SIAM J. APPL. MATH.
Vol. 29, No. 2, September 1975

DIFFERENCE SCHEMES WITH FOURTH ORDER ACCURACY FOR HYPERBOLIC EQUATIONS*

S. ABARBANEL,† D. GOTTLIEB‡ AND E. TURKEL†

Abstract. Two explicit finite difference schemes of fourth order accuracy (in space and time) are presented for the numerical solution of quasi-linear divergence free one-dimensional hyperbolic systems. Both of these schemes are four-step methods, one being a two level scheme, the other using three levels. These algorithms are compared in numerical examples with both second order schemes and with the Kreiss–Oliger method which is fourth order in space and second order in time. The results show that it is most advantageous to use the true fourth order schemes.

1. Introduction. Among the first attempts at solving fluid dynamic problems numerically was the pioneering work of Von Neumann and Richtmyer [16] during and after the second world war. They used a first order Lagrangian scheme and introduced an artificial viscosity to handle shocks. In 1954 Lax [6] showed that by expressing the finite difference equations in divergence form one no longer needed to add an external viscosity factor in order to get the correct shock relations. This was then used to construct a first order Eulerian code. Later, Lax and Wendroff [8] constructed a divergence free second order scheme. Richtmyer [10], [11] then introduced a similar second order algorithm that required two steps per time level but no longer required matrix multiplication and so was much faster.

Multidimensional Eulerian methods do not have the distortion problems associated with the Lagrangian schemes. This together with the accompanying relatively high accuracy led to the general acceptance of multistep methods of second order for solution of problems containing discontinuities. In meteorological problems leap-frog type methods were utilized because of the high dissipation inherent in Lax–Wendroff type schemes.

With the advent of third generation computers more involved problems were tackled, and it was realized that for many problems even higher order methods would be more efficient. Thus, Rusanov [13] as well as Burstein and Mirin [4] introduced a three-step third order generalization of the Richtmyer method. At about the same time, Roberts and Weiss [12] as well as Kreiss and Oliger [5] generalized the leap-frog method and considered schemes that were second order in time but fourth order in space while still being nondissipative. These are applicable to long range meteorological problems where shocks do not appear, but there is a delicate energy balance that needs to be preserved to achieve meaningful results.

* Received by the editors August 8, 1974. This work was supported in part by the Air Force Office of Scientific Research (NAM) through the European Office of Aerospace Research, AFSC, United States Air Force, under Grant AFOSR-72-2370.

† Department of Mathematical Sciences, Tel-Aviv University, Tel-Aviv, Israel.

‡ Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139.

For ordinary differential equations, it has long been recognized that numerical methods of even order have several advantages over methods of odd order. Similar remarks hold for finite difference approximations to hyperbolic systems. When one increases the order from an even order of accuracy to one higher order, the minimal number of points needed at the previous time level increases. However, when one goes from an odd order scheme to the next higher even order, the minimum domain of dependence remains the same. Thus, one can increase the accuracy without increasing the basic complexity of the method. Furthermore, odd order methods require a stabilizing term that is independent of $\Delta t/\Delta x$ for the scheme to be stable (see, e.g., Strang [14]). In nonlinear problems, global stability requirements lead to time steps that are much smaller than those necessary for local stability. In these cases, the odd order schemes will introduce an unnecessarily high dissipation.

Hence, as with Runge–Kutta methods, it is natural to consider fourth order methods. Since in hyperbolic systems the spatial and temporal coordinates are on equal footing, it is desirable to construct a scheme which is fourth order in both space and time. Such a scheme has been presented by Abarbanel and Gottlieb [1]. Here we propose to achieve the same goal by a simpler and more efficient algorithm. One of the major applications of these techniques is to continuous mechanics and so these schemes must also be fourth order for nonlinear equations. Hence, it is not sufficient to verify that the amplification matrix is fourth order for long wave lengths as this only accounts for linear accuracy. For example, the scheme of Balakin [2] is fourth order only for linear problems.

2. Numerical background. In this paper, we shall confine ourselves to the one-dimensional vector equation

$$w_t = f_x, \qquad f = f(w) \tag{1}$$

or

$$w_t = Aw_x. \tag{1'}$$

The introduction of an inhomogenous forcing terming $g(w, x)$ introduces no basic difficulties. In order to simplify the formulas, we assume that f and g do not depend explicitly on t. On practical grounds, this requirement is almost always satisfied. To simplify our notation, we introduce the averaging and differencing operations

$$\mu w_j = \frac{1}{2}(w_{j+1/2} + w_{j-1/2}), \qquad \delta w_j = w_{j+1/2} - w_{j-1/2} \tag{2}$$

and define λ as

$$\lambda = \frac{\Delta t}{\Delta x}, \tag{3}$$

where as usual Δt and Δx are the mesh spacings in time and x, respectively.

We first attempt a straightforward generalization of the Richtmyer two-step second order scheme. Thus, we assume that we have the solution at time $t = n\Delta t$, and we wish the solution at $t = (n + 1)\Delta t$.

Let

$$
\begin{aligned}
w^{(1)} &= \mu w^n + \lambda a\, \delta f^n, \\
w^{(2)} &= \varepsilon w^n + (1 - \varepsilon)\mu^2 w^n + \lambda[\alpha_1\, \delta f^{(1)} + \alpha_2\, \delta\mu f^n], \\
w^{(3)} &= \eta\mu w^n + (1 - \eta)\mu^3 w^n + \lambda[\beta_1\, \delta f^{(2)} + \beta_2\, \delta\mu f^{(1)} + \beta_3\, \delta f^n + \beta_4\, \delta\mu^2 f^n], \\
w^{n+1} = w^{(4)} &= w^n + \lambda[\gamma_1\, \delta f^{(3)} + \gamma_2\, \delta\mu f^{(2)} + \gamma_3\, \delta f^{(1)} + \gamma_4\, \delta\mu^2 f^{(1)} + \gamma_5 \delta\mu f^n \\
&\qquad + \gamma_6\, \delta\mu^3 f^n],
\end{aligned} \tag{4}
$$

where $f^{(k)} = f(w^{(k)}), f^n = f(w(n\,\Delta t, x))$. $a, \varepsilon, \eta, \alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3, \beta_4, \gamma_1, \cdots, \gamma_6$ are fifteen free parameters that we wish to determine and to utilize to get fourth order accuracy.

Experience with second and third order schemes indicates that we would like $w^{(1)}$ to be first order, $w^{(2)}$ second order, $w^{(3)}$ third order while $w^{(4)}$ is fourth order. With this assumption

$$
\varepsilon = 1, \quad \eta = \tfrac{3}{2}, \quad \beta_2 = \gamma_2 = \gamma_3 = \gamma_4 = 0, \tag{4a}
$$

and so we are left with nine parameters. In addition we have the fractional time steps $\theta_1, \theta_2, \theta_3$ at which intermediate approximations are given, i.e.,

$$
w^{(k)} \simeq w(t + \theta_k\, \Delta t) \quad \text{with } \theta_4 = 1.
$$

One then uses a Taylor series expansion for these intermediate steps and require $w^{(k)}$ to be a kth order approximation to $w(t + \theta_k\, \Delta t)$. The first two steps yield a, α_1, α_2 as functions of θ_1 and θ_2 while the third step yields $\beta_1, \beta_3, \beta_4$ and θ_3 as functions of θ_1 and θ_2.

We shall now carry out the fourth step in more detail.

$$
\begin{aligned}
w^{(4)} &= w + \lambda\{\gamma_1\, \delta f^{(3)} + \gamma_5\, \delta\mu f^n + \gamma_6\, \delta\mu^3 f^n\} \\
&= w + \lambda\Big\{\gamma_1\, \delta\Big[f + \theta_3\, \Delta t f_t + \frac{(\theta_3\, \Delta t)^2}{2} f_{tt} + \frac{(\theta_3\, \Delta t)^3}{6} f_{ttt}\Big] \\
&\qquad + \gamma_5\, \delta\mu f^n + \gamma_6\, \delta\mu^3 f^n\Big\} + O((\Delta t)^5) \\
&= w + \frac{\Delta t}{\Delta x}\Big\{\gamma_1\Big[(\Delta x) f_x + \frac{(\Delta x)^3}{24} f_{xxx} + \theta_3(\Delta t)(\Delta x) f_{xt} \\
&\qquad + \frac{\theta_3(\Delta t)(\Delta x)^3}{24} f_{xxxt} + \frac{\theta_3^2(\Delta t)^2(\Delta x)}{2} f_{xttt} \\
&\qquad + \frac{\theta_3^3(\Delta t)^3(\Delta x)}{6} f_{xttt}\Big] + \gamma_5\Big[(\Delta x) f_x + \frac{(\Delta x)^3}{6} f_{xxx}\Big] \\
&\qquad + \gamma_6\Big[(\Delta x) f_x + 5\frac{(\Delta x)^3}{12} f_{xxx}\Big]\Big\} + O((\Delta t)^5 + (\Delta x)^5),
\end{aligned} \tag{5}
$$

but $f_x = w_t$ so

$$w^{(4)} = w + (\gamma_1 + \gamma_5 + \gamma_6)\,\Delta t w_t + \gamma_1\theta_3(\Delta t)^2 w_{tt} + \gamma_1\frac{\theta_3^2}{2}(\Delta t)^3 w_{ttt}$$

$$+ \frac{\gamma_1\theta_3^3(\Delta t)^4}{6} w_{tttt} + \frac{(\gamma_1 + 4\gamma_5 + (10\gamma_6))(\Delta t)(\Delta x)^2}{24} w_{txx}$$

$$+ \frac{\gamma_1\theta_3(\Delta t)^2(\Delta x)^2}{24} w_{ttxx} + O((\Delta t)^5 + (\Delta x)^5).$$

In order for $w^{n+1} = w^{(4)} = w(t + \Delta t) + O((\Delta t)^5)$, we require

$$\gamma_1 + \gamma_5 + \gamma_6 = 1, \qquad \gamma_1\theta_3 = 1/2,$$

$$\gamma_1\frac{\theta_3^2}{2} = 1/6, \qquad \gamma_1\frac{\theta_3^3}{6} = 1/24, \tag{6}$$

$$\gamma_1 + 4\gamma_5 + 10\gamma_6 = 0, \qquad \gamma_1\theta_3 = 0.$$

One can satisfy the last equation by adding an additional term $\gamma_7\,\delta(\mu^2 - 1)f^{(1)}$ to the right-hand side of $w^{(4)}$. Even with this, we have five equations for $\gamma_1, \gamma_5, \gamma_6, \theta_3$, and it can be readily shown that there are no solutions to these equations. Thus, one can not construct a four-step fourth order method where the kth intermediate step has order k.

3. Multilevel formulas. One possible alternative is to construct another $\bar{w}^{(3)}$ at some other intermediate time $\bar{\theta}_3$. Then, one can compute $w^{(4)}$ to fourth order as a function of $w^{(3)}$ and $\bar{w}^{(3)}$, plus terms of lower accuracy. This is basically the approach taken in Abarbanel and Gottlieb [1]. The disadvantage is that one now requires nine steps and so the computational time increases and the method becomes less competitive. Alternatively, one can introduce $\bar{w}^{(2)}$ at the time $w(t - \theta_2\,\Delta t)$. Then, one can use $f^{(2)}$ and $\bar{f}^{(2)}$ to add a term $f_{xtt} = w_{ttt}$. The equations can now be solved and we require only six steps; however, this is still not good enough.

As another alternative, one can relax the requirement that we wish a two level formula and use a three level formula; i.e., we shall use information at time $t - \Delta t$ as well as at time t. Once one is willing to do this, we immediately have another $\bar{w}^{(3)}$ at our disposal. Instead of computing a new $\bar{w}^{(3)}$ as done by Abarbanel and Gottlieb, we use the $w^{(3)}$ computed at the previous time step.

$$\bar{w}^{(3)} = w^{(3)}(t - \Delta t) = w(t - \Delta t + \theta_3\,\Delta t) + O((\Delta t)^4)$$

$$= w(t + (\theta_3 - 1)\,\Delta t) + O((\Delta t)^4) = w(t + \bar{\theta}_3\,\Delta t) + O((\Delta t)^4)$$

with $\bar{\theta}_3 = \theta_3 - 1 < 0$.

We are thus lead to consider the following scheme:

$$
\begin{aligned}
w^{(1)} &= \mu w^n + \lambda a\,\delta f^n,\\
w^{(2)} &= w^n + \lambda[\alpha_1\,\delta f^{(1)} + \alpha_2\,\delta\mu f^n],\\
w^{(3)} &= \left(\frac{3\mu}{2} - \frac{\mu^3}{2}\right)w^n + \lambda[\beta_1\,\delta f^{(2)} + \beta_2\,\delta f^n + \beta_3\,\delta\mu^2 f^n],\\
w^{n+1} &= w + \lambda[\gamma_1\,\delta f^{(3)} + \bar\gamma_1\,\delta\bar f^{(3)} + \gamma_2\,\delta(\mu^2 - 1)f^{(1)} + \gamma_3\,\delta\mu f^{(n)} + \gamma_4\,\delta\mu^3 f^{(n)}],
\end{aligned}
\tag{7}
$$

where $f^{(3)} = f(w^{(3)}(t))$, $\bar f^{(3)} = f(w^{(3)}(t - \Delta t))$.

In nonlinear problems, one frequently does not use a constant time step but allows it to vary from time level to time level. We denote the time difference at the nth step by

$$(\Delta t)_n = t_{n+1} - t_n.$$

Then, in order for (7) to be fourth order, we can choose θ_1 arbitrarily. We then choose

$$(\theta_3)_n = \frac{4z - 3}{6z - 4},$$

where

$$z = \frac{(\theta_3)_{n-1}(\Delta t)_{n-1}}{(\Delta t)_n}$$

and

$$(\theta_2)_n = \tfrac{2}{3}(\theta_3)_n.$$

In the rest of these formulas we drop the subscript n. It is to be understood though that the coefficients in formula (7) change at each time step if $(\Delta t)_n \neq (\Delta t)_{n-1}$. However, since these coefficients are calculated only once at each time level the additional computational time used is small. We let

$$\bar\theta_3 = (\theta_3)_{n-1}, \quad \theta_3 = (\theta_3)_n, \quad \theta_2 = (\theta_2)_n$$

and

$$
\begin{aligned}
\gamma_1 &= -\frac{(\bar\theta_3/2) - 1/3}{\theta_3(\theta_3 - \bar\theta_3)}, \qquad \bar\gamma_1 = \frac{(\theta_3/2) - (1/3)}{\bar\theta_3(\theta_3 - \bar\theta_3)},\\
\gamma_2 &= -\frac{1}{12\theta_1}, \qquad \gamma_3 = \frac{1}{6}\left(10 - \frac{1}{2\theta_1} - 9(\gamma_1 + \bar\gamma_1)\right),\\
\gamma_4 &= -\frac{1}{6}\left(4 - \frac{1}{2\theta_1} - 3(\gamma_1 + \bar\gamma_1)\right),\\
\beta_1 &= \frac{3\theta_3}{4}, \quad \beta_2 = \frac{5\theta_3}{12}, \quad \beta_3 = -\frac{\theta_3}{6},\\
\alpha_1 &= \frac{\theta_2^2}{2\theta_1}, \quad \alpha_2 = \theta_2 - \alpha_1, \quad a = \theta_1.
\end{aligned}
\tag{8}
$$

With these values assigned to the parameters the algorithm (7) is fourth order accurate in both space and time.

As with all multilevel formulas, one must use a separate procedure to compute the first time level.

The stability criterion for multilevel formulas are more difficult to calculate then for two level schemes. Even for a scalar equation, the amplification factor is a matrix, while for a vector equation each component of the amplification matrix G is itself a submatrix. We have not been able to obtain analytically any stability criterion.

Computationally, one finds that the correct condition is

$$\frac{\Delta t}{\Delta x}\rho(A) \leqq 1 \tag{9}$$

where $\rho(A)$ is the spectral radius of $A = \partial f/\partial u$ (see (1′)).

4. Four-step method. A different approach to circumvent the difficulty of constructing a four-step scheme is to insist on a two level formula but to relax the requirement that each intermediate step $w^{(k)}$ have an accuracy of order k. Thus, we only require that $w^{(1)}, w^{(2)}, w^{(3)}$ be of first order while $w^{(4)}$ is to be of fourth order. To find the coefficients in (4), we can no longer do the straight forward Taylor series expansion of $w^{(k)}$ up to order k. Instead we expand $w^{(1)}, w^{(2)}, w^{(3)}$ carrying all terms through fourth order even though they are only first order accurate. Each formula has the same general form as given in (4) but the restrictions given in (4a) are no longer applicable.

For the purposes of the error analysis, the various terms are expanded as follows:

$$w^{(1)} = \mu w^n + \lambda a\, \delta f^n.$$

Using the definition for the averaging and differencing operators μ and δ, we get

$$\mu w = w + \frac{(\Delta x)^2}{8} w_{xx} + O((\Delta x)^4),$$

$$\delta f = f_x + \frac{(\Delta x)^2}{24} f_{xxx} + O((\Delta x)^4), \tag{10}$$

and so

$$w^{(1)} = w + a\,\Delta t\, f_x + \frac{(\Delta x)^2}{8} w_{xx} + \frac{a(\Delta t)(\Delta x)^2}{24} f_{xxx} + O((\Delta x)^4 + (\Delta t)^4);$$

therefore,

$$f^{(1)} = f(w^{(1)})$$

$$= f + f_w\left[a\,\Delta t f_x + \frac{(\Delta x)^2}{8}w_{xx} + \frac{a(\Delta t)(\Delta x)^2}{24}f_{xxx}\right]$$

$$+ \frac{f_{ww}}{2}\left[(a\,\Delta t)^2 f_x^2 + \frac{a(\Delta t)(\Delta x)^2}{24}f_x w_{xx}\right]$$

$$+ \frac{f_{www}}{6}(a\,\Delta t)^3 f_{xxx} + O((\Delta x)^4 + (\Delta t)^4)$$

$$= f + a\,\Delta t f_t + \frac{(a\,\Delta t)^2}{2}A_t w_t + \frac{(a\,\Delta t)^3}{6}(A_{tt}w_{tt} - A_t w_{tt})$$

$$+ a\frac{(\Delta t)(\Delta x)^2}{24}(Af_{xxx} + 3A_t w_{xx}) + O((\Delta x)^4 + (\Delta t)^4),$$

where we have used the formulas

$$f_w = A, \qquad f_{ww} = A_w = A_t/w_t,$$

$$f_{www} = A_{ww} = \frac{A_{tt}}{w_t^2} - \frac{A_t w_{tt}}{w_t^3}. \tag{11}$$

Thus, we have that

$$f_x^{(1)} = w_t + a\,\Delta t w_{tt} + \frac{(a\,\Delta t)^2}{2}(A_t w_t)_x + \frac{(a\,\Delta t)^3}{6}(A_{tt}w_t - A_t w_{tt})_x$$

$$+ \frac{a(\Delta t)(\Delta x)^2}{8}(Af_{xxx} + A_t w_{xx})_x + O((\Delta t)^4 + (\Delta t)^4). \tag{12}$$

We now repeat the process with $w^{(2)}$ and find

$$w^{(2)} = \varepsilon w^n + (1-\varepsilon)\mu^2 w^n + \lambda(\alpha_1\,\delta f^{(1)} + \alpha_2\,\delta\mu f^n)$$

$$= w^n + \frac{1-\varepsilon}{4}(\Delta x)^2 w_{xx} + \Delta t(\alpha_1 f_x^{(1)} + \alpha_2 f_x^n) + \frac{(\Delta x)^2(\Delta t)}{24}(f_{xxx}^{(1)} + 4f_{xxx}^n)$$

$$+ O((\Delta x)^4 + (\Delta t)^4).$$

Since we now have $f_x^{(1)}$ and hence $f_{xxx}^{(1)}$ from (12), we can continue this process until we calculate $w^{(4)}$. When this process is completed one is left with an error of $O((\Delta x)^5)$ with the following equation for $w^{(4)}$:

$$
\begin{aligned}
w^{(4)} = w &+ \Delta t[\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 + \gamma_5 + \gamma_6]w_t \\
&+ (\Delta t)^2[\gamma_1\theta_3 + \gamma_2\theta_2 + (\gamma_3 + \gamma_4)\theta_1]w_{tt} \\
&+ (\Delta t)^3\Big\{\frac{1}{2}[\gamma_1\theta_3^2 + \gamma_2\theta_2^2 + (\gamma_3 + \gamma_4)\theta_1^2](A_t w_t)_x \\
&\qquad + [\gamma_1(\beta_1\theta_2 + \beta_2\theta_1) + \gamma_2\alpha_1\theta_1](Aw_{tt})_x\Big\} \\
&+ (\Delta t)^4\Big\{\frac{1}{6}[\gamma_1\theta_3^3 + \gamma_2\theta_2^3 + (\gamma_3 + \gamma_4)\theta_1^3](A_{tt}w_t)_x \\
&\qquad + \Big[\gamma_1\theta_3(\beta_1\theta_2 + \beta_2\theta_1) + \gamma_2\alpha_1\theta_1\theta_2 \\
&\qquad\quad - \frac{1}{6}(\gamma_1\theta_3^3 + \gamma_2\theta_2^3 + (\gamma_3 + \gamma_4)\theta_1^3)\Big](A_t w_{tt})_x \\
&\qquad + \frac{1}{2}[\gamma_1(\beta_1\theta_2^2 + \beta_2\theta_1^2) + \gamma_2\alpha_1\theta_1^2][A(A_t w_t)_x]_x \\
&\qquad + \gamma_1\beta_1\alpha_1\theta_1[A(Aw_{tt})_x]_x\Big\} \\
&+ (\Delta t)(\Delta x)^2\Big\{\frac{1}{4}\Big[\Big(\frac{3}{2} - \eta\Big)\gamma_1 + \gamma_2(1 - \varepsilon) + \frac{\gamma_3 + \gamma_4}{2}\Big](Aw_{xx})_x \\
&\qquad + \frac{1}{24}[\gamma_1 + 4\gamma_2 + \gamma_3 + 7\gamma_4 + 4\gamma_5 + 10\gamma_6]w_{txx}\Big\} \\
&+ (\Delta t)^2(\Delta x)^2\Big\{\frac{1}{8}[\gamma_1(\beta_2 + 2\beta_1(1 - \varepsilon) + \gamma_2\alpha_1]\cdot[A(Aw_{xx})_x]_x \\
&+ \frac{1}{24}[\gamma_1(\beta_1 + 4\beta_2 + \beta_3 + 7\beta_4 + \gamma_2(\alpha_1 + 4\alpha_2) + (\gamma_3 + \gamma_4)\theta_1](Aw_{txx})_x \\
&+ \frac{1}{8}\Big[2\gamma_1\theta_3\Big(\frac{3}{2} - \eta\Big) + (\gamma_3 + \gamma_4)\theta_1 + 2\gamma_2(1 - \varepsilon)\theta_2\Big](A_t w_{xx})_x \\
&+ \frac{1}{24}\Big[\gamma_1\theta_3 + 4\gamma_2\theta_2 + (\gamma_3 + 7\gamma_4)\theta_1\Big]w_{ttxx}\Big\} + O((\Delta x)^5 + (\Delta t)^5).
\end{aligned}
$$

The requirement that $w^{(4)}$ be fourth order accurate leads to the following equations for the parameters appearing above and in (4).

(13)

1. $\gamma_1\beta_1\alpha_1\theta_1 = \frac{1}{24}$,
2. $\gamma_1\beta_1\theta_2^2 + \gamma_1\beta_2\theta_1^2 + \gamma_2\alpha_1\theta_1^2 = \frac{1}{12}$,
3. $\gamma_1\theta_3^3 + \gamma_2\theta_2^3 + (\gamma_3 + \gamma_4)\theta_1^3 = \frac{1}{4}$,
4. $\gamma_1\beta_1\theta_2\theta_3 + \gamma_1\beta_2\theta_1\theta_3 + \gamma_2\alpha_1\theta_1\theta_2 - \frac{1}{6}(\gamma_1\theta_3^3 + \gamma_2\theta_2^3 + (\gamma_3 + \gamma_4)\theta_1^3) = \frac{1}{12}$,
5. $\gamma_1\theta_3^2 + \gamma_2\theta_2^2 + (\gamma_3 + \gamma_4)\theta_1^2 = \frac{1}{3}$,
6. $\gamma_1\beta_1\theta_2 + \gamma_1\beta_2\theta_1 + \gamma_2\alpha_1\theta_1 = \frac{1}{6}$,
7. $\gamma_1\theta_3 + \gamma_2\theta_2 + (\gamma_3 + \gamma_4)\theta_1 = 0$,
8. $\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 + \gamma_5 + \gamma_6 = 1$,
9. $2\gamma_1\left(\frac{3}{2} - \eta\right) + 2\gamma_2(1 - \varepsilon) + \gamma_3 + \gamma_4 = 0$,
10. $\gamma_1 + 4\gamma_2 + \gamma_3 + 7\gamma_4 + 4\gamma_5 + 10\gamma_6 = 0$,
11. $2\gamma_1\beta_1(1 - \varepsilon) + \gamma_1\beta_2 + \gamma_2\alpha_1 = 0$,
12. $\gamma_1(\beta_1 + 4\beta_2 + \beta_3 + 7\beta_4) + \gamma_2(\alpha_1 + 4\alpha_2) + (\gamma_3 + \gamma_4)\theta_1 = 0$,
13. $2\gamma_1\theta_3\left(\frac{3}{2} - \eta\right) + 2\gamma_2(1 - \varepsilon)\theta_2 + (\gamma_3 + \gamma_4)\theta_1 = 0$,
14. $\gamma_1\theta_3 + 4\gamma_2\theta_2 + (\gamma_3 + 7\gamma_4)\theta_1 = 0$,
15. $a = \theta_1$,
16. $\alpha_1 + \alpha_2 = \theta_2$,
17. $\beta_1 + \beta_2 + \beta_3 + \beta_4 = \theta_3$.

We thus have a system of 17 nonlinear equations for 18 unknowns. It can be shown that these equations require that $\theta_3 = 1$. With regard to the other intermediate times θ_1, θ_2, the general solution consists of two cases:

(a) If θ_1 is chosen equal to θ_2, then one must choose $\theta_1 = \theta_2 = 1/2$; (b) if θ_1 and θ_2 differ, they may be chosen arbitrarily subject to the restriction that they may not be equal to unity. For the first possibility, we get the following choice of parameters:

Case I.

$$\theta_1 = \theta_2 = \frac{1}{2}, \qquad \theta_3 = 1,$$

$$a = \frac{1}{2}, \quad \alpha_1 = \frac{1}{2}, \quad \alpha_2 = 0, \quad \varepsilon = \frac{3}{2}, \quad \eta = \frac{3}{2},$$

$$\beta_1 = 1, \quad \beta_2 = 0, \quad \beta_3 = \frac{1}{2}, \quad \beta_4 = -\frac{1}{2}, \tag{14}$$

$$\gamma_1 = \frac{1}{6}, \quad \gamma_2 = \frac{1}{3}, \quad \gamma_3 = \frac{2}{3}, \quad \gamma_4 = -\frac{1}{3}, \quad \gamma_5 = \frac{1}{4}, \quad \gamma_6 = -\frac{1}{12}.$$

With the second possibility, we have Case II.

Case II.

$$\theta_1 \neq \theta_2, \quad \theta_1 \neq 1, \quad \theta_2 \neq 1 \quad \text{otherwise } \theta_1, \theta_2 \text{ free},$$

$$\gamma_1 = \frac{1/4 - (\theta_1 + \theta_2)/3 + (\theta_1\theta_2/2)}{(1 - \theta_1)(1 - \theta_2)},$$

$$\gamma_2 = \frac{1 - 2\theta_1}{12\theta_2(1 - \theta_2)(\theta_2 - \theta_1)}, \qquad \bar{\gamma} = -\frac{1 - 2\theta_2}{12\theta_1(1 - \theta_1)(\theta_2 - \theta_1)},$$

$$\gamma_4 = -\frac{1/2 + \theta_2 - 2\theta_1 + \theta_1\theta_2 - \theta_2^2}{12\theta_1(1 - \theta_2)(\theta_2 - \theta_1)}, \qquad \gamma_3 = \bar{\gamma} - \gamma_4,$$

$$\gamma_5 = \frac{5}{3} - \frac{3\gamma_1 + 2\gamma_2 + 3\gamma_3 + \gamma_4}{2}, \qquad \gamma_6 = \frac{\gamma_1 + \gamma_3 + \gamma_4}{2} - \frac{2}{3}, \tag{15}$$

$$\varepsilon = 1 - \frac{\theta_2}{2\theta_1} \cdot \frac{1 - 2\theta_2}{1 - 2\theta_1}, \qquad \eta = \frac{3}{2} - \frac{\bar{\gamma}}{2\gamma_1} \frac{\theta_2 - \theta_1}{1 - \theta_2},$$

$$\alpha_1 = \frac{\theta_2 - \theta_1}{2(1 - 2\theta_1)} \cdot \frac{\theta_2}{\theta_1}, \qquad \alpha_2 = \theta_2 - \alpha_1,$$

$$\beta_1 = \frac{\gamma_2(1 - \theta_2)}{\gamma_1}, \qquad \beta_2 = -\frac{\alpha_1\gamma_2}{\gamma_1} + 2\beta_1(1 - \varepsilon),$$

$$\beta_4 = -\frac{\beta_2}{2} - \frac{1}{6} - \frac{\gamma_2}{6\gamma_1}(\alpha_1 + 4\alpha_2) - \frac{\bar{\gamma}}{6\gamma_1}\theta_1, \qquad \beta_3 = 1 - \beta_1 - \beta_2 - \beta_4.$$

Since θ_1 and θ_2 are arbitrary one can choose them to either increase the number of coefficients equal to zero in (4) or can choose θ_1, θ_2 so that both $w^{(2)}$ and $w^{(3)}$ are second order accurate rather than just first order.

Since this scheme uses the same five point lattice as that of Abarbanel and Gottlieb, it is easy to show that they must have the same amplification matrix. Hence, as was shown there, the stability condition is

$$\frac{\Delta t}{\Delta x}\rho(A) \leqq 1 \tag{16}$$

which is the same as found numerically for the 3 level 4-step scheme of the previous section (see (9)). It is clear that this method requires slightly more work than that required by the 3 level scheme since in solving for $w^{(4)}$ we use f^n, $f^{(1)}$, $f^{(2)}$ and $f^{(3)}$ while the 3 level doesn't use $f^{(2)}$. However, if the evaluation of the fluxes is sufficiently complicated this difference in time is insignificant. However, both these methods are significantly faster than that of Abarbanel and Gottlieb since now four flux evaluations are required at each mesh point rather than nine. In the following section, we shall show that both methods developed here are significantly more efficient than either second order techniques or the leap-frog method of Kriess and Oliger which is second order in time and fourth order in space.

5. Results. In this section, we describe the computational results achieved using the fourth order schemes (the two level equation (4) and the multilevel equation (7) previously described) and compare their efficiency with other known codes. We shall discuss four test problems. The first is a nonlinear scalar problem with a decaying solution. The next two problems are linear scalar problems with sinusoidal solutions. One solution consists of a sine wave in a periodic domain while the other allows the wave to travel down an infinite line. Finally, we shall give results for a Riemann problem in fluid dynamics and hence show that these schemes can handle vector equations containing discontinuous flows. We shall thus demonstrate the usefulness of fourth order methods for a large variety of problems. We shall investigate energy conservation properties as well as L_2 and phase errors.

The first problem that we consider is

$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \qquad 1 \leqq x \leqq 2, \quad 0 \leqq t \leqq 10, \tag{16a}$$

$$u(0, x) = 2x, \qquad u(t, 1) = t^2 + 2 - t. \tag{16b}$$

The analytic solution is

$$u(t, x) = (t^2 + 2x)^{1/2} - t. \tag{16c}$$

It is easily verified that $\sqrt{2} \leqq u(0, x) \leqq 2$, $0.09 \leqq u(10, x) \leqq 0.2$, and the solution is well-behaved throughout the domain of interest. The fourth order numerical methods require boundary data at the four points $x = 1$, $1 + \Delta x$, $2 - \Delta x$, 2. At $x = 1$ the boundary data is given by (16b). At the other three points, we assign to u to known anaytic solution (16c). In a future article, we shall discuss other techniques for handling the boundary difficulties. Similarly, for the Richtmyer second order method, we use the analytic solution at the points $x = 1$ and $x = 2$.

In Table 1 we show the effects of varying the θ_i of the intermediate steps for the two level formula as given by (4), (13), (14), (15). The numbers given in this table are the absolute values of the difference between the numerical and analytical solutions at the center of the region. Also given are the computer running times for the various schemes. In this table we have included the three level ((7), (8)) as well as two level ((4), (14), (15)) four-step methods as well as the Richtmyer method [11] and the Kreiss–Oliger algorithm [5]. As is to be expected, the second order method is not comparable for problems where the higher derivatives are small.

Thus, the Richtmyer method with $\Delta x = 1/640$ gives results comparable with the four step techniques at $\Delta x = 1/10$. In addition to the increased storage required by the finer mesh, the second order method takes about 1600 times as much as computer time to achieve comparable accuracy. Since the space derivatives are much smaller than the time derivatives the major error comes from the approximation of the time derivatives. Hence, the Kreiss–Oliger scheme acts as a second order method. Furthermore, it becomes most "efficient" at extremely small ratios of Δt to Δx. As seen in Table 1, the leap-frog fourth order method is even less efficient than the Richtmyer scheme for this decay problem. Using finer meshes but larger $\Delta t/\Delta x$ gives even worse results for the second order time and fourth order in space methods.

TABLE 1
Error at center $\times 10^{-11}$
2 Level

Δx	3 Level	$\theta_- = 1/6$ $\theta_2 = 1/3$	$\theta_1 = 1/4$ $\infty\theta_2 = 1/2$	$\theta_1 = 1/2$ $\theta_2 = 1/2$	Richtmyer	Kreiss CFL = 0.01
$\frac{1}{10}$	4664 / 0.99	2250 / 0.13	1910 / 0.13	975 / 0.13		
$\frac{1}{20}$	289 / 0.31	169 / 0.43	143 / 0.43	764 / 0.43		3.8×10^5 / 17.5
$\frac{1}{40}$	17 / 1.1	11 / 1.6	9.6 / 1.6	5.2 / 1.6	4.7×10^5 / 0.5	
$\frac{1}{80}$	1.0 / 4.3	0.7 / 6.3	0.6 / 6.3	0.3 / 6.3		
$\frac{1}{640}$					1800 / 212	

Numbers above dotted line are the errors, numbers below the dotted line are running time.

From Table 1, we see that for this problem the parameters $\theta_1 = \theta_2 = 1/2$ are optimal. Furthermore, the three level four-step formula is slightly less accurate than the two level algorithm. However, these differences are negligible compared to differences between fourth order methods (in space and time) and second order

methods. Thus, for problems that involve decaying modes the use of fourth order methods in time are essential. We shall now compare these schemes for a wave propagation problem.

In the second problem, we consider the linear equation

$$\begin{aligned} &u_t + u_x = 0, \qquad 0 \leqq x \leqq 1, \quad 0 \leqq t \leqq 10, \\ &u(t, 0) = u(t, 1), \qquad u(0, x) = \sin(2n\pi x) \end{aligned} \tag{17}$$

which has the analytic solution

$$u(t, x) = \sin(2n\pi(x - t)),$$

where n is the number of wave lengths in the interval $0 \leqq x \leqq 1$.

This is a wave propagation problem in which the solution does not decay. In fact, if we let $E(t) = \int_0^1 u^2(x, t)\, dx$, then $E(t)$ is a constant for the analytic solution. In Table 2 we show the numerical solution with $\Delta x = 1/20$ and $n = 1$. We have compared the solutions for the following schemes: (a) Richtmyer 2-step; (b) 3 level 4th order formula (7); (c) 2 level order method (4); since this is a linear problem with constant coefficient all choices of θ_i give *identical* results; (d) SHASTA Code (see [3]); 4th order (space only) leap-frog method of Kreiss and Oliger [5] with a time step

$$\text{(e)}\ \frac{\Delta t}{\Delta x} = 0.65, \quad \text{(f)}\ \frac{\Delta t}{\Delta x} = 0.5, \quad \text{(g)}\ \frac{\Delta t}{\Delta x} = 0.1, \quad \text{(h)}\ \frac{\Delta t}{\Delta x} = 0.05.$$

The time steps for the first three methods were chosen as $\Delta t/\Delta x = 0.9$ and for the SHASTA Code as $\Delta t/\Delta x = 0.45$. These latter ones are chosen as being close to the maximum allowable by stability considerations. The accuracy of the solution can be measured by both the L_2 error

$$(\text{EL})^2 = (\Delta x) \sum (u(t, x) - u_j^n)^2$$

and the conservation of total energy

$$(\text{TE})^2 = \Delta x \sum (u_j^n)^2.$$

The phase errors are not uniquely determined, as they depend on which part of the wave one is examining. Instead it is easiest to estimate the phase errors by simply looking at the graphs.

As expected, the leap-frog type scheme preserves the energy to several significant figures. For this long wave length, all the multistep schemes also preserve the total energy and only the SHASTA algorithm introduces any significant dissipation. Introduction of velocity dependent correction in the SHASTA code does not improve this situation but only causes the sine waves to appear as square waves. Thus, while the SHASTA Code produces sharp shocks it can also steepen smoother waves unless the correction terms are properly chosen. The correct parameters for the correction term are usually not known in advance.

In Table 3 we show the results for the same problem but with $\Delta x = 1/80$. As expected the accuracy improves with the order of the scheme and all methods nearly conserve the total energy even after ten complete time periods have been completed. The error in the SHASTA code decreases as with a second order method

TABLE 2
Problem 2, $\Delta x = 1/20$, $n = 1$

	Richtmyer $\lambda = 0.9$	3 Level $\lambda = 0.9$	2 Level $\lambda = 0.9$	SHASTA $\lambda = 0.45$	Kreiss–Oliger $\lambda = 0.65$	Kreiss–Oliger $\lambda = 0.5$	Kreiss–Oliger $\lambda = 0.25$	Kreiss–Oliger $\lambda = 0.1$	Kreiss–Oliger $\lambda = 0.05$
EL	1.3×10^{-1}	3.2×10^{-3}	2.1×10^{-3}	1.0×10^{-1}	3.0×10^{-1}	1.7×10^{-1}	3.1×10^{-2}	7.0×10^{-3}	1.2×10^{-2}
TE	0.9597	0.9987	0.9992	0.860	0.9998	1.0000	1.0000	0.9999	0.9999
Time	0.24	0.47	0.51	1.0	0.13	0.19	0.28	0.70	1.2

TABLE 3
Problem 2, $\Delta x = 1/80$, $n = 1$

	Richtmyer $\lambda = 0.9$	3 Level $\lambda = 0.9$	2 Level $\lambda = 0.9$	SHASTA $\lambda = 0.45$	Kreiss–Oliger $\lambda = 0.65$	Kreiss–Oliger $\lambda = 0.5$	Kreiss–Oliger $\lambda = 0.25$	Kreiss–Oliger $\lambda = 0.1$	Kreiss–Oliger $\lambda = 0.05$
EL	8.7×10^{-3}	8.6×10^{-6}	8.5×10^{-6}	5.2×10^{-3}	1.9×10^{-2}	1.1×10^{-2}	2.8×10^{-2}	4.9×10^{-4}	5.8×10^{-5}
TE	0.9993	0.9999	0.9999	0.9981	0.9999	1.0000	1.0000	1.0000	1.0000
Time	3.6	6.5	7.3	15.0	1.5	1.8	3.6	8.9	16.2

since the equation we are considering has constant coefficients. The L_2 error of the SHASTA code is slightly smaller than the Richtmyer method but takes about 2.5 times as long due to the increased complexity of the formulas and to the smaller allowable time stop.

The fourth order methods are again seen to be more efficient. Since the domain of dependency of the SHASTA Code is even larger than that for the fourth order methods there seems to be little advantage to the SHASTA algorithm.

In Table 4, we give the results for $n = 4$ and $\Delta x = 1/80$. As expected the accuracy deteriorates as the frequency increases. If we compare this table with Table 2 we see that the L^2 error increases by a factor of 4 when we increase the frequency by a factor of 4 but keep the same number of points per wave length. This is to be expected since we expect the global error to be of the form

$$(\Delta x)^4 u_{xxxxx} + (\Delta t)^4 u_{ttttt}$$

for the fourth order schemes. When we increase the frequency of u by a factor of k, then u_{xxxxx} increases by k^5. In order to keep the points per wave length constant we must decrease Δx by a factor of k which decreases $(\Delta x)^4$ by k^{-4}. In total the error increases by $k^{-4} \cdot k^5 = k$, with $k = 4$ in our case. Thus, in contrast to the conclusions of Kreiss–Oliger, we don't expect the error to be a function only of the number of points per wave length.

From these tables we also see that the largest permissible time step is not the optimal one for the fourth order leap-frog scheme. This is in contrast to Lax–Wendroff type schemes where the optimal time step is the largest permissible without violating stability conditions (see, for example, Turkel [15]). For the second order in time and fourth order in space methods Oliger [9] indicates that the optimal Δt is achieved by keeping $\Delta t/(\Delta x)^2$ constant; i.e., this method has the disadvantage of explicit parabolic schemes. This implies that $\lambda = \Delta t/\Delta x$, which appears in the finite difference equations, is a function of the mesh width as well as the dominant frequencies in the solution. Hence, for higher frequency waves which require finer meshes, the method becomes less competitive. Furthermore, for nonlinear problems the relevant frequencies are not always known in advance.

In Table 4 we give the results for a higher frequency wave $n = 4$. We choose the mesh spacing as $\Delta x = 1/80$ so that the errors for the fourth order methods would be of the order of one percent. Comparing the Kreiss–Oliger scheme with the four-step methods, we see that the true fourth order algorithms have an error 1/3 as large while using 25 percent less computer time then the "best" Kreiss–Oliger scheme ($\Delta t/\Delta x = 1/10$). Hence the four-step methods are more than 4 times as efficient as the fourth order leap-frog schemes for this problem. The Richtmyer scheme requires a mesh of $\Delta x = 1/640$ in order to achieve an L_2 error of the same order of magnitude, i.e., one percent. With this finer mesh, the Richtmyer method takes about 25 times as long as the four-step methods to achieve the same accuracy. Thus, both the uniformly fourth order method and the fourth order in space algorithms are clearly better than the second order Richtmyer methods. The SHASTA method gives a slightly smaller error, for fixed mesh, than the Richtmyer method but is less efficient for this problem due to the large computer times required. Moreover, the energy dissipation for this moderate wave number is quite unacceptable.

TABLE 4
Problem 2, $\Delta x = 1/80$, $n = 4$

	Richtmyer $\lambda = 0.9$	3 Level $\lambda = 0.9$	2 Level $\lambda = 0.9$	SHASTA $\lambda = 0.45$	$\lambda = 0.65$	$\lambda = 0.5$	Kreiss–Oliger $\lambda = 0.25$	$\lambda = 0.1$	$\lambda = 0.05$
EL	5.9×10^{-1}	1.4×10^{-2}	8.6×10^{-3}	3.2×10^{-1}	1.0×10^{0}	6.5×10^{-1}	1.2×10^{-1}	2.78×10^{-2}	4.97×10^{-2}
TE	0.8480	0.9947	0.9970	0.4551	1.0003	0.9997	0.9999	0.9999	0.9999
Time	3.4	6.3	7.3	15.2	1.4	1.8	3.6	8.8	17.4

Before continuing, we wish to warn the reader about the use of computer times in comparing the efficiency of codes. Even under ideal circumstances the same algorithm can be programed in many ways either to optimize time or computer storage or some combination. The ratio of running time for different codes may differ markedly when programmed according to different criteria and certainly will differ for different equations. Thus, for example, the fourth order 2 level scheme was coded in two different ways. One kept all the intermediate levels at all the mesh points in the machine and so optimized running time. Alternatively, we kept only one complete level in the computer at a given time. For the other time levels, we kept a minimum lattice and shifted this from point to point. This minimized storage but increased the running time. For this problem, the ratio of these two running times was about 5 to 2. Should we increase the complexity of the flux term, this ratio would decrease as the shifting of fluxes would now be less consequential. Finally, it should be pointed out that programs are usually not optimized as much as possible and so different programs are even less comparable. Thus, the numbers given in these tables should be used as guides and order of magnitudes only.

As a third problem we consider a packet of 4 waves moving to the right down an infinite line:

$$u_t + u_x = 0, \quad 0 \leqq x \leqq X, \quad 0 \leqq t \leqq 10, \tag{18}$$

$$u(t, x) = \begin{cases} 0, & 0 \leqq x \leqq x_0, \\ \sin[2n\pi(x - x_0)], & x_0 \leqq x \leqq x_0 + 1, \\ 0, & x_0 + 1 \leqq x \leqq X, \end{cases}$$

with solution

$$u(t, x) = \begin{cases} 0, & 0 \leqq x \leqq x_0 + t, \\ \sin[2n\pi(x - x_0 - t)], & x_0 + t \leqq x_0 + t + 1, \\ 0, & x_0 + t + 1 \leqq x \leqq X, \end{cases}$$

X chosen sufficiently large so that $X \gg x_0 + 11$. The results for the various schemes, are shown in Figs. 1–5. They are all given for the same mesh spacing, $\Delta x = 1/40$, because it is our feeling that real-life computations (especially in 2 and 3 space dimensions) are storage-limited rather than computing-time limited. Each graph indicates $\Delta t/\Delta x$, the total running time, the energy conservation (TE = final total energy divided by the initial energy) and the relative L_2 error, EL. As can be seen from the graphs the second order algorithms (Richtmyer and SHASTA) have considerably worse amplitude and phase errors than the 4th order schemes. Even with a finer mesh ($\Delta x = 1/80$) and a running time longer than that of fourth order schemes at $\Delta x = 1/40$, the second order methods give results that are inferior to those of the fourth order method. Among the fourth order schemes, the Kreiss–Oliger one (chosen with the "best" Δt) has a spurious oscillatory wake, with relatively large amplitude, trailing the wave packet. This is due to the energy conserving property of leap-frog type schemes. This phenomena persists with a finer mesh $\Delta x = 1/60$ (see Fig. 6) where the running time is

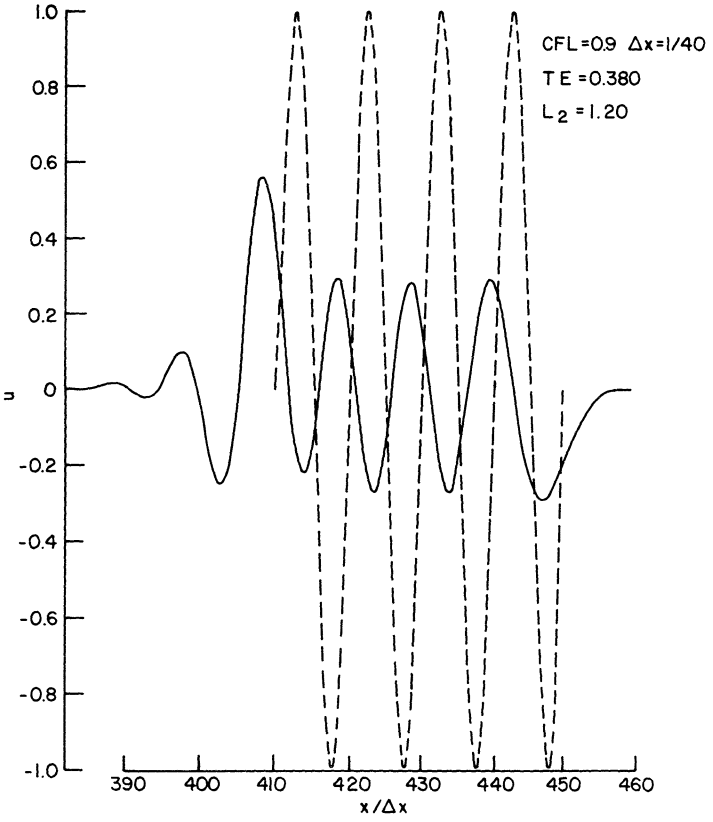

FIG. 1. *Solution to* (18), *with Richtmyer method at* $t = 10.012$. $\Delta x = 1/40$, TE = 0.380, EL = 1.2, *running time* = 7.0 *seconds. The dotted line is the analytic solution while the solid line is the numerical solution.*

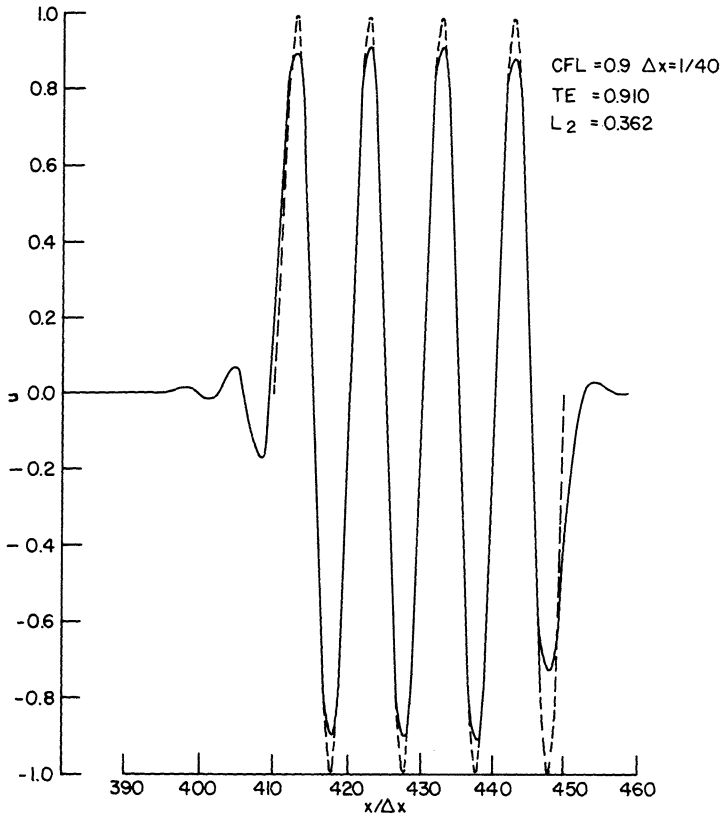

FIG. 2. *Solution to* (18), *using* 2 *level,* 4th *order scheme,* (4), *at* $t = 10.012$. $\Delta x = 1/40$, $\lambda = 0.9$, TE = 0.910, EL = 0.362, *time* = 22.3

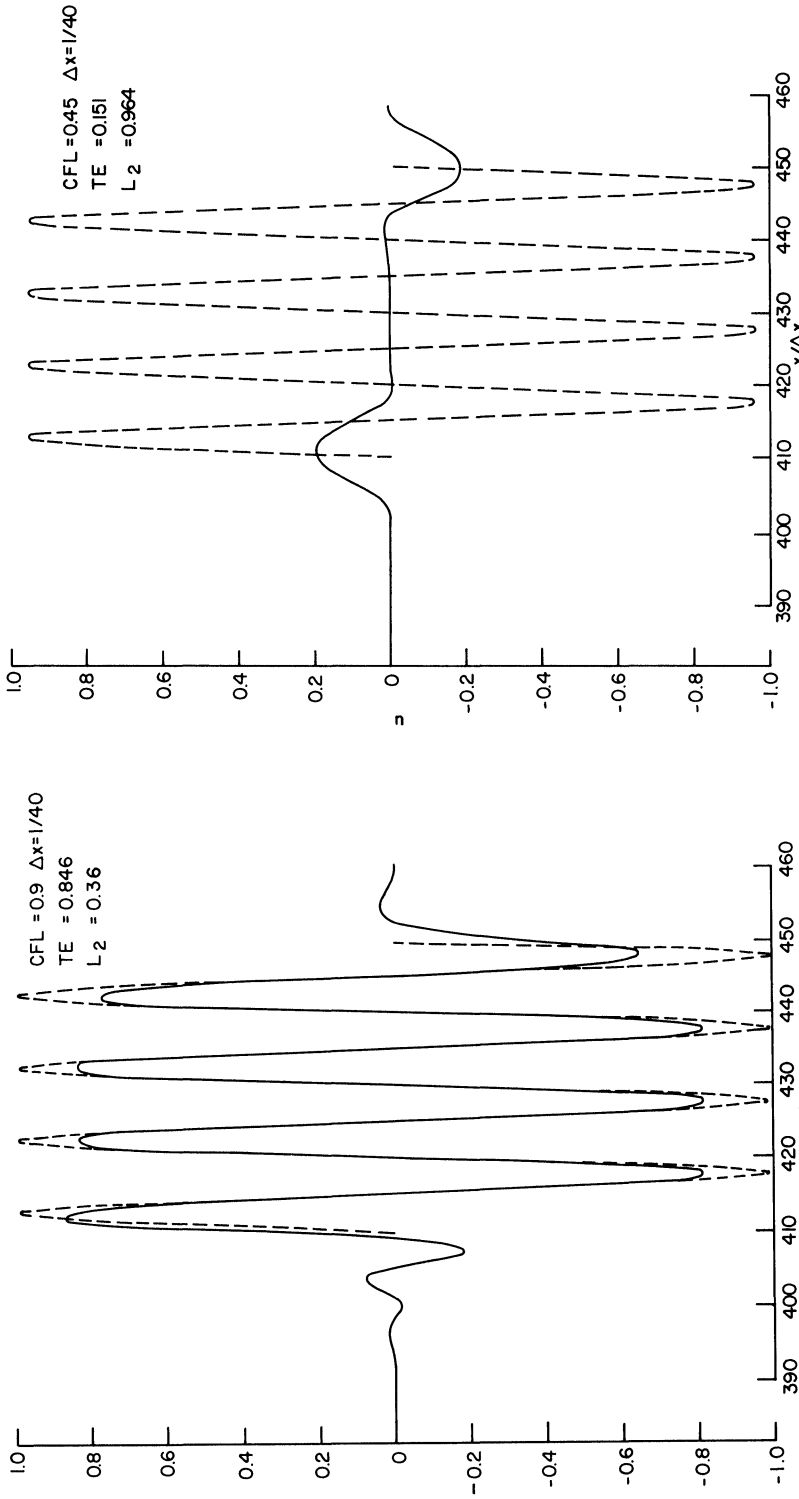

FIG. 4. *Solution to* (18), *using SHASTA Code, at* $t = 10.001$, $\Delta x = 1/40$, $\lambda = 0.45$, TE = 0.151, EL = 0.964, *running time* = 21.0

FIG. 3. *Solution to* (18), *using* 3 *level,* 4th *order scheme,* (7), *at* $t = 10.012$, $\Delta x = 1/40$, $\lambda = 0.9$, TE = 0.846, EL = 0.360, *running time* = 21.8.

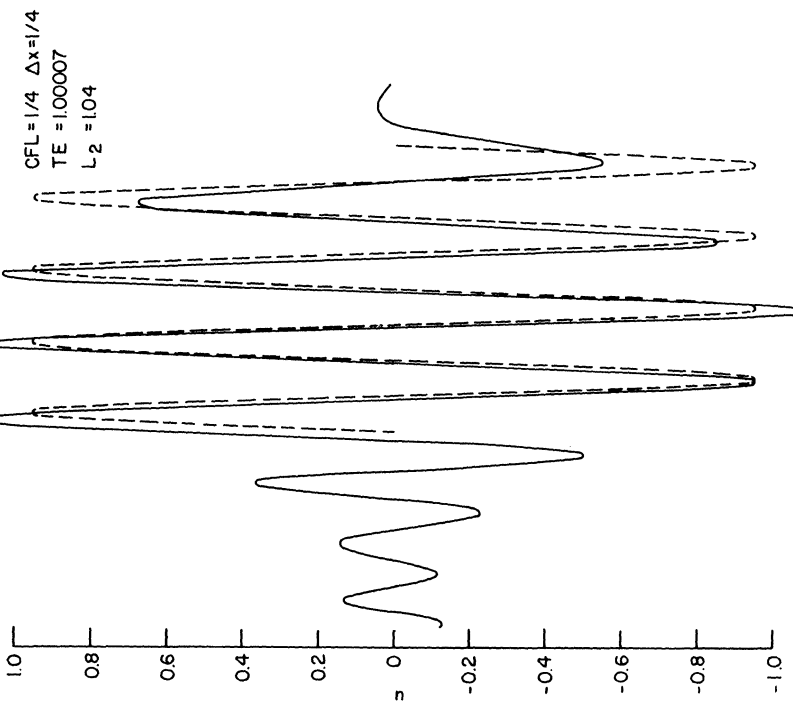

FIG. 5. *Solution to* (18), *using Kreiss–Oliger scheme, at* $t = 10.000$, $\Delta x = 1/40$, $\lambda = 0.25$, TE = 1.0000, EL = 1.04, *running time* = 12.5

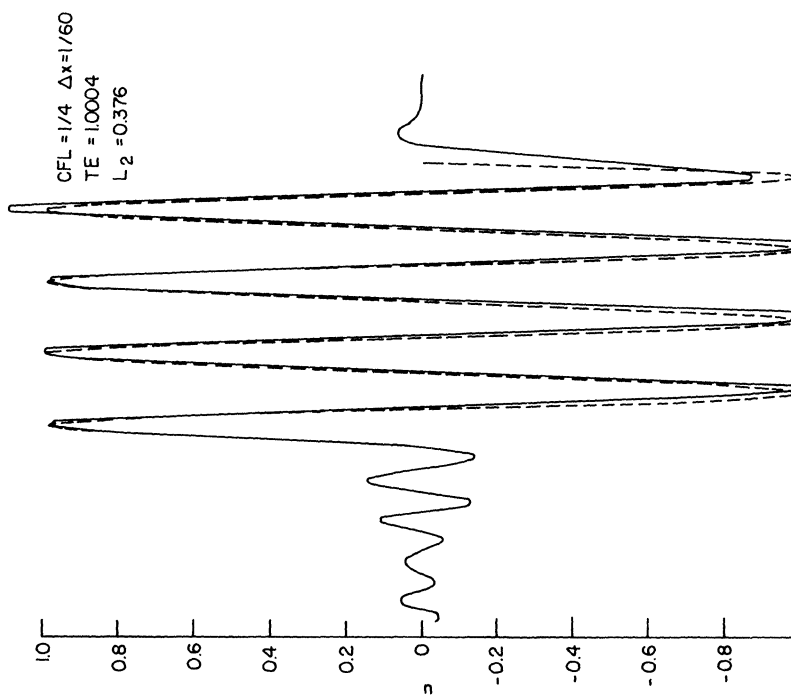

FIG. 6. *Solution to* (18), *using Kreiss–Oliger scheme, at* $t = 10.000$, $\Delta x = 1/60$, $\lambda = 0.25$, TE = 1.0004, EL = 0.0376, *running time* = 22.2

comparable to that of the fourth order method with $\Delta x = 1/40$. These results, including others not shown, indicate that the true fourth order methods (i.e., in space and in time) are superior to other schemes with regard to phase errors, L_2 errors and containment of the trailing oscillations. This holds when the computer running times are kept comparable and even more so when storage requirements are held comparable rather than the running time. There is a slight clipping of the maximum amplitude in the dissipative fourth order methods but considerably less than in the second order methods.

In the last problem, we solve a Riemann problem for fluid dynamics.

$$\begin{aligned} &\rho_t + (\rho u)_x = 0, \\ &(\rho u)_t + (\rho u^2 + p)_x = 0, \\ &E_t + [u(E + p)]_x = 0, \end{aligned} \tag{19}$$

where ρ, u, p and e are, respectively, the density, velocity, pressure and internal energy and $E = f(e + (1/2)u^2)$,

$$p = (\gamma - 1)\rho e.$$

The initial conditions are

$x < 0,$	$x > 0,$
$\rho = \frac{1}{2},$	$\rho = \frac{1}{2.245},$
$u = 0,$	$u = 0.698,$
$p = 0.571,$	$p = 3.528.$

The solution consists of a rarefaction wave moving to the left, a shock propagating to the right followed by a contact discontinuity also moving to the right. Since the shock wave and contact discontinuity originally coincide an artificial viscosity is needed to prevent the density and pressure from becoming negative.

We therefore add to the right-hand side of the fourth equation in system (4) a term that approximates

$$C(\Delta t)(\Delta x)^4 \frac{\partial^2}{\partial x^2}(|u_x| u_{xx}).$$

In Figs. 7a, 7b is shown the result for the 2 level 4-step scheme with $t = 30$ and $C = 0.1$ with Fig. 7a showing the density and 7b the pressure. As expected in regions of sharp gradients, the solution is not better than that of lower order methods (see, for example, Burstein and Mirin [4]). The only advantage is that there will be less smearing of the contact discontinuity. This smearing behaves as $n^{1/k}$ where n is the number of time steps and k is the order of the scheme (see Lax [7]). Nevertheless, this example shows that the four-step methods can automatically handle the shocks as they arise. Hence, the four-step methods are suitable for a large variety of problems which include shocks and wave propagation phenomena.

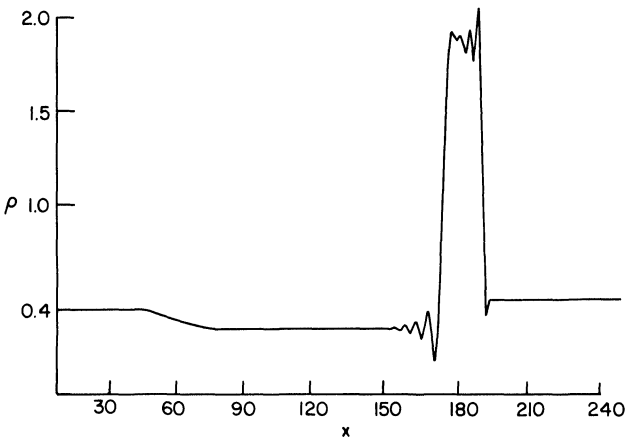

FIG. 7a. *Density for Riemann problem* (19) *using two level four-step method* (4)

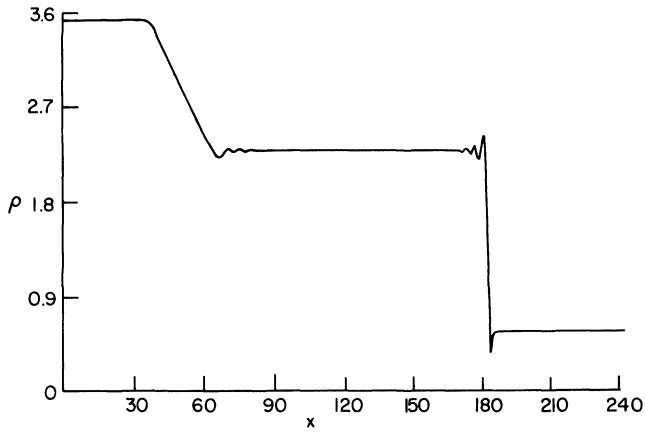

FIG. 7b. *Pressure for same problem as in* 7a

6. Conclusions. The numerical results presented herein indicate that the four-step fourth order accurate algorithms are more efficient than the other tested schemes. For wave propagation problems and errors of the order of one percent, we have shown that the second order methods (Richtmyer and SHASTA) are significantly inferior. For the Kreiss–Oliger algorithm (fourth order in space but second order in time) the "best" $\Delta t/\Delta x$ is not known a priori, and it is dependent on the mesh spacing and the frequencies of the waves in the solution. Thus, high frequency waves (and corresponding fine meshes) will require a disproportionally long running time for their solution. In the numerical examples presented, we used the "best" time step (found experimentally and by reference to the known analytic solution), and still the results of the leap-frog scheme were not as good as those given by the four-step methods. The schemes were compared with regard to phase error, L_2 error and containment of spurious oscillations. For problems with

decaying solutions the use of fourth order methods (in time as well as space) is essential.

It is our feeling that the four-step methods will be increasingly more advantageous as the number of space dimensions increases. At present, we are completing the extension of this work to two and three dimensional algorithms.

Acknowledgments. The numerical work related to this paper was performed on the CDC 6600 computer of Tel-Aviv University.

REFERENCES

[1] S. ABARBANEL AND D. GOTTLIEB, *Higher order accuracy finite difference algorithms for quasi-linear, conservation law hyperbolic systems*, Math. of Computation, 27 (1973), pp. 505–524.

[2] V. B. BALAKIN, *Method of the Runge–Kutta type for gas-dynamics*, U.S.S.R. Computational Math. and Math. Physics, 10 (1970), pp. 208–216.

[3] J. P. BORIS AND D. L. BOOK, *Flux corrected transport. I. SHASTA A fluid transport algorithm that works*, J. Computational Phys., 11 (1973), pp. 38–64.

[4] S. Z. BURSTEIN AND A. A. MIRIN, *Third order difference methods for hyperbolic equations*, Ibid., 5 (1970), pp. 547–557.

[5] H. O. KREISS AND J. OLIGER, *Comparison of accurate methods for the integration of hyperbolic equations*, Tellus, 24 (1972), pp. 199–215.

[6] P. D. LAX, *Weak solutions of nonlinear hyperbolic equations and their numerical computation*, Comm. Pure Appl. Math., 7 (1954), pp. 159–193.

[7] ———, *Hyperbolic systems of conservation laws and the mathematical theory of shock waves*, Regional Conf. Series in Appl. Math., vol. 11, Society for Industrial and Applied Mathematics, Philadelphia, 1973.

[8] P. D. LAX AND B. WENDROFF, *Systems of conservation laws*, Comm. Pure Appl. Math., 13 (1960), pp. 217–237.

[9] J. OLIGER, *Fourth order difference methods for the initial boundary value problems for hyperbolic equations*, Math. of Computation, 27 (1974), pp. 15–26.

[10] R. D. RICHTMYER, *A survey of finite difference methods for nonsteady fluid dynamics*, NCAR Research Tech. notes 63-2, 1963.

[11] R. D. RICHTMYER AND K. W. MORTON, *Difference Methods for Initial Value Problems*, Interscience, New York, 1967.

[12] K. V. ROBERTS AND N. O. WEISS, *Convective difference schemes*, Math. of Computation, 20 (1966), pp. 272–299.

[13] V. V. RUSANOV, *On difference schemes of third order accuracy for nonlinear hyperbolic systems*, J. Computational Phys., 5 (1970), pp. 507–516.

[14] W. G. STRANG, *Trigonometric polynomials and difference methods of maximum accuracy*, J. of Math. and Physics, 41 (1962), pp. 147–154.

[15] E. TURKEL, *Phase error and stability of second order methods for hyperbolic problems, I*, J. Computational Phys., 15 (1974), pp. 226–250.

[16] J. VON NEUMANN AND R. D. RICHTMYER, *A method for the numerical calculation for hydrodynamic shocks*, J. Appl. Phys., 21 (1950), pp. 232–237.