JOURNAL OF COMPUTATIONAL PHYSICS 18, 248–283 (1975)

Flux-Corrected Transport II: Generalizations of the Method

D. L. BOOK, J. P. BORIS, AND K. HAIN

Plasma Physics Division, Naval Research Laboratory, Washington, DC 20375

Received November 26, 1974

The recently developed method of Flux-Corrected Transport (FCT) can be applied to many of the finite-difference transport schemes presently in use. The result is a class of improved algorithms which add to the usual desirable properties of such schemes—conservation, stability, second-order (in some cases) accuracy, etc.—the property of maintaining the intrinsic positivity of quantities like density, energy density, and pressure. Illustrations are given for algorithms of the Lax–Wendroff, leapfrog, and upstreaming types. The errors introduced by the flux-correction process which lies at the heart of the method are cataloged and their effect described. Phoenical FCT, a refinement which minimizes residual diffusive errors, is analyzed. Applications of FCT to general fluid systems, multidimensions, and curvilinear geometry are described. The results of computer tests are shown in which the various types of FCT are compared with one another and with some conventional algorithms.

I. INTRODUCTION

This paper is the second in a planned series of four dealing with the newly developed subject of flux-corrected transport (FCT) algorithms for transient continuity and convective equations. The first paper of the series [1] (which we shall henceforth designate as FCT/1), introduced the subject of flux-corrected transport in the context of a simple one-dimensional algorithm which we named SHASTA. The mesh was uniformly spaced and stationary and only very simple 1-D test problems were presented for illustration and comparison.

The referees, readers, and the many subsequent users asked three important questions:

(1) How does FCT really work; what elements in the interaction of transport, diffusion, and nonlinear flux-correction operations are essential?

(2) What is the optimum FCT algorithm to use in a given circumstance; what is the very best that can be done?

(3) How can the techniques be generalized to complicated fluid problems, curvilinear coordinates, and multidimensions?

This paper essays to answer these three questions in a manner both easily readable and directly applicable. Some general aspects of optimal algorithms and error/stability analyses are deferred, however, until the third paper of the series[2]. It should be emphasized that FCT is a *technique*, not a differencing scheme or a specific algorithm. Mastery of this technique, we believe, enables one to generate painlessly computer programs for solving fluid equations with markedly better accuracy and "physicalness" than conventional codes of comparable speed and memory requirements.

Throughout this introductory section we purposely omit to specify the system of equations being solved. When we discuss specific differencing schemes in what follows, the dependent variable will be denoted ρ. Unless we state otherwise, it is understood that the equation propagating ρ is the continuity equation

$$(\partial\rho/\partial t) + \nabla\cdot(\rho\mathbf{v}) = 0, \tag{1}$$

where $\mathbf{v}$ is a given velocity field. Broadly speaking, most finite-difference equations with convective derivatives can be treated profitably by FCT methods. Examples are the gas dynamic equations, the Vlasov equation, and the MHD inductive law for propagating a magnetic field. On the other hand, equations describing a system in which a moderate amount of diffusion dominates convection generally do not require FCT treatment. For such equations the results are essentially the same with or without FCT.

Flux correction yields qualitatively superior results when the physics of a problem requires that a steep gradient be propagated. Such gradients may represent a shock profile, a shock-tube endwall boundary layer, or an abrupt change in a species concentration. Conventional Eulerian methods of treating such a situation are often unsatisfactory because they introduce a large numerical diffusion, or overcorrect and introduce numerical dispersion, or drastically reduce the mesh size (and hence often the timestep) in the region of the difficulty, or require special knowledge of the solution (e.g., shock fitting and some multidimensional Lagrangian techniques). Most of these problems can be related to a single diseconomy: the finite difference approximation becomes inefficient when steep gradients must be treated. FCT, in contrast, is an automatic way of applying a kind of "minimal fix" which localizes the errors to the troublesome regions alone.

The essential idea of FCT is the application of a corrective diffusion to a dispersive transport scheme, localizing the diffusion in just those regions where nonphysical ripples tend to form on account of dispersion. This corrective diffusion is *nonlinear*: its magnitude depends on the values of ρ from point to point. The diffusion is carried out in a conservative way; that is, whenever a quantity of fluid is subtracted at one point, that same amount is added back on somewhere else. Thus, small corrective quantities of material are shoved from point to point locally, but no net loss or gain to the system as a whole results. However, the equation being differenced need not be in conservative (flux) form. For example, one can difference the pressure equation instead of the energy equation. This approach has been exploited with good results, but for the present general discussion we will assume the equation being differenced describes a conservation condition.

If, for a given velocity field $v(x)$, we knew in advance exactly what a given density profile should look like after transport through one timestep t, it would be easy to calculate where to apply diffusion and how much. But, of course, we do not know. There is apparently no simple criterion for determining the amount of this diffusion a priori. Furthermore, as is shown in Section II, approaches based on simple approximations for this amount do not yield the best results. What FCT attempts to do is apply a sufficient diffusion *everywhere*, then cancel it out with an equal and opposite antidiffusion where it is clearly not needed. The criterion for this is built into the "flux limiter." The flux limiter is a nonlinear operation which transforms the antidiffusion from something which is essentially the negative of the diffusion, to something which differs from this in the neighborhood of steep gradients of ρ.

In FCT/1 it was shown that an FCT algorithm consists of three finite-difference operations: a *transport* and a *diffusion* followed by an *antidiffusion*. The transport and diffusion may be performed either as a single operation or as two successive operations. We can represent the action of these various stages respectively by means of the three symbolic operations T, D, and A. The first two of these are linear. Thus the old values of the density $\{\rho_j{}^0\}$ are carried by $(1 + T)$ into transported values $\{\rho_j{}^T\}$, and can be subsequently diffused to become $\{\rho_j^{TD}\}$, and become after antidiffusion the values at the new timestep, $\{\rho_j{}^1\}$. Symbolically, for the explicit version of FCT,

$$\rho^{TD} = (1 + T + D)\,\rho^0,$$

and

$$\rho^1 = (1 + A)\,\rho^{TD} = (1 + A)(1 + T + D)\,\rho^0.$$

Denoting by F the operation which carries ρ^0 into ρ^1, we have for explicit FCT,

$$F = (1 + A)(1 + T + D).$$

The order in which these operations are carried out is important, as they do not in general commute.

Since all three component operations are conservative, F is also. Furthermore, as a result of the application of the flux limiter, the antidiffusion $1 + A$ is positive. (An operator Ω is said to be positive if $\rho_j > C$, all j, implies $\Omega\rho_j > C$ also.) Similarly, $\rho_j < C$, all j, implies $(1 + A)\,\rho_j < C$. In general, $(1 + D)$ shares this property but $(1 + T)$ does not. It was shown in FCT/1 that the diffusive transport stage of SHASTA, where T and D are combined in a single stage, is positive if

$$| v_j \, \delta t/\delta x | \leqslant \tfrac{1}{2}$$

everywhere. When T and D are distinct, it is necessary to define the latter so that the composite operator $(1 + T + D)$ is positive. If this is done, the total operator F is itself positive. We discuss this point at length in Sections II and III below.

The remainder of the paper is structured as follows:

Section II treats the important subject of the diffusion/antidiffusion process. We discuss variable and adjustable diffusion coefficients, implicit antidiffusion, and phoenical FCT. The term "phoenical" refers to a class of diffusion/antidiffusion prescriptions which leave unchanged the variable being propagated when $(1 + T)$ reduces to the identity operation, i.e., leaves no residual diffusion.

In Section III we discuss generalizations to transport schemes other than SHASTA. The $(1 + T)$ operator can be any of the various finite difference algorithms conventionally used to solve fluid equations. We describe their advantages and disadvantages and illustrate the discussion with simple tests.

In Section IV we discuss the formulation of the flux-limiting prescription and analyze its operation. Several generalizations are described. We investigate the types of error introduced by the limiter and show how to estimate their magnitude. Various devices for avoiding or minimizing such errors are discussed.

In Section V we discuss the application of FCT to multidimensions, to non-Cartesian (curvilinear) geometries, and to the propagation of irrotational vector fields.

In Section VI we compare the results of FCT techniques with those obtained by other methods and present our conclusions. Detailed investigation of the phase and amplitude properties and of stability (linear and nonlinear) of FCT algorithms will be described in the next paper of this series, now in preparation.

II. Diffusion and Antidiffusion

One shortcoming of the original explicit version of the SHASTA FCT algorithm is the tendency for residual velocity-independent diffusion to damp short wavelengths even when the flow velocity vanishes. When the flow is nonzero, the residual diffusion is quite small (of order ϵ^2, where $\epsilon = v\delta t/\delta x$). Even so, any residual diffusion is undesirable. It was shown in Appendix A of FCT/1 that this residual diffusion can be removed using a form of implicit antidiffusion. Nevertheless, using an implicit algorithm can be unsatisfactory for two reasons: (1) The nonlocal aspect of implicit algorithms can give erroneous results in the neighborhood of shocks; and (2) the recursion relations which must be solved do not lend themselves to efficient direct solution on the new vector and pipeline machines.

Two other techniques are available to achieve the same goal. One introduces a variable diffusion/antidiffusion coefficient η; the other, phoenical antidiffusion, involves incrementing the raw antidiffusion fluxes with higher-order terms which cause exact cancellation with the diffusion when $v = 0$. (The term "phoenical" derives from "phoenix": features which sagged into shapelessness in the diffusion stage are "reincarnated" and restored to their original form during antidiffusion.) Here we describe and analyze these two explicit techniques for reducing residual diffusion.

Variable Diffusion/Antidiffusion Coefficient

It is a routine matter to change η from one time or space step to the next; the only precautions necessary are to make sure essentially the same value is used in both diffusion and antidiffusion and to ensure conservation. Since residual diffusion and the errors introduced by the flux correction process (Section IV) are both proportional to η, it is obviously desirable to choose η as small as possible consistent with avoiding nonphysical dispersive ripples.

It is easy to make this condition quantitative. Since most transport schemes we deal with are three-point formulas, we can write the result of transport alone as:

$$(1 + T)\,\rho_j = \rho_j^T = C_j^+\rho_{j+1} + C_j^0\rho_j + C_j^-\rho_{j-1}\,.$$

The C's are in general functions of velocity. Conservation imposes the restriction

$$C_{j-1}^+ + C_j^0 + C_{j+1}^- = 1.$$

The result of simultaneously applying transport and three-point diffusion (with coefficient η taken to be constant in space to simplify the argument) is

$$(1 + T + D)\,\rho_j = \rho_j^{TD} = (C_j^+ + \eta)\,\rho_{j+1} + (C_j^0 - 2\eta)\,\rho_j + (C_j^- + \eta)\,\rho_{j-1}\,. \tag{2}$$

The ρ's may have any positive values we assign them. ρ_j^{TD}, the result of diffusive transport, must be positive. Thus, $C_j^+ + \eta$, $C_j^0 - 2\eta$, $C_j^- + \eta$ should all be positive to insure this. This imposes restrictions both on η and the step size. To see this we note that $C_j^\pm$ can be negative. Hence,

$$\eta \geqslant \max_j(-C_j^\pm) \tag{3}$$

must hold. But $C_j^0 \geqslant 2\eta$ must hold everywhere also. Since $C_j^0 \to 1$ when $v\delta t/\delta x \to 0$ uniformly, this is always possible for sufficiently short timestep δt, provided $\eta < 0.5$. Thus, we find η by taking the equality in (3), and δt is restricted through

$$C_j^0(\delta t) \geqslant 2\eta, \qquad \text{all} \quad j.$$

We illustrate this for the case of SHASTA. From FCT/1, Eq. (7), we find the transport coefficient by subtracting the zeroth-order diffusion ($\eta = \frac{1}{8}$) already included:

$$\begin{aligned} C_j^+ &= \tfrac{1}{2}Q_+^2 - \tfrac{1}{8}, \\ C_j^0 &= Q_+(\tfrac{1}{2} - Q_+) + Q_-(\tfrac{1}{2} - Q_-) + \tfrac{3}{4}, \\ C_j^- &= \tfrac{1}{2}Q_-^2 - \tfrac{1}{8}, \end{aligned}$$

where

$$Q_\pm = (1/2 \mp v_j^{1/2}\, \delta t/\delta x)/[1 \pm (v_{j\pm1}^{1/2} - v_j^{1/2})\, \delta t/\delta x].$$

We consider two models. Suppose we know on physical grounds that all velocities are positive, and bounded by some $\bar{v}$:

$$0 \leqslant v_j \leqslant \bar{v} \text{ (all } j).$$

Then we can set

$$\eta = \eta_{\bar{\epsilon}} = \max_j(-C_j^\pm) = 3\bar{\epsilon}(1 - \bar{\epsilon}/2)/4(1 + \bar{\epsilon})^2,$$

where

$$\bar{\epsilon} = \bar{v}\delta t/\delta x.$$

If the bound $\bar{v}$ is related to δt by

$$\bar{v}\delta t/\delta x = \tfrac{1}{2}$$

(the restriction imposed in the original formulation of SHASTA), then we recover the original coefficient $\eta_{1/2} = 0.125$. If, however, $\bar{\epsilon} = 0.25$ (say), then a coefficient $\eta_{1/4} = 0.105$ suffices.

In the second case, we assume only that the absolute value of velocity is bounded by some (time-dependent) $\bar{v}$:

$$0 < | v_j^{1/2} | < \bar{v}.$$

Then we find that we can set

$$\eta = \max_j(-C_j^\pm) = \bar{\epsilon}/(1 + 2\bar{\epsilon})^2.$$

For $\bar{\epsilon} = 0.25$, $\eta = 9$ suffices. Again, if $\bar{\epsilon} \to 0.5$, $\eta \to 0.125$.

Thus, it is clear that, if we have a bound on velocity, it is possible to improve on the flux-correction by reducing η and hence, the errors introduced by FCT. The reduction in η is often not impressive, but there is essentially no penalty for doing this in most cases. Velocity bounds can be calculated in a single pass through the mesh, or even as it is being updated at the completion of the momentum transport.

It is natural to ask if one can reduce residual diffusion still further by letting η vary as a function of position. The purpose of this is to cut down the numerical errors which scale like η (in the uniform flow case). This improvement is feasible, but calculating η at each point can exact a significant price in CPU time (about 20–30 %). Saving $\{\eta_i\}$ until antidiffusion is carried out also requires storing an additional array, usually a trivial consideration.

This latter drawback may be avoided altogether by using a small local diffusion, just sufficient to wipe out ripples, and not antidiffusing at all. On the face of it, this appears to be big improvement; it looks more efficient than applying a diffusion and then cancelling it out almost entirely. When this prescription is coded as "flux-corrected diffusion" and actually tested, however, the results are disappointing (see Fig. 1). For this purpose the same standard test is used as in

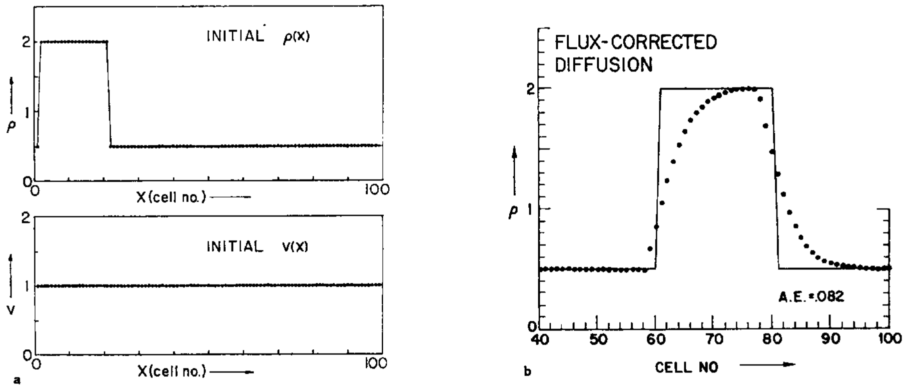

FIG. 1. (a) Initial conditions for the standard square wave test. (b) Result of square wave test with flux-corrected diffusion after 800 cycles. The analytic solution is represented by a solid curve, the computed result by dots. The mean absolute error (A.E.) is 0.082.

FCT/1. The 1-D advective equation is solved on a periodic system 100 mesh points across, with a constant advective velocity v. Except when noted, the timestep δt is taken so that $v\delta t/\delta x = 0.2$, where δx is the mesh space. A square wave 20 mesh points across, with height $\rho_{\max} = 2.0$, is superposed on a background $\rho_{\min} = 0.5$. The square wave starts at the left extremity of the system, propagates to the right, and periodically reenters on the left. Using flux-corrected diffusion the mean absolute error after 800 cycles is 0.082, about 60 % larger than for explicit SHASTA with $\eta = 0.125$, and the square wave has an eroded appearance which compares unfavorably with explicit SHASTA (cf. FCT/1, Figs. 3, 4). Partially offsetting this is the shorter running time associated with flux-corrected diffusion, only ~20 %

larger than with the uncorrected scheme. As a consequence, flux-connected diffusion should be restricted to applications where getting rid of the worst dispersive ripples is enough.

We can understand why better results are achieved by diffusing first, then antidiffusing with corrected fluxes, if we note that the additive diffusion in the definition of ρ_j^{TD}, Eq. (2), changes the phase properties of the algorithm. This would not be the case if the diffusion were applied as a separate operation on ρ_j^T,

$$\rho_j^{TD} = (1 + D)(1 + T)\,\rho_j = (1 + D)\,\rho_j^T,$$

i.e., multiplicatively. In fact, the strong added diffusion improves the phase properties by a factor of four in the case of SHASTA (see FCT/1).

There is another way to make use of spatially varying diffusion in FCT. We observe that, for conservative three-point differencing schemes, whenever the flow velocity actually attains the bound $\bar{v}$, one of the two coefficients $C_j^{\pm}$ vanishes and the transport reduces to a two-point formula, that is, local "upstream" differencing. The implication is that upstream (donor cell) differencing is the diffusive transport scheme which embodies the minimum diffusion necessary to keep ρ positive. The result can then be antidiffused (after appropriate flux correction) to eliminate most of the dissipation. We may expect that the final profile will be improved over the simple first-order "upstream" or "windward" formula, but of course again without the full diffusive phase improvements. In Section III we will see that this is indeed the case.

Phoenical Antidiffusion

When $v \equiv 0$, then the T operation vanishes. We can then write for the density profile before antidiffusion

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

For a Fourier harmonic of wave number k, the analytic solution of Eq. (1) for ρ at the nth timestep is

$$\rho_j = e^{ij\beta(1-\epsilon n)},$$

where $\beta = k\delta x$ and $\epsilon = v\delta t/\delta x$. Hence for $\epsilon = 0$,

$$\rho_j^D/\rho_j = 1 + 2\eta(\cos\beta - 1).$$

If the antidiffusive fluxes are defined (as in explicit SHASTA) by

$$f_{j+1/2} = \eta(\rho_{j+1}^D - \rho_j^D), \tag{4'}$$

then the result of uncorrected antidiffusion is

$$\begin{aligned}\rho_j^{DA}/\rho_j &= [\rho_j^{D}(1 + 2\eta) - \eta(\rho_{j-1}^{D} + \rho_{j+1}^{D})]/\rho_j \\ &= [1 - 2\eta(\cos\beta - 1)][1 + 2\eta(\cos\beta - 1)] \\ &= 1 - 4\eta^2(\cos\beta - 1)^2.\end{aligned}$$

For long wavelengths, this looks like a residual diffusion with coefficient η^2.

If the antidiffusive fluxes had satisfied

$$f_{j+1/2}^{1} = \eta(\rho_{j+1} - \rho_j) \tag{4}$$

rather than using the diffused values ρ^D as in Eq. (4′) above, antidiffusion would have exactly cancelled diffusion:

$$\rho_j^{DA} = \rho_j^{D} - f_{j+1/2}^{1} + f_{j-1/2}^{1} = \rho_j\,.$$

We antempt, then, to define the antidiffusive fluxes in general so that they reduce to (4) when the flow field vanishes.

The simplest such definition is the following one:

$$\rho_j^{1} = \rho_j^{TD} - f_{j+1/2}^{1} + f_{j-1/2}^{1}\,, \tag{5}$$

where

$$f_{j+1/2}^{1} = \eta(\rho_{j+1}^{T} - \rho_j^{T}). \tag{6}$$

Equation (6) clearly has the desired property. Because the antidiffusive fluxes depend on ρ_j^T and not ρ_j^{TD}, they are larger than in the explicit case and thus antidiffusion in this phoenical way clearly has to leave smaller residual diffusion. When v is not identically zero, small additional terms of order ϵ^2 appear. For example, from FCT/1, Eq. (12), we see that SHASTA satisfies

$$\rho_j^{TD}/\rho_j = 1 + (1/4 + \epsilon^2)(\cos\beta - 1) - i\epsilon\sin\beta.$$

Subtracting the zeroth-order diffusion ($\eta = \frac{1}{8}$) yields

$$\rho_j^{T}/\rho_j = 1 + \epsilon^2(\cos\beta - 1) - i\epsilon\sin\beta,$$

so that uncorrected phoenical antidiffusion has an amplification factor

$$\begin{aligned}|\rho_j^{1}/\rho_j^{0}|^2 = 1 - 2\epsilon^2\sin^4(\beta/2)[1 + \tfrac{1}{2}\sin^2(\beta/2)]\{1 + \sin^2(\beta/2) \\ - 2\epsilon^2[1 + \tfrac{1}{2}\sin^2(\beta/2)]\}.\end{aligned} \tag{7}$$

For long wavelengths this differs from unity by a quantity $\sim\epsilon^2(\beta/2)^4$ compared with a residual discrepancy $\sim\frac{1}{4}(\beta/2)^4$ for explicit SHASTA (FCT/1, Eq. (20)). For $\beta \to 2\pi$, the amplification factor becomes $|\rho_j^1/\rho_j|^2 = (1 - 3\epsilon^2)^2$ and thus is stable for $\epsilon^2 < \frac{2}{3}$; the corresponding result for explicit SHASTA is seen to be $\epsilon^2 < \frac{7}{12}$.

We now summarize the methods of explicit, implicit and phoenical diffusion/antidiffusion by means of the operator notation introduced earlier. The three can be written respectively as:

$$\rho_j^E = (1 + A)(1 + T + D)\,\rho_j^0, \tag{8}$$

$$\rho_j^I = (1 + D)^{-1}(1 + T + D)\,\rho_j^0, \tag{9}$$

$$\rho_j^P = [(1 + A)(1 + T) + D]\,\rho_j^0. \tag{10}$$

If one shuts off the flux limiter, thus eliminating the nonlinear effects of flux correction, it is clear that when $T \to 0$ and $A \to -D$, Eqs. (9) and (10) reduce to the identity operation, but Eq. (8) does not.

Figure 2 shows the results of the standard square wave test (see above) using

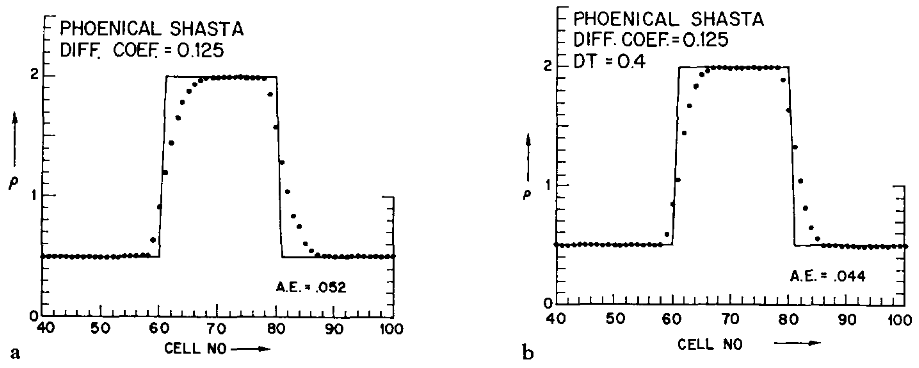

FIG. 2. Square wave test with phoenical SHASTA using timestep (a) $\delta t = 0.2$, 800 cycles, and (b) $\delta t = 0.4$, 400 cycles. Note that the mean absolute error is lower for case (b), because the residual diffusion left by FCT scales with the number of cycles.

phoenical SHASTA with a diffusion/antidiffusion coefficient $\eta = 0.125$. Shown are plots of the density profile for $\delta t = 0.2$ after 800 cycles and for $\delta t = 0.4$ after 400 cycles; the square wave has travelled the same distance in both cases, but ϵ equals 0.2 and 0.4 respectively. Since the residual diffusion depends on the ϵ-independent nonlinear flux correction as well as the well-behaved ϵ dependences, taking fewer but longer timesteps is clearly advantageous. (See FCT/1, Fig. 10, for a comparison with the explicit and implicit methods.)

Phoenical FCT algorithms yield results essentially identical with those of flux-uncorrected schemes in the absence of large gradients, just as ordinary explicit

FCT algorithms do. In addition, though, they leave unchanged steep fronts in regions where no disturbance takes place. This is important in application where a shock propagates into a stationary region in which gradients in concentration or temperature have been previously established. Since there is no appreciable penalty in running time or storage requirements for using phoenical instead of explicit antidiffusion, it is to be preferred in most applications. Therefore, most of the analyses and examples in the following sections will deal with phoenical FCT.

III. Applying Flux Correction to General Difference Schemes

In this section we give prescriptions for flux-correcting three finite-difference schemes commonly used in solving fluid equations: two-step Lax–Wendroff, leapfrog, and upstreaming (or donor-cell). Extension of the technique to many other algorithms in common use should follow by example and be quite straightforward.

Phoenical Lax–Wendroff

Of the three, Lax–Wendroff (L–W) is closest to SHASTA. In fact, SHASTA reduces in the case of uniform flow velocity to L–W with a zeroth-order added diffusion (cf. FCT/1). We consider the continuity equation satisfied by a density $\rho_j{}^n$, carried along by a flow field $v_j{}^n$. Here j labels the mesh point and n takes the values 0, $\frac{1}{2}$ and 1 at the beginning, midpoint, and completion of a timestep, respectively. The midpoint values are defined on a staggered mesh at $j \pm \frac{1}{2}$. Assuming the flow field is known at each time, we solve for ρ^1 as follows:

(a) Advance ρ one-half timestep:

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

(b) Generate diffusive fluxes from first differences of $\rho_j{}^0$

$$f_{j+1/2}^0 = \eta[\rho_{j+1}^0 - \rho_j{}^0].$$

(c) Advance ρ for the whole timestep:

$$\tilde{\rho}_j{}^1 = \rho_j{}^0 - (\delta t/\delta x)[\rho_{j+1/2}^{1/2} v_{j+1/2}^{1/2} - \rho_{j-1/2}^{1/2} v_{j-1/2}^{1/2}].$$

(d) Generate antidiffusive fluxes from first differences of $\tilde{\rho}_j{}^1$

$$f_{j+1/2}^1 = \eta(\tilde{\rho}_{j+1}^1 - \tilde{\rho}_j{}^1).$$

(e) Diffuse the transported ρ using saved first differences:

$$\bar{\rho}_j^{\,1} = \tilde{\rho}_j^{\,1} + f^0_{j+1/2} - f^0_{j-1/2}\,.$$

(f) Take first differences of the diffused, transported density

$$\Delta_{j+1/2} = \bar{\rho}^{1}_{j+1} - \bar{\rho}_j^{\,1}.$$

(g) Limit the antidiffusive fluxes:

$$S = \operatorname{sign}\{f^1_{j+1/2}\}$$

$$f^C_{j+1/2} = S\max\{0, \min[S\cdot\Delta_{j-1/2}\,, |\,f^1_{j+1/2}\,|, S\cdot\Delta_{j+3/2}]\}.$$

(h) Antidiffuse with the limited fluxes:

$$\rho_j^{\,1} = \bar{\rho}_j^{\,1} - f^C_{j+1/2} + f^C_{j-1/2}\,.$$

Steps (a) and (c) are the usual half- and whole-step L–W transport operations. Steps (b) and (d)–(h), introduced by the flux-correction procedure, are easily coded, using two additional scratch arrays. An increase in CPU time of about 50 % over conventional schemes is demanded. However, conventional schemes are never actually run at this maximum timestep. In fluid simulations, the timestep is limited by $(|\,v\,| + c)\,\delta t/\delta x < 1$, where c is the sound velocity, and by accuracy considerations. In direct comparisons of working codes we have made, FCT versions have run much faster for a given accuracy.

Unlike SHASTA, the output from the diffusive transport (step (e)) in the Lax–Wendroff scheme is not intrinsically positive. If $v_j^{\,n}$ is a constant, $\eta \geqslant 0.125$ suffices for positivity (as in SHASTA), but if the velocity varies with position and time, a larger value of η may be required because of the half-step modifications of ρ. For example, arguments similar to those of the preceding section show that if v is bounded, i.e., $|\,v_j^{\,n}\delta t/\delta x\,| < \bar{\epsilon}$, then $\eta = \frac{1}{2}\bar{\epsilon}(1 + \bar{\epsilon})$ suffices for positivity, provided $\bar{\epsilon} \leqslant \frac{1}{2}(-1 + \sqrt{3}) \approx .366$. For larger timesteps violating the latter restriction, because of the properties of the L–W transport stage, no diffusion coefficient guarantees positivity everywhere; this is one of our reasons for preferring SHASTA to Lax–Wendroff in practice. Since L–W and SHASTA are identical when applied to the constant-velocity test version of Eq. (1), the plots in Fig. 2 serve equally well to illustrate flux-corrected L–W. Figure 3 shows the relative efficacy of various levels of diffusion/antidiffusion. The plots show the results of the square wave test for $\eta = 0.0$, 0.0625, 0.125 and 0.5, respectively. Clearly, too much diffusion is as bad as too little. For coupled nonlinear equations in a variable medium, these constant-velocity results are only approximate; but experience shows that they are a very close approximation.

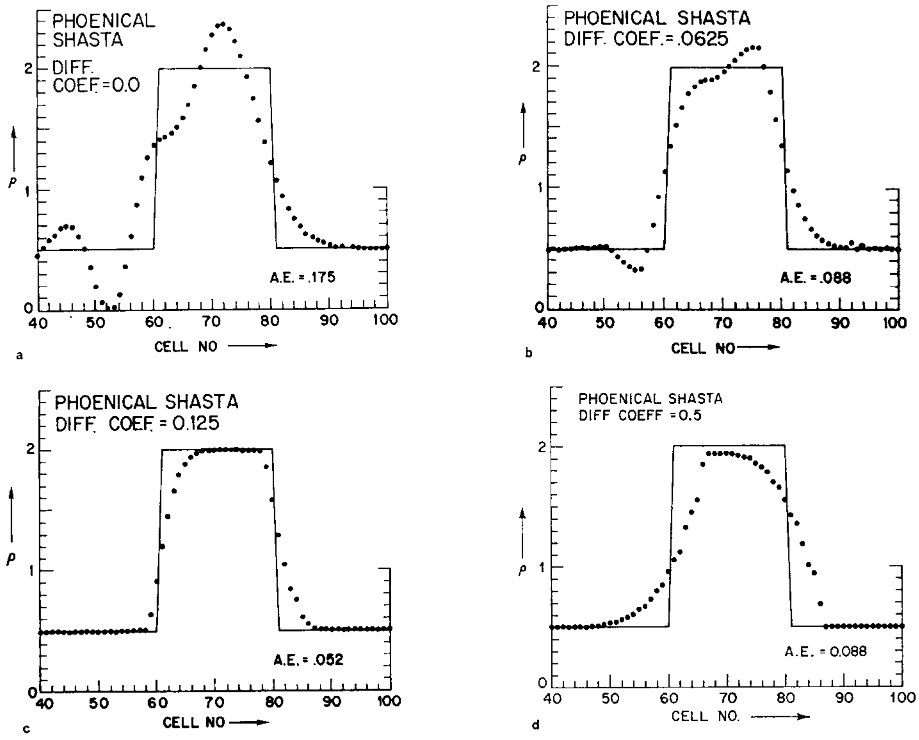

FIG. 3. Phoenical SHASTA (or Lax–Wendroff) with diffusion/antidiffusion coefficient (a) $\eta = 0.0$; (b) $\eta = 1/16$; (c) $\eta = 1/8$; (d) $\eta = 1/2$.

Leapfrog FCT

Implementing FCT in the leapfrog scheme is somewhat harder because two distinct meshes are present. The tendency for solutions on the two meshes to separate (weak instability) is reflected in the computational difficulty in defining a suitable (anti-) diffusion. In general, when a flux connects two density meshes defined at different times, the (anti-) diffusion process in which this flux is employed is no longer conservative on either mesh independently. While it is feasible to apply FCT ignoring the difference between the two time levels in defining fluxes and differences, better results are obtained using two slightly more sophisticated approaches. The first proceeds as follows (we take n, j odd unless otherwise stated).

(a) Compute diffusive fluxes:

$$f^0_{j+1/2} = 2\eta(\rho^{n-1}_{j+1} - \rho_j{}^n),$$
$$f^0_{j-1/2} = 2\eta(\rho_j{}^n - \rho^{n-1}_{j-1}).$$

(b) Advance the densities on the even mesh using the leapfrog algorithm:

$$\rho_{j+1}^{n+1} = \rho_{j+1}^{n-1} - (\delta t/\delta x)(v_{j+2}^{n}\rho_{j+2}^{n} - v_j{}^n\rho_j{}^n).$$

(c) Advance the densities on the odd mesh:

$$\rho_j^{n+2} = \rho_j{}^n - (\delta t/\delta x)(v_{j+1}^{n+1}\rho_{j+1}^{n+1} - v_{j-1}^{n+1}\rho_{j-1}^{n+1}).$$

(d) Compute the antidiffusive fluxes:

$$f_{j+1/2}^{1} = 2\eta(\rho_{j+1}^{n+1} - \rho_j^{n+2}),$$
$$f_{j-1/2}^{1} = 2\eta(\rho_j^{n+2} - \rho_{j-1}^{n+1}).$$

(e) Apply the diffusion:

$$\tilde{\rho}_{j+1}^{n+1} = \rho_{j+1}^{n+1} + f_{j+3/2}^{0} - f_{j+1/2}^{0},$$
$$\tilde{\rho}_j^{n+2} = \rho_j^{n+2} + f_{j+1/2}^{0} - f_{j-1/2}^{0}.$$

(f) Take first differences of the diffused transported ρ:

$$\Delta_{j+1/2} = \tilde{\rho}_{j+1}^{n+1} - \tilde{\rho}_j^{n+2},$$
$$\Delta_{j-1/2} = \tilde{\rho}_j^{n+2} - \tilde{\rho}_{j-1}^{n+1}.$$

(g) Limit the fluxes

$$S = \operatorname{sign}(f_{j+1/2}^{1}),$$
$$f_{j+1/2}^{C} = S \cdot \max[0, \min(S \cdot \Delta_{j-1/2}, |f_{j+1/2}^{1}|, S \cdot \Delta_{j+1/2})] \quad (\text{all } j).$$

(h) Apply the antidiffusion:

$$\rho_j^{n+2} = \tilde{\rho}_j^{n+2} - f_{j+1/2}^{C} + f_{j-1/2}^{C},$$
$$\rho_{j+1}^{n+1} = \tilde{\rho}_{j+1}^{n+1} - f_{j+3/2}^{C} - f_{j-3/2}^{C}.$$

In this prescription 2η is roughly twice the diffusion coefficient used in SHASTA ($\eta = \frac{1}{8}$). Since the FCT process is applied to "clean up" the profile effectively by using a doubled value of η every other timestep, it is not surprising that residual diffusion errors are somewhat larger. This is apparent in the square wave advection test (Fig. 4a). The algorithm is conservative only in a weakened sense:

$$\sum_{\text{odd } j} (\rho_j{}^n + \rho_{j+1}^{n+1}) = \text{const},$$

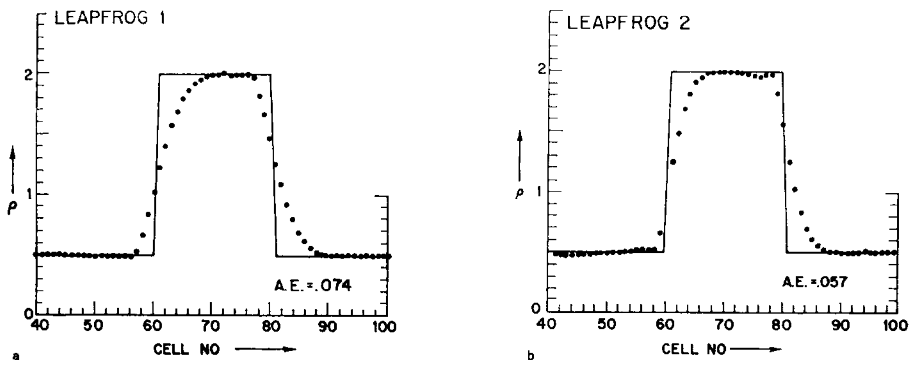

FIG. 4. Square-wave test with two versions of flux-corrected leapfrog.

if computed after the conclusion of the antidiffusion process, that is, every second timestep.

The second method of putting flux correction into leapfrog is more of a brute-force technique. It involves increasing the number of mesh points (and hence the running time and resolution) by a factor of two. In the following description there is thus no restriction on n and j.

(a) Calculate the diffusive fluxes of ρ:

$$f^0_{j+1/2} = \eta(\rho^n_{j+1} - \rho_j{}^n).$$

(b) Advance ρ:

$$\rho_j^{n+2} = \rho_j{}^n - (\delta t/\delta x)(v^{n+1}_{j+1}\rho^{n+1}_{j+1} + v^{n+1}_{j-1}\rho^{n+1}_{j-1}).$$

(c) Calculate antidiffusive fluxes:

$$f^1_{j+1/2} = \eta(\rho^{n+2}_{j+1} - \rho_j^{n+2}).$$

(d) Apply diffusion:

$$\tilde{\rho}_j^{n+2} = \rho_j^{n+2} + f^0_{j+1/2} - f^0_{j-1/2}\,.$$

(e) Take first differences of $\tilde{\rho}$:

$$\Delta_{j+1/2} = \tilde{\rho}^{n+2}_{j+1} - \tilde{\rho}_j^{n+2}.$$

(f) Limit the fluxes:

$$S = \operatorname{sign}(f_{j+1/2}),$$

$$f^C_{j+1/2} = S \cdot \max[0, \min(S \cdot \Delta_{j-1/2}\,, \, |f^1_{j+1/2}|, \, S \cdot \Delta_{j+1/2})].$$

(g) Apply the antidiffusion:

$$\rho_j^{n+2} = \tilde{\rho}_j^{n+2} - f_{j+1/2}^C + f_{j-1/2}^C .$$

This version is phoenical and fully conservative on the separate meshes. The result of propagating a square wave as before is shown in Fig. 4b. The error is smaller than in the first leapfrog method and comparable with that found with Lax–Wendroff. Only the derivative of one mesh interacts with the other. There is nothing in the algorithm which couples the two time levels, so one mesh could separate significantly from the second after many cycles. To prevent this, as in the usual leapfrog algorithm, some mesh averaging could be applied.

Donor Cell (Upstreaming)

Donor-cell (in contrast to Lax–Wendroff and leapfrog) is a first-order scheme. It embodies a *velocity-dependent* diffusion. By applying an antidiffusion stage with the same (velocity-dependent) value for the coefficient, where the antidiffusive fluxes have been corrected as above, we can obtain a fast, positive (in the sense of Section I) algorithm which is essentially second order to boot. The sequence of operation (all velocities assumed positive for simplicity) is

(a) Transport ρ:

$$\tilde{\rho}_j = \rho_j^0 - (\delta t/\delta x)[\rho_j^0 v_j^0 - \rho_{j-1}^0 v_{j-1}^0].$$

(b) Compute the first differences:

$$\Delta_{j+1/2} = \tilde{\rho}_{j+1}^1 - \tilde{\rho}_j^1.$$

(c) Correct the antidiffusive fluxes:

$$\begin{aligned} S &= \operatorname{sign}\{\Delta_{j+1/2}\}, \\ f_{j+1/2} &= S \cdot \max\{0, \min[S \cdot \Delta_{j-1/2}, \eta_{j+1/2} \mid \Delta_{j+1/2} \mid, \\ &\quad S \cdot \Delta_{j+3/2}]\}, \end{aligned}$$

where

$$\begin{aligned} \eta_{j+1/2} &= \tfrac{1}{2}\epsilon_{j+1/2}(1 - \epsilon_{j+1/2}), \\ \epsilon_{j+1/2} &= (\delta t/\delta x) \cdot \max[v_j^0, v_{j+1}]. \end{aligned}$$

(d) Antidiffuse:

$$\rho_j^1 = \rho_j^1 - f_{j+1/2} + f_{j-1/2} .$$

Step (c) requires some comment. When the flow field is uniform ($v_j^n = v_0^n$, all j), the antidiffusive coefficients reduce to $\eta = \frac{1}{2}\epsilon(1 - \epsilon)$, where $\epsilon = v_0^n \delta t/\delta x$. (It is just this choice of η which makes the algorithm second order, as may easily be verified, using the method of Section II.) For $\epsilon = \frac{1}{2}$, $\eta = \frac{1}{8}$ as for SHASTA. When velocities of both signs are present, $| v_j^0 |$ replaces v_j^0 in the definition of $\eta_{j+1/2}$.

Donor-cell FCT is positive because, as with SHASTA, the diffusive transport and the antidiffusive stages are both positive. It is especially useful in applications where the flow field is uniform, although the phase properties are not optimum. Figure 5 shows the result of the square wave test. As is evident, the results are

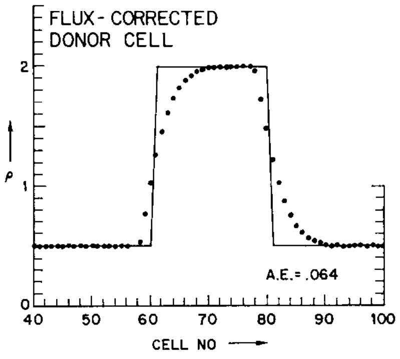

FIG. 5. Square-wave test with flux-corrected donor-cell (upwind or one-sided differencing).

qualitatively similar for all three commonly used algorithms, and in each case very different from the result obtained in the absence of flux correction.

As a rough guideline, we can recommend the following rules for deciding which algorithm to use with flux correction:

(1) For an already existing code, the transport scheme already incorporated may be quite adequate and can be improved by the simple addition of a flux correction.

(2) If it is desired to minimize dispersion while maintaining positivity, one of the versions of SHASTA should be used.

(3) Donor-cell should be used when the values of the flow velocity ase small enough that phase errors are small compared with diffusive errors.

IV. FLUX LIMITERS

The antidiffusion stage of the algorithms described above and in FCT/1 may be written

$$\rho_j^1 = \tilde{\rho}_j^1 - f_{j+1/2}^C + f_{j-1/2}^C , \tag{11}$$

where $\tilde{\rho}_j^1$ is the result of diffusive transport alone and $f_{j\pm1/2}^C$ are the corrected (or limited) fluxes, given by

$$f_{j+1/2}^C = S \cdot \max\{0, \min[S \cdot \Delta_{j-1/2}, |f_{j+1/2}^1|, S \cdot \Delta_{j+3/2}]\}. \tag{12}$$

Here $f_{j+1/2}^1$ is a raw or uncorrected flux, and $S = \operatorname{sign}\{f_{j+1/2}^1\}$. The exact definition of $f_{j+1/2}$ depends on the transport scheme employed, and on whether explicit, phoenical or implicit antidiffusion is used. We will speak of the densities at the two points j and $j + 1$ as being *connected* by $f_{j+1/2}$. When two densities are connected during antidiffusion by a flux, the higher grows at the expense of the lower by the amount of the flux.

The formulation (11)–(12) is known as strong flux correction. It is fast-running and easy to code, but other prescriptions with useful properties are possible. Here we briefly describe how strong flux correction works and then consider modifications.

The input information used consists of the values of $f_{j+1/2}^1$, $\Delta_{j-1/2}$, and $\Delta_{j+3/2}$. Suppose $f_{j+1/2}^1 > 0$. (The opposite case exactly mirrors this one.) Then there are four possibilities:

(a) $\Delta_{j-1/2} > 0, \quad \Delta_{j+3/2} > 0;$
(b) $\Delta_{j-1/2} > 0, \quad \Delta_{j+3/2} \leqslant 0;$
(c) $\Delta_{j-1/2} \leqslant 0, \quad \Delta_{j+3/2} > 0;$
(d) $\Delta_{j-1/2} \leqslant 0, \quad \Delta_{j+3/2} \leqslant 0.$

These are represented schematically in Fig. 6. We consider the cases in order.

(a) This is the "normal" situation. The limiter leaves $f_{j+1/2}$ unchanged unless the jth point would be pushed below the $(j - 1)$st or the $(j + 1)$st would be pushed above the $(j + 2)$nd, in which cases $f_{j+1/2}$ is reduced (corrected) to avoid this.

(b) Since nonvanishing $f_{j+1/2}$ would accentuate the maximum at the $(j + 1)$st point, the limiter sets $f_{j+1/2}^C = 0$.

(c) The limiter sets $f_{j+1/2}^C = 0$ to avoid accentuating the minimum at the jth point.

(d) The limiter again sets $f_{j+1/2}^C = 0$; otherwise *both* extrema would be enhanced.

The limiter thus guarantees that no local extrema ("ripples") form or grow as a result of antidiffusion. We wish to stress, however, that the transport stage can cause local extrema to form and grow as the physics of the problem demands. Errors can arise during antidiffusion in two ways; The raw flux $f_{j+1/2}$ may not be such as to yield the correct antidiffusion even when the limiter plays no role $(f_{j+1/2}^C = f_{j+1/2}^1)$; or the limiter, which is decidedly pessimistic, may overcorrect the fluxes and leave an unnecessarily large net diffusion. We have already treated the

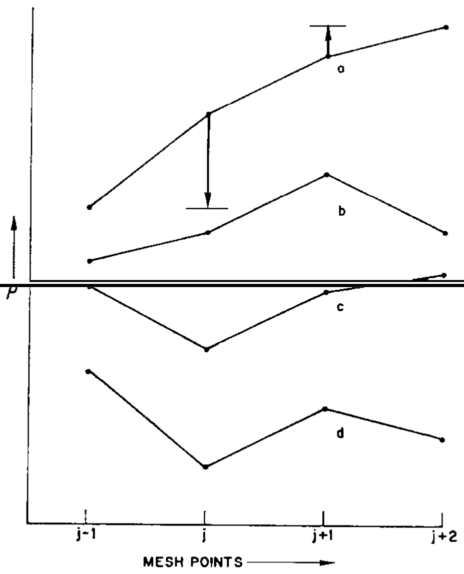

FIG. 6. The four different possibilities in correcting a positive flux connecting the points j and $j + 1$. In (a), the horizontal line segments indicate the distances beyond which the flux limiter will not allow the points j and $j + 1$ to be pushed.

former problem in Section II; here we examine the errors introduced by the limiter itself.

These errors are exhibited primarily in a phenomenon called "clipping." Clipping occurs as a result of the property of Eq. (12) which forbids extrema present after diffusive transport has occurred from being enhanced in the antidiffusive stage.

This is shown clearly in Tables I and II, and in Fig. 7. Consider the sharply peaked density profile whose values are displayed in Table I. We repeatedly diffuse and antidiffuse this density with a coefficient $\eta = \frac{1}{8}$, using *explicit* antidiffusion:

$$\rho_j^D = \rho_j + \eta(\rho_{j+1} - 2\rho_j + \rho_{j-1}),$$

$$\rho_j^{DA} = \rho_j^D - \eta(\rho_j^D - 2\rho_j^D + \rho_{j-1}^D).$$

The result is shown in successive lines of the table. It is, of course, identical with what would be obtained using an explicit FCT algorithm with zero flow velocity. As is clear, the profile is initially strongly diffused. After about 10 cycles, a distinct flattening or "clipping" is apparent at the peak of the profile.

The explanation lies in understanding the action of strong flux correction. Clearly a local extremum (a maximum, say) which flattens when diffusion takes place wants to grow back again during antidiffusion. But this cannot happen with

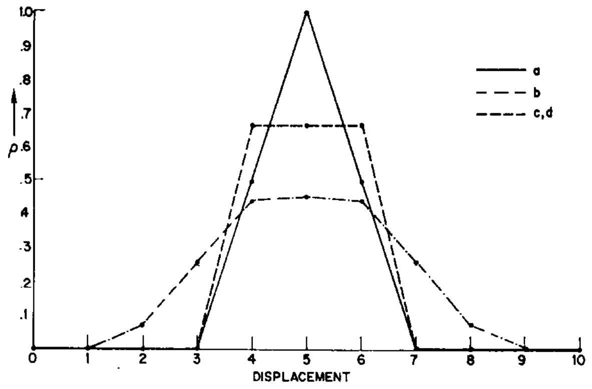

FIG. 7. Results of successively diffusing and antidiffusing the sharply peaked profile (a) 20 cycles with diffusion/antidiffusion coefficient $\eta = 0.2$. Curve (b) is produced by explicit antidiffusion, curves (c) and (d) (indistinguishable on the scale of the plot) by phoenical and implicit antidiffusion.

TABLE I

The result of successive diffusion/antidiffusion operations with flux limiting on a sharp peak (cf. Fig. 7) using $\eta = 1/8$ and explicit antidiffusion. Although 11 points were employed across the mesh, the values found are symmetrical about $j = 6$, $\rho(12 - j) = \rho(j)$, so those to the right of the peak are not shown.

Cycle \ Grid pt.	1	2	3	4	5	6
0	0.0	0.0	0.0	0.0	0.5	1.0
1	0.0	0.0	0.0	0.05520	0.58880	0.71200
2	0.0	0.0	0.0	0.09075	0.57789	0.66272
3	0.0	0.0	0.0	0.12270	0.56290	0.62879
4	0.0	0.0	0.0	0.15037	0.54841	0.60244
5	0.0	0.0	0.0	0.17404	0.53555	0.58083
6	0.0	0.0	0.0	0.19419	0.52446	0.56271
7	0.0	0.0	0.00232	0.20900	0.51498	0.54741
8	0.0	0.0	0.00652	0.21970	0.50655	0.53444
9	0.0	0.0	0.01184	0.22769	0.49882	0.52328
10	0.0	0.0	0.01779	0.23381	0.49164	0.51350
15	0.0	0.0	0.04294	0.24884	0.46686	0.48273
20	0.0	0.0	0.07095	0.25947	0.44239	0.45439

TABLE II

As with Table II, but with implicit or phoenical antidiffusion, which are indistinguishable to within round-off. The entries marked with an asterisk are actually nonzero of order 10^{-7} or less for the implicit case.

Cycle \ Grid pt.	1	2	3	4	5	6
0	0.0	0.0	0.0	0.0	0.5	1.0
1	0.0	0.0	0.0	0.0	0.6	0.8
2	0.0*	0.0*	0.0*	0.0	0.64	0.72
3	0.0*	0.0*	0.0*	0.0	0.656	0.688
4	0.0*	0.0*	0.0*	0.0	0.66496	0.67008
5	0.0*	0.0*	0.0*	0.0	0.66598	0.66803
6	0.0*	0.0*	0.0*	0.0	0.66639	0.66721
7	0.0*	0.0*	0.0*	0.0	0.66656	0.66689
8	0.0*	0.0*	0.0*	0.0	0.66623	0.66675
9	0.0*	0.0*	0.0*	0.0	0.66665	0.66670
10	0.0*	0.0*	0.0*	0.0	0.66666	0.66668
15	0.0*	0.0*	0.0*	0.0*	0.66667	0.66667
20	0.0*	0.0*	0.0*	0.0*	0.66667	0.66667

strong, flux correction, as a sharp feature becomes blunted. The single point at the peak of the maximum shown decreases by virtue of the fluxes it passes to its two neighbors. Each of these passes one flux downhill and one uphill (to the peak); these fluxes represent antagonistic tendencies which almost cancel out, so that the shoulders are eaten away more slowly than the peak. After 20 cycles an approximately flat plateau three points across has formed (Fig. 7b). When this happens, the central point no longer gets pushed downward through diffusion, while antidiffusion is permitted to push back the two points to either side, almost precisely compensating for the effects of diffusion on them. Hence a flat three-point plateau is a nearly stable structure. Clearly clipping is a nonlinear diffusion process which affects the short wavelength components more than the long. The characteristic plateau shape is almost stationary because the short wavelength components have reached their final values, while the long wavelength residual diffusion that remains is very weak (see Fig. 8). When such a plateau is convected across the grid by a smooth flow, the plateau must broaden another grid point or so to permit at least three points to be roughly constant at all times.

Table II and Fig. 7c display the results for the same profile subjected to successive diffusion and phoenical antidiffusion stages. Here the phenomenon is more dramatic. The plateau forms after about three cycles and reaches its asymptotic state (to six-place accuracy) after about eight. To within round-off, we find the same results using implicit antidiffusion. The plateau forms quickly, as in the phoenical case, but now nonzero values for the profile arise to the right and left of the three central points, reflecting the interaction between the nonlocal nature of the implicit operation and machine round-off.

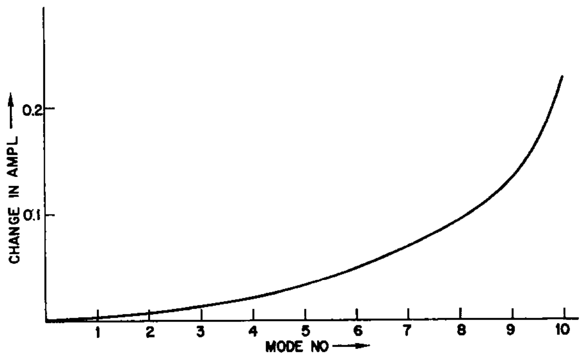

FIG. 8. Wave-number dependence of the clipping phenomenon. The ordinate labels the relative decrease in amplitude after 100 cycles of phoenical diffusion/antidiffusion using strong flux limiting; the abscissia is the mode number $K = 2\pi L/\lambda$, where L is the system size and λ the wavelength.

Whenever strong flux correction is used in propagating a profile with such isolated extrema, characteristic plateaus are observed to form. In the absence of a physical steepening mechanism, resolution on a finer scale than this is precluded (as it would be in any finite difference algorithm). The rapidity with which clipping occurs depends strongly on the breadth of the local extremum. In other words, it is strongly wavelength dependent. Figure 8 shows the dependence on mode number of the amplitude of a given harmonic, initially equal to unity, after 100 cycles of phoenical diffusion/antidiffusion with $\eta = \frac{1}{8}$. The enhanced susceptibility of the shorter wavelengths to clipping is clearly displayed.

A number of specific ideas have been investigated with the goal of eliminating or minimizing the effects of clipping. They generally work but are advantageous only for particular problems where the character of the solution is essentially known in advance. The most useful alternative to strong flux correction is one-sided flux limiting [3]. In some problems the solution is known to be positive and to have the form of a single pulse or peak growing out of a more or less uniform background. An example is the problem of determining the amount of magnetic compression found at the front of a plasma cloud expanding into a stationary magnetized plasma [4]. The amplitude and width of this peak are the quantities of physical interest. FCT wipes out the train of dispersive ripples which would otherwise trail the peak, but the value of the maximum can be too low by 30–40 % in the case of very narrow peaks because of clipping.

Allowing already existing extrema to grow, while legislating against the formation of new ones ("weak" flux correction) does not help. Round-off errors lead to spurious extrema even in a nominally flat profile; initially tiny, they blow up and drive the solution unstable. However, if maxima are allowed to appear or grow while *minima* are restricted as in strong flux correction, the algorithm retains both positivity and stability. Intuitively, this reflects the fact that shifting mass from a mesh point to its neighbor in such a way that the latter rises dramatically makes the former decrease by an equal amount, creating a local minimum. Preventing the formation of minima thus implies a measure of control on the formation of maxima.

The prescription for one-sided flux limiting, replacing Eq. (12) is

$$f^C_{j+1/2} = S \cdot \max\{0, \min[S \cdot \Delta_{j'+1/2}, |f^1_{j+1/2}|]\} \tag{13}$$

with $j' = j - S \cdot 1$ and $S, \Delta_{j+1/2}$, and $f^1_{j+1/2}$ defined as before. Use of Eq. (13) results in profiles where the troughs can be clipped, but not the peaks. If ρ is strictly negative, it is natural to redefine $j' = j + S \cdot 1$. This has the effect of allowing minima to grow, but preventing growth of maxima. And if ρ can take either sign, and we are interested in accurately calculating the regions of largest excursion from $\rho = 0$, the obvious generalization is $j' = j - S \cdot \tilde{S}$, where $\tilde{S} = \text{sign}(\rho_j)$(double one-sided flux limiting).

In addition to the flux compression problem cited above, one-sided and double one-sided limiters have been successfully used in studies of the motion of barium clouds coupled electrostatically to the ionosphere [5] and in construction of a finite-difference Vlasov solver [6]. These examples involve solution of a continuity equation, or a set of continuity equations, with the velocity field derived from a potential field. Application to coupled nonlinear fluid equations in which the velocity is propagated by means of a Navier–Stokes equation tells a different story. Dispersive ripples are almost as bad as without FCT; the only difference is that the ripples are "one-sided." Hence we conclude as a matter of experience that use of one-sided flux limiting should be restricted to solution of convective problems with a prescribed flow field.

In addition to clipping, there is a second kind of nonlinear process called "terracing." It is sometimes observed on the flanks of steep gradients extending over several (~5) mesh points. Qualitatively, it arises because an undamped dispersive ripple on a sufficiently steep slope may distort the profile markedly without actually creating new extrema (Fig. 9a) and hence without signalling its existence to the flux limiter. Eventually such a ripple may grow to the point where the flux limiter comes into play. The effect is then to make a weak shelf or terrace about three points across (Fig. 9b). Comparisons with conventional schemes show that the terraces tend to appear in the same places where large-scale ripples would otherwise form.

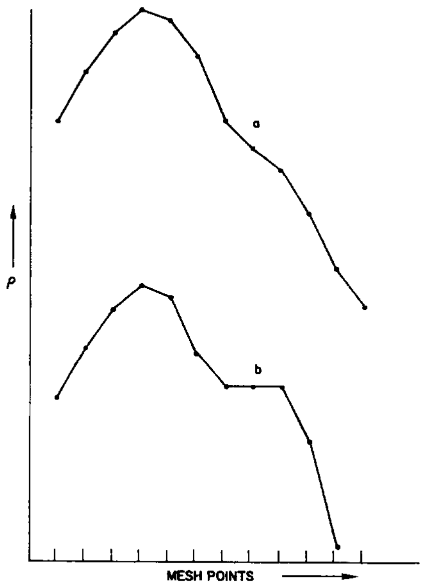

FIG. 9. Schematic portrayal of formation of a "terrace" on the forward side of an advancing region of enhanced fluid density.

~~No convenient means of preventing terracing, other than artificial smoothing,~~ has been developed, although several experimental techniques have received attention. The terracing phenomena really betokens phase errors in the underlying transport algorithm rather than deficiencies of the flux corrector. In any case, if gradients are relatively gentle or a shock-like jump takes place across just one mesh space only, no terraces appear.

V. GENERALIZED GEOMETRIES

Everything said so far about solving the continuity equation has related to a one-dimensional (rectilinear) coordinate system. In this section we show how the techniques already discussed can be carried over to the other coordinate systems in common use. We treat the following topics: multidimensional spaces, curvilinear coordinates, and the propagation of irrotational vector fields.

Multidimensions

We have restricted the discussion to the 1-D Cartesian form of the continuity equation in previous sections. All of the component operations (transport, diffusion and antidiffusion) have obvious generalizations to multidimensions. Let $O_l^{(1)}$ stand for any of the operators T, D, A (here denoting un-flux-corrected antidiffusion), acting on a vector in the $\{\rho\}$ matrix which lies in the direction of the lth coordinate ($l = 1, 2,..., N$). If $O_{l'}^{(1)}$ is a second such operator, clearly $O_l^{(1)}$ and $O_{l'}^{(1)}$ commute. Then the N-dimensional form of the operator in question may be constructed according to either

$$1 + O^{(N)} = \prod_{i=1}^{N} (1 + O_i^{(1)}) \tag{14}$$

or

$$1 + \hat{O}^{(N)} = 1 + \sum_{i=1}^{N} O_i^{(1)}. \tag{15}$$

Equation (14) defines a 3^N-point formula, Eq. (15) a $(2N + 1)$-point formula. Both $O^{(N)}$ and $\hat{O}^{(N)}$ are conservative if $O^{(1)}$ is. If $O^{(1)} = T$ or D, the same is true with respect to the property of positivity, as defined in Section I.

We now wish to generalize the strong flux correction formula (Eq. (12)), but encounter serious difficulties. Any formula which limits a flux f^1 according to what it does to the two points it connects, relative to all their nearest neighbors, would be extremely unwieldy. Worse, conceptual difficulties occur when one tried to define a formula that works at saddle points, for oblique flows, and in almost one-dimensional problems as well as the general case. It is thus not surprising that no N-dimensional equivalent to Eq. (12) which treats all the coordinate axes symmetrically has been developed.

What we have done instead is to employ alternating sweeps through the matrix in each of the coordinate directions, using the one-dimensional form of strong flux correction. It is clearly most efficient to perform the antidiffusion at the same time as the flux limiting, rather than store all the fluxes. For phoenical algorithms, the same economy dictates that the diffusive transport for each coordinate direction be performed immediately before antidiffusing. Thus we are led to an algorithm in which the sequence of operations is $(1 + T_1 + D_1)$, $(1 + A_1)$, $(1 + T_2 + D_2)$, $(1 + A_2)$,..., etc. In other words, a timestep-split generalization of the 1-D algorithms falls out naturally.

The timestep splitting, of course, need only be used be used for the flux correction and antidiffusion portions of the calculation provided appropriate antidiffusive fluxes can be defined. This approach was used in a two-dimensional flux-corrected Lax–Wendroff code with success. In terms of stability properties, however, time-step-split algorithms are more lenient than their unsplit counterparts and the split algorithms are easier to code and use. Therefore we recommend splitting techniques as the easiest and most efficient approach to FCT, whenever the physics of the application permits. In practice, this means in all instances, except when a conservation condition or a differential identity (e.g., irrotationality; see below) must be strictly enforced.

In Cartesian coordinates a single transport module, separately called for propagation in the direction of each successive coordinate axis, is convenient. This is the basis for a number of codes written at NRL [7–9]. An example is SHAS2D, a fully compressible 2-D hydrodynamics code which solves the equations

$$(\partial \rho/\partial t) + \nabla \cdot (\rho \mathbf{v}) = 0; \tag{16}$$

$$(\partial/\partial t)(\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = -\nabla p; \tag{17}$$

$$(\partial E/\partial t) + \nabla \cdot [(p + E) \cdot \mathbf{v}] = 0; \tag{18}$$

$$(\partial \rho_{A,B}/\partial t) + \nabla \cdot (\rho_{A,B} \mathbf{v}) = 0, \tag{19}$$

where ρ_A and ρ_B are "marked" fluids carried around by the flow field $\mathbf{v}$. They satisfy

$$\rho_A + \rho_B = \rho, \tag{20}$$

while

$$p = (\gamma - 1)[E - \tfrac{1}{2}\rho v^2]. \tag{21}$$

In the Appendix it is shown how Eqs. (16)–(20) are differenced. The transport module, called SHASTX, happens to be a phoenical form of SHASTA (see Section II above), but it could as easily be replaced by implicit or explicit SHASTA, Lax–Wendroff, or some other FCT algorithm.

Curvilinear Coordinates

Codes performing analogous operations in $r - z$ and $r - \theta$ geometry have been written in similarly modularized fashion [9]. Some care must be taken in proper treatment of curvilinear scale factors (e.g., the radial weights in polar coordinates). In general it is not desirable to force the same module to perform the transport in all coordinate directions. If the motion is predominantly radial (as in an explosion) only the radial transport need be flux-corrected. Let us consider the problems associated with a one-dimensional radial continuity equation solver. The density ρ satisfies

$$(\partial\rho/\partial t) + r^{-\nu}(\partial/\partial r)(r^{\nu}\rho v) = 0; \tag{22}$$

$\nu + 1 =$ dimensionality (2 or 3). Denote the result of ordinary transport by $\{\rho_j^T)$ and the result of diffusive transport by $\{\rho_j^{DT}\}$. Then the definition of flux-corrected antidiffusion consistent with conservation is

$$\rho_j^1 = \rho_j^{TD} + (r_{j-1/2}/r_j)^{\nu} f_{j-1/2}^C - (r_{j+1/2}/r_j)^{\nu} f_{j+1/2}^C, \tag{23}$$

where

$$r_{j+1/2} = (r_j + r_{j+1})/2,$$

and

$$\begin{aligned} f_{j+1/2}^C = S \cdot \max\{0, \min[&S \cdot (r_j/r_{j-1/2})^{\nu} \Delta_{j-1/2}, |f_{j+1/2}|, \\ &S \cdot (r_{j+1}/r_{j-1/2})^{\nu} \Delta_{j+3/2}]\}. \end{aligned} \tag{24}$$

Here, $f_{j+1/2} = \eta(\rho_{j+1}^T - \rho_j^T)$, $\Delta_{j+1/2} = \rho_{j+1}^T - \rho_j^{TD}$, and $S = \text{sign}(f_{j+1/2})$. The same weights $(r_{j\pm1/2}/r_j)^{\nu}$ also appear in the definition of the diffusion. No other modifications of Cartesian FCT are needed. As with conventional algorithms the usual difficulties at $r = 0$ can be sidestepped if one uses a special prescription for $\nabla \cdot \mathbf{v}$ at the origin or displaces the mesh by a distance small compared with the mesh spacing r, so that the origin does not exactly coincide with the mesh point nearest $r = 0$. In general, cognizance must be taken of axes in the physical system. Special radial modules analogous to the SHASTX module described in the Appendix have been written specifically to provide more flexible and accurate treatments at the axis [9].

Irrotational Vector Felds

The problem of maintaining a divergence-free character in the numerical solution of field equations by finite differencing is not peculiar to FCT [10]. It arises in the case of incompressible flow, where the continuity equation becomes $\nabla \cdot \mathbf{v} = 0$, and in solving Maxwell's equation, where $\nabla \cdot \mathbf{B} = 0$ must hold. For the latter example in ideal magnetohydrodynamics, Faraday's law

$$\partial\mathbf{B}/\partial t = -c\nabla \times \mathbf{E} \tag{25}$$

becomes the inductive law

$$\partial\mathbf{B}/\partial t = \nabla \times (\mathbf{v} \times \mathbf{B}). \tag{26}$$

Equation (26) can be rewritten as

$$(\partial \mathbf{B}/\partial t) + \nabla \cdot (\mathbf{vB}) = -\mathbf{B}\nabla \cdot \mathbf{v}, \tag{27}$$

which resembles closely the fluid equations we have been considering. For the purpose of using FCT, however, the form (26) is more suitable in that $\nabla \cdot \mathbf{B} = 0$ is clearly maintained. In fact, Eq. (26) can be differenced as

$$\mathbf{B}^1 - \mathbf{B}^0 = -c\delta t[\nabla \times \mathbf{E}]^0.$$

If the finite difference approximations to the divergence and curl of a vector field $\mathbf{A}$ are denoted by $\nabla_{fd} \cdot \mathbf{A}$ and $\nabla_{fd} \times \mathbf{A}$, respectively, then we require

$$\nabla_{fd} \cdot \nabla_{fd} \times \mathbf{A} = 0$$

identically (to within round-off). This is trivial to secure in Cartesian coordinates; in curvilinear coordinates it often is not.

Now set

$$\mathbf{B}^{1/2} = \mathbf{B}^0 + (c\delta t/2)\nabla_{fd} \times \mathbf{E}^0; \tag{28}$$

use the material equations (e.g., continuity and momentum) to get $\mathbf{v}^{1/2}$; and define

$$\mathbf{B}^T = \mathbf{B}^0 - c\delta t\, \nabla_{fd} \times \mathbf{E}^{1/2},$$

where $\mathbf{E}^{1/2}$ is determined from $\mathbf{v}^{1/2}$ and $\mathbf{B}^{1/2}$, for example in ideal MHD by

$$\mathbf{E}^{1/2} = -(1/c)\, \mathbf{v}^{1/2} \times \mathbf{B}^{1/2}.$$

The array $\mathbf{B}^T$ represents a solution of Eq. (26) at the new timestep without flux-correction. It resembles Lax–Wendroff, but the expression Eq. (28) for the half timestep value uses $\mathbf{B}^0$ at the mesh point instead of an average. In spite of this, the scheme is generally stable subject to $v\delta t/\delta x < 1$, because $\mathbf{E}^{1/2}$ has to be found on the appropriate mesh by interpolation, a smoothing operation.

To apply phoenical FCT, we define

$$\mathbf{F}^0 = \eta\nabla_{fd} \times \mathbf{B}^0; \tag{29}$$

$$\mathbf{B}^{TD} = \mathbf{B}^T - \nabla_{fd} \times [\eta\nabla_{fd} \times \mathbf{B}^0]; \tag{30}$$

$$\mathbf{F}^1 = \eta\nabla_{fd} \times \mathbf{B}^T; \tag{31}$$

$$\mathbf{B}^1 = \mathbf{B}^{TD} + \nabla_{fd} \times \mathbf{F}^C, \tag{32}$$

where $\mathbf{F}^C$ is a corrected form of $\mathbf{F}^1$. Then $\mathbf{B}^1$ is the desired result and evidently satisfies $\nabla_{fd} \cdot \mathbf{B}^1 = \nabla_{fd} \cdot \mathbf{B}^0 = 0$. The quantity η is a scalar field, representing a variable diffusion coefficient. The definition of $\mathbf{F}^C$ in terms of $\mathbf{F}^1$ and $\mathbf{B}^{TD}$ is the only nontrivial step in the algorithm.

A given component of $\mathbf{F}^C$ in Eq. (32) affects two components of $\mathbf{B}^1$, say $B_1{}^1$ and $B_2{}^1$, through Eq. (23). If we define F^C using a form of Eq. (12) so that ripples in $B_1{}^1$ are suppressed, then $B_2{}^1$ can be badly behaved, and vice versa. No general method of preventing this in the present formulation has been found. The difficulty lies in the fact that there are only N components of $\mathbf{F}^1$ (N being the dimensionality), but N^2 sets of constraints imposed by the requirement of introducing no new extrema in any direction for any component of $\mathbf{B}$.

In some cases of interest this creates no real problem. For example, let coordinate 3 be ignorable and suppose flow occurs predominantly in the direction of coordinate 1. These assumptions are met for strong radial expansions modeled with a spherical or cylindrical 2-D $r - \theta$ code [4]. Now the only nonvanishing component of $\mathbf{F}$ is F_3 and ripples can only develop in B_2. The definition of F^C becomes

$$F^C = S \cdot \max\{0, \min[S \cdot \Delta_-, |F^1|, S \cdot \Delta_+]\}, \tag{33}$$

where $S = -\mathrm{sign}(F^1)$, and the $\Delta_\pm$ are differences of B_2^{TD}. The indices are indicated by ($\pm$) because their exact form depends on the grid staggering. Equation (33) is, of course, equivalent to Eq. (12).

Another approach is also possible. We can write $\mathbf{B}$ as the curl of a vector potential $\mathbf{A}$.

$$\mathbf{B} = \nabla_{fd} \times \mathbf{A}, \tag{34}$$

and the current $\mathbf{J}$ is in turn the curl of $\mathbf{B}$,

$$\mathbf{J} = \nabla_{fd} \times \mathbf{B}. \tag{35}$$

Taking the curl of Eq. (26) gives

$$\partial \mathbf{J}/\partial t = \nabla \times [\mathbf{v} \times (\mathbf{v} \times \mathbf{B})],$$

which can be rewritten in the form

$$(\partial \mathbf{J}/\partial t) + \nabla \cdot \mathbf{J}\mathbf{v} = \mathbf{G}(\mathbf{v}, \mathbf{B}), \tag{36}$$

where $\mathbf{G}$ is a complicated function of the velocity field $\mathbf{v}$, the magnetic field $\mathbf{B}$, and their derivatives. This is particularly useful in two-dimensions. In the ignored direction we get a 2-D scalar convective-like equation for B_3 which can be solved [illegible]

B_1 and B_2, can be determined from the current density J_3 via the vector potential which satisfies a Poisson equation

$$\nabla^2 A_3 = J_3 . \tag{37}$$

Thus B_1 and B_2 as finite difference derivatives of A_3 can be divergence free and yet, as integrals of J_3, they will be smooth and well behaved. Of course, the scalar equation for J_3 is solved by FCT methods for maximum accuracy.

VI. Conclusion

In the Introduction we listed three questions, the answers to which we believe are of interest to many workers in the field of computational fluid dynamics. Question 1 is primarily addressed in Sections II and IV, question 2 in Section III, and question 3 in Section V. In each case we have gone into what may seem to be needlessly grubby detail; our aim, however, has been to tell the reader what he needs to know in order to *use* FCT.

Consistent with this pragmatic approach, we feel that the best way to decided whether FCT is useful for a given fluid problem is to try it. This is readily done; as we have shown, it is usually trivial to "flux-correct" a conventional algorithm. Frequently the dramatic immediate improvement in the physicalness of results

TABLE III

Summary of results of the standard square wave test for the algorithms described in this paper

Algorithm	Mean absolute error
Donor cell	.260
Leapfrog (zero diffusion)	.245
Lax–Wendroff (zero diffusion)	.175
Flux-corrected diffusion	.082
Flux-corrected donor cell	.064
Flux-corrected leapfrog (1st version)	.074
Flux-corrected leapfrog (2nd version)	.057
SHASTA (explicit)	.057
SHASTA (implicit)	.049
SHASTA (phoenical)	.052

near steep gradients is all the comparison one needs. The fourth paper in the present series will describe the application of FCT to about a dozen such problems, along with the computational novelties encountered in the process. Some of these examples were intractable by all other techniques tried.

For the theoretically inclined, results of a test problem possessing an analytic solution may seem desirable. The square wave test used here and in FCT/1 is such a test, as are the shock wave and exploding diaphragm tests described in FCT/1. Rather than presenting the results of a large number of these problems, we here restrict ourselves to a compilation of the mean absolute errors found in the square wave test described in Section II (Table III). While it may be felt that such a test artificially oversimplifies the relative performance of various algorithms, this is generally the case when one imposes the constraint of analytic solubility.

In conclusion, we believe we have shown why FCT works and how to use it in a broad class of problems. We anticipate that other users will experience the same success.

APPENDIX: SHAS2D [8] AND SHASTX [9]

SHAS2D solves the equations of two-dimensional fully compressible hydrodynamics in an FCT finite-difference approximation. This numerical algorithm is specifically designed to handle high Mach-number shocks as well as smooth, nearly incompressible, low Mach-number flows. The physical equations solved by the basic code in conservation form are Eqs. (16)–(21) of Section V.

They are solved by a timestep-splitting procedure using a single subroutine called SHASTX which advances over one timestep a generalized one-dimensional continuity equation of the form

$$\partial F/\partial t = -(\partial/\partial x)(Fv) - S, \tag{A1}$$

where S is a given scalar conservative source term and where the velocity field v is assumed given and fixed throughout the cycle. If we denote by $\mathscr{S}[F, v, S, \delta l, \delta t]$ the "SHASTX" operator which advances $\{F_i\}$ by a timestep δt on a grid of spacing δl using given sequences $\{v_i\}$ and $\{S_i\}$, then a complete cycle of SHAS2D can be written as follows:

the x half cycle

$$\begin{aligned}
v_x{}^0 &\equiv (\rho v_x)^0/\rho^0 \text{ (similarly } v_y{}^0\text{)},\\
P^0 &\equiv (\gamma - 1)[E^0 - \tfrac{1}{2}\rho^0(v_x{}^0)^2 - \tfrac{1}{2}\rho^0(v_y{}^0)^2],\\
\rho^{hx} &= \mathscr{S}[\rho^0, v_x{}^0, 0, \delta x, \delta t/2],\\
(\rho v_x)^{hx} &= \mathscr{S}[(\rho v_x)^0, v_x{}^0, \partial P^0/\partial x, \delta x, \delta t/2],\\
(\rho v_y)^{hx} &= \mathscr{S}[(\rho v_y)^0, v_x{}^0, 0, \delta x, \delta t/2],\\
E^{hx} &= \mathscr{S}[E^0, v_x{}^0, (\partial(P^0 v_x{}^0)/\partial x), \delta x, \delta t/2];
\end{aligned} \tag{A2}$$

the x whole cycle

$$\begin{aligned}
v_x^{hx} &\equiv (\rho v_x)^{hx}/\rho^{hx} \text{ (similarly } v_y^{hx}),\\
P^{hx} &\equiv (\gamma - 1)[E^{hx} - \tfrac{1}{2}\rho^{hx}(v_x^{hx})^2 - \tfrac{1}{2}\rho^{hx}(v_y^{hx})^2],\\
\rho^x &= \mathscr{S}[\rho^0, v_x^{hx}, 0, \delta x, \delta t],\\
(\rho v_x)^x &= \mathscr{S}[(\rho v_x)^0, v_x^{hx}, (\partial P^{hx}/\partial x), \delta x, \delta t],\\
(\rho v_y)^x &= \mathscr{S}[(\rho v_y)^0, v_x^{hx}, 0, \delta x, \delta t],\\
E^x &= \mathscr{S}[E^0, v_x^{hx}, (\partial(P^{hx}v_x^{hx})/\partial x), \delta x, \delta t],\\
\rho_{A,B}^x &= \mathscr{S}[\rho_{A,B}^0, v_x^{hx}, 0, \delta x, \delta t];
\end{aligned} \tag{A3}$$

the y half cycle

$$\begin{aligned}
v_y{}^x &\equiv (\rho v_y)^x/\rho^x \text{ (similarly } v_x{}^x),\\
P^x &\equiv (\gamma - 1)[E^x - \tfrac{1}{2}\rho^x(v_x{}^x)^2 - \tfrac{1}{2}\rho^x(v_y{}^x)^2],\\
\rho^{hy} &= \mathscr{S}[\rho^x, v_y{}^x, 0, \delta y, \delta t/2],\\
(\rho v_x)^{hy} &= \mathscr{S}[(\rho v_x)^x, v_y{}^x, 0, \delta y, \delta t/2],\\
(\rho v_y)^{hy} &= \mathscr{S}[(\rho v_y)^x, v_y{}^x(\partial P^x/\partial y), \delta y, \delta t/2],\\
E^{hy} &= \mathscr{S}[E^x, v_y{}^x, (\partial(P^x v_y{}^x)/\partial y), \delta y, \delta t/2];
\end{aligned} \tag{A4}$$

and finally *the y whole step*

$$\begin{aligned}
v_y^{hy} &= (\rho v_y)^{hy}/\rho^{hy} \text{ (similarly } v_x^{hy}),\\
P^{hy} &= (\gamma - 1)[E^{hy} - \tfrac{1}{2}\rho^{hy}(v_x^{hy})^2 - \tfrac{1}{2}\rho^{hy}(v_y^{hy})^2],\\
(\rho v_x)^n &= \mathscr{S}[(\rho v_x)^x, v_y^{hy}, 0, \delta y, \delta t],\\
(\rho v_y)^n &= \mathscr{S}[(\rho v_y)^x, v_y^{hy}, (\partial P^{hy}/\partial y), \delta y, \delta t],\\
E^n &= \mathscr{S}[E^x, v_y^{hy}, (\partial(P^{hy}v_y^{hy})/\partial y), \delta y, \delta t],\\
\rho_{A,B}^n &= \mathscr{S}[\rho_{A,B}^x, v_y^{hy}, 0, \delta y, \delta t].
\end{aligned} \tag{A5}$$

In Eqs. (A2)–(A5) the superscript "*o*" stands for old, "*hx*" stands for half-step *x*, "*x*" stands for full step *x*, "*hy*" stands for half-step *y*, and "*n*" stands for new. At the completion of all these operations throughout the grid, all of the physical variables have been advanced in time by δt.

The time integration, as can be seen above, is a simple midpoint rule, second-order accurate, where the dependent driving variables $\mathbf{v}$ and P are determined at the half step by a forward differenced first-order algorithm. These half-step quantities are then used to advance all of the physical variables a full step in a time-centered manner. The overall algorithm given has been used effectively on a wide variety of difficult, long running problems.

Since there is no staggering in space or time in this code, the timestep can be varied freely from cycle to cycle. Of course the x half and whole steps consist of one-dimensional integrations of each row (fixed y) independently and the y half and whole steps consist of one-dimensional integrations of each column (fixed x) independently. In SHAS2D these individual integrations in both directions are performed by a single, highly optimized subroutine called SHASTX. The finite-difference formulae within the SHASTX routine itself are given below.

Suppose $\{F_i^0\}$, $\{v_i\}$, $\{S_i\}$, δx and δt are given for $i = 1, 2, \ldots, N$ on a single unstaggered spatial grid and boundary prescriptions are given for determining F_i and F_N using the intermediate values of F_i and the velocities. The following equations then define the $\mathscr{S}$ (SHASTX) operator used above in Eqs. (A2)–(A5) to solve Eq. (A1). First, untransported fluxes and a diffused (but untransported) solution are determined.

$$f_{i+1/2}^0 = \tfrac{1}{8}(F_{i+1}^0 - F_i^0), \tag{A6}$$

$$F_i^D = F_i^0 + (f_{i+1/2}^0 - f_{i-1/2}^0), \tag{A7}$$

where the "D" superscript stands for diffused.

Next, some transport convection factors are computed for each grid point. These factors need only be computed once for all of the physical variables at a given grid point.

$$\epsilon_i^\pm = \tfrac{1}{2} \pm v_i(\delta t/\delta x), \tag{A8}$$

$$Q_i^+ = \epsilon_i^-/(\epsilon_{i+1}^+ + \epsilon_i^-), \tag{A9}$$

$$Q_{i+1}^- = 1 - Q_i^+. \tag{A10}$$

Next a transported diffused solution is computed,

$$F_i^{TD} = 4[Q_i^+]^2 f_{i+1/2}^0 - 4[Q_i^-]^2 f_{i-1/2}^0 + Q_i^+(F_i^0 - S_{i+1/2}) + Q_i^-(F_i^0 - S_{i-1/2}), \tag{A11}$$

and then changes due to transport are computed as follows:

$$\delta F_i^T = F_i^{TD} - F_i^D. \tag{A12}$$

Using these changes due to transport, the fluxes F_i^0 are modified and differences in the transported diffused solutions are computed.

$$\delta F_{i+1/2}^{TD} = F_{i+1}^{TD} - F_i^{TD}, \tag{A13}$$

$$f_{i+1/2}^{T} = f_{i+1/2}^{0} + \tfrac{1}{8}(\delta F_{i+1}^{T} - \delta F_i^{T}). \tag{A14}$$

Using these fluxes and differences, the usual flux correction and antidiffusion is performed as follows:

$$\begin{aligned} f_{i+1/2}^{C} = {} & \operatorname{sgn}(f_{i+1/2}^{T}) \cdot \max\{0, \min[\operatorname{sgn}(f_{i+1/2}^{T}) \cdot \delta F_{i-1/2}^{TD}, \\ & \operatorname{abs}(f_{i+1/2}^{T}), \operatorname{sgn}(f_{i+1/2}^{T}) \cdot \delta F_{i+3/2}^{TD}]\}, \end{aligned} \tag{A15}$$

$$F_i^1 = F_i^{TD} - (f_{i+1/2}^{C} - f_{i-1/2}^{C}). \tag{A16}$$

Equation (A16) is the final flux-corrected transport solution for the cycle. The source term in Eq. (A1) appears in Eq. (A11). When S is $\partial P/\partial x$, for example in Eqs. (A2), $S_{i+1/2}$ is computed as follows:

$$S_{i+1/2} = (\delta t/2\delta x)[P_{i+1}^{0} - P_i^{0}]. \tag{A17}$$

Figure 10 is a simplified flow chart of the SHAS2D logic, reflecting the layout of the main program. After initializing the run and the control data, the input parameters, the physical problem parameters, and the physical variable arrays, the main loop is entered. Each cycle during the main timestep loop is checked for restart and dump conditions. Every cycle of the physical equations of motion begins with a computation of the longest timestep which will be numerically stable for the cycle. This computation is performed in DTSET.

Mr. E. Dent of NRL has constructed a Fortran compatible hand-coded version of SHASTX for the IBM 360/91 at ORNL. The Fortran reference version runs about 50% slower than this new version and now serves primarily to document the algorithm. Running time is also minimized by choosing the maximum stable timestep via the routine DTSET mentioned above and by very careful coding in all of the Fortran subroutines associated with performing the fluid-dynamics part of the calculation.

A series of timings of the pertinent parts of SHAS2D has been performed to help in timing analysis and in code optimization. The Fortran version of SHASTX required about 240μsec/grid point-cycle for the fluid dynamics and the hand-coded version has reduced this to ~160 μsec/grid point-cycle. The entire remainder of the physics calculations (variable selection, boundary-condition calculation, source computation, variable restoration, timestep calculation, and underflow elimination) also takes about 160 μsec/grid point-cycle. By far the largest amount of time here is spent in variable selection, source calculation, and variable restora-

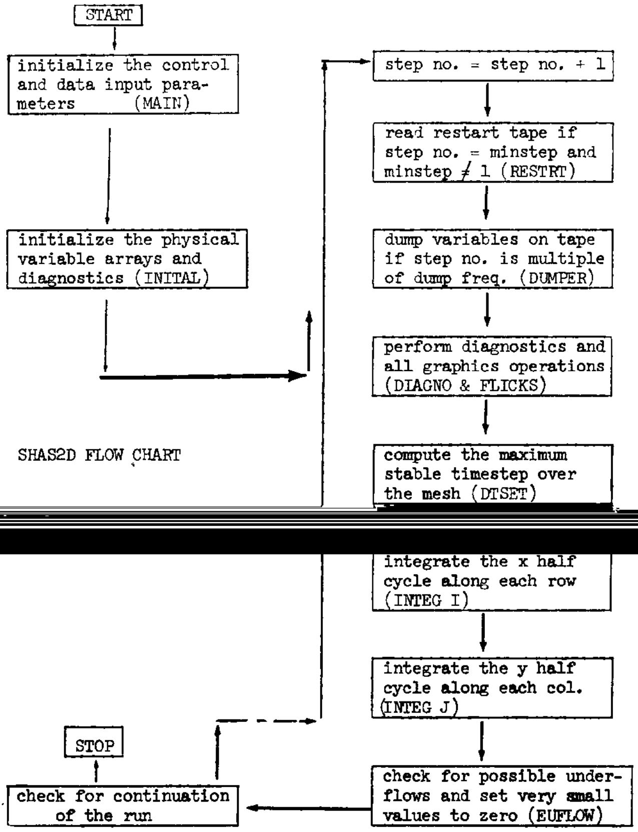
START
initialize the control and data input parameters (MAIN)
initialize the physical variable arrays and diagnostics (INITAL)
SHAS2D FLOW CHART
step no. = step no. + 1
read restart tape if step no. = minstep and minstep ≠ 1 (RESTRT)
dump variables on tape if step no. is multiple of dump freq. (DUMPER)
perform diagnostics and all graphics operations (DIAGNO & FLICKS)
compute the maximum stable timestep over the mesh (DTSET)
integrate the x half cycle along each row (INTEG I)
integrate the y half cycle along each col. (INTEG J)
check for possible underflows and set very small values to zero (EUFLOW)
check for continuation of the run
STOP

FIG. 10. SHAS2D flow chart. The main elements of the program control logic and physics calculations are shown as contained in the main program. INTEGI and INTEGJ contains the detailed solution of the physical equations as described in the Appendix.

tion. Thus a complete cycle on a 40 $\times$ 40 grid takes about .52 sec. For comparison, a Fortran two-dimensional Lax–Wendroff two-step FCT code was written which requires .54 sec/per timestep and must use timesteps roughly a factor $\sqrt{2}$ shorter.

ACKNOWLEDGMENT

This work was supported by the Defense Nuclear Agency.

REFERENCES

1. JAY P. BORIS AND DAVID L. BOOK, *J. Computational Phys.* **11** (1973), 38; referred to in the text as FCT/1.

2. JAY P. BORIS AND DAVID L. BOOK, to be published.
3. Originally suggested by D. V. Anderson and R. W. Clark, personal communication.
4. R. W. CLARK AND D. L. BOOK, *Phys. Fluids* **16** (1973), 720.
5. S. R. GOLDMAN, S. L. OSSAKOW, AND D. L. BOOK, *J. Geophys. Res.* **79** (1974), 1471; A. J. SCANNAPIECO, S. L. OSSAKOW, D. L. BOOK, B. E. MCDONALD, AND S. R. GOLDMAN, *J. Geophys. Res.* **79** (1974), 2913.
6. D. L. BOOK AND J. P. BORIS, to be published.
7. K. HAIN, Comparison of several two-dimensional computing schemes for exploding spheres in the atmosphere, DNA Atmospheric Effects Symposium, San Diego, April, 1973.
8. J. P. BORIS, NRL Memorandum Report 2542, December 1972.
9. J. P. BORIS, J. H. GARDNER, AND S. ZALESAK, Atmospheric hydrocodes using FCT algorithms, DNA Atmospheric Effects Symposium, San Diego, April, 1973.
10. See, for example, PATRICK J. ROACHE, "Computational Fluid Dynamics," Chap. III, Hermosa, Albuquerque, New Mexico, 1972.