MATHEMATICS OF COMPUTATION, VOLUME 29, NUMBER 132
OCTOBER 1975, PAGES 985–994

Higher Order Compact Implicit Schemes for the Wave Equation*

By Melvyn Ciment and Stephen H. Leventhal**

Abstract. Higher order finite-difference techniques have associated large star systems which engender complications near the boundary. In the numerical solution of hyperbolic equations, such boundary conditions require careful treatment since errors or instabilities generated there will in general pollute the entire calculation. To circumvent this difficulty, we use a class of implicit schemes suggested by H.-O. Kreiss, which achieves the highest order of accuracy possible on the smallest (most compact) mesh system. Here we develop a scheme which approximates the wave equation,

$$U_{tt} = a(x, y, t)U_{xx} + b(x, y, t)U_{yy},$$

with fourth order accuracy in space and time. After an appropriate factorization, the resulting set of equations are tridiagonal and hence easily solved. The tridiagonal nature also indicates that the boundary conditions do not create special difficulties. Numerical experiments demonstrate the expected order of convergence and fulfill our expectations on the treatment of boundary conditions. An experimental computation also demonstrates that our results hold on L-shaped domains.

I. Introduction. The design of higher order accurate finite-difference methods for hyperbolic equations is complicated by the need to provide stable schemes which are efficient with respect to operation count and which do not experience difficulties near the boundaries. This last point is especially important to bear in mind when one employs higher order schemes, which characteristically have a large number of mesh points in the associated star of each point. For the second order wave equation, most of the work to date has been involved with second order methods as these will not require fictitious boundary points. Second order methods include the classical second order explicit method [2], and the implicit schemes of von Neumann and Lees [3]. These implicit methods were developed because of their favorable stability condition. Lees [3] devised an alternating direction (ADI) type modification of von Neumann's method which allows one to solve multi-dimensional problems by solving tridiagonal equations.

In this paper, we describe a fourth order implicit scheme. Aside from the higher order accuracy, the most important features of our method is the fact that we solve only tridiagonal equations and that fictitious points are not needed at each time step along the boundary. Our scheme is a modification of a class of implicit difference

Received December 2, 1974.

AMS (MOS) subject classifications (1970). Primary 35L05, 65M05, 65M10; Secondary 65N10.

Key words and phrases. Wave equation, hyperbolic equations, implicit difference scheme, alternating direction method, L-shaped domain, higher order difference method.

*This research was supported by the Naval Surface Weapons Center Independent Research Fund.

**This research was performed while the author was a National Research Council Postdoctoral Fellow.

approximations suggested by Kreiss which achieve highest order of accuracy possible on the smallest (most compact) mesh system. Here we treat the wave equation

$$U_{tt} = a(x, y, t)U_{xx} + b(x, y, t)U_{yy}$$

with fourth order accuracy in space and time on a star which yields tridiagonal equations. Our method closely resembles the work of McKee [4] who claimed to have achieved a fourth order space and time ADI type scheme for the above equation. However, the published version of McKee [4] contains errors which make his equations only second order for time dependent coefficients. McKee has personally communicated his corrected equations to us and plans to publish an erratum. Before his corrected equations were known to us, we had already provided a fourth order scheme. Our derivation in the time-dependent case resulted in an algorithm somewhat more complicated than McKee's. However, we are able to achieve an ADI factorization and arrange the remaining computations in such a way that ultimately one only needs 13 operations (multiply or divide) per grid point in solving the implicit equations at each time step. Moreover, McKee reports that his method is unstable for time-dependent coefficients, whereas our method appears to be stable in all cases tested and numb results are presented to bear this out.

Our scheme is an ADI type factorization which requires no commutation and thus should not be expected to suffer from the defects associated with some ADI schemes which are limited to rectangular regions as in the elliptic case [6]. We present a computation on an L-shaped domain which shows that our method can be extended to more general domains and retain fourth order convergence.

II. Derivation of the Algorithm. A fourth order finite-difference approximation to a second derivative U_{xx} can be obtained by using the five-point explicit difference formula

$$\left(I - \frac{\delta_x^2}{12}\right)\frac{\delta_x^2}{h^2} U_j = U_{xx} + O(h^4), \tag{2.1}$$

where $\delta_x^2 U_j \equiv U_{j+1} - 2U_j + U_{j-1}$ represents the usual second central difference and h is the mesh size in the x direction.

Consider approximating the wave equation $U_{tt} = U_{xx}$, using a fourth order explicit scheme in its most obvious form. This would involve differencing over five time levels and five spatial points; namely, let U_j^n denote an approximation to $U(x_j, t^n)$, then

$$\left(I - \frac{\delta_t^2}{12}\right)\frac{\delta_t^2}{k^2} U_j^n = \left(I - \frac{\delta_x^2}{12}\right)\frac{\delta_x^2}{h^2} U_j^n,$$

where $x_j = jh$ and $t^n = nk$ and where h and k are the spatial and temporal grids, respectively. Aside from the fact that this scheme is unconditionally unstable, there still would be a problem of providing a fictitious point near the boundary. In this paper we follow a suggestion of Kreiss, cited in Orszag [5], as to how to construct higher order compact difference approximations which reduces the number of fictitious points needed near the boundary. They note that by using a Neumann series expansion that

$$Q_x^{-1} \equiv \left(I + \frac{\delta_x^2}{12}\right)^{-1} = \left(I - \frac{\delta_x^2}{12}\right) + O(h^4). \tag{2.2}$$

Substituting (2.2) into (2.1) we obtain

$$\left(I + \frac{\delta_x^2}{12}\right)^{-1} \frac{\delta_x^2}{h^2} U_j = U_{xx} + O(h^4). \tag{2.3}$$

The approximation to U_{xx} is obtained by solving

$$\left(I + \frac{\delta_x^2}{12}\right) U_{xx} = \frac{\delta_x^2}{h^2} U_j + O(h^4). \tag{2.4}$$

Since (2.4) is a tridiagonal system the problem of fictitious points for U_j does not arise. However, one must have boundary values for U_{xx}. How this is handled depends on the analytical circumstances of the problem and will be discussed for our specific case in Section IV.

Let us now formally apply the above implicit difference approximations to the derivatives occurring in the following initial boundary value problem for the two-dimensional wave equation

$$U_{tt} = a(x, y, t)U_{xx} + b(x, y, t)U_{yy} \quad \text{in } \Omega,$$

$$\left.\begin{aligned} U(x, y, 0) &= f(x, y) \\ U_t(x, y, 0) &= g(x, y) \end{aligned}\right\}, \quad (x, y) \in \Omega, \tag{2.5}$$

$$U(x, y, t) = h(x, y, t), \qquad (x, y) \in \partial\Omega,$$

where $a(x, y, t) > 0, b(x, y, t) > 0, \Omega$ is a rectangle in R^2 and $\partial\Omega$ is its boundary. We divide $\Omega \times [0, T]$ into a rectangular mesh with mesh spacings h, k in space and time, respectively. Direct application of (2.3) yields

$$Q_t^{-1} \frac{\delta_t^2}{k^2} U_{j,m}^n = a_{j,m}^n Q_x^{-1} \frac{\delta_x^2}{h^2} U_{j,m}^n + b_{j,m}^n Q_y^{-1} \frac{\delta_y^2}{h^2} U_{j,m}^n, \tag{2.6}$$

where the j, m, and n indices represent displacements in x, y, and t directions, respectively, and Q_x, Q_y, and Q_t are defined as in (2.2).

The approximation represented by (2.6) has truncation error $O(h^4 + k^4)$. However, (2.6) as it stands, seems to suggest that a large number of operations are needed to solve the implicit equations. Clearly, if one could factor the difference operators into the separate spatial variables, then it would only be necessary to solve tridiagonal equations. Here we provide a simple stable ADI type factorization (of a modified equation (2.6)) which remains tridiagonally implicit in its one-dimensional factors. It appears that McKee's corrected equations result in a factored explicit scheme which, unfortunately, is unstable for time-dependent coefficients [4]. Below, we give a simple factorization which is correct for time-dependent coefficients. Modify (2.6) in the following manner:

$$
\begin{aligned}
Q_t^{-1}\frac{\delta_t^2}{k^2}U_{j,m}^n &= a_{j,m}^n Q_x^{-1}\frac{\delta_x^2}{h^2}U_{j,m}^n + b_{j,m}^n Q_y^{-1}\frac{\delta_y^2}{h^2}U_{j,m}^n \\
&\quad -\frac{k^4}{144}Q_t^{-1}\frac{\delta_t^2}{k^2}\left[a_{j,m}^n Q_x^{-1}\frac{\delta_x^2}{h^2}b_{j,m}^n Q_y^{-1}\frac{\delta_y^2}{h^2}\right]U_{j,m}^n.
\end{aligned} \tag{2.7}
$$

The term added on in (2.7) is an $O(k^4)$ term; therefore, the accuracy of the method is unchanged.

Multiplying (2.7) by Q_t we obtain

$$
\begin{aligned}
&\delta_t^2\left[I-\frac{\lambda^2}{12}a_{j,m}^n Q_x^{-1}\delta_x^2\right]\left[I-\frac{\lambda^2}{12}b_{j,m}^n Q_y^{-1}\delta_y^2\right]U_{j,m}^n \\
&\qquad = \lambda^2\left[a_{j,m}^n Q_x^{-1}\delta_x^2 + b_{j,m}^n Q_y^{-1}\delta_y^2\right]U_{j,m}^n
\end{aligned} \tag{2.8}
$$

or

$$
\begin{aligned}
&\left[I-\frac{\lambda^2}{12}a_{j,m}^{n+1}Q_x^{-1}\delta_x^2\right]\left[I-\frac{\lambda^2}{12}b_{j,m}^{n+1}Q_y^{-1}\delta_y^2\right]U_{j,m}^{n+1} \\
&\qquad = 2\left[I-\frac{\lambda^2}{12}a_{j,m}^n Q_x^{-1}\delta_x^2\right]\left[I-\frac{\lambda^2}{12}b_{j,m}^n Q_y^{-1}\delta_y^2\right]U_{j,m}^n \\
&\qquad\quad -\left[I-\frac{\lambda^2}{12}a_{j,m}^{n-1}Q_x^{-1}\delta_x^2\right]\left[I-\frac{\lambda^2}{12}b_{j,m}^{n-1}Q_y^{-1}\delta_y^2\right]U_{j,m}^{n-1} \\
&\qquad\quad +\lambda^2\left[a_{j,m}^n Q_x^{-1}\delta_x^2 + b_{j,m}^n Q_y^{-1}\delta_y^2\right]U_{j,m}^n,
\end{aligned} \tag{2.9}
$$

where $\lambda = k/h$.

The numerical implementation of (2.9) is discussed in Section IV. However, let us note the following:

1. The left-hand side of (2.9) is a factorization into x and y differences which allows us to solve (2.9) by sweeping first in the x and then in the y directions. It will be seen that these sweeps only require the solution of tridiagonal systems.

2. The intermediate boundary conditions necessary for the sweeps are easily obtained using the differential equation and the analytic representation of the difference formulas.

3. The right-hand side of (2.9) is computed as a combination of the two previous right-hand sides, the previous intermediate step, and one tridiagonal sweep.

4. The previous remarks are all easily seen to hold for the obvious generalization to three space dimensions.

III. Stability Analysis. For completeness we include the stability analysis of (2.7) (see [1] for case $a = b = 1$) for the case of constant coefficients a and b. That the scheme is stable for the case of nonconstant coefficients will be apparent from the numerical examples in Section V.

To analyze stability, one substitutes a solution of the form $U_{j,m}^n = \rho^n e^{ij\theta} e^{im\phi}$ into the difference equation (2.7). Observing that the implicit scheme is three-level in time, one obtains a quadratic in ρ

$$\rho^2 + \left[\frac{-2(rs + ab\lambda^4) + 10\lambda^2(ar + bs)}{(rs + \lambda^4 ab) + \lambda^2(ar + bs)}\right]\rho + 1 = 0, \tag{3.1}$$

where $r = (5 + \cos\phi)/(1 - \cos\phi)$, and $s = (5 + \cos\theta)/(1 - \cos\theta)$. It is clear that for the above real polynomial, since the product of the roots is one, that the necessary and sufficient condition for stability is that the discriminant be nonpositive.

This leads to the condition

$$ab\lambda^4 - 2\lambda^2(ar + bs) + rs \geqslant 0, \tag{3.2}$$

on λ and assures that $|\rho| = 1$. The above quadratic inequality on λ^2 holds for all θ and ϕ if λ^2 is less than the smallest λ^2 root obtained by taking equality in (3.2). This implies that

$$\lambda^2 \leqslant \frac{r}{b} + \frac{s}{a} - \sqrt{\left(\frac{r}{b}\right)^2 + \frac{rs}{ab} + \left(\frac{s}{a}\right)^2}. \tag{3.3}$$

If we set $c = \max(a, b)$, then one observes that this last inequality certainly still holds if one replaces a given a or b by c. Since the case when the three are all equal ($a = b = c$) is possible, it is thus necessary and sufficient for stability that

$$\lambda^2 \leqslant \frac{r}{c} + \frac{s}{c} - \frac{1}{c}\sqrt{r^2 + rs + s^2}. \tag{3.4}$$

Now by definition, $2 \leqslant r, s \leqslant \infty$. Inspection reveals that the minimum is assumed at $(2, 2)$. Thus we have for stability

$$c\lambda^2 \leqslant 4 - \sqrt{12} = 2(2 - \sqrt{3})$$

or

$$\sqrt{c}\lambda \leqslant \sqrt{2(2 - \sqrt{3})} = \sqrt{3} - 1. \tag{3.5}$$

IV. Numerical Implementation. The major features to be discussed in the numerical implementation of (2.9) are the splitting of the left-hand side for the x and y sweeps, the computation of the right-hand side, the computation of intermediate boundary conditions, and the initialization of the procedure.

Defining

$$Z_{j,m}^{n+1} = \left[I - \frac{\lambda^2}{12} b_{j,m}^{n+1} Q_y^{-1} \delta_y^2\right] U_{j,m}^{n+1}, \tag{4.1a}$$

then (2.9) becomes

$$\left[I - \frac{\lambda^2}{12} a_{j,m}^{n+1} Q_x^{-1} \delta_x^2\right] Z_{j,m}^{n+1} = G_{j,m}^{n+1}, \tag{4.1b}$$

where $G_{j,m}^{n+1}$ is the right-hand side of (2.9).

Dividing (4.1b) by $a_{j,m}^{n+1}$ and then multiplying by Q_x, the solving of (4.1b) reduces to

$$\left[Q_x \frac{1}{a_{j,m}^{n+1}} - \frac{\lambda^2}{12}\delta_x^2\right] Z_{j,m}^{n+1} = Q_x\left(\frac{G_{j,m}^{n+1}}{a_{j,m}^{n+1}}\right), \tag{4.2a}$$

which only requires the solution of a tridiagonal system for each m.

Similarly, reduce (4.1a) to solving a tridiagonal system for each j by dividing by $b_{j,m}^{n+1}$ and then multiplying by Q_y to obtain

$$\left[Q_y \frac{1}{b_{j,m}^{n+1}} - \frac{\lambda^2}{12}\delta_y^2\right] U_{j,m}^{n+1} = Q_y\left(\frac{Z_{j,m}^{n+1}}{b_{j,m}^{n+1}}\right). \tag{4.2b}$$

Note that the right-hand side of (2.9) is $G_{j,m}^{n+1}$. Thus,

$$\begin{aligned} G_{j,m}^{n+1} &\equiv 2\left[I - \frac{\lambda^2}{12} a_{j,m}^n Q_x^{-1}\delta_x^2\right]\left[I - \frac{\lambda^2}{12} b_{j,m}^n Q_y^{-1}\delta_y^2\right] U_{j,m}^n \\ &\quad -\left[I - \frac{\lambda^2}{12} a_{j,m}^{n-1} Q_x^{-1}\delta_x^2\right]\left[I - \frac{\lambda^2}{12} b_{j,m}^{n-1} Q_y^{-1}\delta_y^2\right] U_{j,m}^{n-1} \\ &\quad + \lambda^2 [a_{j,m}^n Q_x^{-1}\delta_x^2 + b_{j,m}^n Q_y^{-1}\delta_y^2] U_{j,m}^n \\ &= 2G_{j,m}^n - G_{j,m}^{n-1} + \lambda^2 [a_{j,m}^n Q_x^{-1}\delta_x^2 + b_{j,m}^n Q_y^{-1}\delta_y^2] U_{j,m}^n. \end{aligned} \tag{4.3}$$

Hence, the first two terms in $G_{j,m}^{n+1}$ are known from previous time steps and need not be computed. Also known from the previous time step is that

$$W_{j,m}^n \equiv b_{j,m}^n Q_y^{-1}\delta_y^2 U_{j,m}^n = (U_{j,m}^n - Z_{j,m}^n)\frac{12}{\lambda^2}. \tag{4.4}$$

Thus, it is only necessary to compute

$$V_{j,m}^n \equiv Q_x^{-1}\delta_x^2 U_{j,m}^n, \tag{4.5}$$

i.e., we solve the tridiagonal system,

$$Q_x V_{j,m}^n = \delta_x^2 U_{j,m}^n, \tag{4.6}$$

for each m.

From (4.2a), (4.3), (4.4), and (4.6) we see that the intermediate boundary conditions that must be computed are for $Z_{j,m}^{n+1}$ and $V_{j,m}^n$. From (4.1b) and (4.5) it is clear that $Z_{j,m}^{n+1}$ and $V_{j,m}^n$ are respectively approximations to

$$Z(x, y, t) \equiv U(x, y, t) - \frac{k^2}{12} b(x, y, t) U_{yy} + O(k^2 h^4)$$

and

$$V(x, y, t) \equiv h^2 U_{xx}(x, y, t) + O(h^6).$$

To solve for the $Z_{j,m}^n$ and $V_{j,m}^n$ values, it is sufficient to approximate U_{xx} and U_{yy} on the boundary up to fourth order of accuracy.

On y = constant lines U_{xx} may be computed from (4.6); however, U_{yy} may not be computed. Similarly on x = constant lines U_{xx} is not computable and U_{yy} may be computed by solving the tridiagonal system

$$U_{yy} = Q_y^{-1}\frac{\delta_y^2}{h^2} U_{j,m}^n.$$

However, on both these lines U_{tt} may be computed. Therefore, on y = constant lines U_{yy} is obtained from the formula

$$U_{yy} = (U_{tt} - a(x, y, t)U_{xx})/b(x, y, t),$$

and on x = constant lines U_{xx} is obtained from

$$U_{xx} = (U_{tt} - b(x, y, t)U_{yy})/a(x, y, t).$$

The above computation of U_{xx} and U_{yy} on y = constant and x = constant lines requires the value of these functions at the corners. These corner values are obtained by using the fourth order explicit difference formula (2.1) and then a fourth order extrapolation at the two fictitious points required in (2.1). Thus a fourth order approximation at the corner of U_{xx} and U_{yy} requires six points.

As initial conditions for this problem, we are given the value of U and U_t at $t = 0$. From these values there are many ways (e.g. Taylor series) to obtain a fourth order approximation to U at $t = \Delta t$. Therefore, assume that U is known at $t = 0$ and $t = \Delta t$. The problem of initialization is the one of computing the right-hand side $G^2_{j,m}$, i.e.,

$$\begin{aligned} G^2_{j,m} = {} & 2\left[I - \frac{\lambda^2}{12} a^1_{j,m} Q_x^{-1}\delta_x^2\right]\left[I - \frac{\lambda^2}{12} b^1_{j,m} Q_y^{-1}\delta_y^2\right] U^1_{j,m} \\ & - \left[I - \frac{\lambda^2}{12} a^0_{j,m} Q_x^{-1}\delta_x^2\right]\left[I - \frac{\lambda^2}{12} b^0_{j,m} Q_y^{-1}\delta_y^2\right] U^0_{j,m} \\ & + \lambda^2 [a^1_{j,m} Q_x^{-1}\delta_x^2 + b^1_{j,m} Q_y^{-1}\delta_y^2] U^1_{j,m}. \end{aligned} \tag{4.7}$$

It is necessary to compute the last term of (4.7) on every time step, so the procedure for doing this is the same as described above in (4.4)–(4.6). The first and second terms, $G^1_{j,m}$ and $G^0_{j,m}$, are computed in the same fashion, thus it is only necessary to discuss the first term $G^1_{j,m}$. This term is computed in the opposite way to (4.1), i.e., $U^1_{j,m}$ is known and we want to compute

$$G^1_{j,m} = \left[I - \frac{\lambda^2}{12} a^1_{j,m} Q_x^{-1}\delta_x^2\right]\left[I - \frac{\lambda^2}{12} b^1_{j,m} Q_y^{-1}\delta_y^2\right] U^1_{j,m}. \tag{4.8}$$

Noting the definition of $W^1_{j,m}$ and $V^1_{j,m}$, (4.4) and (4.5), we see that by computing these terms separately that

$$G^1_{j,m} = U^1_{j,m} - \frac{\lambda^2}{12} a^1_{j,m} V^1_{j,m} - \frac{\lambda^2}{12} W^1_{j,m} + \frac{\lambda^4}{144} a^1_{j,m} Q_x^{-1}\delta_x^2 W^1_{j,m}. \tag{4.9}$$

Thus we only have to compute $(\lambda^4/144)a^1_{j,m} Q_x^{-1}\delta_x^2 W^1_{j,m}$, which requires the solution of a tridiagonal system. The boundary conditions for this system are obtained by appealing to the corresponding analytical expression as was done previously for $Z^n_{j,m}$ and $V^n_{j,m}$ above. For consistency one should use fourth order accurate boundary conditions; however, for convenience on the initial step, only second order accurate methods were used. This did not change the order of accuracy of the method in the calculations we performed.

Let us perform an operation (multiply and divide) count per time step for the

solving of (2.9). Let J and M be the number of x and y unknown grid points, respectively. For the computation of $V^n_{j,m}$ there are M backsolves, each of which requires (see [2]) $3J - 2$ operations. Solving (4.2a) requires M decompositions and backsolves each of which takes $5J - 4$ operations. Similarly, (4.2b) requires $J(5M - 4)$ operations. Finally, the computation of the boundary conditions requires 2 backsolves with $3J - 2$ operations and 2 backsolves with $3M - 2$ operations. Thus, the total number of operations is $13J \cdot M + 2J - 8$.

V. Numerical Examples. In this section we present three examples demonstrating the fourth order accuracy, the effectiveness, and the stability of the method. The first example was also presented by McKee [4]. We compare our results to his. However, we are unable to explain the different stability characteristics of the two methods. In the second example the solution grows exponentially with time; however, we shall see that the fourth order accuracy and the stability of the method are retained. Finally, the second example is repeated on an L-shaped domain.

Example 5.1. Let Ω be defined by $[\frac{1}{2} \leqslant x, y \leqslant 1]$ and let us define the coefficients, initial conditions, and boundary conditions of (2.5) by

$$\begin{aligned} a(x, y) &= \tfrac{1}{2}(1 - \sin x \sin y), \\ b(x, y) &= \cot^2 x - \cot x \sin y \cos x + \sin x \sin y, \\ f(x, y) &= 0, \\ g(x, y) &= \sin^2 x \sin y, \\ h(x, y, t) &= \sin^2 x \sin y \sin t \quad \text{on } \partial\Omega. \end{aligned}$$

The exact solution is

$$u(x, y, t) = \sin^2 x \sin y \sin t.$$

The first experiment shows the $O(h^4 + k^4)$ accuracy of the method. First the problem was solved with $k = h/2 = .05$ then with $k = h/2 = .025$. The fourth order accuracy predicts a factor sixteen decrease in the error. The results of this experiment showing the fourth order convergence are in Table 5.1.

# TIME STEPS	h	k	L_2-ERROR	L_2-RATE	MAX-ERROR	MAX-RATE
15	.1	.05	2.361129-09	4.1184	1.25968-08	4.2207
30	.05	.025	1.359418-10		6.75579-10	
20	.1	.05	4.610927-09	4.6238	1.92659-08	4.4534
40	.05	.025	1.870124-10		8.7666-10	
30	.1	.05	4.496657-09	4.3577	2.3732-08	4.5980
60	.05	.025	2.193196-10		9.8161-10	

TABLE 5.1

In verifying the accuracy of (2.9) as compared to McKee's method, we see that for $\lambda = .5$ and 200 time steps (i.e., $h = .05$, $k = .025$, $T = 5$) McKee had an absolute maximum relative error of 2×10^{-8} while (2.9) had an absolute maximum relative error of 3.4×10^{-9}.

McKee warns the user that his method has a possible weak instability and an exponential instability when the coefficients depend on time. McKee does not indicate in what type of situations this occurs. For the example which we considered below, (2.9) does not seem to possess time-dependent instabilities.

Example 5.2. Let Ω be defined as above and let us define

$$a(x, y, t) = \frac{x^2}{2(t+5)^2}, \qquad b(x, y, t) = \frac{y^2}{2(t+5)^2},$$

$$f(x, y) = e^{5xy}, \qquad g(x, y) = xye^{5xy}, \qquad h(x, y, t) = e^{xy(t+5)} \quad \text{on } \partial\Omega.$$

The exact solution to (2.5) with the above conditions is $U(x, y, t) = e^{xy(t+5)}$.

Similar experiments to those that were run in Example 5.1 were run for this problem to study the accuracy and stability. The results are in Table 5.2. However, due to the size of the solution, we are concerned with maximum relative error.

# TIME STEPS	h	k	L_2-ERROR	L_2-RATE	RELATIVE MAX-ERROR	RELATIVE MAX-RATE
10	.1	.06	5.904951-04		6.13764-05	
				5.0130		5.3355
20	.05	.03	1.828673-05		1.5200-06	
				4.0087		3.6767
40	.025	.015	1.136046-06		1.1886-07	

TABLE 5.2

Example 5.3. ADI type methods that require commutation of difference operators have been limited to rectangular domains [6]. However, no commutation of operators occurred in the derivation of (2.9), hence (2.9) is still fourth order on nonrectangular domains. In order to test our method in a region where the solution has large truncation error near the reentrant corner, we define Ω as the L-shaped domain derived by taking the rectangle $[1. \leqslant x, y \leqslant 1.5]$ and removing the rectangle $[1.25 < x, y < 1.5]$. Let us solve for the same solution as in Example 5.2. Table 5.3 shows that the fourth order accuracy still holds on the L-shaped region.

# TIME STEPS	h	k	L_2-ERROR	L_2-RATE	RELATIVE MAX-ERROR	RELATIVE MAX-RATE
100	.05	.03	1.37503+01		1.0267-03	
				4.49		6.49
200	.025	.015	6.0967-01		1.1451-05	
				3.92		4.35
400	.0125	.0075	4.0108-02		5.5963-07	

TABLE 5.3

VI. Conclusion and Remarks. The type of factorization given in (2.9) may be achieved also when the equation (2.5) contains lower order terms, i.e.,

$$U_{tt} = a(x, y, t)U_{xx} + b(x, y, t)U_{yy} + c(x, y, t)U_x + d(x, y, t)U_y + e(x, y, t)U. \tag{6.1}$$

As in Section II for second derivatives it is easy to see that the difference operator

$$\left(I + \frac{\delta_x^2}{6}\right)^{-1} \frac{\delta_0^x}{2h} U_j,$$

is a fourth order approximation to U_x, where $\delta_0^x U_j \equiv (U_{j+1} - U_{j-1})/2$ is the first central difference. Denoting $(I + \delta_x^2/6)$ and $(I + \delta_y^2/6)$ by R_x and R_y, respectively, one obtains the following fourth order in time and space difference approximation to (6.1).

$$\begin{aligned} &\delta_t^2 \Gamma_{j,m}^n (I - \lambda^2 \tilde{a}_{j,m}^n Q_x^{-1} \delta_x^2)(I - \lambda^2 h \tilde{c}_{j,m}^n R_x^{-1} \delta_0^x) \\ &\quad \cdot (I - \lambda^2 \tilde{b}_{j,m}^n Q_y^{-1} \delta_y^2)(I - \lambda^2 h \tilde{d}_{j,m}^n R_y^{-1} \delta_0^y) U_{j,m}^n \\ &= \lambda^2 (\tilde{a}_{j,m}^n Q_x^{-1} \delta_x^2 + \tilde{b}_{j,m}^n Q_y^{-1} \delta_y^2 + h \tilde{c}_{j,m}^n R_x^{-1} \delta_0^x + h \tilde{d}_{j,m}^n R_y^{-1} \delta_0^y) U_{j,m}^n, \end{aligned} \tag{6.2}$$

$$\text{where } \Gamma_{j,m}^n = \left(1 - \frac{k^2}{12} e_{j,m}^n\right) \text{ and where } \tilde{\text{symbol}} = \frac{1}{12\Gamma} \cdot \text{symbol}.$$

Details on intermediate boundary conditions still must be looked into for (6.2). In general, (2.9) represents a highly accurate stable method for the solution of second order wave equations. Extension of these ideas to equations with lower order terms are now in progress.

Acknowledgment. The authors are happy to take this opportunity to thank John Bell for his energetic help in getting the above numerical experiments completed during his student traineeship at Naval Surface Weapons Center.

Naval Surface Weapons Center

1. G. FAIRWEATHER & A. R. MITCHELL, "A high accuracy alternating direction method for the wave equation," *J. Inst. Math. Appl.*, v. 1, 1965, pp. 309–316. MR 35 #1224.

2. E. ISAACSON & H. B. KELLER, *Analysis of Numerical Methods*, Wiley, New York, 1966. MR 34 #924.

3. M. LEES, "Alternating direction methods for hyperbolic differential equations," *J. Soc. Indust. Appl. Math.*, v. 10, 1962, pp. 610–616. MR 30 #2694.

4. S. McKEE, "High accuracy A.D.I. methods for hyperbolic equations with variable coefficients," *J. Inst. Math. Appl.*, v. 11, 1973, pp. 105–109.

5. S. A. ORSZAG & M. ISRAELI, "Numerical simulation of viscous incompressible flows," *Annual Review of Fluid Mechanics*. Vol. VI, M. Van Dyke (Editor), Annual Reviews, 1974, pp. 281–318.

6. D. M. YOUNG, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971. MR 46 #4698.