Multi-dimensional asymptotically stable finite difference schemes for the advection-diffusion equation.

Saul Abarbanel*
Adi Ditkowski*
School of Mathematical Sciences
Department of Applied Mathematics
Tel-Aviv University
Tel-Aviv, ISRAEL

Abstract

An algorithm is presented which solves the multi-dimensional advection-diffusion equation on complex shapes to 2^{nd}-order accuracy and is asymptotically stable in time. This bounded-error result is achieved by constructing, on a rectangular grid, a differentiation matrix whose symmetric part is negative definite. The differentiation matrix accounts for the Dirichlet boundary condition by imposing penalty like terms.

Numerical examples in 2-D show that the method is effective even where standard schemes, stable by traditional definitions, fail. It gives accurate, non oscillatory results even when boundary layers are not resolved.

*This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001. S. Abarbanel was also supported in part by the Air Force Office of Scientific research Grant No. AFOSR-F49620-95-1-0074, and by the Department of Energy under grant DOE-DE-FG02-95ER25239.

1 Introduction

Currently there is a growing interest in long time integration for solving problems in areas such as fluid-mechanics, aero-acoustics, electro-magnetics, material-science, and others. Clearly, it will be very advantageous if one could formulate the spatial discretization in a way which guarantees that, for the semi-discrete formulation, the solution-error norm is bounded by the norm of the truncation error. Most, if not all, existing algorithms rely on stability for convergence. However, even stable schemes, which at a given time converge with mesh refinement may have a temporally growing error, [1]. This is particularly true for hyperbolic operators.

This paper considers 2$^{\mathrm{nd}}$-order accurate approximations to model linear advection-diffusion equations in one and more dimensions, on domains which may be irregular. By an irregular domain, we mean a body whose boundary points do not necessarily coincide with nodes of a rectangular mesh.

In section 2 we treat a model "shock-layer" equation (linearized Burger's equation),

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

We develop there the theory for the one dimensional semi-discrete system resulting from the spatial differentiation used in the finite difference algorithm. Energy methods are used in conjunction with "SAT" type terms (see [1], [2]), in order to find boundary treatment and "artificial-viscosity-like terms", that preserve the accuracy of the scheme while constraining an energy norm of the error to be temporally bounded for all $t > 0$ by a "constant" proportional to the norm of the truncation error.

In section 3 it is shown how the methodology developed in section 2 is used as a building block for the multi-dimensional algorithm, even for irregular shapes.

Section 4 presents numerical results. Section 4.1 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.

Section (4.2) considers steady-state solution to a two dimensional scalar model to the boundary layer equations,

$$u_t + au_x + bu_y = \frac{1}{R}u_{yy}; \qquad R \gg 1, \quad b < 0,$$

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

Section (4.3) presents a time dependent example, modeling a boundary-layer being excited sinosoidially,

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

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.

2 The Scalar One Dimensional Case

Consider the scalar advection-diffusion problem

$$\frac{\partial u}{\partial t} = a\frac{\partial u}{\partial x} + \frac{1}{R}\frac{\partial^2 u}{\partial x^2} + f(x,t); \ \ \Gamma_L \leq x \leq \Gamma_R, \ \ t \geq 0, \ \ a > 0, \ \ ^{*} \tag{2.1a}$$

$$u(x,0) = u_0(x), \tag{2.1b}$$

$$u(\Gamma_L, t) = g_L(t), \tag{2.1c}$$

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

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

Let us discritize (2.1) spatially on the following uniform grid:

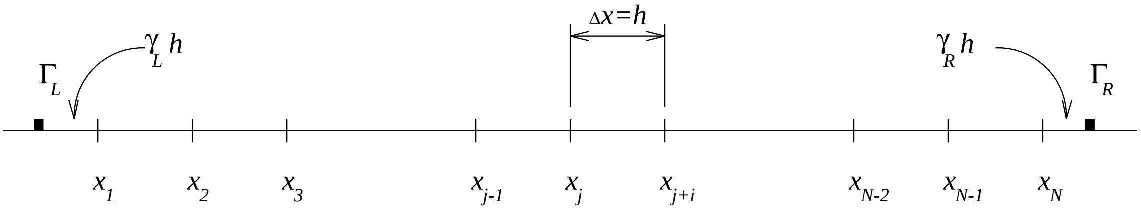

Figure 1: One dimensional grid.

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

[*] The results for the case $a < 0$ are found by an analysis anologus to the one presented in this section, and are presented in Appendix I.

The projection unto the above grid of the exact solution $u(x,t)$ to (2.1), is $u_j(t) = u(x_j,t) \triangleq \mathbf{u}(t)$. Let $\tilde{D}$ be a matrix representing $au_x + \frac{1}{R}u_{xx}$, at internal points without specifying yet how it is being constructed. Then we may write

$$\frac{d}{dt}\mathbf{u}(t) = [\tilde{D}\mathbf{u}(t) + \mathbf{B} + \mathbf{T}] + \mathbf{f}(t), \tag{2.2}$$

where $\mathbf{T}$ is the truncation error due to the numerical differentiation, and $\mathbf{f}(t) = f(x_j,t)$, $1 \leq j \leq N$. The boundary vector $\mathbf{B}$ has entries whose values depend on g_L, g_R, γ_L, γ_R in such a way that $\tilde{D}\mathbf{u} + \mathbf{B}$ represents $au_x + \frac{1}{R}u_{xx}$ everywhere to the desired accuracy. The standard way of finding a numerical approximate solution to (2.1) is to omit $\mathbf{T}$ from (2.2) and solve

$$\frac{d}{dt}\mathbf{v}(t) = \tilde{D}\mathbf{v}(t) + \mathbf{B} + \mathbf{f}(t), \tag{2.3}$$

where $\mathbf{v}(t)$ is the numerical approximation to the projection $\mathbf{u}(t)$. Subtracting (2.3) from (2.2) one gets an equation for the solution error, $\vec{\epsilon}(t) = \mathbf{u}(t) - \mathbf{v}(t)$,

$$\frac{d}{dt}\vec{\epsilon} = \tilde{D}\vec{\epsilon} + \mathbf{T}. \tag{2.4}$$

Our requirement for temporal stability is that $\| \vec{\epsilon} \|$, the L_2 norm of $\vec{\epsilon}$, be bounded by a "constant" proportional to h^m (m being the spatial order of accuracy). Note that this definition is more severe than either the G.K.S. stability criterion [3], or the definition in [1].

It can be shown that if $\tilde{D}$ is constructed 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 rest of this section is devoted to the construction of a scheme of 2^{nd} order spatial accuracy, which is temporally stable for γ_L, γ_R. The basic idea is to follow the procedure used in [2]. The present case is more complicated due to the difficulty in treating the advection term.

Note first that the solution projection $u_j(t)$ satisfies, besides (2.2), the following differential equation:

$$\frac{d\mathbf{u}}{dt} = D\mathbf{u} + \mathbf{T}_e + \mathbf{f}(t), \tag{2.5}$$

where now D is indeed a differentiation matrix, 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 (2.3),

$$\frac{d\mathbf{v}}{dt} = [D\mathbf{v} - \tau_L(A_L\mathbf{v} - \mathbf{g}_L) - \tau_R(A_R\mathbf{v} - \mathbf{g}_R)] + \mathbf{f}(t), \tag{2.6}$$

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:

$$A_L\mathbf{u} = \mathbf{g}_L - \mathbf{T}_L, \qquad A_R\mathbf{u} = \mathbf{g}_R - \mathbf{T}_R, \tag{2.7}$$

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

$$\tau_L = \text{ diag } (\tau_{L_1}, \tau_{L_2}, \ldots, \tau_{L_N}); \qquad \tau_R = \text{ diag } (\tau_{R_1}, \tau_{R_2}, \ldots, \tau_{R_N}). \tag{2.8}$$

Subtracting (2.6) from (2.5) we get

$$\frac{d\vec{\epsilon}}{dt} = [D\vec{\epsilon} - \tau_L A_L \vec{\epsilon} - \tau_R A_R \vec{\epsilon} + \mathbf{T}_1], \tag{2.9}$$

where

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

Taking the scalar product of $\vec{\epsilon}$ with (2.9) one gets:

$$\begin{aligned}\frac{1}{2}\frac{d}{dt} \parallel \vec{\epsilon} \parallel^2 &= (\vec{\epsilon}, (D - \tau_L A_L - \tau_R A_R)\vec{\epsilon}) + (\vec{\epsilon}, \mathbf{T}_1) \\ &= (\vec{\epsilon}, M\vec{\epsilon}) + (\vec{\epsilon}, \mathbf{T}_1).\end{aligned} \tag{2.10}$$

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

$$M = D - \tau_L A_L - \tau_R A_R. \tag{2.11}$$

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

$$(\vec{\epsilon}, (M + M^T)\vec{\epsilon})/2 \leq -c_0 \parallel \vec{\epsilon} \parallel^2, \qquad (c_0 > 0). \tag{2.12}$$

Equation (2.10) then becomes

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

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

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

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

$$\| \vec{\epsilon} \| \leq \frac{\| \mathbf{T}_1 \|_M}{c_0}(1 - e^{-c_0 t}) \tag{2.13}$$

where the "*constant*" $\| \mathbf{T_1} \|_M = \max\limits_{0 \leq \tau \leq t} \| \mathbf{T_1}(\tau) \|$.

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 (2.13) that the norm of the error will be bounded for all t by a constant which is $O(h^m)$ where m is the spatial accuracy of the finite difference scheme (2.6). The numerical solution is then temporally stable.

It can be shown that as $\frac{1}{R} \to 0$, so does c_0. When $c_0 = 0$, the differential inequality is

$$\frac{d}{dt} \| \vec{\epsilon} \| \leq \| \mathbf{T}_1 \| \tag{2.14}$$

leading to

$$\| \vec{\epsilon} \| \leq \| \mathbf{T}_1 \|_M \, t, \tag{2.15}$$

i.e., a linear growth in time, a result typical of hyperbolic systems. This result can also be obtained formally from (2.13) by letting $c_0 \to 0$ for *any* fixed t.

The rest of 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. We shall deal separately with the hyperbolic and parabolic parts of the R.H.S. of (2.11)

Let

$$M = \frac{1}{R}M_P + aM_H = \frac{1}{R}(D_P - \tau_{L_P}A_{L_P} - \tau_{R_P}A_{R_P}) + a(D_H - \tau_{L_H}A_{L_H} - \tau_{R_H}A_{R_H}). \tag{2.16}$$

The parabolic terms are given by:

$$D_P = \frac{1}{h^2}\left[\begin{array}{ccccccccc} 1 & -2 & 1 & 0 & & & & & \\ 1 & -2 & 1 & 0 & & & & & \\ 0 & 1 & -2 & 1 & & & & & \\ 0 & 0 & 1 & -2 & 1 & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & 1 & -2 & 1 & 0 & 0 \\ & & & & & 1 & -2 & 1 & 0 \\ & & & & & & 1 & -2 & 1 \\ & & & & & & 1 & -2 & 1 \end{array}\right]; \tag{2.17}$$

$$\tau_{L_P} = \frac{1}{h^2}\text{diag}\left[\tau_{L_1}^{(P)}, 0, \dots 0\right] = \frac{1}{h^2}\text{diag}\left[\frac{4}{(2+\gamma_L)(1+\gamma_L)}, 0, \dots, 0\right]; \tag{2.18}$$

$$\tau_{R_P} = \frac{1}{h^2}\text{diag}\left[0, 0, \dots, \tau_{R_N}^{(P)}\right] = \frac{1}{h^2}\text{diag}\left[0, 0, \dots, \frac{4}{(2+\gamma_R)(1+\gamma_R)}\right]; \tag{2.19}$$

$$A_{L_P} = \left[\begin{array}{cccccc} \frac{1}{2}(2+\gamma_L)(1+\gamma_L) & -\gamma_L(2+\gamma_L) & \frac{1}{2}(\gamma_L+\gamma_L^2) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{2}(2+\gamma_L)(1+\gamma_L) & -\gamma_L(2+\gamma_L) & \frac{1}{2}(\gamma_L+\gamma_L^2) & 0 & \dots & 0 \end{array}\right]; \tag{2.20}$$

$$A_{R_P} = \left[\begin{array}{cccccc} 0 & \dots & 0 & \frac{1}{2}(\gamma_R+\gamma_R^2) & -\gamma_R(2+\gamma_R) & \frac{1}{2}(2+\gamma_R)(1+\gamma_R) \\ \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & \frac{1}{2}(\gamma_R+\gamma_R^2) & -\gamma_R(2+\gamma_R) & \frac{1}{2}(2+\gamma_R)(1+\gamma_R) \end{array}\right]. \tag{2.21}$$

The hyperbolic terms are given by:

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

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

$$\left. + 2h\tilde{c}\begin{bmatrix} 0 & -1 & 1 & & & & & \\ 1 & -1 & -1 & 1 & & & & \\ 1 & -1 & 0 & -1 & 1 & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & 1 & -1 & 0 & -1 & 1 & \\ & & & & 1 & -1 & -1 & 1 \\ & & & & & 1 & -1 & 0 \end{bmatrix}\right\}, \tag{2.22}$$

where

$$c_k = \frac{1}{N-1}[(c_N - c_1)k + (Nc_1 - c_N)], \tag{2.23}$$

and

$$\tilde{c} = \frac{1}{2}(c_1 - c_N). \tag{2.24}$$

For $a > 0$ in (2.1a), 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 $a > 0$,

$$\tau_{R_H} = \frac{1}{2h}\text{diag}\,[0, 0, \ldots, \tau^{(H)}_{R_{N-1}}, \tau^{(H)}_{R_N}] \tag{2.25}$$

and

$$A_{R_H} = \left[\begin{array}{ccccc} 0 & & & & \\ & \ddots & & & \mathbf{0} \\ & & 0 & & \\ \mathbf{0} & & & -\gamma_R & 1+\gamma_R \\ & & & -\gamma_R & 1+\gamma_R \end{array}\right]. \tag{2.26}$$

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 (2.17) to (2.21), to be

$$\tilde{M}_P = \frac{1}{2h^2}\left[\begin{array}{ccccccccc} -2 & \frac{3\gamma_L - 1}{\gamma_L + 1} & \frac{2-\gamma_L}{2+\gamma_L} & & & & & & \\ \frac{3\gamma_L - 1}{\gamma_L + 1} & -4 & 2 & & & \mathbf{0} & & & \\ \frac{2-\gamma_L}{2+\gamma_L} & 2 & -4 & 2 & & & & & \\ & & 2 & -4 & 2 & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & 2 & -4 & 2 & & \\ \mathbf{0} & & & & & 2 & -4 & 2 & \frac{2-\gamma_R}{2+\gamma_R} \\ & & & & & & 2 & -4 & \frac{3\gamma_R - 1}{\gamma_R + 1} \\ & & & & & & \frac{2-\gamma_R}{2+\gamma_R} & \frac{3\gamma_R - 1}{\gamma_R + 1} & -2 \end{array}\right]. \tag{2.27}$$

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

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

$$\left. + \begin{bmatrix} -2(1-2\alpha) & \dfrac{3\gamma_L - 1}{\gamma_L + 1} - 2\alpha & \dfrac{2-\gamma_L}{2+\gamma_L} & & & & & \\ \dfrac{3\gamma_L - 1}{\gamma_L + 1} - 2\alpha & -4(1-\alpha) & 2(1-\alpha) & & & & & \\ \dfrac{1-\gamma_L}{2+\gamma_L} & 2(1-\alpha) & -2(1-\alpha) & & & & & \Large 0 \\ & & & 0 & & & & \\ & & & & 0 & & & \\ \Large 0 & & & & & -2(1-\alpha) & 2(1-\alpha) & \dfrac{2-\gamma_R}{2+\gamma_R} \\ & & & & & 2(1-\alpha) & -4(1-\alpha) & \dfrac{3\gamma_R - 1}{\gamma_R + 1} - 2\alpha \\ & & & & & \dfrac{2-\gamma_R}{2+\gamma_R} & \dfrac{3\gamma_R - 1}{\gamma_R + 1} - 2\alpha & -2(1-2\alpha) \end{bmatrix} \right\}. \tag{2.28}$$

We look for $1 > \alpha > 0$ such that the second and third matrices in (2.28) are non-positive definite. The first matrix in (2.28) is already negative definite by the argument leading to eq. (2.60), in [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 (2.28) is non-positive definite, see eq. (2.63) & (2.64) in [2]. The third matrix in (2.28) 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.

Next we would like to show that $\tilde{M}_H = \frac{1}{2}(M_H + M_H^T)$ is non-positive definite. Using equations (2.22)–(2.26) we have

$$\tilde{M}_H = \frac{1}{4h} \left[\begin{array}{cccccc} -4-2c_1 & 1+2c_1 & 0 & & & \\ 1+2c_1 & -2c_1 & 0 & & \mathbf{0} & \\ 0 & 0 & 0 & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & \mathbf{0} & & & 2c_N + 2\gamma_R\tau_{N-1}^{(H)} & -1-2c_N-(1+\gamma_R)\tau_{N-1}^{(H)}+\gamma_R\tau_N^{(H)} \\ & & & & -1-2c_N-(1+\gamma_R)\tau_{N-1}^{(H)}+\gamma_R\tau_N^{(H)} & 4+2c_N-2(1+\gamma_R)\tau_N^{(H)} \end{array} \right]. \tag{2.29}$$

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

$$\tilde{M}_H = \frac{1}{4h}[m_{H_1} + m_{H_2} + m_{H_3}], \tag{2.30}$$

where

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

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

$$m_{H_3} = c_N \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 2 & -2 \\ & & & -2 & 2 \end{bmatrix}. \qquad (2.31)$$

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

$$\tau_N^{(H)} = \frac{2+\delta}{1+\gamma_R}, \qquad (\delta \geq 0) \qquad (2.32)$$

$$\tau_{N-1}^{(H)} = -\frac{1-\gamma_R(1-\delta)}{(1+\gamma_R)^2}. \qquad (2.33)$$

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 $-\alpha\pi^2/R$, $0 < \alpha < .275$.

3 The Scalar Two Dimensional Case

We consider an inhomogeneous advection-diffusion equation, with constant coefficients, in a domain Ω. To begin with we shall assume that Ω is convex and has a boundary $\partial\Omega \in \mathcal{C}^2$. The convexity restriction is for the sake of simplicity in presenting the basic idea; it will be removed later. The problem statement is:

$$\frac{\partial u}{\partial t} = a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} + \nu_1\frac{\partial^2 u}{\partial x^2} + \nu_2\frac{\partial^2 u}{\partial y^2} + f(x,y;t); \qquad t > 0, \quad \nu_1, \nu_2 > 0, \tag{3.1a}$$

$$u(x,y,0) = u_0(x,y), \tag{3.1b}$$

$$u(x,y,t)|_{\partial\Omega} = u_B(t). \tag{3.1c}$$

We shall refer to the following grid representation:

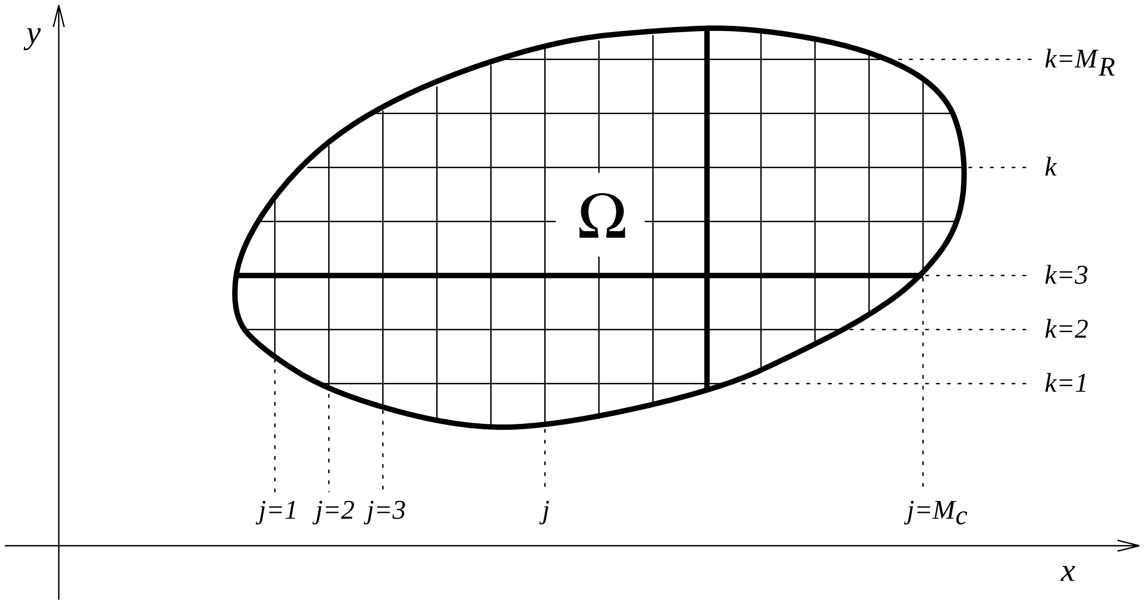

Figure 2: Two dimensional grid.

We have M_R rows and M_C columns inside Ω. Each row and each column has a discretized structure as in the 1-D case, see figure 1. Let the number of grid points in the k^{th} row be denoted by R_k and similarly let the number of 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,

$$\begin{aligned}
\mathbf{U}(t) &= (u_{1,1}, u_{2,1}, \ldots, u_{R_1,1}; u_{1,2}, u_{2,2}, \ldots, u_{R_2,2}; \ldots; u_{1,M_R}, u_{2,M_R}, \ldots, u_{R_{M_R},M_R}) \\
&\equiv (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M_R}). \qquad (3.2)
\end{aligned}$$

Thus, we have arranged the solution projection in vectors according to rows, starting from the bottom of Ω.

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

$$\begin{aligned}
\mathbf{U}^{(C)}(t) &= (u_{1,1}u_{1,2}, \ldots, u_{1,C_1}; u_{2,1}, u_{2,2} \ldots, u_{2,C_2}; \ldots; U_{M_C,1}, u_{M_C,2}, \ldots, u_{M_C,C_{M_C}}) \\
&\equiv (\mathbf{u}_1^{(C)}, \mathbf{u}_2^{(C)}, \ldots, \mathbf{u}_{M_C}^{(C)}). \qquad (3.3)
\end{aligned}$$

Clearly

$$\mathbf{U}^{(C)}(t) = P\mathbf{U}, \qquad (3.4)$$

where P is an orthogonal permutation matrix, of order $\ell \times \ell$, ℓ being the number of grid points within Ω.

The operator $\nu_1 \partial^2/\partial x^2 + a\ \partial/\partial x$ in (3.1a), including the boundary terms, is represented on the k^{th} row by $M_k^{(x)}$, whose structure is given by (2.16) and the definition following it (see (2.17) through (2.26)). Similarly let $M_j^{(y)}$ represent $\nu_2 \partial^2/\partial y^2 + b\ \partial/\partial y$ on the j^{th} column.

With this notation, by analogy to (2.6), the two dimensional semi-discrete problem becomes

$$\frac{d\mathbf{V}}{dt} = (\mathcal{M}^{(x)} + P^T\mathcal{M}^{(y)}P)\mathbf{V} + \mathbf{G}^{(x)} + P^T\mathbf{G}^{(y)} + \mathbf{f}(t), \tag{3.5}$$

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

$$\mathcal{M}^{(x)} = \begin{bmatrix} M_1^{(x)} & & & & \\ & \ddots & & & \\ & & M_k^{(x)} & & \\ & & & \ddots & \\ & & & & M_{M_R}^{(x)} \end{bmatrix}; \;\; \mathcal{M}^{(y)} = \begin{bmatrix} M_1^{(y)} & & & & \\ & \ddots & & & \\ & & M_j^{(y)} & & \\ & & & \ddots & \\ & & & & M_{M_C}^{(y)} \end{bmatrix}, \tag{3.6}$$

and

$$\begin{aligned} \mathbf{G}^{(x)} = G_P^{(x)} + G_H^{(x)} \;\; &= \;\; [(\tau_{L_1}^{(P)}\mathbf{g}_{L_1} + \tau_{R_1}^{(P)}\mathbf{g}_{R_1}), \ldots, (\tau_{L_k}^{(P)}\mathbf{g}_{L_k} + \tau_{R_k}^{(P)}\mathbf{g}_{R_k}), \ldots, \\ & \quad\;\; (\tau_{L_{M_R}}^{(P)}\mathbf{g}_{L_{M_R}} + \tau_{R_{M_R}}^{(P)}\mathbf{g}_{R_{M_R}})] \\ &+ \;\; [(\tau_{L_1}^{(H)}\mathbf{g}_{L_1} + \tau_{R_1}^{(H)}\mathbf{g}_{R_1}), \ldots, (\tau_{L_k}^{(H)}\mathbf{g}_{L_k} + \tau_{R_k}^{(H)}\mathbf{g}_{R_k}), \ldots, \\ & \quad\;\; (\tau_{L_{M_R}}^{(H)}\mathbf{g}_{L_{M_R}} + \tau_{R_{M_R}}^{(H)}\mathbf{g}_{R_{M_R}})], \end{aligned}$$

$$\begin{aligned} \mathbf{G}^{(y)} = G_P^{(y)} + G_H^{(y)} \;\; &= \;\; [(\tau_{B_1}^{(P)}\mathbf{g}_{B_1} + \tau_{T_1}^{(P)}\mathbf{g}_{T_1}), \ldots, (\tau_{B_j}^{(P)}\mathbf{g}_{B_j} + \tau_{T_j}^{(P)}\mathbf{g}_{T_j}), \ldots, \\ & \quad\;\; (\tau_{B_{M_C}}^{(P)}\mathbf{g}_{B_{M_C}} + \tau_{T_{M_C}}^{(P)}\mathbf{g}_{T_{M_C}})] \\ &+ \;\; [(\tau_{B_1}^{(H)}\mathbf{g}_{B_1}\tau_{T_1}^{(H)}\mathbf{g}_{T_1}), \ldots, (\tau_{B_j}^{(H)}\mathbf{g}_{B_j} + \tau_{T_j}^{(H)}\mathbf{g}_{T_j}), \ldots, \\ & \quad\;\; (\tau_{B_{M_C}}^{(H)}\mathbf{g}_{B_{M_C}} + \tau_{T_{M_C}}^{(H)}\mathbf{g}_{T_{M_C}})]. \end{aligned} \tag{3.7}$$

The subscripts B_j ("B" for bottom) play the same role as L_k ("L" for left). The same remark applied to subscripts T_j ("T" for top) and R_k ("R" for right).

Note that $\tau^H_{L_k}(\tau^H_{R_k}) = 0$ when $a > 0(a < 0)$. Similarly $\tau^H_{B_j}(\tau^H_{T_j}) = 0$ when $b > 0(b < 0)$.

Designating the two dimensional array off errors, ϵ_{ij}, by $\mathbf{E} = \mathbf{U} - \mathbf{V}$, the equation for $\mathbf{E}$ becomes

$$\frac{d\mathbf{E}}{dt} = [\mathcal{M}^{(x)} + P^T\mathcal{M}^{(y)}P]\mathbf{E} + \mathbf{T}, \tag{3.8}$$

where $\mathbf{T}$ represents the sum of the various truncation errors.

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

$$\frac{1}{2}\frac{d}{dt} \| \mathbf{E} \|^2 = (\mathbf{E}, (\mathcal{M}^{(x)} + P^T\mathcal{M}^{(y)}P)\mathbf{E}) + (\mathbf{E}, \mathbf{T}). \tag{3.9}$$

By the same argument that follow eq. (3.15) in [2] it is clear that the norm of the error, $\| \mathbf{E} \|$, is bounded by a constant, where the "constant" $\| \mathbf{T} \|_M = \max\limits_{0 \leq \tau \leq t} \| \mathbf{T}(\tau) \|$.

In [2], it was shown that if the domain Ω is not convex or simply connected, the above results still hold. This is also true here.

Note that if $\frac{1}{R} = \nu = 0$ (or $\nu_1 = \nu_2 = 0$ in the 2-D case) then the differentiation operator, M, becomes non-positive definite. In that case, it follows immediately from (3.9) that the bound on the error-norm is not a "constant" but grows *linearly* in time.

4 Numerical Examples

4.1 One Dimensional Case

Here we consider the problem

$$\frac{\partial u}{\partial t} + u_x = \frac{1}{R} u_{xx}, \qquad t \geq 0, \;\; 0 \leq x \leq 1, \tag{4.1.1}$$

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

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

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

The steady state solution to (4.1.1) is:

$$u(x) = \frac{1 - e^{-R(1-x)}}{1 - e^{-R}}. \tag{4.1.2}$$

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

Eq. (4.1.1) 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 approximation $\mathbf{v}$, in this “standard” case, satisfies the following finite difference equation:

$$\frac{1}{2h}(v_{j+1} - v_{j-1}) - \frac{1}{Rh^2}(v_{j+1} - 2v_j + v_{j-1}) = 0, \qquad (0 \leq j \leq n) \tag{4.1.3}$$

with $v_0 = 1$ and $v_N = 0$. The solution to (4.1.3) is:

$$v_j = \frac{\kappa^j - \kappa^{2N-j}}{1-\kappa^{2N}}, \qquad \kappa = \frac{2+hR}{2-hR}. \tag{4.1.4}$$

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 (4.1.1) using the "standard scheme", and using the "bounded-error" algorithm, (2,6), described above are shown in figures 3–8 for $\Delta x = 1/100$ and various values of R. Both schemes were advanced to steady state using 4th-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 $R_C = 10$.

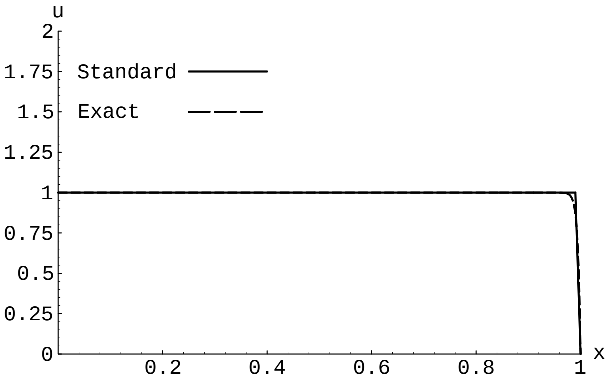

Figure 3: Standard scheme, $R_C = 2$.

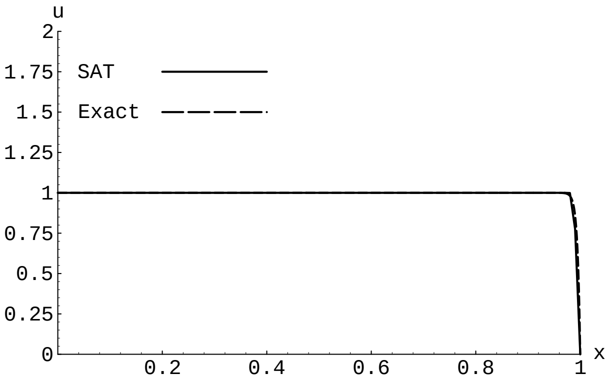

Figure 4: SAT, $R_C = 2$.

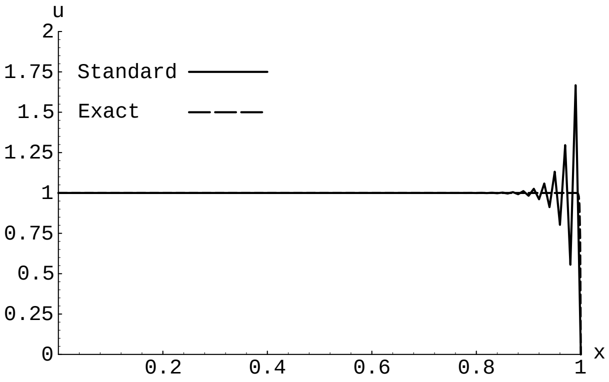

Figure 5: Standard scheme, $R_C = 10$.

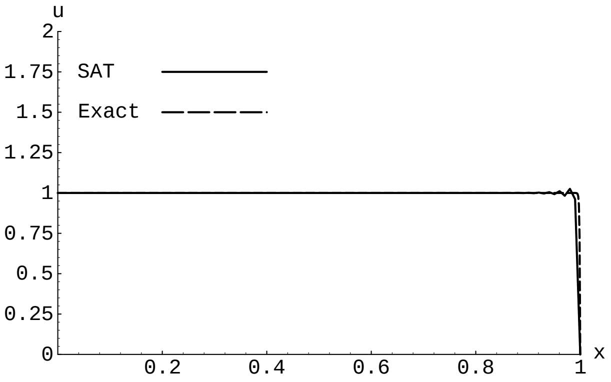

Figure 6: SAT, $R_C = 10$.

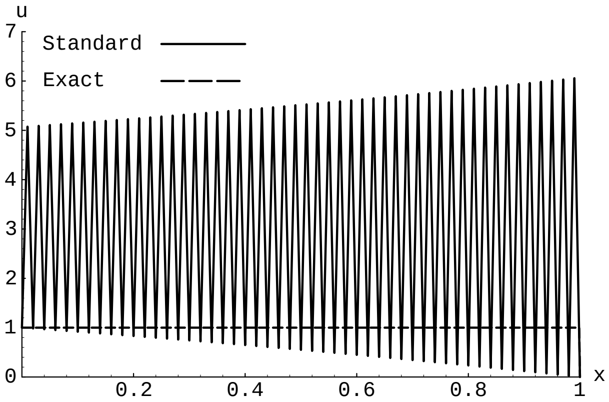

Figure 7: Standard scheme, $R_C = 1000$.

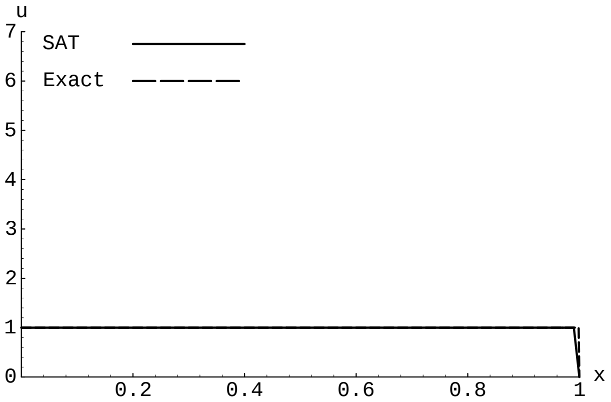

Figure 8: SAT, $R_C = 1000$.

4.2 A Steady State Two Dimensional Case

Here we shall consider a 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.)

$$u_t + au_x + bu_y = \frac{1}{R}u_{yy}; \qquad t \geq 0; 0 \leq x < 1; 0 \leq y \leq 1, \tag{4.2.1}$$

$$u(0, y, t) = \frac{1 - e^{bRy}}{1 - e^{bR}} + \frac{1}{10}bRe^{\frac{bRy}{2}} \sin \pi y, \tag{4.2.1a}$$

$$u(x,0,t) = 0, \tag{4.2.1b}$$

$$u(x,1,t) = 1. \tag{4.2.1c}$$

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 of this problem is:

$$u(x,y) = \frac{1 - e^{bRy}}{1 - e^{bR}} + \frac{1}{10} bRe^{\frac{bRy}{2}} \exp\left[\left(-\frac{b^2R^2}{4} - \pi^2\right)\frac{x}{Ra}\right] \sin \pi y. \tag{4.2.2}$$

Figure 9 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 10 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. [4].

The numerical solution of (4.2.1) using a standard central differencing scheme depends strongly on the value of b (at a given R). Figures 11 and 12 show the 3-D plot of $v_{j,k}$ with $b = -1$ and $b = -4/\sqrt{R} = -4/300$. Figs. 13 and 14 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 figure 11 has nothing to do with the "bumps" in the exact solution (see figure 10). The "peak" occurs way outside the boundary layer, and also the amplitude behavior with the x-coordinate is counter to that of figure 10. 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 15–18.

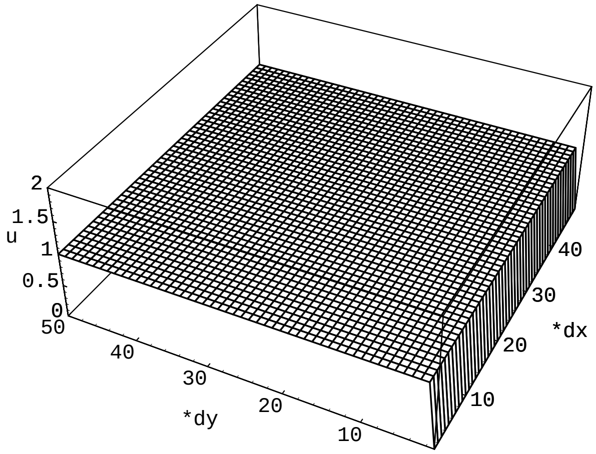

Figure 9: Exact solution.

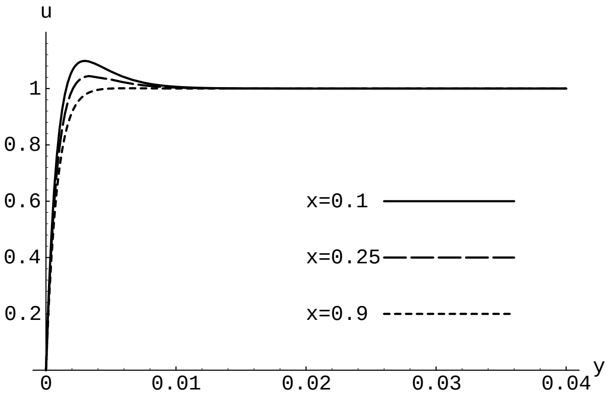

Figure 10: Exact solution near the boundary.

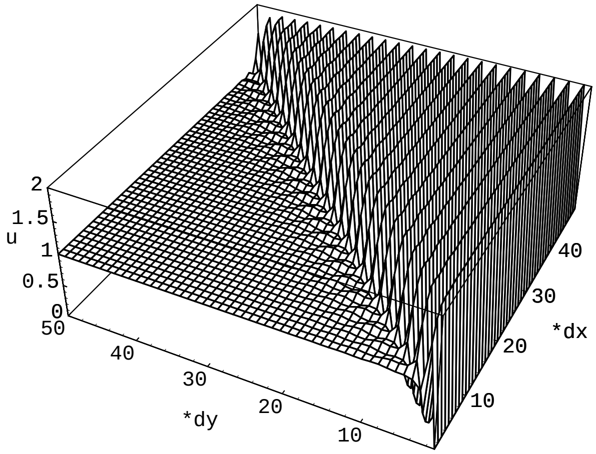

Figure 11: Standard scheme, $b = -1$.

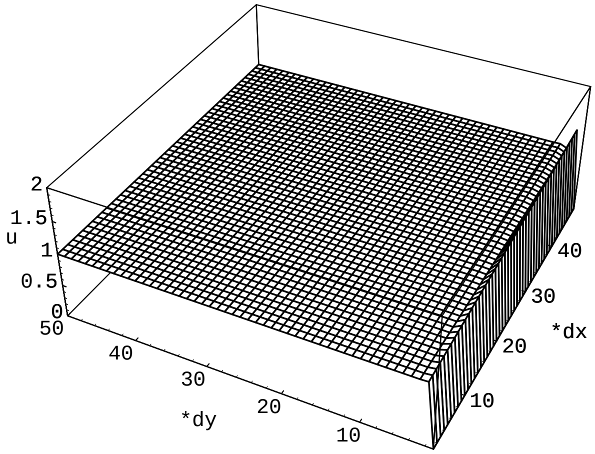

Figure 12: Standard scheme, $b = -4/300$.

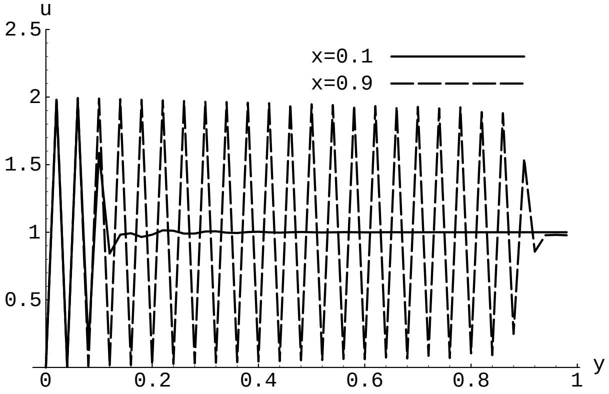

Figure 13: Standard scheme, $b = -1$.

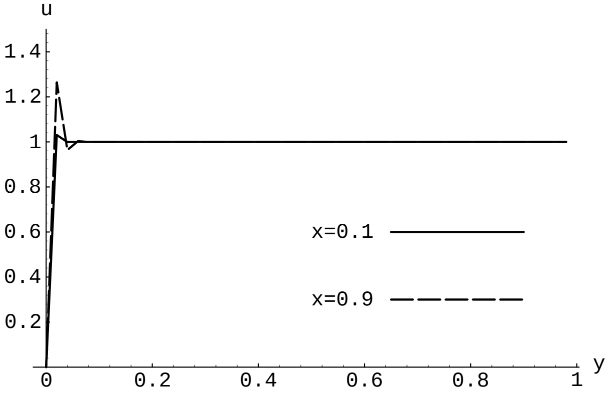

Figure 14: Standard scheme, $b = -4/300$.

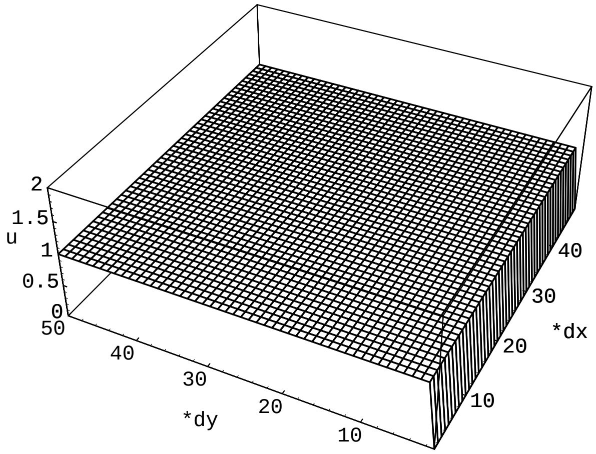

Figure 15: SAT, $b = -1$.

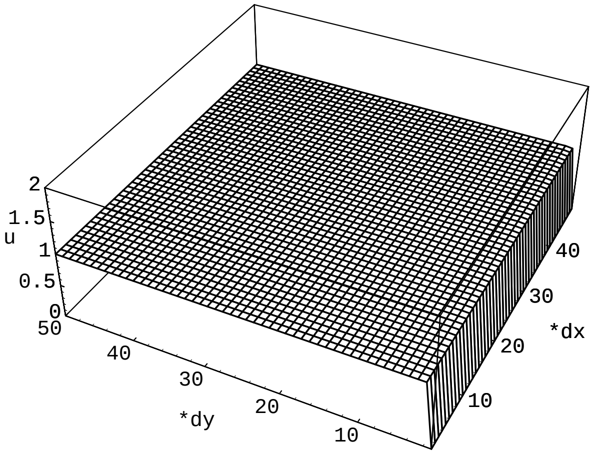

Figure 16: SAT, $b = -4/300$.

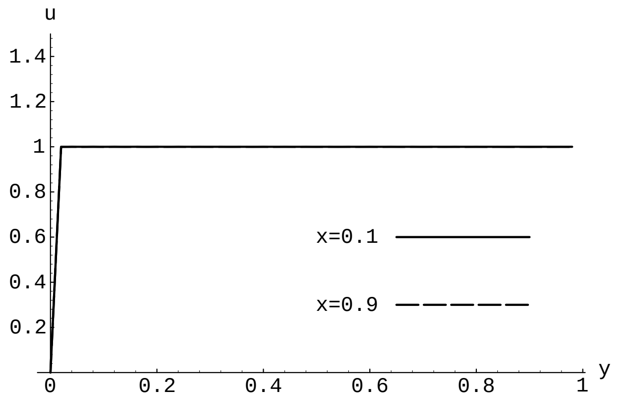

Figure 17: SAT, $b=-1$.

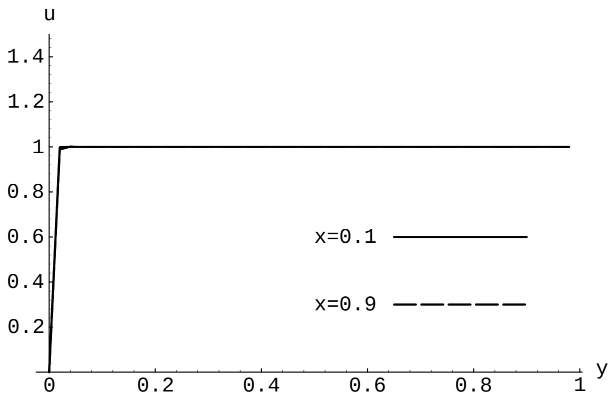

Figure 18: SAT, $b=-4/300$.

It should be noted (see table 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 run time 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 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 19.

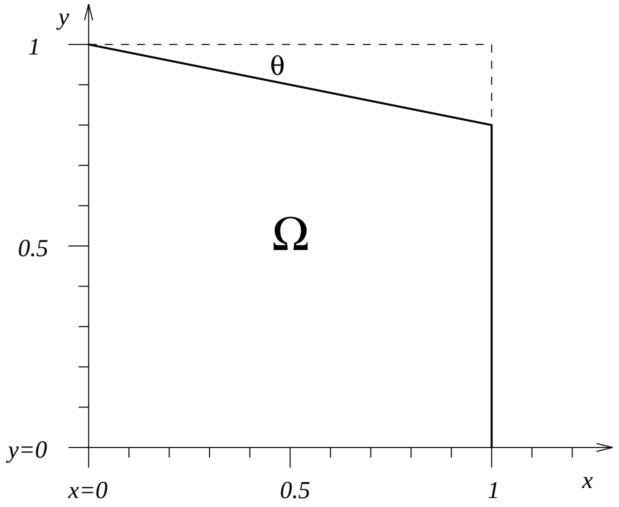

Figure 19: The trapezoid geometry.

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 2, for the case of $\theta = 3.9°$. As in [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 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) is insensitive to the variations in γ caused by the geometry of the domain.

	'time' to	L_2	L_1 norm	L_2 norm	L_∞ norm	max error
	'steady-state'	residual	of the error	of the error	of the 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 2: Trapezoid geometry results.

4.3 A 2-D time dependent example

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

$$u_t + au_x + bu_y = \frac{1}{R}u_{yy} + \sigma b \sin[k(x-at)]; \quad t \geq 0, \quad 0 \leq x < 1, \quad 0 \leq y \leq 1, \tag{4.3.1a}$$

$$u(x,y,0) = \frac{1-e^{bRy}}{1-e^{bR}} + \frac{bR}{10}e^{\frac{bRy}{2}}e^{-(\frac{b^2R^2}{4}+\pi^2)\frac{x}{Ra}}\sin \pi y + y\sigma \sin kx, \tag{4.3.1b}$$

$$u(0,y,t) = \frac{1-e^{bRy}}{1-e^{bR}} + \frac{bR}{10}e^{\frac{bRy}{2}}\sin \pi y - y\sigma \sin kat, \tag{4.3.1c}$$

$$u(x,0,t) = 0, \tag{4.3.1d}$$

$$u(x,1,t) = 1 + \sigma \sin[k(x-at)]. \tag{4.3.1e}$$

The exact solution of (4.3.1) is:

$$u(x,y,t) = \frac{1-e^{bRy}}{1-e^{bR}} + \frac{bR}{10}e^{\frac{bRy}{2}}e^{-(\frac{b^2R^2}{4}+\pi^2)\frac{x}{Ra}}\sin \pi y + y\sigma \sin[k(x-at)]. \tag{4.3.2}$$

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 20 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 21, 22 show the 3-D plots of $v_{j,k}$ for the standard and bounded-error schemes respectively. Figure 23, shows a x-profile of v at $y = .2$, for both schemes and the exact profile, for $b = 1$. Figure 24, gives the same profiles at $y = .8$. These plots bring out the differences in the phase errors of the numerical algorithms. Figures 25–28, repeat the same information as given in figure 21–24, 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.

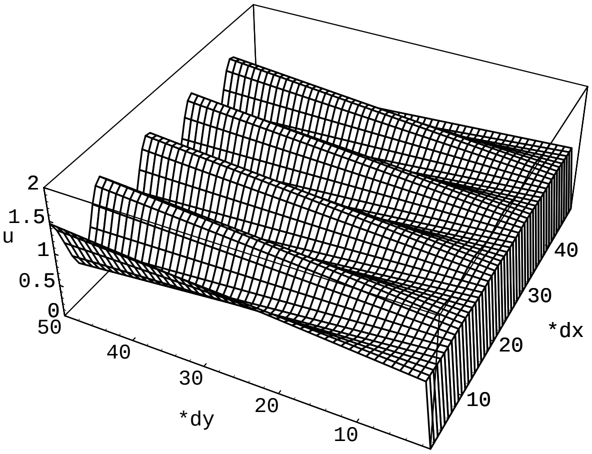

Figure 20: Exact solution.

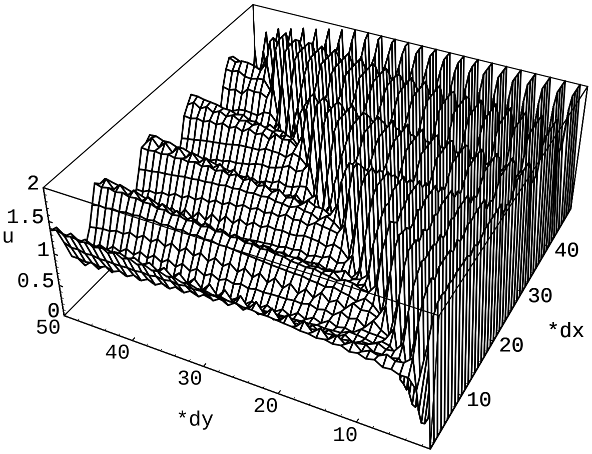

Figure 21: Standard scheme, $b = -1$.

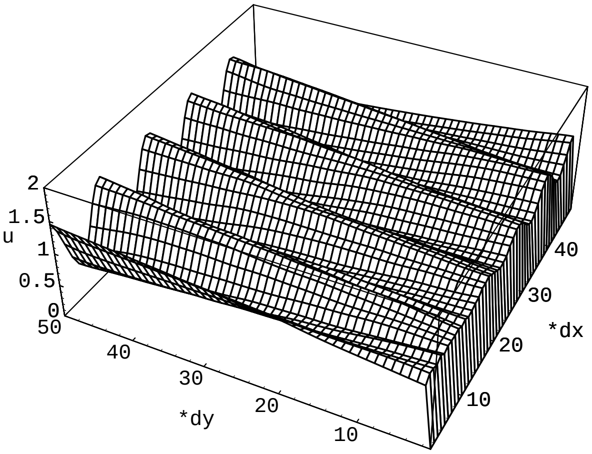

Figure 22: SAT, $b = -1$.

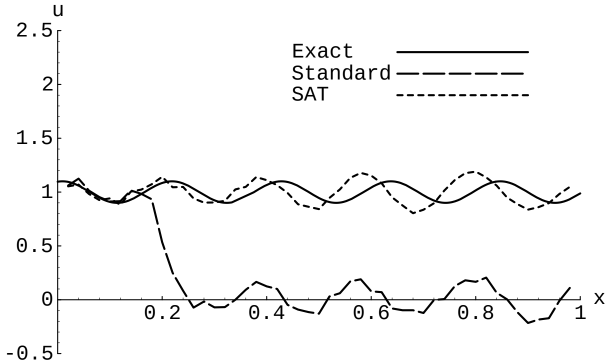

Figure 23: $b = -1$, $y = 0.2$ profiles.

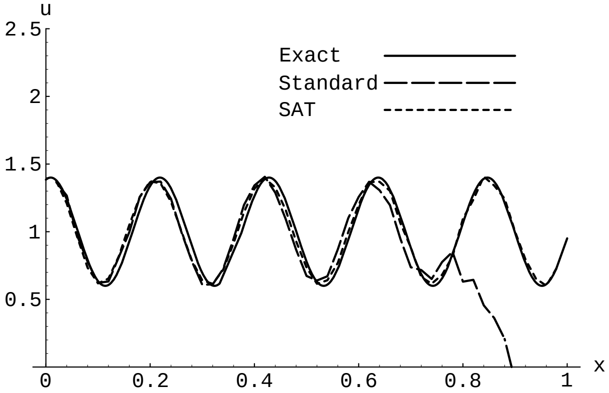

Figure 24: $b = -1$, $y = 0.8$ profiles.

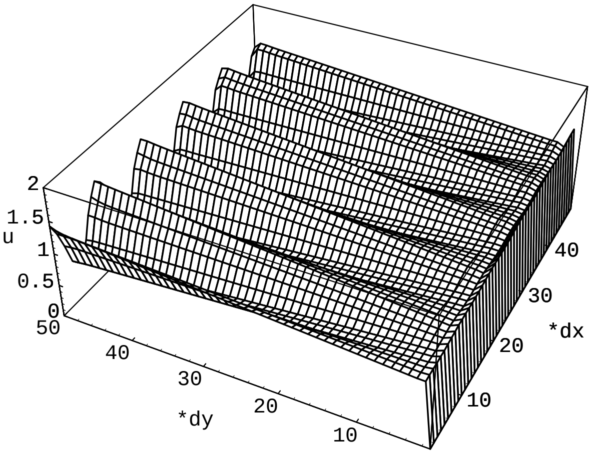

Figure 25: Standard scheme, $b = -4/300$.

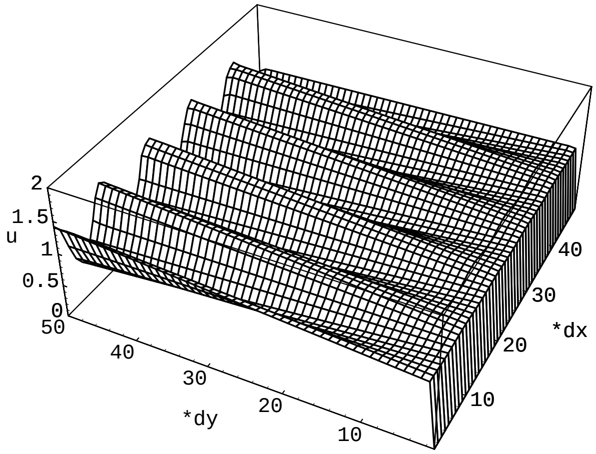

Figure 26: SAT, $b = -4/300$.

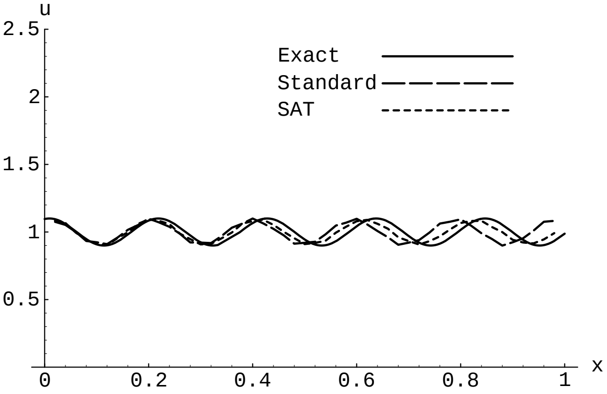

Figure 27: $b = -4/300$, $y = 0.2$ profiles.

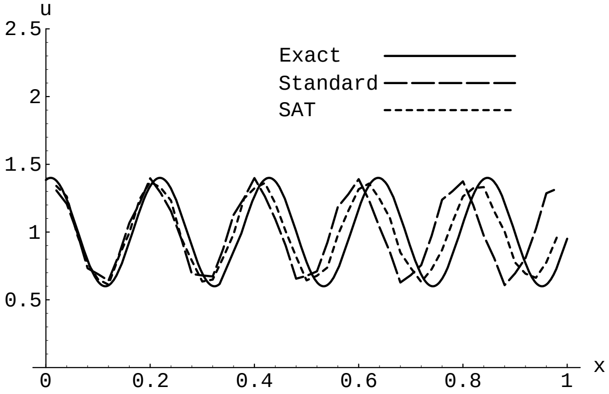

Figure 28: $b = -4/300$, $y = 0.8$ profiles.

5 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 [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 asymptotic temporal stability.

(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 boundedness of the error-norms carry over rigorously to the (linear) multi-dimensional cases.

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

Appendix I

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

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

where

$$c_k = \frac{1}{N-1}[(c_N - c_1)k + (Nc_1 - c_N)], \tag{A.2}$$

and

$$\tilde{c} = \frac{1}{2}(c_1 - c_N). \tag{A.3}$$

For $a < 0$ in (2.1a), 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

$$\tau_{L_H} = \frac{1}{2h}\text{diag }[\tau_{L_1}^{(H)}, \tau_{L_2}^{(H)}, 0, \ldots, 0, 0], \tag{A.4}$$

$$A_{L_H} = \begin{bmatrix} 1+\gamma_L & -\gamma_L & & & \\ 1+\gamma_L & -\gamma_L & & \mathbf{0} & \\ & & 0 & & \\ & & & \ddots & \\ & \mathbf{0} & & & 0 \end{bmatrix}. \tag{A.5}$$

Next we would like to show that $\tilde{M}_H = \frac{1}{2}(M_H + M_H^T)$ is non-negative definite,*then* $a\tilde{M}_H$ *is non-positive definite.* Using equations (A.1)–(A.5) we have

$$\tilde{M}_H = \frac{1}{4h}\begin{bmatrix} -4-2c_1-2(1+\gamma_L)\tau_1^{(H)} & 1+2c_1-(1+\gamma_L)\tau_2^{(H)}+\gamma_L\tau_1^{(H)} & 0 & & & \\ 1+2c_1-(1+\gamma_L)\tau_2^{(H)}+\gamma_L\tau_1^{(H)} & -2c_1+2\gamma_L\tau_2^{(H)} & 0 & & & \mathbf{0} \\ 0 & 0 & 0 & & & \\ & & & \ddots & & \\ \mathbf{0} & & & & 2c_N & -1-2c_N \\ & & & & -1-2c_N & 4+2c_N \end{bmatrix}. \tag{A.6}$$

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

$$\tilde{M}_H = \frac{1}{4h}[m_{H_1} + m_{H_2} + m_{H_3}], \tag{A.7}$$

where

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

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

$$m_{H_3} = c_N \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 2c_N & -1-2c_N \\ & & & -1-2c_N & 4+2c_N \end{bmatrix}. \tag{A.8}$$

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

$$\tau_1^{(H)} = -\frac{2+\delta}{1+\gamma_L} \qquad (\delta \geq 0), \tag{A.9}$$

$$\tau_2^{(H)} = \frac{1-\gamma_L(1-\delta)}{(1+\gamma_L)^2}. \tag{A.10}$$

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 $-\alpha\pi^2/R$, $0 < \alpha < .275$, as in the $a > 0$ case treated in the text.

References

[1] 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 *JCP*.

[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. Submitted to *JCP*.

[3] 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.

[4] 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.