Flux-Corrected Transport.
I. SHASTA, A Fluid Transport Algorithm That Works

JAY P. BORIS AND DAVID L. BOOK

Naval Research Laboratory, Washington, D. C. 20390

Received November 30, 1971

This paper describes a class of explicit, Eulerian finite-difference algorithms for solving the continuity equation which are built around a technique called "flux correction." These flux-conected transport algorithms are of indeterminate order but yield realistic, accurate results. In addition to the mass-conserving property of most conventional algorithms, the FCT algorithms strictly maintain the positivity of actual mass densities so steep gradients and inviscid shocks are handled particularly well. This first paper concentrates on a simple one-dimensional version of FCT utilizing SHASTA, a new transport algorithm for the continuity equation, which is described in detail.

INTRODUCTION

This paper proposes a new approach for numerically solving the continuity equation which yields physically reasonable results even in circumstances where standard algorithms fail. This approach, called flux-corrected transport (FCT), leads to a class of Eulerian finite-difference algorithms which strictly enforce the nonnegative property of realistic mass and energy densities. As a result steep gradients and shocks can be handled particularly well. The method requires no special knowledge about the solution and all internal grid points in the calculation are treated identically.

In most conventional approaches using an Eulerian finite-difference grid [1, 2], the solution of

$$\partial\rho/\partial t = -\nabla\cdot(\rho\mathbf{v}) \tag{1}$$

is approximated by expanding locally up through a given order in the two parameters

$$\delta \equiv \left|\frac{\delta x\,\partial\rho}{\rho\,\partial x}\right|,$$
$$\epsilon \equiv \frac{v\,\delta t}{\delta x}. \tag{2}$$

In pursuing this approach, the standard finite-difference expansions can be endowed with certain desirable qualitative features such as stability and exact conservation. For finite-difference methods of a given order, typically the first or second, the distinguishing qualitative features are determined by the error terms. The crucial importance of the form of the error becomes painfully apparent in regions where ϵ and/or δ is of order unity, that is, in many problems of physical interest.

In regions where the mass density $\rho(x)$ and the flow velocity $\mathbf{v}(x)$ of Eq. (1) are smooth, most second-order schemes such as Lax–Wendroff [3, 4] or leapfrog [1, 4–6] treat the continuity equation quite adequately. In shocks and steep gradients, however, δ is of order unity. In regions of large velocity ϵ is of order unity because one always wants to take as large a timestep as possible. δ is also of order unity near sources of fluid, near sinks, and at interfaces. In all these cases the truncation errors, which are formally of asymptotic order ϵ^3 or $\epsilon\delta^2$, are as large as the solution, so short-wavelength garbage arises in a few cycles. Using higher-order numerical methods in these regions does not help because the error terms at all orders are roughly the same size as the function itself.

The FCT finite-difference technique described here [7] circumvents these steep-gradient problems by requiring another physical property of the continuity equation, positivity, instead of vigorously adhering to an asymptotic ordering. The technique is also stable, mass conservative, and essentially second order in regions where the concept of order is meaningful. An FCT algorithm is of indeterminate order near sharp gradients, the physical behavior of the continuity equation at sharp gradients and discontinuities being folded into the technique directly. Many FCT algorithms are possible; some generalize quite conveniently to two and three dimensions. We will concentrate in this paper on a particular one-dimensional explicit, Eulerian FCT algorithm which we call SHASTA (Sharp And Smooth Transport Algorithm). Several severe computational examples are given using SHASTA to illustrate the general principles of FCT algorithms and their potentialities.

An FCT algorithm consists conceptually of two major stages, a transport or convective stage (Stage I) followed by an antidiffusive or corrective stage (Stage II). Both stages are conservative and maintain positivity. Their interaction enables FCT algorithms to treat strong gradients and shocks without the usual dispersively generated ripples.

Section I of this paper discusses the transport stage of the particular FCT algorithm SHASTA. It has a simple geometric interpretation which generalizes for two- and three-dimensional applications. Section I also contains an analysis of the transport stage. Section II is devoted to the antidiffusion stage of the algorithm (Stage II) which corrects numerical errors introduced in Stage I. The algorithm for Stage II is given and an analysis of error in the overall SHASTA-FCT algorithm is presented.

Section III treats the square-wave problem in one dimension using several conventional difference schemes as well as the FCT technique introduced in this paper and implemented in SHASTA. The concepts of dispersion and diffusion used in Sections I and II are also applied to these conventional schemes. This analysis allows a more complete understanding of the new technique in light of the problems encountered in these earlier finite-difference schemes [1, 4, 8].

Section IV extends the application of the new technique to the full set of fluid equations. The SHASTA algorithm is generalized to include equations of the momentum and pressure type as well as continuity equations. The results of several shock and piston calculations are presented and compared to other numerical methods of solving these problems as reported in the literature [8]. We wish to emphasize here that the new algorithm employs no adjustable artificial viscosity of the von Neumann [1, 9] type and minimizes the undershoot–overshoot phenomena which plague other finite-difference methods [8]. In Section V the workings of the algorithm are described qualitatively and applications of the FCT algorithms are discussed.

I. The Transport Stage (Stage I)

The transport stage of the 1D SHASTA-FCT algorithm has a simple geometric interpretation with extensions to multidimensional calculations. Figure 1(a) shows the Eulerian, uniformly spaced grid at the beginning of the cycle ($t = 0$). The densities $\{\rho_j{}^0\}$ are known and the velocities $\{v_j^{1/2}\}$ are known at $t = \delta t/2$, half a timestep ahead (spatially staggered grids are not used). We seek the densities $\{\rho_j{}^1\}$ at the end of the timestep when $t = \delta t$. Stage I of the algorithm proceeds by considering individually each of the fluid-element trapezoids formed by connecting adjacent density values with straight line segments. This piece-wise linear density profile is fully consistent with the definition of the total mass,

$$M_0 = \sum_j \rho_j \, \delta x. \tag{3}$$

The short arrows at the grid points j and $j + 1$ indicate representative motions the fluid at these grid points might undergo. In the examples and analysis of this paper we consider only $| v \, \delta t/\delta x | < \frac{1}{2}$ so that no grid point can convect past the cell boundaries, indicated by vertical dashed lines in Fig. 1(a), in one cycle. This restriction is probably not strictly necessary but ensures that no two grid points can cross in a single cycle, a necessary condition for this algorithm. It also simplifies the programming immensely.

We follow the motion of this fluid element for one timestep in a Lagrangian sense. The two boundaries move by amounts $v_j^{1/2} \, \delta t$ and $v_{j+1}^{1/2} \, \delta t$. At the end of the

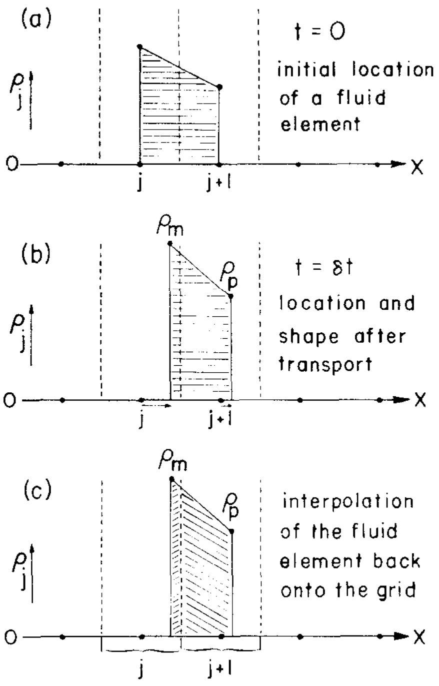
(a)
ρ
j
t = 0
initial location of a fluid element
0
X
j
j+1
(b)
ρm
ρp
t = δt
location and shape after transport
(c)
ρm
ρp
interpolation of the fluid element back onto the grid

FIG. 1. The transport stage of the 1 D FCT algorithm SHASTA (Stage I). Linear interpolations are used throughout so that mass is conserved and the density is nonnegative as long as $| v\delta t/\delta x | < \frac{1}{2}$.

cycle, the fluid element has been convected and deformed as shown in Fig. 1(b). The height of each side of the trapezoid is varied in inverse proportion to the contraction or dilation in x of the base of the trapezoid. Thus

$$\rho_p = \rho_{j+1}^0 \, \delta x/[\delta x + \delta t(v_{j+1}^{1/2} - v_j^{1/2})],$$

and

$$\rho_m = \rho_j^0 \, \delta x/[\delta x + \delta t(v_{j+1}^{1/2} - v_j^{1/2})]. \tag{4}$$

It is clear from these formulas that the area of the fluid element, and hence the mass, is fully conserved. It is also clear that the values ρ_p and ρ_m are always nonnegative if all the $\{\rho_j{}^0\}$ are nonnegative initially and if

$$| v \, \delta t/\delta x | < \tfrac{1}{2}. \tag{5}$$

This ensures that the mass density, interpolated back onto the original grid, is also nonnegative.

The velocities used in Eqs. (4) are evaluated at the centered time but not the centered position with respect to the motion of the fluid at the grid points. Thus the transport is not fully second order. This minor defect is easily rectified by using modified velocities found by linearly interpolating on the grid. That is, we replace $v_j^{1/2}$ in Eqs. (4) [or equivalently Eq. (7) below] with

$$\begin{aligned} \tilde{v}_j^{1/2} &= [1 - \epsilon_j/2]\, v_j^{1/2} + (\epsilon_j/2)\, v_{j+1}^{1/2} \qquad (v_j^{1/2} \geqslant 0), \\ \tilde{v}_j^{1/2} &= [1 - \epsilon_j/2]\, v_j^{1/2} + (\epsilon_j/2)\, v_{j-1}^{1/2} \qquad (v_j^{1/2} < 0), \end{aligned} \tag{6}$$

where $\epsilon_j \equiv | v_j^{1/2}\, \delta t / \delta x |$.

Interpolation of the displaced fluid element back onto the original Eulerian grid is accomplished simply, as shown in Fig. 1(c). The area of the trapezoid lying to the right of the cell boundary [midway between the j-th and the $(j + 1)$ st grid points] can be found by a simple linear interpolation. This amount of fluid is assigned to grid point $j + 1$. The remainder of the fluid in the trapezoid (residing to the left of the cell boundary) is assigned to grid point j, since it lies in the j-th cell. All of the fluid elements are treated in the same way and independently. Thus, a portion of the fluid element trapezoid which extended from $j - 1$ to j initially also gets assigned to grid point j. In a similar way the cell $j + 1$ also gets some of the fluid originating between grid points $j + 1$ and $j + 2$.

The complete transport prescription which relates $\{\rho_j^{n+1}\}$ to $\{\rho_j^n\}$ is

$$\begin{aligned} \rho_j^{n+1} &= \tfrac{1}{2} Q_-^2(\rho_{j-1}^n - \rho_j^n) + \tfrac{1}{2} Q_+^2(\rho_{j+1}^n - \rho_j^n) + (Q_+ + Q_-)\, \rho_j^n, \\ Q_\pm &= \left(\tfrac{1}{2} \mp v_j^{1/2} \frac{\delta t}{\delta x}\right) \Big/ \left[1 \pm (v_{j\pm1}^{1/2} - v_j^{1/2}) \frac{\delta t}{\delta x}\right]. \end{aligned} \tag{7}$$

The $\{\tilde{v}_j^{1/2}\}$ from Eqs. (6) may be used in $Q_\pm$ for greater accuracy.

For a uniform velocity field this reduces to a simpler form,

$$\rho_j^{n+1} = \rho_j^n - \frac{\epsilon}{2}(\rho_{j+1}^n - \rho_{j-1}^n) + \left(\frac{1}{8} + \frac{\epsilon^2}{2}\right)(\rho_{j+1}^n - 2\rho_j^n + \rho_{j-1}^n). \tag{8}$$

This latter formula is a simple two-sided differencing of the convection term plus a strong diffusion. This strong diffusion is the basis of FCT. Without the velocity-independent diffusion, Eq. (8) is identical to the result obtained using the two-step Lax–Wendroff algorithm.

The treatment at the boundaries depends entirely on the physical problem being simulated and will not be considered in detail here since the geometric and numerical implementation of any particular boundary condition is a straightforward generalization of the geometric interpretation shown in Fig. 1.

This SHASTA transport stage is conservative and nonnegative but has a very large, zeroth-order diffusion associated with it, as well as the usual second-order dispersion and velocity-dependent diffusion. This strong diffusion amounts to operating on the initial density profile $\{\rho_j^0\}$ in the following way:

$$\rho_j^1 = \rho_j^0 + \eta(\rho_{j+1}^0 - 2\rho_j^0 + \rho_{j-1}^0). \tag{9}$$

In the zero-velocity case the diffusion coefficient η is strictly $\eta = 0.125$. In the case of nonzero velocities η is roughly 0.125 plus small velocity- and wavenumber-dependent terms.

Consider the slightly more general problem where $\epsilon \equiv v\,\delta t/\delta x$ is constant, greater than zero, but less than 0.5. Let

$$\rho_j^0 \equiv e^{ikj\delta x}, \tag{10}$$

where the cell index j is to be distinguished from $i \equiv \sqrt{(-1)}$, k is the wavenumber, and the superscript is the timestep number. This initial condition corresponds to the physical solution

$$\rho(x, t) = e^{ikx}e^{-ikv(t-t_0)}, \tag{11}$$

an infinite wave propagating to the right. The transport prescription Eq. (8) gives rise to an amplification coefficient for one cycle

$$\rho_j^1/\rho_j^0 = \{1 - (\tfrac{1}{4} + \epsilon^2)(1 - \cos k\,\delta x) - i\epsilon \sin k\,\delta x\}. \tag{12}$$

The two velocity-dependent terms, proportional to ϵ^2 and ϵ, describes the phase propagation of the wave and include small velocity- and wavenumber-dependent errors in both phase and amplitude of the wave. Without the sin $k\,\delta x$ term, Eq. (12) looks like a pure three-point diffusion equation. Some of the $\epsilon^2/2$ term is involved in the wave phase change so the actual diffusion in Eq. (12) has a smaller coefficient than $(1/8 + \epsilon^2/2)$. Furthermore, the velocity-dependent part is also wavenumber dependent.

Using Eq. (12) we find the amplification factor:

$$[\rho_j^1/\rho_j^0]^2 = [1 - \tfrac{1}{4}(1 - \cos k\,\delta x)]^2 - (\epsilon^2/2)(1 - 2\epsilon^2)(1 - \cos k\,\delta x)^2, \tag{13}$$

which is always less than unity for $|\epsilon| < \frac{1}{2}$ (the numerical wave always decays in time and hence the transport scheme is stable).

The analysis of the transport stage of SHASTA so far has considered only the diffusion terms. The phase error in Stage I is also a function of wavenumber $k\,\delta x$ [here identical with the parameter δ defined by Eq. (2)] and velocity ϵ. Define $x_\phi(\epsilon, k)$ to be the position where the imaginary part of ρ_j^1 goes to zero. x_ϕ is zero at $t = 0$ for both the exact solution (11) and the numerical solution (10). Thus at $t = \delta t$ the deviation of $x_\phi(\epsilon, k)$ from the correct value $\epsilon\,\delta x$ measures the phase error as a function of ϵ and k. Setting $\mathrm{Im}(\rho_j^1) = 0$ in Eq. (12) with $x_\phi = j\,\delta x$ gives

$$\tan kx_\phi = \epsilon \sin k\,\delta x/[1 - (\tfrac{1}{4} + \epsilon^2)(1 - \cos k\,\delta x)]. \tag{14}$$

For long wavelengths this gives the relative phase error

$$(x_\phi - \epsilon\,\delta x)/\epsilon\,\delta x \simeq k^2\,\delta x^2[(\epsilon^2/6) - (1/24)] + O(k^4\,\delta x^4). \tag{15}$$

Equation (15) shows that the phases are second-order accurate and that the phase errors become fourth order when $|\epsilon| = \frac{1}{2}$, the maximum allowable velocity here as discussed earlier in connection with Eq. (4). The relative phase error for SHASTA, given by Eq. (15) for long and intermediate wavelengths, is four times smaller than for the standard algorithms shown in Tables I, II, and III. Equation (14) also shows that the numerical phase velocity is smaller than the actual transport velocity for all wavelengths. The very shortest wavelength, $k\,\delta x = \pi$, does not propagate at all according to Eq. (14). Section III of this paper compares these phase errors with those of other widely used methods for solving the continuity equation [4].

II. The Antidiffusion Stage (Stage II)

As seen in the previous section, Stage I of the FCT algorithm SHASTA has quite small phase errors for long and intermediate wavelengths but has a large diffusion which is only weakly velocity dependent. In particular, for zero velocity Stage I reduces to the diffusion equation

$$\rho_j^1 = \rho_j^0 + \tfrac{1}{8}(\rho_{j+1}^0 - 2\rho_j^0 + \rho_{j-1}^0). \tag{16}$$

Removal of this erroneous diffusion by applying an equal and opposite antidiffusion immediately suggests itself. Equation (16) can be inverted in one dimension and solved, since the resulting system is tridiagonal. Thus we could take the result of Stage I $\{\rho_j^1\}$ and find a corrected density $\{\bar{\rho}_j^1\}$ by solving the implicit equation [10]

$$\rho_j^1 = \bar{\rho}_j^1 + \tfrac{1}{8}[\bar{\rho}_{j+1}^1 - 2\bar{\rho}_j^1 + \bar{\rho}_{j-1}^1]. \tag{17}$$

This would effectively remove the factor of 1/4 from Eq. (12) in the $\epsilon = 0$ limit. Appendix A treats the topic of implicit antidiffusion for FCT algorithms. The remainder of this paper concentrates on explicit antidiffusion as providing the simplest form of FCT.

Thus we use the following explicit antidiffusion equation in order to remove the net diffusion of Eqs. (7):

$$\bar{\rho}_j^{\,1} = \rho_j^{\,1} - \tfrac{1}{8}(\rho_{j+1}^1 - 2\rho_j^{\,1} + \rho_{j-1}^1). \tag{18}$$

Notice that this antidiffusion step adds only a real multiplicative factor to Eq. (12) and hence does not disturb the phases. Thus after Fourier decomposition as before,

$$\bar{\rho}_j^{\,1} = [1 + \tfrac{1}{4}(1 - \cos k\, \delta x)]\, \rho_j^{\,1}. \tag{19}$$

The amplitude of the modified $\{\bar{\rho}_j^{\,1}\}$ in terms of the initial $\{\rho_j^{\,0}\}$ is

$$\begin{aligned} |\, \bar{\rho}_j^{\,1}/\rho_j^{\,0}\,|^2 = {} & [1 - \tfrac{1}{16}(1 - \cos k\, \delta x)^2]^2 \\ & - (\epsilon^2/2)(1 - 2\epsilon^2)(1 - \cos k\, \delta x)^2\, [1 + \tfrac{1}{4}(1 - \cos k\, \delta x)]^2. \end{aligned} \tag{20}$$

The velocity-dependent term in Eq. (20) is small but a small velocity-independent "residual diffusion" remains.

The antidiffusion, as given in Eq. (18), is certainly not positive. The simple example of Fig. 2 shows why this is so. The antidiffusion of Stage II, which is only intended to remove numerical errors introduced in Stage I, in fact introduces additional numerical errors at grid points j and $j + 1$ in the figure. New maxima and minima are formed where they are physically unreasonable. The new minimum is actually negative.

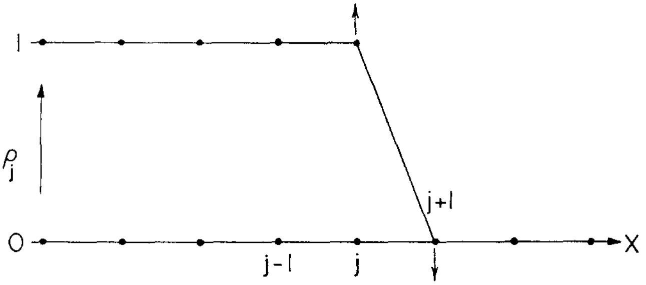

Fig. 2. Showing the non-sign-preserving tendencies of antidiffusion. At gridpoint j a new maximum will be created by antidiffusion and the new minimum at $j + 1$ will be negative.

In removing the problem of nonpositivity it is useful to work with the mass fluxes directly. The antidiffusion formula, Eq. (18), can be rewritten

$$\bar{\rho}_j^{\,1} = \rho_j^{\,1} - f_{j+1/2} + f_{j-1/2}\,, \tag{21}$$

where we have defined the antidiffusive mass fluxes as

$$f_{j\pm1/2} \equiv \pm\tfrac{1}{8}(\rho^1_{j\pm1} - \rho_j^1). \tag{22}$$

Equation (21) has two important properties:

(1) The antidiffusive fluxes $f_{j+1/2}$ and $f_{j-1/2}$ describe explicit transfers of material.

(2) Equation (21) is strictly conservative regardless of the values of the fluxes $\{f_{j+1/2}\}$ because every flux is added once and subtracted once somewhere else (with the obvious exception of boundary points).

The following qualitative limitation on the antidiffusive mass fluxes suggests itself in order that the antidiffusion be nonnegative:

> *The antidiffusion stage should generate no new maxima or minima in the solution, nor should it accentuate already existing extrema.*

Such a prescription obviously maintains positivity. What is not so obvious is how to make the limitation quantitative in a simple way without violating conservation.

This is done by correcting the antidiffusive mass fluxes. The $\{f_{j+1/2}\}$ are limited term by term so that *no antidiffusive-flux transfer of mass can push the density value at any grid point beyond the density value at neighboring points.* This is the origin of the name "flux-corrected transport" and the crux of the method.

The corrected fluxes $\{f^c_{j+1/2}\}$ are given by the formula

$$f^c_{j+1/2} = \operatorname{sgn} \Delta_{j+1/2} \max\{0, \min(\Delta_{j-1/2} \operatorname{sgn} \Delta_{j+1/2}, \text{``}\tfrac{1}{8}\text{''}\, |\Delta_{j+1/2}|, \Delta_{j+3/2} \operatorname{sgn} \Delta_{j+1/2})\}, \tag{23}$$

where

$$\Delta_{j+1/2} \equiv \rho^1_{j+1} - \rho_j^1, \tag{24}$$

and replace the fluxes $\{f_{j+1/2}\}$ in Eq. (21). By means of a few trials, the reader can readily convince himself that this is the quantitative form of the qualitative prescription given above.

The factor $\frac{1}{8}$ of Eq. (18) has been replaced by "$\frac{1}{8}$". The quotation marks indicate that more exact cancellation of errors can be achieved if one expends a small amount of computational effort by including at least rough approximations to the velocity- and wavenumber-dependent corrections [11].

Other prescriptions are possible besides that of Eq. (23), which we term "strong flux correction." These modifications are to be dealt with in a future article as their detailed discussion is more complicated than this introductory article warrants.

In the next section we compare this particular FCT algorithm with other more-or-less standard schemes. These comparisons are made on the simplifying basis that "$\frac{1}{8}$" $= 0.125$.

III. The Square-Wave Test Problems

This section compares the one-dimensional FCT algorithm given in the preceding section with three contemporary, explicit Eulerian, finite-difference schemes. They are the "one-sided" first-order scheme [12], the second-order Lax–Wendroff two-step scheme, and the second-order "leapfrog" scheme. Tables I–III show the amplification factors, phase behavior, and long-wavelength relative phase error of each algorithm applied to the Fourier harmonic with wavenumber $k \equiv 2\pi/\lambda$. The correct theoretical result is also shown for comparison. In this section we assume, as earlier, that v is spatially and temporally constant and define $\epsilon \equiv v\,\delta t/\delta x$. Then the four algorithms and the exact analytic formula are linear operations except for the flux-correction part of Stage II. Each harmonic, as before, can be treated independently and should be a travelling wave of constant amplitude moving with velocity v (independent of k). Finite-difference algorithms in general cause each harmonic to travel at the wrong velocity and cause the amplitude to vary slowly in time.

TABLE I

Comparing the amplification factor squared for four difference schemes and theory. Only the leapfrog method, of those shown, has no damping.

Algorithm	Order	Square of the amplification factor[a]	
Theory	∞	1	
One-Sided	1	$1 - 2\,\lvert\epsilon\rvert\,(1 - \lvert\epsilon\rvert)\,(1 - \cos k\delta x)$	strong damping
Lax–Wendroff	2	$1 - (\epsilon^2 - \epsilon^4)(1 - \cos k\delta x)^2$	weak damping
Leapfrog	2	1 (exact result)	no damping
SHASTA (FCT)	"2"	$[1 - \frac{1}{16}(1 - \cos k\delta x)^2]^2$ $-(\epsilon^2/2)(1 - 2\epsilon^2)(1 - \cos k\delta x)^2\ [1 + \frac{1}{4}(1 - \cos k\delta x)]^2$	

[a] Here $\epsilon \equiv v\delta t/\delta x$.

Consider first Table I for the harmonic amplitudes. The amplification factor is the relative change in the amplitude of the Fourier harmonic during one computational cycle of the given algorithm. Theoretically the amplitude should remain constant for all time. The second-order leapfrog method alone achieves this result—not a great surprise since the leapfrog method is reversible and also gives undamped solutions to wave equations. The other three algorithms give damped solutions with the shorter wavelengths decaying most rapidly. The one-sided method has particularly strong diffusion. The Lax–Wendroff two-step method has much lower intrinsic damping, of order ϵ^2 and $k^4\,\delta x^4$, but clearly leapfrog is best by this criterion. In practical usage, however, both the Lax–Wendroff and leapfrog algorithms require a strong (perhaps local) diffusive smoothing term to keep dispersively generated ripples from pulling the density negative near even moderate gradients and destroying the qualitative profile (see Fig. 3). Thus both Lax–Wendroff and leapfrog are more diffusive in practice than the new algorithm, which guarantees nonnegative values in addition because of the flux-correcting procedure. When velocity- and wavenumber-dependent corrections to Eq. (18) are included, or if the implicit antidiffusion is used, the residual diffusion of the new algorithm can be made still smaller.

TABLE II

The Single-Cycle Phase Shift for Four Difference Schemes and Theory[a]

Algorithm	Order	Phase shift x_Φ in one cycle[b]
Theory	∞	$kx_\Phi = k\epsilon\delta x \; [x_\Phi = v\delta t]$
One-Sided	1	$\tan kx_\Phi = \dfrac{\epsilon \sin k\delta x}{[1 - \lvert\epsilon\rvert(1 - \cos k\delta x)]}$
Lax–Wendroff	2	$\tan kx_\Phi = \dfrac{\epsilon \sin k\delta x}{[1 - \epsilon^2(1 - \cos k\delta x)]}$
Leapfrog	2	$\tan kx_\Phi = \dfrac{[1 - (1 - (\epsilon^2/2)\sin^2 k\delta x)^2]^{1/2}}{(1 - (\epsilon^2/2)\sin^2 k\delta x)}$
SHASTA (FCT)	"2"	$\tan kx_\Phi = \dfrac{\epsilon \sin k\delta x}{[1 - (1/4 + \epsilon^2)(1 - \cos k\delta x)]}$

[a] The expressions are valid for all wavelengths with fixed velocity.
[b] Here $\epsilon \equiv v\delta t/\delta x$.

Table II shows phase behavior as a function of ϵ and k for the exact solution and for the four algorithms; x_ϕ is the position of the $\mathrm{Im}(\rho) = 0$ point at the end of one cycle, assuming $\mathrm{Im}(\rho) = 0$ at $x = 0$ initially. The theoretical result is $x_\phi = v\,\delta t$, independent of k. For long wavelengths, all expressions have the correct limit, as is to be expected. When $k\,\delta x$ equals π (2 cells per wavelength), the phase velocity vanishes for all four algorithms. As an example, consider short wavelengths for which $k\,\delta x = \pi/2$ (four-celled modes) and take $\epsilon = \frac{1}{2}$. Then

$$\begin{aligned} \tan kx_\phi &= 1, && \text{(theory)} \\ \tan kx_\phi &= 1, && \text{(one-sided)} \\ \tan kx_\phi &= 2/3, && \text{(Lax–Wendroff)} \\ \tan kx_\phi &= \sqrt{(15)}/7, && \text{(leapfrog)} \\ \tan kx_\phi &= 1. && \text{(SHASTA)} \end{aligned}$$

SHASTA and the one-sided algorithm give the correct result in this special case. For shorter wavelengths all algorithms give basically nonsensical results.

TABLE III

The Long- and Intermediate-Wavelength Relative Phase Errors are Compared

Algorithm	Order	Relative phase error[b]
Theory	∞	$\dfrac{x_\phi - v\delta t}{v\delta t} = 0$ (no error)
One-Sided	1	$\dfrac{x_\phi - v\delta t}{v\delta t} = -\left(\dfrac{1}{6} - \dfrac{\mid\epsilon\mid}{2} + \dfrac{\epsilon^2}{3}\right) k^2\delta x^2 + O(k^4\delta x^4)$
Lax–Wendroff	2	$\dfrac{x_\phi - v\delta t}{v\delta t} = -\left(\dfrac{1}{6} - \dfrac{\epsilon^2}{6}\right) k^2\delta x^2 + O(k^4\delta x^4)$
Leapfrog	2	$\dfrac{x_\phi - v\delta t}{v\delta t} = -\left(\dfrac{1}{6} - \dfrac{\epsilon^2}{24}\right) k^2\delta x^2 + O(k^4\delta x^4)$
SHASTA (FCT)	"2"	$\dfrac{x_\phi - v\delta t}{v\delta t} = -\left(\dfrac{1}{24} - \dfrac{\epsilon^2}{6}\right) k^2\delta x^2 + O(k^4\delta x^4)$

[a] The FCT algorithm has a four-times smaller error, for small velocities, than the other methods.
[b] Here $\epsilon \equiv v\delta t/\delta x$.

Table III shows the relative phase error for one cycle for the four algorithms. The relative phase error is defined to be $(x_\phi - v\,\delta t)/v\,\delta t$. All four algorithms have phase errors quadratic in $k\,\delta x$. For most problems of interest the ϵ^2 terms can be neglected since $\epsilon^2 \ll 1$. At long wavelengths the one-sided scheme has a term proportional to $\mid \epsilon \mid$ which competes with the 1/6 term when ϵ approaches 1/2. This competition effectively reduces the relative phase error only when ϵ is large.

Lax–Wendroff and leapfrog have comparable relative phase errors at long and intermediate wavelength. The last line of Table III shows that the relative phase errors of the FCT algorithm are typically four times smaller than those of the other methods for small velocity at long and intermediate wavelengths. This result is very important.

Since the one-sided, first-order algorithm has acceptable phase properties even though it is strongly diffusive, one can apply a velocity-dependent antidiffusion with flux correction to reduce the diffusion problem. This has been done in tests, with good results. A strong diffusion can also be added to standard second-order transport algorithms and then removed using the FCT prescription. This generalization, which will be investigated in detail in a future paper, is being tested on many transport algorithms beside SHASTA. The SHASTA-FCT algorithm is the best of the algorithms tested so far and therefore forms the basis of this paper.

In summary, these tables show that the SHASTA-FCT algorithm is also superior to the other methods in regions of x where the solutions are smooth and the flux-corrections unnecessary. The phase behavior of the FCT algorithm is considerably better than for standard methods and the flux-limiting correction of Stage II also gives FCT excellent properties near sharp gradients.

To these results should be added the conclusions of K. W. Morton, who has compared seven different algorithms in somewhat greater detail than we have attempted here [4]. These seven algorithms include most of the basic methods in common use today. In addition to the three basic methods discussed above, he includes treatments of the Crowley fourth-order method [13], the Crank–Nicholson method [10], the Fromm extension of Lax–Wendroff [14], and the Roberts–Weiss fourth-order method [5]. His paper reaches two major conclusions:

(1) The leapfrog algorithm is the best of the generally applicable basic methods. [FCT was not known at that time.]

(2) Implicit and semi-implicit algorithms of the Crank–Nicholson type should be restricted to use in diffusion equations where an explicit time differencing would result in a prohibitively small timestep to ensure numerical stability.

Computer calculations confirm the conclusions of our analysis given above. Tests were performed on density square waves traveling with constant velocity. All four algorithms of Tables I–III were tested with identical initial conditions and identical timestep. The initial condition is identical with Fig. 11(b); a comparison of the four algorithms at two later times is shown in Fig. 3. The square wave is initially 20 cells wide with height 2.0. The background density is 0.5 and constant throughout the rest of the system. The system is 100 cells long with periodic boundary conditions. The velocity was chosen so that $v\,\delta t/\delta x = 0.2$ for all cases. The analytic solution is shown as the solid line. The Lax–Wendroff test was performed with a small additional diffusion to keep the solution from going negative. The leapfrog case was run with the same level of diffusion and looks appreciably better because the undershoots and overshoots are smaller.

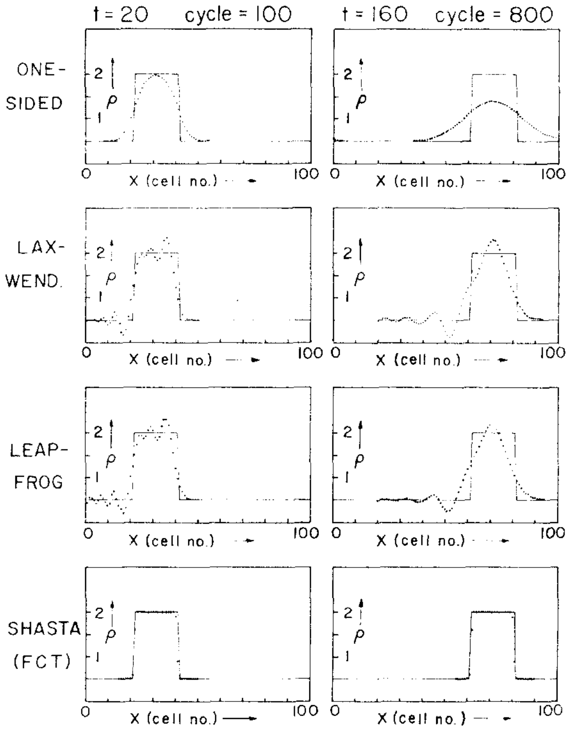

FIG. 3. Square-wave test comparisons at two times during the calculation. The solid line is the analytic solution, the dots are computed values. Computed values at the background level of 0.5 are not all plotted.

No undershoots are visible in the one-sided calculation because of the massive diffusion. The SHASTA one-dimensional FCT calculation shows remarkably good agreement with the exact solution when the k-dependent corrections in the factor "$\frac{1}{8}$" are included approximately [11]. Figure 4 shows a comparison of the SHASTA subroutine results for both the "$\frac{1}{8}$" corrected calculation and the "$\frac{1}{8}$" = 0.125 calculation. The correction enhances the antidiffusion coefficient in regions where short wavelengths predominate. The uncorrected calculation has no overshoots or undershoots but residual fourth-order diffusion rounds the corners of the square wave somewhat. Even this result, however, is obviously qualitatively far superior to the three basic methods shown in Fig. 3.

Figure 5 shows a further test of the 1D FCT algorithm. In this test the continuity equation is solved analytically for a time- and space-dependent density of the form

$$\rho(x, t) = 1 + \tfrac{1}{3}\sin[k_1(v - v_0 t)] + (2t/3t_{\max})\sin(k_2 x). \tag{25}$$

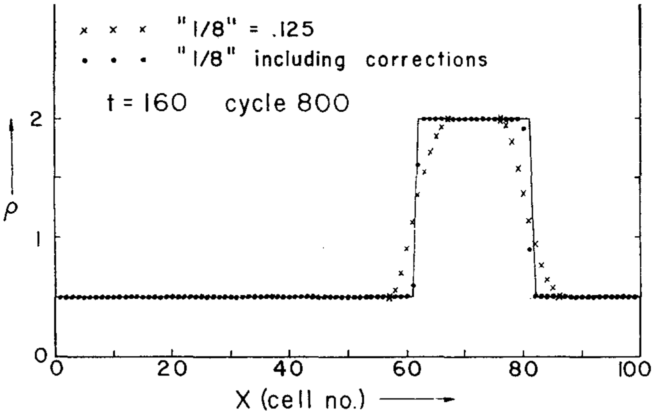

FIG. 4. Comparison of the 1 D FCT algorithm SHASTA with and without k- and v-dependent corrections to the antidiffusion coefficient. Notice the improvement obtained even by the simple formulation "$\frac{1}{8}$" = 0.125 in comparison with the standard schemes of Fig. 3.

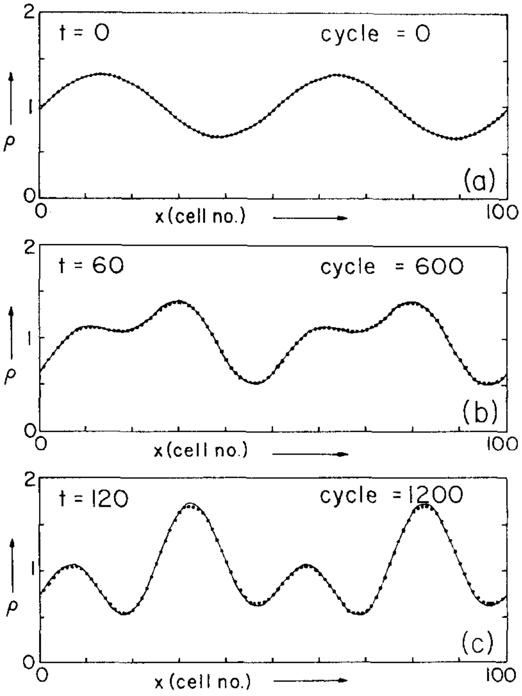

FIG. 5. A test of SHASTA on a problem where new maxima and minima appear physically. Mode 2 is excited initially using an analytic time- and space-dependent velocity and Mode 4 grows with time. The weak clipping phenomenon at maxima and minima is a result of the simple flux-limiting correction of Stage II.

The corresponding velocity field is

$$v(x, t) = (1/\rho)[(1/3)\, v_0\{\sin[k_1(x - v_0 t)] + \sin(k_1 v_0 t)\} \\ + (2/3)(1/k_2 t_{\max})\{\cos(k_2 x) - 1\}], \tag{26}$$

where v_0, k_1, k_2, and $t_{\max}$ are constants and where $\rho(x, t)$ is given by Eq. (25). This exact velocity was used in the SHASTA subroutine to solve the continuity equation numerically for the density. Analytic and computed solutions for the density are plotted for comparison at three times. This problem has a growing sinusoidal density component and a traveling sinusoidal component. Their interaction creates a situation in which new maxima and minima are continually forming and moving physically. It shows that flux limiting, as performed in Stage II of the algorithm, does not preclude the appearance and disappearance of extrema in the solution when they occur physically. The deviations of ~1 % at the sharp maxima and minima of the solution are the result of a "clipping" phenomenon caused by strong flux correction.

IV. The Shock Test Problems

The motivation for developing the FCT algorithms lies in the desire to study

To this end we now consider a one-dimensional, ideal inviscid hydrodynamic system, described self-consistently by the following equations:

$$(\partial\rho/\partial t) + (\partial/\partial x)(\rho v) = 0; \tag{27}$$

$$(\partial/\partial t)(\rho v) + (\partial/\partial x)(\rho v^2) = -(\partial P/\partial x); \tag{28}$$

$$(\partial E/\partial t) + (\partial/\partial x)(v^* E) = 0; \tag{29}$$

where

$$v^* = (1 + P/E)v, \tag{30}$$

$$(\partial T/\partial t) + v(\partial T/\partial x) = (\gamma - 1)\, T(\partial v/\partial x). \tag{31}$$

Here

$$P = \rho T \tag{32}$$

(T has units of velocity squared), and

$$E = P/(\gamma - 1) + (1/2)\,\rho v^2. \tag{33}$$

These equations with $\gamma = 5/3$ were modeled using the algorithm SHASTA, as described in Appendix B. Equation (31) for T is invalid near shocks, as it contains adiabatic heating only, and so was not used. The basic test of the algorithm consists of propagating a shock according to Eqs. (27)–(33), studying the structure of the shock computed, and verifying that the Rankine–Hugoniot relations hold. In the absence of an explicit viscosity, shocks should steepen until the shock front is only one grid space thick, the best numerical resolution possible.

TABLE IV

Density Profiles at and near the Shock Front for March 5 and March 20 Shocks[a]

DENSITY					
	$t = 5.0$			$t = 20.0$	
Grid pt	Mach 5 cycle 100	Mach 20 cycle 500	Grid pt	Mach 5 cycle 400	Mach 20 cycle 2000
left b.c.	3.5714	3.9702	left b.c.	3.5714	3.9702
. . .	. . .	. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .	. . .	. . .
31	3.5701	3.9747	. . .	. . .	. . .
32	3.5701	3.9747	. . .	. . .	. . .
33	3.5701	3.9753	. . .	. . .	. . .
34	3.5700	3.9766	. . .	. . .	. . .
35 ↔	3.4273	3.9766	. . .	. .	. . .
36	1.1566	1.0600	. . .	. . .	. . .
37	0.9995	0.9988	76	3.5713	3.9700
38	0.9995	0.9988	77	3.5712	3.9702
39	0.9999	0.9999	78	3.5712	3.9707
40	1.0000	1.0001	79	3.5712	3.9707
41	1.0000	1.0000	80	3.4038	3.8200
. . .	. . .	. . .	81 ↔	1.1538	0.9978
. . .	. . .	. . .	82	0.9995	0.9978
. . .	. . .	. . .	83	0.9995	0.9982
. . .	. . .	. . .	84	0.9999	1.0001
. . .	. . .	. . .	85	1.0000	1.0001
. . .	. . .	. . .	86	1.0000	1.0000
. . .	. . .	. . .	. . .	. . .	. . .
. . .	. . .	.	. . .	. . .	. . .
. . .	. . .	. . .	. . .	. . .	. . .
right b.c.	1.0000	1.0000	right b.c.	1.0000	1.0000

[a] Arrows indicate positions to which the fronts should have propagated according to theory at $t = 5.0$ and $t = 20.0$. For both examples initial position $x_0 = 20.5$ and velocity of propagation $v_0 = 3.0$. Profiles to the left and right of the transition region are essentially flat and hence are not shown.

Initial conditions are set by specifying ρ_1, v_1, and T_1 on the right boundary, choosing a Mach number M_1, and then calculating the downstream quantities ρ_2, v_2, and T_2 from the Rankine–Hugoniot relations across a shock. A small velocity v_0 is added to v to make the shock travel relative to the fixed grid. The initial shock position is chosen somewhere in the grid, then all physical variables are set to values ρ_1, v_1, T_1, or ρ_2, v_2, T_2 depending on whether the grid point is upstream or downstream from the initial shock position. The boundary conditions consist of holding ρ, v, and T fixed to these upstream and downstream values at the ends of the system. The energy E and pressure p are chosen to be

TABLE V

Velocity Profiles at and near the Shock Front for Mach 5 and Mach 20 Shocks[a]

		VELOCITY			
	$t = 5.0$			$t = 20.0$	
	Mach 5	Mach 20		Mach 5	Mach 20
Grid *pt*	cycle 100	cycle 500	Grid *pt*	cycle 400	cycle 2000
left b.c.	1.1926	−3.5034	left b.c.	1.1926	−3.5034
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
31	1.1946	−3.5011	...	...	...
32	1.1946	−3.4970	...	...	...
33	1.1946	−3.4959	...	...	...
34	1.1946	−3.4948	...	...	...
35	1.1760	−3.6782	...	...	...
36 ↔	−2.8136	−2.1524	...	...	...
37	−3.4742	−2.2841	76	1.1926	−3.5077
38	−3.4568	−2.2827	77	1.1917	−3.5034
39	−3.4551	−2.2815	78	1.1918	−3.5022
40	−3.4548	−2.2814	79	1.1918	−3.5022
41	−3.4549	−2.2815	80	1.1835	−4.0304
...	...	...	81 ↔	−2.8391	−2.2847
...	...	...	82	−3.4742	−2.2849
...	...	...	83	−3.4568	−2.2839
...	...	...	84	−3.4551	−2.2817
...	...	...	85	−3.4548	−2.2818
...	...	...	86	−3.4548	−2.2819
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
right b.c.	−3.4550	−2.2820	right b.c.	−3.4550	−2.2820

[a] Initial conditions and format are those of Table IV.

TABLE VI

Pressure Profiles at and near the Shock Front for Mach 5 and Mach 20 Shocks[a]

PRESSURE					
$t = 5.0$			$t = 20.0$		
Grid *pt*	Mach 5 cycle 100	Mach 20 cycle 500	Grid *pt*	Mach 5 cycle 400	Mach 20 cycle 2000
left b.c.	31.000	499.75	left b.c.	31.000	499.75
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
31	30.966	499.25	...	...	...
32	30.966	499.32	...	...	...
33	30.966	499.45	...	...	...
34	30.965	499.58	...	...	...
35	30.700	497.83	...	...	...
36 ↔	2.491	14.46	...	...	...
37	0.957	0.57	76	30.992	499.29
38	0.997	0.90	77	30.991	499.37
39	1.000	1.01	78	30.990	499.48
40	1.000	0.98	79	30.988	499.48
41	1.000	0.98	80	30.614	472.86
...	...	...	81 ↔	2.096	0.67
...	...	...	82	0.957	0.65
...	...	...	83	0.997	0.88
...	...	...	84	1.000	1.01
...	...	...	85	1.000	1.00
...	...	...	86	1.000	0.99
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
right b.c.	1.000	1.00	right b.c.	1.000	1.00

[a] Initial conditions and format are those of the preceding tables. Note that for the Mach 20 case, the pressure jumps by a factor of 500.

consistent with ρ_1, v_1, T_1, or ρ_2, v_2, T_2 in the initial condition and at the boundaries. The problem becomes nonphysical as soon as the shock reaches one of the boundaries because of the fixed-boundary conditions.

Tables IV, V, and VI show the density, velocity, and pressure jumps, respectively, in the neighborhood of two shocks, a Mach 5 shock and a Mach 20 shock. The grid-point numbers are the corresponding x values because $\delta x = 1$. Each shock has an added velocity $v_0 = 3.0$ so that at time $t = 5.0$ they should have propagated

15 cells from the initial position $x = 20.5$, and at time $t = 20.0$ they should have propagated 60 cells. These exact shock positions are labeled by arrows (↔) in the tables. The right and left boundary values are also given for comparison. These are the exact Rankine–Hugoniot values for the shocks. The pressure exhibits a jump of roughly 500 to 1 in the case of the Mach 20 shock and has a small undershoot associated with the use of energy conservation in determining P on the upstream side of the shock. This can be removed by using Eq. (31) upstream from the shock.

Small, barely discernible ripples propagate with the sound velocity from the jump. These are excited by passage of the shock transition over the grid. Otherwise there is virtually no deviation from the correct, steep profiles. This is as it should be, since inviscid fluids, in theory, exhibit discontinuous shocks.

If we replace the discontinuous jump in the initial conditions with a ramp [Fig. 6(a)], there is little change in the results. After ten–twenty cycles the front steepens up to a one-cell thickness [Fig. 6(b)] and the initial transients from the ramp propagate away downstream. This is of course demanded on physical grounds. The sound waves initially excited by passage of the ramp over the grid die away [Figs. 6(c)–(d)] and by cycle 1000 [Fig. 6(e)] the profile is that of a steep shock wave. The only departure from the results described in Table IV, other than that implied by $v_0 = 1$, is that the shock is slightly weaker at the end of the run (the jump is reduced by about 2 %).

By contrast, shocks propagated with Lax–Wendroff and other conventional schemes using an artificial viscosity [8] exhibit a spurious viscous broadening and large numerical overshoots and undershoots can occur. Without this artificial viscosity, however, even larger and more unphysical ripples can appear, driving temperature, energy, and even density negative.

The SHASTA-FCT algorithm clearly allows propagation of shocks of minimum width (one cell transition length) across an Eulerian grid as shown in Tables IV–VI. This is accomplished without generating large fluctuations or undershoots upstream and overshoots downstream from the shock. While no explicit viscous term is used in the FCT method, the existence of a shock, it can be argued, betokens an "effective viscosity" arising out of the numerics so that shock widths always become about one cell, the best possible resolution. This effective numerical viscosity should not be confused with the linear residual diffusion shown in Eq. (20). Such a small smoothing as given by the linear analysis leading to Eq. (20) could never give decent treatment of shock-like phenomena. Rather it is the highly nonlinear effect from Eq. (23) near a shock transition which generates the effective viscosity. We feel this view tells only a partial truth, however. From another viewpoint, the completely inviscid fluid equations admit weak (discontinuous) solutions. One can also view FCT as generating these weak solutions up to the limit of grid resolution.

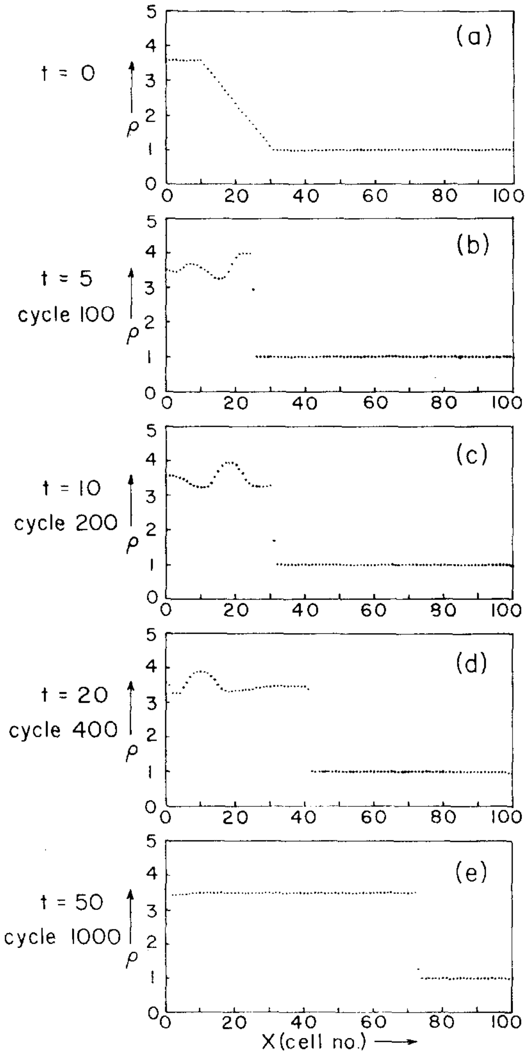

FIG. 6. Evolution of a ramp initial condition into a strong shock. The ramp transient in density, velocity, and pressure leaves sound waves propagating away on the downstream side ot the shock. The correct Rankine–Hugoniot conditions are rapidly achieved.

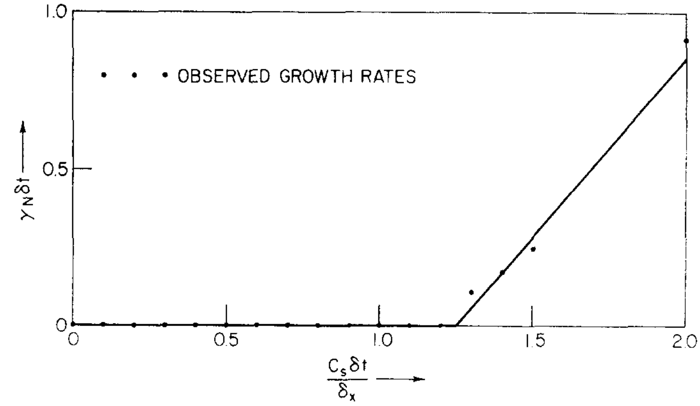

FIG. 7. Growth rate of numerical instability, γ_N, versus Courant number, $C_s\delta t/\delta x$, for the SHASTA-FCT algorithm. Flux limiting plus the relatively nonlocal aspects of the overall algorithm (7 points are involved) permit stable calculation with Courant numbers larger than unity. To obtain the data of this figure δ-function perturbations were applied to a zero velocity, constant-density and -temperature fluid.

The FCT method can be driven unstable (as one would expect because it is explicit) by taking a sufficiently large timestep. Figure 7 shows a plot of numerical growth rate, $\gamma_N \delta t$, versus the Courant number $C_s \delta t/\delta x$ for numerical instability. [redacted] here. Instead we approach the problem empirically. A constant temperature, constant density, stationary fluid is perturbed by a very small velocity impulse (1 % of the sound speed C_s) in a single cell. The evolution of this system is then followed for many cycles using different numerical timesteps. The growth of the peak amplitude, for long timesteps, is thus a measure of numerical instability.

The critical timestep, which can be seen to be between $C_s\, \delta t/\delta x = 1.2$ and 1.3, is somewhat larger than found for other finite-difference algorithms. This increase in the permissible timestep occurs because flux correction strongly suppresses spurious new maxima and minima which occur first at short wavelength in numerical instabilities. If one combines this condition,

$$C_s\, \delta t/\delta x \lesssim 1.25,$$

with the condition for nonnegative densities,

$$v\, \delta t/\delta x < 0.5,$$

it can be seen that Mach .4 flows are integrated accurately with no essential restriction due to the Courant–Friedrichs–Lewy condition [15].

The final tests are modifications of the shock runs described above. First, a supersonic flow is directed across the system against a wall on the left end,

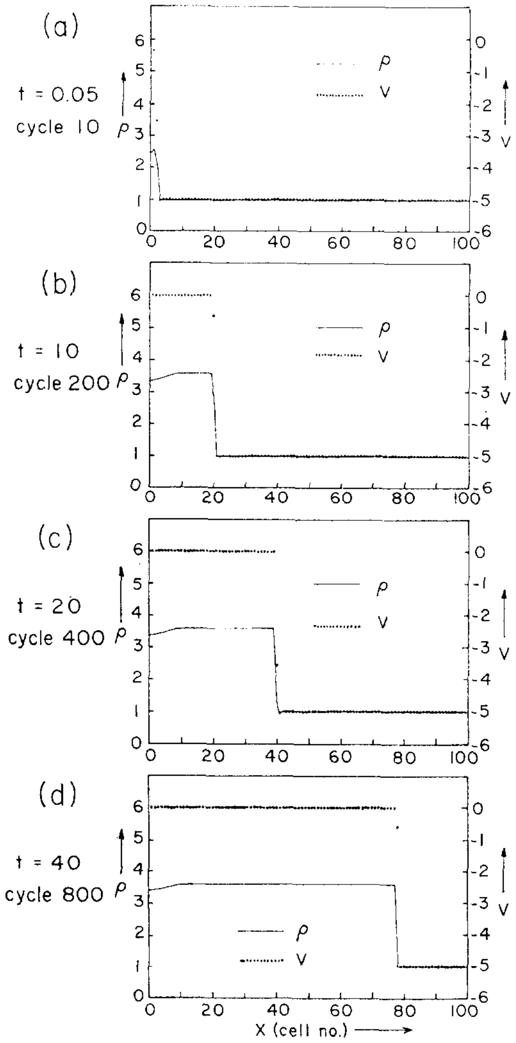

FIG. 8. Supersonic piston calculation using SHASTA-FCT. The left boundary is established as a reflecting wall. Fluid sloshes against this wall from the right. The only remnant of the initial transient (a) is a small contact discontinuity or entropy-mode perturbation against the wall (d) in which ρT = const.

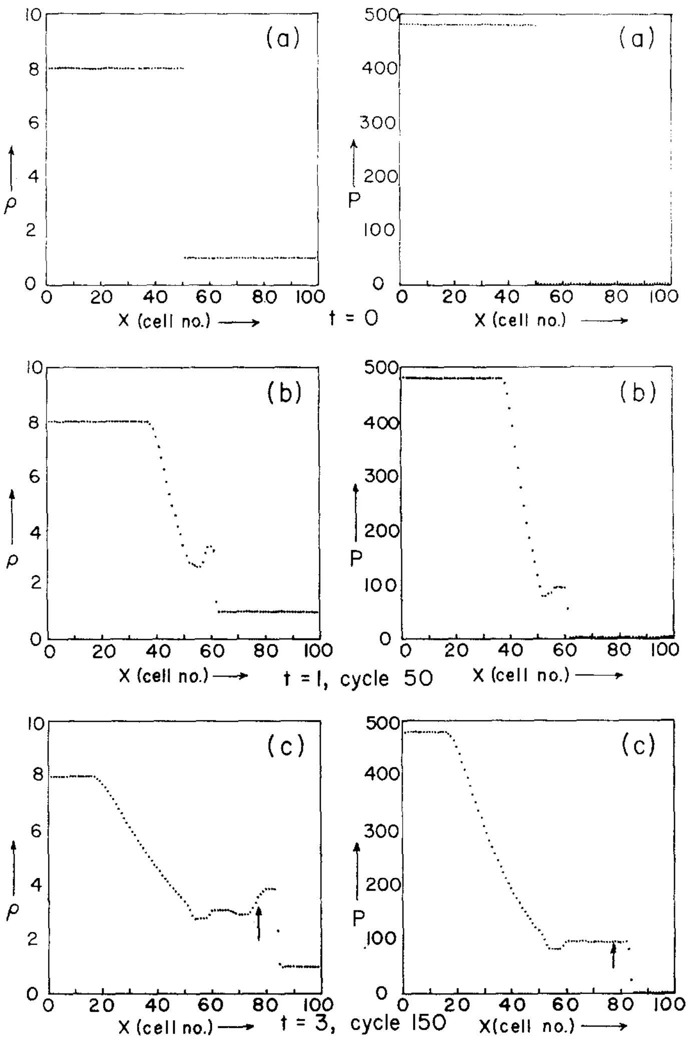

FIG. 9. Exploded diaphragm calculation using SHASTA-FCT. The formation of a right-moving shock by (c) is clearly shown. At this time ($t = 3$, cycle 150) the contact discontinuity can be seen clearly (ρ varying, p constant) near cell 78. The fluctuations in density and pressure arise primarily from the very coarse resolution of the initial transients. The shock steepens as expected.

defined by reflecting boundary conditions. The fluid stopped at the wall piles up and sloshes back [Fig. 8(a)]. The leading edge is a shock which propagates to the right with a velocity determined by the initial condition through the Rankine–Hugoniot relations. Figure 8(b)–(d) shows the density and velocity profiles moving away from the wall where a Mach 3.9 flow has been stopped. The shock has Mach number $M_1 \approx 5.3$, in excellent agreement with the theoretically predicted value of 5.35. The shock develops and steepens properly even in the absence of initialization with the correct jump conditions.

We can also initiate a shock by using initial conditions approximating the explosion of a diaphragm. At time $t = 0$ the system has uniform density $\rho = \rho_1$, zero velocity, and uniform temperature $T = T_1$ to the left of the diaphragm. A lower temperature and density, T_2 and ρ_2, are initialized to the right of the diaphragm. The results of one such calculation are shown in Fig. 9 where $T_1/T_2 = 60$, $\rho_1/\rho_2 = 8$ initially.

Disturbances propagate to the left and the right. A shock wave develops and propagates to the right followed by a rarefaction wave propagating to the left. A contact discontinuity (continuous pressure but discontinuous density and temperature) lies between the shock and the rarefaction wave and represents the vestiges of the original discontinuity at the diaphragm. This contact discontinuity, indicated by arrows in Fig. 9, moves with the local fluid velocity. It displays rounding on account of the residual explicit diffusion in SHASTA because the antidiffusion coefficient has been taken simply as 0.125. Unlike shocks, contact discontinuities have no natural tendency to steepen and are smoothed by thermal conductivity and molecular diffusion in real fluids. Steepeners [11] or implicit antidiffusion (Appendix A) would reduce the extent of the contact discontinuity at the expense of extra computation. Lapidus [16] compares several algorithms on a problem which contains a contact discontinuity and widths vary from 10 to 20 cells compared to the ~5 cell characteristic length of the FCT constant discontinuity.

The shock, observed propagating to the right in Figs. 9(b) and 9(c), once again satisfies the Rankine–Hugoniot relations appropriate to the velocity of propagation and is thus an accurate representation of the weak solutions expected to the hydrodynamical equations. The value of γ in these tests was 5/3.

V. Conclusions

This paper presents a new class of Eulerian, finite-difference algorithms for solving continuity and continuity-like equations. These flux-corrected transport (FCT) algorithms feature a second-order treatment of the convection and dilation terms which is nonnegative as well as conservative. The method has the additional advantageous properties of using a simple nonstaggered grid, of being very stable and easily modularized, and of being easily extended to two and three dimensions. The basic concepts of FCT can even be combined with other more or less standard transport algorithms to obtain vastly improved results.

This paper has concentrated on the one-dimensional version of FCT called SHASTA and has presented several test-problem calculations. FCT algorithms consist of two distinct stages, a first, or transport, stage and a second, flux-corrected antidiffusion stage. The first stage of SHASTA solves the continuity equation using a three-point formula which has an appealing geometric interpretation. Stage I of FCT algorithms in general is characterized by a large numerical error in the form of a strong diffusion. The antidiffusion of Stage II corrects these numerical errors introduced during the transport stage.

The strong diffusion of Stage I, coupled with the antidiffusion and simple flux-correcting procedure of Stage II, makes the FCT algorithms work. This flux correction makes the overall algorithm nonnegative and stable and yet reduces spurious numerical diffusion, the usual stabilizing element of most methods, to a very low level. The condition that no new maxima or minima be generated by the antidiffusion of Stage II seems to be the crucial factor. Qualitatively, the diffusion introduced in Stage I must be larger than any dispersive error. Then the local residual diffusion, taken to be the diffusion of Stage I minus the limited and hence smaller antidiffusion of Stage II, can always be large enough, in principle, to cancel this anomalous dispersion. That is, the FCT algorithms leave behind a large diffusive flux locally which is equal and opposite to the local dispersion error. Asymptotically both terms are of zero order but combine to give an accurate solution which is effectively second order.

This paper has developed the basic concepts of FCT via the one-dimensional algorithm SHASTA. Two other papers in preparation will consider extensions of FCT to other transport algorithms, to multidimensional geometries, and to complex physical problems. A simple timestep-split version using SHASTA for 2D problems and a fully 2D, optimized FCT algorithm have been developed. Results of actual calculations with these codes will be presented in these future papers, for the methods generalize easily to include MHD problems, cylindrical geometry, special treatments of axes [16], and complex multifluid problems with nonvanishing transport and coupling coefficients.

We close this paper with a few timing considerations. In the unoptimized simple 1D forms used for the tests of Section III the Lax–Wendroff method, the leapfrog method, and the one-sided method are all about the same speed, and the SHASTA subroutine is ~2.5 times slower. Additions of viscous terms for shocks, tests of negativity, and smoothing of these other algorithms in practical calculations reduces this ratio considerably. In any case, when effective resolution is taken into account, the FCT algorithms regain this factor easily. In a shock problem, for instance, comparable calculations using one of the basic methods with a von Neumann viscosity would require roughly 2–4 times more cells to achieve comparable resolution. Furthermore, addition of the viscosity is an extra expense. This required decrease in δx implies a corresponding decrease in δt. Thus 4–16 times more grid-point timesteps would have to be performed in 1D using a standard method than would be required of the FCT algorithm. This means, in practical calculations involving steep gradients, that a given error tolerance may actually be achieved by FCT in 2–8 times less computer execution time.

The savings in multidimensions are even more substantial because the basic fluid equations become more complicated. In the two-dimensional FCT fluid code, ρ, ρv_x, ρv_y, and E are integrated simultaneously and self-consistently over a rectangular region. The implicit FCT technique has been superimposed on an optimized Lax–Wendroff two-step algorithm. Preliminary timings on the code show 0.6 sec/cycle on a 40 × 40 grid. When the basic Lax–Wendroff integration is performed alone, the code still requires 0.4 sec/cycle. This added expense for implicit FCT, a cost of about 50 %, can be reduced considerably by using machine language to evaluate Eqs. (17) and (23). A rough count of operations indicates that improvements of order 2.5 can be expected. Thus FCT need add only 20 % to the running time of a fully optimized calculation.

Still more compelling arguments for the FCT approach can be derived by considering strong-shock problems. The use of a sufficiently large artificial viscosity to suppress ripples at the shock in conventional algorithms usually introduces a severe diffusive stability restriction on the timestep because the viscous diffusion dominates the flow. Thus even on grids with the same δx, the FCT algorithms may consume less computer time by taking longer timesteps—and give a much better answer in the bargain.

Appendix A

This appendix treats implicit antidiffusion in greater detail. A minor shortcoming of the explicit antidiffusion formula is that the residual velocity-independent diffusion damps short wavelengths even when the velocity of flow is zero. A stationary density structure is not exactly time independent. To alleviate this problem one expends additional computational effort and uses the implicit form of the antidiffusion formula, Eq. (17).

The implicit antidiffusion can make FCT dissipationless while still retaining the four-fold improvement in relative phase error noted in Section III. To apply implicit antidiffusion, we calculate $\{\bar{\rho}_j\}$ which satisfies

$$\bar{\rho}_j^{1} = \rho_j^{1} - \tfrac{1}{8}(\bar{\rho}_{j+1}^{1} - 2\bar{\rho}_j^{1} + \bar{\rho}_{j-1}). \tag{A1}$$

Equation (A1) involves a simple tridiagonal matrix which can be solved easily [1]. The quantity multiplied by "$\frac{1}{8}$" in Eq. (23) is then replaced by $|\Delta^1_{j+1/2}|$, where

$$\Delta^1_{j+1/2} \equiv \bar{\rho}^1_{j+1} - \bar{\rho}_j^1. \tag{A2}$$

The corrected fluxes from this modified form of Eq. (23) are then used in Eq. (21) exactly as for explicit antidiffusion.

In order to compare amplification factors for the implicit antidiffusion FCT algorithm with Eq. (20) for the explicit version, we again Fourier analyze to find [cf. Eq. (19)]

$$\bar{\rho}_j^1 = \rho_j^1/[1 - \tfrac{1}{4}(1 - \cos k\,\delta x)]. \tag{A3}$$

When combined with Eq. (12) for the convection and diffusion of the SHASTA transport stage, Eq. (A3) gives

$$\frac{\bar{\rho}_j^1}{\rho_j^0} = \frac{[1 - (1/4 + \epsilon^2)(1 - \cos k\,\delta x) - i\epsilon \sin k\,\delta x]}{[1 - (1/4)(1 - \cos k\,\delta x)}\tag{A4}$$

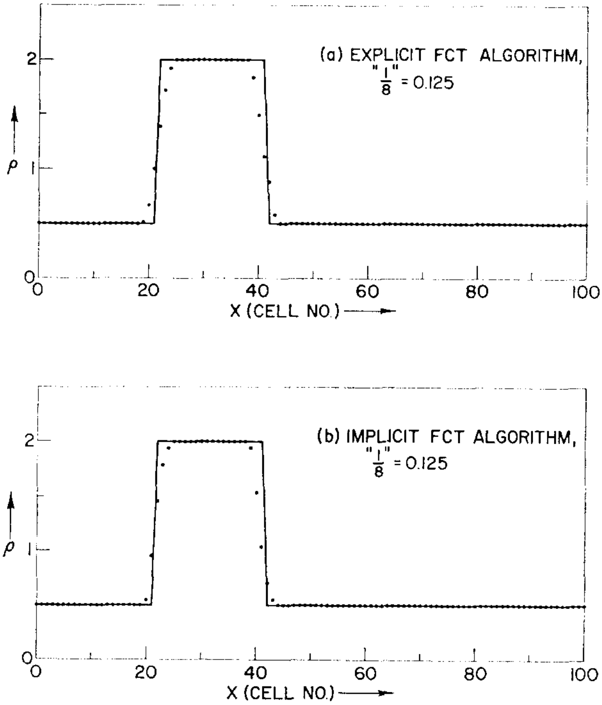

FIG. 10. Comparison of simple explicit ("$\frac{1}{8}$" = 0.125) and simple implicit versions of FCT antidiffusion using the SHASTA transport algorithm. The velocity is unity and $\epsilon = 0.2$. The time is $t = 20$ (cycle = 100). The implicit FCT antidiffusion is marginally better than the explicit version and both are far better than the standard schemes of Fig. 5.

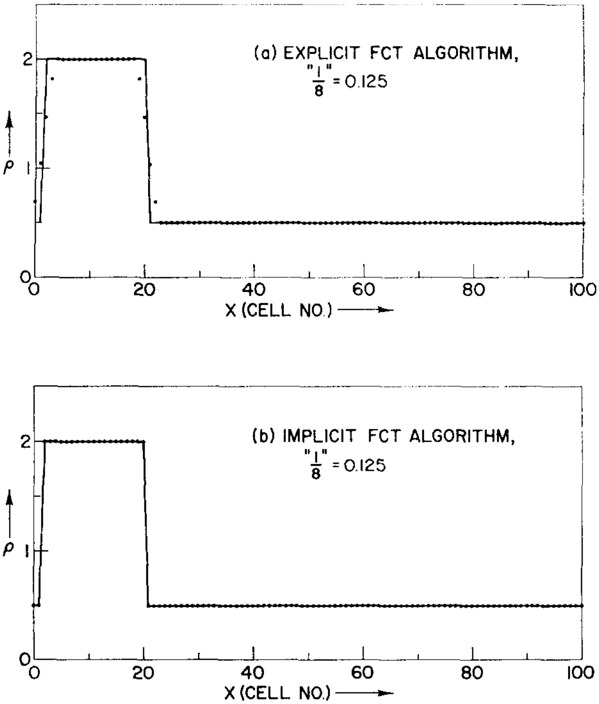

FIG. 11. Same comparison as Fig. 12 except the flow velocity is zero. The solid line again shows the correct solution. Here the implicit FCT algorithm (b) gives the exact answer whereas the somewhat simpler explicit method still has a small residual smoothing.

as the amplification factor for one cycle of the algorithm. The amplitude factor, for comparison with Eq. (20), is

$$\left|\frac{\bar{\rho}_j^{\,1}}{\rho_j^{\,0}}\right|^2 = 1 - \frac{(\epsilon^2/2)(1 - 2\epsilon^2)(1 - \cos k\,\delta x)^2}{[1 - (1/4)(1 - \cos k\,\delta x)]^2}\,. \tag{A5}$$

As the velocity ϵ goes to zero, the modes are completely undamped, as can be seen from Eq. (A5).

A residual diffusion remains when ϵ is nonzero, but is exceedingly small at reasonable wavelengths. Figure 10 shows a comparison of the explicit version and this simple implicit version of the FCT antidiffusion at time $t = 20$ (cycle 100) after the start of the square-wave problem. Both solutions show residual rounding of the square wave but no undershoots or overshoots. Figure 11 shows the same test comparison where the velocity is set to zero rather than unity. In this case the implicit antidiffusion, even with strong flux correction, gives exactly the correct stationary result while the residual diffusion in the explicit case shows up noticeably.

Even in the case of Fig. 10, where v is nonzero, the implicit treatment shows somewhat less residual rounding.

These features of the implicit FCT antidiffusion can be quite important in problems where a disturbance propagates into a stationary but non-uniform region. In the absence of the disturbance the profile in the stationary region should not change. The implicit approach achieves this requirement.

The same requirement could be achieved in the explicit approach by adding a velocity dependent diffusion rather than a velocity independent diffusion. This diffusion coefficient could be chosen just large enough to insure positivity locally and the same level of antidiffusion could be applied in Stage II. Then, when $v = 0$, the diffusion coefficient could be chosen to be zero.

Appendix B

We first compile the full set of equations defining the SHASTA-FCT algorithm in finite-difference form as used in the computational examples of Sections III. We then list the set of fluid equations as finite differenced to solve the hydrodynamic problems of Section IV.

The general continuity equation with forcing term,

$$\partial A/\partial t = -(\partial/\partial x)(AU) - \partial B/\partial x, \tag{B1}$$

is to be solved on a finite-difference mesh of cell size δx to advance $\{A_j\}$ from time t to time $t + \delta t$ using $\{U_j\}$ and $\{B_j\}$. The conservative B term will be used as the pressure to solve the momentum equation. The SHASTA-FCT algorithm is defined as follows:

$$\epsilon_j = U_j\,\delta t/\delta x, \tag{B2}$$

$$Q_{\pm}(j) = (1/2 \mp \epsilon_j)/[1 \pm (\epsilon_{j+1} - \epsilon_j)] \tag{B3}$$

$$\begin{aligned}\bar{A}_j = {}& \tfrac{1}{2}Q_+{}^2(j)[A_{j+1} - A_j] - \tfrac{1}{2}Q_-{}^2(j)[A_j - A_{j-1}] \\ & + Q_+(j)[A_j + (B_{j+1} - B_j)(\delta t/\delta x)] + Q_-(j)[A_j + (B_j - B_{j-1})(\delta t/\delta x)],\end{aligned} \tag{B4}$$

$$\varDelta_{j+1/2} = \bar{A}_{j+1} - \bar{A}_j\,, \tag{B5}$$

$$f^c_{j+1/2} = \operatorname{sgn} \varDelta_{j+1/2} \max\{0, \min[\varDelta_{j+1/2} \operatorname{sgn} \varDelta_{j+1/2}\,, \tfrac{1}{8}\,|\,\varDelta_{j+1/2}\,|, \varDelta_{j+3/2} \operatorname{sgn} \varDelta_{j+1/2}]\}, \tag{B6}$$

$$A_j^{\text{new}} = A_j - (f^c_{j+1/2} - f^c_{j-1/2}). \tag{B7}$$

Equations (B2)–(B7) hold for all interior j. Let us write symbolically

$$\{A_j^{\text{new}}\} = S(\{A_j\}, \{U_j\}, \{B_j\}, \delta t/\delta x) \tag{B8}$$

for the operation of SHASTA-FCT to advance $\{A_j\}$ by a time δt.

Using this notation, we now give the sequence of operations for advancing the dynamical variables $\{\rho_j^0\}$, $\{(\rho v)_j^0\}$, $\{E_j^0\}$ (density, momentum density, and energy density, respectively) of the self-consistent fluid equations, Eqs. (27)–(30) and Eq. (33), one timestep. The new values we denote by $\{\rho_j^1\}$, $\{(\rho v)_j^1\}$, and $\{E_j^1\}$. The algorithm used is

$$v_j^0 = (\rho v)_j^0/\rho_j^0, \tag{B9}$$

$$P_j^0 = (\gamma - 1)[E_j^0 - \tfrac{1}{2}(\rho v)_j^0\, v_j^0], \tag{B10}$$

$$v_j^{*0} = v_j^0(E_j^0 + P_j^0)/E_j^0, \tag{B11}$$

$$\{\rho_j^{1/2}\} = S(\{\rho_j^0\}, \{v_j^0\}, \{0\}, \delta t/2\delta x), \tag{B12}$$

$$\{(\rho v)_j^{1/2}\} = S(\{(\rho v)_j^0\}, \{v_j^0\}, \{P_j^0\}, \delta t/2\delta x), \tag{B13}$$

$$\{E_j^{1/2}\} = S(\{E_j^0\}, \{v_j^{*0}\}, \{0\}, \delta t/2\delta x), \tag{B14}$$

$$v_j^{1/2} = (\rho v)_j^{1/2}/\rho_j^{1/2}, \tag{B15}$$

$$P_j^{1/2} = (\gamma - 1)[E_j^{1/2} - \tfrac{1}{2}(\rho v)_j^{1/2}\, v_j^{1/2}], \tag{B16}$$

$$v_j^{*1/2} = v_j^{1/2}(E_j^{1/2} + P_j^{1/2})/E_j^{1/2}, \tag{B17}$$

$$\{\rho_j^1\} = S(\{\rho_j^0\}, \{v_j^{1/2}\}, \{0\}, \delta t/\delta x), \tag{B18}$$

$$\{(\rho v)_j^1\} = S(\{(\rho v)_j^0\}, \{v_j^{1/2}\}, \{P_j^{1/2}\}, \delta t/\delta x), \tag{B19}$$

$$\{E_j^1\} = S(\{E_j^0\}, \{v_j^{*1/2}\}, \{0\}, \delta t/\delta x). \tag{B20}$$

Acknowledgments

The authors would like to acknowledge discussions with K. Hain, B. E. McDonald, D. V. Anderson, and C. E. Wagner of the Naval Research Laboratory, who will collaborate in writing the second and third papers in this series.

References

1. R. D. Richtlyer and K. W. Morton, "Difference Methods for Initial Value Problems," Interscience Publishers, New York, 1967.
2. Fundamental Methods in Hydrodynamics, Volume 3 of "Methods in Computational Physics," (B. Alder, S. Fernbach, and Manuel Rotenberg, Eds.), Academic Press, New York, 1964.

3. P. D. Lax and B. Wendroff, Systems of Conservation laws, *Comm. Pure Appl. Math.* **13** (1960), 217.
4. K. W. Morton, Stability and convergence in Fluid Flow Problems, *Proc. Roy. Soc.* (*London*) *A* **323** (1971), 237–253.
5. K. V. Roberts and N. O. Weiss, Convective difference Schemes, *Math. Comp.* **20** (1966), 27.
6. G. J. Haltiner, Numerical weather prediction, Naval Weather Research Facility Report NWRF 30-0768-142, July 1968.
7. J. P. Boris, "A Fluid Transport Algorithm That Works," Proceedings of the Seminar Course on Computing as a Language of Physics, 2–20 August 1971, International Centre for Theoretical Physic, Trieste, Italy.
8. A. F. Emery, An evaluation of several differencing methods for inviscid fluid flow problems, *J. Comput. Phys.* **2** (1968), 306–331.
9. J. Von Neumann and R. D. Richtmyer, A method for the numerical calculations of hydrodynamical shocks, *J. Appl. Phys.* **21** (1950), 232.
10. J. Crank and P. Nicholson, A practical method for numerical integration of solutions of partial differential equations of heat conduction type, *Proc. Cambridge Philos. Soc.* **43** (1947), 50.
11. In the square wave tests of FCT (next section) we define $\bar{\rho} = (\rho_{\min}\rho_{\max})^{1/2}$ and set

$$\begin{aligned} \text{“}\tfrac{1}{8}\text{”} &= 0.125[1 + (1 - 0.25\bar{\rho}/|D_{j+1/2}|)^2], \quad |D_{j+1/2}| \geqslant 0.25\,\bar{\rho}, \\ &= 0.125 \text{ otherwise.} \end{aligned}$$

12. One-sided difference schemes of the type we test here are attributed to R. Lelevier by Richtmyer and Morton (1953).
13. W. P. Crowley, *Mon. Weather Rev.* **96** (1968), 1.
14. J. E. Fromm, A method for reducing Dispersion in convective difference schemes, *J. Comput. Phys.* **3** (1968), 176.
15. R. Courant, K. O. Fredrichs, and H. Lewy, *Math. Ann.* **100** (1928), 32.
16. A. Lapidus, Computation of radially symmetric shocked flows, *J. Comput. Phys.* **8** (1971), 106–118.