Algorithm 3.5.1 is modified by changing the acceleration parameter in the A.D.I. equations for  $u$  from  $r = h^2/\Delta t$  to  $r = h^2/(\lambda \Delta t)$ .

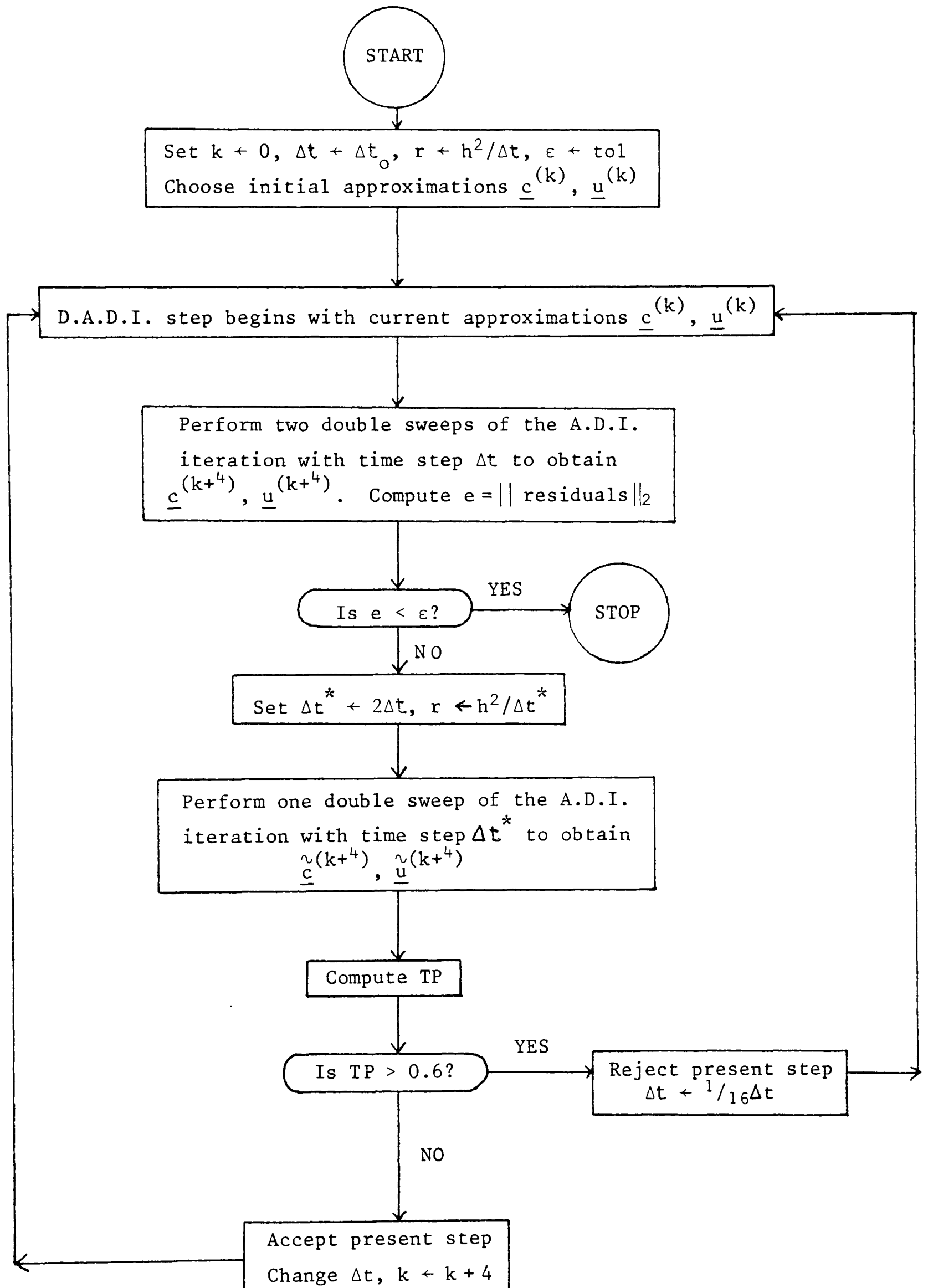


Figure 3.3

### 3.6 Multigrid Method

We now consider a multigrid method of solution to this problem using the accommodative F.A.S. cycling algorithm described in Chapter 2. Let  $G_1, G_2, \dots, G_M$  be a sequence of grids approximating the region  $R = \{ (x,y): 0 \leq x \leq 2, 0 \leq y \leq 4 \}$  with corresponding mesh sizes  $h_1, h_2, \dots, h_M$ . Let  $N_k, k=1, \dots, M$ , be a sequence of integers such that  $N_k h_k = 2$ . Let  $h_k = 2h_{k+1}$  for  $k=1, \dots, M-1$ .

This problem is discretized on each grid  $G_k$  using the finite difference equations derived in Section 3.4. Let the discrete operators  $L_1^k$  and  $L_2^k$  define discretizations of equations (3.1) and (3.2) respectively on  $G_k$ . At internal points these operators are defined by

$$\begin{aligned} (L_1^k c^k)_{i,j} &= c_{i+1,j} + c_{i-1,j} + c_{i,j-1} + c_{i,j+1} - (4+h^2) c_{i,j}, \\ (L_2^k u^k)_{i,j} &= \left( \frac{2}{c_{i+1,j} + c_{i,j}} \right) u_{i+1,j} + \left( \frac{2}{c_{i-1,j} + c_{i,j}} \right) u_{i-1,j} \\ &\quad + \left( \frac{2}{c_{i,j-1} + c_{i,j}} \right) u_{i,j-1} + \left( \frac{2}{c_{i,j+1} + c_{i,j}} \right) u_{i,j+1} \\ &\quad - \left( \left( \frac{2}{c_{i+1,j} + c_{i,j}} \right) + \left( \frac{2}{c_{i-1,j} + c_{i,j}} \right) \right. \\ &\quad \left. + \left( \frac{2}{c_{i,j-1} + c_{i,j}} \right) + \left( \frac{2}{c_{i,j+1} + c_{i,j}} \right) \right) u_{i,j} \end{aligned}$$

Along  $x=0$  the operators are defined by

$$(L_1^k c^k)_{0,j} = 2c_{1,j} + c_{0,j-1} + c_{0,j+1} - (4+h^2)c_{0,j} - \frac{b}{(1+b)} (u_{0,j+1} - u_{0,j-1}),$$

$$\begin{aligned}
(L_2^{k u^k})_{o,j} &= \left( \frac{4}{c_{1,j} + c_{o,j}} \right) u_{1,j} + \left( \frac{2}{c_{o,j+1} + c_{o,j}} \right) u_{o,j+1} + \left( \frac{2}{c_{o,j-1} + c_{o,j}} \right) u_{o,j-1} \\
&\quad - \left( \left( \frac{4}{c_{1,j} + c_{o,j}} \right) + \left( \frac{2}{c_{o,j+1} + c_{o,j}} \right) + \left( \frac{2}{c_{o,j-1} + c_{o,j}} \right) \right) u_{o,j} \\
&\quad - 2(1+b) \log \left( \frac{c_{o,j+1} + c_{o,j}}{c_{o,j-1} + c_{o,j}} \right),
\end{aligned}$$

where  $0 < j < 2N$ .

Along  $x=2$  the operators are defined by

$$(L_2^{k c^k})_{N,j} = 2c_{N-1,j} + c_{N,j+1} + c_{N,j-1} - (4+h^2)c_{N,j} - \frac{1}{(1+b)} (u_{N,j+1} - u_{N,j-1}),$$

$$\begin{aligned}
(L_2^{k u^k})_{N,j} &= \left( \frac{4}{c_{N-1,j} + c_{N,j}} \right) u_{N-1,j} + \left( \frac{2}{c_{N,j+1} + c_{N,j}} \right) u_{N,j+1} + \left( \frac{2}{c_{N,j-1} + c_{N,j}} \right) u_{N,j-1} \\
&\quad - \left( \left( \frac{4}{c_{N-1,j} + c_{N,j}} \right) + \left( \frac{2}{c_{N,j+1} + c_{N,j}} \right) + \left( \frac{2}{c_{N,j-1} + c_{N,j}} \right) \right) u_{N,j} \\
&\quad - \frac{2(1+b)}{b} \log \left( \frac{c_{N,j+1} + c_{N,j}}{c_{N,j-1} + c_{N,j}} \right),
\end{aligned}$$

where  $0 < j < 2N$ .

Along  $y=0$  and  $y=4$  the operator  $L_1^k$  is defined by

$$\begin{aligned}
(L_1^{k c^k})_{i,o} &= 2c_{i,1} + c_{i+1,o} + c_{i-1,o} - (4+h^2)c_{i,o}, \\
(L_1^{k c^k})_{i,2N} &= 2c_{i,2N-1} + c_{i+1,2N} + c_{i-1,2N} - (4+h^2+2hs)c_{i,2N},
\end{aligned}$$

where  $0 < i < N$ .

Finally at the four corners  $L_1^k$  is defined by the following:

$$\begin{aligned}
(L_1^{k c^k})_{o,o} &= 2c_{1,o} + 2c_{o,1} - (4+h^2)c_{o,o} - \frac{2b}{(1+b)} (u_{o,1} - u_{o,o}), \\
(L_1^{k c^k})_{o,2N} &= 2c_{1,2N} + 2c_{o,2N-1} - (4+h^2+2sh)c_{o,2N} - \frac{2b}{(1+b)} (u_{o,2N} - u_{o,2N-1}),
\end{aligned}$$

$$(L_1^k c^k)_{N,0} = 2c_{N-1,0} + 2c_{N,1} - (4+h^2) c_{N,0} - \frac{2}{(1+b)} (u_{N,1} - u_{N,0}),$$

$$(L_1^k c^k)_{N,2N} = 2c_{N-1,2N} + 2c_{N,2N-1} - (4+h^2+2hs) c_{N,2N} - \frac{2}{(1+b)} (u_{N,2N} - u_{N,2N-1}).$$

In defining the operators above we have omitted the superscripts  $k$  on  $c$  and  $u$  on the right-hand sides and the subscripts  $k$  on  $N$ . On the finest grid,  $G_M$ , we have

$$L_1^M c^M = 0, \quad L_2^M u^M = 0.$$

The multigrid algorithm which we use is a natural extension of Algorithm 2.4.1 to solve a system of two coupled equations. An outline of the algorithm, which is flowcharted in Figure 3.4, is given below.

#### Algorithm 3.6.1

(a) We start on the finest grid with initial approximations  $c^M$  and  $u^M$ . Set  $k=M$  and  $f_1^k$  and  $f_2^k$  equal to zero since  $L_1^k c^k = 0$  and  $L_2^k u^k = 0$ . On coarser grids  $f_1^k$  and  $f_2^k$  will denote the modified right-hand sides. Choose  $\epsilon_k$  to be a suitable tolerance.

(b) Put  $\bar{e}_k = 10^{30}$ .

(c) Perform one relaxation sweep over all the equations. Relax the equation  $L_2^k u^k = f_2^k$  first. Then use the new values of  $u^k$  to relax the equation  $L_1^k c^k = f_1^k$ . Gauss-Seidel relaxation is used as our relaxation procedure. Dynamic residuals are computed since they are inexpensive.

$$\text{Set } e_k = \sqrt{[ \| L_2^k u^k - f_2^k \|_2^2 + \| L_1^k c^k - f_1^k \|_2^2 ]}.$$

(d) If  $e_k < \epsilon_k$  i.e. relaxation has sufficiently converged on the current level, go to Step (f). If not, and if convergence is still fast, i.e.

$e_k \leq \eta \bar{e}_k$ , where  $\eta$  is fixed and chosen later, set  $\bar{e}_k = e_k$  and go to Step (c).

If convergence is slow and we are not on the coarsest grid go to Step (e).

If  $k=1$  and  $e_k > \eta \bar{e}_k$  go to Step (c).

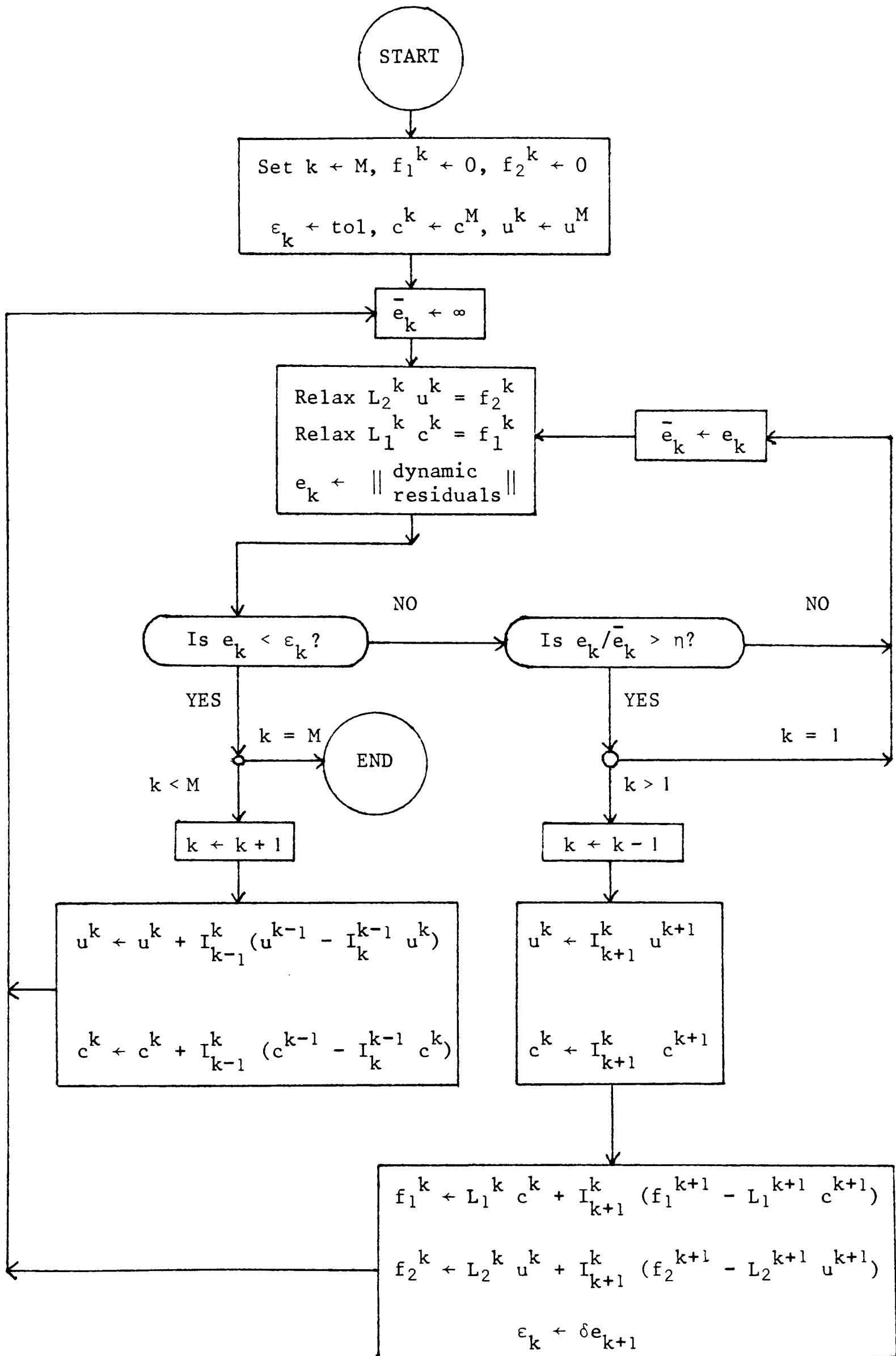


Figure 3.4



(e) Decrease  $k$  by 1. Transfer our current approximations on level  $k+1$  to the new level  $k$  as follows:

$$u^k = I_{k+1}^k u^{k+1}$$

$$c^k = I_{k+1}^k c^{k+1}$$

The right-hand sides for the new level are defined by

$$f_1^k = L_1^k c^k + 4 I_{k+1}^k (f_1^{k+1} - L_1^{k+1} c^{k+1}) ,$$

$$f_2^k = L_2^k u^k + 4 I_{k+1}^k (f_2^{k+1} - L_2^{k+1} u^{k+1})$$

The factor 4 appearing in the above equations is a scaling factor which comes from the fact that we multiplied through by  $h^2$  before defining the difference operators.

Set  $\varepsilon_k = \delta e_{k+1}$  to be the tolerance for the problem on the new level. where  $\delta$  is some parameter. Go to Step (b).

(f) If  $k=M$ , then the problem has been solved to the required tolerance and the algorithm is terminated. If  $k < M$ , we correct the approximation on the next finer grid. Put

$$u^{k+1} = u^{k+1} + I_k^{k+1} (u^k - I_{k+1}^k u^{k+1}) ,$$

$$c^{k+1} = c^{k+1} + I_k^{k+1} (c^k - I_{k+1}^k c^{k+1}) ,$$

where the  $u^{k+1}$ 's and  $c^{k+1}$ 's on the right-hand sides are the previous approximations on the level  $k+1$ . Increase  $k$  by 1 and go to Step (c).

### Multigrid Components

We describe below the components chosen for the above algorithm.

#### (a) Relaxation

Pointwise Gauss-Seidel relaxation is used with the points ordered in the chequerboard (even-odd) manner. The relaxation, performed at Step (c) of Algorithm 3.6.1, is carried out in the following order:

(i) relax the equation  $L_2^k u^k = f_2^k$  at the white (even) points i.e. those for which  $i+j$  is even,

(ii) relax the equation  $L_2^k u^k = f_2^k$  at the black (odd) points i.e. those for which  $i+j$  is odd,

(iii) relax the equation  $L_1^k c^k = f_1^k$  at the white points,

(iv) relax the equation  $L_1^k c^k = f_1^k$  at the black points.

This is just one of a number of ways of performing chequerboard Gauss-Seidel relaxation on these equations. Examples of other ways are:

- (1) perform the above steps in the order (i), (iii), (ii), (iv),
- (2) perform the above steps in the order (iii), (iv), (i), (ii),
- (3) simultaneously relax both equations at the white points, then at the black points.

We experimented using several of these methods and found that, in general, it did not make much difference in the way the equations were relaxed. At the end of Step (c) the residuals of the equation  $L_1^k c^k = f_1^k$  are zero at the black points since at this stage all the black-point equations are simultaneously satisfied.

#### (b) Fine-to-coarse transfer of residuals

Since chequerboard Gauss-Seidel relaxation produces highly oscillating residuals it is not advisable to simply transfer the residuals by injection to a coarser grid. Instead, we transfer the residuals by half-weighting to the coarse grid. If the points of the coarse grid are also white points on the finer grid then the transfer of the residuals of the equation  $L_1^k c^k = f_1^k$  reduces to half-injection i.e.

$$I_k^{k-1} (f_1^k - L_1^k c^k) = 1/2 (f_1^k - L_1^k c^k) .$$

Half-weighting is also used to transfer values of  $c^k$  and  $u^k$  on grid  $G_k$  to the next coarser grid  $G_{k-1}$ .

### (c) Coarse-to-fine transfer of values

Bilinear interpolation is used to transfer the correction to the fine grid to provide a new approximation there.

### 3.7 Numerical Results

When  $s = 0$  the differential equations (3.1) and (3.2) together with the boundary conditions (3.3) to (3.10) possess an analytic solution which is given by

$$c(x,y) = \frac{\{ b \cosh(2-x) + \cosh x \}}{4(1+b) \sinh 2} ,$$

$$u(x,y) = -^{1/4} y ,$$

for  $A = 2$ ,  $B = 4$ . These functions are used as our initial approximation to the solution of the problem for  $s \neq 0$ . Numerical results are presented for  $s = 5$  and  $s = 50$ . The constant  $b$  was given the fixed value 2.7.

In the multigrid method we define a work unit to be the computational work in one relaxation sweep over the finest grid. The step size on the coarsest grid is  $h = 1$ . The values of the parameters  $\eta$  and  $\delta$  in Algorithm 3.6.1 were chosen to be 0.5 and 0.3 respectively. It was found that the choice of  $\eta$  and  $\delta$  was not critical in the sense that values of these parameters in the neighbourhood of the chosen values produced similar efficiency of the algorithm in terms of the number of work units.

The algorithms were terminated when the  $\ell_2$ -norm of the residuals was less than  $10^{-6}$ . Table 3.1 shows how the values of  $c$  and  $u$  at the centre of the diode, the values of  $c$  at the corners, and the values of the constants  $Q$  and  $K$  vary with  $h$  when  $s = 5$ . Table 3.2 (p.60) gives the same information for the case  $s = 50$ .

Tables 3.3 and 3.4 (p.61) give details of the D.A.D.I. method of solution for the cases  $s = 5$  and  $s = 50$  respectively. In these tables we give the number of D.A.D.I. steps required to achieve the desired tolerance and the run time, in seconds, for  $\lambda = 1$  and  $\lambda = 0.05$ . In Tables 3.5 and

3.6 (p.61) we give details of the multigrid method of solution. In these tables the number of work units and the run time, in seconds, is given.

The results we have obtained agree with those obtained by Aitchison (1980). The results given in Tables 3.1 and 3.2 mainly show a good agreement and the convergence is approximately quadratic except at the corners located at the points  $(0,4)$  and  $(2,4)$ . The bad results at these points are due to the singularities. This is illustrated in Tables 3.7, 3.8 and 3.9 (pp.62 and 63) which are difference tables of  $c(1,y)$ ,  $c(0,y)$  and  $(4-y)\log(4-y)$  respectively with  $y$  taking values between 3 and 4 at intervals of 0.125. From Table 3.7 we can see that the differences of  $c(1,y)$  are well-behaved. Table 3.8 illustrates the fact that there is a singularity at the corner  $(0,4)$  since the differences diverge there (see Fox (1957)). A similar behaviour is seen in Table 3.9 for the function  $(4-y)\log(4-y)$ . As we shall see in the next chapter the solution has a singularity involving a term of this form.

We tried various values of  $\lambda \neq 1$  in the D.A.D.I. method and found, by experiment, that the value  $\lambda = 0.05$  produced the fastest convergence. We can see from the computational details that this value of  $\lambda$  is considerably better than  $\lambda = 1$  for the smallest mesh size  $h = 0.125$ . However, even with this choice of  $\lambda$ , the multigrid method performs much better than the D.A.D.I. method on this problem.

It is interesting to note that increasing  $s$  from 5 to 50 hardly affects D.A.D.I. but slows down multigrid a lot e.g. for the mesh size  $h = 0.125$ , 236 work units are required for convergence when  $s = 50$  as opposed to 85 when  $s = 5$ .

Contours of  $c(x,y)$  and  $u(x,y)$  are shown in Figures 3.5 and 3.6 (pp.64 and 65) respectively for the case  $s = 5$ .

h	0.5	0.25	0.125
Q	0.7831	0.7871	0.7884
K	-1.6415	-1.6074	-1.5984
c(1,2)	0.0890	0.0906	0.0910
u(1,2)	-0.5542	-0.5550	-0.5520
c(0,0)	0.2018	0.2071	0.2086
c(2,0)	0.1203	0.1228	0.1235
c(0,4)	0.0591	0.0682	0.0749
c(2,4)	0.0196	0.0203	0.0209

Table 3.1 Details of solution for  $s = 5$

h	0.5	0.25	0.125
Q	0.7445	0.7507	0.7532
K	-1.5745	-1.5351	-1.5239
c(1,2)	0.0862	0.0880	0.0886
u(1,2)	-0.5736	-0.5756	-0.5764
c(0,0)	0.2034	0.2096	0.2114
c(2,0)	0.1223	0.1252	0.1261
c(0,4)	0.0089	0.0120	0.0157
c(2,4)	0.0023	0.0025	0.0027

Table 3.2 Details of solution for  $s = 50$

h	Number of D.A.D.I. steps		Time	
	$\lambda = 1$	$\lambda = 0.05$	$\lambda = 1$	$\lambda = 0.05$
0.5	84	64	2.3	1.9
0.25	140	98	14.0	9.7
0.125	216	122	155.7	88.6

Table 3.3 D.A.D.I. method with  $s = 5$ 

h	Number of D.A.D.I. steps		Time	
	$\lambda = 1$	$\lambda = 0.05$	$\lambda = 1$	$\lambda = 0.05$
0.5	84	72	2.2	1.7
0.25	156	80	15.5	8.0
0.125	240	126	173.8	91.1

Table 3.4 D.A.D.I. method with  $s = 50$ 

h	Number of work units	Time
0.5	42	0.9
0.25	68	2.3
0.125	85	7.3

Table 3.5 Multigrid method with  $s = 5$ 

h	Number of work units	Time
0.5	87	1.3
0.25	77	2.5
0.125	236	17.8

Table 3.6 Multigrid method with  $s = 50$

$y$	$c(1,y)$		$\delta^2$		$\delta^4$
4.000	0.0133				
		82			
3.875	0.0215		-3		
		79		-3	
3.750	0.0294		-6		-4
		73		1	
3.625	0.0367		-5		-1
		68		0	
3.500	0.0435		-5		-1
		63		1	
3.375	0.0498		-6		1
		57		0	
3.250	0.0555		-6		0
		53		0	
3.125	0.0608		-6		
		47			
3.000	0.0655				

Table 3.7 Differences of  $c(1,y)$ 

$y$	$c(0,y)$		$\delta^2$		$\delta^4$
4.000	0.0749				
		240			
3.875	0.0989		-155		
		85		165	
3.750	0.1074		10		-186
		95		-21	
3.625	0.1169		-11		25
		84		4	
3.500	0.1253		-7		-3
		77		1	
3.375	0.1330		-6		0
		71		1	
3.250	0.1401		-5		-2
		66		-1	
3.125	0.1467		-6		
		60			
3.000	0.1527				

Table 3.8 Differences of  $c(0,y)$

y	$(4-y)\log(4-y)$		$\delta^2$		$\delta^4$
4.000	0.0				
		-2599			
3.875	-0.2599		1732		
		-867		-1077	
3.750	-0.3466		655		836
		-212		-241	
3.625	-0.3678		414		143
		212		-98	
3.500	-0.3466		316		34
		528		-64	
3.375	-0.2938		252		22
		780		-42	
3.250	-0.2158		210		10
		990		-32	
3.125	-0.1168		178		
		1168			
3.000	0.0				

Table 3.9 Differences of  $(4 - y)\log(4 - y)$



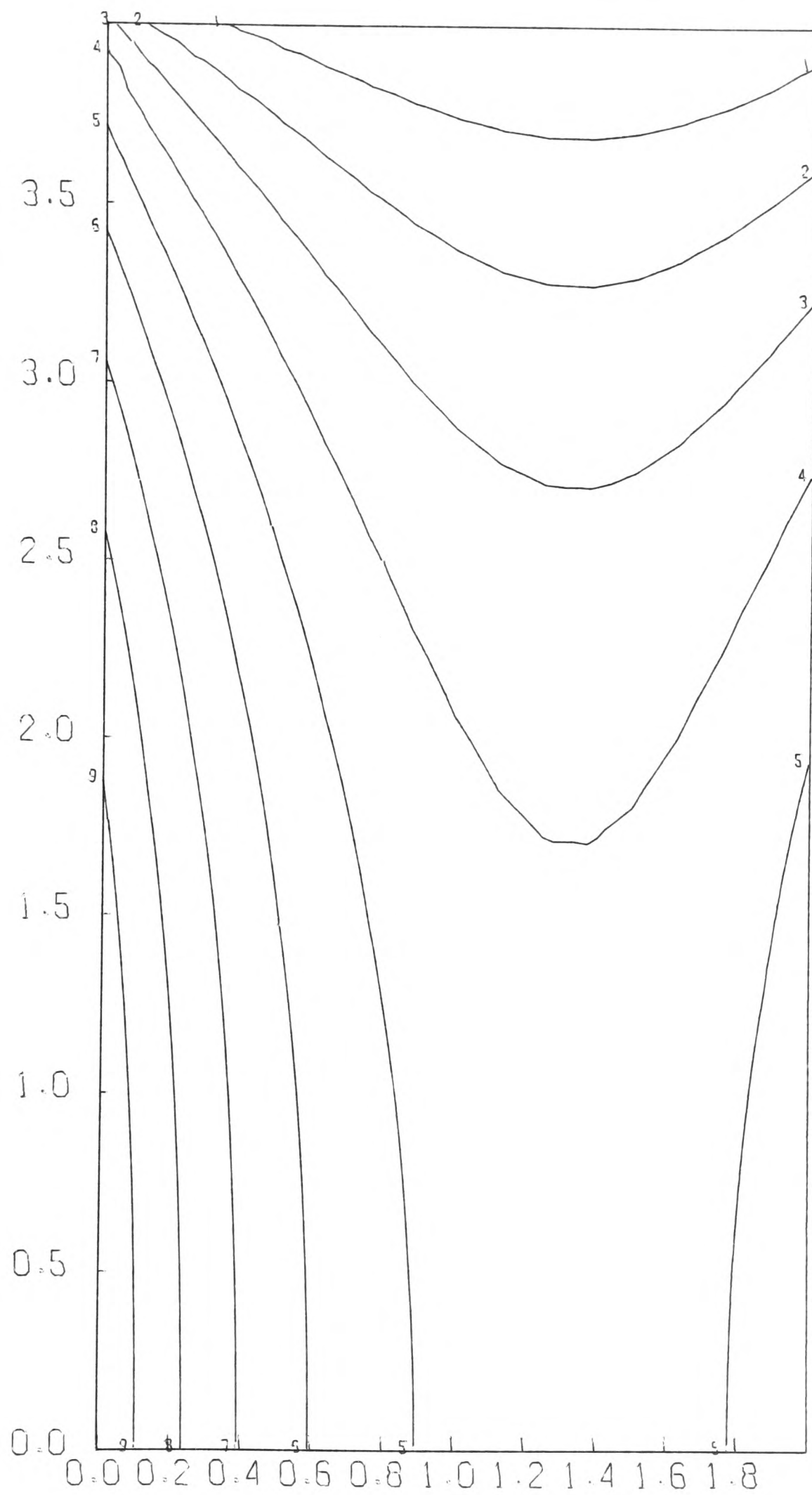


Figure 3.5. Contours of  $c(x,y)$  for  $s=5$   
 Contour levels are 0.03(0.02) 0.19.

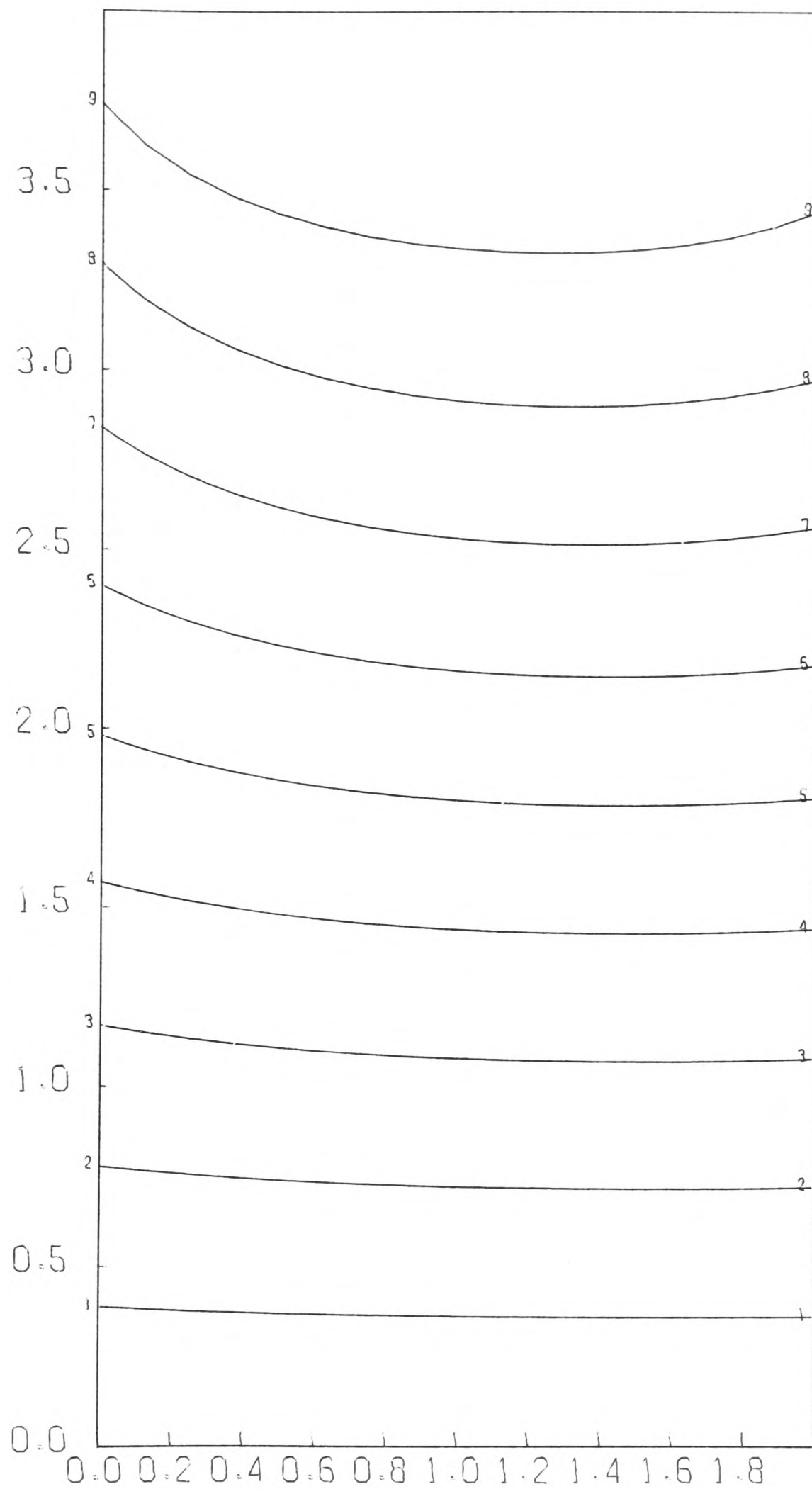


Figure 3.6. Contours of  $u(x, y)$  for  $s=5$   
 Contour levels are  $-0.1(-0.1)-0.9$ .

## Chapter 4

### THE PHILIPS PROBLEM: TREATMENT OF THE SINGULARITY

#### 4.1 Introduction

The Philips problem possesses singularities in the two top corners i.e. at the points  $x=0, y=4$  and  $x=2, y=4$ . In this chapter we look at two methods of improving the approximation in the neighbourhoods of these points. The first method uses a non-uniform grid with a smaller spacing of the grid points in the vicinities of the singularities. This technique, which can also be used to resolve boundary layers, was first proposed by Kálmán de Rivas (1972). The finite difference equations are constructed on this variable grid and solved by the D.A.D.I. method. In the second method we perform a transformation which eliminates the singularities.

#### 4.2 Non-uniform Grids

In most elliptic problems singularities that occur on the boundary do not penetrate into the interior of the region. However, the solution may change very rapidly near the singularity and one or more of its higher derivatives becomes very large, which is the case with the Philips problem. The local truncation error of the finite difference equations involves terms like  $h^p$  multiplying some  $p^{\text{th}}$  derivative, and if these get large near the singularity we would like to make  $h$  correspondingly smaller in these regions. To maintain the order of accuracy in our approximation in situations like these a uniform grid with an extremely fine mesh may be used to obtain a sufficient number of points in the vicinity of the singularity. This is wasteful and expensive since the points are distributed densely away from these regions where they may not be needed. Kálmán de Rivas (1972) suggested a different approach in which a change of independent variable is made which maps the domain into a new co-ordinate system where the variations of the

solution are not so rapid. This approach is sketched out below.

The grid intervals are varied by defining a stretched co-ordinate  $\xi$  such that  $x = x(\xi)$  where the grid intervals  $\Delta\xi$  are constant and  $x$  is the old physical co-ordinate. We can express the first and second derivatives in terms of  $\xi$  to give

$$\frac{\partial v}{\partial x} = \frac{\partial v}{\partial \xi} \cdot \frac{d\xi}{dx} \quad (4.1)$$

and

$$\frac{\partial^2 v}{\partial x^2} = \frac{\partial}{\partial \xi} \left( \frac{\partial v}{\partial \xi} \cdot \frac{d\xi}{dx} \right) \cdot \frac{d\xi}{dx} \quad (4.2)$$

Equations (4.1) and (4.2) can be discretised using central differences to give the following approximations:

$$\frac{\partial v}{\partial x} \approx \frac{v_{i+1} - v_{i-1}}{2\Delta\xi} \cdot \frac{d\xi}{dx} \Big|_{x_i}$$

and

$$\frac{\partial^2 v}{\partial x^2} \approx \frac{1}{\Delta\xi} \left[ \frac{(v_{i+1} - v_i)}{\Delta\xi} \cdot \frac{d\xi}{dx} \Big|_{x_{i+\frac{1}{2}}} - \frac{(v_i - v_{i-1})}{\Delta\xi} \cdot \frac{d\xi}{dx} \Big|_{x_{i-\frac{1}{2}}} \right] \cdot \frac{d\xi}{dx} \Big|_{x_i}$$

where  $v_i$  is the value of  $v$  at the  $i$ th grid point and  $x_i = x(i\Delta\xi)$ .

The transformation can be differentiated numerically using central differences. Thus the approximations to the first and second derivatives become

$$\begin{aligned} \frac{\partial v}{\partial x} &\approx \frac{(v_{i+1} - v_{i-1}))}{2\Delta\xi} \cdot \frac{2\Delta\xi}{(x_{i+1} - x_{i-1})} \\ &= \frac{(v_{i+1} - v_{i-1}))}{(x_{i+1} - x_{i-1})} \end{aligned} \quad (4.3)$$

and

$$\begin{aligned} \frac{\partial^2 v}{\partial x^2} &\approx \frac{1}{\Delta\xi} \left[ \frac{(v_{i+1} - v_i)}{\Delta\xi} \cdot \frac{\Delta\xi}{(x_{i+1} - x_i)} - \frac{(v_i - v_{i-1}))}{\Delta\xi} \cdot \frac{\Delta\xi}{(x_i - x_{i-1})} \right] \cdot \frac{2\Delta\xi}{(x_{i+1} - x_{i-1})} \\ &= \frac{2}{(x_{i+1} - x_{i-1})} \left[ \frac{(v_{i+1} - v_i)}{(x_{i+1} - x_i)} - \frac{(v_i - v_{i-1}))}{(x_i - x_{i-1})} \right] \end{aligned} \quad (4.4)$$

Since we have used central differences in deriving formulae (4.3) and (4.4), the formulae are formally second order in  $\Delta\xi$ . Jones and Thompson (1980) give the exact form of the local truncation errors of these approximations. They also point out that these errors will be small provided that the stretching function has the property that its derivatives are small where the derivatives of the solution are large, and vice versa.

Consider a function defined in the region  $0 \leq x \leq 1$  with a singularity at the point  $x=0$ . The stretching function  $x(\xi)$  should possess the following two properties:

- (a)  $dx/d\xi = 0$  at  $x=0$ . This will ensure a concentration of grid points near  $x=0$ . Elsewhere we should have  $dx/d\xi \neq 0$ .
- b)  $dx/d\xi$  should be finite over the whole interval since if  $dx/d\xi$  becomes infinite at some point then the mapping  $x=x(\xi)$  will give poor resolution near that point. This resolution cannot be improved by increasing the number of points since

$$\Delta x \approx \left( \frac{dx}{d\xi} \right) \cdot \Delta\xi.$$

The following are examples of good stretching functions:

- (i)  $x(\xi) = \xi^2$ . This is a suitable function if there is a singularity or boundary layer at  $x=0$ .
- (ii)  $x(\xi) = \sin^2(\frac{1}{2}\pi\xi)$ . This mapping produces concentrations of grid points near the points  $x=0$  and  $x=1$ .
- (iii)  $x(\xi) = \tan^{-1}(\pi(\xi - \frac{1}{2}))$ . This has a similar effect as (ii).

#### 4.3 D.A.D.I. Method of Solution

The finite difference equations are constructed in a similar way to those on a uniform grid using the integration method of Varga (1962). Let  $x(\xi)$  and  $y(\eta)$  be two grid stretching functions with constant grid intervals  $\Delta\xi$  and  $\Delta\eta$  respectively. These functions will be defined explicitly later. The region  $R = \{(x,y) : 0 \leq x \leq 2, 0 \leq y \leq 4\}$  is covered with a variable grid

defined by the above mappings. Let  $c_{i,j}$  and  $u_{i,j}$  be the values of  $c(x,y)$  and  $u(x,y)$  at the grid point  $(x_i, y_j)$  where  $x_i = x(i\Delta\xi)$  and  $y_j = y(j\Delta\eta)$ . It is convenient to define the following difference notation:

$$\Delta x_r = x_{r+1} - x_r,$$

$$\nabla x_r = x_r - x_{r-1}$$

$$2\mu\delta x_r = x_{r+1} - x_{r-1}$$

Let the region  $r_{i,j}$  be the part of the rectangle  $x_i - \frac{1}{2}\nabla x_i \leq x \leq x_i + \frac{1}{2}\Delta x_i$ ,  $y_j - \frac{1}{2}\nabla y_j \leq y \leq y_j + \frac{1}{2}\Delta y_j$  which lies within  $R$ . A region  $r_{i,j}$  which lies completely within  $R$  is shown in Figure 4.1, bounded by the broken lines.

Let  $s_{i,j}$  be the boundary of the region  $r_{i,j}$ .

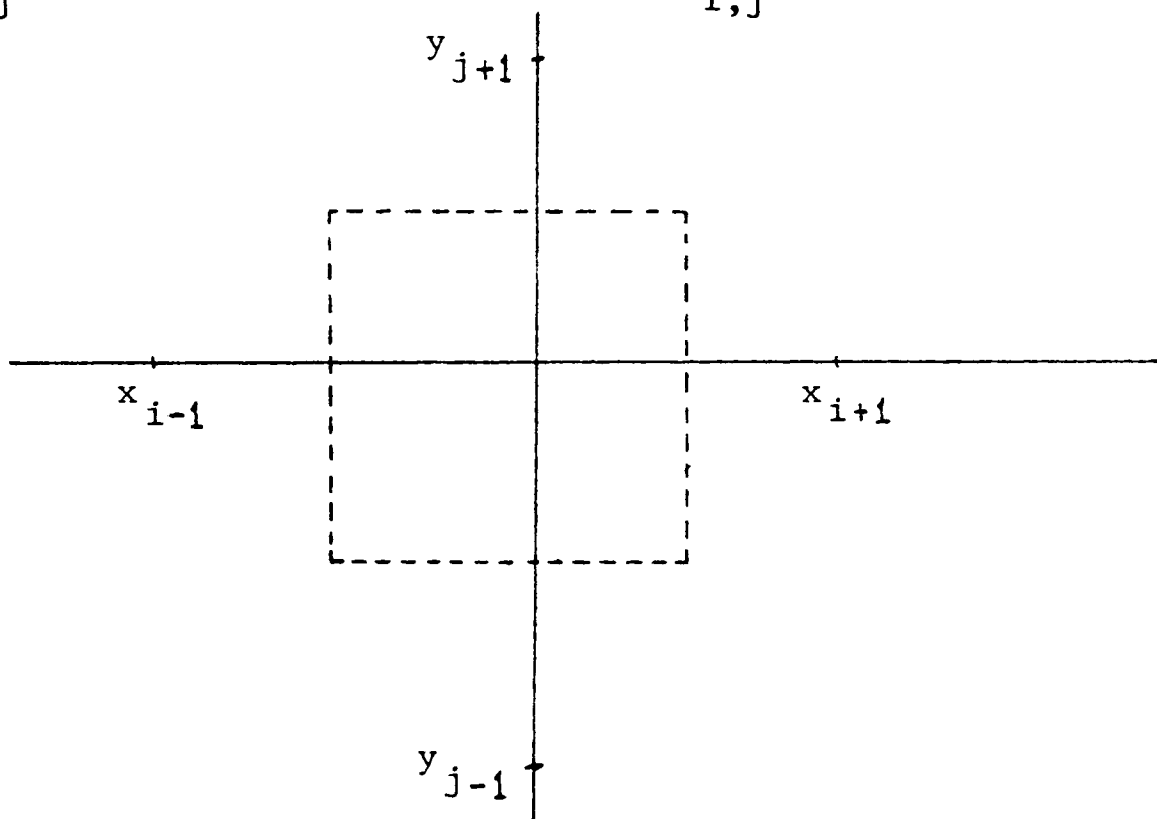


Figure 4.1

Integrating equation (3.1) over the region  $r_{i,j}$  and applying Green's theorem we obtain

$$\int_{s_{i,j}} \frac{\partial c}{\partial n} ds - \iint_{r_{i,j}} c dx dy = 0 \quad (4.5)$$

where  $n$  is the outward drawn normal.

In constructing the finite difference approximation to equation (4.5) we make use of the approximation (4.3). Therefore, at internal points, we

obtain the following approximation

$$\begin{aligned}
 & \frac{(c_{i+1,j} - c_{i,j})}{(x_{i+1} - x_i)} \cdot \frac{1}{2}(y_{j+1} - y_{j-1}) + \frac{(c_{i-1,j} - c_{i,j})}{(x_i - x_{i-1})} \cdot \frac{1}{2}(y_{j+1} - y_{j-1}) \\
 & + \frac{(c_{i,j+1} - c_{i,j})}{(y_{j+1} - y_j)} \cdot \frac{1}{2}(x_{i+1} - x_{i-1}) + \frac{(c_{i,j-1} - c_{i,j})}{(y_j - y_{j-1})} \cdot \frac{1}{2}(x_{i+1} - x_{i-1}) \\
 & - c_{i,j} \cdot \frac{1}{2}(x_{i+1} - x_{i-1}) \cdot \frac{1}{2}(y_{j+1} - y_{j-1}) = 0 .
 \end{aligned}$$

For the purposes of the A.D.I. iteration this can be written in the more convenient form

$$\begin{aligned}
 & \frac{1}{\mu \delta x_i} \left[ \frac{c_{i+1,j}}{\Delta x_i} - \left( \frac{1}{\Delta x_i} + \frac{1}{\nabla x_i} \right) c_{i,j} + \frac{c_{i-1,j}}{\nabla x_i} \right] \\
 & + \frac{1}{\mu \delta y_j} \left[ \frac{c_{i,j+1}}{\Delta y_j} - \left( \frac{1}{\Delta y_j} + \frac{1}{\nabla y_j} \right) c_{i,j} + \frac{c_{i,j-1}}{\nabla y_j} \right] - c_{i,j} = 0 , \quad (4.6)
 \end{aligned}$$

where  $0 < i < N$ ,  $0 < j < 2N$ .

Along  $x=0$  the finite difference approximation is given by

$$\begin{aligned}
 & \frac{(c_{1,j} - c_{0,j})}{(x_1 - x_0)} \cdot \frac{1}{2}(y_{j+1} - y_{j-1}) + \frac{(c_{0,j+1} - c_{0,j})}{(y_{j+1} - y_j)} \cdot \frac{1}{2}(x_1 - x_0) \\
 & \frac{(c_{0,j-1} - c_{0,j})}{(y_j - y_{j-1})} \cdot \frac{1}{2}(x_1 - x_0) - \frac{b}{(1+b)} \cdot \frac{1}{2}(u_{0,j+1} - u_{0,j-1}) \\
 & - c_{0,j} \cdot \frac{1}{2}(x_1 - x_0) \cdot \frac{1}{2}(y_{j+1} - y_{j-1}) = 0 .
 \end{aligned}$$

This can be written in the alternative form

$$\begin{aligned}
 & \frac{2c_{1,j}}{(\Delta x_0)^2} - \frac{2c_{0,j}}{(\Delta x_0)^2} + \frac{1}{\mu \delta y_j} \left[ \frac{c_{0,j+1}}{\Delta y_j} - \left( \frac{1}{\Delta y_j} + \frac{1}{\nabla y_j} \right) c_{0,j} + \frac{c_{0,j-1}}{\nabla y_j} \right] \\
 & - \frac{b}{(1+b)} \cdot \frac{(u_{0,j+1} - u_{0,j-1})}{\mu \delta y_j \cdot \Delta x_0} - c_{0,j} = 0 , \quad (4.7)
 \end{aligned}$$

where  $0 < j < 2N$ .

Equations similar to (4.7) can be derived for points on other parts of the boundary including the corners.

We now consider the discretisation of equation (3.2) on the non-uniform grid. We integrate equation (3.2) over the region  $r_{i,j}$  and apply Green's theorem to obtain

$$\int_{s_{i,j}} \frac{1}{c} \cdot \frac{\partial u}{\partial n} ds = 0 \quad (4.8)$$

At internal parts we obtain the following approximation

$$\begin{aligned} & \frac{(y_{j+1} - y_{j-1})}{(c_{i+1,j} + c_{i,j})} \cdot \frac{(u_{i+1,j} - u_{i,j})}{(x_{i+1} - x_i)} + \frac{(y_{j+1} - y_{j-1})}{(c_{i-1,j} + c_{i,j})} \cdot \frac{(u_{i-1,j} - u_{i,j})}{(x_i - x_{i-1})} \\ & + \frac{(x_{i+1} - x_{i-1})}{(c_{i,j+1} + c_{i,j})} \cdot \frac{(u_{i,j+1} - u_{i,j})}{(y_{j+1} - y_j)} + \frac{(x_{i+1} - x_{i-1})}{(c_{i,j-1} + c_{i,j})} \cdot \frac{(u_{i,j-1} - u_{i,j})}{(y_j - y_{j-1})} = 0 \end{aligned}$$

This can be written in the more convenient form

$$\begin{aligned} & \frac{1}{\mu \delta x_i} \left[ \frac{4 u_{i+1,j}}{\Delta x_i (c_{i+1,j} + c_{i,j})} - \left( \frac{4}{\Delta x_i (c_{i+1,j} + c_{i,j})} + \frac{4}{\nabla x_i (c_{i-1,j} + c_{i,j})} \right) u_{i,j} \right. \\ & \left. + \frac{4 u_{i-1,j}}{\nabla x_i (c_{i-1,j} + c_{i,j})} \right] + \frac{1}{\mu \delta y_j} \left[ \frac{4 u_{i,j+1}}{\Delta y_j (c_{i,j+1} + c_{i,j})} \right. \\ & \left. - \left( \frac{4}{\Delta y_j (c_{i,j+1} + c_{i,j})} + \frac{4}{\nabla y_j (c_{i,j-1} + c_{i,j})} \right) u_{i,j} + \frac{4 u_{i,j-1}}{\nabla y_j (c_{i,j-1} + c_{i,j})} \right] \\ & = 0 \end{aligned} \quad (4.9)$$

where  $0 < i < N$ ,  $0 < j < 2N$ .

Equations similar to (4.9) can be obtained for points which lie on the lines  $x=0$  and  $x=2$ . It can easily be seen that these equations are equivalent to those obtained on the uniform grid by putting  $\mu \delta x_i = h$ ,  $\Delta x_i = h$ ,  $\nabla x_i = h$  etc.

The finite difference equations on the non-uniform grid are solved using the D.A.D.I. algorithm described in Chapter 3. The problem was solved on two non-uniform grids defined by the following grid stretching functions:

$$\begin{aligned} (a) \quad x(\xi) &= 1 + \frac{\tan^{-1} \{ \lambda (\xi - \frac{1}{2}) \}}{\tan^{-1} \{ \frac{1}{2} \lambda \}}, \\ y(\eta) &= \frac{4 \tan^{-1} \{ \mu \eta \}}{\tan^{-1} \{ \mu \}}; \end{aligned} \quad (4.10)$$



$$\begin{aligned}
 \text{(b)} \quad x(\xi) &= 2 \sin^2\{\tfrac{1}{2}\pi\xi\} \\
 y(\eta) &= 4\{1 - (1 - \eta)^2\}
 \end{aligned} \tag{4.11}$$

The values of  $\lambda$  and  $\mu$  were chosen to be 1 and .5 respectively. These grid stretching functions were chosen to give a smaller spacing of the grid points in the neighbourhoods of the singularities. The following constant grid intervals were chosen:

- (i)  $\Delta\xi = 0.25$  ,  $\Delta\eta = 0.125$  ;
- (ii)  $\Delta\xi = 0.125$  ,  $\Delta\eta = 0.0625$  ;
- (iii)  $\Delta\xi = 0.0625$  ,  $\Delta\eta = 0.03125$  .

The grids defined by (i), (ii) and (iii) have respectively  $5 \times 9$ ,  $9 \times 17$  and  $17 \times 33$  points. The non-uniform grids generated by (ii) and defined by the mappings (4.10) and (4.11) are shown in Figures 4.2(a) and 4.2(b) respectively. For the grid shown in Figure 4.2(a) we have  $\Delta x_0 = 0.226$ ,  $\Delta y_0 = 0.882$ ,  $\nabla y_{2N} = 0.037$ . For the grid shown in Figure 4.2(b) we have  $\Delta x_0 = 0.076$ ,  $\Delta y_0 = 0.484$ ,  $\nabla y_{2N} = 0.016$ .

We compare the results obtained on these non-uniform grids with those obtained on a uniform grid with the same number of grid points in each direction. We used as our stopping criterion that the maximum value of the difference between successive iterates at the grid points is less than  $10^{-6}$  in magnitude..

Table 4.1 gives the number of D.A.D.I. steps required to attain the tolerance. Table 4.2 gives the corresponding run time in seconds. The values of  $c(x,y)$  in the corners at the points  $x=0$ ,  $y=4$  and  $x=2$ ,  $y=4$  are given in Table 4.3. We can see from these results that the D.A.D.I. method with grid (b) takes longer to converge than with grid (a). This is because

in grid (b) the points are closer together in the neighbourhoods of the singularities.

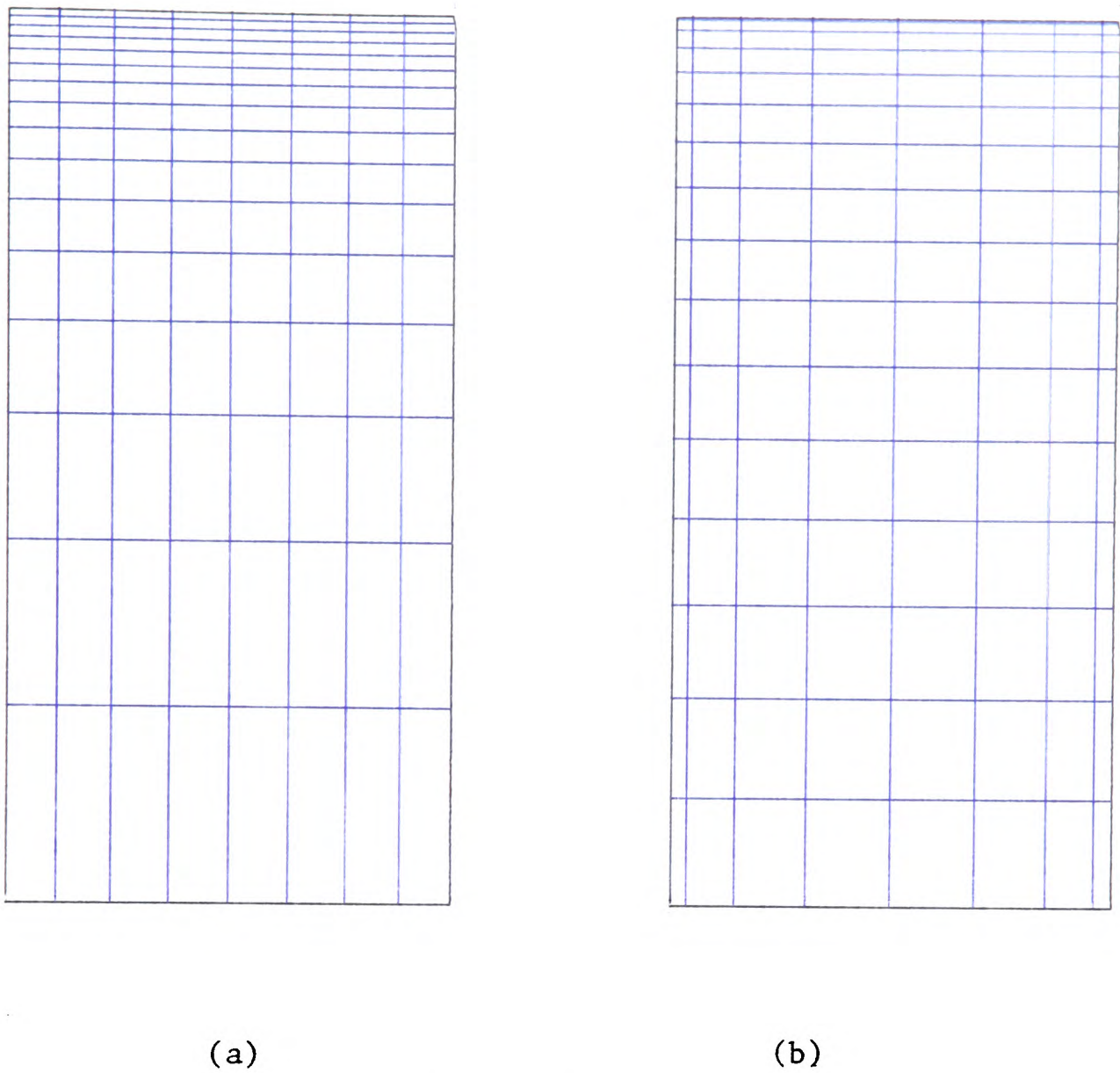


Figure 4.2

Mesh	Uniform Grid	Grid (a)	Grid (b)
5 x 9	204	312	348
9 x 17	288	372	708
17 x 33	552	540	1548

Table 4.1    Number of D.A.D.I. steps

Mesh	Uniform Grid	Grid (a)	Grid (b)
5 x 9	1.2	1.9	2.0
9 x 17	5.2	8.1	15.0
17 x 33	66.1	74.0	211.2

Table 4.2 Computational time

Mesh	Values	Uniform Grid	Grid (a)	Grid (b)
5 x 9	c(0,4)	0.0591	0.0581	0.0629
	c(2,4)	0.0196	0.0188	0.0194
9 x 17	c(0,4)	0.0682	0.0693	0.0758
	c(2,4)	0.0203	0.0200	0.0208
17 x 33	c(0,4)	0.0749	0.0766	0.0819
	c(2,4)	0.0209	0.0209	0.0216

Table 4.3 Values of c(x,y) in the top corners

#### 4.4 Transformation Method

In this method we use a suitable transformation to map the problem in the neighbourhoods of the singularities to one which is free of singularities. As our starting point we assume that we have solved this problem in the rectangle R using a uniform grid. We aim to improve the accuracy of our approximation near the singularities.

Consider the corner at the point  $x=0, y=4$ . The rectangle R is translated so that this point lies at the origin. Let the sector  $S_M$  be defined by

$$S_M = \{(r, \theta) : 0 \leq r \leq Mh, -\frac{1}{2}\pi \leq \theta \leq 0\}$$

where M is a positive integer and h is the mesh size of the uniform grid. This situation is illustrated in Figure 4.3.

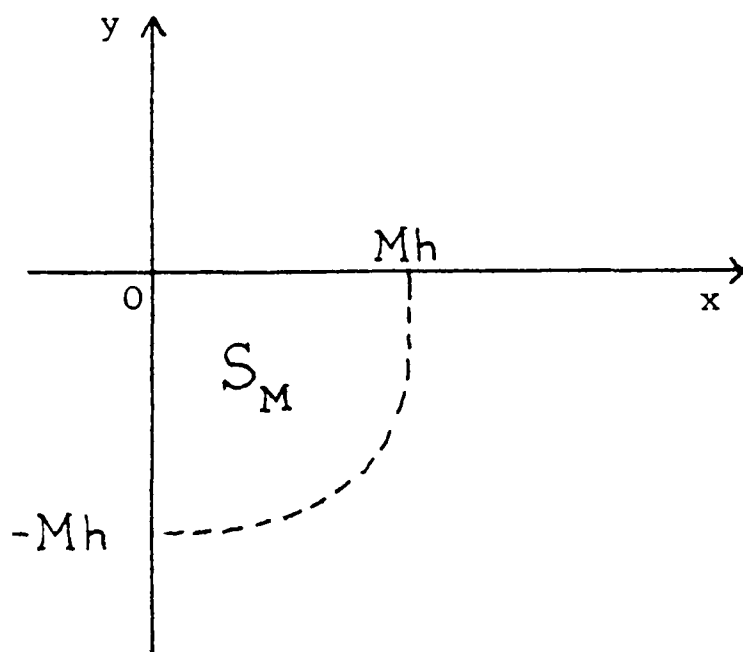


Figure 4.3

By means of the transformation

$$\rho = -\log r, \quad \theta = \tan^{-1}(y/x) \quad (4.12)$$

where  $r^2 = x^2 + y^2$ , the sector  $S_M$  is transformed to the semi-infinite strip  $T_M$  where

$$T_M = \{(\rho, \theta) : -\log(Mh) \leq \rho < \infty, -\frac{1}{2}\pi \leq \theta \leq 0\}$$

With the change of variable (4.12) the partial differential equations (3.1) and (3.2) become

$$\frac{\partial^2 c}{\partial \rho^2} + \frac{\partial^2 c}{\partial \theta^2} = e^{-2\rho} c \quad (4.13)$$

and

$$\frac{\partial}{\partial \rho} \left( \frac{1}{c} \frac{\partial u}{\partial \rho} \right) + \frac{\partial}{\partial \theta} \left( \frac{1}{c} \frac{\partial u}{\partial \theta} \right) = 0 \quad (4.14)$$

As  $\rho \rightarrow \infty$ ,  $r \rightarrow 0$  so for a sufficiently large value of  $\rho$ , say  $P$ , we are sufficiently close to the origin for both  $c$  and  $u$  to be constant along the arc  $r = e^{-P}$ ,  $-\frac{1}{2}\pi \leq \theta \leq 0$ . So we close off the semi-infinite strip at  $\rho = P$  where  $P > -\log(Mh)$ . Let  $T_{M,P}$  be the rectangle

$$\{(\rho, \theta) : -\log(Mh) \leq \rho \leq P, -\frac{1}{2}\pi \leq \theta \leq 0\}.$$

This is illustrated in Figure 4.4.

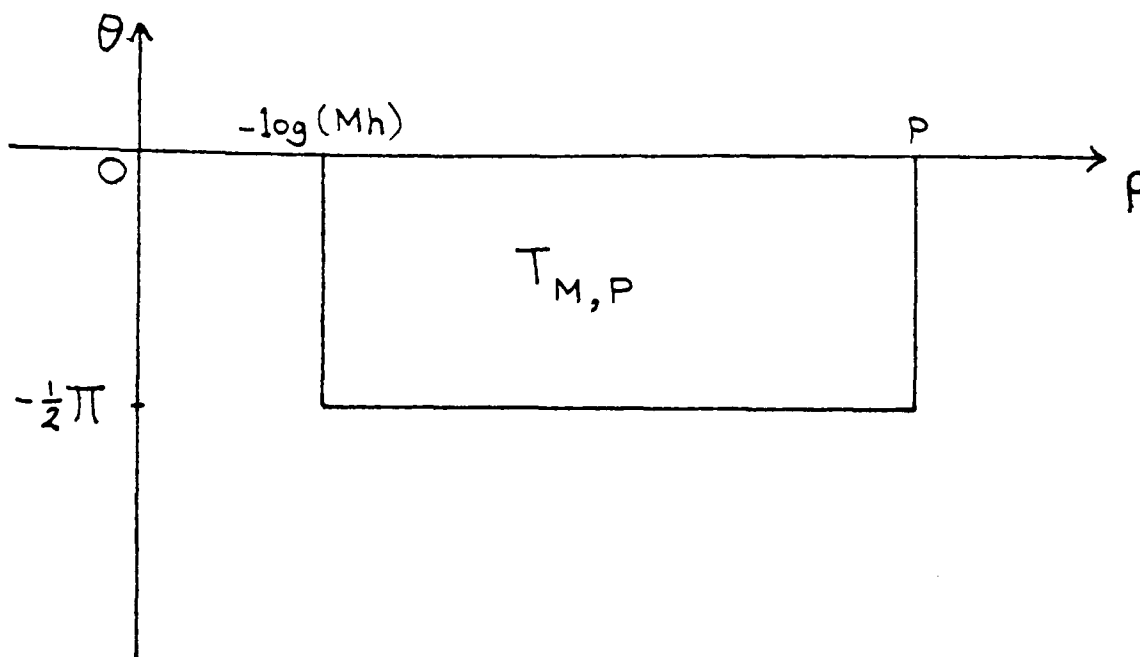


Figure 4.4

We now introduce the boundary conditions used to solve the differential equations (4.13) and (4.14) in the rectangle  $T_{M,P}$ . The lines  $x=0$  ( $-Mh \leq y \leq 0$ ) and  $y=0$  ( $0 \leq x \leq Mh$ ) are mapped under the transformation to the lines  $\theta = -\frac{1}{2}\pi$  and  $\theta = 0$  respectively, for  $-\log(Mh) \leq \rho \leq P$ . Therefore the boundary conditions along these lines are given by

$$\frac{\partial c}{\partial \theta} = \frac{b}{(1+b)} \frac{\partial u}{\partial \rho}, \quad \frac{\partial u}{\partial \theta} = (1+b) \frac{\partial c}{\partial \rho} \quad \text{on } \theta = -\frac{1}{2}\pi,$$

$$\frac{\partial c}{\partial \theta} = -se^{-\rho} c, \quad u = -1 \quad \text{on } \theta = 0$$

Along the line  $\rho = P$  we impose the conditions

$$\frac{\partial c}{\partial \rho} = 0, \quad \frac{\partial u}{\partial \rho} = 0$$

since both  $c$  and  $u$  tend to some limiting value as  $\rho \rightarrow \infty$ . Along the line  $\rho = -\log(Mh)$  we obtain values of  $c$  and  $u$  at certain points by bilinear interpolation. These points will be the mesh points on  $\rho = -\log(Mh)$  and details of the interpolation will be given later.

We are now in a position to solve the partial differential equations

(4.13) and (4.14) together with the associated boundary conditions in the rectangle  $T_{M,P}$ .  $T_{M,P}$  is covered with a rectangular mesh with mesh length  $h_1$  in the  $\rho$ -direction and  $h_2$  in the  $\theta$ -direction. The problem is discretised using the technique described in Section 3.4. The resulting difference equations are solved by the D.A.D.I. method.

The corner situated at the point  $x=2$ ,  $y=4$  can also be treated in a similar fashion. We use the same transformation to map a neighbourhood of this point onto  $T_M$ . The boundary conditions along  $\theta = -\frac{1}{2}\pi$  and  $\theta = 0$  are

$$\frac{\partial c}{\partial \theta} = \frac{1}{(1+b)} \frac{\partial u}{\partial \rho}, \quad \frac{\partial u}{\partial \theta} = \frac{(1+b)}{b} \frac{\partial c}{\partial \rho} \quad \text{on } \theta = -\frac{1}{2}\pi,$$

$$\frac{\partial c}{\partial \theta} = -se^{-\rho}c, \quad u = -1 \quad \text{on } \theta = 0.$$

Let  $W$  be the region which is formed from  $R$  by removing  $(M-1)h$  by  $(M-1)h$  squares in the top right and left hand corners.

#### Algorithm 4.4.1

We give details below of the algorithm used to obtain more accurate approximations to the solution of the Philips problem near the singularities. The algorithm is flowcharted in Figure 4.6.

(a) We solve equations (3.1) and (3.2) in the rectangle  $R$  using a uniform grid and the finite difference equations constructed in Section 3.4. The D.A.D.I. method was used to solve these difference equations. This is described in detail in Algorithm 3.5.1.

(b) Set  $u^I$  and  $c^I$  to be the vectors of the current values of  $u$  and  $c$  respectively at the grid points. Let  $N_1$  and  $N_2$  be integers such that  $N_1 h_1 = P + \log(Mh)$  and  $N_2 h_2 = \frac{1}{2}\pi$ . We require the values of  $u$  and  $c$  at the points of intersection of the lines  $y = -k\pi/(2N_2)$ ,  $k = 0, 1, \dots, N_2$ , with the boundary  $r = Mh$  of the sector  $S_M$ . These will then be the values at the mesh points on the line  $\rho = -\log(Mh)$  in the  $\rho$ - $\theta$  plane. This procedure refers to the transformation of the left-hand sector. A similar procedure is performed

for the right-hand sector.

The interpolation is described for the particular case shown in Figure 4.5. Since we can determine the positions of the points A and B we can find the values of  $u$  and  $c$  at these points by linearly interpolating values at C and D, and D and E respectively. We then linearly interpolate these values at A and B to obtain values at F.

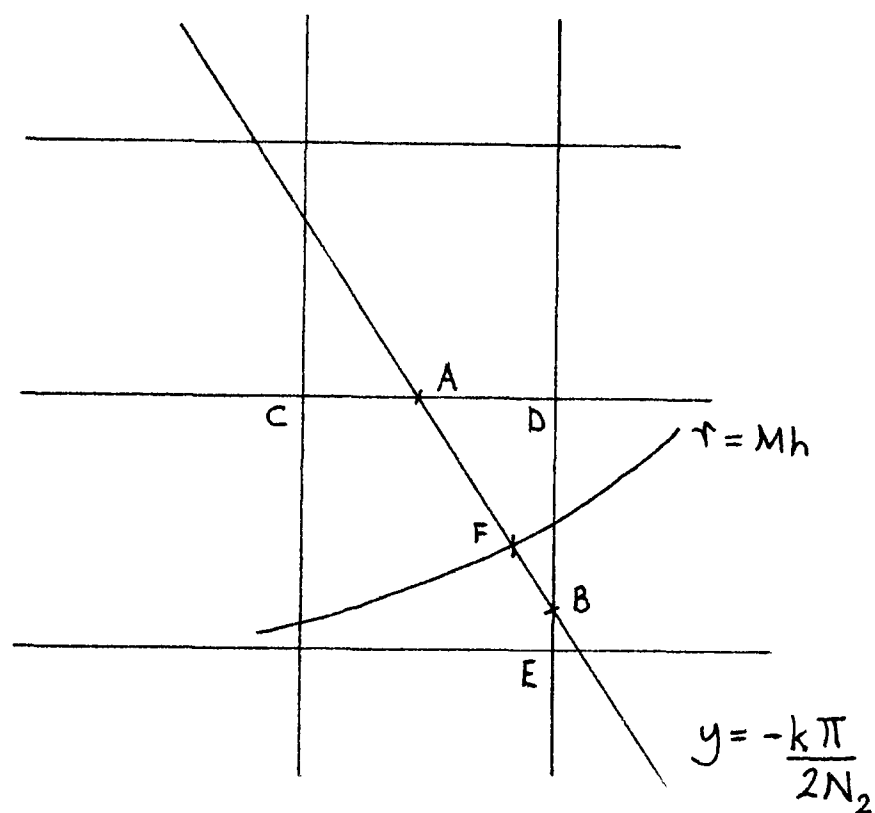


Figure 4.5

(c) Solve the transformed equations (4.13) and (4.14) in each of the rectangles  $T_{M,P}$  starting from some initial guess. When we come to Step (c) other than for the first time we use the previous values to start the iteration. Again we use the D.A.D.I. method to solve the discretised equations. The stopping criterion is that the maximum modulus of the difference in successive iterates is less than  $10^{-4}$ .

(d) Use linear interpolation to calculate values of  $u$  and  $c$  at points of  $T_{M,P}$  which correspond to mesh points of  $R$  lying within the sectors  $S_M$ . This updates the values of  $u$  and  $c$  at the mesh points within  $S_M$ .

(e) Solve the discretised forms of equations (3.1) and (3.2) in  $W$  using the D.A.D.I. method. We use the same stopping criterion as in Step (c).

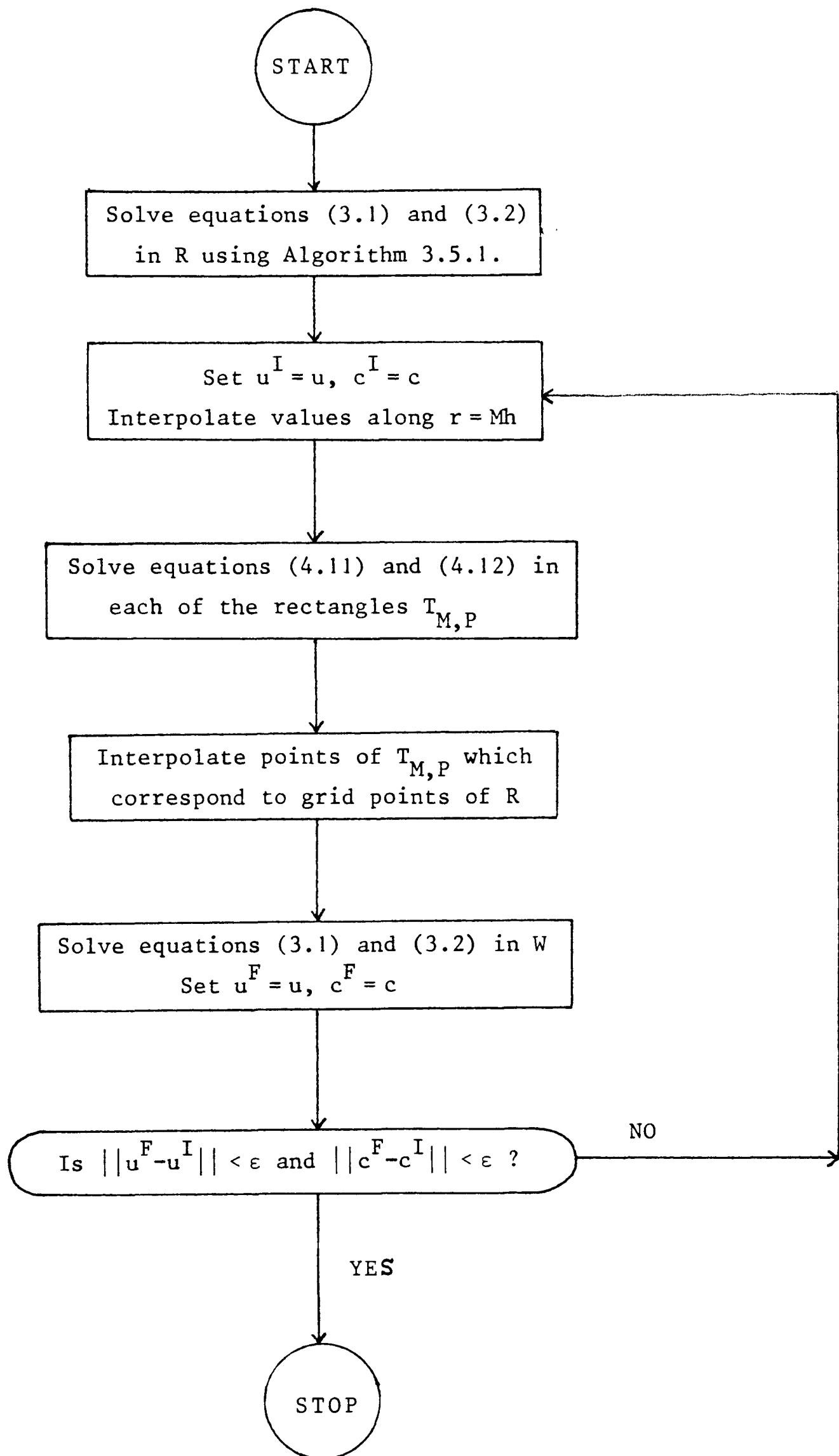


Figure 4.6



Set  $u^F$  and  $c^F$  to be the vectors of the current values of  $u$  and  $c$  respectively at the grid points of  $R$ .

(f) If  $\|u^F - u^I\|_\infty < \epsilon$  and  $\|c^F - c^I\|_\infty < \epsilon$ , where  $\epsilon$  is some tolerance, then the algorithm is terminated. If not, go to Step (b).

Numerical results are shown below. The mesh length,  $h$ , of the uniform grid on  $R$  was chosen to be 0.125. The values of  $N_1$ ,  $N_2$  and  $h_1$  were chosen to be 37, 4 and 0.25 respectively. This means that in the case when  $M=2$  we have that  $P=10.63$ . This value of  $P$  corresponds to the value of  $r=2.4 \times 10^{-5}$ . The tolerance,  $\epsilon$ , was chosen to be  $10^{-4}$ . Figure 4.7 shows the two grids i.e. those on  $R$  and  $S_M$ , superposed, for  $M=6$ . The value of  $M$  was chosen so that the two sectors  $S_M$  do not intersect. So, in this case we have that  $M \leq 7$ . The values of  $c$  in the top left-hand and right-hand corners are shown in Table 4.4 for various values of  $M$ .

$M$	$c(0,4)$	$c(2,4)$
2	0.0831	0.0211
3	0.0789	0.0210
4	0.0767	0.0211
5	0.0768	0.0209
6	0.0769	0.0212

Table 4.4

#### 4.5 Conclusions

The iterative solution of finite difference equations constructed on a non-uniform grid usually presents great difficulties. This is due to the problem of finding suitable parameters for the acceleration of convergence of any selected iterative method. Hence, an advantage of the D.A.D.I. method over standard iterative methods for solving problems of this type is that we do not require an a priori choice of parameters to accelerate convergence.

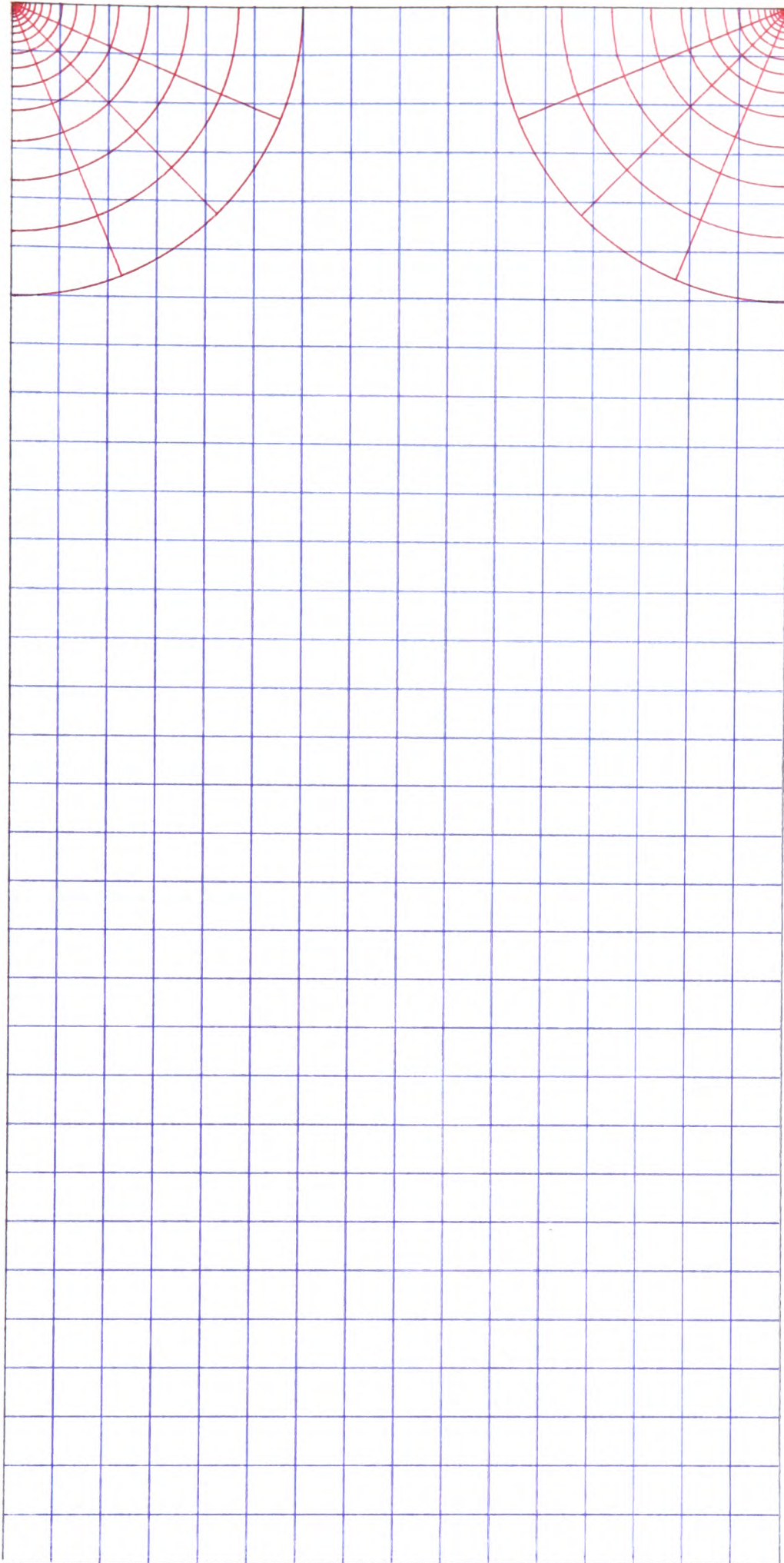


Figure 4.7

Although the local truncation error of the finite difference approximations to the first and second derivatives on the non-uniform grid are formally second order in  $\Delta\xi$ , the term of  $O(\Delta\xi^2)$  contains derivatives of the stretching function which will be large in certain areas if there is strong mesh stretching and large variation in the grid. The local truncation error of the first derivative is given by the expression

$$-(\Delta\xi)^2 \left\{ \frac{1}{4} \frac{d^2x}{d\xi^2} \frac{\partial^2 v}{\partial x^2} + \frac{1}{6} \left( \frac{dx}{d\xi} \right)^2 \frac{\partial^3 v}{\partial x^3} \right\} + O(\Delta\xi^3).$$

Near a singularity the higher derivatives of  $v$  become large and therefore the coefficient of  $(\Delta\xi)^2$  is small provided that the derivatives of the stretching function are small there. For the grid transformation given by (4.11) this is not the case and so the coefficient of  $O(\Delta\xi^2)$  of the local truncation error of the resulting approximation near the singularities is not small.

Tables 4.5 and 4.6 are tables of difference of  $c(0,y)$  in terms of equal intervals in the stretched co-ordinates given by (4.10) and (4.11) respectively. These can be compared with the corresponding tables in Chapter 3. The differences in Table 4.5 are reasonably well-behaved suggesting that the approximation obtained using the grid given by (4.10) is fairly accurate. The differences in Table 4.6 appear to be rather less satisfactory.

However, it is preferable to leave ourselves with the task of solving a non-singular problem, when this is possible, for which numerical methods of finite difference type have a much sounder basis. This is an advantage of the transformation method which maps the problem in the vicinities of the singularities into a problem free of singularities. From Table 3.8 we see that the effects of the singularities are local in nature and so the finite difference approximation computed at points at a reasonable distance from the singularities is accurate.

y	c(0,y)		$\delta^2$		$\delta^4$		$\delta^6$
4.000	0.0766						
		58					
3.982	0.0824		-17				
		41		7			
3.963	0.0865		-10		-1		
		31		6		-3	
3.942	0.0896		-4		-4		6
		27		2		3	
3.920	0.0923		-2		-1		-2
		25		1		1	
3.897	0.0948		-1		0		-1
		24		1		0	
3.872	0.0972		0		0		
		24		1			
3.845	0.0996		1				
		25					
3.816	0.1021						

Table 4.5 Differences of c(0,y) with grid (a)

From the above discussion we would expect the results from the D.A.D.I. method with grid defined by (4.10) and the transformation method to be accurate and indeed there is good agreement between them. However, it is worth noting that the D.A.D.I. method with a variable grid is easier to implement than the transformation method.

y	c(0,y)		$\delta^2$		$\delta^4$		$\delta^6$
4.000	0.0819						
		13					
3.996	0.0832		7				
		20		-3			
3.984	0.0852		4		5		
		24		2		-5	
3.965	0.0876		6		0		-3
		30		2		-8	
3.938	0.0906		8		-8		25
		38		-6		17	
3.902	0.0942		2		9		-31
		40		3		-14	
3.859	0.0982		5		-5		
		45		-2			
3.809	0.1027		3				
		48					
3.750	0.1075						

Table 4.6 Differences of c(0,y) with grid (b)

## Chapter 5

### THE BIHARMONIC EQUATION

#### 5.1 Introduction

Let  $\Omega$  be an open bounded domain in  $\mathbb{R}^2$  with boundary  $\Gamma$  and define  $\bar{\Omega} = \Omega \cup \Gamma$ . Consider the following biharmonic problem in  $\bar{\Omega}$ :

$$\nabla^4 u(x,y) = \frac{\partial^4 u}{\partial x^4} + 2 \frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4} = f(x,y), \quad (x,y) \in \Omega, \quad (5.1)$$

$$u(x,y) = g_1(x,y), \quad (x,y) \in \Gamma, \quad (5.2)$$

and

$$\frac{\partial u}{\partial n}(x,y) = g_2(x,y), \quad (x,y) \in \Gamma, \quad (5.3)$$

where  $\partial/\partial n$  is the derivative in the direction of the outward normal to the boundary and  $f$ ,  $g_1$  and  $g_2$  are given functions defined respectively on  $\Omega$  and  $\Gamma$ . The operator  $\nabla^4$  is known as the biharmonic operator. Applications of problems involving equations of the form (5.1) occur in linear elasticity and in fluid flow.

In this chapter we investigate A.D.I. and multigrid methods for solving the biharmonic equation in the unit square. We begin by reviewing an A.D.I. method, proposed by Conte and Dames(1958), for solving the biharmonic equation and show how the D.A.D.I. method can be extended to solve problems of this type. We finish by looking at various multigrid methods including one which uses A.D.I. as its smoothing procedure. A problem from elasticity is used to compare these methods.

#### 5.2 Finite Difference Equations

Suppose that  $\Omega$  is the open region  $\{(x,y) : 0 < x < 1, 0 < y < 1\}$ . We cover  $\Omega$  with a square grid of mesh length  $h = 1/N$  where  $N$  is a positive integer. Let  $u_{i,j}$  be the value of  $u(x,y)$  at the grid point  $(x_i, y_j)$  where  $x_i = ih$  and  $y_j = jh$ . At a grid point  $(x_i, y_j)$  for which  $2 < i, j < N-2$ , it can be shown using Taylor's series that a finite difference approximation to (5.1) is

$$\begin{aligned}
\frac{1}{h^4} \left\{ 20 u_{i,j} - 8(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) \right. \\
+ 2(u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1}) \\
\left. + (u_{i+2,j} + u_{i-2,j} + u_{i,j+2} + u_{i,j-2}) \right\} = f(x_i, y_j) \quad (5.4)
\end{aligned}$$

The local truncation error of this discretization is  $O(h^2)$  provided that the sixth order derivatives of  $u$  are bounded over  $\Omega$ . The above approximation to the biharmonic operator can be represented by the 13-point finite difference molecule of Figure 5.1.

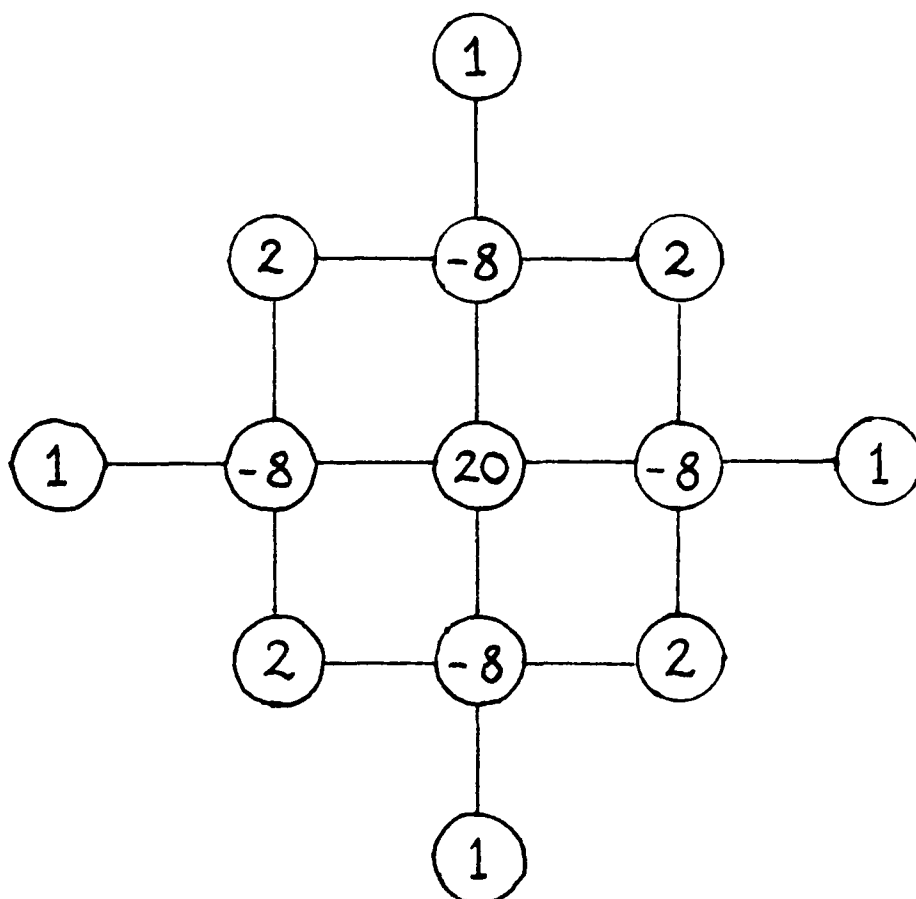


Figure 5.1

The finite difference approximation to (5.1) at grid points in  $\Omega$  which are one mesh length from the boundary is constructed by applying (5.4) at these points and eliminating the exterior imaginary points using a central difference approximation to the Neumann boundary condition (5.3). We introduce the following notation:

$$\delta_x^2 u_{i,j} = u_{i+1,j} - 2u_{i,j} + u_{i-1,j},$$

and

$$\delta_x^4 u_{i,j} = u_{i+2,j} - 4 u_{i+1,j} + 6 u_{i,j} - 4 u_{i-1,j} + u_{i-2,j} .$$

With this notation the approximation (5.4) to the biharmonic problem (5.1) can be written in the form

$$\delta_x^4 u_{i,j} + 2 \delta_y^2 \delta_x^2 u_{i,j} + \delta_y^4 u_{i,j} = h^4 f(x_i, y_j) . \quad (5.5)$$

### 5.3 A.D.I. Method

In the A.D.I. method we convert equation (5.1) to the parabolic equation

$$\frac{\partial u}{\partial t} + \nabla^4 u = f(x, y) \quad (5.6)$$

whose steady state solution, if one exists, solves (5.1). The A.D.I. scheme described here was proposed for biharmonic problems by Conte and Dames (1958). The scheme is an extension of a method developed by Douglas and Rachford (1956) for heat conduction problems.

The A.D.I. scheme of Conte and Dames for solving (5.6) is

$$(rI + \delta_x^4) u_{i,j}^{(n+1)} = (rI - 2 \delta_y^2 \delta_x^2 - \delta_y^4) u_{i,j}^{(n)} + h^4 f_{i,j} , \quad (5.7)$$

and

$$(rI + \delta_y^4) u_{i,j}^{(n+2)} = r u_{i,j}^{(n+1)} + \delta_y^4 u_{i,j}^{(n)} , \quad (5.8)$$

where  $r = h^4/\Delta t$ . Equations (5.7) and (5.8) represent one double sweep of the A.D.I. iteration. We are not really interested in solving the parabolic equation (5.6) accurately for finite times, but in reaching the steady state solution as quickly as possible. Bearing this in mind we can treat  $r$  as a parameter which may be varied after each double sweep of the iteration in order to accelerate convergence.

The finite difference equations (5.7) are solved along horizontal lines. Along each of these lines we have a system of  $N-1$  equations in  $N-1$  unknowns. The coefficient matrices of these systems are symmetric, positive definite and  $\text{quidiagonal}$ . This means that the solution of these



systems by Gaussian elimination and back-substitution is stable without the need for interchanges. An efficient algorithm for solving quidiagonal systems is given in Appendix I. Similarly, the finite difference equations (5.8) are solved along vertical lines.

To investigate the convergence of the above scheme we eliminate  $u_{i,j}^{(n+1)}$  from (5.7) and (5.8) to obtain

$$(rI + \delta_x^4)(rI + \delta_y^4) u_{i,j}^{(n+2)} = (rI - \delta_x^2 \delta_y^2)^2 u_{i,j}^{(n)} + h^4 f_{i,j}. \quad (5.9)$$

We define the error by

$$e_{i,j}^{(n)} = u_{i,j}^{(n)} - u_{i,j}. \quad (5.10)$$

The error then satisfies the following equation

$$(rI + \delta_x^4)(rI + \delta_y^4) e_{i,j}^{(n+2)} = (rI - \delta_x^2 \delta_y^2)^2 e_{i,j}^{(n)}. \quad (5.11)$$

The eigenfunctions of (5.11) are of the form

$$e_{i,j}^{(n)} = A \sin\left(\frac{pi\pi}{N}\right) \sin\left(\frac{qj\pi}{N}\right)$$

where A is a constant. A proof of this is given by Heller (1960).

Suppose that

$$e_{i,j}^{(n+2)} = \lambda_{p,q} e_{i,j}^{(n)}, \quad p, q = 1, 2, \dots, N-1. \quad (5.12)$$

We substitute (5.12) into (5.11) to obtain

$$\lambda_{p,q} (rI + \delta_x^4)(rI + \delta_y^4) e_{i,j}^{(n)} = (rI - \delta_x^2 \delta_y^2)^2 e_{i,j}^{(n)}. \quad (5.13)$$

It can easily be verified that

$$\delta_x^2 e_{i,j}^{(n)} = -4 \sin^2(p\pi/2N) e_{i,j}^{(n)},$$

and

$$\delta_y^2 e_{i,j}^{(n)} = -4 \sin^2(q\pi/2N) e_{i,j}^{(n)}.$$

Substituting these two expressions into (5.13) we obtain

$$\lambda_{p,q} = \frac{(r - 16 \sigma_p^2 \sigma_q^2)^2}{(r + 16 \sigma_p^4)(r + 16 \sigma_q^4)} \quad (5.14)$$

where

$$\sigma_p = \sin(p\pi/2N)$$

and

$$\sigma_q = \sin(q\pi/2N) .$$

From (5.14) we can see that  $0 \leq \lambda_{p,q} < 1$  for all  $r > 0$ . Hence the A.D.I. scheme given by (5.7) and (5.8) is convergent for all positive values of  $r$ .

Conte and Dames (1958) show how to determine a parameter sequence for the biharmonic problem. They used the following sequence of parameters

$$r_k = \frac{16}{(0.2)^{1-k}}, \quad k = 1, \dots, M, \quad (5.15)$$

where  $M$  is defined by

$$M - 1 < 1 + \frac{4 \log(\sigma_1)}{\log(0.2)} \leq M.$$

This choice of parameters results in a reduction of  $10^{-2}$  in the initial error per cycle.

#### 5.4 Dynamic A.D.I. Method

We show how the dynamic A.D.I. (D.A.D.I.) method described in Chapter 1 can be extended to the biharmonic equation. In the notation of Chapter 1 the discretized form of equation (5.1) can be written in the form

$$(H + V)^2 \underline{u} = \underline{b} \quad (5.16)$$

where  $\underline{u}$  is the vector containing the unknown values of  $u$  at the grid points. The matrices  $H$  and  $V$  are symmetric, positive definite and commuting. The following analysis holds for any matrices  $H$  and  $V$  with

these properties. The fact that  $H$  and  $V$  commute means that they have a common set of eigenvectors (see Heller (1960), for example). For the analysis of the step size strategy we make the assumption that the error at each stage is concentrated mainly in a single eigenspace. So we assume that the error  $\underline{e}^{(n)}$  at the  $n$ th step is an eigenfunction for  $H$  and  $V$  corresponding to eigenvalues  $a$  and  $b$  respectively.

We are able to compute the convergence factor and the test parameter as functions of  $a\Delta t$  and  $b\Delta t$ . Substituting (5.14) into (5.12) we obtain

$$\underline{e}^{(n+2)} = \frac{(1 - ab\Delta t)^2}{(1 + a^2\Delta t)(1 + b^2\Delta t)} \underline{e}^{(n)} = R(a\Delta t, b\Delta t) \underline{e}^{(n)}. \quad (5.17)$$

Hence the convergence factor,  $CF$ , for two double sweeps is given by

$$CF = \frac{\|\underline{e}^{(n+4)}\|}{\|\underline{e}^{(n)}\|} = R^2(a\Delta t, b\Delta t).$$

The test parameter,  $TP$ , is given by

$$TP = \frac{\|\underline{e}^{(n+4)} - \tilde{\underline{e}}^{(n+4)}\|}{\|\underline{e}^{(n+4)} - \underline{e}^{(n)}\|} = \left| \frac{R^2(a\Delta t, b\Delta t) - R(2a\Delta t, 2b\Delta t)}{R^2(a\Delta t, b\Delta t) - 1} \right|$$

where  $\tilde{\underline{e}}^{(n+4)} = \tilde{\underline{u}}^{(n+4)} - \underline{u}$  and  $\tilde{\underline{u}}^{(n+4)}$  is the result of an A.D.I. double sweep with step size  $2\Delta t$  starting from  $\underline{u}^{(n)}$ .

Our objective is to find a step size strategy which keeps  $\Delta t$  within a region of fast convergence. This is done by investigating graphically whether it is possible to make  $CF$  small by changing  $\Delta t$  to keep  $TP$  within a certain range. As before we can normalize by assuming  $b \leq a = 1$ . We then consider the graphs of  $(CF)^{\frac{1}{2}}$  and  $TP$  as functions of  $\Delta t$  for different values of  $b$ . Four of these graphs were plotted on a log-log scale, for  $b = 1, 0.5, 0.1, 0.05$ , and are shown in Figures 5.2 to 5.5. We notice that  $(CF)^{\frac{1}{2}}$  is quite small provided that  $TP$  is in the range  $(0.1, 0.3]$ . We find that we can employ the same strategy as the one devised for second order equations. So when  $TP$  falls in the intervals  $(-\infty, 0.05]$ ,  $(0.05, 0.1]$ ,

$(0.1, 0.3]$ ,  $(0.3, 0.4]$ ,  $(0.4, 0.6]$  one accepts the present step and changes  $\Delta t$  by the factor of 4, 2, 1,  $1/2$ ,  $1/4$  respectively for the next step. If TP falls in the interval  $(0.6, \infty)$  we reject the present step and start the step again with  $\Delta t$  reduced by  $1/16$ . We recall that a D.A.D.I. step comprises the following operations:

- (i) perform two double sweeps of A.D.I. with time step  $\Delta t$ ,
- (ii) return to the beginning of the step and perform a single A.D.I. double sweep with time step  $2\Delta t$ ,
- (iii) compute TP and change  $\Delta t$  for the next step.

With this strategy the shifted  $\Delta t$  quite often falls within a region of fast convergence. In designing this strategy the aim has been to damp out the higher order eigencomponents using small  $\Delta t$  before increasing  $\Delta t$  to damp out the lower order eigencomponents.

Doss and Miller (1979) show that the strategy described in Chapter 1 appears to give good results for second order equations in situations which are beyond the realm of the assumptions. Therefore, we would expect the D.A.D.I. method described in this chapter to work well on more general fourth order problems like

$$a(x,y) \frac{\partial^4 u}{\partial x^4} + b(x,y) \frac{\partial^4 u}{\partial x^2 \partial y^2} + c(x,y) \frac{\partial^4 u}{\partial y^4} = d(x,y) .$$

The A.D.I. method will probably be less efficient in solving such problems because of the difficulty in finding a suitable parameter sequence.

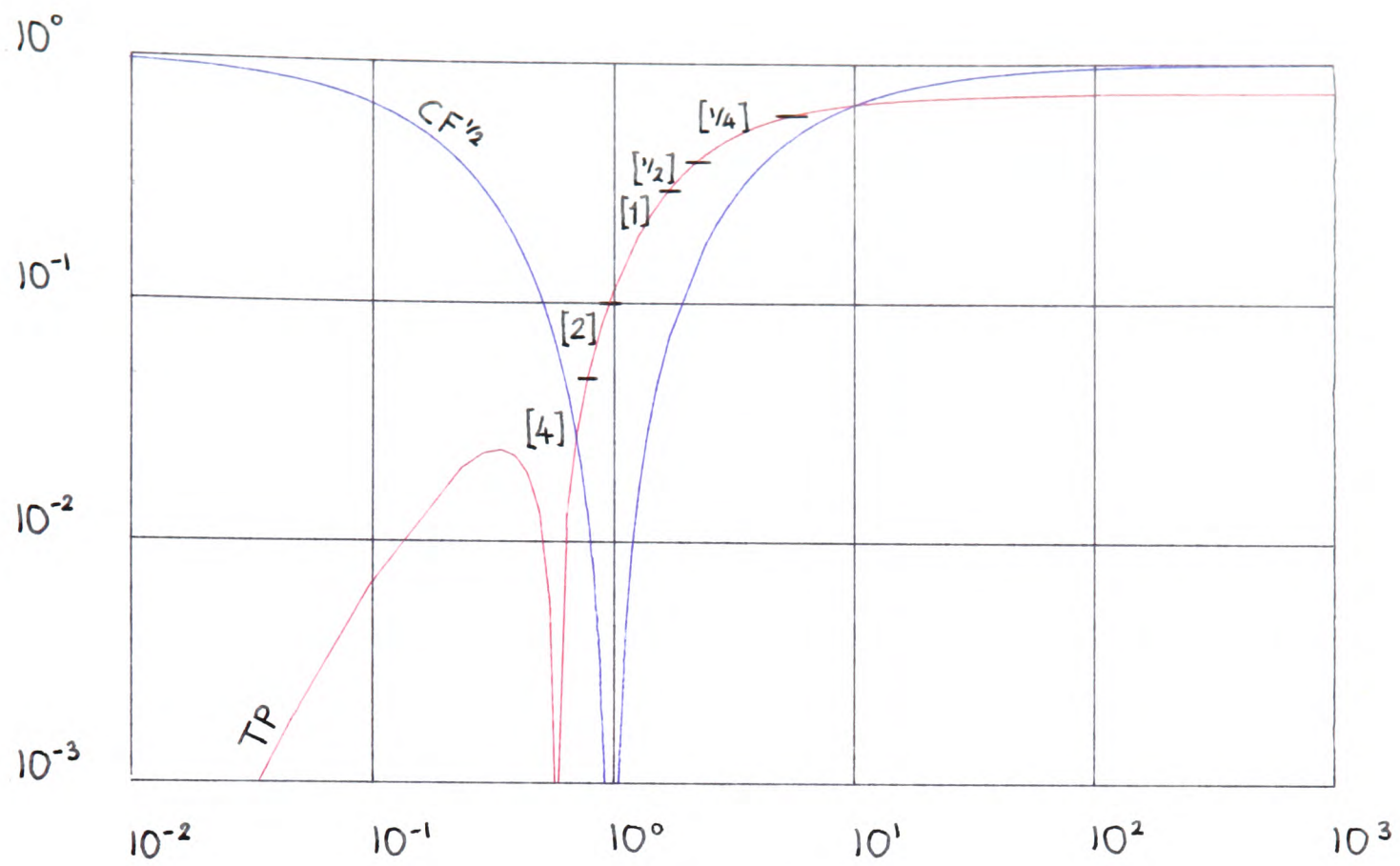


Figure 5.2.  $(CF)^{1/2}$  and  $TP$  vs.  $\Delta t$  for  $b = 1$

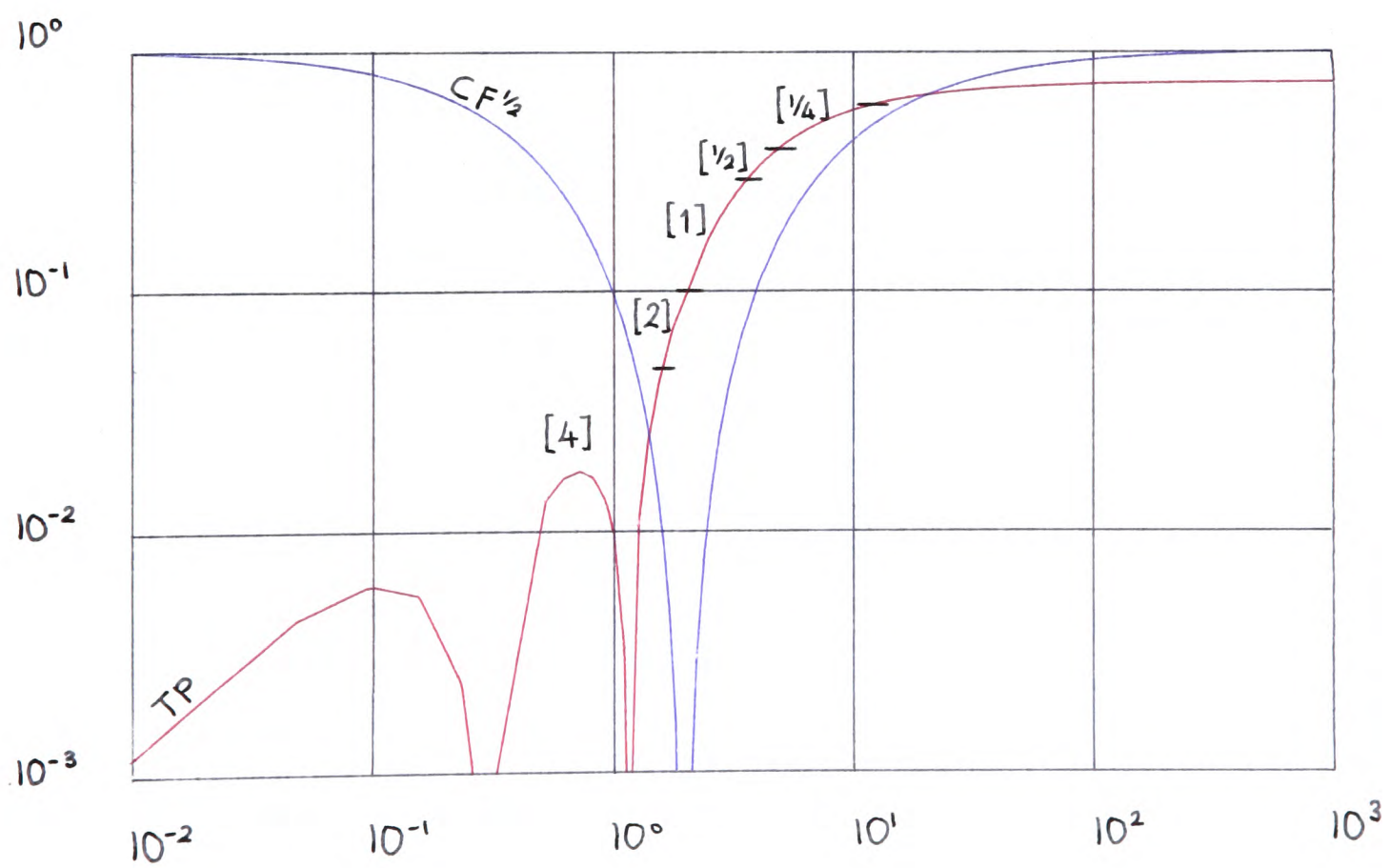


Figure 5.3.  $(CF)^{1/2}$  and  $TP$  vs.  $\Delta t$  for  $b = 0.5$

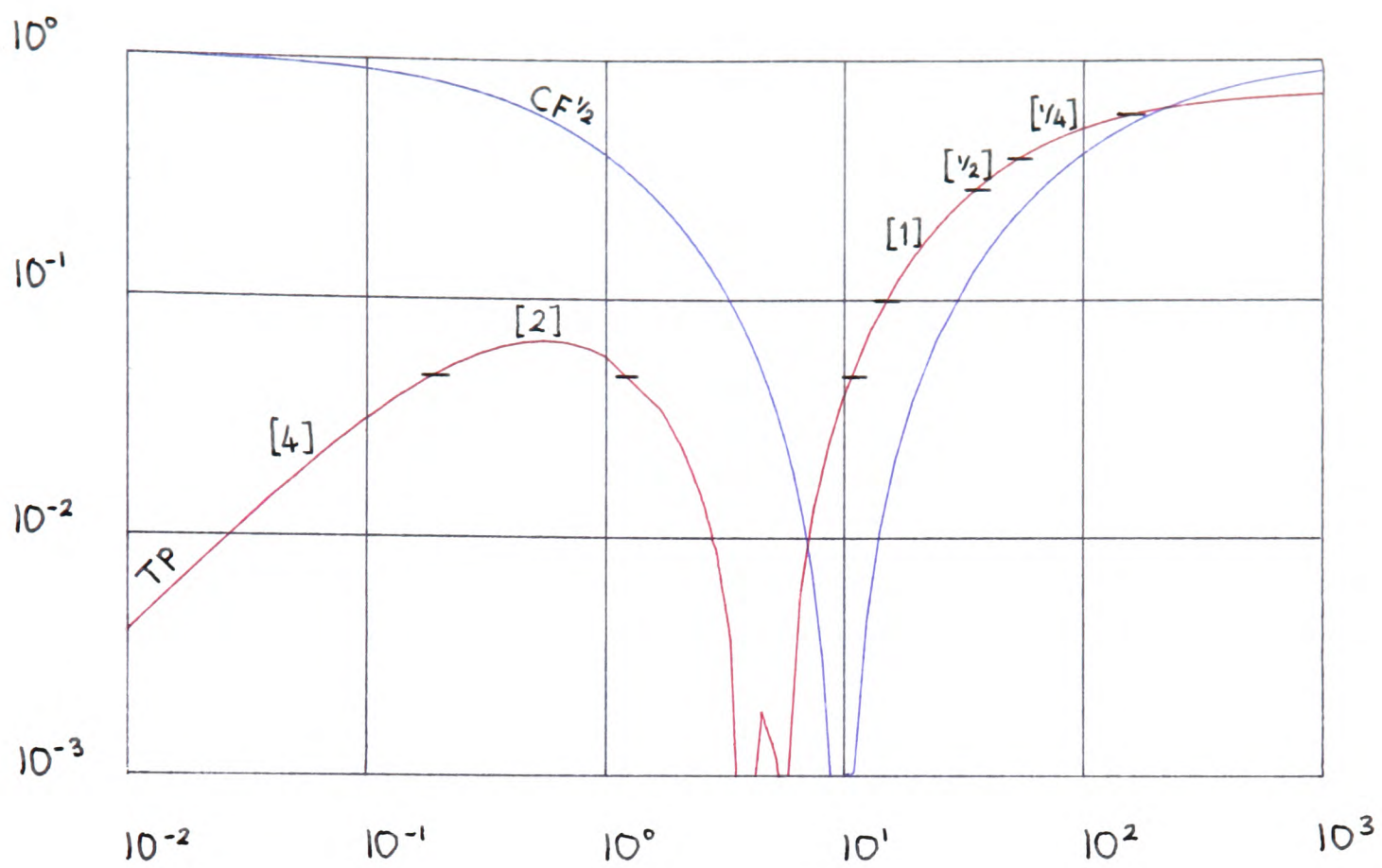


Figure 5.4.  $(CF)^{\frac{1}{2}}$  and TP vs.  $\Delta t$  for  $b = 0.1$ .

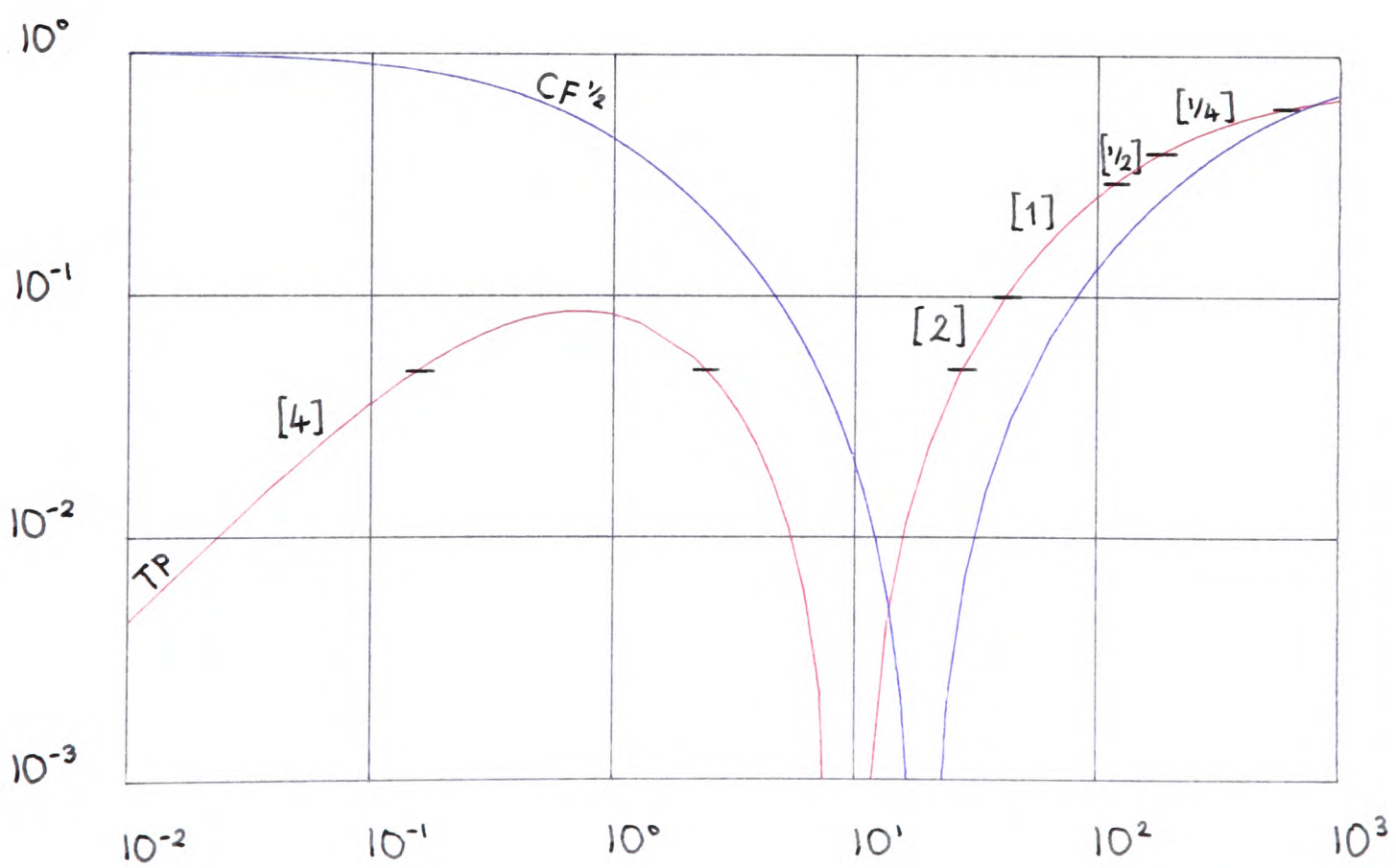


Figure 5.5.  $(CF)^{\frac{1}{2}}$  and TP vs.  $\Delta t$  for  $b = 0.05$ .

### 5.5. Multigrid using point relaxation

Let  $G_1, G_2, \dots, G_M$  be a sequence of grids approximating the same region  $\Omega$  with corresponding mesh sizes  $h_1, h_2, \dots, h_M$ . Let  $h_k = 2h_{k+1}$ ,  $k = 1, \dots, M-1$ . Since the biharmonic equation is linear the Correction Storage (C.S.) scheme is used. This scheme is described in Chapter 2. On any grid  $G_k$  we have, from equation (5.4), that

$$\begin{aligned} \frac{1}{h_k^4} \left\{ 20 u_{i,j}^k - 8(u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k) \right. \\ + 2(u_{i+1,j-1}^k + u_{i+1,j+1}^k + u_{i-1,j-1}^k + u_{i-1,j+1}^k) \\ + (u_{i+2,j}^k + u_{i-2,j}^k + u_{i,j+2}^k + u_{i,j-2}^k) \left. \right\} \\ = f_{i,j}^k, \end{aligned} \quad (5.18)$$

for  $i = 3, \dots, N_k - 2$ ,  $j = 3, \dots, N_k - 2$  where  $N_k h_k = 1$ . For  $k < M$ ,  $u^k$  denotes some correction to  $u^{k+1}$ , and  $f^k$  denotes some modified right-hand side.

When  $i = 2$  the finite difference equations are

$$\begin{aligned} \frac{1}{h_k^4} \left\{ 21 u_{2,j}^k - 8(u_{3,j}^k + u_{1,j}^k + u_{2,j+1}^k + u_{2,j-1}^k) \right. \\ + (2(u_{3,j-1}^k + u_{3,j+1}^k + u_{1,j-1}^k + u_{1,j+1}^k) \\ + (u_{4,j}^k + u_{2,j+2}^k + u_{2,j-2}^k) \left. \right\} \\ = f_{2,j}^k, \end{aligned} \quad (5.19)$$

for  $j = 3, \dots, N_k - 2$ . Similar expressions are obtained along  $x = (N_k - 1)h_k$  for  $j = 3, \dots, N_k - 2$ , and along  $y = h_k$  and  $y = (N_k - 1)h_k$  for  $i = 3, \dots, N_k - 2$ . At the point for which  $i = 2$  and  $j = 2$  the finite difference equation is

$$\frac{1}{h_k^4} \left\{ 22 u_{2,2}^k - 8(u_{3,2}^k + u_{1,2}^k + u_{2,3}^k + u_{2,1}^k) \right. \\
+ 2(u_{3,1}^k + u_{3,3}^k + u_{1,1}^k + u_{1,3}^k) \\
\left. + (u_{4,2}^k + u_{2,4}^k) \right\} = f_{2,2}^k \quad (5.20)$$

Similar expressions can be obtained for the points corresponding to

$$i = 2, j = N_k - 1; i = N_k - 1, j = 2; i = N_k - 1, j = N_k - 1.$$

### Components of the multigrid procedure

#### (a) relaxation

As our smoothing routine for the biharmonic problem we use pointwise Gauss-Seidel relaxation and consider two orderings of the grid points. The first is the lexicographic ordering which yields a smoothing factor of 0.8 by performing an analysis similar to that given in Section 2.3 for Laplace's equation. The second orders the points in the four-coloured manner. This means that in performing a relaxation sweep over the above finite difference equations, the points are ordered in the following way:

- (i)  $(x_i, y_j)$  with  $i$  and  $j$  both odd,
- (ii)  $(x_i, y_j)$  with  $i$  and  $j$  both even,
- (iii)  $(x_i, y_j)$  with  $i$  odd and  $j$  even,
- (iv)  $(x_i, y_j)$  with  $i$  even and  $j$  odd.

This ordering is illustrated in Figure 5.6 with points of kind (i), (ii), (iii) and (iv) denoted by R, B, G and Y respectively for  $N = 8$ .

#### (b) fine-to-coarse transfer of values

The transfer of residuals to the coarse grid is carried out by full-weighting, i.e.

$$u_{i,j}^{k-1} = \frac{1}{4} u_{i,j}^k + \frac{1}{8} (u_{2i+1,2j}^k + u_{2i-1,2j}^k + u_{2i,2j+1}^k + u_{2i,2j-1}^k) \\
+ \frac{1}{16} (u_{2i+1,2j+1}^k + u_{2i+1,2j-1}^k + u_{2i-1,2j+1}^k + u_{2i-1,2j-1}^k) .$$



	R	Y	R	Y	R	Y	R
	G	B	G	B	G	B	G
	R	Y	R	Y	R	Y	R
	G	B	G	B	G	B	G
	R	Y	R	Y	R	Y	R
	G	B	G	B	G	B	G
	R	Y	R	Y	R	Y	R

Figure 5.6. Four-coloured ordering of grid points

### (c) interpolation

Here we use the rule that the order of interpolation should be no less than the order of the differential equation. Brandt (1977) shows that if a smaller order of interpolation is used then even a small and smooth residual function may produce large high frequency residuals, and a significant amount of computational work will be required to smooth them out. This effect was demonstrated in numerical experiments by Brandt (1973). Hence, cubic interpolation is used to transfer the correction to the fine grid. For example,

$$u_{2i,2j}^k = u_{2i,2j}^k + u_{i,j}^{k-1},$$

$$u_{2i+1,2j}^k = u_{2i+1,2j}^k + \frac{1}{16} (-u_{i-1,j}^{k-1} + 9u_{i,j}^{k-1} + 9u_{i+1,j}^{k-1} - u_{i+2,j}^{k-1})$$

### 5.6 Multigrid using A.D.I.

Firstly, we investigate the smoothing properties of the Douglas-Rachford A.D.I. scheme for the biharmonic equation. The Douglas-Rachford

method is described by the following iterative scheme:

$$(rI + \delta_x^4) u_{r,s}^{(n+1)} = (rI - 2\delta_y^2 \delta_x^2 - \delta_y^4) u_{r,s}^{(n)} + h^4 f_{r,s},$$

$$(rI + \delta_y^4) u_{r,s}^{(n+2)} = r u_{r,s}^{(n+1)} + \delta_y^4 u_{r,s}^{(n)}.$$

Let  $w_{r,s}^{(n)} = u_{r,s} - u_{r,s}^{(n)}$ ,  $w_{r,s}^{(n+1)} = u_{r,s} - u_{r,s}^{(n+1)}$ ,  $w_{r,s}^{(n+2)} = u_{r,s} - u_{r,s}^{(n+2)}$

where  $u_{r,s}$  is the true solution of the difference equations. To study the

$\theta = (\theta_1, \theta_2)$  Fourier component of the error functions  $w^{(n)}$ ,  $w^{(n+1)}$  and  $w^{(n+2)}$  we put

$$w_{r,s}^{(n)} = A_\theta e^{i(r\theta_1 + s\theta_2)}, \quad w_{r,s}^{(n+1)} = B_\theta e^{i(r\theta_1 + s\theta_2)}, \quad w_{r,s}^{(n+2)} = C_\theta e^{i(r\theta_1 + s\theta_2)}.$$

The convergence factor of the  $\theta$  component over one A.D.I. sweep is given by

$$\mu(\theta) = \left| \frac{C_\theta}{A_\theta} \right| = \frac{(r - 16\sin^2(\frac{1}{2}\theta_1)\sin^2(\frac{1}{2}\theta_2))^2}{(r + 16\sin^4(\frac{1}{2}\theta_1))(r + 16\sin^4(\frac{1}{2}\theta_2))} \quad (5.21)$$

We want to find the maximum value of  $\mu$  over those values of  $\theta$  for which  $|\theta|$  lies in the range  $[\frac{1}{2}\Pi, \Pi]$  where  $|\theta| = \max(|\theta_1|, |\theta_2|)$ . This is the range of high frequency components i.e. the range of components which cannot be approximated on the coarser grid.

Let  $x = \sin^2(\frac{1}{2}\theta_1)$ , then since (5.21) is symmetric in  $\theta_1$  and  $\theta_2$  we can consider the problem of finding the maximum value of  $\mu$  over those values of  $\theta$  for which  $\frac{1}{2}\Pi \leq \theta_1 \leq \Pi$  i.e.  $\frac{1}{2} \leq x \leq 1$ .

Now we can write

$$\mu(\theta) = \frac{g(x)}{(r + 16p^4)},$$

where

$$g(x) = \frac{(r - 16p^2x)^2}{(r + 16x^2)}$$

and  $p = \sin(\frac{1}{2}\theta_2)$ . So our first problem is to find the maximum value of

$g(x)$  in the interval  $[\frac{1}{2}, 1]$  for fixed  $r$  and  $p$ .

We can differentiate  $g(x)$  to obtain

$$g'(x) = \frac{-32r (r - 16p^2x) (x + p^2)}{(r + 16x^2)^2}.$$

So  $g'(x) = 0$  when  $x = r/16p^2$  or  $x = -p^2$ . Since  $g(r/16p^2) = 0$ , the maximum value of  $g$  in the interval occurs at one of the end points of the interval. We consider both cases separately.

(i) Suppose that the maximum value of  $g$  occurs at  $x = \frac{1}{2}$ . Let  $y = \sin^2(\frac{1}{2}\theta_2)$  and consider

$$f(y) = \frac{(r - 8y)^2}{(r + 4)(r + 16y^2)}$$

where  $0 \leq y \leq 1$ . We want to find the maximum value of  $f(y) = \mu(\frac{1}{2}\Pi, \theta)$  for  $y$  in the range  $[0, 1]$

$$\text{Now } f'(y) = - \frac{16r(r + 4)(r - 8y)(2y + 1)}{(r + 4)^2(r + 16y^2)^2}$$

and therefore  $f'(y) = 0$  when  $y = r/8$  or  $y = -\frac{1}{2}$ . Since  $f(r/8) = 0$  the maximum value of  $f$  in  $[0, 1]$  occurs at one of the end points.

(ii) Suppose that the maximum values of  $g$  occurs at  $x = 1$ . Let  $y = \sin^2(\frac{1}{2}\theta_2)$  and consider

$$\ell(y) = \frac{\ell(r - 16y)^2}{(r + 16)(r + 16y^2)}$$

where  $0 \leq y \leq 1$ . We want to find the maximum value of  $\ell(y) = \mu(\Pi, \theta_2)$  for  $y$  in the range  $[0, 1]$ . We find that  $\ell'(y) = 0$  when  $y = r/16$  or  $y = -1$ . Since  $\ell(r/16) = 0$  the maximum value of  $\ell$  in  $[0, 1]$  occurs at one of the end points.

From the above results it follows that

$$\bar{\mu} = \max \left\{ \frac{r}{(r + 4)}, \frac{(r - 8)^2}{(r + 4)(r + 16)}, \frac{r}{(r + 16)}, \left( \frac{(r - 16)}{(r + 16)} \right)^2 \right\}$$

The following inequalities hold:

$$(a) \quad \frac{r}{(r+4)} > \frac{r}{(r+16)} \quad \text{for all } r > 0 ,$$

$$(b) \quad \left( \frac{r-16}{r+16} \right)^2 > \frac{r}{(r+4)} \quad \text{for } 0 < r < 3.2 ,$$

$$(c) \quad \frac{(r-8)^2}{(r+4)(r+16)} > \frac{r}{(r+4)} \quad \text{for } 0 < r < \sqrt{5}-1 ,$$

$$(d) \quad \frac{(r-8)^2}{(r+4)(r+16)} > \left( \frac{r-16}{r+16} \right)^2 \quad \text{for } r > 80/7$$

$$\therefore \bar{\mu} = \begin{cases} \mu(\Pi, \Pi) = \left( \frac{r-16}{r+16} \right)^2 , & \text{if } 0 < r \leq 3.2 , \\ \mu(\frac{1}{2}\Pi, 0) = \frac{r}{(r+4)} , & \text{if } r \geq 3.2 . \end{cases}$$

So the value of  $r$  which minimizes  $\bar{\mu}$  is  $r = 3.2$  which yields the smoothing factor  $\bar{\mu} = 4/9$ . The Douglas-Rachford A.D.I. scheme is incorporated into a multigrid method. The method is the same as the one described in Section 5.5 with the exception that at the smoothing stage a double sweep of A.D.I. is performed instead of a Gauss-Seidel sweep of the finite difference equations.

### 5.7 Numerical Results

We apply the numerical methods described in this chapter to the problem of flexure of flat elastic plates. The transverse displacements in a thin flat elastic plate caused by small flexural distortions satisfy the biharmonic equation

$$\nabla^4 u = f .$$

When the edges of the plate are clamped then the displacement  $u(x,y)$  and its normal derivative  $\partial u / \partial n$  are given on the boundary. A detailed description of this problem is given by Southwell (1956).

Consider a square plate, defined by  $-\frac{1}{2} \leq x \leq \frac{1}{2}$ ,  $-\frac{1}{2} \leq y \leq \frac{1}{2}$ , with clamped

edges under uniform transverse pressure. This situation is illustrated in Figure 5.7.

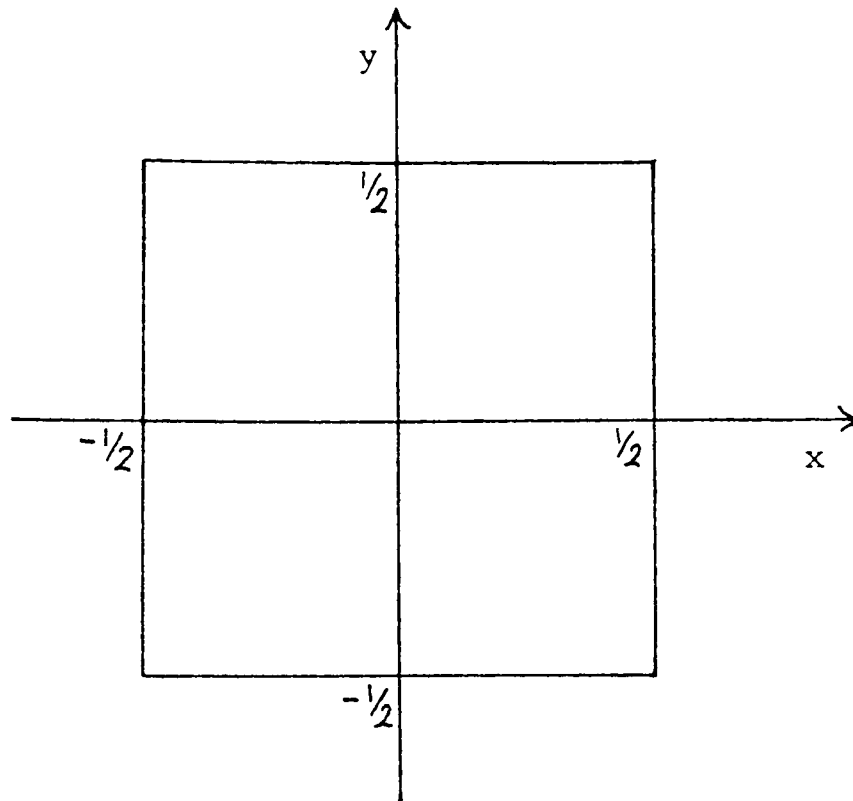


Figure 5.7

We consider the case where  $f = 4096$  and the boundary conditions are

$$u = \frac{\partial u}{\partial n} = 0$$

along the edges  $x = \pm \frac{1}{2}$  and  $y = \pm \frac{1}{2}$ . This problem has 8-fold symmetry. In the calculations which follow, however, no advantage is taken of this fact.

We first compare the following A.D.I. methods on this problem:

- (a) ADI. The Douglas-Rachford A.D.I. method with the parameter sequence given by (5.15).
- (b) DADI. The dynamic A.D.I. method.

The algorithms were terminated when the  $\ell_2$ -norm of the residuals was less than  $10^{-4}$ . Tables 5.1 and 5.2 show how the number of double sweeps and run time vary respectively with grid size. For DADI the number in brackets after the number of double sweeps denotes the number of D.A.D.I. steps.

Method	$h = 1/8$	$h = 1/16$	$h = 1/32$	$h = 1/64$
ADI	52	65	84	103
DADI	77(26)	122(41)	152(51)	188(63)

Table 5.1 Number of double sweeps

Method	$h = 1/8$	$h = 1/16$	$h = 1/32$	$h = 1/64$
ADI	0.5	1.6	6.4	30.9
DADI	0.9	2.7	11.6	57.6

Table 5.2 Computational time (seconds)

For this example we see that ADI is more efficient than DADI. However, for more difficult problems it is probable that we will not be able to determine a suitable sequence of parameters for the A.D.I. method.

We turn our attention now to the following multigrid methods:

- (a) MGLEX. The C.S. scheme is used with Gauss-Seidel for smoothing with the points ordered lexicographically.
- (b) MGFOCOL. This is the same as MGLEX but with the relaxation performed with the points ordered in the four-coloured manner.
- (c) MGADI. The C.S. scheme is used with A.D.I. ( $r = 3.2$ ) as its smoothing procedure.

We define a work unit as in Chapter 2. So for the multigrid method with Gauss-Seidel relaxation, for example, we define a work unit to be the computational work in one relaxation sweep over the finest grid  $G_M$ . The grid size on the coarsest grid is  $h = 0.125$ . The value of the switching parameter  $\eta$  in MGLEX and MGFOCOL was chosen to be 0.8 and 0.6 respectively. We took the value of  $\delta$  (see Algorithm 2.4.1) to be 0.2. For MGADI we used the same choice of parameters as for MGFOCOL. The algorithms were again terminated when the  $\ell_2$ -norm of the residuals was less than  $10^{-4}$ .

Tables 5.3 and 5.4 show how the number of work units and run time vary respectively with grid size.

Method	$h = 1/16$	$h = 1/32$	$h = 1/64$
MGLEX	216	189	161
MGFOCOL	202	142	129
MGADI	88	88	70

Table 5.3 Number of work units

Method	$h = 1/16$	$h = 1/32$	$h = 1/64$
MGLEX	3.7	10.4	31.7
MGFOCOL	4.0	9.0	29.2
MGADI	2.6	8.9	27.2

Table 5.4 Computational time (seconds)

These three methods are comparable in terms of computational time. We see that fewer work units are required when we relax the equations using the four-coloured ordering as opposed to the lexicographic ordering. This demonstrates that for this problem the smoothing factor per sweep of the Gauss-Seidel iterative method using the four-coloured ordering is smaller than that using the lexicographic ordering. In using MGADI on this problem we have been able to determine a value of  $r$  which minimizes the smoothing factor per sweep. For more difficult problems it is likely that we will not be able to do this. It is interesting to note that the number of work units decreases as the grid size decreases and more grids are incorporated into the multigrid procedure.

### 5.8 The Coupled Equation Approach

The biharmonic problem given by equations (5.1) to (5.3) can also be formulated in the following way:

$$\nabla^2 u = v, \quad (5.22)$$

$$\nabla^2 v = f, \quad (5.23)$$

defined in the domain  $\Omega$ , with the boundary conditions

$$u(x,y) = g_1(x,y), \quad (5.24)$$

$$\frac{\partial u}{\partial n}(x,y) = g_2(x,y), \quad (5.25)$$

on  $\Gamma$ . Equations (5.22) and (5.23) represent a coupled pair of second order equations. These equations are discretized using the standard five-point approximation to the Laplacian operator. We consider a multi-grid procedure for solving this coupled pair of equations. A difficulty with this formulation of the problem is that equation (5.22) has two boundary conditions whereas equation (5.23) has none.

On any grid  $G_k$ ,  $k=1,\dots,M$ , equations (5.22) and (5.23) are discretized as follows:

$$u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k - 4u_{i,j}^k = h_k^2 v_{i,j}^k, \quad (5.26)$$

$$v_{i+1,j}^k + v_{i-1,j}^k + v_{i,j+1}^k + v_{i,j-1}^k - 4v_{i,j}^k = h_k^2 f_{i,j}^k, \quad (5.27)$$

where  $i,j=1,2,\dots,N_k-1$ . Here we are assuming that  $\Omega$  is the open region  $\{(x,y) : 0 < x < 1, 0 < y < 1\}$ . On the boundary of  $\Omega$  the values of  $u$  are given. At mesh points along the line  $x=0$  we apply the difference formula (5.26), dropping the superscript  $k$ , to obtain

$$u_{1,j} + u_{-1,j} + u_{0,j+1} + u_{0,j-1} - 4u_{0,j} = h^2 v_{0,j}. \quad (5.28)$$

We use central differences to approximate the Neumann boundary condition (5.25). Therefore at mesh points along  $x=0$  we have



$$u_{1,j} - u_{-1,j} = 2h g_2(0,jh). \quad (5.29)$$

Eliminating  $u_{-1,j}$  between (5.28) and (5.29) we obtain

$$v_{0,j} = \frac{2u_{1,j} + u_{0,j+1} + u_{0,j-1} - 2hg_2(0,jh) - 4u_{0,j}}{h^2}, \quad (5.30)$$

$j = 1, \dots, N_k - 1$ . All values on the right-hand side of (5.30) are known except  $u_{1,j}$ . This means that we shall need to update the boundary values  $v_{0,j}$  at the beginning of each relaxation sweep. Equations similar to (5.30) are obtained along the other parts of the boundary.

One relaxation sweep over the grid  $G_k$  is defined by the following steps:

- (i) set up the boundary conditions for the finite difference equations (5.27) using equations (5.30) and similar ones,
- (ii) perform a relaxation sweep of equations (5.27) using Gauss-Seidel with the points ordered lexicographically,
- (iii) perform a relaxation sweep of equations (5.26) using Gauss-Seidel with the points ordered lexicographically.

Since the differential equations (5.22) and (5.23) are linear the correction storage algorithm is used. The residuals are transferred to coarser levels of discretization by injection. For the coarse-to-fine transfer we use linear interpolation for the correction to  $v$  values. For the correction to  $u$  values we consider both linear and cubic interpolation. Here again we use the rule that the order of interpolation should not be less than the order of the differential equation. Now  $u$  satisfies the second order equation (5.22) so we can use linear interpolation. However,  $u$  also satisfies the biharmonic equation so we are also justified in using cubic interpolation. We have two boundary conditions associated with  $u$  which means that cubic interpolation can be efficiently implemented near the boundaries.

The method was used to solve the problem of flexure of flat elastic

plates described in the previous section. We denote by MGLIN and MGCUB the methods which use linear and cubic interpolation respectively for the coarse-to-fine transfer of the correction to  $u$  values. The grid size of the coarsest grid is  $0.25$ . The algorithms were terminated when the  $\ell_2$ -norm of the residuals was less than  $10^{-6}$ .

Tables 5.5 and 5.6 show how the number of work units and run time vary with grid size respectively.

Method	$h = 1/8$	$h = 1/16$	$h = 1/32$	$h = 1/64$
MGLIN	81	100	105	107
MGCUB	59	71	75	83

Table 5.5 Number of work units

Method	$h = 1/8$	$h = 1/16$	$h = 1/32$	$h = 1/64$
MGLIN	0.8	1.8	5.0	17.3
MGCUB	0.9	1.5	4.3	15.8

Table 5.6 Computational time (seconds)

From the above tables we see that fewer work units are required for MGCUB than MGLIN. However, more work is required to perform cubic interpolation than linear interpolation. Taking this into account MGCUB is still the more efficient method in terms of computational time. These results cannot be compared with those in the previous section since here we are solving different finite difference equations.

## Chapter 6

### NATURAL CONVECTION IN AN ENCLOSED CAVITY

#### 6.1 Introduction

The problem described in this chapter was proposed by Jones (1979a) as a comparison problem for testing computer codes for a variety of practical problems. The problem is that of buoyancy driven flow in a vertical, rectangular cavity whose vertical sides are at different temperatures and whose horizontal surfaces are insulated. This is known as the 'double glazing' or 'window cavity' problem and has many applications, particularly in the field of thermal insulation. The most well-known application is double glazing where a stagnant layer of air acts as an insulant between a warm room and a cold outside. The formulation of the problem is in terms of stream function, vorticity and temperature. We eliminate vorticity to obtain a coupled pair of partial differential equations, one of which is fourth order. A D.A.D.I. method for solving these equations numerically is described. For large non-dimensional temperature differences characterized by the Rayleigh number the flow patterns develop strong boundary layers. These boundary layers are resolved using the grid stretching techniques described in Chapter 4. Numerical results and contour plots are given.

#### 6.2 Formulation of the Problem

We are concerned with the problem of fluid flow induced in an upright, rectangular cavity described in non-dimensional terms by  $0 \leq x \leq 1$ ,  $0 \leq z \leq 1$ , with  $z$  vertically upwards. The cavity has different constant temperatures on the vertical walls,  $T_1$  on the hot wall and  $T_2$  on the cooler wall, and has insulated horizontal walls. We shall consider the two-dimensional flow of a Boussinesq fluid of Prandtl number 0.71 in which the flow takes

place perpendicular to the walls. The Boussinesq approximation (see Mallinson and De Vahl Davis (1973)) assumes that the physical properties and the density are constant except in the buoyancy term in the equations where the density is taken into account. This approximation is quite realistic and can give rise to predictions that are in good agreement with experiment (Jones (1979b)) for small temperature differences. The governing equations are considerably simplified by use of this approximation. A detailed description of this problem is given by Jones (1979c).

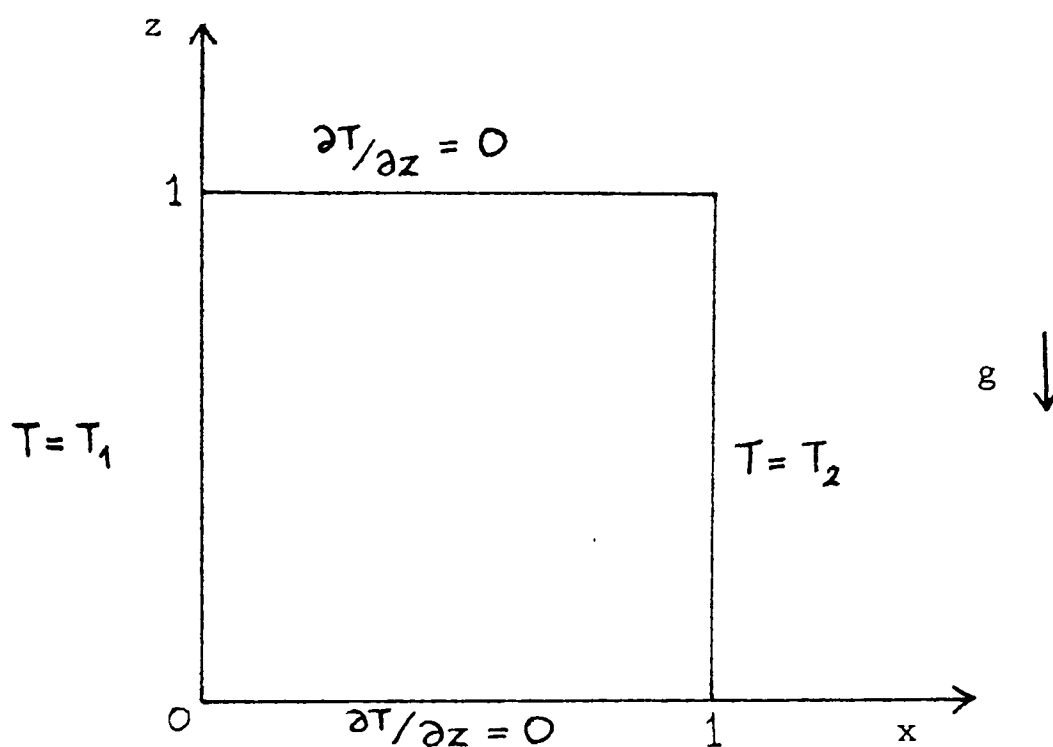


Figure 6.1

A non-dimensional temperature,  $T$ , is defined by

$$T = \frac{(T^* - T_2)}{(T_1 - T_2)},$$

where  $T^*$  is the temperature. The equations representing the conservation of mass, momentum and energy may be written as

$$\nabla \cdot \underline{v} = 0, \quad (6.1)$$

$$(\nabla \times \underline{v}) \times \underline{v} = -\nabla p - Ra Pr \underline{T_k} - Pr \nabla^2 \underline{v}, \quad (6.2)$$

$$\nabla \cdot (\underline{v} T) = \nabla^2 T, \quad (6.3)$$

(Mallinson and De Vahl Davis (1977)) where  $\underline{v} = (u, 0, w)$ ,  $\underline{k} = (0, 0, 1)$  and  $p$  is the perturbation pressure. The Rayleigh number is given by

$$Ra = g \beta (T_1 - T_2) / \kappa \nu$$

and the Prandtl number by

$$Pr = \nu / \kappa ,$$

where  $g$  is the acceleration due to gravity,  $\beta$  the coefficient of volumetric expansion and  $\nu$  the coefficient of kinematic viscosity.

Mallinson and De Vahl Davis (1977) show that the governing equations may be recast in the form

$$Pr \nabla^2 \zeta + Pr Ra \frac{\partial T}{\partial x} = u \frac{\partial \zeta}{\partial x} + w \frac{\partial \zeta}{\partial z} , \quad (6.4)$$

$$-\zeta = \nabla^2 \psi , \quad (6.5)$$

$$\frac{\partial}{\partial x} (uT) + \frac{\partial}{\partial z} (wT) = \nabla^2 T , \quad (6.6)$$

$$u = - \frac{\partial \psi}{\partial z} , \quad w = \frac{\partial \psi}{\partial x} , \quad (6.7)$$

where  $\zeta$  is known as the vorticity and  $\psi$  the stream function. Equations (6.4) to (6.7) represent what is known as the vorticity-stream function formulation of the problem.

The boundary conditions are

$$\psi = \frac{\partial \psi}{\partial n} = 0 , \quad \text{on } \Gamma ,$$

where  $\Gamma$  is the boundary of  $\Omega = \{(x, z) : 0 \leq x \leq 1, 0 \leq z \leq 1\}$  ,

$$T = 1 , \quad \text{on } x = 0 ,$$

$$T = 0 , \quad \text{on } x = 1 ,$$

$$\frac{\partial T}{\partial n} = 0 , \quad \text{on } z = 0 \text{ and } z = 1$$

Eliminating the velocities  $u$ ,  $w$  and the vorticity  $\zeta$  using (6.5) and (6.7) we obtain

$$\nabla^4 \psi - Ra \frac{\partial T}{\partial x} + \frac{1}{Pr} \frac{\partial \psi}{\partial z} \frac{\partial (\nabla^2 \psi)}{\partial x} - \frac{1}{Pr} \frac{\partial \psi}{\partial x} \frac{\partial (\nabla^2 \psi)}{\partial z} = 0, \quad (6.8)$$

$$\nabla^2 T - \frac{\partial (wT)}{\partial z} - \frac{\partial (uT)}{\partial x} = 0. \quad (6.9)$$

The system (6.8) and (6.9) represents a fourth order equation for  $\psi$  and a second order equation for  $T$ . The major interest is in heat transfer so there are two important quantities namely the temperature fields and the overall heat transfer defined by a Nusselt number

$$Nu = \int_0^1 \frac{\partial T}{\partial x} dz \Big|_{x=0}$$

In order to solve this problem using an A.D.I. method we must first convert equations (6.8) and (6.9) to the parabolic equations

$$\begin{aligned} \frac{1}{\lambda} \frac{\partial \psi}{\partial t} + \nabla^4 \psi - Ra \frac{\partial T}{\partial x} + \frac{1}{Pr} \frac{\partial \psi}{\partial z} \frac{\partial (\nabla^2 \psi)}{\partial x} \\ - \frac{1}{Pr} \frac{\partial \psi}{\partial x} \frac{\partial (\nabla^2 \psi)}{\partial z} = 0, \end{aligned} \quad (6.10)$$

$$\frac{\partial T}{\partial t} = \nabla^2 T - \frac{\partial (uT)}{\partial x} - \frac{\partial (wT)}{\partial z}, \quad (6.11)$$

whose steady state solution, if one exists, solves (6.8) and (6.9). The parameter  $\lambda$  plays the same role as it did in the Philips problem of Chapter 3. It means that for  $\lambda \neq 1$  we are effectively using different time scales for the two equations. Mallinson and De Vahl Davis (1973) also suggest this technique.

### 6.3 Solution of a Nonlinear Equation

In a numerical process to find the solution of equations (6.8) and (6.9) we use (6.8) to solve for  $\psi$  and (6.9) to solve for  $T$ . Equation (6.8) is nonlinear with respect to  $\psi$  and it can be written in the form

$$L(\psi) = f(T), \quad (6.12)$$

where  $L$  is a nonlinear operator.

A Newton-type method is used to solve equation (6.12). Suppose that  $\psi^*$  is some approximation to the solution of (6.12). We replace  $L$  by its linearization about  $\psi^*$  and then attempt to partially solve the linearized problem

$$L'(\psi^*) \cdot (\psi - \psi^*) + L(\psi^*) - f(T) \simeq 0. \quad (6.13)$$

We use the D.A.D.I. method to solve this equation. The linearization is updated after each D.A.D.I. step. This will be explained in detail later.

The Fréchet derivative of  $L$  at  $\psi$  (see Appendix II) is given by

$$\begin{aligned} L'(\psi) \cdot \phi &= \nabla^4 \phi + \frac{1}{\text{Pr}} \frac{\partial \psi}{\partial z} \frac{\partial}{\partial x} (\nabla^2 \phi) + \frac{1}{\text{Pr}} \left\{ \frac{\partial}{\partial x} (\nabla^2 \psi) \right\} \frac{\partial \phi}{\partial z} \\ &\quad - \frac{1}{\text{Pr}} \frac{\partial \psi}{\partial x} \frac{\partial}{\partial z} (\nabla^2 \phi) - \frac{1}{\text{Pr}} \left\{ \frac{\partial}{\partial z} (\nabla^2 \psi) \right\} \frac{\partial \phi}{\partial x}. \end{aligned} \quad (6.14)$$

#### 6.4 Finite Difference Equations

We cover  $\Omega$  with a square grid of mesh length  $h = 1/N$  where  $N$  is a positive integer. Let  $\psi_{i,j}$  and  $T_{i,j}$  be the values of  $\psi$  and  $T$  at the point  $(x_i, z_j)$  respectively where  $x_i = ih$  and  $z_j = jh$ . Equations (6.8) and (6.9) are discretized using central differences to give

$$\begin{aligned} \frac{1}{h^4} \{ \delta_x^4 \psi_{i,j} + 2 \delta_y^2 \delta_x^2 \psi_{i,j} + \delta_y^4 \psi_{i,j} \} &- \frac{\text{Ra}(T_{i+1,j} - T_{i-1,j})}{2h} \\ - \frac{1}{\text{Pr}} u_{i,j} \frac{(P_{i+1,j} - P_{i-1,j})}{2h} &- \frac{1}{\text{Pr}} w_{i,j} \frac{(P_{i,j+1} - P_{i,j-1})}{2h} = 0, \end{aligned} \quad (6.15)$$

and

$$\begin{aligned} \frac{1}{h^2} \{ \delta_x^2 T_{i,j} + \delta_y^2 T_{i,j} \} &- \frac{(u_{i+1,j} T_{i+1,j} - u_{i-1,j} T_{i-1,j})}{2h} \\ - \frac{(w_{i,j+1} T_{i,j+1} - w_{i,j-1} T_{i,j-1})}{2h} &= 0, \end{aligned} \quad (6.16)$$

where

$$u_{i,j} = - \frac{(\psi_{i,j+1} - \psi_{i,j-1})}{2h}, \quad (6.17)$$

$$w_{i,j} = \frac{(\psi_{i+1,j} - \psi_{i-1,j})}{2h}, \quad (6.18)$$

$$p_{i,j} = \frac{\delta_x^2 \psi_{i,j} + \delta_y^2 \psi_{i,j}}{h^2}. \quad (6.19)$$

The normal derivative boundary conditions are also discretized using central differences and these are used to remove exterior imaginary points from equations (6.15) and (6.16) near the boundary. So we have  $2N(N-1)$  equations for the  $2N(N-1)$  unknowns  $\psi_{i,j}$ ,  $i = 1, \dots, N-1$ ,  $j = 1, \dots, N-1$ , and  $T_{i,j}$ ,  $i = 1, \dots, N-1$ ,  $j = 0, \dots, N$ .

Equation (6.15) can be written in the form

$$L_h(\psi_{i,j}) = f_h(T_{i,j}),$$

which is the discrete form of equation (6.12), where  $L_h$  is a nonlinear discrete operator.

### 6.5 Method of Solution

We describe how the D.A.D.I. method outlined in Chapter 1 for second order equations and extended in Chapter 5 to fourth order equations can be applied to this problem. Since equations (6.10) and (6.11) are parabolic they may be advanced in time by a direct method and the complete solution procedure may be regarded as a single iterative scheme. Our interest is not in solving the parabolic equations (6.10) and (6.11) accurately for finite times but to reach the steady state solution as soon as possible. We therefore use the D.A.D.I. method which uses a strategy that attempts to keep the time step  $\Delta t$  within a region of fast convergence.

We start from initial guesses  $\underline{T}^{(0)}$  and  $\underline{\psi}^{(0)}$  which are vectors of the unknown values of  $T$  and  $\psi$  respectively at the grid points at time  $t = 0$ . Suppose that we have reached the  $n$ th time step, where  $n = 2m$  and  $m$  is even, and that the current approximations to  $T$  and  $\psi$  at the grid points are  $\underline{T}^{(n)}$  and  $\underline{\psi}^{(n)}$  respectively. Starting from these approximations we



describe a step of the D.A.D.I. method with time step  $\Delta t$ .

We begin by putting

$$e_{i,j}^{(n)} = 0$$

for  $i = 0, 1, \dots, N$ ,  $j = 0, 1, \dots, N$ .

The Peaceman-Rachford A.D.I. scheme is used to advance equation (6.11) in time. We solve the following systems along lines in the x-direction:

$$- T_{i-1,0}^{(n+1)} + (2 + r_1) T_{i,0}^{(n+1)} - T_{i+1,0}^{(n+1)} = 2T_{i,1}^{(n)} - (2 - r_1) T_{i,0}^{(n)}, \quad (6.20)$$

$$i = 1, \dots, N-1,$$

$$- (1 + \frac{1}{2} h u_{i-1,j}^{(n)}) T_{i-1,j}^{(n+1)} + (2 + r_1) T_{i,j}^{(n+1)} - (1 - \frac{1}{2} h u_{i+1,j}^{(n)}) T_{i+1,j}^{(n+1)}$$

$$= (1 + \frac{1}{2} h w_{i,j-1}^{(n)}) T_{i,j-1}^{(n)} - (2 - r_1) T_{i,j}^{(n)} + (1 - \frac{1}{2} h w_{i,j+1}^{(n)}) T_{i,j+1}^{(n)}, \quad (6.21)$$

$$i = 1, \dots, N-1, \quad j = 1, \dots, N-1,$$

$$- T_{i-1,N}^{(n+1)} + (2 + r_1) T_{i,N}^{(n+1)} - T_{i+1,N}^{(n+1)} = 2T_{i,N-1}^{(n)} - (2 - r_1) T_{i,N}^{(n)}, \quad (6.22)$$

$$i = 1, \dots, N-1,$$

with  $T_{o,j}^{(n+1)} = 1$  and  $T_{N,j}^{(n+1)} = 0$  for  $j = 0, 1, \dots, N$ , and where  $r_1 = h^2/\Delta t$ .

The following systems are solved along lines in the z-direction:

$$- (1 + \frac{1}{2} h w_{i,j-1}^{(n)}) T_{i,j-1}^{(n+2)} + (2 + r_1) T_{i,j}^{(n+2)} - (1 - \frac{1}{2} h w_{i,j+1}^{(n)}) T_{i,j+1}^{(n+2)}$$

$$= (1 + \frac{1}{2} h u_{i-1,j}^{(n)}) T_{i-1,j}^{(n+1)} - (2 - r_1) T_{i,j}^{(n+1)} + (1 - \frac{1}{2} h u_{i+1,j}^{(n)}) T_{i+1,j}^{(n+1)}, \quad (6.23)$$

with  $T_{i,-1}^{(n+2)} = T_{i,1}^{(n+2)}$  and  $T_{i,N+1}^{(n+2)} = T_{i,N-1}^{(n+2)}$  for  $i = 1, \dots, N-1$ . In equations

(6.21) and (6.23) we have used the notation

$$u_{i,j}^{(n)} = - \frac{(\psi_{i,j+1}^{(n)} - \psi_{i,j-1}^{(n)})}{2h}, \quad (6.24)$$

and

$$w_{i,j}^{(n)} = \frac{(\psi_{i+1,j}^{(n)} - \psi_{i-1,j}^{(n)})}{2h} . \quad (6.25)$$

We advance  $\psi$  in time using the Douglas-Rachford A.D.I. scheme to solve the fourth order linear equation (6.13). A step of Newton's method is used to update the linearization after each D.A.D.I. step. We solve the following systems, described by  $j=1, \dots, N-1$ , along lines in the x-direction:

$$\begin{aligned} & \left\{ 1 + \frac{h}{2Pr} u_{i,j}^{(n)} \right\} e_{i-2,j}^{(n+1)} - \left\{ 4 + \frac{h}{Pr} u_{i,j}^{(n)} - \frac{h^2}{4Pr} (p_{i,j+1}^{(n)} - p_{i,j-1}^{(n)}) \right\} e_{i-1,j}^{(n+1)} \\ & + \{ 6 + r_2 \} e_{i,j}^{(n+1)} - \left\{ 4 - \frac{h}{Pr} u_{i,j}^{(n)} + \frac{h^2}{4Pr} (p_{i,j+1}^{(n)} - p_{i,j-1}^{(n)}) \right\} e_{i+1,j}^{(n+1)} \\ & + \left\{ 1 - \frac{h}{2Pr} u_{i,j}^{(n)} \right\} e_{i+2,j}^{(n+1)} = \left\{ 8 - \frac{2h}{Pr} w_{i,j}^{(n)} - \frac{h^2}{4Pr} (p_{i+1,j}^{(n)} - p_{i-1,j}^{(n)}) \right\} e_{i,j+1}^{(n)} \\ & + \left\{ 8 + \frac{2h}{Pr} w_{i,j}^{(n)} + \frac{h^2}{4Pr} (p_{i+1,j}^{(n)} - p_{i-1,j}^{(n)}) \right\} e_{i,j-1}^{(n)} + \left\{ 4 + \frac{h}{Pr} u_{i,j}^{(n)} \right\} e_{i-1,j}^{(n)} \\ & + \left\{ 4 - \frac{h}{Pr} u_{i,j}^{(n)} \right\} e_{i+1,j}^{(n)} - \{ 14 - r_2 \} e_{i,j}^{(n)} - \left\{ 2 - \frac{h}{2Pr} u_{i,j}^{(n)} - \frac{h}{2Pr} w_{i,j}^{(n)} \right\} e_{i+1,j+1}^{(n)} \\ & - \left\{ 2 - \frac{h}{2Pr} u_{i,j}^{(n)} + \frac{h}{2Pr} w_{i,j}^{(n)} \right\} e_{i+1,j-1}^{(n)} - \left\{ 2 + \frac{h}{2Pr} u_{i,j}^{(n)} - \frac{h}{2Pr} w_{i,j}^{(n)} \right\} e_{i-1,j+1}^{(n)} \\ & - \left\{ 2 + \frac{h}{2Pr} u_{i,j}^{(n)} + \frac{h}{2Pr} w_{i,j}^{(n)} \right\} e_{i-1,j-1}^{(n)} - \left\{ 1 + \frac{h}{2Pr} w_{i,j}^{(n)} \right\} e_{i,j-2}^{(n)} \\ & - \left\{ 1 + \frac{h}{2Pr} w_{i,j}^{(n)} \right\} e_{i,j+2}^{(n)} + L_h(\psi_{i,j}^{(n)}) - f_h(T_{i,j}^{(n)}) , \quad (6.26) \\ & i = 1, \dots, N-1, \end{aligned}$$

with homogeneous Dirichlet and Neumann boundary conditions, and where

$$r_2 = h^4/(\lambda \Delta t)$$

The following equations, described by  $i=1, \dots, N-1$  are solved along lines in the z-direction:

$$\begin{aligned}
& \left\{ 1 + \frac{h}{2P_r} w_{i,j}^{(n)} \right\} e_{i,j-2}^{(n+2)} - \left\{ 4 + \frac{h}{P_r} w_{i,j}^{(n)} + \frac{h^2}{4P_r} (p_{i+1,j}^{(n)} - p_{i-1,j}^{(n)}) \right\} e_{i,j-1}^{(n+2)} \\
& + \{6 + r_2\} e_{i,j}^{(n+2)} - \left\{ 4 - \frac{h}{P_r} w_{i,j}^{(n)} - \frac{h^2}{4P_r} (p_{i+1,j}^{(n)} - p_{i-1,j}^{(n)}) \right\} e_{i,j+1}^{(n+2)} \\
& + \left\{ 1 - \frac{h}{2P_r} w_{i,j}^{(n)} \right\} e_{i,j+2}^{(n+2)} = \left\{ 1 + \frac{h}{2P_r} w_{i,j}^{(n)} \right\} e_{i,j-2}^{(n)} \\
& - \left\{ 4 + \frac{h}{P_r} w_{i,j}^{(n)} + \frac{h^2}{4P_r} (p_{i+1,j}^{(n)} - p_{i-1,j}^{(n)}) \right\} e_{i,j-1}^{(n)} + 6 e_{i,j}^{(n)} + r e_{i,j}^{(n+1)} \\
& - \left\{ 4 - \frac{h}{P_r} w_{i,j}^{(n)} - \frac{h^2}{4P_r} (p_{i+1,j}^{(n)} - p_{i-1,j}^{(n)}) \right\} e_{i,j+1}^{(n)} + \left\{ 1 - \frac{h}{2P_r} w_{i,j}^{(n)} \right\} e_{i,j+2}^{(n)}, \\
& j = 1, \dots, N-1. \tag{6.27}
\end{aligned}$$

In equations (6.26) and (6.27) we have used the notation

$$p_{i,j}^{(n)} = \frac{\delta^2 \psi_{i,j}^{(n)} + \delta^2 \psi_{i,j}^{(n)}}{h^2} \tag{6.28}$$

The coefficient matrices of the systems of equations given by (6.26) for  $j=1, \dots, N-1$ , and (6.27) for  $i=1, \dots, N-1$ , are  $\overset{n}{\underset{\Lambda}{\text{quidiagonal}}}$  and so the algorithm described in Appendix I can be used to solve them. We define an A.D.I. step to be the process by which we obtain  $\underline{T}^{(n+2)}$  and  $\underline{e}^{(n+2)}$  from  $\underline{T}^{(n)}$  and  $\underline{e}^{(n)}$  i.e. the solution of equations (6.20) to (6.23) (6.26) and (6.27). Starting with the approximations  $\underline{T}^{(n+2)}$  and  $\underline{e}^{(n+2)}$  we perform a second A.D.I. step using the same time step  $\Delta t$  to obtain  $\underline{T}^{(n+4)}$  and  $\underline{e}^{(n+4)}$ . We now carry out the bookkeeping stage of the D.A.D.I. step. Starting from  $\underline{T}^{(n)}$  and  $\underline{e}^{(n)}$  we perform an A.D.I. step with time step  $2\Delta t$  to obtain  $\tilde{\underline{T}}^{(n+4)}$  and  $\tilde{\underline{e}}^{(n+4)}$ . We compute the test parameter, TP, which is defined in earlier chapters. If  $TP > 0.6$  then we reject the present step and start the step again with  $\Delta t$  reduced by a factor of  $1/16$ . If  $TP \leq 0.6$  we accept the present D.A.D.I. step and change  $\Delta t$  according to a strategy which is described in the algorithm which follows. We use one step of Newton's method to obtain a new approximation,  $\psi_{i,j}^{(n+4)}$ , to  $\psi_{i,j}$  given by

$$\psi_{i,j}^{(n+4)} = \psi_{i,j}^{(n)} - e_{i,j}^{(n+4)},$$

$i = 1, \dots, N-1, j = 1, \dots, N-1$ . We then use these new values of  $\psi$  to update the linearization before starting the next D.A.D.I. step.

#### Algorithm 6.5.1

The following algorithm, which is flowcharted in Figure 6.2, is used to solve the 'double glazing' problem.

- (a) Set  $n=0$  where  $n$  is the current time level. Choose an initial time step  $\Delta t_0$  and set  $r_1 = h / \Delta t_0$ ,  $r_2 = h^4 / (\lambda \Delta t_0)$ . Choose initial approximations  $\underline{T}^{(0)}$  and  $\underline{\psi}^{(0)}$ . Choose  $\epsilon$  to be a suitable tolerance.
- (b) Begin a step of the D.A.D.I. method. The current approximations to  $T$  and  $\psi$  are  $\underline{T}^{(n)}$  and  $\underline{\psi}^{(n)}$  respectively. These values are used to form  $\underline{u}^{(n)}$ ,  $\underline{w}^{(n)}$ ,  $\underline{p}^{(n)}$ ,  $\underline{L}_h(\underline{\psi}^{(n)})$  and  $\underline{f}_h(\underline{T}^{(n)})$  which are required in the coefficients and right-hand sides of the A.D.I. equations. Set  $\underline{e}^{(n)} = 0$ .
- (c) Perform two A.D.I. steps with time step  $\Delta t$  to obtain the approximations  $\underline{T}^{(n+4)}$  and  $\underline{e}^{(n+4)}$  at the  $(n+4)$ th time step. Compute  $d$  which is the maximum modulus of the difference between successive iterates on even time steps i.e.

$$d = \max \left\{ \begin{aligned} &\| \underline{T}^{(n+2)} - \underline{T}^{(n)} \|_{\infty}, \quad \| \underline{e}^{(n+2)} - \underline{e}^{(n)} \|_{\infty}, \\ &\| \underline{T}^{(n+4)} - \underline{T}^{(n+2)} \|_{\infty}, \quad \| \underline{e}^{(n+4)} - \underline{e}^{(n+2)} \|_{\infty} \end{aligned} \right\}$$

- (d) If  $d < \epsilon$  i.e. we have attained the desired tolerance, then the algorithm is terminated. If not, go to Step (e).
- (e) Carry out the bookkeeping part of the D.A.D.I. step. Starting with  $\underline{T}^{(n)}$  and  $\underline{e}^{(n)}$  perform an A.D.I. step with time step  $\Delta t^* = 2\Delta t$  and set

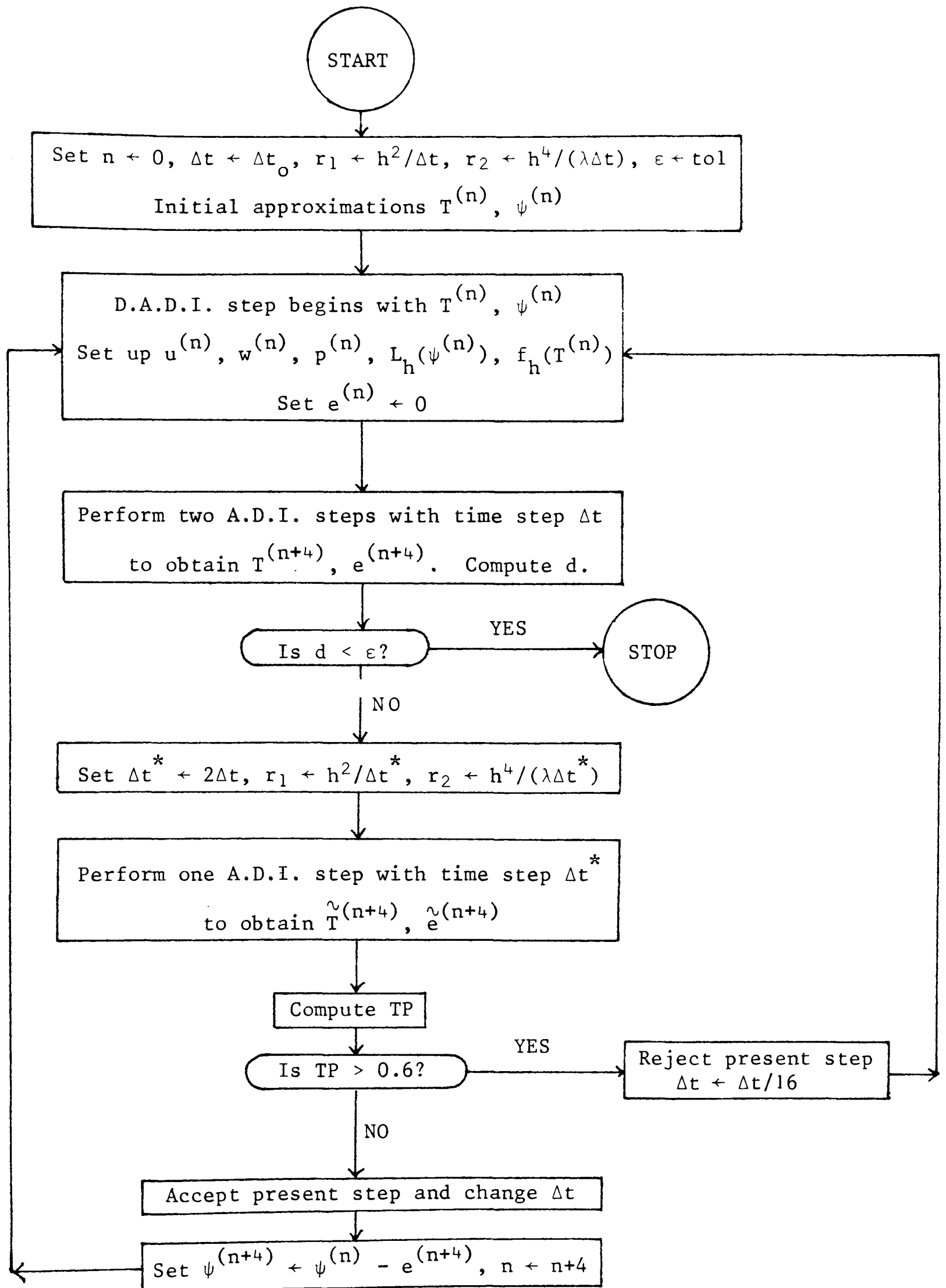


Figure 6.2

$r_1 = h^2/\Delta t^*$ ,  $r_2 = h^4/(\lambda \Delta t^*)$ . Let the resulting approximations be  $\underline{\tilde{T}}^{(n+4)}$  and  $\underline{\tilde{e}}^{(n+4)}$

(f) Compute the test parameter, TP, given by

$$TP = \sqrt{[SUM/ASUM]},$$

where

$$SUM = \|\underline{T}^{(n+4)} - \underline{\tilde{T}}^{(n+4)}\|_2^2 + \|\underline{e}^{(n+4)} - \underline{\tilde{e}}^{(n+4)}\|_2^2$$

and

$$ASUM = \|\underline{T}^{(n+4)} - \underline{T}^{(n)}\|_2^2 + \|\underline{e}^{(n+4)} - \underline{e}^{(n)}\|_2^2$$

If  $TP > 0.6$  then we reject the present D.A.D.I. Step, replace  $\Delta t$  by  $\Delta t/16$  and go to step (b). This is an indication that instabilities are beginning to occur and so we decrease  $\Delta t$  to bypass them. If  $TP \leq 0.6$  then we accept the present D.A.D.I. step and change  $\Delta t$  according to the strategy: if TP falls in the intervals  $[0, 0.05]$ ,  $(0.05, 0.1]$ ,  $(0.1, 0.3]$ ,  $(0.3, 0.4]$ ,  $(0.4, 0.6]$  change  $\Delta t$  by the factors 4, 2,  $\sqrt{3}$ ,  $1/2$ ,  $1/4$  respectively for the next D.A.D.I. step. Go to Step (g). The strategies that we have used in earlier chapters to solve Poisson's equation, the coupled system of equations which comprise the Philips problem and the biharmonic equation led us to this choice of strategy.

(g) Use one step of Newton's method to update the approximation to  $\psi$  by setting

$$\underline{\psi}^{(n+4)} = \underline{\psi}^{(n)} - \underline{e}^{(n+4)}$$

Replace  $n$  by  $n+4$  and go to Step (b).

## 6.6 Numerical Results

The initial approximations  $\underline{T}^{(0)}$  and  $\underline{\psi}^{(0)}$  are chosen to be the values at the grid points of the functions  $T_I$  and  $\psi_I$  respectively, where

$$T_I(x, y) = 1 - x, \quad \psi_I(x, y) = 0.$$

Algorithm 6.5.1. was run for different grid sizes and for Rayleigh numbers of  $10^3$ ,  $10^4$ ,  $10^5$  and  $10^6$ . The quidiagonal systems are solved using the algorithm given in Appendix I. A similar algorithm is used for the tridiagonal systems (6.20) to (6.23). The stopping criterion is defined at Step (c) of Algorithm 6.5.1. The tolerance,  $\epsilon$ , was chosen to be  $10^{-6}$  for all grid sizes except  $h=1/64$ , and  $5 \times 10^{-5}$  for the grid size  $h=1/64$ .

We experimented with different values of the parameter  $\lambda$ . For  $\lambda=1$  we found that for large values of the Rayleigh number the iterates oscillated due to the interaction between the equations. For suitable choices of this parameter the instabilities were controlled and the method converged. The value of  $\lambda$  was chosen to be 0.05, 0.01, 0.002 and 0.002 respectively for the Rayleigh numbers  $10^3$ ,  $10^4$ ,  $10^5$  and  $10^6$ . The choice of  $\lambda$  was not critical in the sense that similar convergence behaviour was observed for a wide range of values in the neighbourhood of the chosen  $\lambda$ . In the D.A.D.I. method of solution of the Philips problem in Chapter 3 a similar choice of  $\lambda$  gave a big improvement in convergence.

Suppose that for a particular value of the Rayleigh number the problem has been solved numerically on a grid of mesh size  $h$ . To find the solution on a grid of mesh size  $1/2h$  values interpolated from those obtained on the coarser grid are used as our initial approximation. Cubic interpolation is used for values of  $\psi$  and linear interpolation for values of  $T$ . It must be noted that we are not using a multigrid method but simply using the best available information to start each calculation.

The average value of the Nusselt number is calculated using the trapezoidal rule, where we use the following approximation to the normal derivative of  $T$  on the hot wall:

$$\frac{\partial T(0,z)}{\partial n} \simeq \frac{-3T(0,z) + 4T(h,z) - T(2h,z)}{2h}.$$

Table 6.1 contains the average Nusselt number on the hot wall and the maximum and minimum local Nusselt numbers on the hot wall, and their location. Table 6.2 contains the maximum vertical velocity on the horizontal mid-plane and its location, the maximum horizontal velocity on the vertical mid-plane and its location, and the maximum absolute value of the stream function and its location. Table 6.3 contains the number of D.A.D.I. steps and the run time, in seconds, required to reach the convergence criterion. Contours for the temperature and stream function are shown in Figures 6.3 to 6.10 for different values of the Rayleigh number.

The results we have obtained mainly show a good agreement and the convergence is approximately quadratic. Also it is clear that some form of non-uniform grid is needed to resolve the boundary layers accurately when  $Ra$  is large and this might even have the effect of improving the rate of D.A.D.I. convergence. This is investigated in the following section.

Thirty contributions to this comparison problem are contained in a report edited by Jones and Thompson (1981). The results presented in Tables 6.1 and 6.2 agree well with those deemed to be the best in this report apart from the value of  $\overline{Nu}$  at  $Ra = 10^6$  which is a little high. It is difficult to draw a comparison between the methods contained in this report and the method described in this chapter due to non-conformity of computers, stopping criterion, grid size, etc. However, the D.A.D.I. method seems a perfectly feasible method of solving this problem and could well be made even more competitive by 'fine tuning'. In solving a biharmonic equation for the stream function, by eliminating vorticity, this method has the attraction of avoiding the need for a vorticity boundary condition.



Ra	Mesh	Average Nusselt number	Maximum Nusselt number		Minimum Nusselt number	
		$\overline{Nu}$	$Nu_{max}$	@ z =	$Nu_{min}$	@ z =
$10^3$	17 x 17	1.120	1.515	0.86	0.697	0.0
	33 x 33	1.118	1.508	0.91	0.693	0.0
	65 x 65	1.118	1.507	0.91	0.692	0.0
$10^4$	17 x 17	2.357	3.838	0.81	0.615	0.0
	33 x 33	2.270	3.628	0.84	0.592	0.0
	65 x 65	2.250	3.554	0.86	0.587	0.0
$10^5$	17 x 17	5.146	8.423	0.88	0.854	0.0
	33 x 33	4.758	8.508	0.91	0.755	0.0
	65 x 65	4.573	7.972	0.92	0.735	0.0
$10^6$	33 x 33	10.062	19.085	0.91	0.926	0.03
	65 x 65	9.272	19.590	0.95	0.999	0.0

Table 6.1 Heat Transfer Results

Ra	Mesh	Vertical velocity on horizontal mid-plane		Horizontal velocity on vertical mid-plane		Maximum Stream function		
		w <sub>max</sub>	@ x =	u <sub>max</sub>	@ z =	ψ <sub>max</sub>	@ x =	@ z =
10 <sup>3</sup>	17 x 17	3.692	0.19	3.657	0.19	1.219	0.5	0.5
	33 x 33	3.692	0.81	3.651	0.81	1.186	0.5	0.5
	65 x 65	3.694	0.17	3.649	0.19	1.177	0.5	0.5
10 <sup>4</sup>	17 x 17	19.718	0.88	16.621	0.19	5.427	0.5	0.5
	33 x 33	19.623	0.88	16.269	0.81	5.163	0.5	0.5
	65 x 65	19.602	0.88	16.206	0.83	5.096	0.5	0.5
10 <sup>5</sup>	17 x 17	69.306	0.94	40.558	0.13	11.725	0.69	0.63
	33 x 33	68.680	0.06	36.155	0.16	10.129	0.72	0.59
	65 x 65	68.553	0.06	35.097	0.86	9.742	0.72	0.61
10 <sup>6</sup>	33 x 33	218.711	0.03	75.595	0.84	20.020	0.16	0.47
	65 x 65	216.657	0.97	67.179	0.16	17.561	0.84	0.55

Table 6.2 Velocities and Stream Function Values

Ra	Mesh	Number of D.A.D.I. steps	Time
$10^3$	17 x 17	24	4.9
	33 x 33	19	14.9
	65 x 65	9	30.3
$10^4$	17 x 17	36	6.7
	33 x 33	36	26.9
	65 x 65	21	66.0
$10^5$	17 x 17	111	21.1
	33 x 33	92	68.9
	65 x 65	49	147.5
$10^6$	33 x 33	362	268.5
	65 x 65	335	1012.0

Table 6.3 Computational Results

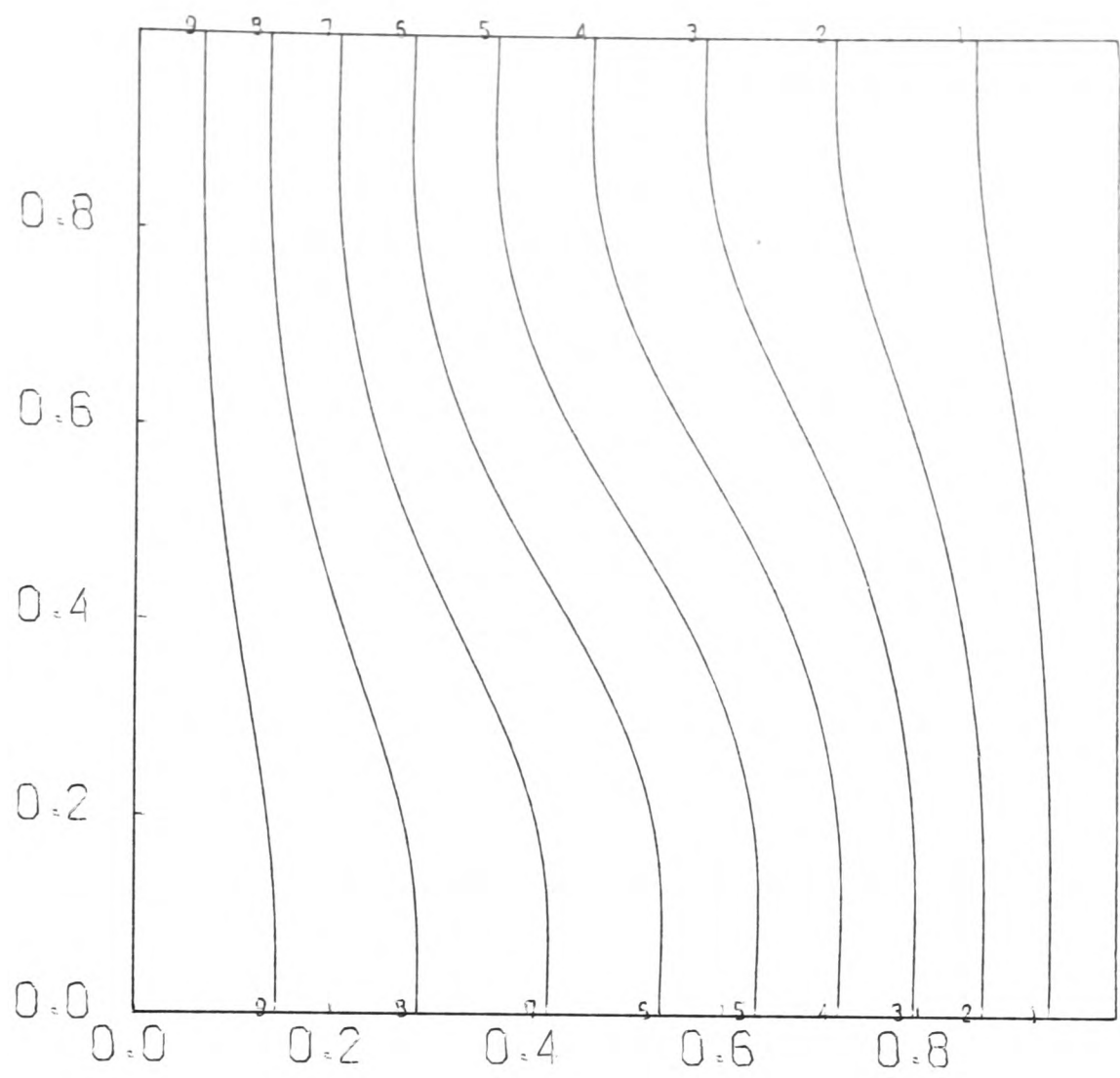


Figure 6.3. Temperature contours for  $Ra = 10^3$ .  
Contour levels are  $0.1(0.1)0.9$ .

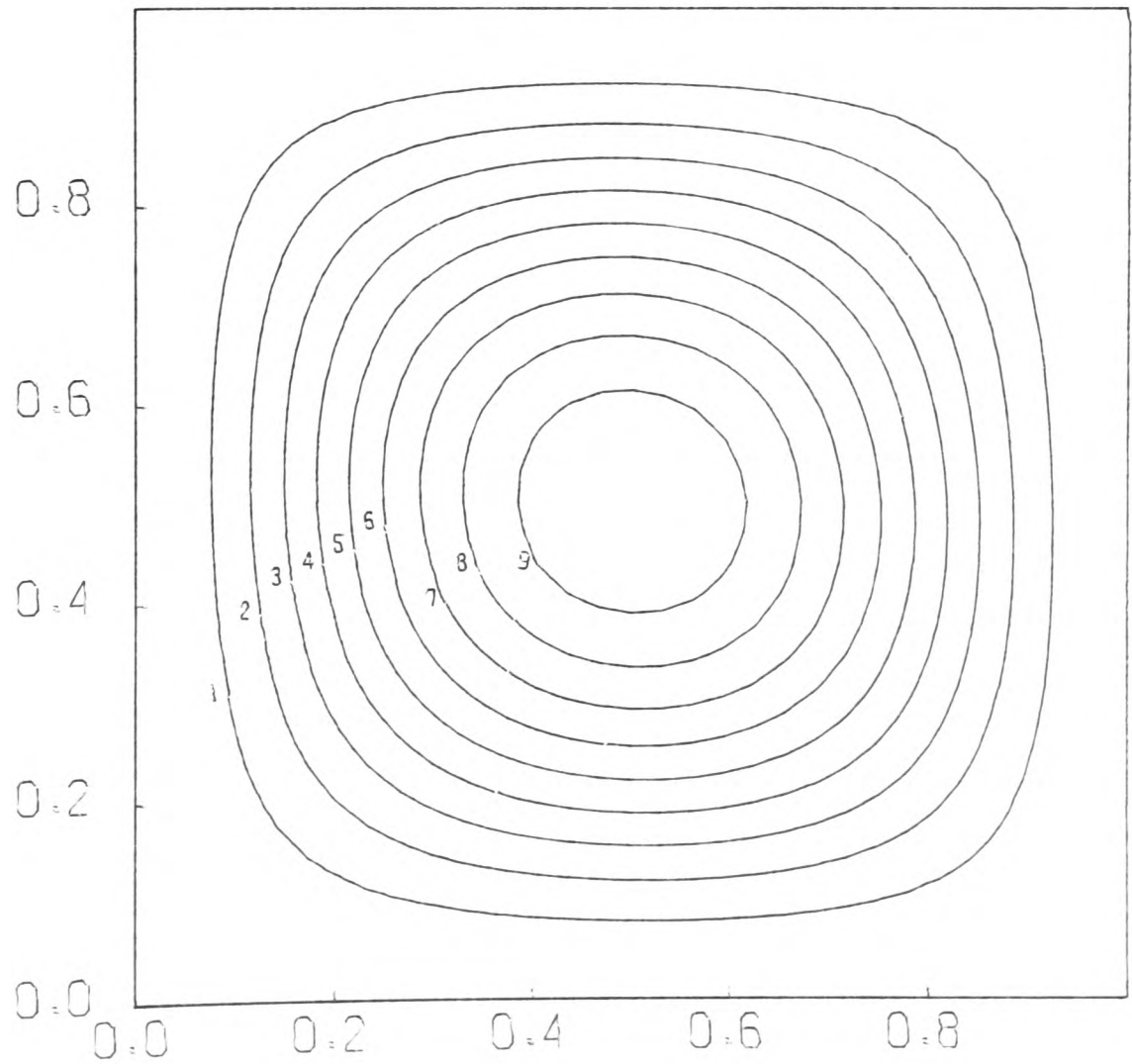


Figure 6.4. Stream function contours for  $Ra = 10^3$ .  
Contour levels are  $-0.12(-0.12)-1.08$ .

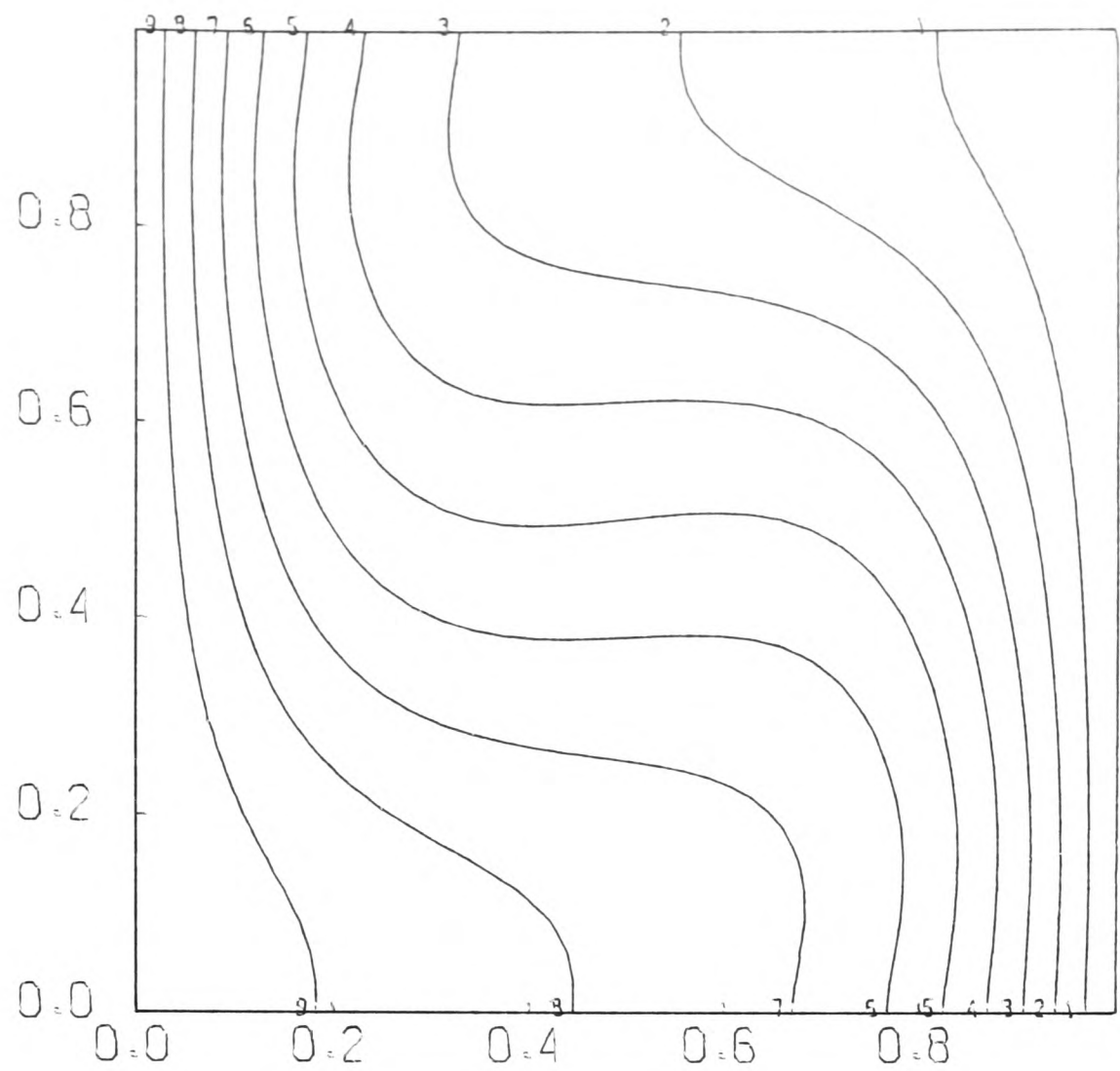


Figure 6.5. Temperature contours for  $Ra = 10^4$ .  
Contour levels are  $0.1(0.1)0.9$ .

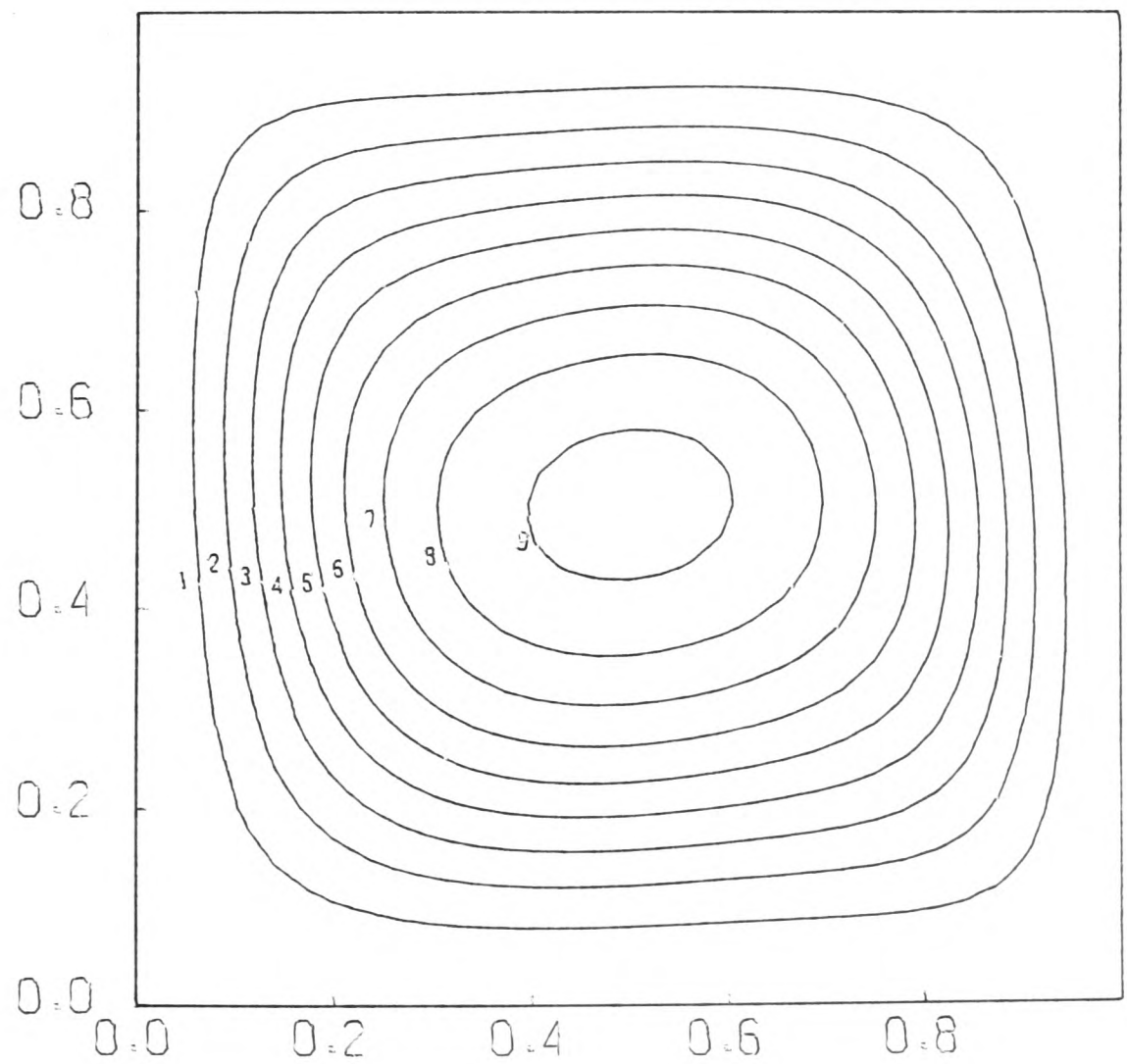


Figure 6.6. Stream function contours for  $Ra = 10^4$ .  
Contour levels are  $-0.552(-0.552)-4.968$

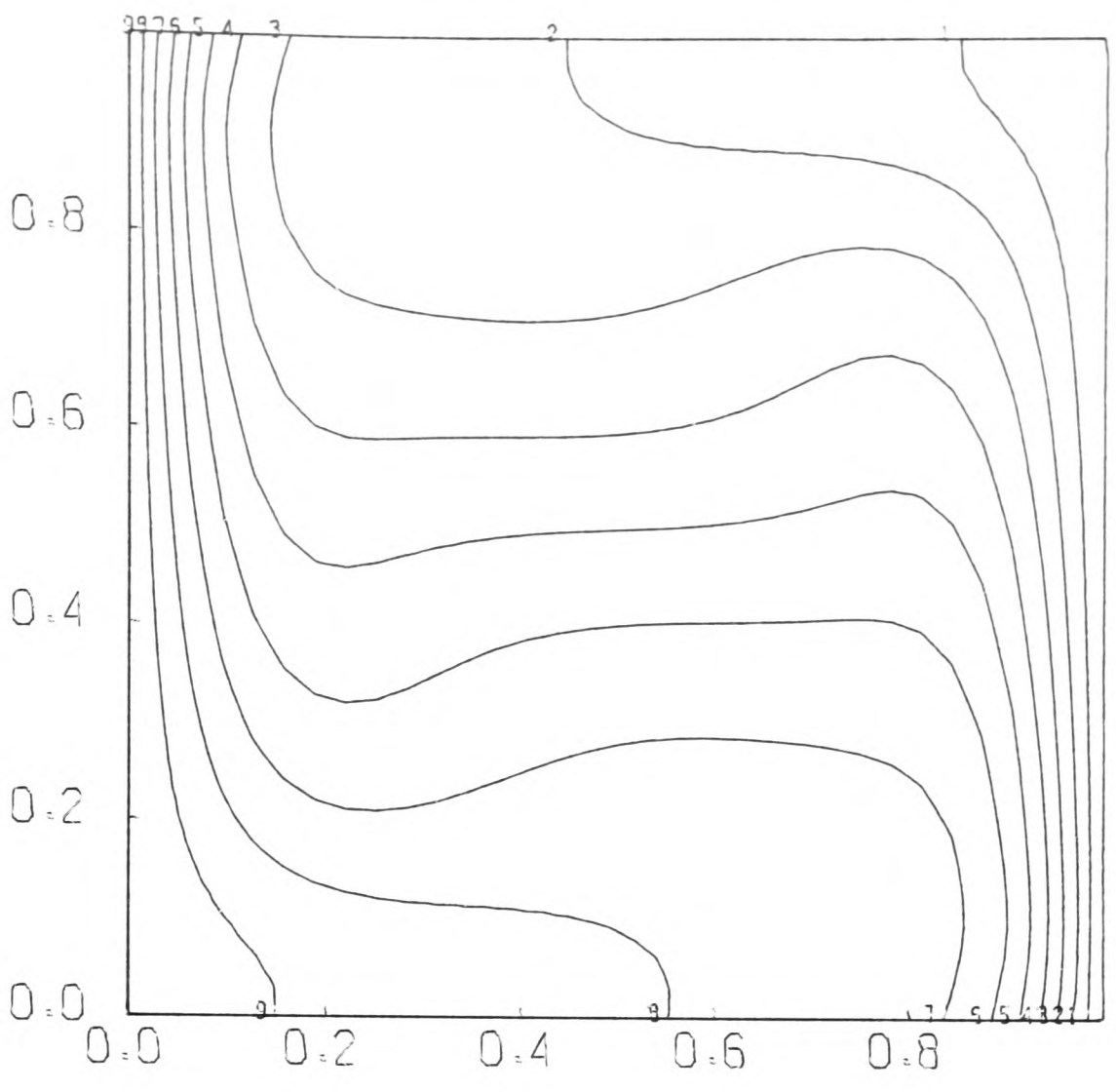


Figure 6.7. Temperature contours for  $Ra = 10^5$ .  
Contour levels are  $0.1(0.1)0.9$ .

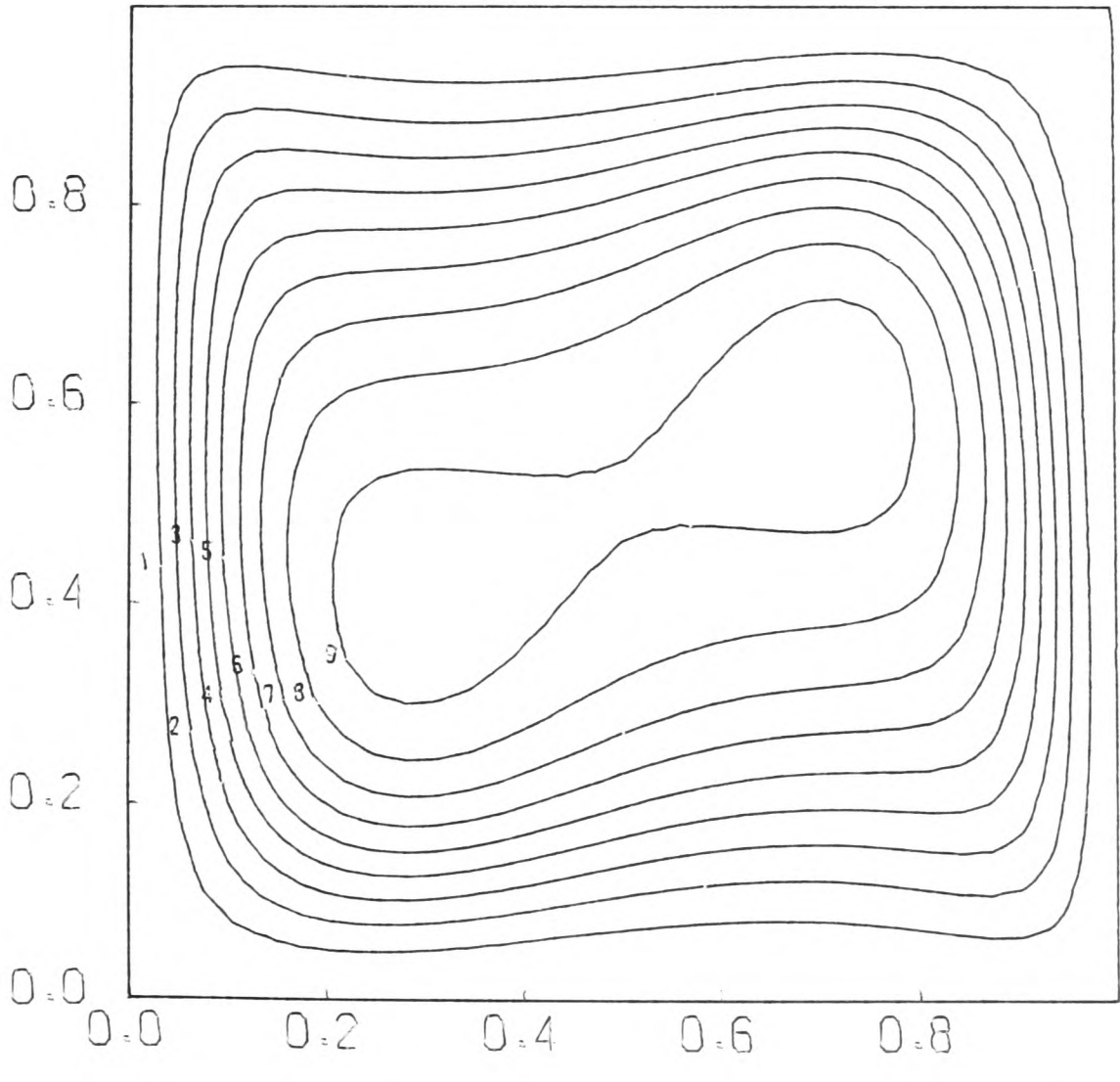


Figure 6.8 Stream function contours for  $Ra = 10^5$ .  
Contour levels are  $-1.06(-1.06)-9.54$ .

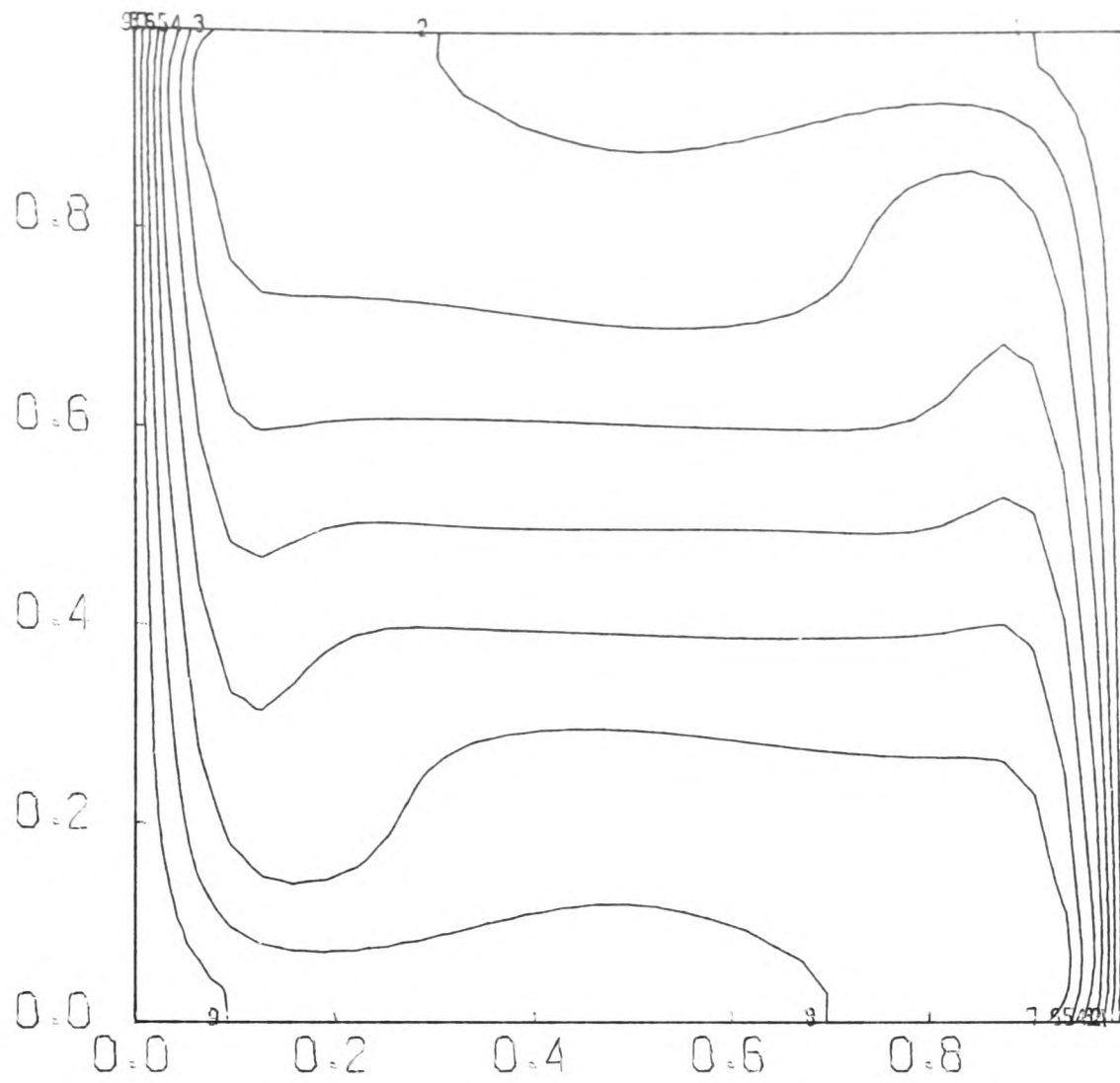


Figure 6.9. Temperature contours for  $Ra = 10^6$ .  
Contour levels are  $0.1(0.1)0.9$ .

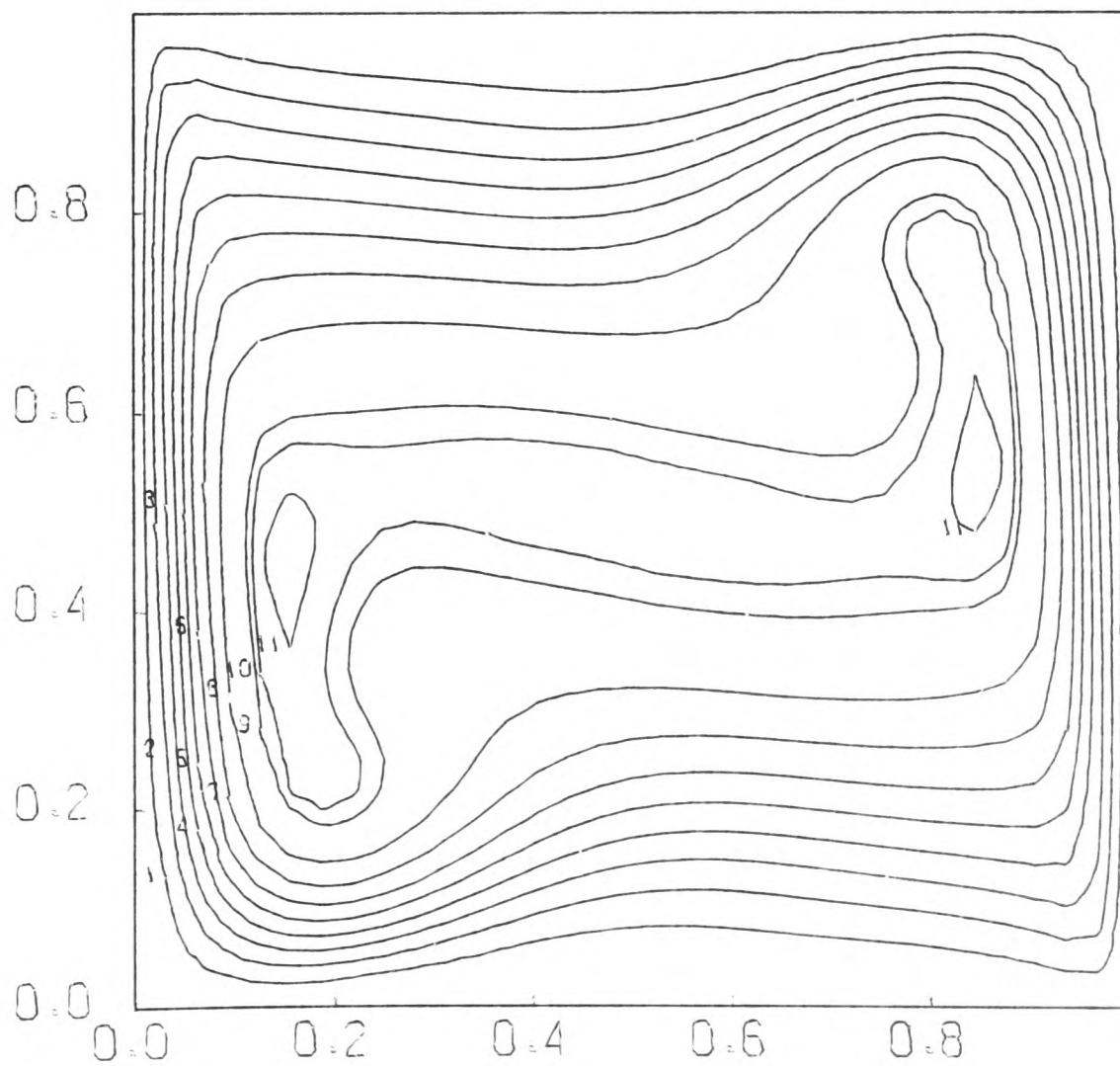


Figure 6.10. Stream function contours for  $Ra = 10^6$ .  
Contour levels are  $-2.1(-2.1)-18.9, -19.3, -19.8$ .

### 6.7 Non-uniform grids

For large values of the Rayleigh number the flows develop strong boundary layers. This effect can be seen in Figure 6.10, for example. When  $Ra = 10^6$  we have had to use an extremely fine mesh,  $65 \times 65$ , in order to resolve these boundary layers. This is rather wasteful since the points are also densely distributed away from these layers where they are not needed. Here we apply the technique of Kálmán de Rivas (1972), which was described in Section 4.3, to resolve the boundary layers using a non-uniform grid.

Basically, the idea is to make a change of independent variable so that the domain is mapped into a new co-ordinate system where the variations of the solution are not so rapid. The grid intervals are varied by defining a stretched co-ordinate  $\xi$  such that  $x = x(\xi)$  where the grid intervals  $\Delta\xi$  are constant and  $x$  is the old physical co-ordinate. The mapping is chosen so that the solution, when regarded as a function of the new variable  $\xi$ , has no boundary layers.

In Chapter 4 we expressed the first and second derivatives in terms of  $\xi$ . In a similar way we can express the fourth derivative in terms of  $\xi$  in the following manner

$$\frac{\partial^4 v}{\partial x^4} = \frac{\partial}{\partial \xi} \left( \frac{\partial}{\partial \xi} \left( \frac{\partial}{\partial \xi} \left( \frac{\partial v}{\partial \xi} \cdot \frac{d\xi}{dx} \right) \frac{d\xi}{dx} \right) \frac{d\xi}{dx} \right) \frac{d\xi}{dx}. \quad (6.29)$$

Equation (6.29) can be discretized using central differences to give

$$\begin{aligned} \frac{\partial^4 v}{\partial x^4} \simeq & \frac{4}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)(x_{i+2} - x_i)} \left[ \frac{(v_{i+2} - v_{i+1})}{(x_{i+2} - x_{i+1})} - \frac{(v_{i+1} - v_i)}{(x_{i+1} - x_i)} \right] \\ & - \frac{4}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)(x_i - x_{i-1})} \left[ \frac{(v_{i+1} - v_i)}{(x_{i+1} - x_i)} - \frac{(v_i - v_{i-1})}{(x_i - x_{i-1})} \right] \\ & + \frac{4}{(x_{i+1} - x_{i-1})(x_i - x_{i-1})(x_i - x_{i-2})} \left[ \frac{(v_i - v_{i-1})}{(x_i - x_{i-1})} - \frac{(v_{i-1} - v_{i-2})}{(x_{i-1} - x_{i-2})} \right], \end{aligned} \quad (6.30)$$



where  $v_i$  is the value of  $v$  at the  $i$ th grid point and  $x_i = x(i\Delta\xi)$ . In deriving the approximation (6.30) we have differentiated the transformation numerically using central differences. Finite difference approximations for other high order derivatives are obtained in a similar way.

Let  $x(\xi)$  and  $z(\eta)$  be two grid stretching functions with constant grid intervals  $\Delta\xi$  and  $\Delta\eta$  respectively. These functions will be defined later. The region  $\Omega = \{ (x,z) : 0 \leq x \leq 1, 0 \leq z \leq 1 \}$  is covered with a variable grid defined by the above mappings. We define  $T_{i,j}$  and  $\psi_{i,j}$  to be the values of  $T(x,z)$  and  $\psi(x,z)$  respectively at the grid point  $(x_i, z_j)$  where  $x_i = x(i\Delta\xi)$  and  $z_j = z(j\Delta\eta)$ . The finite difference equations are constructed using approximations like (6.30) for the derivatives. The system of finite difference equations is solved using the D.A.D.I. method by incorporating the mesh stretching technique into Algorithm 6.5.1. This means that if we let the grid stretching functions define a uniform grid then we obtain the same equations and hence the same results as when we apply Algorithm 6.5.1. with the same constant grid interval.

The problem is solved on the following non-uniform grids:

$$\begin{aligned} \text{(a)} \quad x(\xi) &= \sin^2(1/2\pi\xi) , \\ z(\eta) &= \sin^2(1/2\pi\eta) . \end{aligned} \tag{6.31}$$

$$\begin{aligned} \text{(b)} \quad x(\xi) &= 6\xi^5 - 15\xi^4 + 10\xi^3 , \\ z(\eta) &= 6\eta^5 - 15\eta^4 + 10\eta^3 . \end{aligned} \tag{6.32}$$

These grid stretching functions give a smaller spacing of the grid points near the boundaries. The function  $f(x) = 6x^5 - 15x^4 + 10x^3, x \in [0,1]$  , was constructed subject to the following conditions:

(i)  $f'(0) = 0$  and  $f'(1) = 0$ . These conditions ensure a concentration of grid points near  $x = 0$  and  $x = 1$ .

(ii)  $f''(0)$  and  $f''(1) = 0$ . In the boundary layers the derivatives of the solution are large and so to keep the local truncation error small as well as

formally second order we need the derivatives of  $f$  to be small there (see Jones and Thompson (1980)).

(iii)  $f(0) = 0$  and  $f(1) = 1$ .

(iv) A symmetric distribution of points about  $x = 1/2$ .

These conditions define a suitable non-uniform grid for this problem.

The non-uniform grid defined by (6.32) is shown in Figure 6.11 for

$\Delta\xi = \Delta\eta = 0.04$ .

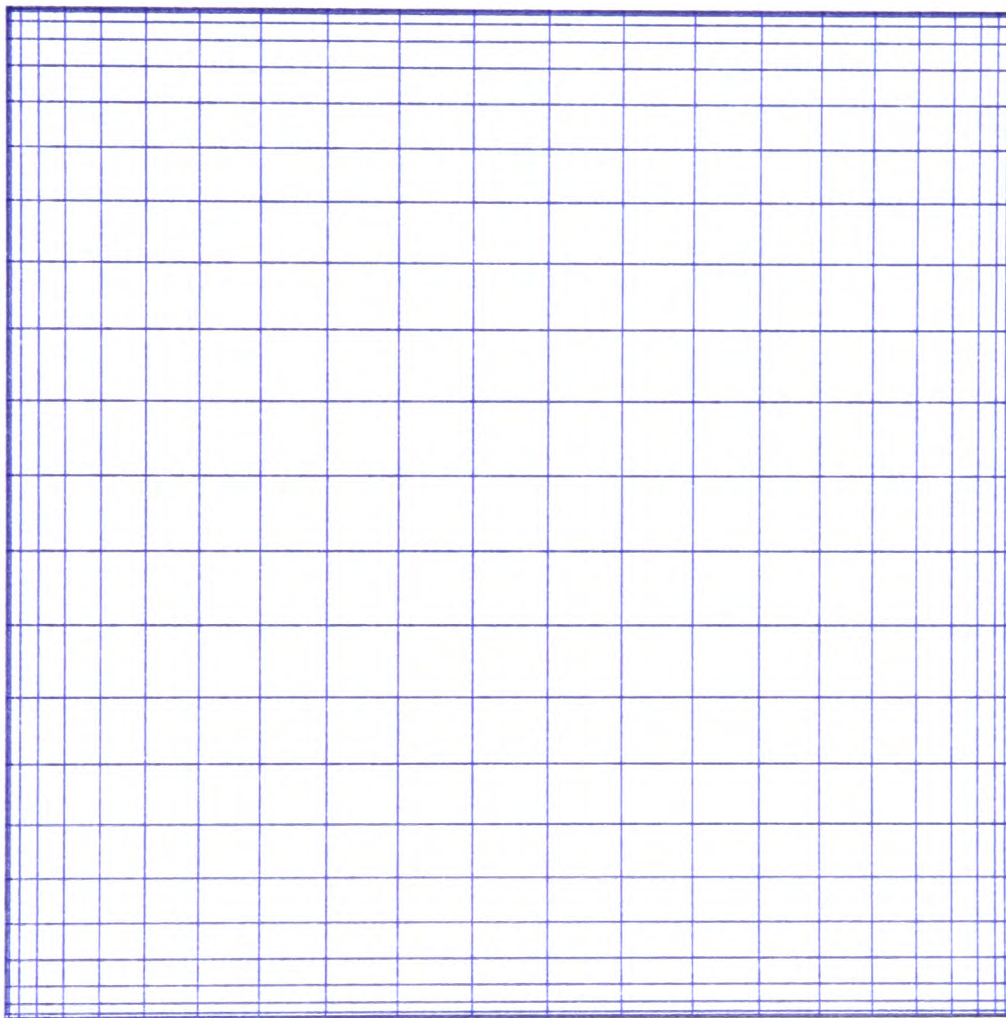


Figure 6.11. Non-uniform grid (b)

The problem was solved for Rayleigh numbers of  $10^5$  and  $10^6$  on the  $26 \times 26$  non-uniform grids defined by (6.31) and (6.32) using a modification of Algorithm 6.5.1. for non-uniform grids. One of the quantities of major interest is the average value of the Nusselt number along the hot wall. In Table 6.4 we compare values of this quantity obtained on

a uniform  $26 \times 26$  mesh, the stretched  $26 \times 26$  meshes defined by (6.31) and (6.32), and a uniform  $65 \times 65$  mesh. In Tables 6.5 and 6.6 we compare the number of D.A.D.I. steps and run time respectively to reach the convergence criterion, which is the same as that used for the results in Section 6.6.

In Table 6.4 we see that we have obtained a good estimate of the Nusselt number by using a stretched grid even though we have only a  $26 \times 26$  mesh. For small values of the Rayleigh number the use of a stretched grid has little effect. It is only when the Rayleigh number becomes large that we obtain an improvement by using non-uniform grids. We can see from Table 6.6 that solving this problem for  $Ra = 10^6$  on a uniform  $65 \times 65$  mesh is very time consuming. The use of non-uniform grids to resolve the boundary layers has had the effect of improving the rate of D.A.D.I. convergence. Approximately twice as many D.A.D.I. steps were required for convergence of the method using the uniform  $26 \times 26$  mesh than for the stretched  $26 \times 26$  meshes.

The use of non-uniform grids has allowed us to resolve the boundary layers using comparatively few mesh points. Reasonable accuracy has also been obtained on the stretched grids compared with an extremely fine mesh.

The D.A.D.I. method for solving this problem on a non-uniform grid seems to have worked well. Finally, we refer back to Chapter 4 where we commented that an advantage of the D.A.D.I. method over standard iterative methods of solving finite difference equations on a non-uniform grid is that we <sup>do</sup> not require an a priori choice of parameters to accelerate convergence.

Ra	Uniform mesh	Stretched mesh	Stretched mesh	Uniform mesh
	26 x 26	(a) 26 x 26	(b) 26 x 26	65 x 65
10 <sup>5</sup>	4.889	4.596	4.595	4.573
10 <sup>6</sup>	10.163	9.123	9.066	9.272

Table 6.4 Average value of the Nusselt number

Ra	Uniform mesh	Stretched mesh	Stretched mesh	Uniform mesh
	26 x 26	(a) 26 x 26	(b) 26 x 26	65 x 65
10 <sup>5</sup>	151	85	84	80
10 <sup>6</sup>	676	334	369	426

Table 6.5 Number of D.A.D.I. steps

Ra	Uniform mesh	Stretched mesh	Stretched mesh	Uniform mesh
	26 x 26	(a) 26 x 26	(b) 26 x 26	65 x 65
10 <sup>5</sup>	68	85	84	238
10 <sup>6</sup>	299	332	364	1280

Table 6.6 Computational time (seconds)

## APPENDIX I

Algorithm for solving a quidiagonal system

The following quidiagonal system is typical of those arising from biharmonic problems:

$$\begin{bmatrix}
 C_1 & D_1 & E_1 & & & \\
 B_2 & C_2 & D_2 & E_2 & & \\
 A_3 & B_3 & C_3 & D_3 & E_3 & \\
 & & \cdot & \cdot & \cdot & \cdot \\
 & & & \cdot & \cdot & \cdot \\
 & & & & A_{N-2} & B_{N-2} & C_{N-2} & D_{N-2} \\
 & & & & & A_{N-1} & B_{N-1} & C_{N-1}
 \end{bmatrix}
 \begin{bmatrix}
 u_1 \\
 u_2 \\
 u_3 \\
 \cdot \\
 \cdot \\
 u_{N-2} \\
 u_{N-1}
 \end{bmatrix}
 =
 \begin{bmatrix}
 F_1 \\
 F_2 \\
 F_3 \\
 \cdot \\
 \cdot \\
 F_{N-2} \\
 F_{N-1}
 \end{bmatrix}
 \quad (I.1)$$

This system can be written in the matrix form

$$S \underline{u} = \underline{F}. \quad (I.2)$$

The following Gaussian elimination scheme may be used to solve the above system. Let

$$\begin{aligned}
 \alpha_1 &= C_1 \\
 \beta_1 &= D_1/\alpha_1, \quad \beta_0 = 0, \quad \beta_{N-1} = 0 \\
 \gamma_1 &= E_1/\alpha_1, \quad \gamma_0 = 0, \quad \gamma_{N-1} = 0, \quad \gamma_{N-2} = 0
 \end{aligned}$$

For  $j = 2, \dots, N-1$  define recursively

$$\begin{aligned}
 \delta_j &= B_j - A_j \beta_{j-2}, \\
 \alpha_j &= C_j - A_j \gamma_{j-2} - \delta_j \beta_{j-1}, \\
 \beta_j &= (D_j - \delta_j \gamma_{j-1}) / \alpha_j, \\
 \gamma_j &= E_j / \alpha_j.
 \end{aligned}$$

Next we construct the corresponding new right-hand sides:

$$\begin{aligned}
 p_0 &= 0, \\
 p_1 &= F_1 / \alpha_1, \\
 p_j &= (F_j - A_j p_{j-2} - \delta_j p_{j-1}) / \alpha_j, \quad j = 2, \dots, N-1.
 \end{aligned}$$

## Appendix I....

The solution of (I.1) is then given by the following formulae

$$u_{N-1} = p_{N-1} ,$$

$$u_j = p_j - \beta_j u_{j+1} - \gamma_j u_{j+2} , \quad j = N-2, N-3, \dots, 1.$$

The stability of this algorithm, which does not use interchanges, is guaranteed if the matrix  $S$  in (I.2) is symmetric and positive definite.

## APPENDIX II

Let  $L : A \rightarrow B$  where  $A$  and  $B$  are real normed vector spaces.

Definition

If  $L(\psi + \phi) - L(\psi) = M(\phi) + E(\phi)$ , where  $M$  is a bounded linear operator,  $M : A \rightarrow B$  and  $\|E(\phi)\| / \|\phi\| \rightarrow 0$  as  $\|\phi\| \rightarrow 0$  then the function  $L$  is said to be Fréchet differentiable at  $\psi$  and we write  $L'(\psi) = M$ .

We define

$$C^\infty(\Omega) = \{ \phi : \Omega \rightarrow \mathbb{R} \mid \phi \text{ and all of its partial derivatives exist and are continuous} \}.$$

This space is a normed vector space with norm defined by

$$\|\phi\| = \max_{i,j} \left\{ \sup_{\Omega} \left| \frac{\partial^{i+j} \phi}{\partial x^i \partial z^j} \right| \right\},$$

where  $i$  and  $j$  are non-negative integers.

If  $L$  is defined by (6.8) and (6.12) then

$$L(\psi + \phi) - L(\psi) = M(\phi) + E(\phi),$$

where

$$\begin{aligned} M(\phi) &= \nabla^4 \phi + \frac{1}{Pr} \frac{\partial \psi}{\partial z} \frac{\partial}{\partial x} (\nabla^2 \phi) + \frac{1}{Pr} \left( \frac{\partial}{\partial x} (\nabla^2 \psi) \right) \frac{\partial \phi}{\partial z} \\ &- \frac{1}{Pr} \frac{\partial \psi}{\partial x} \frac{\partial}{\partial z} (\nabla^2 \phi) - \frac{1}{Pr} \left( \frac{\partial}{\partial z} (\nabla^2 \psi) \right) \frac{\partial \phi}{\partial x}, \end{aligned} \quad (II.1)$$

and

$$E(\phi) = \frac{1}{Pr} \frac{\partial \phi}{\partial z} \frac{\partial}{\partial x} (\nabla^2 \phi) - \frac{1}{Pr} \frac{\partial \phi}{\partial x} \frac{\partial}{\partial z} (\nabla^2 \phi). \quad (II.2)$$

Clearly  $M$  is both bounded and linear. Moreover, we have that

$$\|E(\phi)\| \leq \frac{1}{Pr} \|\phi_z\| \|\nabla^2 \phi\|_x + \frac{1}{Pr} \|\phi_x\| \|\nabla^2 \phi\|_z. \quad (II.3)$$

Now we have that

$$\|\phi_z\| = \max_{i,j} \left\{ \sup_{\Omega} \left| \frac{\partial^{i+j+1} \phi}{\partial x^i \partial z^{j+1}} \right| \right\}$$

Appendix II.....

$$\leq \max_{i,j} \left\{ \sup_{\Omega} \left| \frac{\partial^{i+j} \phi}{\partial x^i \partial z^j} \right| \right\}$$

$$= \| \phi \| ,$$

We obtain similar expressions for the other terms in (II.3). Therefore

$$\| E(\phi) \| \leq \frac{4}{Pr} \| \phi \|^2$$

Hence  $\| E(\phi) \| / \| \phi \| \rightarrow 0$  as  $\| \phi \| \rightarrow 0$  and so we deduce that  $L$  is

Fréchet differentiable at  $\psi$  with  $L'(\psi) = M$ .



## REFERENCES

- AITCHISON, J.M. (1980). Numerical modelling of PIN diodes. Submitted to *J. Inst. Maths. Applies*.
- AITCHISON, J.M. and BERZ, F. (1981). The effect of surface recombination on PIN diodes. *Solid-St. Electron.* **24**, 795-804.
- ASTRACHANCEV, G.P. (1971). An iterative method of solving elliptic net problems. *U.S.S.R. Comp. Maths. Math. Phys.* **11**, 2, 171-182.
- BAKHVALOV, N.S. (1966). On the convergence of a relaxation method with natural constraints on the elliptic operator. *U.S.S.R. Comp. Maths. Math. Phys.* **6**, 5, 101-135.
- BRANDT, A. (1973). Multi-level adaptive technique (MLAT) for fast numerical solutions to boundary value problems. Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics (Paris, 1972), *Lect. Notes in Phys.* **18**, 82-89. Berlin and New York: Springer-Verlag.
- BRANDT, A. (1977). Multi-level adaptive solutions to boundary-value problems. *Math. Comp.* **31**, 330-390.
- CONTE, S. and DAMES, R.J. (1958). An alternating direction scheme for the biharmonic difference equations. *Math. Tables Aids Comput.* **12**, 198-205.
- DOSS, S. and MILLER, K. (1979). Dynamic A.D.I. methods for elliptic equations. *SIAM J. Numer. Anal.* **16**, 837-856.
- 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.
- FEDORENKO, R.P. (1962). A relaxation method for solving elliptic difference equations. *U.S.S.R. Comp. Maths. Math. Phys.* **1**, 1092-1096.
- FEDORENKO, R.P. (1964). The speed of convergence of one iterative process. *U.S.S.R. Comp. Maths. Math. Phys.* **4**, 3, 227-235.
- FOERSTER, H. and WITSCH, K. (1981). On efficient multigrid software for elliptic problems on rectangular domains. Preprint no.458, Sonderforschungsbereich 72, University of Bonn, Bonn.
- FOX, L. (1957). The numerical solution of two-point boundary problems in ordinary differential equations. Oxford: Clarendon Press.
- HACKBUSCH, W. (1979). On the fast solution of nonlinear elliptic equations. *Numer. Math.* **32**, 83-95.
- HACKBUSCH, W. (1980). Convergence of multigrid iterations applied to difference equations. *Math. Comp.* **34**, 425-440.
- HELLER, J. (1960). Simultaneous, successive and alternating direction iteration schemes. *J. Soc. Indust. Appl. Math.* **8**, 150-174.

- ISAACSON, E. and KELLER, H.B. (1966). Analysis of numerical methods. New York: John Wiley.
- JONES, I.P. (1979a). *J. Fluid Mech.* 95, 4, inside back cover.
- JONES, I.P. (1979b). A numerical study of natural convection in an air-filled cavity: comparison with experiment. *Num. Heat. Transf.* 2, 193-213.
- JONES, I.P. (1979c). A comparison problem for numerical methods in fluid dynamics: the 'double-glazing' problem. In: Numerical Methods in Thermal Problems. (R.W. Lewis and K. Morgan, Eds.). Swansea: Pineridge Press.
- JONES, I.P. and THOMPSON, C.P. (1980). On the use of non-uniform grids in finite difference calculations. A.E.R.E. Report R.9765. H.M.S.O.
- JONES, I.P. and THOMPSON, C.P. (Eds.) (1981). Numerical solutions for a comparison problem on natural convection in an enclosed cavity. A.E.R.E. Report R. 9950. H.M.S.O.
- KALNAY DE RIVAS, E. (1972). On the use of non-uniform grids in finite difference equations. *J. Comp. Phys.* 10, 202-210.
- MALLINSON, G.D. and DE VAHL DAVIS, G. (1973). The method of the false transient for the solution of coupled elliptic equations. *J. Comp. Phys.* 12, 435-461.
- MALLINSON, G.D. and DE VAHL DAVIS, G. (1977). Three-dimensional natural convection in a box: a numerical study. *J. Fluid Mech.* 83, 1, 1-31.
- PEACEMAN, D.W. and RACHFORD, H.H. (1955). The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.* 3, 28-41.
- SMITH, G.D. (1978). Numerical solution of partial differential equations. Oxford: Clarendon Press.
- SOUTHWELL, R.V. (1935). Stress calculation in frameworks by the method of systematic relaxation of constraints, I, II. *Proc. Roy. Soc. London Ser. A.* 151, 56-95.
- SOUTHWELL, R.V. (1956). Relaxation methods in theoretical physics, Vol.II. Oxford: Clarendon Press.
- VARGA, R.S. (1962). Matrix iterative analysis. New Jersey: Prentice-Hall.
- WACHSPRESS, E.L. (1962). Optimum alternating-direction-implicit iteration parameters for a model problem. *J. Soc. Indust. Appl. Math.* 10, 339-350.
- WESSELING, P. (1980). The rate of convergence of a multiple grid method. In: Numerical Analysis (G.A. Watson, Ed.). Lect. Notes in Math. 773. Berlin: Springer-Verlag.
- WESSELING, P. and SONNEVELD, P. (1979). Numerical experiments with a multiple grid and a preconditioned Lanczos type method. Report NA-32, Delft University of Technology.

- WILLIAMS, J. (1979). Alternating direction implicit methods. In: A Survey of Numerical Methods for Partial Differential Equations. (I. Gladwell and R. Wait, Eds.). Oxford: Clarendon Press.
- YOUNG, D.M. (1971). Iterative solution of large linear systems. New York: Academic Press.