JOURNAL OF COMPUTATIONAL PHYSICS **133,** 279–288 (1997)
ARTICLE NO. CP975653

Asymptotically Stable Fourth-Order Accurate Schemes for the Diffusion Equation on Complex Shapes[1]

Saul Abarbanel and Adi Ditkowski

School of Mathematical Sciences, Department of Applied Mathematics, Tel Aviv University, Tel Aviv, Israel

Received February 27, 1996; revised November 14, 1996

An algorithm which solves the multidimensional diffusion equation on complex shapes to fourth-order accuracy and is asymptotically stable in time is presented. 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. The ability of the paradigm to be applied to arbitrary geometric domains is an important feature of the algorithm. © 1997 Academic Press

1. INTRODUCTION

Recently there has been renewed interest in finite-difference algorithms of high order of accuracy (fourth and above), for both hyperbolic and parabolic pde's (see, for example, [1–3]). The advantages of high-order accuracy schemes, especially for truly time-dependent problems, are often offset by the difficulty of imposing stable boundary conditions. Even when the scheme is stable in the sense of Gustafsson, Kreiss, and Sundström (GKS), the error may increase exponentially in time.

This paper is concerned with fourth-order approximations to the long-time solutions of the diffusion equation in one and two dimensions, on irregular domains. By an irregular domain, we mean a body whose boundary points do not coincide with nodes of a rectangular mesh.

In Section 2 we develop the theory for the one-dimensional semidiscrete system resulting from the spatial differentiation used in the finite-difference algorithm. Energy methods are used in conjunction with simultaneous approximation terms (SAT) (see [1]) in order to find boundary conditions 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 truncation error.

In Section 3 it is shown how the methodology developed in Section 2 is used as a building block for the multidimensional algorithm, even for irregular shapes containing "holes".

Section 4 presents numerical results in two space dimensions illustrating the long-time temporal stability of the method, in contradistinction to "standard" methods for a Cartesian grid on irregular shapes.

2. THE ONE-DIMENSIONAL CASE

We consider the problem

$$\frac{\partial u}{\partial t} = k\frac{\partial^2 u}{\partial x^2} + f(x,t); \quad \Gamma_L \le x \le \Gamma_R, t \ge 0, k > 0 \tag{2.1a}$$

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

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

$$u(\Gamma_R, t) = g_R(t) \tag{2.1d}$$

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

Let us spatially discretize (2.1a) on the uniform grid presented in Fig. 1. 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 onto the grid in Fig. 1 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 the second partial derivative with respect to x, at "internal" points without specifying yet how it is being built. Then we may write

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

where $\mathbf{T}$ is the truncation error due to the numerical differentiation and $\mathbf{f}(t) = f(x_j, t)$, $1 \le j \le N$. The boundary

[1] This research was supported by the National Aeronautics and Space Administration under NASA Contract NAS1-19480 while the authors were in the residence of the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA. S. Abarbanel was also supported in part by the Air Force Office of Scientific Research under Grant AFOSR-F49620-95-1-0074, and by the Department of Energy under Grant DOE-DE-FG02-95ER25239.

0021-9991/97 $25.00

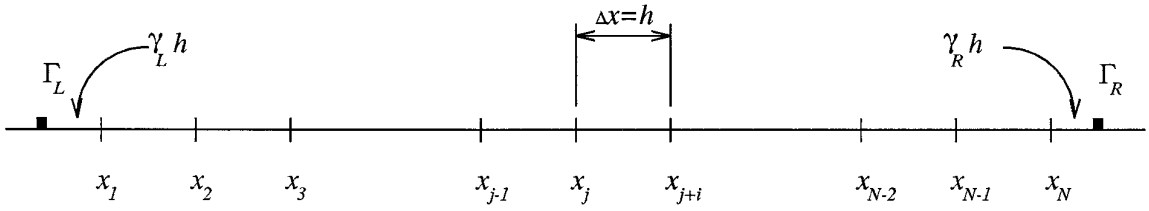

FIG. 1. One-dimensional grid.

vector $\mathbf{B}$ has entries whose values depend on g_L, g_R, γ_L, and γ_R in such a way that $\tilde{D}\mathbf{u} + \mathbf{B}$ represents the second derivative 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

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

where $\mathbf{v}(t)$ is the numerical approximation to the projection $\mathbf{u}(t)$. An equation for the solution error vector, $\vec{\varepsilon}(t) = \mathbf{u}(t) - \mathbf{v}(t)$, can be found by subtracting (2.3) from (2.2):

$$\frac{d}{dt}\vec{\varepsilon} = k\tilde{D}\vec{\varepsilon}(t) + k\mathbf{T}(t). \tag{2.4}$$

Our requirement for *temporal stability is that* $\|\vec{\varepsilon}\|$, *the* L_2 *norm of* $\vec{\varepsilon}$, *be bounded by a "constant" proportional to* h^m (m being the spatial order of accuracy) *for all* $t < \infty$. Note that this definition is more severe than either the GKS stability criterion [4] or the definition in [1].

It can be shown that if $\tilde{D}$ is constructed in a standard manner, i.e., the numerical second derivative is symmetric away from the boundaries, and near the boundaries one uses nonsymmetric differentiation, then there are ranges of values of γ_R and γ_L for which $\tilde{D}$ is not negative definite. Since in the multidimensional case one may encounter all values of $0 \leq \gamma_L, \gamma_R < 1$, this is unacceptable.

The rest of this section is devoted to the construction of a scheme of fourth-order spatial accuracy, which is temporally stable for all γ_L, γ_R.

The basic idea is to use a penalty-like term as in the SAT procedure of Ref. [1]; here, however, it will be modified and applied in a different manner.

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

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

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 semidiscrete problem for $\mathbf{v}(t)$ be, instead of (2.3),

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

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

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

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

$$\tau_L = \mathrm{diag}(\tau_{L_1}, \tau_{L_2}, ..., \tau_{L_N}); \quad \tau_R = \mathrm{diag}(\tau_{R_1}, ..., \tau_{R_N}). \tag{2.8}$$

Subtracting (2.6) from (2.5) we get

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

where

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

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

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

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

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

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

$$(\vec{\varepsilon}, (M + M^T)\vec{\varepsilon}/2) \le -c_0 \|\vec{\varepsilon}\|^2 \quad (c_0 > 0). \tag{2.12}$$

Equation (2.10) then becomes

$$\frac{1}{2}\frac{d}{dt}\|\vec{\varepsilon}\|^2 \le -kc_0\|\vec{\varepsilon}\|^2 + k(\vec{\varepsilon}, \mathbf{T}_1)$$

and using Schwarz's inequality we get after dividing by $\|\vec{\varepsilon}\|$

$$\frac{d}{dt}\|\vec{\varepsilon}\| \le -kc_0\|\vec{\varepsilon}\| + k\|\mathbf{T}_1\|$$

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

$$\|\vec{\varepsilon}\| \le \frac{\|\mathbf{T}_1\|_M}{c_0}(1 - e^{kc_0 t}), \tag{2.13}$$

where the *"constant"* $\|\mathbf{T}_1\|_M = \max_{0\le\tau\le t}\|\mathbf{T}_1(\tau)\|$.

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.

The rest of this section is devoted to this task for the case of $m = 4$, i.e., a fourth-order accurate finite-difference algorithm.

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

$$\frac{1}{12h^2}\begin{bmatrix}
45 & -154 & 214 & -156 & 61 & -10 & & & & & & \\
10 & -15 & -4 & 14 & -6 & 1 & & & & & & \\
-1 & 16 & -30 & 16 & -1 & & & & & & & \\
 & -1 & 16 & -30 & 16 & -1 & & & & & & \\
 & & -1 & 16 & -30 & 16 & -1 & & & & & \\
 & & & & & \ddots & & & & & & \\
 & & & & & & \ddots & & & & & \\
 & & & & & & & \ddots & & & & \\
 & & & & & & -1 & 16 & -30 & 16 & -1 & \\
 & & & & & & & -1 & 16 & -30 & 16 & -1 \\
 & & & & & & 1 & -6 & 14 & -4 & -15 & 10 \\
 & & & & & & -10 & 61 & -156 & 214 & -154 & 45
\end{bmatrix}. \tag{2.14}$$

The upper two rows and the lower two rows represent nonsymmetric 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 thus neither is the symmetric part of $\frac{1}{2}(D + D^T)$.

The matrices A_L, A_R, τ_L, and τ_R and the construction of the symmetric part of M are given in the Appendix.

Indeed it can be 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 (2.13) is valid. For details the reader is referred to [5].

3. THE TWO-DIMENSIONAL CASE

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

$$\frac{\partial u}{\partial t} = k\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + f(x, y; t); \quad (x, y) \in \Omega; t \ge 0; k > 0 \tag{3.1a}$$

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

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

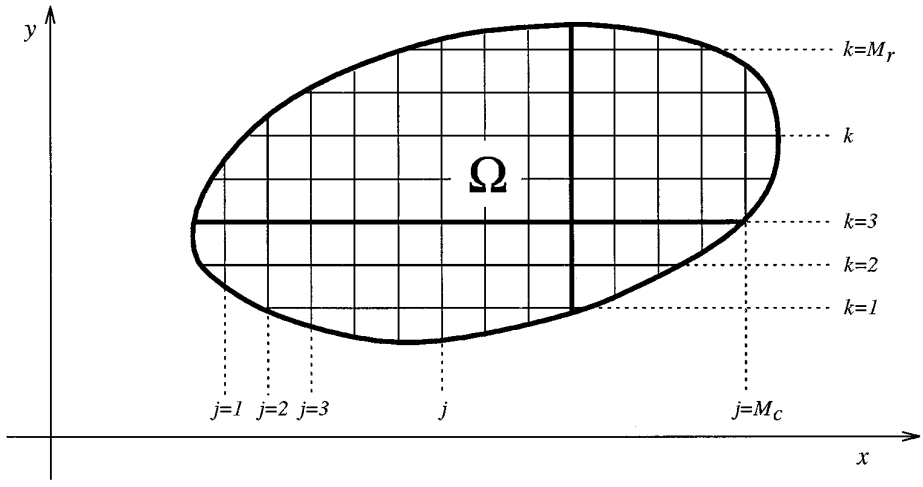

FIG. 2. Two-dimensional grid.

We shall refer to the grid representation in Fig. 2. We have M_R rows and M_C columns inside Ω. Each row and each column have a discretized structure as in the 1-D case; see Fig. 1. Let the number of grid points in the kth row be denoted by R_k and similarly let the number of grid points in the jth 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,

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

Thus, we have arranged the solution projection array 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:

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

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

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

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

The second-derivative operator $\partial^2/\partial x^2$ in (3.1a) is represented on the kth row by the differentiation matrix $D_k^{(x)}$, whose structure is given by (2.14). Similarly let $\partial^2/\partial y^2$ be given on the jth columns by $D_j^{(y)}$, whose structure is also given by (2.14). With this notation the Laplacian of the solution projection is

$$\begin{aligned}\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) u_{ij}(t) &= (\mathscr{D}^{(x)}\mathbf{U} + \mathscr{D}^{(y)}\mathbf{U}^{(C)}\\ &\quad + \mathbf{T}_e^{(x)} + \mathbf{T}_e^{(y)})_{ij},\end{aligned} \qquad (3.5)$$

where

$$\begin{aligned}\mathscr{D}^{(x)} &= \begin{bmatrix} D_1^{(x)} & & \\ & D_2^{(x)} & \\ & & D_{M_R}^{(x)} \end{bmatrix};\\ \mathscr{D}^{(y)} &= \begin{bmatrix} D_1^{(y)} & & \\ & D_2^{(y)} & \\ & & D_{M_C}^{(y)} \end{bmatrix},\end{aligned} \qquad (3.6)$$

where $\mathscr{D}^{(x)}$ and $\mathscr{D}^{(y)}$ are $(l \times l)$ matrices and have the block structures shown. $\mathbf{T}_e^{(x)}$ and $\mathbf{T}_e^{(y)}$ are the truncation errors associated with $\mathscr{D}^{(x)}$ and $\mathscr{D}^{(y)}$, respectively. We now call attention to the fact that $\mathscr{D}^{(x)}$ and $\mathscr{D}^{(y)}$ do not operate on the same vector. This is fixed using (3.4):

$$\nabla^2 u_{ij}(t) = \nabla^2 \mathbf{U} = (\mathscr{D}^{(x)} + P^{\mathrm{T}}\mathscr{D}^{(y)}P)\mathbf{U} + \mathbf{T}_{\mathrm{e}}^{(x)} + P^{\mathrm{T}}\mathbf{T}_{\mathrm{e}}^{(y)}. \tag{3.7}$$

Thus (3.1a) becomes, by analogy to (2.5),

$$\frac{d\mathbf{U}}{dt} = k(\mathscr{D}^{(x)} + P^{\mathrm{T}}\mathscr{D}^{(y)}P)\mathbf{U} + k(\mathbf{T}_{\mathrm{e}}^{(x)} + P^{\mathrm{T}}T_{\mathrm{e}}^{(y)}) + \mathbf{f}(t), \tag{3.8}$$

where $\mathbf{f}(t)$ is $f(x, y; t)$ arranged by *rows* as a vector.

Before proceeding to the semidiscrete problem let us define

$$M_k^{(x)} = D_k^{(x)} - \tau_{\mathrm{L}_k} A_{\mathrm{L}_k} - \tau_{\mathrm{R}_k} A_{\mathrm{R}_k}, \tag{3.9}$$

where τ_{L_k} and A_{L_k} are the τ_{L} and A_{L} defined in the Appendix, appropriate to the kth row; similarly for τ_{R_k} and A_{R_k}. In the same way, define

$$M_j^{(y)} = D_j^{(y)} - \tau_{\mathrm{B}_j} A_{\mathrm{L}_j} - \tau_{\mathrm{T}_j} A_{\mathrm{R}_j}, \tag{3.10}$$

where B and T stand for bottom and top.

We can now write the semidiscrete problem by analogy to (2.6)

$$\frac{d\mathbf{V}}{dt} = k(\mathscr{M}^{(x)} + P^{\mathrm{T}}\mathscr{M}^{(y)}P)\mathbf{V} + k\mathbf{G}^{(x)} + kP^{\mathrm{T}}\mathbf{G}^{(y)} + \mathbf{f}(t), \tag{3.11}$$

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

$$\mathscr{M}^{(x)} = \begin{bmatrix} M_1^{(x)} & & & \\ & M_2^{(x)} & & \\ & & & M_{M_R}^{(x)} \end{bmatrix}; \qquad \mathscr{M}^{(y)} = \begin{bmatrix} M_1^{(y)} & & & \\ & M_2^{(y)} & & \\ & & & M_{M_C}^{(y)} \end{bmatrix}; \tag{3.12}$$

and

$$\begin{aligned} \mathbf{G}^{(x)} &= [(\tau_{\mathrm{L}_1}\mathbf{g}_{\mathrm{L}_1} + \tau_{\mathrm{R}_1}\mathbf{g}_{\mathrm{R}_1}), \ldots, (\tau_{\mathrm{L}_k}\mathbf{g}_{\mathrm{L}_k} + \tau_{\mathrm{R}_k}\mathbf{g}_{\mathrm{R}_k}), \ldots, (\tau_{\mathrm{L}_{M_R}}\mathbf{g}_{\mathrm{L}_{M_R}} + \tau_{\mathrm{R}_{M_R}}\mathbf{g}_{\mathrm{R}_{M_R}})], \\ \mathbf{G}^{(y)} &= [(\tau_{\mathrm{B}_1}\mathbf{g}_{\mathrm{B}_1} + \tau_{\mathrm{T}_1}\mathbf{g}_{\mathrm{T}_1}), \ldots, (\tau_{\mathrm{B}_j}\mathbf{g}_{\mathrm{B}_j} + \tau_{\mathrm{T}_j}\mathbf{g}_{\mathrm{T}_j}), \ldots, (\tau_{\mathrm{B}_{M_C}}\mathbf{g}_{\mathrm{B}_{M_C}} + \tau_{\mathrm{T}_{M_C}}\mathbf{g}_{\mathrm{T}_{M_C}})]. \end{aligned} \tag{3.13}$$

Subtracting (3.11) from (3.8) we get in a fashion similar to the derivation of (2.9)

$$\frac{d\mathbf{E}}{dt} = k[\mathscr{M}^{(x)} + P^{\mathrm{T}}\mathscr{M}^{(y)}P]\mathbf{E} + k\mathbf{T}_2, \tag{3.14}$$

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

$$\frac{1}{2}\frac{d}{dt}\|\mathbf{E}\|^2 = k(\mathbf{E}, (\mathscr{M}^{(x)} + P^{\mathrm{T}}\mathscr{M}^{(y)}P)\mathbf{E}) + k(\mathbf{E}, \mathbf{T}_2). \tag{3.15}$$

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

$$\tfrac{1}{2}[(\mathscr{M}^{(x)} + \mathscr{M}^{(x)^{\mathrm{T}}}) + P^{\mathrm{T}}(\mathscr{M}^{(y)} + \mathscr{M}^{(y)^{\mathrm{T}}})P]. \tag{3.16}$$

Clearly $\mathscr{M}^{(x)} + \mathscr{M}^{(x)^{\mathrm{T}}}$ and $\mathscr{M}^{(y)} + \mathscr{M}^{(y)^{\mathrm{T}}}$ are block-diagonal matrices with typical blocks given by $M_k^{(x)} + M_k^{(x)^{\mathrm{T}}}$ and $M_j^{(y)} + M_j^{(y)^{\mathrm{T}}}$. We have already shown in the one-dimensional case that each one of those blocks is negative definite and bounded away from zero by $\pi^2/24$. Therefore the operator (3.16) is also negative definite and bounded away from zero. The rest of the proof follows the one-dimensional case and thus the norm of the error, $\|\mathbf{E}\|$, is bounded by a constant.

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 Fig. 3) are taken as *separate* vectors.

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 $l \times l$. Note that adding more "holes" inside $\partial\Omega$ does not change the general approach.

4. NUMERICAL EXAMPLE

In this section we describe numerical results for the problem

$$\frac{\partial u}{\partial t} = k(u_{xx} + u_{yy}) + f(x, y, t) \quad (x, y) \in \Omega, t > 0, \tag{4.1}$$

where Ω is the region contained between a circle of radius $r_0 = 1/2$ and inner circle of radius $r_{\mathrm{i}} \leq 0.1$. The inner circle is *not concentric* with the outer one. Specifically Ω is described by

$$\{(x - 0.5)^2 + (y - 0.5)^2 \leq \tfrac{1}{4}\} \cap \{(x - 0.6)^2 + (y - 0.5)^2 \geq (0.1 - \delta)^2; 0 < \delta < 0.1\}. \tag{4.2}$$

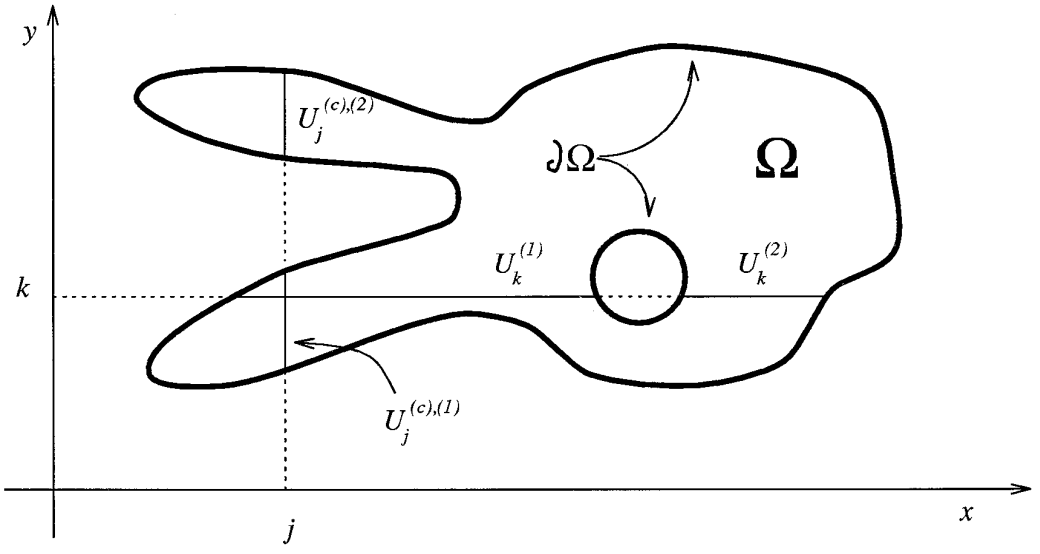

FIG. 3. Two-dimensional grid, nonconvex domain.

The Cartesian grid in which Ω is embedded spans $0 \le x, y \le 1$. We took $\Delta x = \Delta y$, and ran several cases with $\Delta x = 1/50, 1/75, 1/100$. The geometry thus looks as shown in Fig. 4.

The source function $f(x, y, t)$ was chosen different from zero so that we could assign an exact analytic solution to (4.1). This enables one to compute the error $\varepsilon_{ij} = U_{ij} - V_{ij}$ "exactly" (to machine accuracy). We chose $k = 1$ and

$$u(x, y, t) = 1 + \cos(10t - 10x^2 - 10y^2). \tag{4.3}$$

This leads to

$$f(x, y, t) = 400(x^2 + y^2)\cos(10t - 10x^2 - 10y^2) - 50\sin(10t - 10x^2 - 10y^2). \tag{4.4}$$

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

The problem (4.1), (4.2), (4.4) was solved using both a standard fourth-order algorithm (a 2-D version of (2.3)) and the new SAT, or bounded-error, approach described in Section 3. The temporal advance was via a fourth-order Runge–Kutta.

The standard algorithm was run for $\Delta x = 1/50$ and a range of $0 \le \delta < 0.01$ ($0.09 < r_i \le 0.1$). We found that for $\delta \ge 0.0017323$, the runs were stable and the error was bounded for "long" times (10^5 time steps, or equivalently $t = 2$). For $0 \le \delta < 0.0017233$ the results began to diverge exponentially from the analytic solution. The "point of

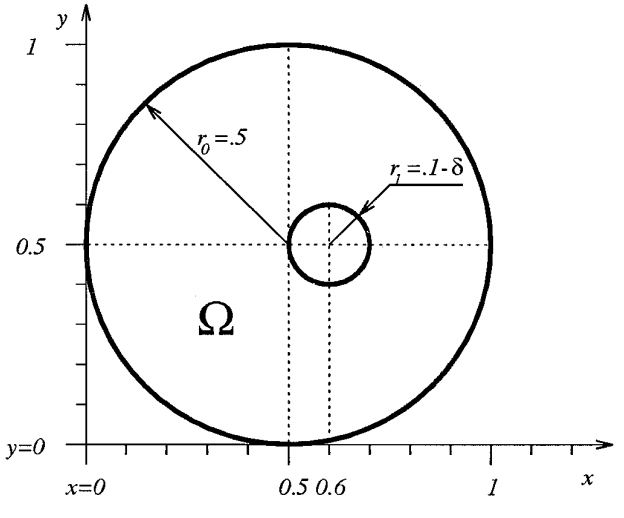

FIGURE 4

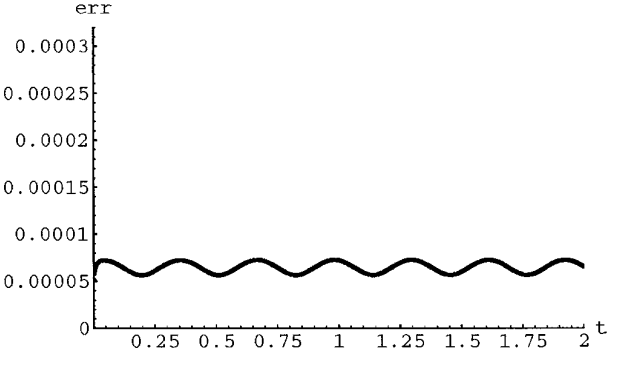

FIG. 5. $\delta = 0.0017325$, standard scheme.

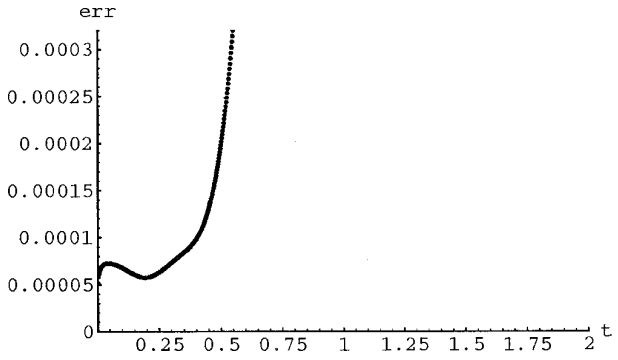

FIG. 6. $\delta = 0.0017323$, standard scheme.

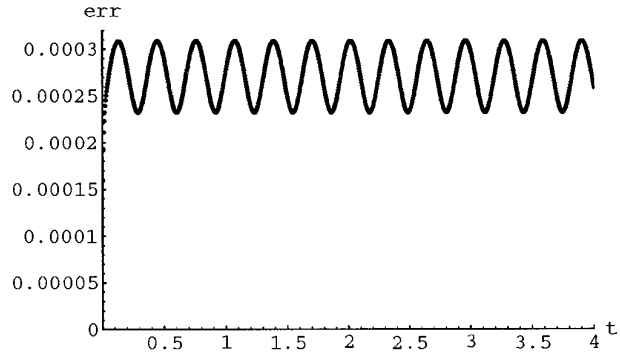

FIG. 8. $\delta = 0$, SAT scheme.

departure'' depended on δ. A discussion of these results is deferred to the next section. Figures 5, 6, and 7 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 described in Section 3 (see Eq. (3.5)), and the results are shown in Figs. 8, 9, 10, and 11. 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 $1/100$. Figures 12, 13, and 14 show the logarithmic slope of the L_2, L_1, and L_∞ errors to be less than -4; i.e., we indeed have a fourth-order method. 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.

A study of the effect of the size of Δt shows that the instabilities exhibited above are due to the time step being near the CFL limit. It is interesting that this CFL limit depends so strongly on the geometry.

5. CONCLUSIONS

(i) The theoretical results show that one must be very careful when using an algorithm whose differentiation matrix, and thus 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 ensures, in the 1-D case, the temporally asymptotic stability. In the 2-D case, even though each of the block submatrices of the $l \times l$ x-and-y differentiation matrices has only negative (real-part) eigenvalues, it is not ensured that the sum of the two $l \times l$ 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 a ''paradoxical'' situation, where 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.

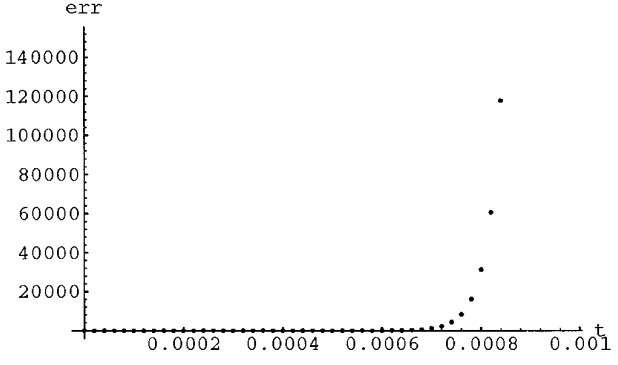

FIG. 7. $\delta = 0.0015$, standard scheme.

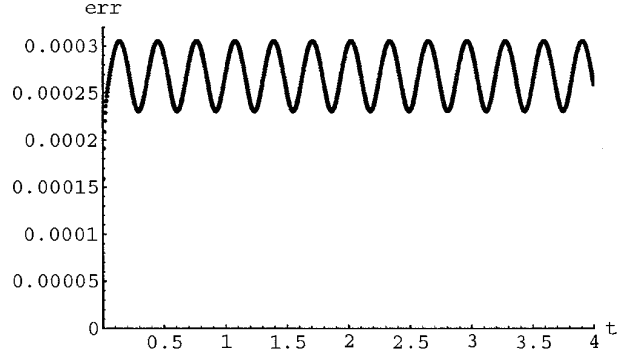

FIG. 9. $\delta = 0.0015$, SAT scheme.

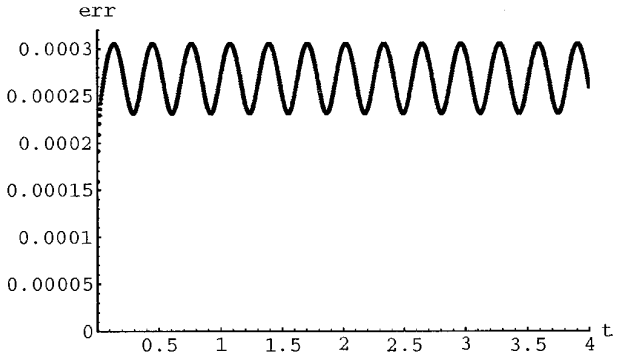

FIG. 10. $\delta = 0.0017323$, SAT scheme.

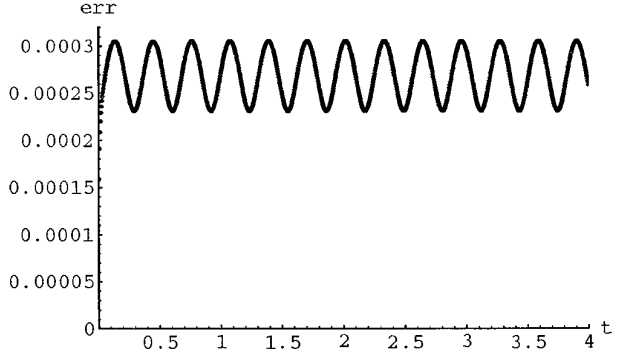

FIG. 11. $\delta = 0.0017325$, SAT scheme.

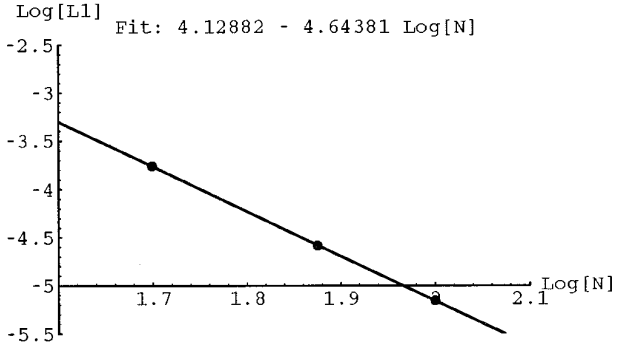

FIG. 12. Order of accuracy L_1.

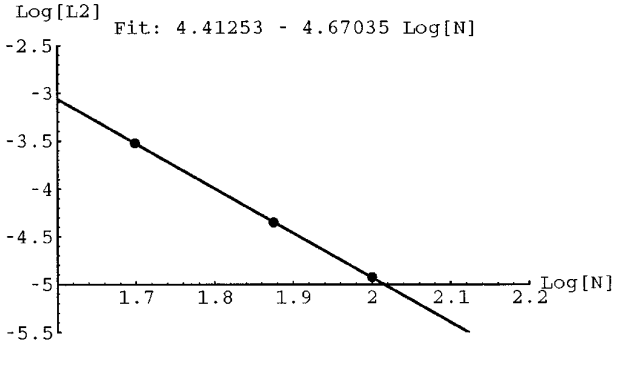

FIG. 13. Order of accuracy L_2.

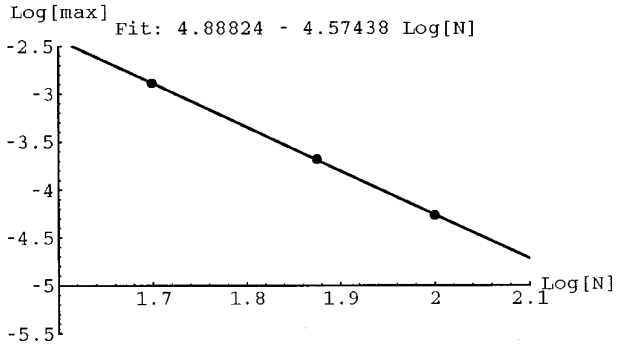

FIG. 14. Order of accuracy L_∞.

It is also interesting to note that if one uses explicit standard methods then the allowable CFL may decrease extremely rapidly with the change in the geometry that causes the decrease in the γ's. This point is brought out in Figs. 5 to 7.

(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 k, in the equation

$$u_t = k\Delta^2 u,$$

is a function of the spatial coordinates, $k = k(x, y, z)$, the previous analysis goes through but the energy estimate for the error is now for a different, but equivalent, norm.

APPENDIX

In order to construct M we need to specify A_{L}, A_{R}, τ_{L}, and τ_{R}. We construct A_{L} as

$$A_{\mathrm{L}} = A_{\alpha}^{(\mathrm{L})} + c_{\mathrm{L}} A_{\mathrm{e}}^{(\mathrm{L})}, \tag{A.1}$$

where

$$A_{\alpha}^{(\mathrm{L})} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & 0 & \dots & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & 0 & \dots & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & 0 & \dots & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & 0 & \dots & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & 0 & \dots & 0 \\ \vdots & & & & & & & \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & 0 & \dots & 0 \end{bmatrix}, \tag{A.2}$$

$$c_{\mathrm{L}} = \mathrm{diag}[-20\alpha_1/71, 0, \dots, 0], \tag{A.3}$$

$$A_e^{(L)} = \begin{bmatrix} -1 & 5 & -10 & 10 & -5 & 1 & 0 & \dots & 0 \\ -1 & 5 & -10 & 10 & -5 & 1 & 0 & \dots & 0 \\ \vdots \\ -1 & 5 & -10 & 10 & -5 & 1 & 0 & \dots & 0 \end{bmatrix}. \qquad (A.4)$$

The α's are given by

$$\begin{aligned} \alpha_1 &= 1 + \frac{25}{12}\gamma_L + \frac{35}{24}\gamma_L^2 + \frac{5}{12}\gamma_L^3 + \frac{1}{24}\gamma_L^4 \\ \alpha_2 &= -\left(4\gamma_L + \frac{13}{3}\gamma_L^2 + \frac{3}{2}\gamma_L^3 + \frac{1}{6}\gamma_L^4\right) \\ \alpha_3 &= 3\gamma_L + \frac{19}{4}\gamma_L^2 + 2\gamma_L^3 + \frac{1}{4}\gamma_L^4 \\ \alpha_4 &= -\left(\frac{4}{3}\gamma_L + \frac{7}{3}\gamma_L^2 + \frac{7}{6}\gamma_L^3 + \frac{1}{6}\gamma_L^4\right) \\ \alpha_5 &= \frac{1}{4}\gamma_L + \frac{11}{24}\gamma_L^2 + \frac{1}{4}\gamma_L^3 + \frac{1}{24}\gamma_L^4. \end{aligned} \qquad (A.5)$$

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 represent $(\partial^5 v_1/\partial x^5)h^5$. Since c_L (see (A.3)) 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 (2.11) or (2.6) we must define τ_L,

$$\tau_L = \frac{1}{12h^2}\,\mathrm{diag}[\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, 0, \dots, 0], \qquad (A.6)$$

where

$$\begin{aligned} \tau_1 &= 71/2\alpha_1 \\ \tau_2 &= (-94 - \alpha_2\tau_1)/\alpha_1 \\ \tau_3 &= (113 - \alpha_3\tau_1)/\alpha_1 \\ \tau_4 &= (-56 - \alpha_4\tau_1)/\alpha_1 \\ \tau_5 &= (11 - \alpha_5\tau_1)/\alpha_1. \end{aligned} \qquad (A.7)$$

The right boundary treatment is constructed in a similar fashion, and the formulae corresponding to (A.1)–(A.7) become

$$A_R = A_\alpha^{(R)} + c_R A_e^{(R)}, \qquad (A.8)$$

$$A_\alpha^{(R)} = \begin{bmatrix} 0 & \dots & 0 & \alpha_{N-4} & \alpha_{N-3} & \alpha_{N-2} & \alpha_{N-1} & \alpha_N \\ 0 & \dots & 0 & \alpha_{N-4} & \alpha_{N-3} & \alpha_{N-2} & \alpha_{N-1} & \alpha_N \\ 0 & \dots & 0 & \alpha_{N-4} & \alpha_{N-3} & \alpha_{N-2} & \alpha_{N-1} & \alpha_N \\ 0 & \dots & 0 & \alpha_{N-4} & \alpha_{N-3} & \alpha_{N-2} & \alpha_{N-1} & \alpha_N \\ 0 & \dots & 0 & \alpha_{N-4} & \alpha_{N-3} & \alpha_{N-2} & \alpha_{N-1} & \alpha_N \\ \vdots \\ 0 & \dots & 0 & \alpha_{N-4} & \alpha_{N-3} & \alpha_{N-2} & \alpha_{N-1} & \alpha_N \end{bmatrix}, \qquad (A.9)$$

$$c_R = \mathrm{diag}[0, 0, \dots, 0, -20\alpha_N/71], \qquad (A.10)$$

$$A_e^{(R)} = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 & -5 & 10 & -10 & 5 & -1 \\ 0 & 0 & \dots & 0 & 1 & -5 & 10 & -10 & 5 & -1 \\ \vdots \\ 0 & 0 & \dots & 0 & 1 & -5 & 10 & -10 & 5 & -1 \end{bmatrix}. \qquad (A.11)$$

The α's are here

$$\begin{aligned} \alpha_N &= 1 + \tfrac{25}{12}\gamma_R + \tfrac{35}{24}\gamma_R^2 + \tfrac{5}{12}\gamma_R^3 + \tfrac{1}{24}\gamma_R^4 \\ \alpha_{N-1} &= -(4\gamma_R + \tfrac{13}{3}\gamma_R^2 + \tfrac{3}{2}\gamma_R^3 + \tfrac{1}{6}\gamma_R^4) \\ \alpha_{N-2} &= 3\gamma_R + \tfrac{19}{4}\gamma_R^2 + 2\gamma_R^3 + \tfrac{1}{4}\gamma_R^4 \\ \alpha_{N-3} &= -(\tfrac{4}{3}\gamma_R + \tfrac{7}{3}\gamma_R^2 + \tfrac{7}{6}\gamma_R^3 + \tfrac{1}{6}\gamma_R^4) \\ \alpha_{N-4} &= \tfrac{1}{4}\gamma_R + \tfrac{11}{24}\gamma_R^2 + \tfrac{1}{4}\gamma_R^3 + \tfrac{1}{24}\gamma_R^4 \end{aligned} \qquad (A.12)$$

$$\tau_R = \frac{1}{12h^2}\,\mathrm{diag}[0, \dots, \tau_{N-4}, \tau_{N-3}, \tau_{N-2}, \tau_{N-1}, \tau_N], \qquad (A.13)$$

$$\begin{aligned} \tau_N &= 71/2\alpha_N \\ \tau_{N-1} &= (-94 - \alpha_{N-1}\tau_N)/\alpha_N \\ \tau_{N-2} &= (113 - \alpha_{N-2}\tau_N)/\alpha_N \\ \tau_{N-3} &= (-56 - \alpha_{N-3}\tau_N)/\alpha_N \\ \tau_{N-4} &= (11 - \alpha_{N-4}\tau_N)/\alpha_N. \end{aligned} \qquad (A.14)$$

We are now ready to construct

$$\begin{aligned} \tfrac{1}{2}(M + M^T) = \tfrac{1}{2}\{D + D^T &- [\tau_L(A_\alpha^{(L)} + c_L A_e^{(L)}) \\ &+ \tau_R(A_\alpha^{(R)} + c_R A_e^{(R)})] \\ &- [\tau_L(A_\alpha^{(L)} + c_L A_e^{(L)}) \\ &+ \tau_R(A_\alpha^{(R)} + c_R A_e^{(R)})]^T\}. \end{aligned} \qquad (A.15)$$

Upon using equations (2.14) and (A.1)–(A.15) one gets

$$\frac{M+M^{\mathrm{T}}}{2}=\frac{1}{24h^2}\begin{bmatrix} & & & & & & 0 & & & & & & & & \\ & & W^{(\mathrm{L})} & & & & 0 & & & & & & & & \\ & & & & & & -2 & 0 & & & & & & & \\ & & & & & & 32 & -2 & & & & & 0 & & \\ 0 & \dots & 0 & -2 & 32 & -60 & 32 & -2 & & & & & & & \\ & & & & -2 & 32 & -60 & 32 & -2 & & & & & & \\ & & & & & & -2 & 32 & -60 & 32 & -2 & & & & \\ & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \\ & & & & & & & -2 & 32 & -60 & 32 & -2 & 0 & \dots & 0 \\ & & & 0 & & & & & -2 & 32 & & & & & \\ & & & & & & & & & -2 & & & & & \\ & & & & & & & & & 0 & & & W^{(\mathrm{R})} & & \\ & & & & & & & & & 0 & & & & & \end{bmatrix}. \quad \text{(A.16)}$$

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

$$W^{(\mathrm{L})} = W_1^{(\mathrm{L})} + W_2^{(\mathrm{L})} \quad \text{(A.17)}$$

$$W^{(\mathrm{R})} = W_1^{(\mathrm{R})} + W_2^{(\mathrm{R})} \quad \text{(A.18)}$$

$$W_{1_{ij}}^{(\mathrm{L})} = \begin{Bmatrix} 0, & i=1 \text{ or } j=1 \\ -(\alpha_i\tau_j + \alpha_j\tau_i), & i,j \neq 1 \end{Bmatrix}, \quad 1 < i,j < 5 \quad \text{(A.19)}$$

$$W_{1_{ij}}^{(\mathrm{R})} = \begin{Bmatrix} 0, & i=N \text{ or } j=N \\ -(\alpha_{N-i}\tau_{N-j} + \alpha_{N-j}\tau_{N-i}) \end{Bmatrix}, \quad 0 \leq N-i, N-j \leq 4 \quad \text{(A.20)}$$

$$W_2^{(\mathrm{L})} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -30 & 12 & 13 & -6 & 1 \\ 0 & 12 & -60 & 32 & -2 & 0 \\ 0 & 13 & 32 & -60 & 32 & -2 \\ 0 & -6 & -2 & 32 & -60 & 32 \\ 0 & 1 & 0 & -2 & 32 & -60 \end{bmatrix}. \quad \text{(A.21)}$$

$$W_2^{(\mathrm{R})} = \begin{bmatrix} -60 & 32 & -2 & 0 & 1 & 0 \\ 32 & -60 & 32 & -2 & -6 & 0 \\ -2 & 32 & -60 & 32 & 13 & 0 \\ 0 & -2 & 32 & -60 & 12 & 0 \\ 1 & -6 & 13 & 12 & -30 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}. \quad \text{(A.22)}$$

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 (*J. Comput. Phys.,* to appear).
2. P. Olson, *Summation by Parts, Projections and Stability,* RIACS Technical Report 93.04 (June 1993).
3. B. Strad, Summation by parts for finite difference approximations for d/dx (Department of Scientific Computing, Upsala University, Upsala, Sweden, August 1991).
4. B. Gustafsson, H. O. Kreiss, and A. Sundström, Stability theory of difference approximations for mixed initial boundary value problems. II, *Math. Comput.* **26,** 649 (1972).
5. S. Abarbanel and A. Ditkowski, *Multi-dimensional Asymptotically Stable 4th-Order Accurate Schemes for the Diffusion Equation,* ICASE Report No. 96-8, February 1996.