TEL-AVIV UNIVERSITY
The Raymond and Beverly Sackler Faculty of Exact Sciences
School of Mathematical Sciences

Bounded-Error Finite Difference Schemes for Initial Boundary Value Problems on Complex Domains

Thesis submitted for the degree "Doctor of Philosophy"

by

Adi Ditkowski

Submitted to the Senate of Tel-Aviv University
June 1997

This work was carried out under the supervision of

Professor Saul S. Abarbanel

This work is dedicated to the memory of my late father

HAIM DITKOWSKI

I wish to express my gratitude and deepest appreciation to Professor Saul S. Abarbanel for his guidance, counseling, encouragement and for his friendship.

I would like to thank my wife Sigal, my mother Yona, and my sister Idit for standing by me during the past years.

I wish to thank my colleagues in TAU. In particular, I would like to thank Doron Levy for the fruitful discussions, and for the time we spent together.

I would also like to thank the Institute for Computer Applications in Science and Engineering (ICASE) , NASA Langley Research Center, Hampton, VA, for its warm hospitality.

Finally, I would like to thank the Roberto Nemirovsky doctoral fellowship and the Bauer-Neuman Chair in applied mathematics and theoretical mechanics for supporting this research.

Contents

Introduction 1

1 General theory and description of the method 7

1.1 The one dimensional case 7

1.2 The multi dimensional case 11

2 The diffusion equation 17

2.1 Construction of the scheme 17

2.2 Numerical example 23

2.3 Conclusions 27

2.4 Appendix 28

3 The advection diffusion equation 37

3.1 Construction of the scheme 38

3.2 Numerical examples 44

3.2.1 One dimensional case 44

3.2.2 A steady state two dimensional case 46

3.2.3 A 2-D time dependent example 51

3.3 Conclusions 53

3.4 Appendix, The case $a < 0$ 54

4 Mixed derivatives and parabolic systems 57

4.1 The scalar mixed derivatives problem 57

4.2 Second order scheme for the scalar mixed derivatives problem 63

4.3 Parabolic systems; the diffusion part of the Navier-Stokes equations . . . 69

5 Bounded error schemes for the wave equation **76**

5.1 Description of the method 76

5.2 Construction of the scheme 81

Summary **86**

Bibliography **88**

Introduction

This thesis is concerned with the construction of bounded-error finite-difference numerical schemes, for initial boundary value problems (IBVP), on complex, multi-dimensional shapes. Applications where such problems arise are, among others, heat transfer, acoustic and electro-magnetic wave propagation and fluid dynamics. In constructing numerical schemes for these problems some difficulties may arise. Some of them are:

1. *Imposing stable boundary conditions:* Close to the boundary one-sided approximations have to be used. These approximations have to be stable, fulfill the boundary conditions and maintain the global accuracy of the scheme.

2. *Irregular domains:* In multi-dimensional domains the boundary may not necessarily coincide with the nodes of the mesh.

3. *Long time integration:* The error may grow rapidly in time even when the scheme is stable in the classical sense.

4. *Low viscosity:* For some schemes large numerical oscillations may occur when the boundary-layers are not resolved.

5. *Lack of 'modularity':* Suppose we have two differentiation matrices D_x and D_y in the x and y directions respectively, all the eigenvalues of which have non positive real part (i.e. they are strictly stable, see the definition below). The sum of these matrices will not necessarily preserve this virtue. This is an example of lack of modularity.

In this thesis a method is presented that allows one to construct finite-difference schemes which resolve some of these difficulties.

The standard way to construct finite-difference schemes that numerically solve a PDE, or a system of PDE's, on a given mesh is to find a *consistent* approximation to the given problem and then prove that the approximation is *stable*. Lax's equivalence theorem insures that the scheme *converges*, i.e. that for a fixed time T the numerical solution to the scheme converges to the analytic solution of the differential problem as the mesh size $h \to 0$, see for example [33], [13] . Note that even when the scheme is stable the error may grow exponentially in time, see [5]. Additional types of stability are *strong stability* which means that an estimate of the solution at any given time is obtained in terms of the forcing function, initial data and boundary data; and *strict stability* which means that the energy dissipation introduced by the boundaries is essentially preserved by the numerical scheme. In the case of semi-discrete schemes strict stability implies that all the eigenvalues of the coefficient matrix of the corresponding ODE system have non positive real part.

The stability analysis for fully-discrete algorithms can be a difficult task. The analysis is somewhat simpler when carried out for semi-discrete schemes. If one converts a semi-discrete scheme to a fully-discrete one by using Runge-Kutta or other multi-step methods, then stability is assured under conditions given in Kreiss and Wu [22] or Levy and Tadmor [23].

There are essentially two methods to analyze stability of finite-difference approximations of PDE's with non-periodic boundary conditions: the energy method and the Laplace transform method. The first paper in the area of the Laplace transform method was written by Goganov and Ryabenkii [14] in 1963. This gave necessary normal-mode conditions for stability, analogous to the Von Neumann necessary condition for pure initial value problem. The sufficient condition was given by Kreiss [18] and the complete theory for dissipative schemes was presented in [19]. Extentions of Kreiss results, including some nondissipative and implicit methods were in Osher [30]. A general stability theory, for hyperbolic IBVP, based on the Laplace transform method, for the fully-discrete case, was presented by Gustafsson, Kreiss and Sundström (G.K.S.) [12]. This was a breakthrough in the study of the stability analysis of finite-difference schemes. Later, the theory was generalized to the semi-discrete case by Strikwerda [37]. In [13] Gustafsson, Kreiss and Oliger prove that, under mild assumptions, G.K.S. stability

leads to strong stability. It should be realized however that the above theories are one-dimensional in nature. In the d-dimensional case they are applicable only if $d-1$ boundary conditions are periodic. Even then the application of this theory in several space dimensions is complicated. In fact, to my knowledge, there is only one example of this type of stability analysis carried out for two-dimensional 2×2 hyperbolic systems, see [4].

Energy methods for stability analysis were introduced in the 50's, see [33]. In the 70's Kreiss and Scherer [20] [21] proved the existence of difference operators approximating a hyperbolic PDE using a summation by parts formula and weighted norms. They also used a projection operator to impose the boundary conditions on semi-discrete schemes. Lately this technique was generalized by Olsson [28] [29] and Strand [35][36] who proved strict stability for a larger class of operators and orders of accuracy. Another approach to impose boundary conditions and to insure strict stability was presented by Carpenter, Gottlieb and Abarbanel [8] who used simultaneous approximation terms (SAT) to treat boundary conditions. Using this technique high order implicit schemes were constructed.

Many problems that arise from applications are concerned with solving PDE.'s in complex multidimensional geometries. Typical problems are fluid flow and electromagnetic wave propagation. Some of the methods which attack this problems are unstructured grids, body-fitted coordinates, and overlapping meshes. These methods are successfully applied in CFD. However, they suffer from two major drawbacks - the length of time it takes to generate the grid, measured in weeks for typical large scale CFD computation, and the complexity added to the computation due to the fact that the PDE's have to be transformed to the new coordinate system. Another approach is to use a cartesian grid. This method uses a 'simple' cartesian grid, but requires a complex treatment at boundaries which do not necessarily coincide with grid nodes or cell-surfaces. Papers describing the use of cartesian grids using finite difference algorithms and finite volume methods were published in the past twenty years, see for example [32] [31] [9] [11]. Lately there is a growing interest in cartesian grid methods combined with mesh refinement. One of the goals of current research is to automate grid generation, see for example [27] [6] [24] [25] and also some of the home-pages devoted to cartesian grids, e.g., Dr. John Melton :http://oldwww.nas.nasa.gov/~melton/cartesian.html and Capt.

Michael Aftosmis: http://george.arc.nasa.gov/~aftosmis/. Currently there are several industrial CFD codes that use cartesian grids, for example TRANAIR and MGAERO. The Finite-Difference Time-Domain (FDTD) method, was first applied by Yee in 1966 [38] to the problem of solving numerically Maxwell's equations. Yee used two cartesian grids, an electric field **E**-grid and a magnetic field **H**-grid which are offset from each other both spatially and temporally. A leap-frog scheme was utilized to advance the fields in time. When FDTD methods are applied to problems on complex geometries, "staircased" or "lego-type" approximations are used to represent the boundaries. These approximations to the geometry may, in certain cases, lead to significant errors [7] [15]. Mesh refinement, as well as other approaches, are used to overcome this difficulty and to resolve small-scale structures, such as a narrow slot; see [26] [17]. See also the FDTD survey paper [34].

One of the difficulties that may arise in the analysis of finite-difference schemes is that some of the properties of the discrete operators are not necessarily preserved when the operators are added. For example, even if we have strictly stable differentiation matrices D_x and D_y (i.e. all their eigenvalues have non positive real part), it is not assured that the representation of $\partial/\partial x + \partial/\partial y$ by $D_x + D_y$ will possess only eigenvalues with non positive real part. This implies that it is the matrix $D_x + D_y$ that should be checked for stability. This analysis may be extremely complicated on complex multidimensional geometries. Thus arises the motivation to construct 'modular' schemes, in the sense that the properties of the operators that are essential for stability, or convergence, are preserved when the operators are added.

In this work a methodology for constructing finite-difference semi-discrete schemes, for initial boundary value problems (IBVP), on complex, multi-dimensional shapes is presented. The implementation of this methodology for constructing finite-difference approximations for various operators and orders of accuracy is discussed and selected numerical verifications are presented.

Unlike the 'standard approach' where convergence is proven by using stability, here we prove convergence directly, by deriving an equation for the error and bounding the error norm. The standard requirement of stability is replaced by *error boundness*, which means that the error norm is bounded by a function of the time t, the mesh size h and

the exact solution to the differential problem u (typically a Sobolev norm of u), i.e. $\| \epsilon \| < F(u,h,t), \quad F < \infty \; \forall \; t < \infty, \quad F \to 0$ as $h \to 0$. Throughout this work we use the error boundness in a stricter sense. We require that $\| \epsilon \|$, the L_2 norm of ϵ, be bounded by a "constant" proportional to h^m (m being the spatial order of accuracy) for all $t < \infty$, or at most grow linearly in time, the time coefficient being proportional to h^m. By Lax's equivalence theorem, a scheme which possesses an error bounded by a linear growth in time is strictly stable. Additionally, the use of error-boundedness analysis enables us not only to prove convergence directly, but also to get an estimate on the actual error generated by the scheme.

The method presented here for building up semi-discrete schemes on complex, multi-dimensional shapes has as its starting point the construction of one dimensional schemes on a uniform grid with boundary points that do not necessarily coincide with the extremal nodes of the mesh. The boundary conditions are imposed using simultaneous approximation terms (SAT) which are a generalization of the penalty method presented in [8]. The 1-D schemes are built in a way that the coefficient matrix of the corresponding ODE system which represents the error evolution in time is negative definite (N.D.) and bounded away from 0 by a constant independent of the size of the matrix, or is at least non-positive definite (N.P.D.). These properties, N.D. or N.P.D., enable us to prove that the scheme is error-bounded by a "constant", or error-bounded by linear growth in time, respectively. Since a sum of two negative (non-positive) definite matrices is a negative (non-positive) definite matrix, a multidimensional scheme can be built by adding differentiation operators each of which is negative (non-positive) definite. This is the sense in which such schemes are 'modular'.

The error boundness proof and details of constructing multidimensional schemes on complex shapes are given in chapter 1.

In chapter 2 a 4^{th}-order accurate scheme is developed for solving the diffusion equation in one or more dimensions, on irregular domains. The scheme is constructed on a rectangular grid using the method presented in chapter 1. Numerical examples in 2-D show that the method is effective even where standard schemes, stable by traditional definitions, fail. The general theory and the results presented in this chapter were published in [2].

In chapter 3 the methodology presented in chapter 1 is used to develop second order accurate schemes which solve multi-dimensional linear hyperbolic and diffusion equations on complex shapes. These algorithms are used to solve the linear advection-diffusion equation, including the low viscosity case. Numerical examples show that the method can give a good approximation to the solution outside the boundary-layer even when the viscosity is low, and the grid is coarse with respect to the boundary layer thickness. Standard schemes converge much slower and generate oscillations. The material presented in this chapter was published in [3].

In chapter 4 the methodology presented in chapter 1 is adopted to construct schemes for parabolic equations and systems containing mixed-derivatives. A second order accurate bounded-error scheme is constructed to represent a scalar mixed-derivatives parabolic operator. This scheme is then generalized to solve the diffusion part of the Navier-Stokes equations in two and three space dimensions. This generalization may also be applicable to other parabolic systems.

In chapter 5 the same method is adopted to solve the wave equation. The difficulties that arise from attempts to apply the method presented in chapter 1 naively are discussed, and a way to resolve them and to solve the problem is proposed. Consequently we have a second-order accurate bounded-error scheme to solve the wave equation on complex shapes.

Chapter 1

General theory and description of the method

1.1 The one dimensional case

We consider the following problem

$$\frac{\partial u}{\partial t} = L(u) + f(x,t); \qquad \Gamma_L \leq x \leq \Gamma_R, \;\; t \geq 0 \tag{1.1.1a}$$

$$u(x,0) = u_0(x) \tag{1.1.1b}$$

$$B_L(u(\Gamma_L,t)) = g_L(t) \tag{1.1.1c}$$

$$B_R(u(\Gamma_R,t)) = g_R(t) \tag{1.1.1d}$$

Where L, B_L, B_R are linear differential operators. For example: in the inhomogeneous diffusion equation with Dirichlet boundary condition $L(u) = k\frac{\partial^2 u}{\partial x^2} + f(x,t)$; $k > 0$, $B_L(u(\Gamma_L,t)) = u(\Gamma_L,t)$, and $B_R(u(\Gamma_R,t)) = u(\Gamma_R,t)$. In the inhomogeneous hyperbolic equation with Dirichlet boundary condition $L(u) = a\frac{\partial u}{\partial x} + f(x,t)$; for $a < 0$, $B_L(u(\Gamma_L,t)) = u(\Gamma_L,t)$, and no boundary condition is given on the right side.

Let us spatially discretize (1.1.1a) on the following uniform grid:

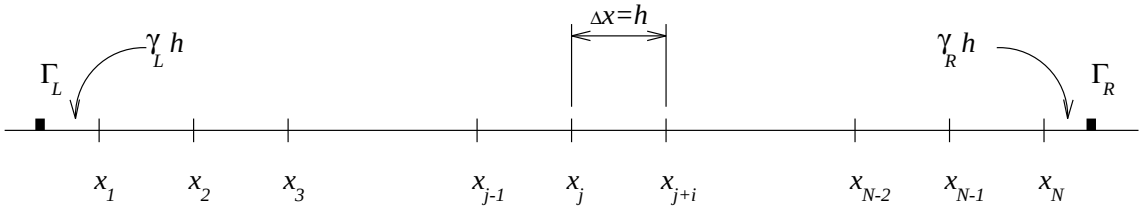

Figure 1.1: One dimensional grid.

Note that the boundary points do not necessarily coincide with x_1 and x_N. Set $x_{j+1} - x_j = h$, $1 \le j \le N-1$; $x_1 - \Gamma_L = \gamma_L h$, $0 \le \gamma_L < 1$; $\Gamma_R - x_N = \gamma_R h$, $0 \le \gamma_R < 1$.

The projection of the exact solution $u(x,t)$ to (1.1.1) onto the above grid is $u_j(t) = u(x_j,t) \stackrel{\triangle}{=} \mathbf{u}(t)$, similarly $f_j(t) = f(x_j,t) \stackrel{\triangle}{=} \mathbf{f}(t)$ is the projection of the inhomogeneous term. Let $\tilde{D}$ be a matrix representing numerical approximation to the differential operator L at internal points, without specifying yet how it is being built. Then we may write

$$\frac{d}{dt}\mathbf{u}(t) = \tilde{D}\mathbf{u}(t) + \mathbf{B} + \mathbf{f} + \mathbf{T} \tag{1.1.2}$$

where $\mathbf{T}$ is the truncation error due to the numerical differentiation. The boundary vector $\mathbf{B}$ has entries whose values depend on $g_L, g_R, \gamma_L, \gamma_R$ in such a way that $\tilde{D}\cdot + \mathbf{B}$ represents the differential operator L everywhere to the desired accuracy. The standard way of finding a numerical approximate solution to (1.1.1) is to omit $\mathbf{T}$ from (1.1.2) and solve

$$\frac{d}{dt}\mathbf{v}(t) = \tilde{D}\mathbf{v}(t) + \mathbf{B} + \mathbf{f} \tag{1.1.3}$$

where $\mathbf{v}(t)$ is the numerical approximation to the projection $\mathbf{u}(t)$. An equation for the solution error vector, $\boldsymbol{\epsilon}(t) = \mathbf{u}(t) - \mathbf{v}(t)$, can be found by subtracting (1.1.3) from (1.1.2):

$$\frac{d}{dt}\,\boldsymbol{\epsilon} = \tilde{D}\,\boldsymbol{\epsilon}(t) + \mathbf{T}(t) \tag{1.1.4}$$

Unlike the 'standard approach' where convergence is proven by using stability, here we prove convergence directly, by deriving an equation for the error and bounding the error norm. The standard requirement of stability is replaced by *error boundness* which means that the error norm is bounded by a function of the time t, the mesh size h and the exact solution to the differential problem u (typically a Sobolev norm of u), i.e. $\| \boldsymbol{\epsilon} \| < F(u,h,t)$, $\quad F < \infty \ \forall\ t < \infty$, $\quad F \to 0$ as $h \to 0$. Throughout this work, our requirement for *error boundness is that either* $\| \boldsymbol{\epsilon} \|$, *the* L_2 *norm of* $\boldsymbol{\epsilon}$, *be bounded by a "constant" proportional to* h^m (m being the spatial order of accuracy) *for all* $t < \infty$, *or at most grow linearly in time, the time coefficient being proportional to* h^m. Note that this requirement is more severe than strong stability, which allows for exponential temporal growth of the error.

In many cases it can be shown that if $\tilde{D}$ is constructed using central differencing in

a standard manner, i.e., away from the boundaries the numerical second derivative is symmetric and the numerical first derivative is antisymmetric, (and near the boundaries one uses "non-symmetric" differentiation), then there are ranges of γ_R and γ_L for which $\tilde{D}$ is not negative definite. Since in the multi-dimensional case one may encounter all values of $0 \leq \gamma_L, \gamma_R \leq 1$, this is unacceptable.

The "common practice" is to use $\tilde{D}\mathbf{v}+\mathbf{B}$ to approximate $\tilde{D}\mathbf{u}+\mathbf{B}$. Here, however, the basic idea is to use a differentiation matrix D to approximate the differential operator L everywhere and use a penalty-like term in the numerical algorithm to represent the boundary conditions. This way of imposing the boundary conditions gives us more degrees of freedom in the construction of the scheme. Special properties of the scheme, like non-positive definiteness in L_2 norm, can now be achieved using these extra degrees of freedom. This idea of using penalty-like term was presented in the SAT procedure of ref [8]; here, however, it will be modified and applied in a different manner.

Note first that the solution projection $u_j(t)$ satisfies, besides (1.1.2), the following differential equation:

$$\frac{d\mathbf{u}}{dt} = D\mathbf{u} + \mathbf{f} + \mathbf{T}_e \tag{1.1.5}$$

where now D is indeed a *representation* of L, *that does not use the boundary values*, and therefore $\mathbf{T}_e \neq \mathbf{T}$ but it too is a truncation error due to differentiation.

Next let the semi-discrete problem for $\mathbf{v}(t)$ be, instead of (1.1.3),

$$\frac{d\mathbf{v}}{dt} = [D\mathbf{v} - \tau_L(A_L\mathbf{v} - \mathbf{g}_L) - \tau_R(A_R\mathbf{v} - \mathbf{g}_R)] + \mathbf{f} \tag{1.1.6}$$

where $\mathbf{g}_L = (1, \ldots, 1)^T g_L(t)$; $\mathbf{g}_R = (1, \ldots, 1)^T g_R(t)$, are vectors created from the left and right boundary values as shown. The matrices A_L and A_R are defined by the relations:

$$A_L\mathbf{u} = \mathbf{g}_L - \mathbf{T}_L; \qquad A_R\mathbf{u} = \mathbf{g}_R - \mathbf{T}_R, \tag{1.1.7}$$

i.e., each row in $A_L(A_R)$ is composed of the coefficients extrapolating $\mathbf{u}$ to its boundary value $\mathbf{g}_L(\mathbf{g}_R)$, at $\Gamma_L(\Gamma_R)$ to within the desired order of accuracy. (The error is then $\mathbf{T}_L(\mathbf{T}_R)$).

The diagonal matrices τ_L and τ_R are given by

$$\tau_L = \text{ diag } (\tau_{L_1}, \tau_{L_2}, \ldots, \tau_{L_N}); \qquad \tau_R = \text{ diag } (\tau_{R_1}, \tau_{R_2}, \ldots, \tau_{R_N}) \ . \tag{1.1.8}$$

Though in principle the penalty terms $(A_L\mathbf{v} - \mathbf{g}_L)$ and $(A_R\mathbf{v} - \mathbf{g}_R)$ are added to each point, in practice they are added just near the boundaries, i.e. most of the τ_{L_j}s and τ_{R_j}s are zero. Subtracting (1.1.6) from (1.1.5) we get

$$\frac{d\,\boldsymbol{\epsilon}}{dt} = [D\,\boldsymbol{\epsilon} - \tau_L A_L\,\boldsymbol{\epsilon} - \tau_R A_R\,\boldsymbol{\epsilon}] + \mathbf{T}_1 \tag{1.1.9}$$

where

$$\mathbf{T}_1 = \mathbf{T}_e - \tau_L \mathbf{T}_L - \tau_R \mathbf{T}_R$$

Taking the scalar product of $\boldsymbol{\epsilon}$ with (1.1.9) one gets:

$$\begin{aligned} \frac{1}{2}\frac{d}{dt} \parallel \boldsymbol{\epsilon} \parallel^2 &= (\boldsymbol{\epsilon}, (D - \tau_L A_L - \tau_R A_R)\,\boldsymbol{\epsilon}) + (\boldsymbol{\epsilon}, \mathbf{T}_1) \\ &= (\boldsymbol{\epsilon}, M\,\boldsymbol{\epsilon}) + (\boldsymbol{\epsilon}, \mathbf{T}_1) \end{aligned} \tag{1.1.10}$$

We notice that $(\boldsymbol{\epsilon}, M\,\boldsymbol{\epsilon})$ is $(\boldsymbol{\epsilon}, (M + M^T)\,\boldsymbol{\epsilon})/2$, where

$$M = D - \tau_L A_L - \tau_R A_R. \tag{1.1.11}$$

If $M + M^T$ can be made negative definite then

$$(\boldsymbol{\epsilon}, (M + M^T)\,\boldsymbol{\epsilon})/2 \leq -c_0 \parallel \boldsymbol{\epsilon} \parallel^2, \qquad (c_0 > 0), \tag{1.1.12}$$

where $-c_0$ is an upper bound on the largest eigenvalue of $M + M^T$. Equation (1.1.10) then becomes

$$\frac{1}{2}\frac{d}{dt} \parallel \boldsymbol{\epsilon} \parallel^2 \leq -c_0 \parallel \boldsymbol{\epsilon} \parallel^2 + (\boldsymbol{\epsilon}, \mathbf{T}_1),$$

and using Schwarz's inequality we get after dividing by $\parallel \boldsymbol{\epsilon} \parallel$

$$\frac{d}{dt} \parallel \boldsymbol{\epsilon} \parallel \leq -c_0 \parallel \boldsymbol{\epsilon} \parallel + \parallel \mathbf{T}_1 \parallel$$

and therefore (using the fact that $\mathbf{v}(0) = \mathbf{u}(0)$)

$$\parallel \boldsymbol{\epsilon} \parallel \leq \frac{\parallel \mathbf{T_1} \parallel_M}{c_0}(1 - e^{-c_0 t}) \tag{1.1.13}$$

where the *"constant"* $\parallel \mathbf{T_1} \parallel_M = \max_{0 \leq \tau \leq t} \parallel \mathbf{T_1}(\tau) \parallel$. This *"constant"* is function of the exact solution u and its derivatives.

If we indeed succeed in constructing M such that $M + M^T$ is negative definite, with $c_0 > 0$ independent of the size of the matrix M as it increases, then it follows

from (1.1.13) that the norm of the error will be bounded for all t by a constant which is $O(h^m)$ where m is the order of the spatial accuracy of the finite difference scheme (1.1.6). The algorithm is then a *bounded error scheme*, and as $h \to 0$, $\mathbf{v}(t)$ converges to $\mathbf{u}(t)$.

When $c_0 = 0$, as in the case of hyperbolic equations, the differential inequality is

$$\frac{d}{dt} \parallel \boldsymbol{\epsilon} \parallel \leq \parallel \mathbf{T}_1 \parallel \tag{1.1.14}$$

leading to

$$\parallel \boldsymbol{\epsilon} \parallel \leq \parallel \mathbf{T}_1 \parallel_M t \ , \tag{1.1.15}$$

i.e. a linear growth in time, a result typical of hyperbolic systems; convergence, however, is unaffected.

1.2 The multi dimensional case

In this section we show how to use a one-dimensional scheme, whose properties were described in the previous section, as a building block for multi-dimensional schemes of the type $\frac{\partial u}{\partial t} = \sum_{r=1}^{d} L^{(x_r)}(u) + f(\mathbf{x},t)$; $\mathbf{x} = (x_1, \ldots, x_d) \in R^d$. For the sake of simplicity we describe in detail the construction in the two-dimensional case ($d = 2$). A brief explanation on how to construct a multi-dimensional schemes in d dimensions is given at the end of this section. The more general case, where the evolution operator contains mixed derivatives, will be examined in chapter 4.

We consider the following inhomogeneous linear differential equation, with constant coefficients, in a domain Ω. To begin with we shall assume that Ω is convex and has a boundary curve $\partial\Omega \in \mathcal{C}^2$. The convexity restriction is for the sake of simplicity in presenting the basic idea; it will be removed later. We thus have

$$\frac{\partial u}{\partial t} = L^{(x)}(u) + L^{(y)}(u) + f(x,y,t); \qquad x,y \in \Omega; \ \ t \geq 0 \tag{1.2.1a}$$

$$u(x,y,0) = u_0(x,y) \tag{1.2.1b}$$

$$B(u(x,y,t))|_{\partial\Omega} = u_B(t) \tag{1.2.1c}$$

We shall refer to the following grid representation:

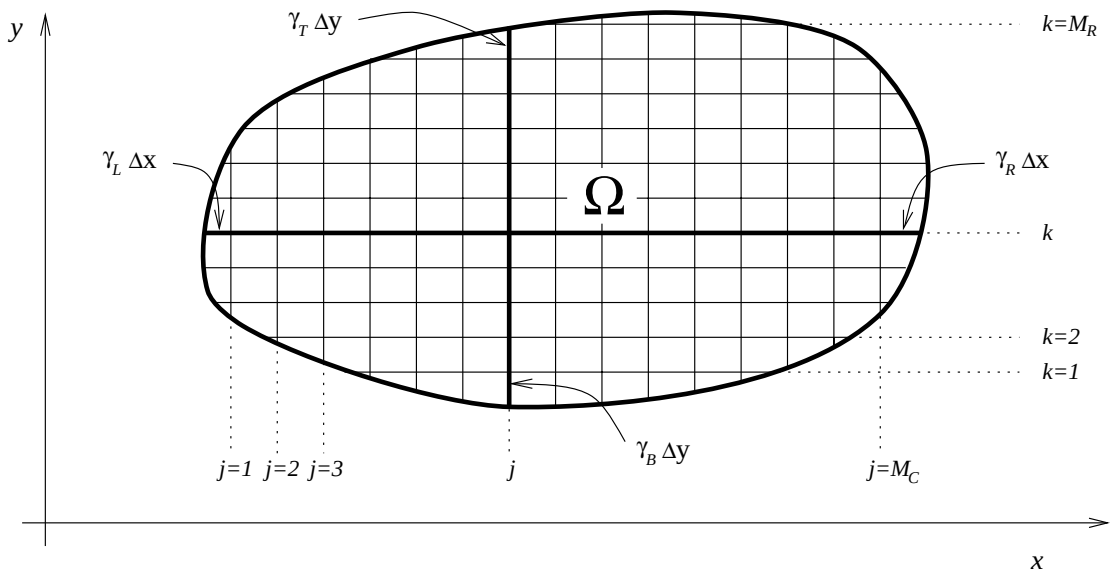

Figure 1.2: Two dimensional grid.

We have M_R rows and M_C columns inside Ω. Each row and each column has a discretized structure as in the one 1-D case, see Figure 1.2, with Δx not necessarily being equal to Δy. Let the number of grid points in the k^{th} row be denoted by R_k and similarly let the number of grid points in the j^{th} column be C_j. Let the solution projection be designated by $U_{j,k}(t)$. By $\mathbf{U}(t)$ we mean, by analogy to the 1-D case,

$$\begin{aligned} \mathbf{U}(t) &= (u_{1,1}, u_{2,1}, \ldots, u_{R_1,1}; u_{1,2}, u_{2,2}, \ldots, u_{R_2,2}; \ldots; u_{1,M_R}, u_{2,M_R}, \ldots u_{R_{M_R},M_R}) \\ &\equiv (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M_R}). \end{aligned} \tag{1.2.2}$$

One should note that the x-position of $j = 1$ is different on different rows (i.e. different k's).

Thus, we have arranged the solution projection array in vectors according to rows, starting from the bottom of Ω. The projection of the forcing function, $\mathbf{F}(t)$, was also arranged in the same manner.

If we arrange this array by columns (instead of rows) we will have the following structure

$$\begin{aligned} \mathbf{U}^{(C)}(t) &= (u_{1,1}, u_{1,2}, \ldots, u_{1,C_1}; u_{2,1}, u_{2,2}, \ldots, u_{2,C_2}; \ldots; u_{M_C,1}, u_{M_C,2}, \ldots, u_{M_C,C_{M_C}}) \\ &\equiv (\mathbf{u}_1^{(C)}, \mathbf{u}_2^{(C)}, \ldots, \mathbf{u}_{M_C}^{(C)}) \end{aligned} \tag{1.2.3}$$

Since $\mathbf{U}^{(C)}(t)$ is just a permutation of $\mathbf{U}(t)$, there must exist an orthogonal matrix P such that

$$\mathbf{U}^{(C)}(t) = P\mathbf{U}. \tag{1.2.4}$$

If the length of $\mathbf{U}(t)$ is ℓ, then P is an $\ell \times \ell$ matrix whose each row contains $\ell - 1$ zeros and a single 1 somewhere.

The differential operator $L^{(x)}$ in (1.2.1a) is represented on the k^{th} row by the differentiation matrix $D_k^{(x)}$, whose structure is the same as that of D in (1.1.5). Similarly let $L^{(y)}$ be given on the j^{th} column by $D_j^{(y)}$, whose structure is also given by (1.1.5). With this notation the projection of $L^{(x)} + L^{(y)}$ is:

$$\left(L^{(x)} + L^{(y)}\right) u_{ij}(t) = \left[\mathcal{D}^{(x)}\mathbf{U} + \mathcal{D}^{(y)}\mathbf{U}^{(C)} + \mathbf{T}_e^{(x)} + \mathbf{T}_e^{(y)}\right]_{ij} \tag{1.2.5}$$

where $\mathcal{D}^{(x)}$ and $\mathcal{D}^{(y)}$ are $(\ell \times \ell)$ matrices with a block structures shown in (1.2.6).

$$\mathcal{D}^{(x)} = \begin{bmatrix} D_1^{(x)} & & & \\ & D_2^{(x)} & & \\ & & \ddots & \\ & & & D_{M_R}^{(x)} \end{bmatrix}; \mathcal{D}^{(y)} = \begin{bmatrix} D_1^{(y)} & & & \\ & D_2^{(y)} & & \\ & & \ddots & \\ & & & D_{M_C}^{(y)} \end{bmatrix} \tag{1.2.6}$$

$\mathbf{T}_e^{(x)}$ and $\mathbf{T}_e^{(y)}$ are the truncation errors associated with $\mathcal{D}^{(x)}$ and $\mathcal{D}^{(y)}$, respectively. We now call attention to the fact that $\mathcal{D}^{(x)}$ and $\mathcal{D}^{(y)}$ do not operate on the same vector. This is fixed using (1.2.4):

$$\left(L^{(x)} + L^{(y)}\right) u_{ij}(t) = \left(L^{(x)} + L^{(y)}\right) \mathbf{U} = (\mathcal{D}^{(x)} + P^T\mathcal{D}^{(y)}P)\mathbf{U} + \mathbf{T}_e^{(x)} + P^T\mathbf{T}_e^{(y)} \tag{1.2.7}$$

Thus (1.2.1a) becomes, by analogy to (1.1.5),

$$\frac{d\mathbf{U}}{dt} = (\mathcal{D}^{(x)} + P^T\mathcal{D}^{(y)}P)\mathbf{U} + (\mathbf{T}_e^{(x)} + P^T\mathbf{T}_e^{(y)}) + \mathbf{F} \tag{1.2.8}$$

Before proceeding to the semi-discrete problem let us define:

$$M_k^{(x)} = D_k^{(x)} - \tau_{L_k} A_{L_k} - \tau_{R_k} A_{R_k} \tag{1.2.9}$$

where τ_{L_k}, A_{L_k} are the τ_L and A_L defined in section 1.1, appropriate to the k^{th} row; similarly for τ_{R_k} and A_{R_k}. In the same way, define

$$M_j^{(y)} = D_j^{(y)} - \tau_{B_j} A_{B_j} - \tau_{T_j} A_{T_j} \tag{1.2.10}$$

where the subscripts B and T stand for bottom and top respectively.

We can now write the semi-discrete problem by analogy to (1.1.6)

$$\frac{d\mathbf{V}}{dt} = (\mathcal{M}^{(x)} + P^T\mathcal{M}^{(y)}P)\mathbf{V} + \mathbf{G}^{(x)} + P^T\mathbf{G}^{(y)} + \mathbf{F} \quad ^{\dagger} \tag{1.2.11}$$

where $\mathbf{V}$ is the numerical approximation to $\mathbf{U}$;

$$\mathcal{M}^{(x)} = \begin{bmatrix} M_1^{(x)} & & & \\ & M_2^{(x)} & & \\ & & \ddots & \\ & & & M_{M_R}^{(x)} \end{bmatrix}; \mathcal{M}^{(y)} = \begin{bmatrix} M_1^{(y)} & & & \\ & M_2^{(y)} & & \\ & & \ddots & \\ & & & M_{M_C}^{(y)} \end{bmatrix}; \tag{1.2.12}$$

and

$$\begin{aligned} \mathbf{G}^{(x)} &= \left[(\tau_{L_1}\mathbf{g}_{L_1} + \tau_{R_1}\mathbf{g}_{R_1}), \ldots, (\tau_{L_k}\mathbf{g}_{L_k} + \tau_{R_k}\mathbf{g}_{R_k}), \ldots, (\tau_{L_{M_R}}\mathbf{g}_{L_{M_R}} + \tau_{R_{M_R}}\mathbf{g}_{R_{M_R}})\right], \\ \mathbf{G}^{(y)} &= \left[(\tau_{B_1}\mathbf{g}_{B_1} + \tau_{T_1}\mathbf{g}_{T_1}), \ldots, (\tau_{B_j}\mathbf{g}_{B_j} + \tau_{T_j}\mathbf{g}_{T_j}), \ldots, (\tau_{B_{M_C}}\mathbf{g}_{B_{M_C}} + \tau_{T_{M_C}}\mathbf{g}_{T_{M_C}})\right]. \end{aligned} \tag{1.2.13}$$

Subtracting (1.2.11) from (1.2.8) we get in a fashion similar to the derivation of (1.1.9):

$$\frac{d\mathbf{E}}{dt} = (\mathcal{M}^{(x)} + P^T\mathcal{M}^{(y)}P)\mathbf{E} + \mathbf{T_2} \tag{1.2.14}$$

where $\mathbf{E} = \mathbf{U} - \mathbf{V}$ is the two dimensional array of the errors, ϵ_{ij}, arranged by rows as a vector. $\mathbf{T_2}$ is proportional to the truncation error.

The time rate of change of $\| \mathbf{E} \|^2$ is given by

$$\frac{1}{2}\frac{d}{dt} \| \mathbf{E} \|^2 = (\mathbf{E}, (\mathcal{M}^{(x)} + P^T\mathcal{M}^{(y)}P)\mathbf{E}) + (\mathbf{E}, \mathbf{T_2}) \tag{1.2.15}$$

The symmetric part of $\mathcal{M}^{(x)} + P^T\mathcal{M}^{(y)}P$ is given by

$$\frac{1}{2}[(\mathcal{M}^{(x)} + \mathcal{M}^{(x)^T}) + P^T(\mathcal{M}^{(y)} + \mathcal{M}^{(y)^T})P] \tag{1.2.16}$$

---

† Note that when this scheme is used in practice, the 1-D algorithem (1.1.6) is implemented on each row, to compute the numerical approximation to $L^{(X)}$, and on each column, to compute the numerical approximation to $L^{(y)}$. Therefor the scheme may be written as:

$$\frac{d\mathbf{v_{ij}}}{dt} = \left[\mathcal{M}^{(x)}\mathbf{V} + \mathbf{G}^{(x)} + \mathcal{M}^{(y)}\mathbf{V}^{(C)} + \mathbf{G}^{(y)} + \mathbf{F}\right]_{ij}.$$

Writing the scheme in the manner of equation (1.2.11), enables us to use standard linear-algebra manipulations, and thus make the proof easier.

Clearly $\mathcal{M}^{(x)} + \mathcal{M}^{(x)^T}$ and $\mathcal{M}^{(y)} + \mathcal{M}^{(y)^T}$ are block-diagonal matrices with typical blocks given by $M_k^{(x)} + M_k^{(x)^T}$ and $M_j^{(y)} + M_j^{(y)^T}$. We have already shown in the one dimensional case that each one of those blocks is either negative definite and bounded away from zero by c_0, or is non-positive definite. Therefore the operator (1.2.16) is also negative definite and bounded away from zero, or non-positive definite. The rest of the proof follows the one dimensional case and thus the norm of the error, $\| E \|$, is bounded by a constant, or, in the hyperbolic case, at most grows linearly with t.

If the domain Ω is not convex or simply connected then either rows or columns, or both, may be "interrupted" by $\partial\Omega$. In that case the values of the solution on each "internal" interval (see Figure 1.3 below) are taken as *separate* vectors.

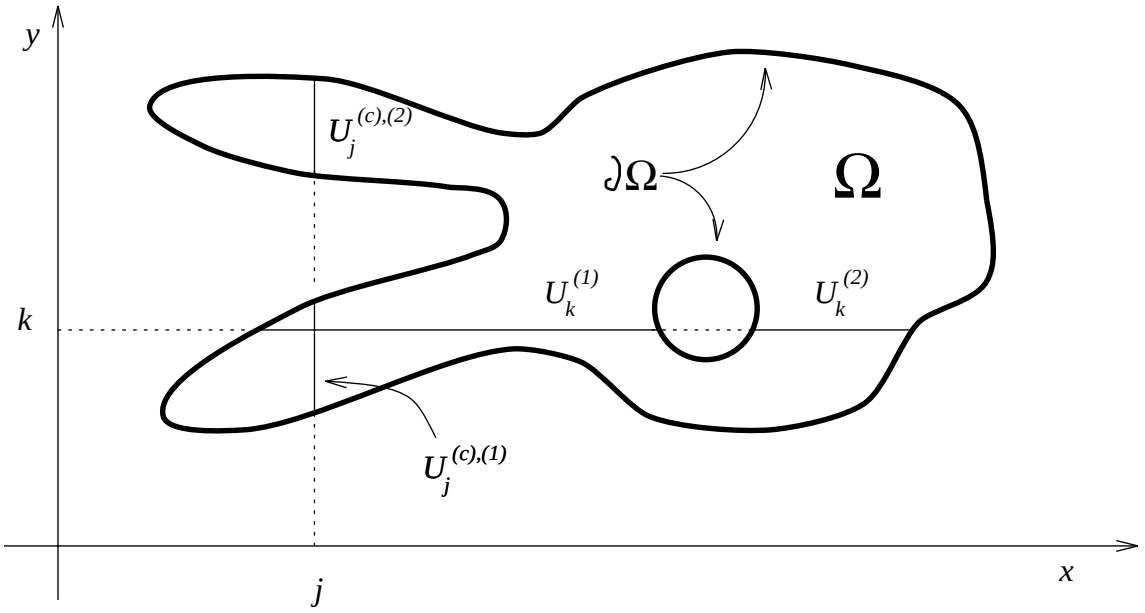

Figure 1.3: Two dimensional grid, non convex domain.

Decomposing "interrupted" vectors in this fashion leaves the previous analysis unchanged. The length of $\mathbf{U}$ (or $\mathbf{U}^{(C)}$) is again ℓ, where ℓ is the number of grid nodes inside Ω. The differentiation and permutation matrices remain $\ell \times \ell$. Note that adding more "holes" inside $\partial\Omega$ does not change the general approach.

If the domain $\Omega \in R^d$, in practice algorithm (1.1.6) is implemented on each direction x_r, $r = 1, \ldots, d$, in the same way it was implemented for each row, to compute in the

two-dimensional case, see footnote to equation (1.2.11). In order to prove the convergence of this algorithm we write the scheme as in equation (1.2.11), using the proper permutation matrices. Then we get a bound on on the error norm as was done in the two-dimensional case.

The following chapters describe how to construct M, see (1.1.11), for various operators and orders of accuracy, so that the basic assumption of the theory, i.e. that M is either negative definite or non-positive definite is fulfilled. Selected numerical verifications are also presented.

Chapter 2

The diffusion equation

This chapter is concerned with 4^{th}-order approximations to the the diffusion equation in one and two dimensions, on irregular domains, using the method presented in chapter 1.

In section 2.1 we develop the one-dimensional semi-discrete algorithm that approximate second derivative to 4^{th}-order accuracy. This scheme satisfies the requirement presented in chapter 1, therefore the scheme is bounded by a *"constant"* and can be generalized to the multi-dimensional case, i.e. the Laplace operator.

Section 2.2 presents numerical results in two space dimensions demonstrating the error boundeness of the scheme, constructed in section 2.1. These results show that the method is effective even where standard schemes, stable by traditional definitions, fail.

2.1 Construction of the scheme

We consider the following problem

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} + f(x,t); \qquad \Gamma_L \leq x \leq \Gamma_R, \quad t \geq 0, \quad k > 0 \tag{2.1.1a}$$

$$u(x,0) = u_0(x) \tag{2.1.1b}$$

$$u(\Gamma_L, t) = g_L(t) \tag{2.1.1c}$$

$$u(\Gamma_R, t) = g_R(t) \tag{2.1.1d}$$

and $f(x,t) \in C^4$.

As in chapter 1 let us discretize (2.1.1a) spatially on the following uniform grid:

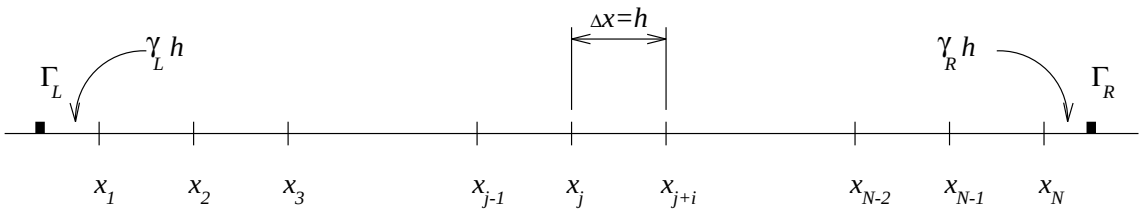

Figure 2.1: One dimensional grid.

Note that the boundary points do not necessarily coincide with x_1 and x_N. Set $x_{j+1} - x_j = h$, $1 \leq j \leq N-1$; $x_1 - \Gamma_L = \gamma_L h$, $0 \leq \gamma_L < 1$; $\Gamma_R - x_N = \gamma_R h$, $0 \leq \gamma_R < 1$.

The projection of the exact solution $u(x,t)$ to (2.1.1) onto the above grid is $u_j(t) = u(x_j,t) \stackrel{\triangle}{=} \mathbf{u}(t)$, similarly $f_j(t) = f(x_j,t) \stackrel{\triangle}{=} \mathbf{f}(t)$ is the projection of the inhomogeneous term. Let $\mathbf{v}(t)$ be the numerical approximation to the projection $\mathbf{u}(t)$, given by equation (1.1.6).

$$\begin{aligned} \frac{d\mathbf{v}}{dt} &= [D\mathbf{v} - \tau_L(A_L\mathbf{v} - \mathbf{g}_L) - \tau_R(A_R\mathbf{v} - \mathbf{g}_R)] + \mathbf{f}(t) \\ &= M\mathbf{v} + \tau_L\mathbf{g}_L + \tau_R\mathbf{g}_R + \mathbf{f}(t) \end{aligned} \tag{2.1.2}$$

where

$$\frac{d\mathbf{u}}{dt} = D\mathbf{u} + \mathbf{f}(t) + \mathbf{T}_e.$$

$\mathbf{g}_L = (1,\ldots,1)^T g_L(t)$; $\mathbf{g}_R = (1,\ldots,1)^T g_R(t)$, are vectors created from the left and right boundary values as shown. The matrices A_L and A_R are defined by the relations:

$$A_L\mathbf{u} = \mathbf{g}_L - \mathbf{T}_L; \qquad A_R\mathbf{u} = \mathbf{g}_R - \mathbf{T}_R,$$

i.e., each row in $A_L(A_R)$ is composed of the coefficients extrapolating $\mathbf{u}$ to its boundary value $\mathbf{g}_L(\mathbf{g}_R)$, at $\Gamma_L(\Gamma_R)$ to within the desired order of accuracy. (The error is then $\mathbf{T}_L(\mathbf{T}_R)$).

The diagonal matrices τ_L and τ_R are given by

$$\tau_L = \text{ diag } (\tau_{L_1}, \tau_{L_2}, \ldots, \tau_{L_N}); \qquad \tau_R = \text{ diag } (\tau_{R_1}, \tau_{R_2}, \ldots, \tau_{R_N})$$

The numerical solution error vector is given by,

$$\boldsymbol{\epsilon}(t) = \mathbf{u}(t) - \mathbf{v}(t)$$

where

$$\frac{d\,\boldsymbol{\epsilon}}{dt} = M\mathbf{v} + \mathbf{T}_1$$

and

$$\mathbf{T}_1 = \mathbf{T}_e - \tau_L \mathbf{T}_L - \tau_R \mathbf{T}_R.$$

The rest of this section is concerned with the case of $m = 4$, i.e, a spatially fourth order accurate finite difference algorithm.

Let the $n \times n$ differentiation matrix, D, be given by

$$\frac{1}{12h^2}\begin{bmatrix}
45 & -154 & 214 & -156 & 61 & -10 & & & & & & \\
10 & -15 & -4 & 14 & -6 & 1 & & & & & & \\
-1 & 16 & -30 & 16 & -1 & & & & & & & \\
 & -1 & 16 & -30 & 16 & -1 & & & & & & \\
 & & -1 & 16 & -30 & 16 & -1 & & & & & \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \\
 & & & & -1 & 16 & -30 & 16 & -1 & & & \\
 & & & & & -1 & 16 & -30 & 16 & -1 & & \\
 & & & & & & -1 & 16 & -30 & 16 & -1 & \\
 & & & & & & 1 & -6 & 14 & -4 & -15 & 10 \\
 & & & & & & -10 & 61 & -156 & 214 & -154 & 45
\end{bmatrix} \tag{2.1.3}$$

The upper two rows and the lower two rows represent non-symmetric fourth order accurate approximation to the second derivative without using boundary values. The internal rows are symmetric and represent central differencing approximation to u_{xx} to the same order. Note that D is not negative definite, and neither is the symmetric part of $\frac{1}{2}(D + D^T)$ which is given by:

$$\frac{1}{24h^2}\begin{bmatrix}
90 & -144 & 213 & -156 & 61 & -10 & & & & & & & \\
-144 & -30 & 12 & 13 & -6 & 1 & & & & & & & \\
213 & 12 & -60 & 32 & -2 & 0 & & & & & & & \\
-156 & 13 & 32 & -60 & 32 & -2 & & & & & & & \\
61 & -6 & -2 & 32 & -60 & 32 & -2 & & & & & & \\
-10 & 1 & 0 & -2 & 32 & -60 & 32 & -2 & & & & & \\
 & & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \\
 & & & & & -2 & 32 & -60 & 32 & -2 & 0 & 1 & -10 \\
 & & & & & & -2 & 32 & -60 & 32 & -2 & -6 & 61 \\
 & & & & & & & -2 & 32 & -60 & 32 & 13 & -156 \\
 & & & & & & & 0 & -2 & 32 & -60 & 12 & 213 \\
 & & & & & & & 1 & -6 & 13 & 12 & -30 & -144 \\
 & & & & & & & -10 & 61 & -156 & 213 & -144 & 90
\end{bmatrix}$$

In order to construct M we need to specify A_L, A_R, τ_L and τ_R. We construct A_L as follows:

$$A_L = A_\alpha^{(L)} + c_L A_e^{(L)} \tag{2.1.4}$$

where

$$A_\alpha^{(L)} = \begin{bmatrix}
\alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & 0 & \dots & 0 \\
\alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & 0 & \dots & 0 \\
\vdots & & & & & & & \\
\alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & 0 & \dots & 0
\end{bmatrix}, \tag{2.1.5}$$

$$c_L = \text{ diag } [-20\alpha_1/71, \ \ 0, \dots, 0] \tag{2.1.6}$$

$$A_e^{(L)} = \begin{bmatrix}
-1 & 5 & -10 & 10 & -5 & 1 & 0 & \dots & 0 \\
-1 & 5 & -10 & 10 & -5 & 1 & 0 & \dots & 0 \\
\vdots & & & & & & & & \\
-1 & 5 & -10 & 10 & -5 & 1 & 0 & \dots & 0
\end{bmatrix}. \tag{2.1.7}$$

The α's are given by

$$\alpha_1 = 1 + \frac{25}{12}\gamma_L + \frac{35}{24}\gamma_L^2 + \frac{5}{12}\gamma_L^3 + \frac{1}{24}\gamma_L^4$$

$$\alpha_2 = -\left(4\gamma_L + \frac{13}{3}\gamma_L^2 + \frac{3}{2}\gamma_L^3 + \frac{1}{6}\gamma_L^4\right)$$

$$\alpha_3 = 3\gamma_L + \frac{19}{4}\gamma_L^2 + 2\gamma_L^3 + \frac{1}{4}\gamma_L^4 \tag{2.1.8}$$

$$\alpha_4 = -\left(\frac{4}{3}\gamma_L + \frac{7}{3}\gamma_L^2 + \frac{7}{6}\gamma_L^3 + \frac{1}{6}\gamma_L^4\right)$$

$$\alpha_5 = \frac{1}{4}\gamma_L + \frac{11}{24}\gamma_L^2 + \frac{1}{4}\gamma_L^3 + \frac{1}{24}\gamma_L^4$$

Note that $A_\alpha^{(L)}\mathbf{v}$ gives a vector whose components are the extrapolated value of $\mathbf{v}$ at $x = \Gamma_L$ (i.e., $v_{\Gamma_L}(t)$), to fifth order accuracy; while $A_e^{(L)}\mathbf{v}$ gives a vector whose components represents $(\partial^5 v_1/\partial x^5)h^5$. Since c_L (see 2.1.6) is of order unity, then $A_L\mathbf{v} = (A_\alpha^{(L)} + c_L A_e^{(L)})\mathbf{v}$ represents an extrapolation of $\mathbf{v}$ to v_{Γ_L} to fifth order.

Before using A_L in (2.1.2) we define τ_L:

$$\tau_L = \frac{1}{12h^2}\mathrm{diag}[\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, 0, \ldots, 0] \tag{2.1.9}$$

where

$$\begin{aligned} \tau_1 &= 71/2\alpha_1 \\ \tau_2 &= (-94 - \alpha_2\tau_1)/\alpha_1 \\ \tau_3 &= (113 - \alpha_3\tau_1)/\alpha_1 \\ \tau_4 &= (-56 - \alpha_4\tau_1)/\alpha_1 \\ \tau_5 &= (11 - \alpha_5\tau_1)/\alpha_1 \end{aligned} \tag{2.1.10}$$

The right boundary treatment is constructed in a similar fashion, and the formulae corresponding to (2.1.5) - (2.1.11) become:

$$A_R = A_\alpha^{(R)} + c_R A_e^{(R)}, \tag{2.1.11}$$

$$A_\alpha^{(R)} = \begin{bmatrix} 0 & \ldots & \ldots & 0 & 0 & \alpha_{N-4} & \alpha_{N-3} & \alpha_{N-2} & \alpha_{N-1} & \alpha_N \\ 0 & \ldots & \ldots & 0 & 0 & \alpha_{N-4} & \alpha_{N-3} & \alpha_{N-2} & \alpha_{N-1} & \alpha_N \\ \vdots & & & & & & & & & \\ 0 & \ldots & \ldots & 0 & 0 & \alpha_{N-4} & \alpha_{N-3} & \alpha_{N-2} & \alpha_{N-1} & \alpha_N \end{bmatrix}, \tag{2.1.12}$$

$$c_R = \mathrm{diag}[0, 0, \ldots, 0, \quad -20\alpha_N/71] \tag{2.1.13}$$

$$A_e^{(R)} = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 & -5 & 10 & -10 & 5 & -1 \\ 0 & 0 & \dots & 0 & 1 & -5 & 10 & -10 & 5 & -1 \\ \vdots & & & & & & & & & \\ 0 & 0 & \dots & 0 & 1 & -5 & 10 & -10 & 5 & -1 \end{bmatrix} \tag{2.1.14}$$

The α's here are:

$$\begin{aligned} \alpha_N &= 1 + \frac{25}{12}\gamma_R + \frac{35}{24}\gamma_R^2 + \frac{5}{12}\gamma_R^3 + \frac{1}{24}\gamma_R^4 \\ \alpha_{N-1} &= -\left(4\gamma_R + \frac{13}{3}\gamma_R^3 + \frac{3}{2}\gamma_R^3 + \frac{1}{6}\gamma_R^4\right) \\ \alpha_{N-2} &= 3\gamma_R + \frac{19}{4}\gamma_R^2 + 2\gamma_R^3 + \frac{1}{4}\gamma_R^4 \\ \alpha_{N-3} &= -\left(\frac{4}{3}\gamma_R + \frac{7}{3}\gamma_R^2 + \frac{7}{6}\gamma_R^3 + \frac{1}{6}\gamma_R^4\right) \\ \alpha_{N-4} &= \frac{1}{4}\gamma_R + \frac{11}{24}\gamma_R^2 + \frac{1}{4}\gamma_R^3 + \frac{1}{24}\gamma_R^4, \end{aligned} \tag{2.1.15}$$

$$\tau_R = \frac{1}{12h^2}\mathrm{diag}[0, \dots, \tau_{N-4}, \tau_{N-3}, \tau_{N-2}, \tau_{N-1}, \tau_N], \tag{2.1.16}$$

$$\begin{aligned} \tau_N &= 71/2\alpha_N \\ \tau_{N-1} &= (-94 - \alpha_{N-1}\tau_N)/\alpha_N \\ \tau_{N-2} &= (113 - \alpha_{N-2}\tau_N)/\alpha_N \\ \tau_{N-3} &= (-56 - \alpha_{N-3}\tau_N)/\alpha_N \\ \tau_{N-4} &= (11 - \alpha_{N-4}\tau_N)/\alpha_N \end{aligned} \tag{2.1.17}$$

Note that the extrapolating errors T_L and T_R are only $O(h^5)$. Therefore the scheme near the boundaries is third order accurate.

In order to show that the error estimate (1.1.13) is valid, i.e. the error is bounded by a "*constant*", it is sufficient to prove that $\frac{1}{2}(M + M^T)$ is negative definite, and its eigenvalues are bounded away from zero. The proof is done by writing $\frac{1}{2}(M + M^T)$ as a sum of five matrices. One matrix is negative definite and the other are non-positive

ones. The details are given in the appendix to this chapter, section 2.4, where it is proven that $\frac{1}{2}(M + M^T)$ is indeed negative definite, and its eigenvalues are bounded away from zero by $(-\pi^2/24)$, even as $N \to \infty$.

In the 2-D case the mesh size h becomes Δx or Δy respectively, when used in conjunction with $\mathcal{D}^{(x)}$ or $\mathcal{D}^{(y)}$.

2.2 Numerical example

In this section we describe numerical results for the following problem:

$$\frac{\partial u}{\partial t} = \nu(u_{xx} + u_{yy}) + f(x, y, t), \qquad (x, y) \in \Omega, \quad t > 0, \tag{2.2.1}$$

where Ω is the region contained between a circle of radius $r_o = 1/2$ and inner circle of radius $r_i \leq 0.1$. The inner circle is *not concentric* with the outer one. Specifically Ω is described by

$$\left\{(x - .5)^2 + (y - .5)^2 \leq 1/4\right\} \cap \left\{(x - .6)^2 + (y - .5)^2 \geq (.1 - \delta)^2; 0 \leq \delta \leq .1\right\} \tag{2.2.2}$$

The geometry thus looks as follows:

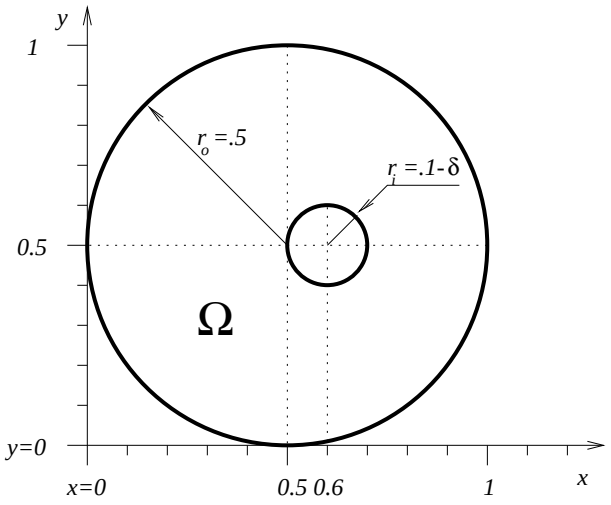

Figure 2.2:

The Cartesian grid in which Ω is embedded spans $0 \leq x, y \leq 1$. We took $\Delta x = \Delta y$, and ran several cases with $\Delta x = 1/50,\ 1/75,\ 1/100$.

The source function $f(x,y,t)$ was chosen different from zero so that we could assign an exact analytic solution to (2.2.1). This enables one to compute the error $E_{ij} = U_{ij} - V_{ij}$ "exactly" (to machine accuracy). We chose $\nu = 1$ and

$$u(x,y,t) = 1 + \cos(10t - 10x^2 - 10y^2) \tag{2.2.3}$$

This leads to

$$\begin{aligned} f(x,y,t) &= 400(x^2+y^2)\cos(10t - 10x^2 - 10y^2) \\ &- 50\sin(10t - 10x^2 - 10y^2) \end{aligned} \tag{2.2.4}$$

From the expression for $u(x,y,t)$ one obtains the boundary and initial conditions.

The problem (2.2.1), (2.2.2), (2.2.4) was solved using both a "standard" fourth order algorithm and the new "SAT", or "bounded error", see footnote to equation (1.2.11). The temporal advance was via a fourth order Runge-Kutta.

The standard algorithm was run for $\Delta x = 1/50$ and a range of $0 \le \delta < .01$ ($.09 < r_i \le .1$). We found that for $\delta \ge .0017323$, the runs were stable and the error bounded for "long" times (10^5 time steps, or equivalently $t = 2$). For $0 \le \delta < .0017233$ the results began to diverge exponentially from the analytic solution. The "point of departure" depended on δ. A discussion of these results is deferred to the next section. Figures 2.3 to 2.5 show the L_2-norm of the error vs. time for different radii of the inner "hole".

The same configurations were also run using the "bounded error" algorithm, and the results are shown in figures 2.6 to 2.9. It is seen that for δ's for which the standard methods fails, the new algorithm still has a bounded error, as predicted by the theory.

To check on the order of accuracy, the "SAT" runs (with $\delta = 0$) were repeated for $\Delta x = \Delta y = 1/75$ and $1/100$. Figure 2.10, 2.11, and 2.12 show the logarithmic slope of the L_2, L_1 and L_∞ errors to be less than -4; i.e., we indeed have a 4^{th} order method. That the slopes are larger in magnitude than 4.5 is attributed to the fact that as $\Delta x = \Delta y$ decreases the percentage of "internal" points increases (the boundary points have formally only 3^{rd} oder accuracy). It is therefore possible that if the number of grid points was increased much further, the slope would tend to -4. Lack of computer resources prevented checking this point further. (For $\Delta x = 0.01$, running 20,000 time steps, $t = 1.0$, the cpu time on a CRAY YMP is about 5 hours). It should also be noted

that the "bounded-error" algorithm was run with a time step, Δt, twice as large as the one used in the standard scheme. At this larger Δt the standard scheme "explodes" immediately.

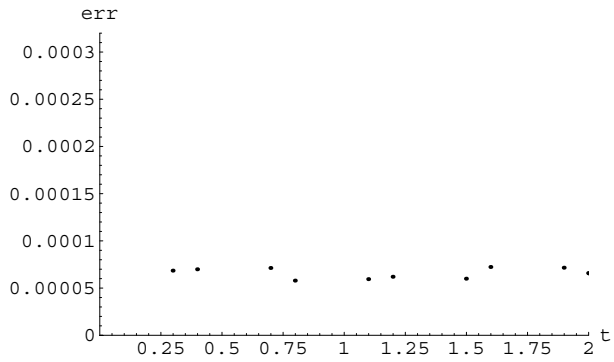

Figure 2.3: $\delta = 0.0017325$, Standard scheme

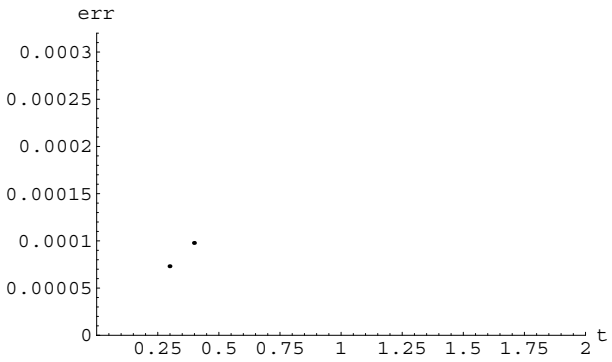

Figure 2.4: $\delta = 0.0017323$, Standard scheme

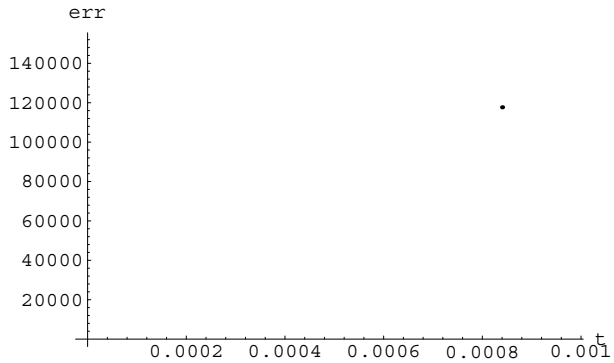

Figure 2.5: $\delta = 0.0015$, Standard scheme

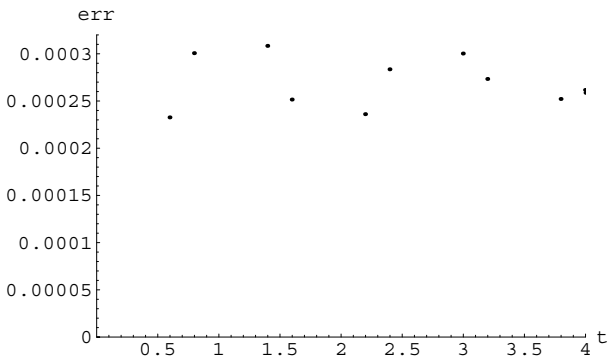

Figure 2.6: $\delta = 0$, SAT scheme

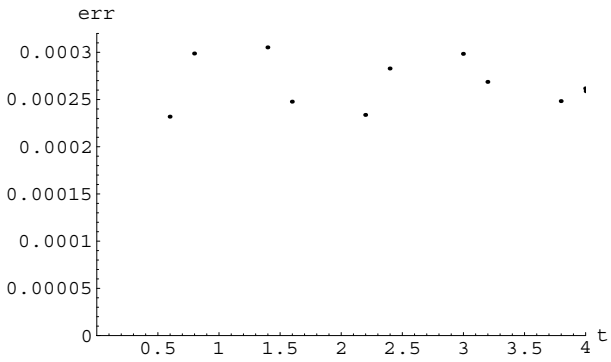

Figure 2.7: $\delta = 0.0015$, SAT scheme

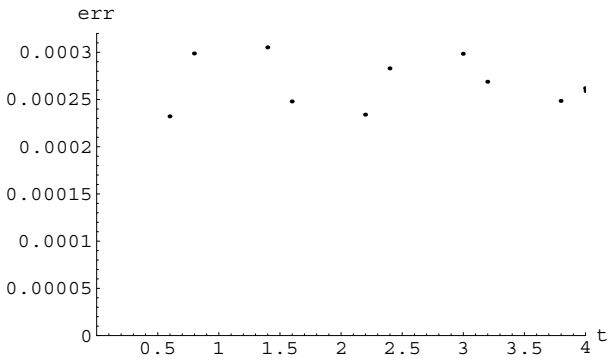

Figure 2.8: $\delta = 0.0017323$, SAT scheme

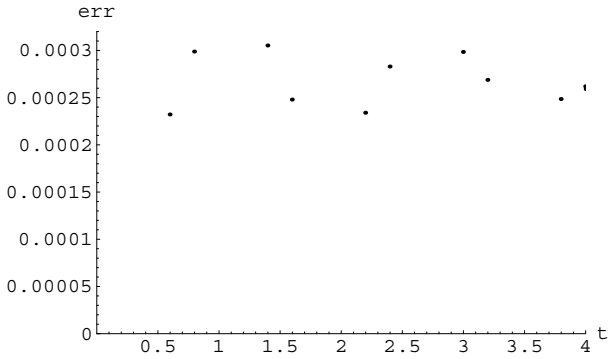

Figure 2.9: $\delta = 0.0017325$, SAT scheme

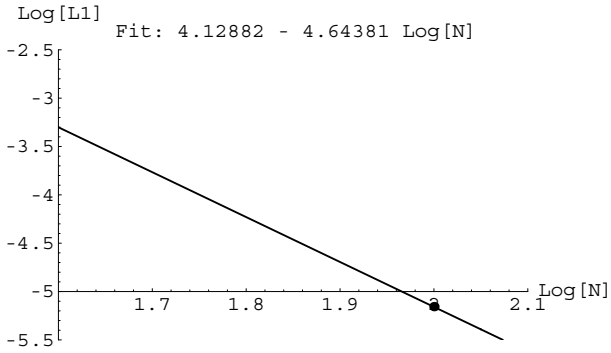

Figure 2.10: Order of accuracy L_1

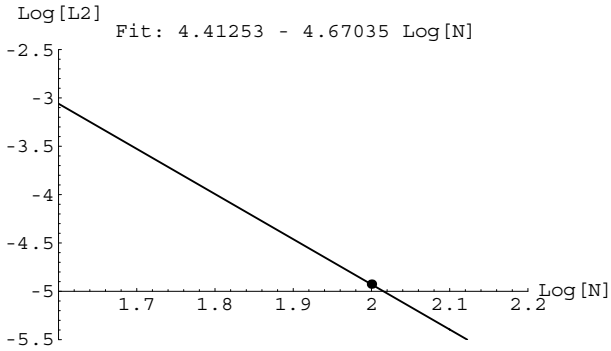

Figure 2.11: Order of accuracy L_2

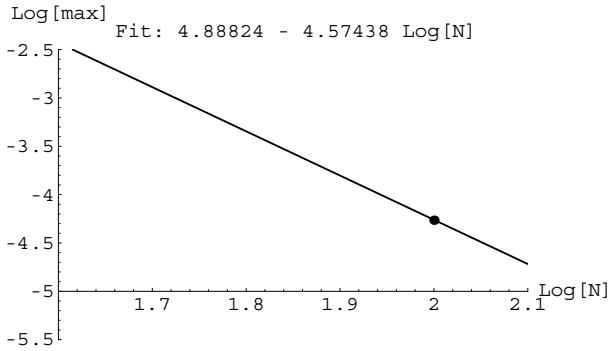

Figure 2.12: Order of accuracy L_∞

A study of the effect of the size of Δt shows that the instabilities exhibited above by the "standard" scheme are due to the time-step being near the C.F.L.-limit. The reason for this strong dependence on the geometry is that the "standard" differentiation matrix has entries which are $O\left(1/(h^2\gamma)\right)$. In the "SAT" differentiation matrix the entries are $O\left(1/(h^2(1+\gamma))\right)$. This instability problem could probably be solved also by using an

implicit method for the time integration.

2.3 Conclusions

(i) The theoretical results show that one has to be very careful when using an algorithm whose differentiation matrix, or rather its symmetric part, is not negative definite. For some problems, such "standard" schemes will give good answers (i.e., bounded errors) and for others instability will set in. Thus, for example, the "standard" scheme for the 1-D case has a matrix which, for all $0 < \gamma_L, \gamma_R < 1$, though not negative definite has eigenvalues with negative real parts. This assures, in the 1-D case, the error boundness. In the 2-D case, even though each of the block sub-matrices of the $\ell \times \ell$ x-and-y differentiation matrices has only negative (real-part) eigenvalues, it is not assured that the sum of the two $\ell \times \ell$ matrices will have this property. This depends, among other things, on the shape of the domain and the mesh size (because the mesh size determines, for a given geometry, the γ_L and γ_R's along the boundaries).

Thus we might have the "paradoxical" situation, that for a given domain shape, successive mesh refinement could lead to instability due to the occurrence of destabilizing γ's. This cannot happen if one constructs, as was done here, a scheme whose differentiation matrices have symmetric parts that are negative definite.

It is also interesting to note that if one uses explicit standard method then the allowable C.F.L. may decrease extremely rapidly with change in the geometry that causes decrease in the γ's. This point is brought out in figures 2.3 to 2.5.

(ii) Note that the construction of the 2-D algorithm, and its analysis, which were based on the 1-D case, can be extended in a similar (albeit more complex) fashion to higher dimensions.

(iii) Also note that if the diffusion coefficient ν, in the equation

$$u_t = \nu \nabla^2 u$$

is a function of the spatial coordinates, $\nu = \nu(x, y, z)$, the previous analysis goes

through but the energy estimate for the error is now for a different, but equivalent norm.

2.4 Appendix

Using D given in (2.1.3) and A_L, τ_L, A_R, τ_R given in equations (2.1.5) to (2.1.18) we are now ready to construct

$$\begin{aligned}\frac{1}{2}(M+M^T) = \frac{1}{2}\{D+D^T \quad &- \quad [\tau_L(A_\alpha^{(L)} + c_L A_e^{(L)}) + \tau_R(A_\alpha^{(R)} + c_R A_e^{(R)})] \\ &- \quad [\tau_L(A_\alpha^{(L)} + c_L A_e^{(L)}) + \tau_R(A_\alpha^{(R)} + c_R A_e^{(R)})]^T\}\end{aligned} \tag{2.4.1}$$

Upon using equations (2.1.3)-(2.1.18) in (2.4.1) one gets:

$$\frac{M+M^T}{2} =$$

$$\frac{1}{24h^2}\begin{bmatrix}
 & & & & & & 0 & & & & & & & & \\
 & & W^{(L)} & & & & 0 & & & & & & & & \\
 & & & & & & -2 & 0 & & & & & & & \\
 & & & & & & 32 & -2 & & & & & 0 & & \\
0 & \dots & 0 & -2 & 32 & -60 & 32 & -2 & & & & & & & \\
 & & & & -2 & 32 & -60 & 32 & -2 & & & & & & \\
 & & & & & -2 & 32 & -60 & 32 & -2 & & & & & \\
 & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \\
 & & & & & & & -2 & 32 & -60 & 32 & -2 & 0 & \dots & 0 \\
 & & & 0 & & & & & -2 & 32 & & & & & \\
 & & & & & & & & & -2 & & & & & \\
 & & & & & & & & & 0 & & & W^{(R)} & & \\
 & & & & & & & & & 0 & & & & &
\end{bmatrix}$$

(2.4.2)

where $W^{(L)}$ and $W^{(R)}$ are 6×6 blocks given by:

$$W^{(L)} = W_1^{(L)} + W_2^{(L)} \tag{2.4.3}$$

$$W^{(R)} = W_1^{(R)} + W_2^{(R)} \tag{2.4.4}$$

$$W_{1_{ij}}^{(L)} = \left\{ \begin{array}{ll} 0 & i=1 \text{ or } j=1 \\ & \\ -(\alpha_i\tau_j + \alpha_j\tau_i) & i,j \neq 1 \end{array} \right\} \qquad 1 < i,j < 5 \tag{2.4.5}$$

$$W_{1_{ij}}^{(L)} = \left\{ \begin{array}{ll} 0 & i=N \text{ or } j=N \\ & \\ -(\alpha_{N-i}\tau_{N-j} + \alpha_{N-j}\tau_{N-i}) & \end{array} \right\} \qquad 0 \leq N-i, N-j \leq 4 \tag{2.4.6}$$

$$W_2^{(L)} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -30 & 12 & 13 & -6 & 1 \\ 0 & 12 & -60 & 32 & -2 & 0 \\ 0 & 13 & 32 & -60 & 32 & -2 \\ 0 & -6 & -2 & 32 & -60 & 32 \\ 0 & 1 & 0 & -2 & 32 & -60 \end{bmatrix} \tag{2.4.7}$$

$$W_2^{(R)} = \begin{bmatrix} -60 & 32 & -2 & 0 & 1 & 0 \\ 32 & -60 & 32 & -2 & -6 & 0 \\ -2 & 32 & -60 & 32 & 13 & 0 \\ 0 & -2 & 32 & -60 & 12 & 0 \\ 1 & -6 & 13 & 12 & -30 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{2.4.8}$$

The next task is to show that $\tilde{M} = \frac{1}{2}(M + M^T)$ is negative definite. We write the symmetric matrix $\tilde{M}$ as a sum of five symmetric matrices,

$$\tilde{M} = \frac{1}{24h^2}\left[\beta_0\tilde{M}_1 + 2\tilde{M}_2 + (24-\beta_0)\tilde{M}_3 + \tilde{M}_4 + \tilde{M}_5\right]. \tag{2.4.9}$$

We shall show that $\tilde{M}_1$ is negative definite, and that $\tilde{M}_j (j = 2, \ldots, 5)$ are non-positive definite. The $\tilde{M}$'s are given by

$$\tilde{M}_1 = \begin{bmatrix} -\frac{1}{2\beta_0} & 0 & 0 & & & & & & \\ 0 & -2 & 1 & 0 & 0 & & & & \\ 0 & 1 & -2 & 1 & 0 & & & & \\ 0 & 0 & 1 & -2 & 1 & & & & \\ 0 & 0 & 0 & 1 & -2 & 1 & & & \\ & & & & \ddots & \ddots & \ddots & & \\ & & & & & 1 & -2 & 1 & 0 \\ & & & & & & 1 & -2 & 0 \\ & & & & & & 0 & 0 & -\frac{1}{2\beta_0} \end{bmatrix} = M_1^L + \hat{M}_1 + M_1^R \tag{2.4.10}$$

where

$$M_1^L = \begin{bmatrix} -1/2\beta_0 & 0 & & \\ 0 & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix}, M_2^R = \begin{bmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & 0 \\ & & 0 & -1/2\beta_0 \end{bmatrix}$$

and $\hat{M}_1$ is the remaining $(N-2)\times(N-2)$ middle block.

$$\tilde{M}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & & & & & & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & & & & & & & \\ 0 & 0 & 0 & 0 & 2 & -5 & 4 & -1 & & & & & & \\ & & & & -1 & 4 & -6 & 4 & -1 & & & & & \\ & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \\ & & & & & & -1 & 4 & -6 & 4 & -1 & & & \\ & & & & & & & -1 & 4 & -5 & 2 & 0 & 0 & 0 \\ & & & 0 & & & & & -1 & 2 & -1 & 0 & 0 & 0 \\ & & & & & & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.4.11}$$

$$\tilde{M}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & & & & 0 & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & & & & & & \\ & & & & & & 1 & -2 & 1 & & & & & \\ & & & & & & & & \ddots & \ddots & \ddots & & & \\ & & & & & & & & & 1 & -2 & 1 & & \\ & & & & & & & & & & 1 & -1 & 0 & 0 \\ & & & 0 & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & 0 & 0 & 0 \end{bmatrix} \tag{2.4.12}$$

$$\tilde{M}_4 = \left[\begin{array}{c|c|c|c|c|c} -1/2 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & \begin{array}{c} -30+2\beta \\ -2\alpha_2\tau_2 \end{array} & \begin{array}{c} 12-\beta \\ -(\alpha_2\tau_3+\alpha_3\tau_2) \end{array} & \begin{array}{c} 13 \\ -(\alpha_2\tau_4+\alpha_4\tau_2) \end{array} & \begin{array}{c} -6 \\ -(\alpha_2\tau_5+\alpha_5\tau_2) \end{array} & 1 \\ \hline 0 & \begin{array}{c} 12-\beta \\ -(\alpha_2\tau_3+\alpha_3\tau_2) \end{array} & \begin{array}{c} -60+2\beta \\ -2\alpha_3\tau_3 \end{array} & \begin{array}{c} 32-\beta \\ -(\alpha_3\tau_4+\alpha_4\tau_3) \end{array} & \begin{array}{c} -2 \\ -(\alpha_3\tau_5+\alpha_5\tau_3) \end{array} & 0 \\ \hline 0 & \begin{array}{c} 13 \\ -(\alpha_2\tau_4+\alpha_4\tau_2) \end{array} & \begin{array}{c} 32-\beta \\ -(\alpha_3\tau_4+\alpha_4\tau_3) \end{array} & \begin{array}{c} -60+2\beta \\ -2\alpha_4\tau_4 \end{array} & \begin{array}{c} 32-\beta \\ -(\alpha_4\tau_5+\alpha_5\tau_4) \end{array} & -2 \\ \hline 0 & \begin{array}{c} -6 \\ -(\alpha_2\tau_5+\alpha_5\tau_2) \end{array} & \begin{array}{c} -2 \\ -(\alpha_3\tau_5+\alpha_5\tau_3) \end{array} & \begin{array}{c} 32-\beta \\ -(\alpha_4\tau_5+\alpha_5\tau_4) \end{array} & \begin{array}{c} -58+\beta \\ -2\alpha_5\tau_5 \end{array} & 28-\beta \\ \hline 0 & 1 & 0 & -2 & 28-\beta & -26+\beta \end{array}\right] \tag{2.4.13}$$

$$\tilde{M}_5 = \left[\begin{array}{c|c|c|c|c|c} -26+\beta & 28-\beta & -2 & 0 & 1 & \\ & & & & & 0 \\ \hline 28-\beta & \begin{array}{c}-58+2\beta\\ -2\alpha_{N-4}\tau_{N-4}\end{array} & \begin{array}{c}32-\beta\\ -(\alpha_{N-3}\tau_{N-4}\\ +\alpha_{N-4}\tau_{N-3})\end{array} & \begin{array}{c}-2\\ -(\alpha_{N-2}\tau_{N-4}\\ +\alpha_{N-4}\tau_{N-2})\end{array} & \begin{array}{c}-6\\ -(\alpha_{N-1}\tau_{N-4}\\ +\alpha_{N-4}\tau_{N-1})\end{array} & 0 \\ \hline -2 & \begin{array}{c}32-\beta\\ -(\alpha_{N-3}\tau_{N-4}\\ +\alpha_{N-4}\tau_{N-3})\end{array} & \begin{array}{c}-60+2\beta\\ -2\alpha_{N-3}\tau_{N-3}\end{array} & \begin{array}{c}32-\beta\\ -(\alpha_{N-2}\tau_{N-3}\\ +\alpha_{N-3}\tau_{N-2})\end{array} & \begin{array}{c}13\\ -(\alpha_{N-1}\tau_{N-3}\\ +\alpha_{N-3}\tau_{N-1})\end{array} & 0 \\ \hline 0 & \begin{array}{c}-2\\ -(\alpha_{N-2}\tau_{N-4}\\ +\alpha_{N-4}\tau_{N-2})\end{array} & \begin{array}{c}32-\beta\\ -(\alpha_{N-2}\tau_{N-3}\\ +\alpha_{N-3}\tau_{N-2})\end{array} & \begin{array}{c}-60+2\beta\\ -2\alpha_{N-2}\tau_{N-2}\end{array} & \begin{array}{c}12-\beta\\ -(\alpha_{N-1}\tau_{N-2}\\ +\alpha_{N-2}\tau_{N-1})\end{array} & 0 \\ \hline 1 & \begin{array}{c}-6\\ -(\alpha_{N-1}\tau_{N-4}\\ +\alpha_{N-4}\tau_{N-1})\end{array} & \begin{array}{c}13\\ -(\alpha_{N-1}\tau_{N-3}\\ +\alpha_{N-3}\tau_{N-1})\end{array} & \begin{array}{c}12-\beta\\ -(\alpha_{N-1}\tau_{N-2}\\ +\alpha_{N-2}\tau_{N-1})\end{array} & \begin{array}{c}-30+2\beta\\ -2\alpha_{N-1}\tau_{N-1}\end{array} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & -1/2 \end{array}\right]. \tag{2.4.14}$$

Let us consider $\hat{M}_1$ - see (2.4.10); it may be decomposed as follows:

$$\hat{M}_1 = -\begin{bmatrix} 1 & -1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & -1 \\ & & & & 1 \end{bmatrix}\begin{bmatrix} 1 & & & & \\ -1 & \ddots & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & & & & \\ & \ddots & & & \\ & & 0 & & \\ & & & \ddots & \\ & & & & -1 \end{bmatrix} \tag{2.4.15}$$

The last matrix in non-positive definite. The first term is a product of a regular matrix with its transpose, hence its negative is a negative definite matrix. Thus we established that $\hat{M}_1$ is negative definite for any finite dimension N. All its eigenvalues are negative. It remains to show that the eigenvalues of $\tilde{M}_1/h^2$ (see (2.4.9)) are bounded away from zero by a constant as $h \to 0$ $(N \to \infty)$.

Consider a symmetric tridiagonal matrix S with, like $\hat{M}_1$, constant diagonals:

$$S = \begin{bmatrix} b & a & 0 & & & \\ a & b & a & & & \\ 0 & a & b & a & & \\ & \ddots & \ddots & \ddots & & \\ & & & a & b & a \\ & & & & a & b \end{bmatrix}. \tag{2.4.16}$$

Designate by D_j the determinant of the upper-left $j \times j$ sub-matrix. Thus $D_1 = b$, $D_2 = \det \begin{bmatrix} b & a \\ a & b \end{bmatrix}$, etc.

We have then $D_1 = b$, $D_2 = b^2 - a^2$ and in general

$$D_j = bD_{j-1} - a^2 D_{j-2}, \qquad (3 \leq j \leq N) \tag{2.4.17}$$

It can be shown that the solution to the recursion relation (2.4.17) is

$$D_j = -\frac{1}{a^2}\left[\frac{A}{\mu_1^j} + \frac{B}{\mu_2^j}\right] \tag{2.4.18}$$

where

$$\mu_1 = \frac{1}{2a^2}\left[b + \sqrt{b^2 - 4a^2}\right] \tag{2.4.19}$$

$$\mu_2 = \frac{1}{2a^2}\left[b - \sqrt{b^2 - 4a^2}\right] \tag{2.4.20}$$

$$A = \frac{1}{\mu_1 - \mu_2}\left[(D_2 - bD_1)\,\mu_1 + D_1\right] \tag{2.4.21}$$

$$B = \frac{1}{\mu_1 - \mu_2}\left[(D_2 - bD_1)\,\mu_2 + D_1\right] \tag{2.4.22}$$

We have already shown that $\tilde{M}_1$ is negative definite. The eigenvalues of $\hat{M}_1$ are found from

$$\det(\tilde{M}_1 - I\lambda) = \left(-\frac{1}{2\beta_0} - \lambda\right) \cdot \det(\hat{M}_1 - \lambda I) \cdot \left(-\frac{1}{2\beta_0} - \lambda\right) = 0 \tag{2.4.23}$$

thus either $\lambda = -1/2\beta_0 < 0$ (because β_0 will be taken positive) or $\lambda =$ eigenvalue of $\hat{M}_1 < 0$. We would like to investigate the behavior of the eigenvalues of $\frac{\beta_0}{24h^2}\tilde{M}_1$. In particular we would like to show that these eigenvalues (which are negative) are bounded

away from zero. To show this we analyze the behavior of $\hat{M}_1 - \lambda I$ as N increases. We now take $S = \hat{M}_1 - \lambda I$. Its determinant is given by D_{N-2}. Substituting (2.4.19)-(2.4.22) into (2.4.18) with $j = N-2$ we get after some elementary manipulations

$$D_{N-2} = \frac{2^{N-2}}{\rho r^{N-3}} \sin(N-1)\theta \tag{2.4.24}$$

where

$$\begin{aligned} \rho &= \sqrt{4-b^2}; \;\; b = -2-\lambda; \;\; a = 1 \\ r &= \sqrt{b^2+\rho^2} = 2 \\ \theta &= \tan^{-1}(\rho/b) \end{aligned} \tag{2.4.25}$$

From (2.4.23) we require

$$D_{N-2} = 0 \tag{2.4.26}$$

This is equivalent, see (2.4.24), to requiring

$$\theta = \frac{k\pi}{N-1}, \qquad k = 1, \ldots, N-2. \tag{2.4.27}$$

From the definition of θ and (2.4.25) we obtain

$$\tan\left(\frac{k\pi}{N-1}\right) = -\frac{\sqrt{-\lambda(\lambda+4)}}{2+\lambda}, \qquad (\lambda < 0). \tag{2.4.28}$$

Squaring (2.4.28) we get a quadratic equation for λ, the solution of which is

$$\begin{aligned} \lambda &= -2\left[1 \pm \left(1 + \tan^2\left(\frac{k\pi}{N-1}\right)\right)^{-1/2}\right] \\ &= -2\left[1 \pm \cos\left(\frac{k\pi}{N-1}\right)\right]. \end{aligned} \tag{2.4.29}$$

For any fixed N, the smallest values of $|\lambda|$ is given by (2.4.28) for $k = 1$,

$$\lambda_{\max} = -\min_k |\lambda| = -2\left[1 - \cos\left(\frac{\pi}{N-1}\right)\right]. \tag{2.4.30}$$

As N increases, we have

$$\begin{aligned} \lambda_{\max} &\to -2\left[1 - \left(1 - \frac{\pi^2}{2(N-1)^2} + O\left(\frac{1}{N^4}\right)\right)\right] \\ &= -\frac{\pi^2}{(N-1)^2} \approx -\pi^2 h^2. \end{aligned} \tag{2.4.31}$$

Thus the eigenvalues of $\hat{M}_1/24h^2$ (and hence of $\tilde{M}_1/24h^2$) are bounded away from zero by the value $-\left(\frac{\pi^2}{24}\right)$.

We now consider $\tilde{M}_2$. One can verify that

$$\tilde{M}_2 = -\hat{M}_2\hat{M}_2^T \tag{2.4.32}$$

where

$$\hat{M}_2 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & 0 & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & -2 & 1 & 0 & & & & & & \\
 & & & & & 0 & 1 & -2 & 1 & & & & & & \\
 & & & & & & & 1 & -2 & 1 & & & & & \\
 & & & & & & & & \ddots & \ddots & \ddots & & & & \\
 & & & & & & & & & 1 & -2 & 1 & 0 & & \\
 & & & 0 & & & & & & 0 & 1 & -2 & 0 & & \\
 & & & & & & & & & & 0 & 1 & 0 & 0 & 0 \\
 & & & & & & & & & & & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & 0 & 0 & 0 & 0
\end{bmatrix} \tag{2.4.33}$$

Therefore $\tilde{M}_2$ is non-positive definite. In a similar fashion $\tilde{M}_3$ is non-positive definite because

$$M_3 = -\hat{M}_3\hat{M}_3^T \tag{2.4.34}$$

with

$$\hat{M}_3 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & & & & & & & & & \\
 & & & & & & 1 & -1 & & & & & & & & \\
 & & & & & & & \ddots & \ddots & & & & & & & \\
 & & & & & & & & 1 & -1 & & & & & & \\
 & & & & & & & & & 1 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}. \tag{2.4.35}$$

The matrices $\tilde{M}_4$ and $\tilde{M}_5$ are $N \times N$ matrices with zero entries except for 6×6 upper-left (lower-right) blocks. It is sufficient to show that these blocks are negative definite. This was done symbolically using the Mathematica software and plotted for $0 \leq \gamma_L, \gamma_R < 1$ and $\beta_0 = 1$. $\tilde{M}_4$ and $\tilde{M}_5$ are indeed negative definite for, $0 \leq \gamma_R, \gamma_L < 1$. Thus we have shown that $\tilde{M} = \frac{1}{2}(M + M^T)$ is indeed negative definite, and its eigenvalues are bounded away from zero by $(-\pi^2/24)$, even as $N \to \infty$, and the error estimate (1.1.13) is valid.

Chapter 3

The advection diffusion equation

This chapter considers 2^{nd}-order accurate approximations to model linear advection-diffusion equations in one and more dimensions, on domains which may be irregular.

In section 3.1 we treat a model "shock-layer" equation (linearized Burger's equation),

$$u_t + a u_x = \frac{1}{R} u_{xx}; \quad t \geq 0, \;\; 0 < x < 1; \;\; R \gg 1.$$

We develop there a 2^{nd}-order one dimensional semi-discrete scheme using the methodology presented in chapter 1. By constructing the scheme this way we can be sure that the energy norm of the error to be temporally bounded for all $t > 0$ by a "constant" proportional to the norm of the truncation error and that this property is also valid for the multi-dimensional scheme.

Section 3.2 presents numerical results. Subsection 3.2.1 deals with the steady state solution to the "shock-layer" equation for a large range of the "Reynolds number", R. Oscillations that appear in the numerical solution when using a standard central finite-differencing, are eliminated (or dramatically reduced) when the bounded-error algorithm is used.

Subsection 3.2.2 considers steady-state solution to a two dimensional scalar model to the boundary layer equations,

$$u_t + a u_x + b u_y = \frac{1}{R} u_{yy}; \quad R \gg 1, \;\; b < 0,$$

both for rectangular and trapezoidal domains. Again, the bounded-error algorithm out-performs the standard scheme in ways described therein.

Subsection 3.2.3 presents a time dependent example, modeling a boundary-layer being excited sinusoidally,

$$u_t + au_x + bu_y = \frac{1}{R}u_{yy} + \sigma b \sin[k(x - at)].$$

Here, aside from the usual performance criteria, such as error-norms and quality of the velocity profiles, we see that the error-bounded algorithm also has a significantly smaller phase error.

3.1 Construction of the scheme

Consider the scalar advection-diffusion problem

$$\frac{\partial u}{\partial t} = a\frac{\partial u}{\partial x} + \frac{1}{R}\frac{\partial^2 u}{\partial x^2} + f(x,t); \quad \Gamma_L \le x \le \Gamma_R, \quad t \ge 0, \quad a > 0 \ ^{\dagger} \tag{3.1.1a}$$

$$u(x,0) = u_0(x) \tag{3.1.1b}$$

$$u(\Gamma_L,t) = g_L(t) \tag{3.1.1c}$$

$$u(\Gamma_R,t) = g_R(t)$$

and $f(x,t) \in \mathcal{C}^2$.

Let us discretize (3.1.1) spatially on the same grid as in chapters 1 and 2, and using the same notation for the numerical approximation and for the error vector. The scheme gets the form:

$$\frac{d\mathbf{v}}{dt} = M\mathbf{v} + \tau_L \mathbf{g}_L + \tau_R \mathbf{g}_R + \mathbf{f}(t) \tag{3.1.2a}$$

where

$$M = \frac{1}{R}M_P + aM_H = \frac{1}{R}(D_P - \tau_{L_P}A_{L_P} - \tau_{R_P}A_{R_P}) + a(D_H - \tau_{L_H}A_{L_H} - \tau_{R_H}A_{R_H}) \tag{3.1.2b}$$

This section is devoted to the task of constructing M in the case of $m = 2$, i.e., a second order accurate finite difference algorithm.

† The results for the case $a < 0$ are found by an analysis anologus to the one presented in this section, and are presented in the appendix to this chapter, section 3.4.

We shall deal separately with the hyperbolic and parabolic parts of the r.h.s. of (3.1.2b). The parabolic terms are given by:

$$D_P = \frac{1}{h^2}\begin{bmatrix} 1 & -2 & 1 & 0 & & & & & \\ 1 & -2 & 1 & 0 & & & & & \\ 0 & 1 & -2 & 1 & & & & & \\ 0 & 0 & 1 & -2 & 1 & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & 1 & -2 & 1 & 0 & 0 \\ & & & & & 1 & -2 & 1 & 0 \\ & & & & & & 1 & -2 & 1 \\ & & & & & & 1 & -2 & 1 \end{bmatrix}; \tag{3.1.3}$$

$$\tau_{L_P} = \frac{1}{h^2}\text{diag}\left[\tau_{L_1}^{(P)}, 0, \ldots 0\right] = \frac{1}{h^2}\text{diag}\left[\frac{4}{(2+\gamma_L)(1+\gamma_L)}, 0, \ldots, 0\right]; \tag{3.1.4}$$

$$\tau_{R_P} = \frac{1}{h^2}\text{diag}\left[0, 0, \ldots, \tau_{R_N}^{(P)}\right] = \frac{1}{h^2}\text{diag}\left[0, 0, \ldots, \frac{4}{(2+\gamma_R)(1+\gamma_R)}\right]; \tag{3.1.5}$$

$$A_{L_P} = \begin{bmatrix} \frac{1}{2}(2+\gamma_L)(1+\gamma_L) & -\gamma_L(2+\gamma_L) & \frac{1}{2}(\gamma_L+\gamma_L^2) & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{2}(2+\gamma_L)(1+\gamma_L) & -\gamma_L(2+\gamma_L) & \frac{1}{2}(\gamma_L+\gamma_L^2) & 0 & \ldots & 0 \end{bmatrix}; \tag{3.1.6}$$

$$A_{R_P} = \begin{bmatrix} 0 & \ldots & 0 & \frac{1}{2}(\gamma_R+\gamma_R^2) & -\gamma_R(2+\gamma_R) & \frac{1}{2}(2+\gamma_R)(1+\gamma_R) \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \ldots & 0 & \frac{1}{2}(\gamma_R+\gamma_R^2) & -\gamma_R(2+\gamma_R) & \frac{1}{2}(2+\gamma_R)(1+\gamma_R) \end{bmatrix}. \tag{3.1.7}$$

The hyperbolic terms are given by:

$$D_H = \frac{1}{2h}\left\{\begin{bmatrix} -2 & 2 & & & & & \\ -1 & 0 & 1 & & & & \\ & -1 & 0 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 0 & 1 & 0 \\ & & & & -1 & 0 & 1 \\ & & & & & -2 & 2 \end{bmatrix}\right.$$

$$
+\begin{bmatrix} c_1 & & & & & & \\ & c_2 & & & & & \\ & & c_3 & & & & \\ & & & \ddots & & & \\ & & & & c_{N-2} & & \\ & & & & & c_{N-1} & \\ & & & & & & c_N \end{bmatrix}
\begin{bmatrix} -1 & 2 & -1 & & & & \\ 0 & -1 & 2 & -1 & & & \\ 1 & -2 & 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 0 & 2 & -1 \\ & & & 1 & -2 & 1 & 0 \\ & & & & 1 & -2 & 1 \end{bmatrix}
$$

$$
+2h\tilde{c}\begin{bmatrix} 0 & -1 & 1 & & & & & \\ 1 & -1 & -1 & 1 & & & & \\ 1 & -1 & 0 & -1 & 1 & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & 1 & -1 & 0 & -1 & 1 & \\ & & & & 1 & -1 & -1 & 1 \\ & & & & & 1 & -1 & 0 \end{bmatrix} \Bigg\} \tag{3.1.8}
$$

where

$$
c_k = \frac{1}{N-1}[(c_N - c_1)k + (Nc_1 - c_N)], \tag{3.1.9}
$$

and

$$
\tilde{c} = \frac{1}{2}(c_1 - c_N). \tag{3.1.10}
$$

We note that the first term in (3.1.8) is the usual second order differentiation matrix, The additional matrices may be viewed as *connectivity terms* that allow the non-positive definite property to be maintained as we go from one corner to the other. In the parabolic case the need did not arise because there are penalty terms available at both corners. These terms are $O(h^2)$, thus maintaining accuracy. One effect of using these connectivity terms is that the 'core' of D_H is not Toeplitz any more.

For $a > 0$ in (3.1.1a), the left boundary is, for the hyperbolic part, an "outflow" boundary on which we do not prescribe a "hyperbolic boundary condition", therefore, in this case $\tau_{L_H} = 0$. When $a < 0$, then $\tau_{R_H} = 0$ – see the Appendix for details.

Here, with $a > 0$,

$$
\tau_{R_H} = \text{ diag } [0, 0, \ldots, \tau^{(H)}_{R_{N-1}}, \tau^{(H)}_{R_N}] \tag{3.1.11}
$$

and

$$
A_{R_H} = \begin{bmatrix} 0 & & & \\ & \ddots & 0 & \\ & 0 & & \\ 0 & & -\gamma_R & 1+\gamma_R \\ & & -\gamma_R & 1+\gamma_R \end{bmatrix} \tag{3.1.12}
$$

Next we shall show that the parabolic part of M is negative definite. The symmetric part of M_P, $\tilde{M}_P = \frac{1}{2}(M_P + M_P^T)$, is found using equations (3.1.3) to (3.1.7), to be

$$\tilde{M}_P = \frac{1}{2h^2}\begin{bmatrix}
-2 & \frac{3\gamma_L-1}{\gamma_L+1} & \frac{2-\gamma_L}{2+\gamma_L} & & & & & & \\
\frac{3\gamma_L-1}{\gamma_L+1} & -4 & 2 & & & & & 0 & \\
\frac{2-\gamma_L}{2+\gamma_L} & 2 & -4 & 2 & & & & & \\
 & & 2 & -4 & 2 & & & & \\
 & & & \ddots & \ddots & \ddots & & & \\
 & & & & 2 & -4 & 2 & & \\
 & 0 & & & & 2 & -4 & 2 & \frac{2-\gamma_R}{2+\gamma_R} \\
 & & & & & & 2 & -4 & \frac{3\gamma_R-1}{\gamma_R+1} \\
 & & & & & & \frac{2-\gamma_R}{2+\gamma_R} & \frac{3\gamma_R-1}{\gamma_R+1} & -2
\end{bmatrix} \tag{3.1.13}$$

We now decompose $\tilde{M}_P$ as follows:

$$\tilde{M}_P = \frac{1}{2h^2}\left\{ \alpha \begin{bmatrix}
-4 & 2 & & & & & \\
2 & -4 & 2 & & & & \\
 & 2 & -4 & 2 & & & \\
 & & \ddots & \ddots & \ddots & & \\
 & & & 2 & -4 & 2 & \\
 & & & & 2 & -4 & 2 \\
 & & & & & 2 & -4
\end{bmatrix} + (1-\alpha) \begin{bmatrix}
0 & 0 & 0 & & & & & \\
0 & 0 & 0 & & & & & \\
0 & 0 & -2 & 2 & & & & \\
 & & 2 & -4 & 2 & & & \\
 & & & \ddots & \ddots & \ddots & & \\
 & & & & 2 & -4 & 2 & 0 \\
 & & & & 0 & 2 & -2 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & 0
\end{bmatrix} \right.$$

$$
+\left\{\begin{bmatrix}
-2(1-2\alpha) & \frac{3\gamma_L-1}{\gamma_L+1}-2\alpha & \frac{2-\gamma_L}{2+\gamma_L} & & & \\
\frac{3\gamma_L-1}{\gamma_L+1}-2\alpha & -4(1-\alpha) & 2(1-\alpha) & & 0 & \\
\frac{1-\gamma_L}{2+\gamma_L} & 2(1-\alpha) & -2(1-\alpha) & & & \\
 & & & 0 & & \\
 & & & -2(1-\alpha) & 2(1-\alpha) & \frac{2-\gamma_R}{2+\gamma_R} \\
 & 0 & & 2(1-\alpha) & -4(1-\alpha) & \frac{3\gamma_R-1}{\gamma_R+1}-2\alpha \\
 & & & \frac{2-\gamma_R}{2+\gamma_R} & \frac{3\gamma_R-1}{\gamma_R+1}-2\alpha & -2(1-2\alpha)
\end{bmatrix}\right\}
$$

(3.1.14)

We look for $1 > \alpha > 0$ such that the second and third matrices in (3.1.14) are non-positive definite. The first matrix in (3.1.14) is already negative definite by the argument leading to eq. (2.4.31), in the appendix to chapter 2. By the same argument it immediately follows that its largest eigenvalue is smaller than $-\alpha\pi^2$. For $0 < \alpha < 1$, the second matrix in (3.1.14) is non-positive definite, see eq. (2.4.34) and (2.4.35) in that appendix. The third matrix in (3.1.14) has two square 3×3 corners which are negative for $0 < \alpha < .275$. This completes the proof that $\tilde{M}_P$ is indeed negative definite (for a certain range of α).

Next we would like to show that $\tilde{M}_H = \frac{1}{2}(M_H + M_H^T)$ is non-positive definite. Using equations (3.1.8)–(3.1.12) we have

$$\tilde{M}_H = \frac{1}{4h}\begin{bmatrix} -4-2c_1 & 1+2c_1 & 0 & & & \\ 1+2c_1 & -2c_1 & 0 & & & \\ & & & & & 0 \\ 0 & 0 & 0 & & & \\ & & & \ddots & & \\ & 0 & & 2c_N + 2\gamma_R\tau_{N-1}^{(H)} & -1-2c_N-(1+\gamma_R)\tau_{N-1}^{(H)}+\gamma_R\tau_N^{(H)} & \\ & & & -1-2c_N-(1+\gamma_R)\tau_{N-1}^{(H)}+\gamma_R\tau_N^{(H)} & 4+2c_N-2(1+\gamma_R)\tau_N^{(H)} & \end{bmatrix}$$

(3.1.15)

We now write $\tilde{M}_H$ as the sum of three "corner-matrices",

$$\tilde{M}_H = \frac{1}{4h}[m_{H_1} + m_{H_2} + m_{H_3}] \tag{3.1.16}$$

where

$$m_{H_1} = \begin{bmatrix} -4-2c_1 & 1+2c_1 & & & \\ & & 0 & & \\ 1+2c_1 & -2c_1 & & & \\ & & 0 & & \\ & 0 & & \ddots & \\ & & & & 0 \end{bmatrix},$$

$$m_{H_2} = \begin{bmatrix} 0 & & & & \\ & 0 & & 0 & \\ & & \ddots & & \\ & 2\gamma_R\tau_{N-1}^{(H)} & & -1-(1+\gamma_R)\tau_{N-1}^{(H)}+\gamma_R\tau_N^{(H)} & \\ 0 & & & & \\ & -1-(1+\gamma_R)\,\tau_{N-1}^{(H)}+\gamma_R\tau_N^{(H)} & & 4-2(1+\gamma_R)\tau_N^{(H)} & \end{bmatrix},$$

$$m_{H_3} = c_N \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 2 \quad -2 \\ & & & -2 \quad 2 \end{bmatrix}. \tag{3.1.17}$$

Clearly m_{H_3} is N.P.D (non-positive definite) for $\forall c_N \leq 0$. Also, m_{H_1} is N.P.D for $c_1 \geq 1/4$. A simple computation shows that m_{H_2} is N.P.D if τ_{N-1} and τ_N satisfy

$$\tau_N^{(H)} = \frac{2+\delta}{1+\gamma_R} \qquad (\delta \geq 0) \tag{3.1.18}$$

$$\tau_{N-1}^{(H)} = -\frac{1-\gamma_R(1-\delta)}{(1+\gamma_R)^2} \tag{3.1.19}$$

Thus we have proved that $\tilde{M}_H$ is indeed non-positive definite, and therefore $\tilde{M} = \frac{1}{R}\tilde{M}_P + a\tilde{M}_H$ is negative definite for $\forall \frac{1}{R}, a > 0$, with its eigenvalues bounded away below zero by $-\alpha\pi^2/R$, $0 < \alpha < .275$, and the $\tau^{(H)}$'s satisfying (3.1.18) and (3.1.19).

3.2 Numerical examples

3.2.1 One dimensional case

Here we consider the problem

$$\frac{\partial u}{\partial t} + u_x = \frac{1}{R}u_{xx} \qquad t \geq 0, \;\; 0 \leq x \leq 1 \tag{3.2.1}$$

$$u(0,t) = 1$$

$$u(1,t) = 0$$

$$u(x,0) = u_0(x)$$

The steady state solution to (3.2.1) is:

$$u(x) = \frac{1 - e^{-R(1-x)}}{1 - e^{-R}} \tag{3.2.2}$$

Note that $R\ (= 1/\nu)$ plays the role of Reynolds number in this model for a "linear shock layer".

Eq. (3.2.1) was solved numerically by two methods. In one ,referred to as "standard", we use central differencing for the spatial differentiation, and 4^{th}-order Runge-Kutta in time. In this "standard" case, there is no need for special treatment at the boundaries.

The numerical steady state approximation $\mathbf{v}$, ($\frac{\partial \mathbf{v}}{\partial t} = 0$), in this "standard" case, satisfies, for $(\gamma_L = \gamma_R = 1)$, the following finite difference equation:

$$\frac{1}{2h}(v_{j+1} - v_{j-1}) - \frac{1}{Rh^2}(v_{j+1} - 2v_j + v_{j-1}) = 0, \qquad (1 \leq j \leq N-1) \tag{3.2.3}$$

with $v_0 = 1$ and $v_N = 0$. The solution to (3.2.3) is:

$$v_j = \frac{\kappa^j - \kappa^{2N-j}}{1 - \kappa^{2N}}, \qquad \kappa = \frac{2 + hR}{2 - hR}. \tag{3.2.4}$$

Notice that if the "cell Reynolds number", $R_C = hR > 2$, then $\kappa < 0$ and the numerical solution, v_j, will be oscillatory. If $R_C < 2$ then we resolve the "shock layer" (or "boundary layer") and the solution will be smooth.

Numerical steady-state solutions of (3.2.1) using the "standard scheme", and using the "bounded-error" algorithm, (3.1.2), described above are shown in Figures 3.1 to 3.6 for $\Delta x = 1/100$ and various values of R. Both schemes were advanced to steady state using 4^{th}-order Runge-Kutta. It is clear that when $R_C < 2$, both schemes give good results. For $R_C = 10$ $(R = 1000)$ both show oscillations, but the new algorithm approximates the exact solution much better. When $R_C = 10^3$ $(R = 10^5)$, the "standard" numerical solution is useless while the "bounded-error" scheme gives excellent results; in fact far better than for $R_C = 10$.

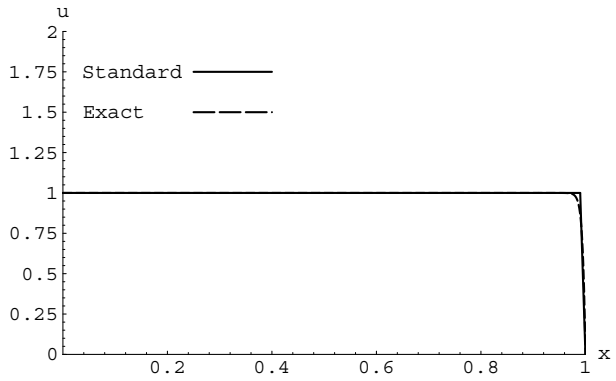

Figure 3.1: Standard scheme, $R_C = 2$

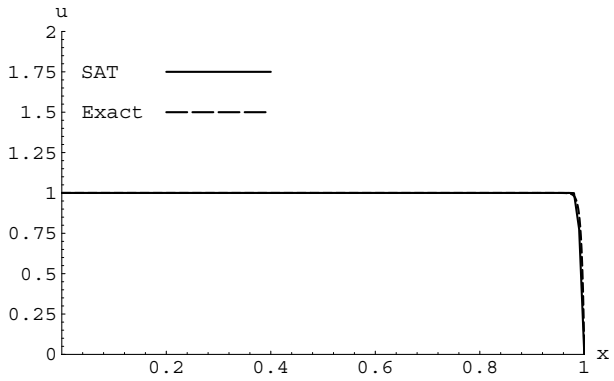

Figure 3.2: SAT, $R_C = 2$

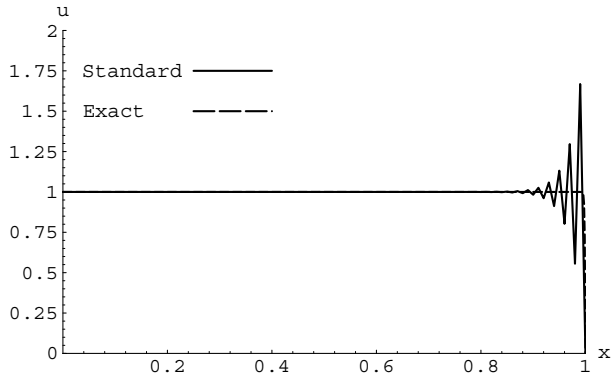

Figure 3.3: Standard scheme, $R_C = 10$

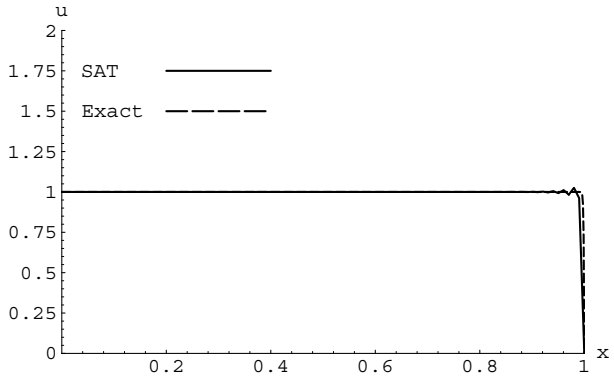

Figure 3.4: SAT, $R_C = 10$

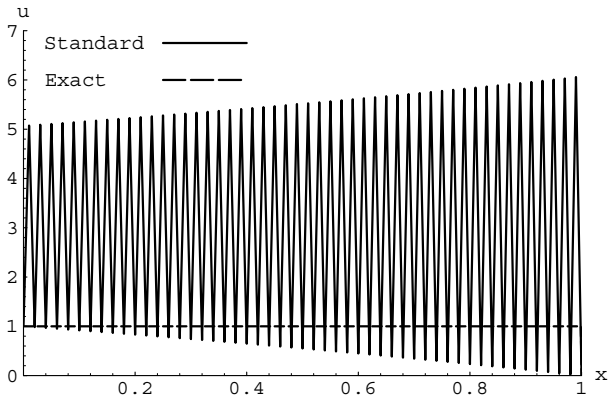

Figure 3.5: Standard scheme, $R_C = 1000$

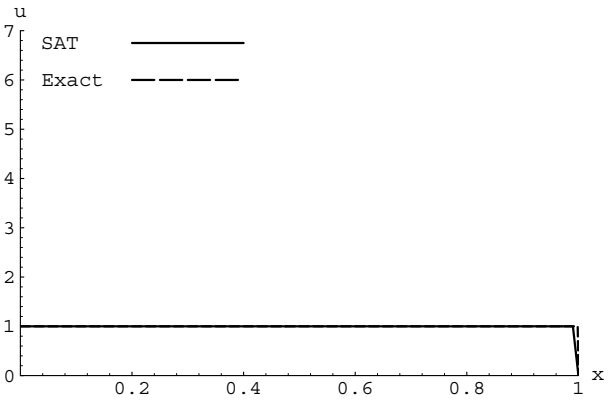

Figure 3.6: SAT, $R_C = 1000$

3.2.2 A steady state two dimensional case

Here we shall consider a linear steady-state problem, which models, in a way, the 2-D boundary layer equations. The formulation is as follows: (the time derivative is left in the equation, since the approach to steady state will be via temporal advance.)

$$u_t + au_x + bu_y = \frac{1}{R}u_{yy}; \qquad t \geq 0;\ \ 0 \leq x < 1;\ \ 0 \leq y \leq 1 \tag{3.2.5}$$

$$u(0,y,t) = \frac{1-e^{bRy}}{1-e^{bR}} + \frac{1}{10}bRe^{\frac{bRy}{2}}\sin \pi y \tag{3.2.5a}$$

$$u(x,0,t) = 0 \tag{3.2.5b}$$

$$u(x,1,t) = 1 \tag{3.2.5c}$$

We also take $a = 1$, and in order to have a growing "boundary layer" on $y = 0$, we must set $b < 0$.

The analytic solution to this problem is:

$$u(x,y) = \frac{1-e^{bRy}}{1-e^{bR}} + \frac{1}{10}bRe^{\frac{bRy}{2}} \exp\left[\left(-\frac{b^2R^2}{4} - \pi^2\right)\frac{x}{Ra}\right]\sin \pi y \tag{3.2.6}$$

Figure 3.7 is a 3-D rendition of $u(x,y)$ for $R = 90,000$. (This 3-D plot looks the same to the eye for various $-1 < b < -4/\sqrt{R} = -4/300$.) Figure 3.8 is a plot of the "velocity profile" inside the "boundary-layer" $(0 < y < .04)$ at $x = .1, .25, .9$ and $b = -4/\sqrt{R}$. The "bumps" at $x = .1$ and $x = .25$ may be considered as "emulating" results of fluid mechanics computation for an incompressible flow near the entrance to a channel, see e.g. [1].

The numerical solution of (3.2.5) using a standard central differencing scheme depends strongly on the value of b (at a given R). Figures 3.9 and 3.10 show the 3-D

plot of $v_{j,k}$ with $b = -1$ and $b = -4/\sqrt{R} = -4/300$. Figs. 3.11 and 3.12 show the profiles at $x = .1$ and $x = .9$ for $b = -1$ and $-\frac{4}{300}$, respectively. It should be emphasized that the "peak" in Figures 3.10 and 3.12 has nothing to do with the "bumps" in the exact solution (see Figure 3.8). The "peak" occurs way outside the boundary layer, and also the amplitude behavior with the x-coordinate is counter to that of Figure 3.12 The "peak" is due to a purely numerical oscillation.

The same series of plots, but as computed by the new algorithm, is shown in Figures 3.13 to 3.16.

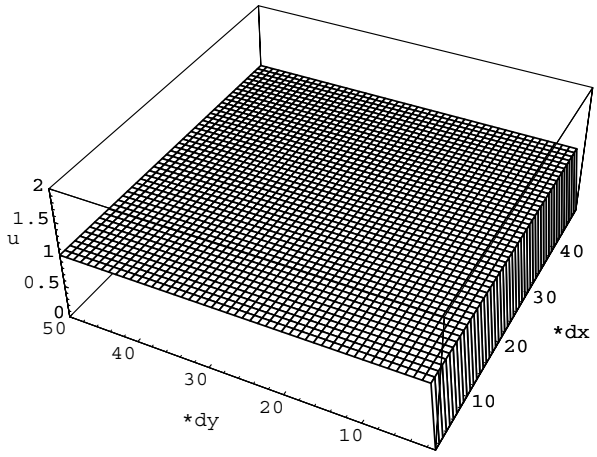

Figure 3.7: Exact solution

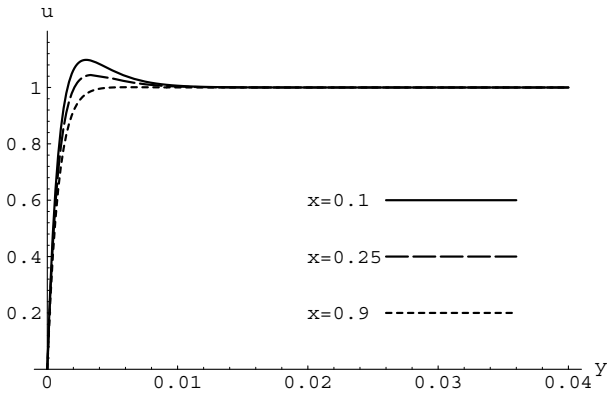

Figure 3.8: Exact solution near the boundary

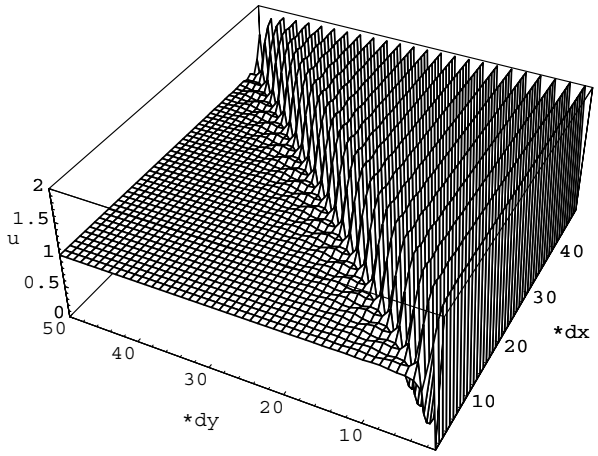

Figure 3.9: Standard scheme, $b = -1$

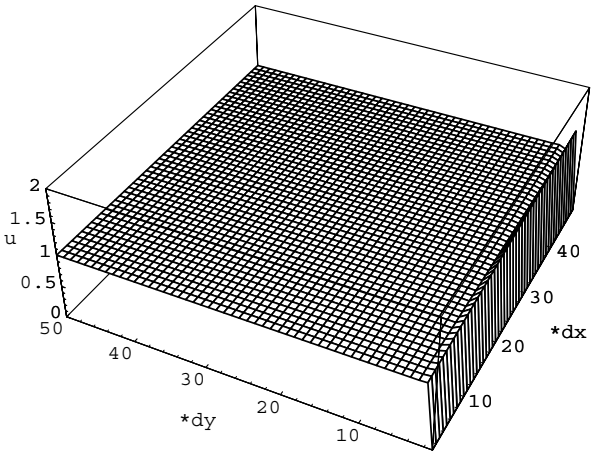

Figure 3.10: Standard scheme, $b = -4/300$

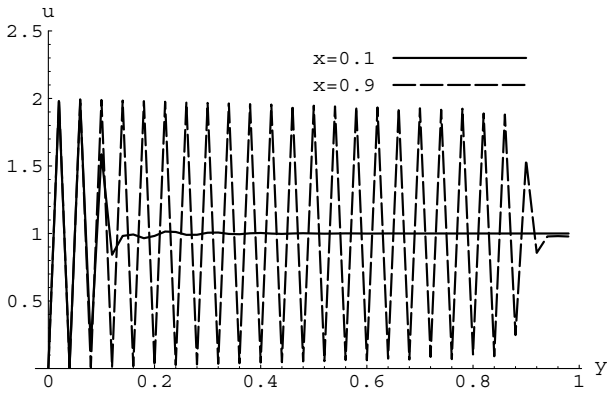

Figure 3.11: Standard scheme, $b = -1$

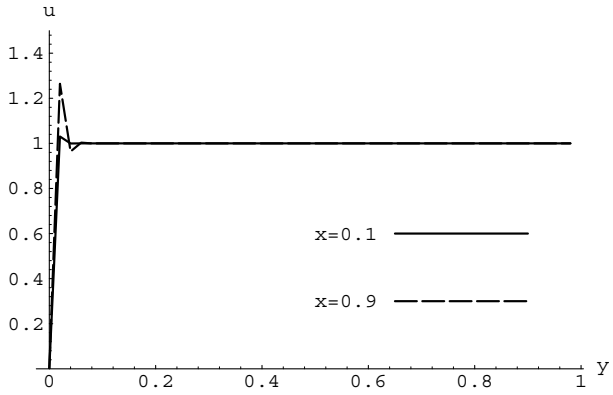

Figure 3.12: Standard scheme, $b = -4/300$

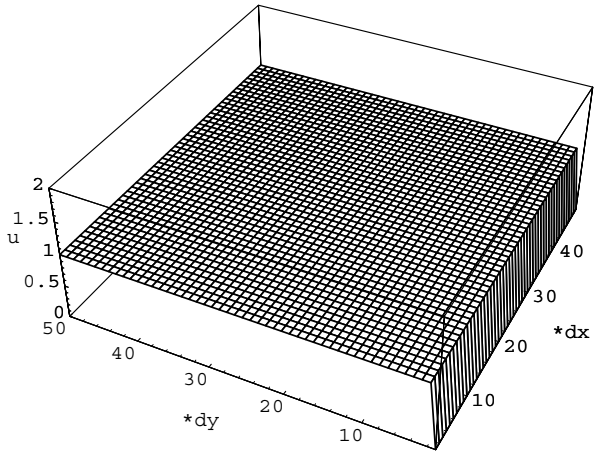

Figure 3.13: SAT, $b = -1$

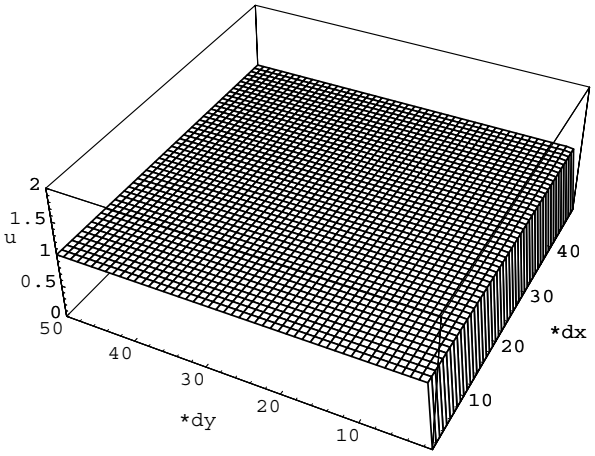

Figure 3.14: SAT, $b = -4/300$

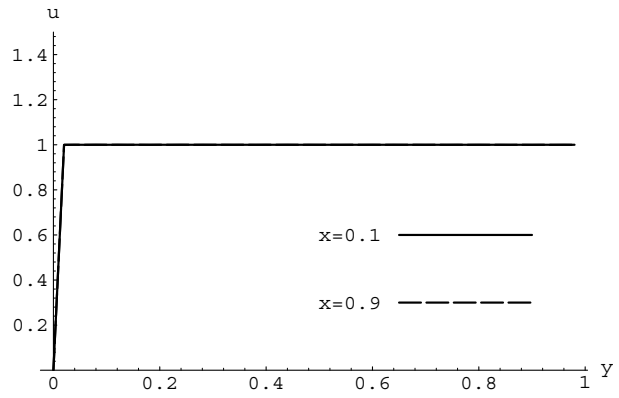

Figure 3.15: SAT, $b = -1$

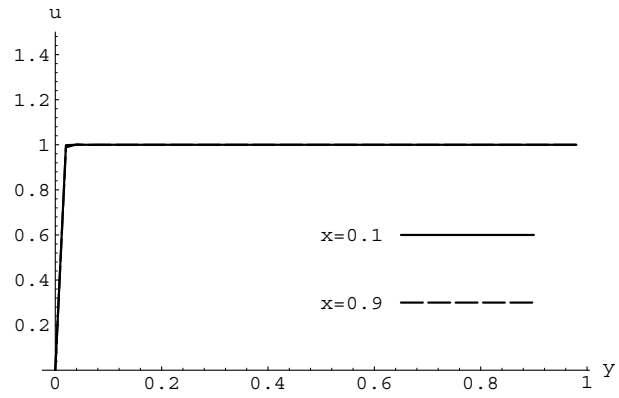

Figure 3.16: SAT, $b = -4/300$

It should be noted (see table 3.1) that the "bounded-error" algorithm converges to steady state (residual L_2 norm $< 10^{-13}$) an order of magnitude faster than the standard scheme when using the same Δt, while cpu-time/iteration is about the same. The standard scheme may be run at bigger Δt (by about a factor of 2) while the SAT algorithm was already at its maximum CFL number. If we let each scheme run at its own maximum Δt then the non-dimensional times to get to a steady-state in both cases are about equal, but the difference in errors remains.

	'time' to 'steady-state'	L_2 residual	L_1 norm of the error	L_2 norm of the error	L_∞ norm of the error	max error location
$b = -1$						
SAT	21.09	9.911e-14	8.805e-05	1.076e-04	3.108e-04	45, 46
Standard	417	9.987e-14	0.485139	0.674233	-1.00423	10, 4
$b = -4/300$						
SAT	52.64	9.943e-14	1.665e-04	1.142e-03	0.01220	50, 2
Standard	416	9.967e-14	3.362e-03	2.447e-02	-0.2864	50, 2

Table 3.1: Rectangular geometry results

We also ran the same equations for a non-strictly rectangular geometry, where the upper boundary instead of being $y = 1$ is $y = 1 - (\tan\theta)x$, where θ is the angle which the upper boundary makes with the x-axis, see Figure 3.17.

	'time' to 'steady-state'	L_2 residual	L_1 norm of the error	L_2 norm of the error	L_∞ norm of the error	max error location
$b = -4/300$						
SAT	52.56	9.984e-14	1.707e-04	1.156e-03	0.01220	50, 2
Standard	401.11	9.995e-14	3.448e-03	2.479e-02	-0.2864	50, 2

Table 3.2: Trapezoid geometry results

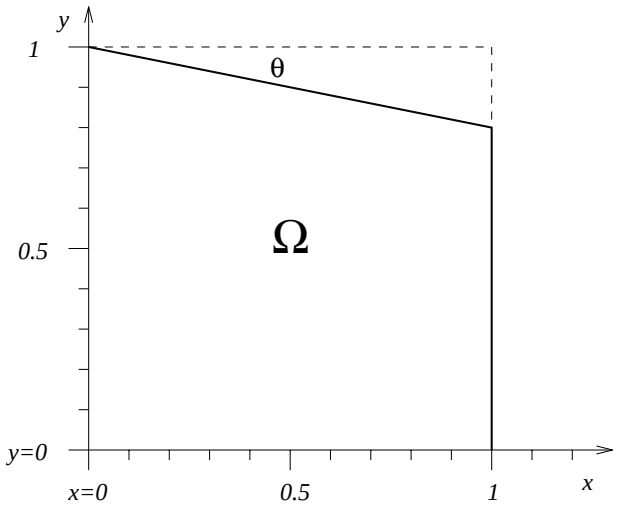

Figure 3.17:

For many θ's the results of the performance of the two schemes are unaffected by the change. However, there are some θ's for which the standard scheme converges to steady state much slower than before at its own maximum allowed Δt, while the performance of the bounded-error algorithm remains the same as before. For example, see table 3.2, for the case of $\theta = 3.9°$. As in chapter 2, the point is that for non-rectangular geometry the distance that a boundary is away from a computational mode, γh, might become extremely small and this causes the deterioration in the performance of the standard scheme. Here it is reflected in the fact that the standard scheme cannot "support" the larger allowed Δt that can be achieved for the case $\theta = 0$. For more complex geometries it is very difficult to predict a-priori what range the values of γ will take. The SAT methods (the bounded error algorithm) are insensitive to the variations in γ caused by the geometry of the domain.

3.2.3 A 2-D time dependent example

To check on the temporal "performance" of the bounded-error scheme, we considered the following problem:

$$u_t + au_x + bu_y = \frac{1}{R}u_{yy} + \sigma b \sin[k(x-at)]; \;\; t \geq 0, \;\; 0 \leq x < 1, \;\; 0 \leq y \leq 1 \tag{3.2.7a}$$

$$u(x,y,0) = \frac{1-e^{bRy}}{1-e^{bR}} + \frac{bR}{10}e^{\frac{bRy}{2}}e^{-(\frac{b^2R^2}{4}+\pi^2)\frac{x}{Ra}}\sin \pi y + y\sigma \sin kx \tag{3.2.7b}$$

$$u(0,y,t) = \frac{1-e^{bRy}}{1-e^{bR}} + \frac{bR}{10}e^{\frac{bRy}{2}}\sin \pi y - y\sigma \sin kat \tag{3.2.7c}$$

$$u(x,0,t) = 0 \tag{3.2.7d}$$

$$u(x,1,t) = 1 + \sigma \sin[k(x-at)] \tag{3.2.7e}$$

The exact solution of (3.2.7) is:

$$u(x,y,t) = \frac{1-e^{bRy}}{1-e^{bR}} + \frac{bR}{10}e^{\frac{bRy}{2}}e^{-(\frac{b^2R^2}{4}+\pi^2)\frac{x}{Ra}}\sin \pi y + y\sigma \sin[k(x-at)] \tag{3.2.8}$$

Again we take $a = 1$, $R = 90{,}000$, $b = -1$, and $-\frac{4}{\sqrt{R}}$. The parameters σ and k have certain constraints. If we want $u > 0$, we must take $\sigma < 1$. The number of computational nodes, N, puts a lower bound of $2\pi N$ on the wave-length, $1/k$, i.e., $1 < k < 2\pi N$. In the actual computations we used $\sigma = 1/2$ and $k = 30$. All the plots for this time dependent case are shown for $t = 10$. Figure 3.18 shows a 3-D plot of $u(x,y,10)$. As in the steady-state case, the plot looks the same to the eye for various $-1 < b < -4/\sqrt{R} = -4/300$. Figures 3.19, and 3.20 show the 3-D plots of $v_{j,k}$ for the standard and bounded-error schemes respectively. Figure 3.21, shows a x-profile of v at $y = .2$, for both schemes and the exact profile, for $b = 1$. Figure 3.22, gives the same profiles at $y = .8$. These plots bring out the differences in the phase errors of the numerical algorithms. Figures 3.23 to 3.26, repeat the same information as given in Figures 3.19 to 3.22 , but for $b = -4\sqrt{R} = -4/300$. The efficacy of the bounded-error algorithm is quite evident – even when $b = -4/\sqrt{R}$, where the norm-errors away from the boundary layer are not dissimilar. The phase error of the right running waves is quite a bit smaller in the case of the proposed present scheme.

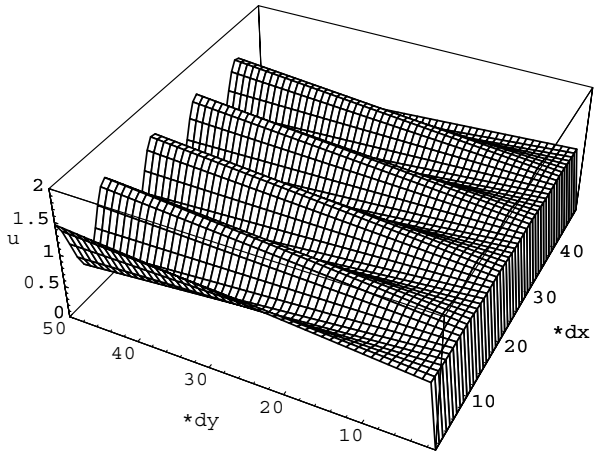

Figure 3.18: Exact solution.

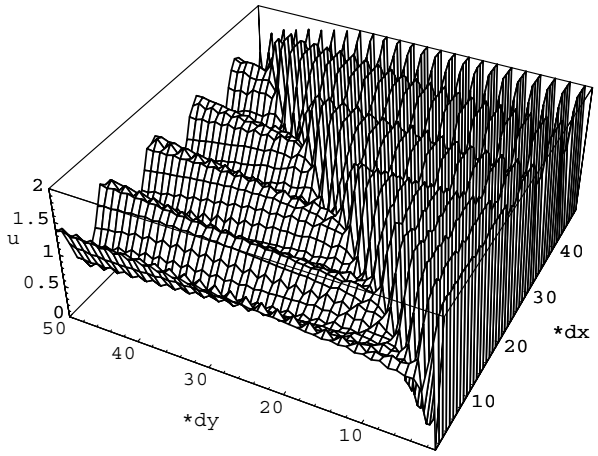

Figure 3.19: Standard scheme, $b = -1$.

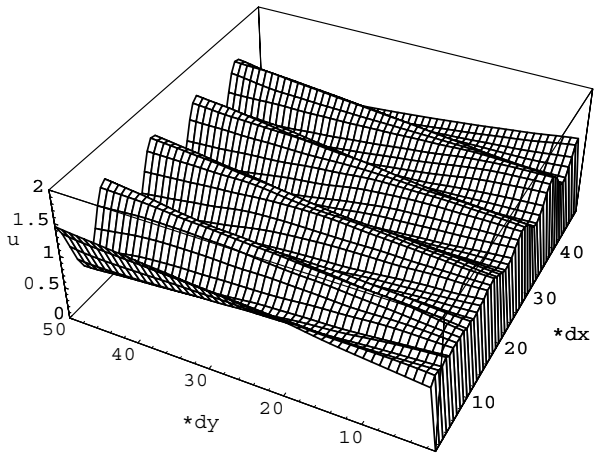

Figure 3.20: SAT, $b = -1$.

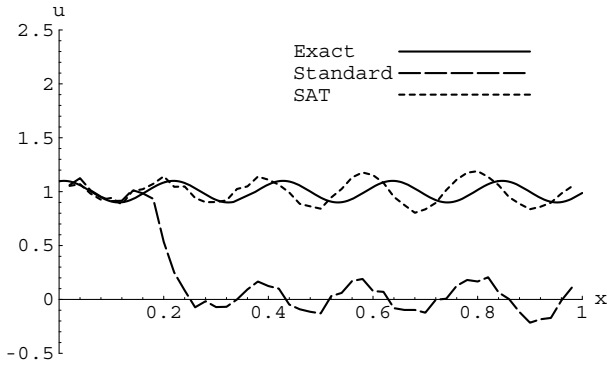

Figure 3.21: $b = -1$, $y = 0.2$ profiles.

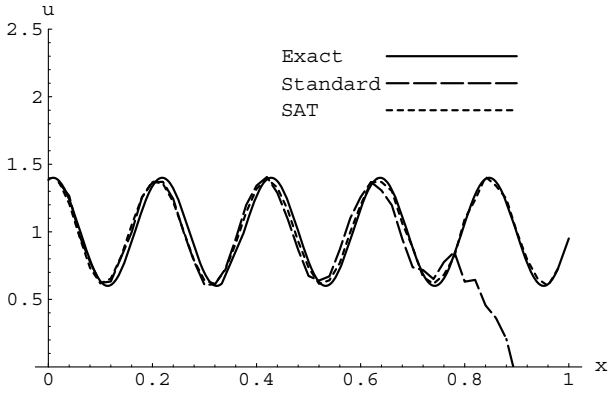

Figure 3.22: $b = -1$, $y = 0.8$ profiles.

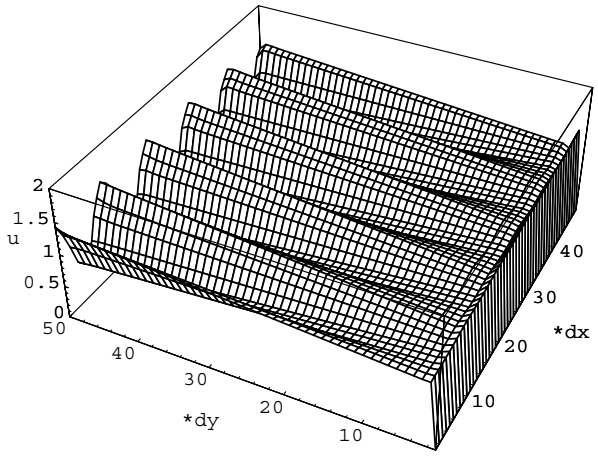

Figure 3.23: Standard scheme, $b = -4/300$.

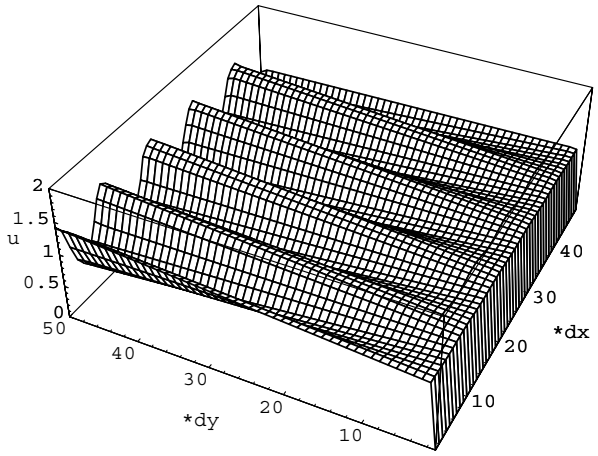

Figure 3.24: SAT, $b = -4/300$.

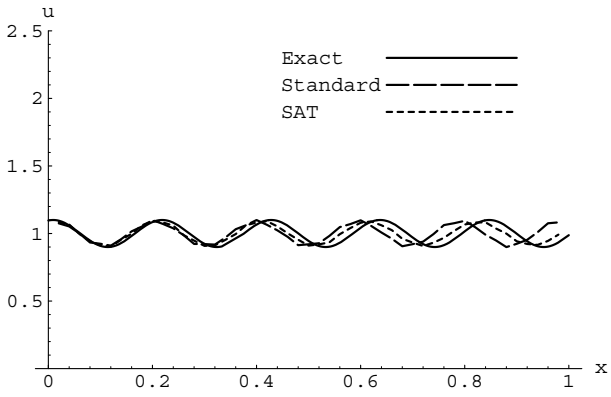

Figure 3.25: $b = -4/300$, $y = 0.2$ profiles.

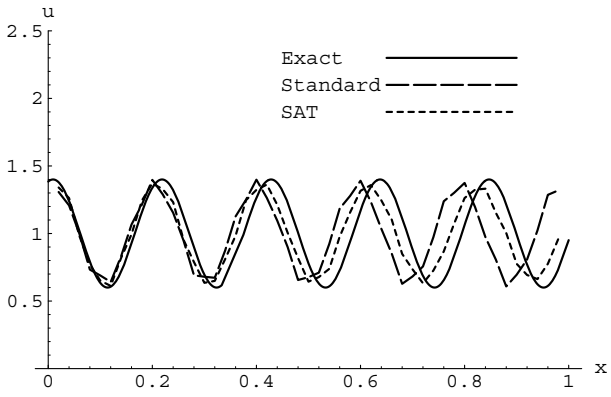

Figure 3.26: $b = -4/300$, $y = 0.8$ profiles.

3.3 Conclusions

(i) A second order method has been developed which renders spatial second derivative finite difference operators negative definite. This is not surprising, since negative definiteness was achieved for 4^{th} order parabolic operators in chapter 2, see also [2].

(ii) A second order method has been developed which renders spatial first derivative finite difference operators non-positive definite. For the case when boundary points do not coincide with grid nodes ($\gamma \neq 1$), this is a new result.

(iii) The results (i) and (ii) allow us to construct a solution operator for the advection-diffusion problem (and, of course, the diffusion equation) which is negative definite,

thereby ensuring that this scheme is indeed error bounded.

(iv) The construction of these operators allows an immediate simple generalization to multi-dimensional problems, on complex domains which are covered by rectangular meshes. The proofs of the boundness of the error-norms carry over rigorously to the (linear) multi-dimensional cases.

(v) Numerous numerical examples demonstrate the efficacy of this methodology.

3.4 Appendix, The case $a < 0$

As in the $a > 0$ case the hyperbolic terms are given by:

$$
\begin{aligned}
D_H = \frac{1}{2h}\Bigg\{ & \begin{bmatrix}
-2 & 2 & & & & & \\
-1 & 0 & 1 & & & & \\
 & -1 & 0 & 1 & & & \\
 & & \ddots & \ddots & \ddots & & \\
 & & & -1 & 0 & 1 & 0 \\
 & & & & -1 & 0 & 1 \\
 & & & & & -2 & 2
\end{bmatrix} \\
& + \begin{bmatrix}
c_1 & & & & & & \\
 & c_2 & & & & & \\
 & & c_3 & & & & \\
 & & & \ddots & & & \\
 & & & & c_{N-2} & & \\
 & & & & & c_{N-1} & \\
 & & & & & & c_N
\end{bmatrix}
\begin{bmatrix}
-1 & 2 & -1 & & & & \\
0 & -1 & 2 & -1 & & & \\
1 & -2 & 0 & 2 & -1 & & \\
 & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & & 1 & -2 & 0 & 2 & -1 \\
 & & & 1 & -2 & 1 & 0 \\
 & & & & 1 & -2 & 1
\end{bmatrix} \\
& + 2h\tilde{c}
\begin{bmatrix}
0 & -1 & 1 & & & & & \\
1 & -1 & -1 & 1 & & & & \\
1 & -1 & 0 & -1 & 1 & & & \\
 & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
 & & 1 & -1 & 0 & -1 & 1 & \\
 & & & & 1 & -1 & -1 & 1 \\
 & & & & & 1 & -1 & 0
\end{bmatrix}\Bigg\}
\end{aligned}
\tag{3.4.1}
$$

where

$$
c_k = \frac{1}{N-1}[(c_N - c_1)k + (Nc_1 - c_N)], \tag{3.4.2}
$$

and

$$
\tilde{c} = \frac{1}{2}(c_1 - c_N). \tag{3.4.3}
$$

For $a < 0$ in (3.1.1a), the right boundary is, for the hyperbolic part, an "outflow" boundary on which we do not prescribe a "hyperbolic boundary condition", therefore, in this case $\tau_{R_H} = 0$, and

$$\tau_{L_H} = \text{ diag } [\tau_{L_1}^{(H)}, \tau_{L_2}^{(H)}, 0, \ldots, 0, 0] \tag{3.4.4}$$

$$A_{L_H} = \begin{bmatrix} 1+\gamma_L & -\gamma_L & & & \\ 1+\gamma_L & -\gamma_L & & 0 & \\ & & 0 & & \\ & & & \ddots & \\ & & 0 & & 0 \end{bmatrix} \tag{3.4.5}$$

Since $a < 0$, and we want $a\tilde{M}_H$ to be non-positive definite, we need to show that $\tilde{M}_H = \frac{1}{2}(M_H + M_H^T)$ is non-negative definite. Using equations (3.4.1)–(3.4.5) we have

$$\tilde{M}_H = \frac{1}{4h} \begin{bmatrix} -4-2c_1-2(1+\gamma_L)\tau_1^{(H)} & 1+2c_1-(1+\gamma_L)\tau_2^{(H)}+\gamma_L\tau_1^{(H)} & 0 & & & \\ 1+2c_1-(1+\gamma_L)\tau_2^{(H)}+\gamma_L\tau_1^{(H)} & -2c_1+2\gamma_L\tau_2^{(H)} & 0 & & & 0 \\ 0 & 0 & 0 & & & \\ & & & \ddots & & \\ 0 & & & & 2c_N & -1-2c_N \\ & & & & -1-2c_N & 4+2c_N \end{bmatrix} \tag{3.4.6}$$

We now write $\tilde{M}_H$ as the sum of three "corner-matrices",

$$\tilde{M}_H = \frac{1}{4h}[m_{H_1} + m_{H_2} + m_{H_3}] \tag{3.4.7}$$

where

$$m_{H_1} = c_1 \begin{bmatrix} -2 & 2 & & & \\ 2 & -2 & & 0 & \\ & & 0 & & \\ 0 & & & \ddots & \\ & & & & 0 \end{bmatrix},$$

$$m_{H_2} = \begin{bmatrix} -4-2(1+\gamma_L)\tau_1^{(H)} & 1-(1+\gamma_L)\tau_2^{(H)}+\gamma_L\tau_1^{(H)} & 0 & & \\ 1-(1+\gamma_L)\tau_2^{(H)}+\gamma_L\tau_1^{(H)} & +2\gamma\tau_2^{(H)} & 0 & 0 & \\ 0 & 0 & 0 & & \\ & & & 0 & \\ & 0 & & & \ddots \\ & & & & 0 \end{bmatrix},$$

$$m_{H_3} = \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 2c_N & -1-2c_N \\ & & & -1-2c_N & 4+2c_N \end{bmatrix} \tag{3.4.8}$$

Clearly m_{H_1} is N.N.D (non-negative definite) for $\forall c_1 \leq 0$. Also, m_{H_3} is N.N.D for $c_N \geq -1/4$. A simple computation shows that m_{H_2} is N.N.D if τ_1 and τ_2 satisfy

$$\tau_1^{(H)} = -\frac{2+\delta}{1+\gamma_L} \qquad (\delta \geq 0) \tag{3.4.9}$$

$$\tau_2^{(H)} = \frac{1-\gamma_L(1-\delta)}{(1+\gamma_L)^2} \tag{3.4.10}$$

Thus we have proved that $\tilde{M}_H$ is indeed non-negative definite, and therefore $\tilde{M} = \frac{1}{R}\tilde{M}_P + a\tilde{M}_H$ is negative definite for $\forall \frac{1}{R} > 0$, with its eigenvalues bounded away from zero by $-\alpha\pi^2/R$, $0 < \alpha < .275$, and $\tau^{(H)}$'s satisfying (3.4.9) and (3.4.10), as in the $a > 0$ case treated in the text.

Chapter 4

Mixed derivatives and parabolic systems

In the previous chapters we first developed a one-dimensional scheme, and then generalized it to the multi-dimensional case, following the outline presented in chapter 1. However, since a mixed derivative problem is inherently multi-dimensional, a different strategy should be adopted. In section 4.1 two methods to tackle the mix derivative scalar problem are presented. In section 4.2 a 2^{nd} order scheme is presented that solves this problem. In section 4.3 parabolic systems are discussed, and a scheme is constructed that solves the diffusion part of the Navier-Stokes equations in two and three space dimensions.

4.1 The scalar mixed derivatives problem

Consider the scalar problem

$$\frac{\partial u}{\partial t} = \sum_{i,j=1}^{d} c_{ij} \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} u; \qquad (x_1, \ldots, x_d) \in \Omega \subset R^d; \quad t \geq 0 \tag{4.1.1a}$$

$$u(x_1, \ldots, x_d, 0) = u_0(x_1, \ldots, x_d) \tag{4.1.1b}$$

$$u(x_1, \ldots, x_d, t)|_{\partial\Omega} = u_B(t) \tag{4.1.1c}$$

It is assumed that this differential problem is strictly-stable in the sense of Petrowski, i.e. $\sum_{i,j=1}^{d} c_{ij}\eta_i\eta_j > \delta > 0$ for all $\sum_{j=1}^{d} \eta_j^2 = 1$, see for example [10], and that $c_{ij} = c_{ji}$

Note that equation (4.1.1) can also be written as:

$$\frac{\partial u}{\partial t} = (\partial_{x_1}, \ldots, \partial_{x_d})\, C \begin{pmatrix} \partial_{x_1} \\ \vdots \\ \partial_{x_d} \end{pmatrix} u \tag{4.1.2}$$

where the matrix C has c_{ij} as its entries. Using this notation, the matrix C is symmetric and positive-definite.

One way to attack this mixed derivative problem problem is to convert equation (4.1.1) to its canonical form, $u_t = \nabla^2 u$, using a change of variables. This standard technique can be found in many P.D.E. textbooks, for example [16]. In the new variables the problem was already solved in chapter 2.

In some cases this method can be very effective. Often however, especially when parabolic *systems* have to be solved, a change of variables may not be possible since a change of variables that diagonalizes one equation in a system may not diagonalize the others. The rest of this section is devoted to the description of an alternative approach, which will be demonstrated in two space dimensions. A second-order accuracy scheme will be built in the next section. It should be emphasized that this procedure might not always work. However, under certain conditions, bounded-error schemes for scalar equations and systems can be generated by using method. These conditions are discussed in the next section.

We consider the scalar problem

$$\frac{\partial u}{\partial t} = a^2(1+\mu_1^2)u_{xx} + 2abu_{xy} + b^2(1+\mu_2^2)u_{yy}; \qquad (x,y) \in \Omega; \quad t \geq 0 \tag{4.1.3a}$$

$$u(x,y,0) = u_0(x,y) \tag{4.1.3b}$$

$$u(x,y,t)|_{\partial\Omega} = u_B(t) \tag{4.1.3c}$$

Let us use a multi-dimensional grid and the notation presented in section 1.2. Then the projection of the exact solution $\mathbf{u}(t)$ satisfies:

$$\begin{aligned} \frac{d\mathbf{U}}{dt} &= a^2(1+\mu_1^2)\mathcal{D}^{(xx)}\mathbf{U} + ab\left[\mathcal{D}_2^{(x)}\, P^T\mathcal{D}_1^{(y)}P + P^T\mathcal{D}_2^{(y)}P\, \mathcal{D}_1^{(x)}\right]\mathbf{U} + \\ & \quad b^2(1+\mu_2^2)P^T\mathcal{D}^{(yy)}P\mathbf{U} + \mathbf{T} \end{aligned} \tag{4.1.4}$$

where

$$\mathcal{D}^{(xx)} = \begin{bmatrix} D_1^{(xx)} & & & \\ & D_2^{(xx)} & & \\ & & \ddots & \\ & & & D_{M_R}^{(xx)} \end{bmatrix}; \mathcal{D}^{(yy)} = \begin{bmatrix} D_1^{(yy)} & & & \\ & D_2^{(yy)} & & \\ & & \ddots & \\ & & & D_{M_C}^{(yy)} \end{bmatrix} \quad (4.1.5)$$

i.e. the matrix $\mathcal{D}^{(xx)}$ is a diagonal-block matrix, the k^{th} block in its diagonal, $D_k^{(xx)}$, is a differentiation sub-matrix which represents second x derivative when applied to the k^{th} row. Similarly the matrix $\mathcal{D}^{(yy)}$ is also a diagonal-block matrix, and the j^{th} block in its diagonal, $D_j^{(yy)}$, is a differentiation sub-matrix which represents second y derivative when applied to the j^{th} column. The matrices

$$\mathcal{D}_1^{(x)} = \begin{bmatrix} D_{1_1}^{(x)} & & & \\ & D_{1_2}^{(x)} & & \\ & & \ddots & \\ & & & D_{1_{M_R}}^{(x)} \end{bmatrix}; \mathcal{D}_2^{(x)} = \begin{bmatrix} D_{2_1}^{(x)} & & & \\ & D_{2_2}^{(x)} & & \\ & & \ddots & \\ & & & D_{2_{M_R}}^{(x)} \end{bmatrix} \quad (4.1.6)$$

and

$$\mathcal{D}_1^{(y)} = \begin{bmatrix} D_{1_1}^{(y)} & & & \\ & D_{1_2}^{(y)} & & \\ & & \ddots & \\ & & & D_{1_{M_C}}^{(y)} \end{bmatrix}; \mathcal{D}_2^{(y)} = \begin{bmatrix} D_{2_1}^{(y)} & & & \\ & D_{2_2}^{(y)} & & \\ & & \ddots & \\ & & & D_{2_{M_C}}^{(y)} \end{bmatrix} \quad (4.1.7)$$

are defined in a similar way, with the difference that the matrices $D_{1_k}^{(x)}$, $D_{2_k}^{(x)}$, $D_{1_j}^{(y)}$ and $D_{2_j}^{(y)}$ represents first derivative.

Now we can write the scheme:

$$\begin{aligned} \frac{d\mathbf{V}}{dt} &= a^2(1+\mu_1^2)\mathcal{M}^{(xx)}\mathbf{V} + \\ &\quad ab\left[\mathcal{M}_2^{(x)}\, P^T\mathcal{M}_1^{(y)}P + P^T\mathcal{M}_2^{(y)}P\, \mathcal{M}_1^{(x)}\right]\mathbf{V} + \\ &\quad b^2(1+\mu_2^2)P^T\mathcal{M}^{(yy)}P\mathbf{V} + \mathbf{G}^{(x)} + P^T\mathbf{G}^{(y)} \end{aligned} \quad (4.1.8)$$

The $\mathcal{M}$ matrices have the same structure as the $\mathcal{D}$ matrices do, with the sub-matrices defined in a manner similar to that of chapter 1:

$$
\begin{aligned}
M_k^{(xx)} &= D_k^{(xx)} - \tau_{L_k} A_{L_k} - \tau_{R_k} A_{R_k} \\
M_{1_k}^{(x)} &= D_{1_k}^{(x)} - \sigma_{L_k} A_{L_k} - \sigma_{R_k} A_{R_k} \\
M_{2_k}^{(x)} &= D_{2_k}^{(x)} \\
M_j^{(yy)} &= D_j^{(yy)} - \tau_{B_j} A_{B_j} - \tau_{T_j} A_{T_j} \\
M_{1_j}^{(y)} &= D_{1_j}^{(y)} - \sigma_{B_j} A_{B_j} - \sigma_{T_j} A_{T_j} \\
M_{2_j}^{(y)} &= D_{2_j}^{(y)}
\end{aligned}
\tag{4.1.9}
$$

and

$$
\begin{aligned}
\mathbf{G}^{(x)} &= \Big[(\tau_{L_1}\mathbf{g}_{L_1} + \tau_{R_1}\mathbf{g}_{R_1}), \ldots, (\tau_{L_k}\mathbf{g}_{L_k} + \tau_{R_k}\mathbf{g}_{R_k}), \ldots, (\tau_{L_{M_R}}\mathbf{g}_{L_{M_R}} + \tau_{R_{M_R}}\mathbf{g}_{R_{M_R}})\Big] + \\
&\quad \Big[(\sigma_{L_1}\mathbf{g}_{L_1} + \sigma_{R_1}\mathbf{g}_{R_1}), \ldots, (\sigma_{L_k}\mathbf{g}_{L_k} + \sigma_{R_k}\mathbf{g}_{R_k}), \ldots, (\sigma_{L_{M_R}}\mathbf{g}_{L_{M_R}} + \sigma_{R_{M_R}}\mathbf{g}_{R_{M_R}})\Big], \\
\mathbf{G}^{(y)} &= \Big[(\tau_{B_1}\mathbf{g}_{B_1} + \tau_{T_1}\mathbf{g}_{T_1}), \ldots, (\tau_{B_j}\mathbf{g}_{B_j} + \tau_{T_j}\mathbf{g}_{T_j}), \ldots, (\tau_{B_{M_C}}\mathbf{g}_{B_{M_C}} + \tau_{T_{M_C}}\mathbf{g}_{T_{M_C}})\Big] + \\
&\quad \Big[(\sigma_{B_1}\mathbf{g}_{B_1} + \sigma_{T_1}\mathbf{g}_{T_1}), \ldots, (\sigma_{B_j}\mathbf{g}_{B_j} + \sigma_{T_j}\mathbf{g}_{T_j}), \ldots, (\sigma_{B_{M_C}}\mathbf{g}_{B_{M_C}} + \sigma_{T_{M_C}}\mathbf{g}_{T_{M_C}})\Big].
\end{aligned}
\tag{4.1.10}
$$

After subtracting (4.1.8) from (4.1.4) we get:

$$
\frac{d\mathbf{E}}{dt} = a^2(1+\mu_1^2)\mathcal{M}^{(xx)}\mathbf{E} + ab\left[\mathcal{M}_2^{(x)}\ P^T\mathcal{M}_1^{(y)}P + P^T\mathcal{M}_2^{(y)}P\ \mathcal{M}_1^{(x)}\right] + b^2(1+\mu_2^2)P^T\mathcal{M}^{(yy)}P\mathbf{E} + \mathbf{T}
\tag{4.1.11}
$$

where $\mathbf{E} = \mathbf{U} - \mathbf{V}$. The time rate of change of $\| \mathbf{E} \|^2$ is given by:

$$
\begin{aligned}
\frac{1}{2}\frac{d}{dt}\| \mathbf{E} \|^2 &= a^2(1+\mu_1^2)\left(\mathbf{E}, \mathcal{M}^{(xx)}\mathbf{E}\right) + ab\left(\mathbf{E}, \mathcal{M}_2^{(x)}\ P^T\mathcal{M}_1^{(y)}P\mathbf{E}\right) + \\
&\quad ab\left(\mathbf{E}, P^T\mathcal{M}_2^{(y)}P\ \mathcal{M}_1^{(x)}\mathbf{E}\right) + b^2(1+\mu_2^2)\left(\mathbf{E}, P^T\mathcal{M}^{(yy)}P\mathbf{E}\right) + (\mathbf{E}, \mathbf{T})
\end{aligned}
\tag{4.1.12}
$$

Equation (4.1.12) may be written as:

$$
\begin{aligned}
\frac{1}{2}\frac{d}{dt}\| \mathbf{E} \|^2 &= a^2(1+\mu_1^2)\left(\mathbf{E}, \mathcal{M}^{(xx)}\mathbf{E}\right) + ab\left({\mathcal{M}_2^{(x)}}^T\mathbf{E}, P^T\mathcal{M}_1^{(y)}P\mathbf{E}\right) + \\
&\quad ab\left(P^T{\mathcal{M}_2^{(y)}}^T P\mathbf{E}, \mathcal{M}_1^{(x)}\mathbf{E}\right) + b^2(1+\mu_2^2)\left(\mathbf{E}, P^T\mathcal{M}^{(yy)}P\mathbf{E}\right) + (\mathbf{E}, \mathbf{T})
\end{aligned}
$$

Now we may use the Schwarz inequality and get:

$$\begin{aligned}\frac{1}{2}\frac{d}{dt} \parallel \mathbf{E} \parallel^2 &\leq a^2(1+\mu_1^2)(\mathbf{E}, \mathcal{M}^{(xx)}\mathbf{E}) + ab \parallel \mathcal{M}_2^{(x)^T}\mathbf{E} \parallel \parallel P^T\mathcal{M}_1^{(y)}P\mathbf{E} \parallel + \\ & ab \parallel P^T\mathcal{M}_2^{(y)^T}P\mathbf{E} \parallel \parallel \mathcal{M}_1^{(x)}\mathbf{E} \parallel + b^2(1+\mu_2^2)(\mathbf{E}, P^T\mathcal{M}^{(yy)}P\mathbf{E}) + \parallel \mathbf{E} \parallel \parallel \mathbf{T} \parallel\end{aligned}$$

Then by using the inequality:

$$\alpha\beta \leq \frac{\alpha^2+\beta^2}{2} \tag{4.1.13}$$

we may write

$$\begin{aligned}\frac{1}{2}\frac{d}{dt} \parallel \mathbf{E} \parallel^2 &\leq a^2\left(\mathbf{E}, \left[(1+\mu_1^2)\mathcal{M}^{(xx)} + \frac{1}{2}\left(\mathcal{M}_1^{(x)^T}\mathcal{M}_1^{(x)} + \mathcal{M}_2^{(x)}\mathcal{M}_2^{(x)^T}\right)\right]\mathbf{E}\right) + \\ & b^2\left(\mathbf{E}, P^T\left[(1+\mu_2^2)\mathcal{M}^{(yy)} + \frac{1}{2}\left(\mathcal{M}_1^{(y)^T}\mathcal{M}_1^{(y)} + \mathcal{M}_2^{(y)}\mathcal{M}_2^{(y)^T}\right)\right]P\mathbf{E}\right) + \\ & \parallel \mathbf{E} \parallel \parallel \mathbf{T} \parallel\end{aligned} \tag{4.1.14}$$

If the matrices

$$\left[(1+\mu_1^2)\mathcal{M}^{(xx)} + \frac{1}{2}\left(\mathcal{M}_1^{(x)^T}\mathcal{M}_1^{(x)} + \mathcal{M}_2^{(x)}\mathcal{M}_2^{(x)^T}\right)\right] \tag{4.1.15}$$

and

$$\left[(1+\mu_2^2)\mathcal{M}^{(yy)} + \frac{1}{2}\left(\mathcal{M}_1^{(y)^T}\mathcal{M}_1^{(y)} + \mathcal{M}_2^{(y)}\mathcal{M}_2^{(y)^T}\right)\right] \tag{4.1.16}$$

are negative definite and bounded away from 0 by $-c_0,\ c_0 > 0$ then the error norm $\parallel \mathbf{E} \parallel$ is bounded, see chapter 1 for the details. Note that each of these matrices is a diagonal-block-matrix, therefor it is sufficient to prove that each of their blocks

$$\left[(1+\mu_1^2)M_k^{(xx)} + \frac{1}{2}\left(M_{1_k}^{(x)^T}M_{1_k}^{(x)} + M_{2_k}^{(x)}M_{2_k}^{(x)^T}\right)\right] \tag{4.1.17}$$

and

$$\left[(1+\mu_2^2)M_j^{(yy)} + \frac{1}{2}\left(M_{1_j}^{(y)^T}M_{1_j}^{(y)} + M_{2_j}^{(y)}M_{2_j}^{(y)^T}\right)\right] \tag{4.1.18}$$

are negative definite and bounded away from 0.

We can try to solve equation (4.1.1), in more than two space dimensions ($d > 2$), using the same method. Writing the scheme, by analogy to equation (4.1.8) we get:

$$\frac{d\mathbf{V}}{dt} = \sum_{j=1}^{d} c_{jj} P_j^T \mathcal{M}^{(x_j x_j)} P_j \mathbf{V} + \sum_{j=1}^{d} \sum_{i=1}^{j-1} c_{ij} \left[P_i^T \mathcal{M}_2^{(x_i)} P_i \; P_j^T \mathcal{M}_1^{(x_j)} P_j + P_j^T \mathcal{M}_2^{(x_j)} P_j \; P_i^T \mathcal{M}_1^{(x_i)} P_i \right] \mathbf{V} + \sum_{j=1}^{d} P_j^T \mathbf{G}^{(x_j)} \quad , \tag{4.1.19}$$

where the P_j's are the permutation matrices defined in section 1.2, (we take $P_1 = I$) and the $\mathcal{M}$'s and $\mathbf{G}$'s are defined in a way similar to equations (4.1.9) and (4.1.10) respectively. Now we write $c_{ij} = c_{ij}^{(i)} \; c_{ij}^{(j)}, \;\; i \neq j$, (thus for $d = 2$, $c_{ij} = c_{1\,2} = c_{1\,2}^{(1)} \; c_{1\,2}^{(2)} = ab$), and after doing the same manipulations leading to equation (4.1.14) we get:

$$\frac{1}{2}\frac{d}{dt} \parallel \mathbf{E} \parallel^2 \leq \sum_{j=1}^{d} \left(\mathbf{E}, P_j^T \left[c_{jj} \mathcal{M}^{(x_j x_j)} + \frac{1}{2} \sum_{i \neq j} c_{ij}^{(j)^2} \left(\mathcal{M}_1^{(x_j)^T} \mathcal{M}_1^{(x_j)} + \mathcal{M}_2^{(x_j)} \mathcal{M}_2^{(x_j)^T} \right) \right] P_j \mathbf{E} \right) + \parallel \mathbf{E} \parallel \parallel \mathbf{T} \parallel . \tag{4.1.20}$$

And as in the two-dimensional case we require that for all j, $j = 1, \ldots, d$ the matrices

$$\left[c_{jj} \mathcal{M}^{(x_j x_j)} + \frac{1}{2} \sum_{i \neq j}^{d} c_{ij}^{(j)^2} \left(\mathcal{M}_1^{(x_j)^T} \mathcal{M}_1^{(x_j)} + \mathcal{M}_2^{(x_j)} \mathcal{M}_2^{(x_j)^T} \right) \right], \tag{4.1.21}$$

or, equivalently, their diagonal-blocks

$$\left[c_{jj} M_k^{(x_j x_j)} + \frac{1}{2} \sum_{i \neq j}^{d} c_{ij}^{(j)^2} \left(M_{1_k}^{(x_j)^T} M_{1_k}^{(x_j)} + M_{2_k}^{(x_j)} M_{2_k}^{(x_j)^T} \right) \right], \tag{4.1.22}$$

be negative definite and bounded away from 0. It is shown in the next section that this requirement is more difficult to satisfy than in the two-dimensional case.

4.2 Second order scheme for the scalar mixed derivatives problem

In this section we construct a second order accuracy scheme to the problem (4.1.3) using the method presented in the previous section.

The matrices $M_k^{(xx)} = D_k^{(xx)} - \tau_{L_k} A_{L_k} - \tau_{R_k} A_{R_k}$ are constructed from:

$$D_k^{(xx)} = \frac{1}{h^2} \begin{bmatrix} 1 & -2 & 1 & 0 & & & & & \\ 1 & -2 & 1 & 0 & & & & & \\ 0 & 1 & -2 & 1 & & & & & \\ 0 & 0 & 1 & -2 & 1 & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & 1 & -2 & 1 & 0 & 0 \\ & & & & & 1 & -2 & 1 & 0 \\ & & & & & & 1 & -2 & 1 \\ & & & & & & 1 & -2 & 1 \end{bmatrix}; \tag{4.2.1}$$

For simplicity we take

$$\tau_{L_k} = \frac{1}{h^2} \text{diag} \left[\frac{7 - 5\gamma_L}{2}, 0, \ldots, 0, 0 \right]; \tag{4.2.2}$$

$$\tau_{R_k} = \frac{1}{h^2} \text{diag} \left[0, 0, \ldots, 0, \frac{7 - 5\gamma_R}{2} \right]; \tag{4.2.3}$$

$$A_{L_k} = \begin{bmatrix} \frac{1}{2}(2+\gamma_L)(1+\gamma_L) & -\gamma_L(2+\gamma_L) & \frac{1}{2}(\gamma_L + \gamma_L^2) & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{2}(2+\gamma_L)(1+\gamma_L) & -\gamma_L(2+\gamma_L) & \frac{1}{2}(\gamma_L + \gamma_L^2) & 0 & \ldots & 0 \end{bmatrix}; \tag{4.2.4}$$

$$A_{R_k} = \begin{bmatrix} 0 & \ldots & 0 & \frac{1}{2}(\gamma_R + \gamma_R^2) & -\gamma_R(2+\gamma_R) & \frac{1}{2}(2+\gamma_R)(1+\gamma_R) \\ \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & 0 & \frac{1}{2}(\gamma_R + \gamma_R^2) & -\gamma_R(2+\gamma_R) & \frac{1}{2}(2+\gamma_R)(1+\gamma_R) \end{bmatrix}. \tag{4.2.5}$$

The matrices $M_{1_k}^{(x)} = D_{1_k}^{(x)} - \sigma_{L_k} A_{L_k} - \sigma_{R_k} A_{R_k}$ are composed of:

$$D_{1_k}^{(x)} = \frac{1}{2h}\begin{bmatrix} -3+c_1 & 4-3c_1 & -1+3c_1 & -c_1 & & & & & & \\ -1 & 0 & 1 & & & & & & & \\ & -1 & 0 & 1 & & & & & & \\ & & -1 & 0 & 1 & & & & & \\ & & & \ddots & \ddots & \ddots & & & & \\ & & & & -1 & 0 & 1 & 0 & & \\ & & & & & -1 & 0 & 1 & 0 & \\ & & & & & & -1 & 0 & 1 & 0 \\ & & & & & c_N & 1-3c_N & -4+3c_N & 3-c_N & \end{bmatrix}; \tag{4.2.6}$$

Where $c_1 = c_N = 1$.

$$\sigma_{L_k} = \frac{1}{h^2}\text{diag}\left[\sigma_{L_1}, \sigma_{L_2}, 0, \ldots, 0, 0\right]; \tag{4.2.7}$$

$$\sigma_{R_k} = \frac{1}{h^2}\text{diag}\left[0, 0, \ldots, 0, \sigma_{R_{N-1}}, \sigma_{R_N}\right]; \tag{4.2.8}$$

but we take $\sigma_{L_1} = \sigma_{L_2} = \sigma_{R_{N-1}} = \sigma_{R_N} = 0$, i.e. we don't use the penalty terms for $M_{1_k}^{(x)}$.

Finally the matrices $M_{2_k}^{(x)} = D_{2_k}^{(x)}$ are given by:

$$D_{2_k}^{(x)} = \frac{1}{2h}\begin{bmatrix} -2 & 2 & & & & & \\ -1 & 0 & 1 & & & & \\ & -1 & 0 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 0 & 1 & 0 \\ & & & & -1 & 0 & 1 \\ & & & & & -2 & 2 \end{bmatrix} \tag{4.2.9}$$

The definitions of the matrices $M_j^{(yy)}$,$M_{1_j}^{(y)}$ and $M_{2_j}^{(y)}$ are analogous.

After substituting (4.2.1) to (4.2.9) into (4.1.17) and denoting by $\tilde{M}$ the symmetric part of this matrix one gets,

$$\tilde{M} = \frac{1}{16h^2}\begin{bmatrix}
m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & & & & & & & \\
m_{1,2} & m_{2,2} & m_{2,3} & m_{2,4} & & & 0 & & & & \\
m_{1,3} & m_{2,3} & m_{3,3} & m_{3,4} & m_2 & & & & & & \\
m_{1,4} & m_{2,4} & m_{3,4} & m_{4,4} & m_1 & m_2 & & & & & \\
 & & m_2 & m_1 & m_0 & m_1 & m_2 & & & & \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\
 & & & & m_2 & m_1 & m_0 & m_1 & m_2 & & \\
 & & & & & m_2 & m_1 & m_{N-3,N-3} & m_{N-2,N-3} & m_{N-1,N-3} & m_{N,N-3} \\
 & & & & & & m_2 & m_{N-2,N-3} & m_{N-2,N-2} & m_{N-1,N-2} & m_{N,N-2} \\
 & & & & 0 & & & m_{N-1,N-3} & m_{N-1,N-2} & m_{N-1,N-1} & m_{N,N-1} \\
 & & & & & & & m_{N,N-3} & m_{N,N-2} & m_{N,N-1} & m_{N,N}
\end{bmatrix}, \tag{4.2.10}$$

where

$$\begin{aligned}
m_0 &= 8 - 32(1+\mu_1^2) \\
m_1 &= 16(1+\mu_1^2) \\
m_2 &= -4,
\end{aligned}$$

$$\begin{aligned}
m_{1,1} &= 26 - 4(1+\mu_1^2)(10 + 11\gamma_L - 8\gamma_L^2 - 5\gamma_L^3) \\
m_{1,2} &= -4(1+\mu_1^2)(2 - 14\gamma_L + 3\gamma_L^2 + 5\gamma_L^3) \\
m_{1,3} &= -14 + 2(1+\mu_1^2)(4 - 7\gamma_L - 2\gamma_L^2 + 5\gamma_L^3) \\
m_{1,4} &= 4 \\
m_{2,2} &= 8 - 32(1+\mu_1^2)
\end{aligned}$$

$$
\begin{aligned}
m_{2,3} &= 4 + 16(1+\mu_1^2) \\
m_{2,4} &= -6 \\
m_{3,3} &= 16 - 32(1+\mu_1^2) \\
m_{3,4} &= -4 + 16(1+\mu_1^2) \\
m_{4,4} &= 10 - 32(1+\mu_1^2)
\end{aligned}
$$

and

$$
\begin{aligned}
m_{N,N} &= 26 - 4(1+\mu_2^2)(10 + 11\gamma_R - 8\gamma_R^2 - 5\gamma_R^3) \\
m_{N,N-1} &= -4(1+\mu_2^2)(2 - 14\gamma_R + 3\gamma_R^2 + 5\gamma_R^3) \\
m_{N,N-2} &= -14 + 2(1+\mu_2^2)(4 - 7\gamma_R - 2\gamma_R^2 + 5\gamma_R^3) \\
m_{N,N-3} &= 4 \\
m_{N-1,N-1} &= 8 - 32(1+\mu_2^2) \\
m_{N-1,N-2} &= 4 + 16(1+\mu_2^2) \\
m_{N-1,N-3} &= -6 \\
m_{N-2,N-2} &= 16 - 32(1+\mu_2^2) \\
m_{N-2,N-3} &= -4 + 16(1+\mu_2^2) \\
m_{N-3,N-3} &= 10 - 32(1+\mu_2^2)
\end{aligned}
$$

We now decompose $\tilde{M}$ as follows:

$$
\tilde{M} = \frac{1}{16h^2}\left[16\mu_1^2\alpha\tilde{M}_1 + 16\mu_1^2(1-\alpha)\tilde{M}_2 + 4\tilde{M}_3 + \tilde{M}_4 + \tilde{M}_5\right] \tag{4.2.11}
$$

where:

$$
\tilde{M}_1 = \begin{bmatrix}
-2 & 1 & & & & & & \\
1 & -2 & 1 & & & & & \\
 & 1 & -2 & 1 & & & & \\
 & & \ddots & \ddots & \ddots & & & \\
 & & & & 1 & -2 & 1 & \\
 & & & & & 1 & -2 & 1 \\
 & & & & & & 1 & -2
\end{bmatrix}, \tag{4.2.12}
$$

$$\tilde{M}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & & & & & & & \\ 0 & 0 & 0 & 0 & & & & & & & \\ 0 & 0 & 0 & 0 & & & & & & & \\ 0 & 0 & 0 & -1 & 1 & & & & & & \\ & & & 1 & -2 & 1 & & & & & \\ & & & & \ddots & \ddots & \ddots & & & & \\ & & & & & 1 & -2 & 1 & & & \\ & & & & & 0 & 1 & -1 & 0 & 0 & 0 \\ & & & & & & & 0 & 0 & 0 & 0 \\ & & & & & & & 0 & 0 & 0 & 0 \\ & & & & & & & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.2.13}$$

$$\tilde{M}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & & & & & & & \\ 0 & 0 & 0 & 0 & & & & & & 0 & \\ 0 & 0 & -1 & 2 & -1 & & & & & & \\ 0 & 0 & 2 & -5 & 4 & -1 & & & & & \\ & & -1 & 4 & -6 & 4 & -1 & & & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & & -1 & 4 & -6 & 4 & -1 & & \\ & & & & & -1 & 4 & -5 & 2 & 0 & 0 \\ & 0 & & & & & -1 & 2 & -1 & 0 & 0 \\ & & & & & & & 0 & 0 & 0 & 0 \\ & & & & & & & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.2.14}$$

$$\tilde{M}_4 = \frac{1}{16h^2} \left[\begin{array}{c|c|c|c} +m_{1,1} \\ 32\mu_1^2\alpha & m_{1,2} \\ & -16\mu_1^2\alpha & m_{1,3} & m_{1,4} \\ \hline m_{1,2} & m_{2,2} & m_{2,3} & m_{2,4} \\ -16\mu_1^2\alpha & +32\mu_1^2\alpha & -16\mu_1^2\alpha & \\ \hline m_{1,3} & m_{2,3} & m_{3,3} & m_{3,4} \\ & -16\mu_1^2\alpha & +32\mu_1^2\alpha + 4 & -16\mu_1^2\alpha - 8 \\ \hline m_{1,4} & m_{2,4} & m_{3,4} & m_{4,4} \\ & & -16\mu_1^2\alpha - 8 & +16(1+\alpha)\mu_1^2 + 20 \\ \\ & & 0 & \end{array}\right] \begin{matrix} \\ \\ \\ 0 \\ \\ \\ \\ 0 \end{matrix} \tag{4.2.15}$$

and

$$\tilde{M}_5 = \frac{1}{16h^2}\left[\begin{array}{c|c|c|c} 0 & 0 & & \\ \hline & \begin{array}{c} m_{N-3,N-3} \\ +16(1+\alpha)\mu_1^2+20 \end{array} & \begin{array}{c} m_{N-2,N-3} \\ -16\mu_1^2\alpha-8 \end{array} & m_{N-1,N-3} \quad m_{N,N-3} \\ \hline & \begin{array}{c} m_{N-2,N-3} \\ -16\mu_1^2\alpha-8 \end{array} & \begin{array}{c} m_{N-2,N-2} \\ +32\mu_1^2\alpha+4 \end{array} & \begin{array}{c} m_{N-1,N-2} \\ -16\mu_1^2\alpha \end{array} \quad m_{N,N-2} \\ \hline 0 & m_{N-1,N-3} & \begin{array}{c} m_{N-1,N-2} \\ -16\mu_1^2\alpha \end{array} & \begin{array}{c} m_{N-1,N-1} \\ +32\mu_1^2\alpha \end{array} \quad \begin{array}{c} m_{N,N-1} \\ -16\mu_1^2\alpha \end{array} \\ \hline & m_{N,N-3} & m_{N,N-2} & \begin{array}{c} m_{N,N-1} \\ -16\mu_1^2\alpha \end{array} \quad \begin{array}{c} m_{N,N} \\ +32\mu_1^2\alpha \end{array} \end{array}\right] . \tag{4.2.16}$$

$\tilde{M}_1$ is negative definite by the argument leading to eq. (2.4.31), in the appendix to chapter 2. By the same argument it immediately follows that its largest eigenvalue is smaller than $-h^2\pi^2$. $\tilde{M}_2$ and $\tilde{M}_3$ are non-positive definite, see equations (2.4.32) to (2.4.35) in that appendix. The matrices $\tilde{M}_4$ and $\tilde{M}_5$ are non positive definite for all $0 \leq \gamma_L, \gamma_R \leq 1$, $0 \leq \alpha \leq \alpha_{\max}(\mu_1)$. For $\mu_1^2 \leq \mu_{\min}^2 \approx 0.4342$ there is no positive α. If, for example, we take $\mu_1^2 = 1/2$ α ranges from 0 to 0.02 . For $\mu_1^2 > 1/2$ $\alpha_{\max}$ increases with μ_1^2. These results were verified using the Mathematica software. We get the same expressions for the differentiation matrices in the y direction, the only difference being that in the y direction we have μ_2 instead of μ_1. Thus we have proven that if $\mu_1^2, \mu_2^2 \geq 1/2$ then from (4.1.14), (4.2.11) and the fact that $\tilde{M}_1$ is bounded away from 0 by $-h^2\pi^2$ it follows that

$$\begin{aligned} \frac{1}{2}\frac{d}{dt} \parallel \mathbf{E} \parallel^2 \quad &\leq \quad -0.02\left[a^2(\mu_1^2 - 1/2) + b^2(\mu_2^2 - 1/2)\right](\mathbf{E}, \mathbf{E}) + \\ &\qquad \parallel \mathbf{E} \parallel \parallel \mathbf{T} \parallel . \end{aligned}$$

Then by using the definition:

$$c_0 = 0.02\left[a^2(\mu_1^2 - 1/2) + b^2(\mu_2^2 - 1/2)\right] ,$$

we get that

$$\| \mathbf{E} \| \leq \frac{\| \mathbf{T} \|_M}{c_0}(1 - e^{-c_0 t}) \tag{4.2.17}$$

where the *"constant"* $\| \mathbf{T} \|_M = \max_{0 \leq \tau \leq t} \| \mathbf{T}(\tau) \|$. This *"constant"* is a function of the exact solution u and its derivatives.

The value of the maximal α, ($\alpha_{\max} = 0.02$), and the minimal value of μ_1^2 and μ_2^2, (1/2), could probably be improved by using different values for the coefficients τ's ,σ's and c's in (4.2.2), (4.2.3) and (4.2.6) to (4.2.8).

It should be noted that when we write equation (4.1.1) in the form (4.1.2), then in order to prove the negative-definiteness of the discrete differentiation operator in (4.1.8), the diagonal of the matrix C should be very large with respect to the off-diagonal terms (by a factor of 1.5). This problem limits the usefulness of this approach, especially in the case of many space dimensions, $d \geq 3$, where the ratio between the diagonal and off-diagonal terms becomes larger. For example, if the matrix C has the form:

$$c_{ij} = \begin{cases} c_i c_j & \text{if } i \neq j \\ \\ s c_j & \text{if } i = j \end{cases} ,$$

then by equating eq. (4.1.22) to eq. (4.1.17) one can see that s should be at least $1.5(d-1)$.

In the general case, see equation (4.1.20), a ratio between the coefficient of $\mathcal{M}^{(x_j x_j)}$ and the coefficient of $\left(\mathcal{M}_1^{(x_j)^T} \mathcal{M}_1^{(x_j)} + \mathcal{M}_2^{(x_j)} \mathcal{M}_2^{(x_j)^T} \right)$ of least 3 will assure the negative definiteness. When this ratio is grater the 3 we can bound the error-norm by a constant using the same considerations leading to eq. (4.1.17).

4.3 Parabolic systems; the diffusion part of the Navier-Stokes equations

The analysis of general parabolic systems is much more complex than that for the scalar equations. It can be carried out for some important parabolic systems, such as diffusion part of the Navier-Stokes equations, which will now be considered.

Consider the problem

$$\frac{\partial u^{(1)}}{\partial t} = \frac{4}{3} u_{xx}^{(1)} + u_{yy}^{(1)} + \frac{1}{3} u_{xy}^{(2)}; \qquad (x, y) \in \Omega; \quad t \geq 0$$

$$
\begin{aligned}
\frac{\partial u^{(2)}}{\partial t} &= u^{(2)}_{xx} + \frac{4}{3}u^{(2)}_{yy} + \frac{1}{3}u^{(1)}_{xy}; \qquad (x,y) \in \Omega; \quad t \geq 0 \\
u^{(1)}(x,y,0) &= u^{(1)}_0(x,y); \qquad (x,y) \in \Omega \\
u^{(2)}(x,y,0) &= u^{(2)}_0(x,y); \qquad (x,y) \in \Omega \\
u^{(1)}(x,y,t)|_{\partial\Omega} &= u^{(1)}_B(t) \\
u^{(2)}(x,y,t)|_{\partial\Omega} &= u^{(2)}_B(t)
\end{aligned}
\tag{4.3.1}
$$

where the Reynolds number, R_e, has been absorbed in the temporal variable t.

Let us discretize (4.3.1) in the same way as we discretized the scalar equation (4.1.3).

$$
\begin{aligned}
\frac{d\mathbf{U^{(1)}}}{dt} &= \frac{4}{3}\mathcal{D}^{(xx)}\mathbf{U^{(1)}} + P^T\mathcal{D}^{(yy)}P\mathbf{U^{(1)}} + \\
&\quad \frac{1}{6}\left[\mathcal{D}^{(x)}_2\ P^T\mathcal{D}^{(y)}_1 P + P^T\mathcal{D}^{(y)}_2 P\ \mathcal{D}^{(x)}_1\right]\mathbf{U^{(2)}} + \mathbf{T^{(1)}} \\
\frac{d\mathbf{U^{(2)}}}{dt} &= \mathcal{D}^{(xx)}\mathbf{U^{(2)}} + \frac{4}{3}P^T\mathcal{D}^{(yy)}P\mathbf{U^{(2)}} + \\
&\quad \frac{1}{6}\left[\mathcal{D}^{(x)}_2\ P^T\mathcal{D}^{(y)}_1 P + P^T\mathcal{D}^{(y)}_2 P\ \mathcal{D}^{(x)}_1\right]\mathbf{U^{(1)}} + \mathbf{T^{(2)}}
\end{aligned}
\tag{4.3.2}
$$

The definitions for the $\mathcal{D}$'s and P are given in (4.1.5) to (4.1.7). We can now write the scheme, analogous to equation (4.1.8).

$$
\begin{aligned}
\frac{d\mathbf{V^{(1)}}}{dt} &= \frac{4}{3}\mathcal{M}^{(xx)}\mathbf{V^{(1)}} + P^T\mathcal{M}^{(yy)}P\mathbf{V^{(1)}} + \\
&\quad \frac{1}{6}\left[\mathcal{M}^{(x)}_2\ P^T\mathcal{M}^{(y)}_1 P + P^T\mathcal{M}^{(y)}_2 P\ \mathcal{M}^{(x)}_1\right]\mathbf{V^{(2)}} + \mathbf{G}^{(x)}_{11} + P^T\mathbf{G}^{(y)}_{11} + \mathbf{G}^{(x)}_{12} + P^T\mathbf{G}^{(y)}_{12} \\
\frac{d\mathbf{V^{(2)}}}{dt} &= \mathcal{M}^{(xx)}\mathbf{V^{(2)}} + \frac{4}{3}P^T\mathcal{M}^{(yy)}P\mathbf{V^{(2)}} + \\
&\quad \frac{1}{6}\left[\mathcal{M}^{(x)}_2\ P^T\mathcal{M}^{(y)}_1 P + P^T\mathcal{M}^{(y)}_2 P\ \mathcal{M}^{(x)}_1\right]\mathbf{V^{(1)}} + \mathbf{G}^{(x)}_{21} + P^T\mathbf{G}^{(y)}_{21} + \mathbf{G}^{(x)}_{22} + P^T\mathbf{G}^{(y)}_{22}
\end{aligned}
\tag{4.3.3}
$$

where the $\mathcal{M}$'s are given in (4.1.9), the $\mathbf{G}$'s are analogous to those in equation (4.1.10), and the first number in the subscripts indicate in which equation the $\mathbf{G}$'s are and the second one indicate whether the boundary conditions are of $u^{(1)}$ or $u^{(2)}$. For example $\mathbf{G}^{(x)}_{21}$ reflects the boundary values for $u^{(1)}$ in the x direction (rows) and it appears in the second equation. After subtracting (4.3.2) from (4.3.3) we get:

$$
\begin{aligned}
\frac{d\mathbf{E}^{(1)}}{dt} &= \frac{4}{3}\mathcal{M}^{(xx)}\mathbf{E}^{(1)} + P^T\mathcal{M}^{(yy)}P\mathbf{E}^{(1)} + \\
&\quad \frac{1}{6}\left[\mathcal{M}_2^{(x)}\ P^T\mathcal{M}_1^{(y)}P + P^T\mathcal{M}_2^{(y)}P\ \mathcal{M}_1^{(x)}\right]\mathbf{E}^{(2)} + \mathbf{T}^{(1)} \\
\frac{d\mathbf{E}^{(2)}}{dt} &= \mathcal{M}^{(xx)}\mathbf{E}^{(2)} + \frac{4}{3}P^T\mathcal{M}^{(yy)}\mathbf{E}^{(2)}P + \\
&\quad \frac{1}{6}\left[\mathcal{M}_2^{(x)}\ P^T\mathcal{M}_1^{(y)}P + P^T\mathcal{M}_2^{(y)}P\ \mathcal{M}_1^{(x)}\right]\mathbf{E}^{(1)} + \mathbf{T}^{(2)}
\end{aligned}
\tag{4.3.4}
$$

Taking the scalar product of $\mathbf{E}^{(1)}$ with the first equation and of $\mathbf{E}^{(2)}$ with the second one one gets after adding the two equations:

$$
\begin{aligned}
\frac{1}{2}\frac{d}{dt}\left(\| \mathbf{E}^{(1)} \|^2 + \| \mathbf{E}^{(2)} \|^2\right) &= \frac{4}{3}\left(\mathbf{E}^{(1)}, \mathcal{M}^{(xx)}\mathbf{E}^{(1)}\right) + \left(\mathbf{E}^{(1)}, P^T\mathcal{M}^{(yy)}P\mathbf{E}^{(1)}\right) + \\
&\quad \frac{1}{6}\left(\mathbf{E}^{(1)}, \left[\mathcal{M}_2^{(x)}\ P^T\mathcal{M}_1^{(y)}P + P^T\mathcal{M}_2^{(y)}P\ \mathcal{M}_1^{(x)}\right]\mathbf{E}^{(2)}\right) + \\
&\quad \left(\mathbf{E}^{(2)}, \mathcal{M}^{(xx)}\mathbf{E}^{(2)}\right) + \frac{4}{3}\left(\mathbf{E}^{(2)}, P^T\mathcal{M}^{(yy)}P\mathbf{E}^{(2)}\right) + \\
&\quad \frac{1}{6}\left(\mathbf{E}^{(2)}, \left[\mathcal{M}_2^{(x)}\ P^T\mathcal{M}_1^{(y)}P + P^T\mathcal{M}_2^{(y)}P\ \mathcal{M}_1^{(x)}\right]\mathbf{E}^{(1)}\right) + \\
&\quad + \left(\mathbf{E}^{(1)}, \mathbf{T}^{(1)}\right) + \left(\mathbf{E}^{(2)}, \mathbf{T}^{(2)}\right)
\end{aligned}
\tag{4.3.5}
$$

Then after doing the same manipulations leading to equation (4.1.14) we get:

$$
\begin{aligned}
\frac{1}{2}\frac{d}{dt}\left(\| \mathbf{E}^{(1)} \|^2 + \| \mathbf{E}^{(2)} \|^2\right) &\leq \left(\mathbf{E}^{(1)}, \left[\frac{4}{3}\mathcal{M}^{(xx)} + \frac{1}{12}\left(\mathcal{M}_1^{(x)^T}\mathcal{M}_1^{(x)} + \mathcal{M}_2^{(x)}\mathcal{M}_2^{(x)^T}\right)\right]\mathbf{E}^{(1)}\right) + \\
&\quad \left(\mathbf{E}^{(1)}, P^T\left[\mathcal{M}^{(yy)} + \frac{1}{12}\left(\mathcal{M}_1^{(y)^T}\mathcal{M}_1^{(y)} + \mathcal{M}_2^{(y)}\mathcal{M}_2^{(y)^T}\right)\right]P\mathbf{E}^{(1)}\right) + \\
&\quad \left(\mathbf{E}^{(2)}, \left[\mathcal{M}^{(xx)} + \frac{1}{12}\left(\mathcal{M}_1^{(x)^T}\mathcal{M}_1^{(x)} + \mathcal{M}_2^{(x)}\mathcal{M}_2^{(x)^T}\right)\right]\mathbf{E}^{(2)}\right) + \\
&\quad \left(\mathbf{E}^{(2)}, P^T\left[\frac{4}{3}\mathcal{M}^{(yy)} + \frac{1}{12}\left(\mathcal{M}_1^{(y)^T}\mathcal{M}_1^{(y)} + \mathcal{M}_2^{(y)}\mathcal{M}_2^{(y)^T}\right)\right]P\mathbf{E}^{(2)}\right) + \\
&\quad \| \mathbf{E}^{(1)} \| \| \mathbf{T}^{(1)} \| + \| \mathbf{E}^{(2)} \| \| \mathbf{T}^{(2)} \| .
\end{aligned}
\tag{4.3.6}
$$

We can now define the error vector $\mathbf{E}$ as a $2N$ long vector whose first N entries are the entries of $\mathbf{E^{(1)}}$ and the other N entries are the ones of $\mathbf{E^{(2)}}$. In a similar way we can define the truncation-error vector, $\mathbf{T}$. Using these definitions the norms of $\mathbf{E}$ and $\mathbf{T}$ are:

$$\| \mathbf{E} \| = \frac{1}{\sqrt{2}} \sqrt{\| \mathbf{E^{(1)}} \|^2 + \| \mathbf{E^{(2)}} \|^2} \tag{4.3.7}$$

and

$$\| \mathbf{T} \| = \frac{1}{\sqrt{2}} \sqrt{\| \mathbf{T^{(1)}} \|^2 + \| \mathbf{T^{(2)}} \|^2}. \tag{4.3.8}$$

If we use the second order differentiation matrices presented in the previous section, see equations (4.2.1) to (4.2.9), then using the arguments presented in the last paragraph of that section and using, e.g., $\frac{4}{3} = \frac{1}{6}(1+\mu_1^2)$, etc, we may write the following inequality:

$$\frac{1}{2}\frac{d}{dt} \| \mathbf{E} \|^2 \leq -0.02\frac{11}{6} \| \mathbf{E} \|^2 + \| \mathbf{E} \| \| \mathbf{T} \| \tag{4.3.9}$$

Now by using the definition:

$$c_0 = 0.02\frac{11}{6},$$

we get that

$$\| \mathbf{E} \| \leq \frac{\| \mathbf{T} \|_M}{c_0}(1 - e^{-c_0 t}) \tag{4.3.10}$$

where the *"constant"* $\| \mathbf{T} \|_M = \max_{0 \leq \tau \leq t} \| \mathbf{T}(\tau) \|$. This *"constant"* is function of the exact solution $(u^{(1)}, u^{(2)})^T$ and its derivatives.

We can also apply the same procedure to the diffusion part of the Navier-Stokes equations in three-dimensions.

Consider the problem

$$
\begin{aligned}
\frac{\partial u^{(1)}}{\partial t} &= \frac{4}{3}u^{(1)}_{xx} + u^{(1)}_{yy} + u^{(1)}_{zz} + \frac{1}{3}u^{(2)}_{xy} + \frac{1}{3}u^{(3)}_{xz}; \qquad (x,y,z) \in \ \Omega; \ \ t \geq 0 \\
\frac{\partial u^{(2)}}{\partial t} &= u^{(2)}_{xx} + \frac{4}{3}u^{(2)}_{yy} + u^{(2)}_{zz} + \frac{1}{3}u^{(1)}_{xy} + \frac{1}{3}u^{(3)}_{yz}; \qquad (x,y,z) \in \ \Omega; \ \ t \geq 0 \\
\frac{\partial u^{(3)}}{\partial t} &= u^{(3)}_{xx} + u^{(3)}_{yy} + \frac{4}{3}u^{(3)}_{zz} + \frac{1}{3}u^{(1)}_{xz} + \frac{1}{3}u^{(2)}_{yz}; \qquad (x,y,z) \in \ \Omega; \ \ t \geq 0 \\
u^{(1)}(x,y,z,0) &= \ u^{(1)}_0(x,y,z); \qquad (x,y,z) \in \ \Omega \\
u^{(2)}(x,y,z,0) &= \ u^{(2)}_0(x,y,z); \qquad (x,y,z) \in \ \Omega \\
u^{(3)}(x,y,z,0) &= \ u^{(3)}_0(x,y,z); \qquad (x,y,z) \in \ \Omega \\
u^{(1)}(x,y,z,t)|_{\partial\Omega} &= u^{(1)}_B(t) \\
u^{(2)}(x,y,z,t)|_{\partial\Omega} &= u^{(2)}_B(t) \\
u^{(3)}(x,y,z,t)|_{\partial\Omega} &= u^{(3)}_B(t)
\end{aligned}
\tag{4.3.11}
$$

We write the scheme, analogous to equation (4.3.3).

$$
\begin{aligned}
\frac{d\mathbf{V^{(1)}}}{dt} \ = \ & \frac{4}{3}\mathcal{M}^{(xx)}\mathbf{V^{(1)}} + P_2^T\mathcal{M}^{(yy)}P_2\mathbf{V^{(1)}} + P_3^T\mathcal{M}^{(zz)}P_3\mathbf{V^{(1)}} + \\
& \frac{1}{6}\left[\mathcal{M}_2^{(x)}\ P_2^T\mathcal{M}_1^{(y)}P_2 + P_2^T\mathcal{M}_2^{(y)}P_2\ \mathcal{M}_1^{(x)}\right]\mathbf{V^{(2)}} + \\
& \frac{1}{6}\left[\mathcal{M}_2^{(x)}\ P_3^T\mathcal{M}_1^{(z)}P_3 + P_3^T\mathcal{M}_2^{(z)}P_3\ \mathcal{M}_1^{(x)}\right]\mathbf{V^{(3)}} + \\
& \mathbf{G}_{11}^{(x)} + P_2^T\mathbf{G}_{11}^{(y)} + P_3^T\mathbf{G}_{11}^{(z)} + \\
& \mathbf{G}_{12}^{(x)} + P_2^T\mathbf{G}_{12}^{(y)} + P_3^T\mathbf{G}_{12}^{(z)} + \\
& \mathbf{G}_{13}^{(x)} + P_2^T\mathbf{G}_{13}^{(y)} + P_3^T\mathbf{G}_{13}^{(z)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{d\mathbf{V^{(2)}}}{dt} &= \mathcal{M}^{(xx)}\mathbf{V^{(2)}} + \frac{4}{3}P_2^T\mathcal{M}^{(yy)}P_2\mathbf{V^{(2)}} + P_3^T\mathcal{M}^{(zz)}P_3\mathbf{V^{(2)}} + \\
&\frac{1}{6}\left[\mathcal{M}_2^{(x)}\ P_2^T\mathcal{M}_1^{(y)}P_2 + P_2^T\mathcal{M}_2^{(y)}P_2\ \mathcal{M}_1^{(x)}\right]\mathbf{V^{(1)}} + \\
&\frac{1}{6}\left[P_2^T\mathcal{M}_2^{(y)}P_2\ P_3^T\mathcal{M}_1^{(z)}P_3 + P_3^T\mathcal{M}_2^{(z)}P_3\ P_2^T\mathcal{M}_1^{(y)}P_2\right]\mathbf{V^{(3)}} + \\
&\mathbf{G}_{21}^{(x)} + P_2^T\mathbf{G}_{21}^{(y)} + P_3^T\mathbf{G}_{21}^{(z)} + \\
&\mathbf{G}_{22}^{(x)} + P_2^T\mathbf{G}_{22}^{(y)} + P_3^T\mathbf{G}_{22}^{(z)} + \\
&\mathbf{G}_{23}^{(x)} + P_2^T\mathbf{G}_{23}^{(y)} + P_3^T\mathbf{G}_{23}^{(z)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{d\mathbf{V^{(3)}}}{dt} &= \mathcal{M}^{(xx)}\mathbf{V^{(3)}} + P_2^T\mathcal{M}^{(yy)}P_2\mathbf{V^{(3)}} + \frac{4}{3}P_3^T\mathcal{M}^{(zz)}P_3\mathbf{V^{(3)}} + \\
&\frac{1}{6}\left[\mathcal{M}_2^{(x)}\ P_3^T\mathcal{M}_1^{(z)}P_3 + P_3^T\mathcal{M}_2^{(z)}P_3\ \mathcal{M}_1^{(x)}\right]\mathbf{V^{(1)}} + \\
&\frac{1}{6}\left[P_2^T\mathcal{M}_2^{(y)}P_2\ P_3^T\mathcal{M}_1^{(z)}P_3 + P_3^T\mathcal{M}_2^{(z)}P_3\ P_2^T\mathcal{M}_1^{(y)}P_2\right]\mathbf{V^{(2)}} + \\
&\mathbf{G}_{31}^{(x)} + P_2^T\mathbf{G}_{31}^{(y)} + P_3^T\mathbf{G}_{31}^{(z)} + \\
&\mathbf{G}_{32}^{(x)} + P_2^T\mathbf{G}_{32}^{(y)} + P_3^T\mathbf{G}_{32}^{(z)} + \\
&\mathbf{G}_{33}^{(x)} + P_2^T\mathbf{G}_{33}^{(y)} + P_3^T\mathbf{G}_{33}^{(z)}
\end{aligned}
$$

(4.3.12)

Then after doing the same manipulations leading to equation (4.3.9) and the analogous definitions for $\mathbf{E}$ and $\mathbf{T}$,

$$\| \mathbf{E} \| = \frac{1}{\sqrt{3}}\sqrt{\| \mathbf{E^{(1)}} \|^2 + \| \mathbf{E^{(2)}} \|^2 + \| \mathbf{E^{(3)}} \|^2}$$

and

$$\| \mathbf{T} \| = \frac{1}{\sqrt{3}}\sqrt{\| \mathbf{T^{(1)}} \|^2 + \| \mathbf{T^{(2)}} \|^2 + \| \mathbf{T^{(3)}} \|^2},$$

we get, as before:

$$\frac{1}{2}\frac{d}{dt} \parallel \mathbf{E} \parallel^2 \leq -0.02\frac{11}{6} \parallel \mathbf{E} \parallel^2 + \parallel \mathbf{E} \parallel \parallel \mathbf{T} \parallel \tag{4.3.13}$$

Now by using the definition:

$$c_0 = 0.02\frac{11}{6},$$

we get that

$$\parallel \mathbf{E} \parallel \leq \frac{\parallel \mathbf{T} \parallel_M}{c_0}(1 - e^{-c_0 t}) \tag{4.3.14}$$

where the *"constant"* $\parallel \mathbf{T} \parallel_M = \max_{0 \leq \tau \leq t} \parallel \mathbf{T}(\tau) \parallel$.

Chapter 5

Bounded error schemes for the wave equation

5.1 Description of the method

We consider the following problem

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + f(x,t); \qquad \Gamma_L \leq x \leq \Gamma_R, \;\; t \geq 0 \tag{5.1.1a}$$

$$u(x,0) = u_0(x) \tag{5.1.1b}$$

$$\frac{\partial}{\partial t} u(x,0) = u_{t_0}(x) \tag{5.1.1c}$$

$$u(\Gamma_L, t) = g_L(t) \tag{5.1.1d}$$

$$u(\Gamma_R, t) = g_R(t) \tag{5.1.1e}$$

and $f(x,t) \in C^2$.

Let us discretize (5.1.1) spatially on the same grid and use the same notation as in in the previous chapters.

We would like to construct a semi-discrete finite difference scheme to solve (5.1.1) without first writing it as a system of first order P.D.E's. The reason for constructing a direct solver for the wave-equation is demonstrated for the two-dimensional case:

Consider,

$$\begin{aligned}
\frac{\partial^2 u}{\partial t^2} &= \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \\
u(x,y,t) &= u_B(x,y,t), \quad (x,y) \in \partial\Omega \\
u(x,y,0) &= u_0(x,y) \\
\frac{\partial}{\partial t} u(x,y,0) &= u_{t_0}(x,y)
\end{aligned}$$

Converting the above to a system of first order P.D.E's, using the standard substitution: $v^I = u_t; v^{II} = u_x; v^{III} = u_y$ we get:

$$\frac{\partial}{\partial t}\begin{pmatrix} v^I \\ v^{II} \\ v^{III} \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{\partial}{\partial x}\begin{pmatrix} v^I \\ v^{II} \\ v^{III} \end{pmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \frac{\partial}{\partial y}\begin{pmatrix} v^I \\ v^{II} \\ v^{III} \end{pmatrix} \tag{5.1.2}$$

Unlike the one-dimensional case, where the right hand side of the system can be diagonalized, this is not the case here, as the two matrices cannot be diagonalized simultaneously. This means that boundary values must be assigned to v^{II} and v^{III}. These values cannot be derived directly from the original wave equation, nor from a procedure analogous to the 1-D characteristic decomposition. Additionally it should also be noted that if we solve the wave equation in d-dimensional space on a grid with N grid points by converting it to the d-dimensional version of (5.1.2), then after discretization we have an O.D.E system with $(1+d)N$ 'variables'. When (5.1.1) is solved by the method to be proposed shortly, see (5.1.4), one gets an O.D.E system with only N 'variables', or $2N$ 'variables' if one solves (5.1.11).

Since, unlike the previous cases, equation (5.1.1) has a second time derivative, attempts to apply naively the methods presented in chapter 1 will fail. The reason is that if we write a discrete approximation to equation (5.1.1),

$$\frac{d^2\mathbf{u}}{dt^2} = D\mathbf{u} + \mathbf{f}(t) + \mathbf{T}_e \tag{5.1.3}$$

and the numerical scheme,

$$\frac{d^2\mathbf{v}}{dt^2} = [D\mathbf{v} - \tau_L(A_L\mathbf{v} - \mathbf{g}_L) - \tau_R(A_R\mathbf{v} - \mathbf{g}_R)] + \mathbf{f}(t) \tag{5.1.4}$$

as in chapter 1, the equation for the error-vector $\boldsymbol{\epsilon}$ becomes,

$$\begin{aligned} \frac{d^2\,\boldsymbol{\epsilon}}{dt^2} &= [D\,\boldsymbol{\epsilon} - \tau_L A_L\,\boldsymbol{\epsilon} - \tau_R A_R\,\boldsymbol{\epsilon}] + \mathbf{T} \\ &= M\,\boldsymbol{\epsilon} + \mathbf{T} \end{aligned} \tag{5.1.5}$$

where

$$M = D - \tau_L A_L - \tau_R A_R$$

is a negative-definite matrix and

$$\mathbf{T} = \mathbf{T}_e - \tau_L \mathbf{T}_L - \tau_R \mathbf{T}_R = (T_1, \ldots, T_m, \ldots, T_N)^T \ .$$

If the matrix M can be diagonalized, as can be shown for the M's used in chapters 2 and 3, then

$$M = Q^{-1}\Lambda Q,$$

with the diagonal matrix, Λ, having the eigenvalues of M. Using the definition $\boldsymbol{\mu} = Q\,\boldsymbol{\epsilon}$, equation (5.1.5) becomes,

$$\begin{aligned} \frac{d^2\,\boldsymbol{\mu}}{dt^2} &= \Lambda\,\boldsymbol{\mu} + Q\mathbf{T} \\ &= \Lambda\,\boldsymbol{\mu} + \hat{\mathbf{T}}. \end{aligned} \tag{5.1.6}$$

This is an un-coupled system of O.D.E's. The general solution for the m^{th} equation, is:

$$\mu_m(t) = c_{m1} \exp\left(\sqrt{\lambda_m}\ t\right) + c_{m2} \exp\left(-\sqrt{\lambda_m}\ t\right) + \frac{1}{\sqrt{\lambda_m}} \int_0^t \sinh\left(\sqrt{\lambda_m}\ (t-s)\right) \hat{T}_m(s)\ ds \ .$$

Recalling that at $t = 0$, $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_t = \mathbf{0}$, we have, at $t = 0$, $\boldsymbol{\mu} = \boldsymbol{\mu}_t = \mathbf{0}$ and the solution for (5.1.6) is:

$$\mu_m(t) = \frac{1}{\sqrt{\lambda_m}} \int_0^t \sinh\left(\sqrt{\lambda_m}\,(t-s)\right) \hat{T}_m(s)\, ds\ . \tag{5.1.7}$$

Note that unless all the eigenvalues of M are real and non-positive some of the $\sqrt{\lambda_m}$'s will have a positive real part. In that case at least one of the μ_m may grow exponentially in time. In order to prevent this, we have to demand that M, in addition to being negative-definite, also possess only real eigenvalues. Furthermore in order to use the one-dimensional scheme as a building block for multi-dimensional schemes in the way presented in the second section of chapter 1, M should be built in a way that verifies that these properties will be carried over to the multi-dimensional differentiating matrix. One way to achieve this goal is to construct M as a negative-definite *symmetric* matrix. Then, an estimate on the error bound can be derived directly from the solution (5.1.7),

$$|\mu_m(t)| \leq \frac{1}{\sqrt{|\lambda_m|}} \hat{T}_{m_M} t$$

where $\hat{T}_{m_M} = \max_{0\leq s\leq t} |\hat{T}_m(s)|$. Then, for a normalized Q

$$\| \boldsymbol{\epsilon} \| = \| \boldsymbol{\mu} \| \leq \frac{1}{c_0} \| \hat{\mathbf{T}}_M \| t \tag{5.1.8}$$

where $c_0 = \min_{m=1,\ldots,N} \sqrt{|\lambda_m|}$. Therefore $\| \boldsymbol{\epsilon} \|$ grows at most linearly with t. Alternatively one may use an energy method in a way similar to that presented in chapter 1. Taking the scalar product of $\boldsymbol{\epsilon}_t$ with (5.1.5) one gets:

$$(\ \boldsymbol{\epsilon}_t,\ \boldsymbol{\epsilon}_{tt}) = (\ \boldsymbol{\epsilon}_t, M\ \boldsymbol{\epsilon}) + (\ \boldsymbol{\epsilon}_t, \mathbf{T}).$$

Since M is a symmetric matrix this equation can be written as:

$$\frac{1}{2}\frac{d}{dt}\left[\| \boldsymbol{\epsilon}_t \|^2 - (\ \boldsymbol{\epsilon}, M\ \boldsymbol{\epsilon})\right] = (\ \boldsymbol{\epsilon}_t, \mathbf{T})\ .$$

After integrating from 0 to t one gets:

$$\begin{aligned}\frac{1}{2}\left[\| \boldsymbol{\epsilon}_t \|^2 - (\boldsymbol{\epsilon}, M\,\boldsymbol{\epsilon})\right] &= \int_0^t (\boldsymbol{\epsilon}_s, \mathbf{T})ds \\ &= \int_0^t \left[(\boldsymbol{\epsilon}, \mathbf{T})_s - (\boldsymbol{\epsilon}, \mathbf{T}_s)\right] ds.\end{aligned}$$

Using Schwarz's inequality, $-(\boldsymbol{\epsilon}, M\,\boldsymbol{\epsilon}) \geq c_0 \| \boldsymbol{\epsilon} \|^2$ and the fact that at $t = 0$, $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_t = \mathbf{0}$ one gets:

$$\begin{aligned}\frac{1}{2}c_0 \| \boldsymbol{\epsilon} \|^2 &\leq (\boldsymbol{\epsilon}, \mathbf{T}) - \int_0^t (\boldsymbol{\epsilon}, \mathbf{T}_s)ds \\ &\leq \| \boldsymbol{\epsilon} \| \| \mathbf{T} \| + \int_0^t \| \boldsymbol{\epsilon} \| \| \mathbf{T}_s \| \, ds. \qquad (5.1.9)\end{aligned}$$

Denoting $\| \mathbf{T} \|_M = \max_{0\leq\tau\leq t} \| \mathbf{T}(\tau) \|$, $\| \mathbf{T}_t \|_M = \max_{0\leq\tau\leq t} \| \mathbf{T}_\tau(\tau) \|$, $\| \boldsymbol{\epsilon} \|_M = \max_{0\leq\tau\leq t} \| \boldsymbol{\epsilon}(\tau) \|$ and by t_M the time where $\| \boldsymbol{\epsilon} \|$ gets that maximum value, $(0 \leq t_M \leq t)$, we can evaluate equation (5.1.9) at $t = t_M$ by:

$$\| \boldsymbol{\epsilon} \|_M{}^2 \leq \frac{2}{c_0}\left[\| \boldsymbol{\epsilon} \|_M \| \mathbf{T} \|_M + \| \boldsymbol{\epsilon} \|_M \| \mathbf{T}_t \|_M \; t_M \right] .$$

After a division by $\| \boldsymbol{\epsilon} \|_M$ one gets:

$$\begin{aligned}\| \boldsymbol{\epsilon} \| &\leq \| \boldsymbol{\epsilon} \|_M \\ &\leq \frac{2}{c_0}\left[\| \mathbf{T} \|_M + \| \mathbf{T}_t \|_M \; t_M \right] \\ &\leq \frac{2}{c_0}\left[\| \mathbf{T} \|_M + \| \mathbf{T}_t \|_M \; t \right] \qquad (5.1.10)\end{aligned}$$

i.e., a linear growth in time. It is not surprising that the bound (5.1.10) is not as 'sharp' as (5.1.8), since the latter was derived directly from the exact solution (5.1.7).

As mentioned before, the multi-dimensional case

$$\frac{\partial^2 u}{\partial t^2} = \nabla^2 u + f(x_1, \ldots, x_d, t)$$

on complex shapes is completely analogous to the method indicated in chapter 1 and the proofs go over in the same manner.

Note also that instead of solving (5.1.4) directly as a 2^{nd} order ODE system in time one can solve

$$\begin{aligned} \frac{d\mathbf{w}}{dt} &= [D\mathbf{v} - \tau_L(A_L\mathbf{v} - \mathbf{g}_L) - \tau_R(A_R\mathbf{v} - \mathbf{g}_R)] + \mathbf{f} \\ \frac{d\mathbf{v}}{dt} &= \mathbf{w} \ . \end{aligned} \tag{5.1.11}$$

The number of 'variables' has increased from N to $2N$ but one gains in the simplicity of the time integration.

5.2 Construction of the scheme

This section is devoted to the task of constructing a symmetric negative definite M for the case of $m = 2$, i.e., a second order accurate finite difference algorithm.

Let

$$D = \frac{1}{h^2}\left\{ \begin{bmatrix} 1 & -2 & 1 & 0 & & & & & \\ 1 & -2 & 1 & 0 & & & & & \\ 0 & 1 & -2 & 1 & & & & & \\ 0 & 0 & 1 & -2 & 1 & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & 1 & -2 & 1 & 0 & 0 \\ & & & & & 1 & -2 & 1 & 0 \\ & & & & & & 1 & -2 & 1 \\ & & & & & & 1 & -2 & 1 \end{bmatrix} \right.$$

$$+ \begin{bmatrix} 0 & & & & & & \\ & c_2 & & & & & \\ & & c_3 & & & & \\ & & & \ddots & & & \\ & & & & c_{N-2} & & \\ & & & & & c_{N-1} & \\ & & & & & & 0 \end{bmatrix} \left. \begin{bmatrix} 0 & 0 & 0 & 0 & & & \\ 1 & -3 & 3 & -1 & & & \\ -1 & 4 & -6 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -6 & 4 & -1 \\ & & & -1 & 3 & -3 & 1 \\ & & & 0 & 0 & 0 & 0 \end{bmatrix} \right]$$

$$-\tilde{c}\left\{\begin{bmatrix} 0 & 0 & 0 & & & & \\ 0 & 1 & -2 & 1 & & & \\ -1 & 2 & 0 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -1 & 2 & 0 & -2 & 1 \\ & & & -1 & 2 & -1 & 0 \\ & & & 0 & 0 & 0 & 0 \end{bmatrix}\right\} \tag{5.2.1}$$

where

$$c_k = c_2 + \frac{c_{N-1} - c_2}{N-3}(k-2)\ , \tag{5.2.2}$$

and

$$\tilde{c} = \frac{c_{N-1} - c_2}{N-3}\ . \tag{5.2.3}$$

Note, that as in chapter 3, we had to resort to using *connectivity terms* in (5.2.1).

$$A_L = \begin{bmatrix} \frac{1}{2}(2+\gamma_L)(1+\gamma_L) & -\gamma_L(2+\gamma_L) & \frac{1}{2}(\gamma_L+\gamma_L^2) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{2}(2+\gamma_L)(1+\gamma_L) & -\gamma_L(2+\gamma_L) & \frac{1}{2}(\gamma_L+\gamma_L^2) & 0 & \dots & 0 \end{bmatrix}; \tag{5.2.4}$$

$$A_R = \begin{bmatrix} 0 & \dots & 0 & \frac{1}{2}(\gamma_R+\gamma_R^2) & -\gamma_R(2+\gamma_R) & \frac{1}{2}(2+\gamma_R)(1+\gamma_R) \\ \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & \frac{1}{2}(\gamma_R+\gamma_R^2) & -\gamma_R(2+\gamma_R) & \frac{1}{2}(2+\gamma_R)(1+\gamma_R) \end{bmatrix}. \tag{5.2.5}$$

$$\tau_L = \frac{1}{h^2}\mathrm{diag}\left[\tau_{L_1}, \tau_{L_2}, \tau_{L_3}, 0, \dots, 0, 0\right]; \tag{5.2.6}$$

$$\tau_R = \frac{1}{h^2}\mathrm{diag}\left[0, 0, \dots, 0, \tau_{R_{N-2}}, \tau_{R_{N-1}}, \tau_{R_N}\right]; \tag{5.2.7}$$

In order to make the matrix $M = D - \tau_L A_L - \tau_R A_R$ symmetric we choose:

$$
\begin{aligned}
c_2 &= \frac{(1-\gamma_L)\,\gamma_L}{2} \\
c_{N-1} &= \frac{(1-\gamma_R)\,\gamma_R}{2} \\
\tau_{L_2} &= \frac{3-\gamma_L-2\,\gamma_L\,\tau_{L_1}}{1+\gamma_L} \\
\tau_{L_3} &= \frac{-2+\gamma_L+\gamma_L\,\tau_{L_1}}{2+\gamma_L} \\
\tau_{R_{N-1}} &= \frac{3-\gamma_R-2\,\gamma_R\,\tau_{R_N}}{1+\gamma_R} \\
\tau_{R_{N-2}} &= \frac{-2+\gamma_R+\gamma_R\,\tau_{R_N}}{2+\gamma_R}
\end{aligned}
\tag{5.2.8}
$$

τ_{L_1} and τ_{R_N} will be determine later.

We now decompose M as follows:

$$
M = \frac{1}{h^2}\left[\alpha M_1 + (1-\alpha)M_2 + M_3 + M_4 + M_5\right] \tag{5.2.9}
$$

where:

$$
M_1 = \begin{bmatrix}
-2 & 1 & & & & & & \\
1 & -2 & 1 & & & & & \\
 & 1 & -2 & 1 & & & & \\
 & & \ddots & \ddots & \ddots & & & \\
 & & & 1 & -2 & 1 & & \\
 & & & & 1 & -2 & 1 & \\
 & & & & & 1 & -2 & \\
 & & & & & & 1 & -2
\end{bmatrix}, \tag{5.2.10}
$$

$$
M_2 = \begin{bmatrix}
0 & 0 & 0 & & & & & & \\
0 & 0 & 0 & & & & & & \\
0 & 0 & -1 & 1 & & & & & \\
 & & 1 & -2 & 1 & & & & \\
 & & & \ddots & \ddots & \ddots & & & \\
 & & & & 1 & -2 & 1 & & \\
 & & & & 0 & 1 & -1 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & 0
\end{bmatrix}, \tag{5.2.11}
$$

$$M_3 = -\left\{ \begin{bmatrix} 0 & 0 & 0 & & & & \\ 0 & -1 & 1 & & & & \\ 0 & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & 0 \\ & & & & 1 & -1 & 0 \\ & & & & 0 & 0 & 0 \end{bmatrix} \right.$$

$$\left. \begin{bmatrix} 0 & & & & & \\ & c_2 & & & & \\ & & c_3 & & & \\ & & & \ddots & & \\ & & & & c_{N-2} & \\ & & & & & c_{N-1} \\ & & & & & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & & & & \\ 0 & -1 & 1 & & & & \\ 0 & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & 0 \\ & & & & 1 & -1 & 0 \\ & & & & 0 & 0 & 0 \end{bmatrix} \right\}, \qquad (5.2.12)$$

$$M_4 = \begin{bmatrix} m_4^{1,1} & m_4^{1,2} & m_4^{1,3} & \\ m_4^{1,2} & m_4^{2,2} & m_4^{2,3} & 0 \\ m_4^{1,3} & m_4^{2,3} & m_4^{3,3} & \\ & 0 & & 0 \end{bmatrix} \qquad (5.2.13)$$

where:

$$\begin{aligned} m_4^{1,1} &= 1 + 2\alpha - \frac{(1+\gamma_L)(2+\gamma_L)\tau_{L_1}}{2} \\ m_4^{1,2} &= -2 - \alpha + \gamma_L(2+\gamma_L)\tau_{L_1} \\ m_4^{1,3} &= 1 - \frac{\gamma_L(1+\gamma_L)\tau_{L_1}}{2} \\ m_4^{2,2} &= 2\alpha + \frac{7\gamma_L - 4(1+\gamma_L^2) - {\gamma_L}^2(2+\gamma_L)(1+4\tau_{L_1})}{2(1+\gamma_L)} \\ m_4^{2,3} &= 1 - \alpha - \frac{3\gamma_L}{2} + \frac{{\gamma_L}^2}{2} + {\gamma_L}^2\tau_{L_1} \\ m_4^{3,3} &= 2\alpha + \frac{-4 + {\gamma_L}^2 - {\gamma_L}^3 - {\gamma_L}^2(1+\gamma_L)\tau_{L_1}}{2(2+\gamma_L)} \end{aligned}$$

and

$$M_5 = \begin{bmatrix} 0 & & 0 & \\ & m_5^{N-2,N-2} & m_5^{N-1,N-2} & m_5^{N,N-2} \\ 0 & m_5^{N-1,N-2} & m_5^{N-1,N-1} & m_5^{N,N-1} \\ & m_5^{N,N-2} & m_5^{N,N-1} & m_5^{N,N} \end{bmatrix} \tag{5.2.14}$$

where:

$$\begin{aligned} m_5^{N,N} &= 1 + 2\alpha - \frac{(1+\gamma_R)(2+\gamma_R)\tau_{R_N}}{2} \\ m_5^{N,N-1} &= -2 - \alpha + \gamma_R(2+\gamma_R)\tau_{R_N} \\ m_5^{N,N-2} &= 1 - \frac{\gamma_R(1+\gamma_R)\tau_{R_N}}{2} \\ m_5^{N-1,N-1} &= 2\alpha + \frac{7\gamma_R - 4(1+\gamma_R^2) - {\gamma_R}^2(2+\gamma_R)(1+4\tau_{R_N})}{2(1+\gamma_R)} \\ m_5^{N-1,N-2} &= 1 - \alpha - \frac{3\gamma_R}{2} + \frac{{\gamma_R}^2}{2} + {\gamma_R}^2\tau_{R_N} \\ m_5^{N-2,N-2} &= 2\alpha + \frac{-4 + {\gamma_R}^2 - {\gamma_R}^3 - {\gamma_R}^2(1+\gamma_R)\tau_{R_N}}{2(2+\gamma_R)}. \end{aligned}$$

The matrix M_1 is negative-definite and bounded away from 0 by $h^2\pi^2$ by the argument leading to eq. (2.4.31), see appendix to chapter 2. M_2 is non-positive definite, see eq. (2.4.34) and (2.4.35) in that appendix. From (5.2.2), (5.2.3) and (5.2.8) follows that $c_k \geq 0$, $k = 1, \ldots, N$, therefore, the matrix M_3 is non-positive. For a given value of $0 \leq \alpha \leq 1$, τ_{L_1} and τ_{R_N} can be found such that the matrices M_4 and M_5 will be non-positive, for all γ_L and γ_R. For example: for $\alpha = 1/10$, $\tau_{L_1} = \tau_{R_N} = 4$, for $\alpha = 1/2$, $\tau_{L_1} = \tau_{R_N} = 9$ and for $\alpha = 8/10$, $\tau_{L_1} = \tau_{R_N} = 24$. This completes the proof that M is indeed a negative-definite matrix, bounded away from 0 by $\alpha\pi^2$. Therefore the norm of the error vector $\| \boldsymbol{\epsilon} \|$ can grow at most linearly in time, see equations (5.1.8) and (5.1.10).

Summary

In this work a methodology for constructing finite-difference semi-discrete schemes, for initial boundary value problems (IBVP), on complex, multi-dimensional shapes was presented. Its starting point is the development of one-dimensional schemes on a uniform mesh with boundary points that do not necessarily coincide with the extremal nodes of the grid. The 1-D construction was done in such a way that the coefficient matrix of the corresponding ODE system (which represents the error evolution in time) is negative definite and bounded away from 0 by a constant independent of the size of the matrix, or is at least non-positive definite. This construction was carried out by imposing the boundary conditions using simultaneous approximation terms (SAT). Using the fact that the coefficient matrix is negative, or non-positive, definite it was proved that the one dimensional scheme is error-bounded and that this scheme can be used as a building block for multi-dimensional, error-bounded algorithms. The general theory was given in chapter 1.

In chapters 2 and 3 the methodology presented in chapter 1 was used to develop second and fourth order accurate approximations for $\partial^2/\partial x^2$ and a second order accurate approximation for $\partial/\partial x$. Using these approximations error-bounded schemes were constructed for the one and multi-dimensional diffusion and linear advection-diffusion equations. Numerical examples show that the method is effective even where standard schemes, stable by traditional definitions, fail.

In chapter 4 the methodology presented in chapter 1 was adopted to construct error-bounded schemes for parabolic equations and systems containing mixed-derivatives. In chapter 5 the method was modified to solve the wave equation.

The efficacy of the method was demonstrated via the various numerical examples. The operators developed herein can be used as 'off the shelf operators' for other differential equations of mathematical physics. However the severe restrictions on the approximations, i.e. the negative or non-positive definiteness in standard L_2 norm, make the construction far from being trivial. The development of fourth order accurate approximations for $\partial/\partial x$ as well as schemes for hyperbolic systems is left for future research. The imposition of other boundary conditions such as Neumann or absorbing boundary conditions is also left for future study.

Bibliography

[1] S. Abarbanel, S. Bennet, A. Brandt, J. Gillis, Velocity Profiles of flow at Low Reynolds Numbers. *Journal of Applied Mechanics* 37E, **1**, 1970. 1-3.

[2] S. Abarbanel, A. Ditkowski, Multi-Dimensional Asymptotically Stable 4^{th}-Order Accurate Schemes for the Diffusion Equation. ICASE Report No.96-8, February 1996. Also, Asymptotically Stable Fourth-Order Accurate Schemes for the Diffusion Equation on Complex Shapes. *J. Comput. Phys.*, **133**(2), 1997.

[3] S. Abarbanel, A. Ditkowski, Multi-dimensional asymptotically stable schemes for advection-diffusion equations.ICASE report 47-96. To appear *Computers and Fluids.*

[4] S. Abarbanel, D.Gottlieb, Stability of Tow-Dimensional Initial Boundary Value Problems Using Leap-Frog Type Systems. *Math. of Comp.*, **33**(148), 1979. 1145-1155.

[5] S. Abarbanel, D.Gottlieb, Spurious Frequencies as a Result of Numerical Boundary Treatments. ICASE Report 90-73, October 1990.

[6] M. Berger, R. LeVeque,Stable Boundary Conditions for Cartesian Grid Calculations, ICASE Report No. 90-37, 1990.

[7] A. C. Cangellaris, D. B. Wright. Analysis of the numerical error caused by the stair-stepped approximation of a conducting boundary in FDTD simulations of electromagnetic phenomena. *IEEE Trans. Antennas Propagat.*, **39**(10), 1518-1525, 1991.

[8] M.H. Carpenter, D.Gottlieb and S. Abarbanel, Time Stable Boundary Conditions for Finite Difference Schemes Solving Hyperbolic Systems: Methodology and Application to High Order Compact Schemes. NASA Contractor Report 191436, ICASE Report 93-9, To appear *J. Comput. Phys.*.

[9] D. Clarke, M. Salas and H. Hassan, Euler Calculations for Multi-Element Airfoils using Cartesian Grids, *AIAA Journal*, **24**(2), 1986.

[10] A. Friedman. Partial Differential Equations of Parabolic Type. Prentice-Hall, 1964.

[11] R. Gaffney, H. Hassan, and M. Salas, Euler Calculations for Wings using Cartesian Grids, *AIAA paper*, 87-0356, 1987.

[12] B. Gustafsson, H.O. Kreiss, and A. Sundström, Stability Theory of Difference Approximations for Mixed Initial Boundary Value Problems. II, *Math. Comp.* **26**, 1972. 649-686.

[13] B. Gustafsson, H.O. Kreiss, and J. Oliger, Time Dependent Problems and Difference Methods. John Wiley & Sons, Inc., 1995.

[14] S.K. Goganov and V.S. Ryabenkii, Spectral Criteria for the Stability of Boundary-Value Problems for Non-Self-Adjoint Difference Equations, *Uspeki Mat.* **18 VIII**, 3-15, 1963.

[15] R. Holland. Pitfalls of staircase meshing. *IEEE Trans. Electromagn. Compat.*, **35**(4), 434–439, 1993.

[16] F. John. Partial Differential Equations. Fourth edition, Springer-Verlag, 1982.

[17] E. A.Jones, W. T. Joines. Improved Computational Efficiency by Using Sub-Regions in FDTD Simulations. Conference proceedings of the 13$^{\text{th}}$ Annual Review of Progress in Applied Computational Electromagnetics, 322-329, 1997.

[18] H.O. Kreiss. Difference Approximations for the Initial Boundary Value Problem for Hyperbolic Differential Equations. *Numerical Solutions of Nonlinear Partial Differential Equations*, edited by D Greenspan, Wiley, New York, 1966.

[19] H.O. Kreiss. Stability Theory for Difference approximations of Mixed Initial Boundary Value Problem. **I.** *Math. Comp.*, **22**, 703-714, 1968.

[20] H.O. Kreiss, G. Scherer. Finite Element and Finite Difference Methods for Hyperbolic Partial Differential Equations.*Mathematical Aspects of Finite Element in Partial Differential Equations*, Academic Press, Inc., 1974.

[21] H.O. Kreiss, G. Scherer. On the Existence of Energy estimates for Difference Approximations for Hyperbolic Systems. Technical report, Dept. of Scientific Computing, Uppsala University, 1977

[22] H.O. Kreiss, L. Wu. On the Stability definition of Difference Approximations for the Initial Boundary Value Problems. *Appl. Num. Math.*, **12**, 1993, 213-227.

[23] D. Levy, E. Tadmor. From Semi-Discrete to Fully-Discrete: The Stability of Runge-Kutta Schemes by the Energy Method. *SIAM Review*, to appear.

[24] J. E. Melton, M. J. Berger, M. J. Aftosmis and M. D. Wong, 3D Applications of a Cartesian Grid Euler Method, *AIAA Paper*, 95-0853, 1995.

[25] J. Melton, F. Enomoto and M. Berger, 3D Automatic Cartesian Grid Generation for Euler Flows. *AIAA Paper*, 93-3386-CP, 1993.

[26] P. Monk and E. Süli. A convergence analysis of Yee's scheme on non-uniform grids. *SIAM J. on Numer. Anal.*, **31**(2), 393–412, 1994.

[27] K. Morinishi, A Finite Difference Solution of the Euler Equations on Non-body-fitted Cartesian Grids,*Computers Fluids*, **21**(3), 331-344, 1992.

[28] P. Olsson, Summation by Parts, Projections and Stability. I. *Math. of Comp.*, **64**(211), 1995. 1035-1065.

[29] P. Olsson, Summation by Parts, Projections and Stability. II. *Math. of Comp.*, **64**(212), 1995. 1473-1493.

[30] S. Osher. Systems of Difference Equations with General Homogeneous Boundary Conditions. *Tran. Amer. Math. Soc.*, **v.137**, 1969, pp. 177-201.

[31] J. Purvis, J. Burkhalter. Prediction of Critical Mach Number for Store Configurations. *AIAA Journal*, **17**(2), 1979.

[32] T. A. Reyhner. Cartesian Mesh Solution for Axisymmetric Transonic Potential Flow Around Inlets,*AIAA Journal*, **15**(5), 1977, pp. 624-631.

[33] R.D. Richtmyer, K.W. Morton, Difference Methods for Initial-Value Problems. Second edition. John Wiley & Sons, Inc., 1967.

[34] K.L. Shlager, J.B. Schneider. A Selective Survey of the Finite-Difference Time-Domain Literature. *IEEE Antennas and Propagation Magazine*, **37**(4), 39-56, 1995.

[35] B. Strand, Summation by Parts for Finite Difference Approximations for d/dx. *J. Comput. Phys.*, **110**, 1994. 47-67.

[36] B. Strand, High Order Difference Approximations for Hyperbolic Initial boundary Value Problems. Thesis, Dept of Scientific Computing, Uppsala University, Uppsala, Sweden, 1996

[37] J.C. Strikwerda. Initial Boundary Value Problems for Method of Lines. *J. Comput. Phys.*, **34**, 1980. 94-110.

[38] K. S. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas Propagat.* , **AP-14**(4), 302–307, 1966.