JOURNAL OF COMPUTATIONAL PHYSICS 20, 397–431 (1976)

Flux-Corrected Transport. III. Minimal-Error FCT Algorithms

J. P. BORIS AND D. L. BOOK

Naval Research Laboratory, Washington, D.C. 20375

Received March 31, 1975; revised August 7, 1975

This paper presents an error analysis of numerical algorithms for solving the convective continuity equation using flux-corrected transport (FCT) techniques. The nature of numerical errors in Eulerian finite-difference solutions to the continuity equation is analyzed. The properties and intrinsic errors of an "optimal" algorithm are discussed and a flux-corrected form of such an algorithm is demonstrated for a restricted class of problems. This optimal FCT algorithm is applied to a model test problem and the error is monitored for comparison with more generally applicable algorithms. Several improved FCT algorithms are developed and judged against both standard flux-uncorrected transport algorithms and the optimal algorithm. These improved FCT algorithms are found to be four to eight times more accurate than standard non-FCT algorithms, nearly twice as accurate as the original SHASTA FCT algorithm, and approach the accuracy of the optimal algorithm.

I. INTRODUCTION

This paper presents an analysis of numerical errors in the solution of the convective continuity equation. Emphasis is placed on developing improved flux-corrected transport algorithms and on evaluating these and other algorithms relative to the performance of an "optimal" algorithm on simple test problems. Flux-corrected transport (FCT) is a technique for constructing positivity-preserving finite-difference algorithms [1–4] to solve generalized continuity equations for problems involving strong shocks or sharp shear-layer transitions where the characteristic scale lengths for the desired solution are as small as or smaller than the computational zone size. In such regions it is relatively meaningless to approximate a generalized continuity equation

$$(\partial\rho/\partial t) + \mathbf{v} \cdot \nabla\rho = -\rho\nabla \cdot \mathbf{v} + s(\mathbf{x}, t, \rho, \text{etc.}) \tag{1}$$

by standard linear Taylor series or finite-difference approaches because the error terms are the same size as the solution sought.

The FCT technique achieves its success by replacing strict asymptotic ordering expansions with the much more flexible physical fact that positivity must be maintained in the vicinity of strong gradients. Thus, flux-correction, when applied to a convected conserved quantity such as the mass density in hydrodynamics, is sufficient to guarantee that an everywhere nonnegative density profile remains nonnegative even near large gradients or in regions of strong evacuation. This important additional physical property is folded into the FCT algorithms through the flux-correction formula introduced in [1], hereafter referred to as FCT/I. Some generalizations of the FCT technique are introduced in [4], referred to as FCT/II, which extends the techniques to simple curvilinear geometries and basic transport algorithms. The present paper is primarily concerned with minimizing the remaining numerical error to develop improved FCT algorithms.

An FCT algorithm for solving Eq. (1) on a finite-difference grid consists conceptually of four sequential steps to advance the vector of grid density values $\{\rho_j(t)\}$ by one timestep to $\{\rho_j(t + \delta t)\}$. The first step consists of a highly diffusive approximation to the continuity equation:

$$\tilde{\rho}_j(t + \delta t) = \rho_j(t) + \delta t\, \dot{\rho}_j(t + \delta t/2) + \{\nu_{j+(1/2)}[\rho_{j+1}(t) - \rho_j(t)] - \nu_{j-(1/2)}[\rho_j(t) - \rho_{j-1}(t)]\}. \quad (2)$$

Here, $\dot{\rho}_j$ signifies a conservative finite-difference approximation to Eq. (1) and the last term is a strong added diffusion that, by itself, is an error with respect to Eq. (1). While $\dot{\rho}_j$ may contain errors of other types, we assume that all of the numerical error that can be classed as a simple three-point diffusion is included in the third term. The second step consists of computing some provisional fluxes preparatory to removing the three-point diffusive error term shown in Eq. (2). These fluxes, for explicit FCT, are

$$\tilde{\phi}_{j+(1/2)} = \nu_{j+(1/2)}[\tilde{\rho}_{j+1}(t + \delta t) - \tilde{\rho}_j(t + \delta t)]. \quad (3)$$

The fluxes are then corrected according to the following simple prescription in a third step:

The antidiffusion of $\{\tilde{\rho}_j\}$ using the corrected fluxes $\{\phi_{j+(1/2)}\}$ should generate no new maxima or minima in the solution, nor should it accentuate already existing extrema.

Equation (4) below is one particular mathematical formulation of this prescription and forms the basis of the FCT technique (FCT/I).

$$\phi_{j+(1/2)} = \operatorname{sgn} \tilde{\Delta}_{j+(1/2)} \max\{0, \min[\tilde{\Delta}_{j-(1/2)} \operatorname{sgn} \tilde{\Delta}_{j+(1/2)}, |\tilde{\phi}_{j+(1/2)}|, \tilde{\Delta}_{j+(3/2)} \operatorname{sgn} \tilde{\Delta}_{j+(1/2)}]\}, \quad (4)$$

where $\tilde{\Delta}_{j+(1/2)} \equiv \tilde{\rho}_{j+1}(t + \delta t) - \tilde{\rho}_j(t + \delta t)$. The fourth step performs the antidiffusion,

$$\rho_j(t + \delta t) = \tilde{\rho}_j(t + \delta t) - \phi_{j+(1/2)} + \phi_{j-(1/2)} . \tag{5}$$

Equation (5) removes that part of the diffusive error in Eq. (2) that does not lead to extraneous new maxima and minima in the solution.

Although new maxima and minima that would be caused by the corrective antidiffusive step are suppressed, physically valid new extrema arising from the $\dot{\rho}_j$ term in Eq. (2) appear naturally in the solution, because the change from $\{\tilde{\rho}_j\}$ to $\{\rho_j(t + \delta t)\}$ is checked for new extrema, not the change from $\{\rho_j(t)\}$ to $\{\rho_j(t + \delta t)\}$. The numerically induced new extrema that appear near strong gradients in non-FCT algorithms arise from two sources of error, dispersion at short wavelengths and intrinsic Gibbs phenomena brought on by the finite grid. FCT rectifies the most odious ramifications of these two types of error by leaving behind in selected regions a diffusive smoothing error, which is equal and opposite to the dispersive and Gibbs errors, and therefore cancels them. Near maxima or minima in a profile FCT leaves a residual diffusion that eventually replaces a one-point maximum or minimum with a three-point wide plateau (FCT/II). This plateau is impervious to further effect by FCT and is called "clipping." It is the price paid for the extraordinary stability and accuracy of FCT algorithms near strong gradients.

An interesting side effect of FCT diffusion and antidiffusion, as noted for the original FCT algorithm Shasta [1, 2], is the roughly fourfold reduction of dispersion in the basic finite-difference approximation itself. Even in the absence of the flux-correction formula Eq. (4), the process of strongly diffusing and then antidiffusing the solution greatly reduces the numerical errors. We will exploit this fact systematically.

Generalizations of the FCT technique to more complicated continuity-like equations, to multidimensions, to non-Cartesian geometries, and to complicated realistic magnetohydrodynamic and turbulence problems have been accomplished since the original document FCT/I was submitted for publication. Many of these are covered in [3] and the second paper in this series, FCT/II. Some of the current applications of the original FCT presentation and recent generalizations are discussed by Anderson [5], Book [6, 7], Boris [8, 9], Gardner [10], Liewer [11, 12] and Freeman [13].

Many different approaches have been tried to improve numerical solutions of continuity equations. These include characteristic methods [14], characteristic particle methods [15], finite element methods [16], splines [17], and finite-difference methods [18]. Characteristic and spectral methods give superb results in many cases, but lack general applicability. Characteristic methods, for example, usually break down when a real physical diffusion term is present, and spectral methods

can become expensive when complicated nonlinearities are present. Characteristic particle methods get over the nonlinear and diffusion-term stumbling blocks nicely, but run into massive amounts of computer time when the number of "particles" has to be made large to reduce fluctuations. Finite-element and spline methods also give excellent results where they are applicable, but their complexity and computational cost are often prohibitively high. Thus, attention naturally turns time after time to the simple, generally applicable, computationally efficient finite-difference formulations.

Our interest is centered primarily on Eulerian finite-difference solutions to Eq. (1) using a discrete grid of values to represent the mathematically continuous functions ρ, $\mathbf{v}$, and s. The Eulerian restriction is certainly not necessary to FCT ([4] and [8] describe Lagrangian FCT algorithms), but in many practical circumstances, the Eulerian approach is the simplest. In any case, Lagrangian techniques still have unavoidable Eulerian aspects. In ideal hydrodynamics, a grid moving with velocity $\mathbf{v}$ ensures that no mass crosses its cell boundaries, but the momentum and energy of the flow cross the Lagrangian boundaries. As is well known, the ideal hydrodynamic energy equation,

$$\partial E/\partial t = -\nabla \cdot [(E + P)\,\mathbf{v}], \tag{6}$$

actually has associated with it a Lagrangian velocity

$$\mathbf{v}^* = ((E + P)/E)\,\mathbf{v}, \tag{7}$$

which is parallel to and somewhat larger than $\mathbf{v}$. No energy is transported across cell boundaries moving at v^*. Of course, if the grid moves at $\mathbf{v}^*$, so that the energy is Lagrangian, the density *cannot* be Lagrangian, since $\mathbf{v}^*$ differs from $\mathbf{v}$ whenever $P > 0$. In other words, convective flows across cell boundaries present almost as big a problem for Lagrangian finite-difference algorithms as for Eulerian.

We concentrate primarily on the one-dimensional incompressible continuity equation, often called the advective equation, for $\rho(x, t)$ with a given constant velocity on a uniform spatial grid. Numerical algorithms for this problem can be analyzed theoretically in great detail. Fourier analysis of the constant-velocity version of Eq. (1) is introduced in Section II, as a mechanism for cataloguing three types of numerical error that can be identified. In Section II, we also discuss briefly a model test problem for the one-dimensional incompressible continuity equation. This square-wave problem is used to compare various algorithms in a fashion consistent with previous reports.

Section III considers the properties of an optimal algorithm for the model test problem based on Fourier-transforming the incompressible (constant-velocity) continuity equation. This algorithm, while quite restricted in its applicability, suffers no amplitude or dispersion errors. Thus, it forms an ideal method of

exploring the irreducible finite-interval Gibbs error relative to more generally applicable algorithms.

Having isolated and displayed an essentially irreducible error, we undertake a treatment of amplitude errors in Section IV. We develop several zero-residual-diffusion algorithms appropriate for use with FCT whose linear amplification factors are identically unity for all Fourier harmonics. These algorithms help to clarify the relationship between antidiffusion and amplitude errors, but do not reduce the error level from that found in the original explicit Shasta algorithm [1–3]. This result suggests that phase errors due to dispersion account for the dominant inaccuracies in finite-difference algorithms.

Section V presents several phase-improved generally applicable FCT algorithms, which show that, even in the presence of nonzero residual diffusion, overall errors can be made almost as small as for the optimal algorithm of Section III. Phase improvement is achieved by adjusting the level of diffusion and antidiffusion so that phase errors are reduced in the overall algorithm from second order in $k\,\delta x$, to fourth order. Two extremely good algorithms are derived and compared.

Section VI briefly extends the consideration of FCT algorithms to include some aspects of linear and nonlinear stability.

Section VII presents a brief summary and discusses the following six conclusions, the main results of this paper.

1. An optimal algorithm is not perfect because of essentially irreducible finite-interval Gibbs effects.

2. The optimal Fourier FCT algorithm actually can be approached quite closely by useable finite-difference FCT algorithms.

3. Modest phase errors are generally more destructive than modest amplitude errors because the former can be secular.

4. Improved phase-error FCT algorithms with modest but nonzero residual diffusion appear to be quite sufficient.

5. The good FCT algorithms are about an order of magnitude more accurate than more-or-less standard non-FCT algorithms.

6. Flux-corrected transport seems to strongly repress or eliminate the nonlinear instabilities that result in grid-separation errors in many long runs using standard algorithms.

For purposes of clarity and continuity in exposition, the text of this paper consists mostly of discussion of the concepts of optimal algorithms and phase and amplitude improvements. As far as possible, the numerical analysis on which the paper rests is kept to the background, in the form of an Appendix, which may be omitted in a first reading and referred to as necessary.

II. Errors in Solving the Continuity Equation by Finite Differences

We now consider the types of error inherent in numerical solutions of the continuity equation (1), which can be written in conservative form without sources as

$$\partial\rho/\partial t = -\nabla \cdot (\rho \mathbf{v}). \tag{8}$$

Here, $\rho(\mathbf{x}, t)$ is the density function, which in general, depends on position and time, and $\mathbf{v}(x, t)$ is a velocity flow field for that density. For our analysis, we consider the one-dimensional version of Eq. (8) with $\mathbf{v}(\mathbf{x}, t)$ a given constant V_0. One then knows the analytic solution of Eq. (8),

$$\rho(x, t) = \rho(x - V_0 t, 0), \tag{9}$$

given the initial profile $\rho(x, 0)$.

We assume that our knowledge of $\rho(x, 0)$ is itself limited, extending only to the initial values of ρ on a set of N discrete gridpoints with separation δx. These values we denote as $\{\rho_j{}^0\}$ $(0 \leqslant j \leqslant N - 1)$, where $\rho_N{}^0 = \rho_0{}^0$. Furthermore, we only expect our finite-difference algorithms to yield approximations to $\rho(x, n\,\delta t)$ at discrete times $t = 0, \delta t, 2\,\delta t, \ldots$ on the same discrete spatial grid; these approximate solutions of Eq. (8) we denote by $\{\rho_j{}^n\}$.

Since Eq. (8) is linear with constant coefficients under the above assumptions and has periodic boundary conditions, one can Fourier analyze $\rho(x, t)$ in space:

$$\rho(x, t) = \sum_{k=-\infty}^{\infty} \hat{\rho}_k(t)\, e^{(2\pi i k x/L)}, \tag{10}$$

where $L \equiv N\,\delta x$ is the length of the system. The infinite discrete set of harmonic coefficients $\{\hat{\rho}_k(t)\}$ can be found as an integral over the continuous density function $\rho(x, t)$,

$$\hat{\rho}_k(t) \equiv (1/L) \int_0^L \rho(x', t)\, e^{(-2\pi i k x'/L)}\, dx', \tag{11}$$

where it is clear that $\hat{\rho}_k(t) = \hat{\rho}_{-k}(t)$ when $\rho(x, t)$ is real. This Fourier analysis discretizes simply as follows:

$$\rho_j{}^n = \sum_{k=(-N/2)+1}^{N/2} \hat{\rho}_k{}^n e^{(2\pi i k j \delta x/L)}, \tag{12}$$

where

$$\hat{\rho}_k{}^n \equiv \frac{1}{N} \sum_{j=0}^{N-1} \rho_j{}^n e^{(-2\pi i k j \delta x/L)}. \tag{13}$$

From the reality of $\{\rho_j{}^n\}$, we get all the negative k harmonics by conjugacy.

The advantage of Fourier analysis lies in the particularly simple form of the closed solution, Eq. (9), in k-space.

$$\hat{\rho}_k(t) = \hat{\rho}_k(0)\, e^{(2\pi i k V_0 t/L)}. \tag{14}$$

The Fourier harmonics of $\rho(x, t)$ advance uniformly in phase with an unchanging amplitude for all time. Furthermore, since the equation being studied is linear, it is sufficient to know the behavior of each harmonic independently; a complicated profile can be built up by superposition. In terms of the finite discrete harmonics of Eq. (13), which we assume to represent a numerical approximation to $\rho(x, t)$ on a grid hereafter, we note that the time-discretized analog to Eq. (14),

$$\hat{\rho}_k{}^n = \hat{\rho}_k{}^0 e^{(2\pi i k V_0 n \delta t/L)}, \tag{15}$$

would only be true if the algorithm were optimal.

In general, Eq. (15) does not hold for a finite-difference algorithm, so it is valuable to catalogue the types of numerical error that can occur. The first such error, one that dominates many numerical solutions of the continuity equation that must display a nonnegative solution, is damping. Numerical diffusion is a particular form of harmonic damping that arises from often unintended approximations to the second-order diffusion equation,

$$\partial\rho/\partial t = D(\partial^2\rho/\partial x^2), \tag{16}$$

such as the simple three-point difference formula,

$$\rho_j{}^1 = \rho_j{}^0 + (D\ \delta t/\delta x^2)[\rho_{j+1}^0 - 2\rho_j{}^0 + \rho_{j-1}^0]. \tag{17}$$

The measure of damping for each algorithm is a quantity called the amplification factor,

$$A_k \equiv |\ \hat{\rho}_k^{n+1}/\hat{\rho}_k{}^n\ |, \tag{18}$$

which can be defined for each harmonic. When the harmonic is damped $A_k < 1$. When the harmonic is unstable $A_k > 1$. When the algorithm has zero residual damping (ZRD), $A_k = 1$ for all Fourier harmonics.

The second source of error that can plague numerical solutions of the continuity equation is numerical dispersion. Dispersive errors occur when the phase velocity of some harmonics differs from the velocity of the flow V_0 in Eq. (15). This type of error occurs because the spatial and temporal derivatives in Eq. (8) are being approximated by finite differences, and hence, the harmonic phase velocity is

no longer independent of the harmonic index k. To study the numerical dispersion, we define a relative phase error

$$R(k) \equiv \frac{V_\phi(k) - V_0}{V_0}, \tag{19}$$

where $V_\phi(k)$ is the numerical phase velocity of harmonic k and V_0 is the fluid velocity.

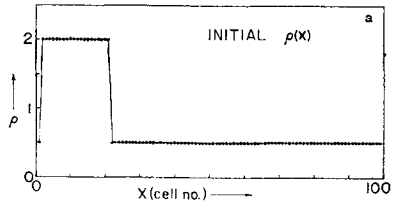
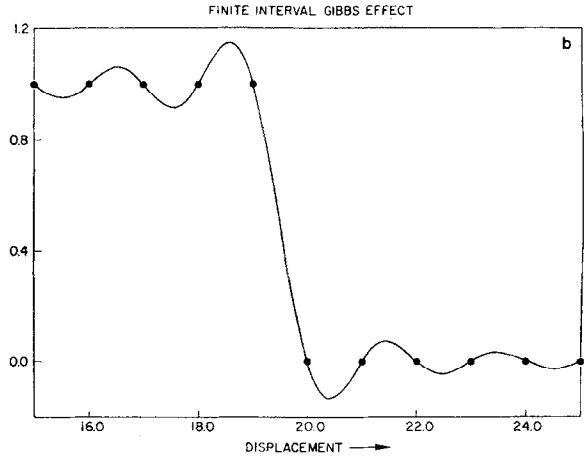


FIG. 1(a). Initial conditions for the square wave test problem. The velocity is taken to be constant in space and time and the boundary conditions on the 100-point mesh are periodic. The density square wave of height 2.0 is 20 cells across. The background density is 0.5. The square wave should propagate across the grid unchanged.



(b). The Gibbs phenomenon at the edge of a square wave. The points show the numerical values of density for the leading edge of a square wave with transition from 0 to 1. The solid line is the continuous finite Fourier expansion which passes through all the grid values. This Fourier expression is the smoothest function passing through all the grid values and displays overshoots and undershoots. These oscillations, intrinsic to the finite grid representation, can force negative density, and hence, must be numerically truncated to faithfully represent a physical density.

The third type of error observed in solving Eq. (8) on a discrete grid in space and time is associated with the Gibbs phenomenon. Errors of this type arise because $\rho(x, 0)$ is known at only N distinct points in space, and hence, only N distinct harmonics can contribute. The behavior between the gridpoints, undetermined in principle, actually is specified completely once the Fourier spectrum is truncated as in Eq. (12). The truncation of the spectrum amounts to saying that the unknown behavior of $\rho(x, 0)$ between the gridpoints should be as "smooth" as possible. This means that oscillations between the (finite number of) gridpoints cannot be avoided.

We will use the squarewave test problem introduced in FCT/I to illustrate the x-space effects of our various algorithms and algorithm modifications. Through this vehicle, the qualitative effects of damping and dispersion on each individual harmonic can be demonstrated as well as the overall effects on a given profile. As seen in Fig. 1(a), the squarewave test has 100 gridpoints in periodic geometry. A square step 20 points wide is propagated at a constant velocity $V_0 = 1$. Initially the 100 values $\{\rho_j^0\}$ $(j = 1, 2, ..., 100)$ are as shown. At each cycle of a test calculation 100 new values $\{\rho_j^n\}$ have to be determined unambiguously. The measure of error we will use for these tests is the average absolute error

$$\text{A.E.} \equiv (1/100) \sum_{j=1}^{100} | \rho_j^n - \rho_j^{\text{analytic}}(n\ \delta t)|. \tag{20}$$

Figure 1(b) demonstrates the Gibbs phenomenon for this problem by plotting the continuous function $\rho(x, 0)$ corresponding to the discrete values of the squarewave test problem. The Gibbs oscillations between the gridpoints are clearly seen with peak overshoots and undershoots of about 15%. If left uncorrected, these oscillations could make an algorithm negative even in the complete absence of harmonic damping or dispersion. The Gibbs oscillations represent an irreducible error that cannot be eliminated by improvement of the numerical algorithm alone; the grid must be refined as well.

Figures 2(a) and 2(b) compare leapfrog and Lax–Wendroff algorithms with and without added numerical smoothing to show the relative effects of dispersion and damping. When $\nu = 0$ in both cases, the added damping is zero. The dispersive errors dominate and nonpositive transport occurs with large errors. Somewhat smaller values of A.E. are achieved by adding a nonphysical damping as shown. The large errors from short wavelength dispersion are reduced when these improperly calculated harmonics are nonphysically damped to low amplitude. Too much damping, however, is generally the bane of numerical algorithms. Figure 3(a) shows the donor cell algorithm (flux uncorrected) on the test problem. This algorithm, as seen in FCT/I, has second order relative phase errors, but has strong first-order numerical damping coefficients. The value of A.E. achieved,

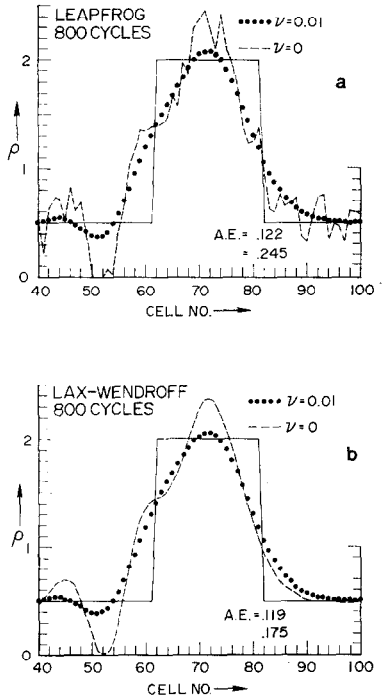


FIG. 2. Comparison of leapfrog and Lax–Wendroff algorithms on the square wave test. (a) The reversible leapfrog algorithm with and without additional diffusive damping. (b) The Lax–Wendroff algorithm with and without additional diffusive damping. The secular short wavelength dispersion errors are so bad that added error in the form of nonphysical damping actually improves the solution (smaller values of the average absolute error, A.E.)

0.260, is worse than the standard damped Lax–Wendroff and leapfrog algorithms (A.E. = 0.175 and A.E. = 0.245, respectively) and far worse than the reversible FCT algorithm shown for comparison in Fig. 3(b).

In fact, this very broad comparison of these standard treatments with the FCT solution graphically illustrates the reason for such interest in FCT algorithms. In FCT/I, Figs. 10 and 11, we see the corresponding calculations for the explicit and implicit SHASTA algorithms. For these early FCT algorithms, A.E. = 0.057, and A.E. = 0.049, respectively. Figure 5 of FCT/II shows a flux-corrected explicit donor cell, which has A.E. = 0.064, again a big improvement.

The remainder of the paper is devoted to determining the relative importance of the Gibbs error, amplitude errors, and dispersion errors, and to developing modified FCT finite-difference algorithms to eliminate or minimize them. We feel that one should try to enforce the following six constraints in constructing these finite-difference algorithms.

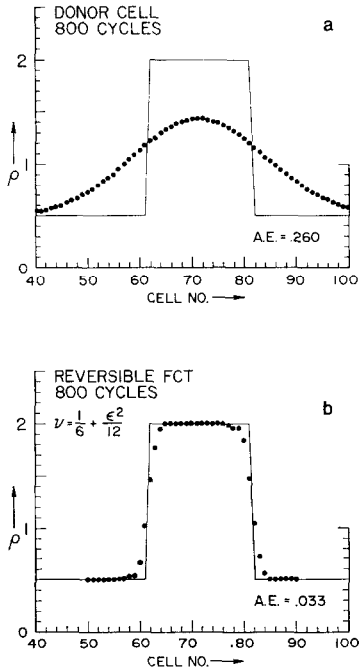


FIG. 3. Comparison of simple donor cell and a reversible FCT algorithm on the square wave test. (a) The simple donor cell algorithm is second-order accurate in phase but has a strong first-order diffusion and a correspondingly huge value of A.E. (b) The reversible FCT algorithm, here with the diffusion coefficient ν chosen to minimize phase errors, gives the lowest error of any finite-difference algorithm tested to date. A.E. = 0.033 is within 50% of optimum and 8 times smaller than the donor cell value A.E. = 0.260.

1. Exact conservation properties of the partial differential equations should be mirrored in the finite-difference approximations. Thus, where a conserved integral of the system exists, such as total mass or total energy, a corresponding numerical integral or sum should be exactly conserved.

2. The algorithm should ensure stability of all the harmonics in some useful range of the parameters δx and δt. If even one harmonic is linearly unstable, having an amplification factor greater than unity, the long term accuracy of the algorithm is suspect.

3. The algorithm should be effectively single- or double-step in the temporal integration. If an algorithm requires past values of the dependent variables at several time levels, the storage requirements of the algorithm can become prohibitive when generalized to multidimensional calculations.

4. The overall algorithm should be at least second order in regions of the problem where the concept of order is related usefully to accuracy. The relative phase errors should decrease for long wavelengths at least as $(k\,\delta x)^2$, and the amplification should generally correspond to a numerical damping coefficient no larger than $\epsilon^2 = (v\,\delta t/\delta x)^2$. This requirement is included to provide at least a minimal long term accuracy.

5. The nonnegative property of convective and continuity equations should be maintained by the algorithm wherever applicable. This important aspect of these equations is generally ignored by most algorithms. Its maintenance by FCT algorithms is the major source of their success. Generally speaking, enforcing this condition by a conservative linear diffusion operator leads to too much numerical diffusion to satisfy requirement 4.

6. The algorithm should not be built around special (and rather singular) properties, such as giving exactly the correct answer when $\epsilon \equiv v\,\delta t/\delta x = 1$. The algorithm should be generalizable to variable flow velocities and grid spacing, and should not be problem-dependent.

III. An Optimal Algorithm, Fourier FCT

The three sorts of numerical error that were identified in Section II should be reduced as much as possible in order to generate an optimal algorithm for transporting a density structure across an Eulerian grid. As we have seen, the Fourier transform allows complete control over the phase and amplitude of the harmonic components of the density structure being transported when the grid is uniform and the velocity field is constant. Therefore, it is not surprising that the Fourier transform forms the basis for developing what we might call an optimal algorithm for treating the advective continuity equation. We first Fourier transform the density field $\rho(x, t)$, and then advance each of the Fourier harmonics according to Eq. (14). These new and accurate harmonics then can be Fourier synthesized on the finite-difference grid according to the inverse-Fourier transform formula Eq. (12).

If the quantity $V_0\,\delta t$ is an integral-multiple of δx, the solution is exact. The situation is not so good when the distribution is transported a fraction of a cell. Because of the truncated Fourier representation, the Gibbs phenomenon is operative. Transport over a fraction of a cell brings out the hidden oscillations of the function that exist between the specified gridpoints. Thus, although the Fourier transform has zero residual damping and zero phase error, the irreducible Gibbs phenomenon requires us to enforce nonnegativity on the solution. Our optimal algorithm, therefore, combines the good amplitude and phase properties of the Fourier transform with the properties of flux-corrected transport to maintain a

nonnegative solution. If the Fourier solution is diffused at each cycle, and then antidiffused by equal amounts, the antidiffusion being performed implicitly, the final solution in the linear sense still has zero residual damping and zero phase error. Properly applied, the diffusion will eliminate the nonnegativity caused by the Gibbs oscillations and the flux-correction formula applied to the antidiffusive fluxes will then insure against new maxima or minima. The effect of the flux-correction formula will be to leave a residual diffusion near sharp gradients to remove the nonnegative tendencies inherent in any finite-grid representation. The asymptotic order of the solution will no longer be infinite as with the pure Fourier transform, but the solution itself will be much more reasonable and reliable in the context of physically complicated calculations.

Another way of looking at this algorithm is to consider a given profile of densities to be transported and then to ask what is the closest discrete approximation for which the Fourier synthesis has no extra maxima or minima between the grid-points. If we were to take the desired profile and replace it with the smoothed profile (whose Fourier interpolation is monotonic between the gridpoints), the Fourier transform would then be a "perfect algorithm" for solving the advective equation. No new maxima or minima could be generated, the phase errors and amplitude errors would be nonexistent, and the algorithm would be reversible so that the original solution could always be reconstructed after many cycles from the final solution.

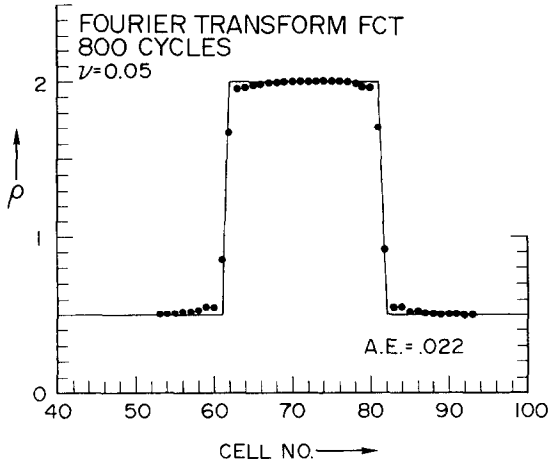


FIG. 4. Use of the optimal Fourier transform FCT algorithm on the square wave test. With $\nu = \frac{1}{20}$ after 800 cycles the absolute error A.E. = 0.022, the smallest value obtained with any positive algorithm. This represents (roughly) a minimal error since phases and amplitudes are treated exactly. The only inaccuracy arises from the finite discrete representation of physically continuous functions.

Clearly, the smaller the diffusion and antidiffusion in this limited but optimal algorithm, consistent with nonnegativity, the better. However, we found by test calculations that the final solution and the asymptotic value of A.E. are quite insensitive to the level of diffusion and antidiffusion. Figure 4 shows results of using the optimal algorithm on the square-wave test problem using $\nu = 0.05$. The value of A.E. for these calculations is about 0.022, more than 10 times better than the standard algorithms previously tested. If flux correction had not been used with the Fourier algorithm, the errors would have been oscillatory as the solution moved through the grid, going to zero whenever the profile had moved an integral number of gridpoints. The mean values of A.E., however, would have been about as large as those determined using the optimal algorithm with FCT and the solution would have been nonpositive through much of the calculation.

We certainly do not mean to imply by these test calculations that the Fourier transform with FCT is the best algorithm to use in general. Transformation techniques are very complicated to use when a nonuniform mesh is considered, or when the solution and the equations have strong nonlinearities. A very good example would be a simple calculation with spatially varying velocity. While our Fourier transforming would still be possible, the interaction of the nonlinear terms would certainly confuse the concepts of phase and amplitude significantly. Furthermore, and this is the strongest reason for not relying more heavily on transform techniques, the expense of performing such transform calculations far exceeds the gains that can be realized over good finite-difference algorithms. The Fourier approach is basically one of performing an N-point approximation to the derivatives, where N is the number of cells in the system. Even using the specialized folding techniques of the fast Fourier transform still leaves a very expensive process.

Rather, this optimal algorithm is included to point up the importance and the irreducible nature of the Gibbs phenomenon. Having evaluated the effective magnitude of the Gibbs errors, we have a realistic basis for comparison with more flexible, inexpensive finite-difference algorithms.

IV. Zero Residual Damping Algorithms

All of the algorithms discussed here are given in Appendix A in FCT form with an unspecified diffusion coefficient ν and an unspecified antidiffusion coefficient μ. We now wish to use some of this built-in freedom to improve the accuracy of the calculations. Although the analyses of Sections IV and V are linear, the test calculations are actually performed using the full FCT algorithm. ("Cheating" to take advantage of the linearity would make meaningful comparisons impossible; cf. requirement 6 of Section II.) Heavy use of the Appendix here allows a more transparent development of the various ideas we wish to test.

The algorithms we consider are all basically three-point algorithms in the diffusive transport stage. In other words,

$$\hat{\rho}_k^{\,1} = a_j\rho_{j+1}^0 + b_j\rho_j^{\,0} + c_j\rho_{j-1}^0 \tag{21}$$

for diffusive transport. There are three free parameters here. There is also another free parameter, the antidiffusion coefficient μ, making a total of four that can be chosen to give the resulting algorithms up to four desirable properties. Two of the four properties are locked in. One free parameter must be chosen to ensure conservation and the second must be chosen to ensure that the advection term is accurate. The remaining two parameters are available to establish further important properties for the algorithms.

In this section, we look at algorithms that minimize the residual linear damping left by the diffusion and antidiffusion stages. The idea here is that numerical diffusion has invalidated many convection calculations. How small can this be made, consistent with linear stability of all harmonics? Ideally, $|A|^2 = 1$ is sought for all allowed values of β and ϵ, in which case the term zero residual damping (ZRD) is applied.

1. *Shasta (Lax–Wendroff) ZRD*

Equation (A.10) gives the Shasta transfer function for explicit and implicit antidiffusion (since our test problem is linear, everything said here and in the following for Shasta applies equally well for Lax–Wendroff). The corresponding amplification factor squared given in Eq. (A.11), consists of the product of two terms. If the remaining parameters ν and μ are properly chosen in Eq. (A.11) the dependence on β and ϵ can be made to cancel out entirely provided implicit antidiffusion (minus signs in exponents) is also used. The value of ν is chosen to complete the square of the first term in (A.11) in curly brackets. That term is then in the form of a diffusion term squared and μ can be chosen to give exactly the same antidiffusion. Thus

$$\nu = \tfrac{1}{4}(1 - \epsilon^2) \tag{22a}$$

and

$$\mu = \tfrac{1}{4}(1 - \epsilon^2), \tag{22b}$$

are the appropriate coefficients for Shasta ZRD. When plugged into Eq. (A.11), these yield

$$|A|^2_{\mathrm{SHA}} = \frac{[1 - \tfrac{1}{2}(1 - \epsilon^2)(1 - \cos\beta)]^2}{[1 - \tfrac{1}{2}(1 - \epsilon^2)(1 - \cos\beta)]^2} = 1. \tag{23}$$

Thus, the choices (22a) and (22b) for ν and μ give the desired result of zero residual damping.

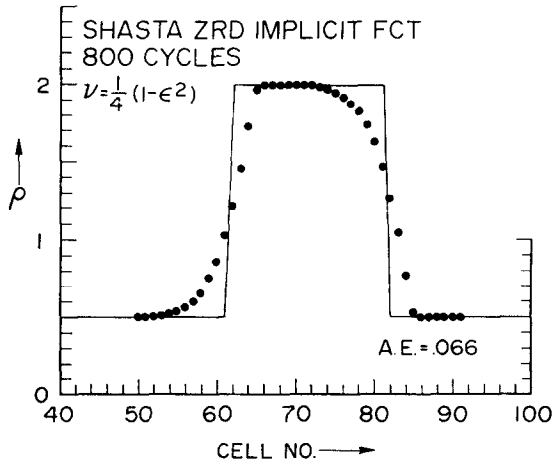


FIG. 5. Result of using Shasta ZRD on the square wave test. Implicit antidiffusion is required for zero residual damping and ν is chosen to complete the square in the squared amplification factor so that two equal implicit antidiffusion steps can exactly cancel the damping. The Shasta ZRD result, A.E. = 0.066, is worse than simple FCT algorithm because even though the residual damping is zero, the phase properties are much worse than in the simpler Shasta algorithms.

Figure 5 shows the result of using this algorithm on our square wave test problem. The average absolute error is actually worse than in previous cases even though there is no linear residual damping. This result is puzzling until one considers the effect of Eqs. (22) on the dispersion errors. The original Shasta algorithm used $\nu = \mu = \frac{1}{8}$, which was found to have very good effects on the phase errors. In FCT/I, the form of the relative phase error for long wavelengths was found to be $R = \beta^2[-(1/24) + (\epsilon^2/6)] + O(\beta^4)$, when $\nu = (1/8)$. When $\nu = (1/4)(1 - \epsilon^2)$, we find now

$$R_{\text{SHA}} = \tfrac{1}{12}[1 - \epsilon^2]\,\beta^2 + O(\beta^4) \tag{24}$$

The phase errors are now about twice as large as in the original algorithm even though the amplitude errors are reduced. Furthermore, the phase errors now have the opposite sign, the import of which can be seen by comparing Fig. 5 and Fig. 2 of FCT/II. The difference in sign of the relative phase errors causes the two figures to look rather like mirror images. The ZRD Shasta algorithm has A.E. = 0.066, about 35% more than the original implicit version, which has A.E. = 0.049.

Another factor, besides the increased phase errors, is the increased diffusion coefficient. We are diffusing and antidiffusing by much more than is actually necessary to maintain positivity. Thus, the residual nonlinear errors introduced by the flux corrector also can be somewhat larger. The conclusion one draws immediately is that good phase properties are more important than ideal amplitude properties. This conclusion is reinforced throughout the remainder of the paper.

2. *Donor Cell ZRD*

A zero residual damping algorithm can also be constructed around implicit flux-corrected donor cell (FCT/II). Again, ν and μ in Eq. (A.15) are chosen to eliminate the variations of $|A|^2_{DC}$ with β and ϵ. The resulting values are

$$\nu = \tfrac{1}{4}(1 - \epsilon)^2 \tag{25a}$$

and

$$\mu = \tfrac{1}{4}(1 - \epsilon^2), \tag{25b}$$

and they ensure

$$|A|^2_{DC} = 1 \tag{26}$$

for all allowed β and ϵ.

Donor cell FCT is another three-point algorithm with four free parameters. Since we are imposing the same restrictions as those used above for Shasta, it is not surprising that the linear transfer functions for the two cases become identical. We find in both cases

$$A_{DC} = A_{SHA} = \frac{[1 - \tfrac{1}{2}(1 + \epsilon^2)(1 - \cos\beta) - i\epsilon \sin\beta]}{[1 - \tfrac{1}{2}(1 - \epsilon^2)(1 - \cos\beta)]}. \tag{27}$$

The nonlinear properties, of course, are different for the two cases so the algorithms as actually used in a complicated code will differ somewhat.

3. *Phoenical Algorithms (no ZRD)*

The point of phoenical antidiffusion, introduced in FCT/II, is to remove the tridiagonal sweep needed in implicit algorithms. The transfer functions (A.18) and (A.22) are such that $|A|_{PSH} = 1$ and $|A|_{PDC} = 1$ identically when $\epsilon = 0$. There is no residual damping when there is no flow. This is a third constraint $\nu = \mu$, which has been built into the phoenical algorithms so there remains only one free parameter, not enough to eliminate β *and* ϵ dependences of $|A|^2$. Thus, ZRD versions of three-point phoenical algorithms are not generally possible. The fourth parameter has to be chosen to minimize amplitude errors, for example, at long wavelength. It turns out that

$$\nu = \tfrac{1}{4}(1 - \epsilon^2) \tag{28}$$

also minimizes the long wavelength amplitude errors in phoenical Shasta.

4. *Reversible FCT (ZRD)*

Equations (A.26–A.30) in Appendix A, describe a reversible algorithm for transport, where ν is now a free parameter. Because the algorithm is time symmetric, Eq. (A.28) holds for all ϵ, β, and ν anyway. The algorithm is truly ZRD, and ν is left free to adjust the phase properties as discussed in the next section.

In summary, zero residual damping (ZRD) versions of FCT algorithms are possible, based on several different three-point transport algorithms. Extra velocity-dependent diffusion is added to the diffusive transport to put the amplification factors in the form of a perfect square of two diffusion steps. The implicit anti-diffusion then leaves $|A|^2 = 1$ identically. The phoenical algorithms cannot be treated in this way, but a reversible ZRD algorithm is possible in which the diffusion coefficient ν is (as yet) unspecified. Our comparison of the ZRD algorithms with other calculations have shown conclusively, however, that phase errors due to numerical dispersion are generally more serious than residual amplitude errors and the irreducible Gibbs phenomenon combined.

V. Low Phase Error Algorithms

It should not be surprising that the phase properties of an algorithm are more important than the usual sorts of amplitude errors. Damping generally leaves the long wavelengths untouched while removing the very short wavelengths. Since these short wavelength harmonics of the solution generally suffer the most dispersion anyway, damping at the short wavelengths can sometimes actually reduce the overall A.E. in conjunction with dispersion. The phase properties are practically more important because phase errors grow secularly when the velocity is predominantly in one direction. The difference in position between the correct phase front and the numerically computed phase front increases linearly in time when the velocity is constant.

Clearly, the phase errors are not secular when the velocity is oscillatory and the distance of oscillatory motion is small compared to spatial wavelengths of interest. In such specialized situations, the phase errors increase in one direction for half a cycle and then increase in the other for half a cycle. The net integrated phase errors go to zero on the average, making amplitude errors (damping) the major remaining source of numerical error. In such special situations, the ZRD algorithms discussed in the previous section would probably be best. (See Section VI for a special case of such motion).

Here, we consider more or less uniform flows where dispersion is secular, and hence, reductions in phase errors should improve the solutions appreciably. Appendix A gives expansions of the relative phase error, $R = (X - V_0\,\delta t/V_0\,\delta t)$, for long wavelength (small β). Here, X is the distance that a phase front of wavenumber $\beta = k\,\delta x$ moves in one timestep. The exact value is $V_0\,\delta t$, of course. Values of ν then can be chosen to reduce the relative phase error from second order to fourth order in β. Since explicit and implicit antidiffusion in the three-point diffusive transport algorithms do not effect the phase properties, μ then can be chosen to minimize the residual long wavelength amplitude errors. In the

phoenical and reversible algorithms, other properties have been built in, removing this freedom to choose μ after the fact.

1. *Shasta Explicit or Implicit (LPE)*

Equation (A.13) gives the long wavelength expansion of the expression for relative phase errors in Shasta. The second-order terms vanish when we choose

$$\nu = \tfrac{1}{6}(1 - \epsilon^2), \tag{29}$$

and the residual fourth-order term is

$$R_{\mathrm{SHA}} \approx \beta^4 \left[-\frac{1}{180} + \frac{\epsilon^2}{36} + \frac{\epsilon^4}{45} \right] + O(\beta^6). \tag{30}$$

This should be compared with the corresponding result given in Eq. (24) for the zero residual damping algorithm.

The choice of antidiffusion coefficient μ must be such as to ensure stability for all harmonics. The most stringent case is for implicit antidiffusion. When the corresponding

$$\mu = \tfrac{1}{6}(1 - \epsilon^2) \tag{31}$$

is chosen, $| A |^2_{\mathrm{SHA}}$ from Eq. (A.11) becomes

$$| A |^2_{\mathrm{SHA}} = 1 - \frac{(\epsilon^2/3)(1 - \epsilon^2)(1 - \cos\beta)^2}{[1 - (1/3)(1 - \epsilon^2)(1 - \cos\beta)]^2}. \tag{32}$$

This is the best one can do, as any larger value of μ would give unstable positive terms of order β^2.

2. *Donor Cell Explicit or Implicit (LPE)*

The same trick can be applied to flux-corrected donor cell. We choose

$$\nu = \frac{1}{6} - \frac{\epsilon}{2} + \frac{\epsilon^2}{3} = \frac{1}{6}(1 - \epsilon)(1 - 2\epsilon) \tag{33}$$

to cancel the β^2 term in Eq. (A.17). The remaining relative phase error is

$$R_{\mathrm{DC}} = \beta^4 \left[-\frac{1}{180} + \frac{\epsilon^2}{36} + \frac{\epsilon^4}{45} \right] + O(\beta^6). \tag{34}$$

This term of order β^4 is identical to the corresponding Shasta error, Eq. (30). Here again, as in the case of zero residual damping, we are applying the same constraints and the two algorithms become linearly identical. Letting

$$\mu = \tfrac{1}{6}(1 - \epsilon^2) \tag{35}$$

as before, gives exactly the same residual amplification factor, Eq. (32), as found for the implicit Shasta algorithm.

Figure 6(a) shows our test calculation performed using these low-phase-error three-point transport algorithms. The resulting profile looks qualitatively better than the ZRD algorithms and the standard algorithms. The value of the average absolute error, A.E. = 0.034, is smaller than the original Shasta algorithm using either explicit, implicit, or phoenical antidiffusion. This clearly demonstrates that most of the remaining error in finite-difference algorithms, over and above the A.E. = 0.022 value from the Gibbs phenomenon, can be tuned out of the method by properly structuring the phase properties. In fact, these LPE algorithms are within 3% or so of the very best finite-difference algorithms that have been discovered so far.

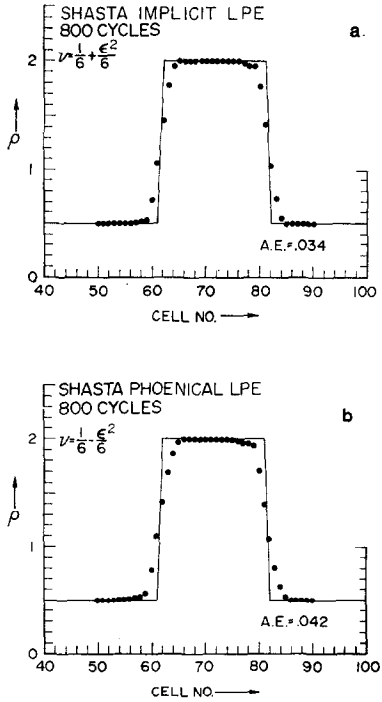


FIG. 6. Comparison of implicit and phoenical low phase error Shasta on the square wave test problem. Using $\nu = \frac{1}{6}(1 - \epsilon^2)$ in each case reduces phase errors from second-order to fourth-order in $k\delta x$, and hence, dispersive ripples are minimized, making the work on the flux corrector much easier. (a) Implicit antidiffusion gives almost as good a result as with the reversible FCT algorithm but (b) phoenical antidiffusion is not quite as accurate. It does have the advantage of being local and not requiring the solution of a tridiagonal system of equations and is more accurate than the simplest FCT algorithms.

3. *Phoenical Shasta (LPE)*

Although phoenical algorithms have only one free parameter left, it can be chosen to eliminate the β^2 term in Eq. (A.21). The value of ν to do this is

$$\nu = \tfrac{1}{6}(1 - \epsilon^2),$$

exactly the same value needed to phase-correct the nonphoenical three-point algorithms. The residual damping for this LPE modification to phoenical Shasta is

$$|A|^2_{\mathrm{PSH}} = 1 - \frac{\epsilon^2}{3}(1 - \epsilon^2)(1 - \cos\beta)^2 - \frac{4\epsilon^2}{9}(1 - \epsilon^2)^2(1 - \cos\beta)^3 - \frac{1}{9}(1 - \epsilon^2)^3(1 - \cos\beta)^4. \tag{37}$$

The remaining fourth-order phase errors become

$$R_{\mathrm{PSH}} = \beta^4\left(-\frac{1}{30} + \frac{\epsilon^2}{12} - \frac{\epsilon^4}{20}\right) + O(\beta^6). \tag{38}$$

Since the phoenical algorithms are relatively easy to use, being local and nonimplicit, the apparent advantages of low-phase-error implicit Shasta may be overshadowed by the expense of an implicit calculation and the nonlinear ramifications of its nonlocal character. Figure 6(b) uses the LPE phoenical algorithm to perform the test calculation. In this case, A.E. = 0.042, the best result obtained using a strictly local algorithm.

4. *Phoenical Donor Cell (LPE)*

The phoenical version of LPE donor cell is achieved by setting

$$\nu = \frac{1}{6} - \frac{\epsilon}{2} + \frac{\epsilon^2}{3}, \tag{39}$$

the same value that worked for the explicit and implicit donor cell versions. The residual fourth-order term is

$$R_{\mathrm{PDC}} = \beta^4\left[-\frac{1}{30} + \frac{\epsilon}{6} - \frac{\epsilon^2}{3} - \frac{\epsilon^3}{3} - \frac{2\epsilon^4}{15}\right] + O(\beta^6). \tag{40}$$

This phoenical version of donor cell is not identical to phoenical Shasta because the velocity dependent and independent parts of the algorithm are now treated differently so adjustment of the single parameter ν no longer can make the two algorithms equivalent.

The residual damping has the form

$$|A|^2_{\text{PDC}} = 1 - 2\epsilon(1-\epsilon)(1-\cos\beta) - \frac{2}{3}\epsilon(1-\epsilon)(1-2\epsilon)^2(1-\cos\beta)^2 + \frac{2\epsilon^2}{9}(1-\epsilon)^2(1-2\epsilon)^2(1-\cos\beta)^3. \tag{41}$$

This damping is quite a bit larger than the phoenical Shasta LPE values given by Eq. (37) because the velocity dependent damping of the basic donor cell algorithm is linear rather than quadratic. Therefore, for $1 - \cos\beta \ll 1$, the linear term dominates. Thus, the phoenical Shasta LPE algorithm appears better than the corresponding donor cell algorithm.

5. *Reversible FCT (LPE + ZRD)*

In the previous section and in Appendix A, a reversible (and hence, ZRD) algorithm was introduced which permitted flux-correction. The algorithm achieves ZRD through its implicit reversible form and thus the value of ν remained free to improve the phase properties. The term of order β^2 in Eq. (A.30) can be eliminated by choosing

$$\nu = \frac{1}{6} + \frac{\epsilon^2}{12}. \tag{42}$$

The remaining fourth-order term is then

$$R_{\text{REV}} = \beta^4\left[-\frac{1}{180} + \frac{\epsilon^2}{144} - \frac{\epsilon^4}{720}\right] + O(\beta^6). \tag{43}$$

Figure 3(b) shows the test problem solved using this algorithm REVFCT. The value of the error, A.E. = 0.033, is the smallest value obtained by any practical finite-difference algorithm. This error is only 50% larger than the optimal value obtained with Fourier FCT, and the algorithm admits trivial extensions to variable velocity and nonuniform spatial grids.

The REVFCT algorithm can be excellent for many pure linear convection problems. Because the transport term as well as the antidiffusion terms is implicit in REVFCT, however, the results on the 1D hydrodynamic shock problem were disappointing. The tridiagonal nonlocal aspect of the algorithm allowed information about the advancing shock front to propagate upstream further than the flux correction procedure could cope with (about a cell or two). Diffusive errors could not be arranged to cancel the residual dispersion near the shock, and hence, the solutions were unacceptable. This objection to implicit treatments of the convective terms would seem to be quite general even though only a specific instance was tested.

In summary, low phase error (LPE) versions of FCT algorithms show much more promise than their ZRD counterparts for improving the already good FCT schemes. The best algorithms tested were implicit Shasta and REVFCT, giving an error only 50% larger than optimal. However, phoenical Shasta LPE is the simplest, all-around algorithm. This local easily generalizable algorithm has performed very well in many of our complex nonlinear plasma codes and has been applied to multidimensional problems and other types of equations.

VI. Nonlinear Stability

In Sections III–V, a number of FCT algorithms were defined and analyzed. Their linear amplification factors $|A|^2$ were determined as functions of time-step and harmonic wave number through the parameters ϵ and β, respectively. For

TABLE I

Ranking of Various Algorithms with Respect to Absolute Error on the Square Wave Test Problem

Algorithms	Absolute error	ν	μ
Optimal Fourier FCT (implicit)	0.022	0.050	0.050
Reversible FCT (LPE)	0.033	$\frac{1}{6}(1+\frac{1}{2}\epsilon^2)$	$\frac{1}{6}(1+\frac{1}{2}\epsilon^2)$
Shasta[a] (implicit LPE)	0.034	$\frac{1}{6}(1-\epsilon^2)$	$\frac{1}{6}(1-\epsilon^2)$
Donor cell (implicit LPE)	0.034	$\frac{1}{6}-(\epsilon/2)+(\epsilon^2/3)$	$\frac{1}{6}(1-\epsilon^2)$
Shasta[a] phoenical (LPE)	0.042	$\frac{1}{6}(1-\epsilon^2)$	$\frac{1}{6}(1-\epsilon^2)$
Shasta[a] (implicit FCT)	0.049	$\frac{1}{8}$	$\frac{1}{8}$
Shasta[a] (phoenical FCT)	0.052	$\frac{1}{8}$	$\frac{1}{8}$
Shasta[a] (explicit FCT)	0.057	$\frac{1}{8}$	$\frac{1}{8}$
Donor cell (explicit FCT)	0.064	0	$(\epsilon/2)(1-\epsilon)$
Shasta[a] (implicit ZRD)	0.066	$\frac{1}{4}(1-\epsilon^2)$	$\frac{1}{4}(1-\epsilon^2)$
Donor cell (implicit ZRD)	0.066	$\frac{1}{4}(1-\epsilon)^2$	$\frac{1}{4}(1-\epsilon^2)$
Lax–Wendroff (diffused)	0.119	0.010	0
Leapfrog (diffused)	0.122	0.010	0
Lax–Wendroff (simple)	0.175	0	0
Leapfrog (simple)	0.245	0	0
Donor cell (simple)	0.260	0	0

[a] Also applies to Lax–Wendroff in constant velocity case, making allowance for intrinsic diffusion of $\frac{1}{8}$ in Shasta, which is included in ν here.

TABLE II

Positivity and Linear Stability of FCT Algorithms

Algorithm	Conditions to ensure[a]	
	Positivity ($\lvert \epsilon \rvert \leqslant$)	Stability ($\lvert \epsilon \rvert \leqslant$)
Shasta (explicit $\nu = \frac{1}{8}$)	$\frac{1}{2}$*	$(\frac{7}{12})^{(1/2)}$
Shasta (implicit $\nu = \frac{1}{8}$)	$\frac{1}{2}$*	$(\frac{1}{2})^{(1/2)}$
Shasta (phoenical $\nu = \frac{1}{8}$)	$\frac{1}{2}$*	$(\frac{1}{2})^{(1/2)}$
Shasta (implicit ZRD)	$\frac{1}{2}$*	—
Shasta (explicit LPE)	$\frac{1}{2}$	1
Shasta (implicit LPE)	$\frac{1}{2}$	1
Shasta (phoenical LPE)	$\frac{1}{2}$	1
Donor cell (explicit $\nu = 0$)	1	1
Donor cell (implicit $\nu = 0$)	1	1
Donor cell (implicit ZRD)	1	—
Donor cell (explicit LPE)	1	1
Donor cell (implicit LPE)	1	1
Donor cell (phoenical LPE)	1	1
Reversible FCT (LPE)	0	—
Fourier FCT	0	—

[a] Bounds on ϵ imposed by (i) positivity of the transport stage, and (ii) linear stability for all harmonics, assuming uniform velocity. Asterisks denote forms of Shasta that are positive for all $\lvert \epsilon \rvert \leqslant 1$ when v is uniform (whereupon the transport stage becomes identical with Lax–Wendroff). Where entries under (ii) are omitted, the algorithms are formally linearly stable for all ϵ (ZRD).

each algorithm, there exists a critical value $\bar{\epsilon}$, such that for all β and all $\epsilon \leqslant \bar{\epsilon}$, $\lvert A \rvert^2 \leqslant 1$. The maximum timestep δt imposed by the requirement of linear stability is determined through $\bar{\epsilon} = V \delta t_{\max}/\delta x$. The values of $\bar{\epsilon}$ for the algorithms considered in this paper are compiled in Table II.

For a nonlinear coupled set of fluid equations, however, a guarantee of linear stability is not enough. In such a system, energy tends to cascade from large scale sizes down to small ones. This effect is normally entirely physical, although exaggerated by numerical truncation errors in finite-difference representations. If sufficient dissipation (physical or numerical) is present, this energy is removed or degraded into heat. But if there is no dissipation, the energy can be aliased into the long-wavelength modes that are usually the ones of greatest interest. This

process, first described by Phillips [19], manifests itself as a grid-separation instability. It is a serious numerical problem in many meterological applications [20] and in other situations where very long running times are employed.

We now show an example of how this instability arises in a problem where unstaggered two-step Lax–Wendroff differencing is used. Letting ρ_j^0, $\rho_j^{1/2}$ and ρ_j^1 represent the density on the jth mesh point before transport, after the first step, and after the second step, respectively, we have

$$\rho_j^{1/2} = \frac{1}{2}(\rho_{j+1}^0 + \rho_{j-1}^0) - \frac{\delta t}{4\delta x}(\rho_{j+1}^0 V_{j+1} - \rho_j^0 V_j), \tag{44}$$

$$\rho_j^1 = \rho_j^0 - \frac{\delta t}{2\delta x}(\rho_{j+1}^{1/2} V_{j+1} - \rho_{j-1}^{1/2} V_{j-1}). \tag{45}$$

Here, following Gerrity [21], we take the velocity in the form

$$V_j = V + \tilde{V} e^{i\pi j} = V[1 - (-1)^j \gamma], \tag{46}$$

where $\gamma = \tilde{V}/V$ is the relative amplitude of a two-cell wiggle imposed on a uniform flow field. V and $\tilde{V}$ are both independent of position and time. This oscillation in V may be thought of as being induced by a corresponding oscillation in ρ through the nonlinear coupling between the equation propagating velocity and the continuity equation.

Combining Eqs. (44) and (45) and using (46) for V_j, we obtain the following result for the case of j even:

$$\rho_j^1 = \rho_j^0[1 - \epsilon^2(1 - \gamma^2)] - (\epsilon(1 - \gamma)/2)\{\rho_{j+2}^0[1 - \epsilon(1 + \gamma)] - \rho_{j-2}^0[1 + \epsilon(1 + \gamma)]\},$$

where $\epsilon = (v\,\delta t/2\,\delta x)$. (For j odd, the sign of γ is reversed.) Writing

$$\rho_j^n = \zeta^n \exp(ikj\,\delta x),$$

where ζ is the amplification factor, we find

$$|\zeta|^2 = [1 + \epsilon^2(1 - \gamma^2)(\cos\alpha - 1)]^2 + \epsilon^2(1 - \gamma)^2 \sin^2\alpha, \tag{47}$$

with $\alpha = 2k\,\delta x$. It is noteworthy that for this differencing scheme, even mesh points are connected only to even, and odd points to odd.

For $\gamma = 0$, Eq. (47) reduces to the usual linear amplification for Lax–Wendroff differencing and $|\zeta| \leqslant 1$ provided $\epsilon \leqslant 1$ (the CFL limit). But for arbitrary nonvanishing γ, Eq. (47) implies instability ($|\zeta| > 1$) for all modes satisfying

$$1 > \cos\alpha > 1 - \frac{4|\gamma|}{1 + |\gamma| - \epsilon^2(1 - \gamma^2)(1 - |\gamma|)}. \tag{48}$$

For $\gamma > 0$ ($\gamma < 0$), values of ρ_j are amplified only on the odd (even) points. The maximum amplification, which occurs at

$$\cos \hat{\alpha} \equiv \frac{1 - |\gamma|}{1 + |\gamma|} \frac{1 - \epsilon^2(1 - \gamma^2)}{1 - \epsilon^2(1 + |\gamma|)^2},$$

is

$$|\zeta|^2 = \frac{1 - \epsilon^2(1 + |\gamma|)(1 - 3|\gamma|)}{1 - \epsilon^2(1 - |\gamma|)^2} \approx 1 + \frac{4\epsilon^2\gamma^2}{(1 - \epsilon^2)^2}, \qquad |\gamma| \ll 1. \quad (49)$$

For a simulation to exhibit this instability, a relatively quiescent state must be followed for a long time. Growth is initially slow, but the grid separation in ρ feeds back into V, so that γ actually increases in time. The amplification rate ζ increases without bound, implying the solutions become singular at some finite time (such instabilities are sometimes termed *explosive*).

It is a simple matter to flux-correct the scheme of Eqs. (44) and (45), as described in FCT/II. The result is to couple the odd and even points both linearly (through the diffusion/antidiffusion operation) and nonlinearly (through the flux limiter). The former mechanism has a decided inhibiting effect on the instability, while the latter completely stabilizes it.

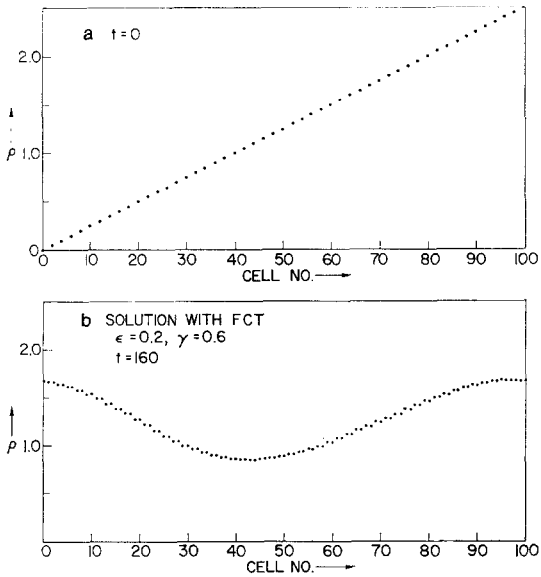


FIG. 7. Solution of a nonlinear instability test problem using flux-corrected transport. (a) Initial conditions are a sawtooth to excite all harmonics. The velocity field has a large oscillatory component. (b) After 800 cycles a nonlinearly stable FCT solution is shown.

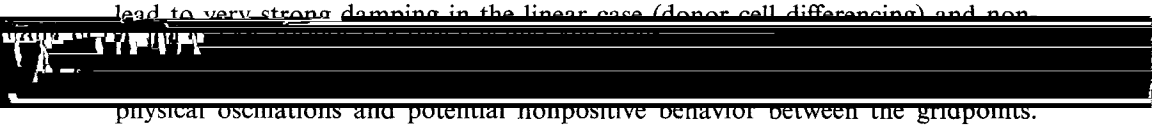
To illustrate this discussion, we provide a numerical example. Fig. 7 shows a ramp profile, chosen for the initial state because all the Fourier harmonics enter into the superposition with nonzero amplitudes. This was solved on the same mesh employed in the square wave tests using the differencing scheme just discussed. The solution was propagated for 800 cycles or until values exceeding 10^{10} in magnitude were obtained. For example, at $t = 800\ \delta t$, amplitudes of both signs with magnitudes $\sim 8 \times 10^8$ were attained for $\epsilon = 0.2$ and $\gamma = 0.6$. (The maximum amplification factor given by Eq. (49) for these parameters is $|\tilde{\zeta}|^2 = 1.058$, hence $|\tilde{\zeta}|^{800} \approx 6 \times 10^9$. This multiplies 0.19, the initial amplitude of the fastest growing mode, yielding a result in agreement with the amplitude observed.) *In contrast, the FCT solution satisfies* $0.86 \lesssim \rho \lesssim 1.67$.

It should be emphasized that the nonlinear instability is extremely prevalent. It does not depend on complete decoupling of the grid into odd and even sets of points, as occurs here; this particular differencing scheme was chosen because it makes the algebra simpler. Even those schemes like staggered Lax–Wendroff etc., which mix odd and even points, do not prevent aliasing of energy from short to long scale sizes [20]. The nonlinear properties of the flux-limiting stage of FCT, however, do provide a kind of nonlinear numerical dissipation restricted to the shortest scale sizes. This appears to eliminate the nonlinear grid-separation phenomena.

VII. Conclusions

In this paper, we have considered the problem of reducing the errors in numerical solutions of the continuity equation (1). Specifically, the properties of several flux-corrected transport algorithms were studied and the algorithms modified to reduce undesirable amplitude and phase errors arising from the use of local finite-difference approximations to derivatives. Three types of error were isolated: amplitude errors (damping or instability), phase errors (dispersion), and intrinsic grid representation errors (the Gibbs phenomenon).

The Gibbs errors arise because only a finite number of values, the density function values at the numerical gridpoints, are being used to represent a physically continuous profile. The behavior of the profile between the gridpoints is unknown and various interpolation forms or splines could be assumed. The simplest of these lead to very strong damping in the linear case (donor cell differencing) and non-



physical oscillations and potential nonpositive behavior between the gridpoints. Smoothing off the Gibbs oscillations using FCT (Section III) gave us a measure of this intrinsic irreducible error.

The phase and amplitude errors were adjusted in each of the several finite-

difference algorithms considered by changing the strengths and forms of the diffusion and antidiffusion coefficients. We looked at the extremes of the problem by minimizing phase errors and amplitude errors independently. A truly complete optimization would involve considering a whole spectrum of ν and μ values for each algorithm but the original unmodified forms of the FCT algorithms used values of μ and ν roughly midway between the ZRD and LPE limits. Thus, the range of possible values is well covered by test cases as well as analysis. Furthermore, the results suggest strongly that low-phase-error (LPE) algorithms are significantly superior to zero residual damping (ZRD) algorithms. Therefore, it is unlikely that better versions of the given algorithms exist than the LPE versions suggested and tested. Of course, much work remains to generalize this analysis rigorously to nonconstant velocity and grid spacing.

Throughout the paper, we restricted consideration to single step (in time) algorithms satisfying the conditions laid out in Section II. The algorithms considered were all linearly stable and conservative before, during, and after the application of flux correction. The diffusion and antidiffusion coefficients were chosen so the resultant algorithms would be linearly second-order, while FCT guaranteed positivity. All except the optimal Fourier FCT are generalizable to variable grid/variable velocity flow fields. We do not claim to have exhausted the list of possible or interesting algorithms, but have surveyed the field rather thoroughly. There do exist many double step and nonlocal algorithms that have been used and tested thoroughly in the past, but by and large these do not lend themselves particularly well to flux correction (cf. the discussion of leapfrog and two-step Lax–Wendroff in FCT/II) and hence, do not guarantee positivity.

We found that three algorithms looked particularly good. All are low-phase-error (LPE) algorithms. For purely convective problems reversible FCT LPE and implicit Shasta LPE were by far the best generalizable finite-difference algorithms. Their errors, given in Table I, are within 50% of optimal on our test problem. Due to its nonlocal implicit nature, however, the REVFCT algorithm was found to do poorly on nonlinear shock problems and to cost a little more to run.

The third algorithm, phoenical Shasta LPE, overcomes both these objections admirably, at the price of a little accuracy. We can see from Table I, however, that phoenical Shasta LPE is still within a factor of two of optimal and five or six times more accurate than standard flux-uncorrected algorithms. Our results enable us to draw the following conclusions:

1. One of the three types of error, the Gibbs phenomenon, is intrinsic to the finite discrete spatial resolution. This error can give negative densities so some damping is required, at least initially, of all algorithms to avoid this (Section II).

2. The effective size of this minimal error was measured using a Fourier FCT algorithm with perfect phase and amplitude properties (Section III). Several

useable low phase error (LPE) algorithms were devised using FCT, which approach this minimal error within a factor of two (Section IV).

3. Modest numerical dispersion is generally a more serious source of error than modest harmonic damping errors (Section IV and V). The reason is that phase differences between harmonics increase secularly in time, while amplitude errors are self-eliminating.

4. The minimal error due to the Gibbs phenomenon can be approached quite closely by LPE FCT algorithms such as phoenical Shasta (Section V). Reducing the dispersion to fourth order appears to be quite adequate even with some residual amplitude damping.

5. The best of the FCT algorithms are about on order of magnitude better than some of the ordinary continuity-equation algorithms (Table I). Even the original Shasta algorithm and some of the variations are five or six times better than the old methods (Sections IV and V). Zero residual damping (ZRD) algorithms were actually somewhat worse in the test problem than the original Shasta explicit FCT (Section IV). Of course, in oscillatory flows, the phase errors are not generally secular so ZRD algorithms may be most valuable in such cases.

6. FCT seems to strongly repress or eliminate the nonlinear grid-separation instabilities that can arise when the flow field can have an oscillatory component (Section VI). The strongly nonlinear effect of the flux correction stage of the algorithms shuts off the nonlinear growth when it reaches a large enough amplitude to just form new nonphysical maxima and minima in the solution.

APPENDIX A: NUMERICAL ANALYSIS

Several different algorithms are considered and compared in the paper. This Appendix lists the various numerical formulae used for this purpose. We assume a constant velocity in the x direction and consider the one-dimensional continuity equation

$$\frac{\partial \rho}{\partial t} = -V_0 \frac{\partial \rho}{\partial x}, \tag{A.1}$$

on a fixed uniform grid with spacing δx. The timestep is δt and we define

$$\epsilon \equiv V_0 \, \delta t / \delta x, \tag{A.2}$$

the fraction of a cell crossed by the flow in one timestep.

Equation (A.1) is linear, so we can Fourier analyze and examine the evolution of a single harmonic. The initial condition is

$$\rho_j{}^0 = \rho^0 e^{ikj\delta x}, \tag{A.3}$$

where k is the wavenumber of the particular harmonic being considered. On a finite discrete mesh (periodic boundary conditions)

$$k = (2\pi m/L) \quad (m = 0, 1, \ldots, N), \tag{A.4}$$

where $L = N\,\delta x$. Thus, the quantity

$$\beta = k\,\delta x, \tag{A.5}$$

which will appear throughout our analysis, ranges from 0 to 2π in value.

We are concerned with the following algorithms: (1) Shasta with explicit (implicit) antidiffusion; (2) donor cell with explicit (implicit) antidiffusion; (3) phoenical Shasta; (4) phoenical donor cell, and (5) a new algorithm REVFCT, a linearly reversible continuity equation algorithm. The three types of antidiffusion (explicit, implicit, and phoenical) are described in the proceeding paper of this series, FCT/II.

The present paper ignores leapfrog algorithms and does not explicitly consider Lax-Wendroff algorithms. Previously published tests of flux-corrected leapfrog show no particular advantages over Shasta or Lax–Wendroff, and there are minor difficulties in applying FCT to leapfrog. In the linear constant-velocity test problem used as the focus for most of this paper, the results for the Shasta transport algorithm apply equally well for Lax–Wendroff. (For constant velocity, Shasta is just Lax–Wendroff with an added three-point diffusion with nondimensional coefficient $\frac{1}{8}$).

The transfer function A relates $\{\rho_j{}^0\}$ to the density function $\{\rho_j{}^1\}$ one timestep later, where for brevity the dependence on wavenumber k is suppressed. Thus

$$\rho_j{}^1 = A_{\mathrm{ALGO}}\rho_j{}^0; \tag{A.6}$$

here, the subscript on A will indicate the algorithm. The amplification factor for each algorithm is just $|A|_{\mathrm{ALGO}}$, the magnitude of the transfer function.

$$|A|_{\mathrm{ALGO}} = [\mathrm{Re}^2(\rho^1/\rho^0) + \mathrm{Im}^2(\rho^1/\rho^0)]^{1/2}. \tag{A.7}$$

The angle of A in the complex plane can be related to the phase properties of the algorithm for the harmonic being considered. In a time δt, the phase front for any harmonic should propagate a distance $x = V_0\,\delta t$. In actuality, each harmonic propagates a different distance $x_{\mathrm{ALGO}}(k)$. The different phase properties

of each harmonic constitute the dispersion of the algorithm. The phase propagation distance is calculated from the inverse trigonometric relation of

$$\tan kx_{\mathrm{ALGO}}(k) \equiv -\mathrm{Im}(A_{\mathrm{ALGO}})/\mathrm{Re}(A_{\mathrm{ALGO}}). \tag{A.8}$$

The explicit dependence on k of x_{ALGO} in the following will also be dropped.

Finally, for each algorithm and harmonic we can define a relative phase error R_{ALGO}, defined to be

$$R_{\mathrm{ALGO}} = (x_{\mathrm{ALGO}} - V_0\,\delta t)/V_0\,\delta t. \tag{A.9}$$

This will usually be expanded as a power series in β.

1. *Shasta Explicit or Implicit Algorithm (SHA)*

$$A_{\mathrm{SHA}} = [1 - (2\nu + \epsilon^2)(1 - \cos\beta) - i\epsilon \sin\beta][1 \pm 2\mu(1 - \cos\beta)]^{\pm 1}. \tag{A.10}$$

The upper (lower) sign gives the explicit (implicit) form of A_{SHA}. Here, ν and μ are, respectively, the nondimensional diffusion, and the antidiffusion coefficients. The μ and ν do not need to be equal in general, but μ should always be small enough so that $|A|_{\mathrm{ALGO}} \leqslant 1$ for all k.

For Shasta (as for most other algorithms) the phase properties are unaffected by the type of antidiffusion, even though the amplification factors do depend on whether explicit or implicit forms are used.

$$|A|^2_{\mathrm{SHA}} = \{1 - 4\nu(1 - \cos\beta) + [2\nu + \epsilon^2)^2 - \epsilon^2](1 - \cos\beta)^2\}$$
$$\times \{1 \pm 2\mu(1 - \cos\beta)\}^{\pm 2}, \tag{A.11}$$

Equation (20) of FCT/I is just a special case of (A.11) with $\nu = \mu = \frac{1}{8}$ and using the upper (plus) sign.

The phase properties are derived from

$$\tan kx_{\mathrm{SHA}} = \frac{\epsilon \sin\beta}{[1 - (2\nu + \epsilon^2)(1 - \cos\beta)]}, \tag{A.12}$$

which has the following relative phase error for small β

$$R_{\mathrm{SHA}} \approx \beta^2\left[\nu - \frac{1}{6} + \frac{\epsilon^2}{6}\right] + \beta^4\left[\frac{1}{120} - \frac{\nu}{4} + \frac{\epsilon^2}{24} + \nu^2 - \frac{\epsilon^4}{20}\right] + O(\beta^6). \tag{A.13}$$

2. *Donor Cell Explicit or Implicit (DC)*

$$A_{\mathrm{DC}} = [1 - (2\nu + |\epsilon|)(1 - \cos\beta) - i\epsilon \sin\beta][1 \pm 2\mu(1 - \cos\beta)]^{\pm 1}, \quad \text{(A.14)}$$

where the upper signs again give explicitly antidiffused donor cell. Then, taking $\epsilon > 0$ (for the general case, ϵ is replaced by $|\epsilon|$), we find

$$|A|^2_{\mathrm{DC}} = \{1 - (4\nu + 2\epsilon - 2\epsilon^2)(1 - \cos\beta) + [(2\nu + \epsilon)^2 - \epsilon](1 - \cos\beta)^2\} \times \{1 \pm 2\mu(1 - \cos\beta)\}^{\pm 2}. \quad \text{(A.15)}$$

$$\tan kx_{\mathrm{DC}} = \frac{\epsilon \sin\beta}{[1 - (2\nu + \epsilon)(1 - \cos\beta)]} \quad \text{(A.16)}$$

$$R_{\mathrm{DC}} \approx \beta^2\left[\nu - \frac{1}{6} + \frac{\epsilon}{2} - \frac{\epsilon^2}{3}\right] + \beta^4\left[\frac{1}{120} - \frac{\nu}{4} - \frac{\epsilon}{8} + \nu^2 + \frac{5\epsilon^2}{12} - \frac{\epsilon^3}{2} + \frac{\epsilon^4}{5} + \nu\epsilon - \nu\epsilon^2\right] + O(\beta^6), \quad \text{(A.17)}$$

for long wavelengths (small β).

3. *Phoenical Shasta (PSH)*

$$A_{\mathrm{PSH}} = [1 - \epsilon^2(1 \cos\beta) - 2\nu\epsilon^2(1 - \cos\beta)^2] - i\epsilon \sin\beta[1 + 2\nu(1 - \cos\beta)]. \quad \text{(A.18)}$$

In the phoenical algorithms, we are adding the physical constraint that $|A_{\mathrm{PSH}}| \equiv 1$ for all harmonics β (i.e., $k\delta x$) when $\epsilon = 0$. This constraint removes one degree of freedom and we must set $\mu = \nu$ to ensure this useful property.

$$|A|^2_{\mathrm{PSH}} = \{1 - \epsilon^2(1 - \cos\beta) - 2\nu\epsilon^2(1 - \cos\beta)^2\}^2 - \epsilon^2(1 - \cos\beta)^2 \times [1 + 2\nu(1 - \cos\beta)]^2 + 2\epsilon^2(1 - \cos\beta)[1 + 2\nu(1 - \cos\beta)]^2. \quad \text{(A.19)}$$

$$\tan kx_{\mathrm{PSH}} = \frac{\epsilon \sin\beta[1 + 2\nu(1 - \cos\beta)]}{[1 - \epsilon^2(1 - \cos\beta) - 2\nu\epsilon^2(1 - \cos\beta)^2]}. \quad \text{(A.20)}$$

$$R_{\mathrm{PSH}} = \beta^2\left[\nu - \frac{1}{6} + \frac{\epsilon^2}{6}\right] + \beta^4\left[\frac{1}{120} - \frac{\nu}{4} + \frac{\epsilon^2}{24} - \frac{\epsilon^4}{20}\right] + O(\beta)^6. \quad \text{(A.21)}$$

To this order, Eq. (A.21) differs from the corresponding Eq. (A.13) only in the absence of a term ν^2 multiplying β^4; a similar relationship holds for donor cell (Eqs. (A.17) and (A.25) below).

4. *Phoenical Donor Cell (PDC)*

$$A_{\mathrm{PDC}} = [1 - \epsilon(1 - \cos\beta) - 2\nu\epsilon(1 - \cos\beta)^2] - i\epsilon \sin\beta[1 + 2\nu(1 - \cos\beta)]. \tag{A.22}$$

$$|A|^2_{\mathrm{PDC}} = \{1 - \epsilon(1 - \cos\beta) - 2\nu\epsilon(1 - \cos\beta)^2\}^2 + [2\epsilon^2(1 - \cos\beta) - \epsilon^2(1 - \cos\beta)^2][1 + 2\nu(1 - \cos\beta)]^2. \tag{A.23}$$

$$\tan kx_{\mathrm{PDC}} = \frac{\epsilon \sin\beta[1 + 2\nu(1 - \cos\beta)]}{[1 - \epsilon(1 - \cos\beta) - 2\nu\epsilon(1 - \cos\beta)]}. \tag{A.24}$$

$$R_{\mathrm{PDC}} = \beta^2\left[\nu - \frac{1}{6} + \frac{\epsilon}{2} - \frac{\epsilon^2}{3}\right] + \beta^4\left[\frac{1}{120} - \frac{\nu}{4} - \frac{\epsilon}{8} + \nu\epsilon - \nu\epsilon^2 + \frac{5\epsilon^2}{12} - \frac{\epsilon^3}{2} + \frac{\epsilon^4}{5}\right] + O(\beta^6). \tag{A.25}$$

5. *Reversible Flux-Corrected Transport (REV)*

An implicit algorithm has been introduced in this paper with both the zero residual damping (ZRD) and low phase errors (LPE) properties.

Linear reversibility again requires $\mu = \nu$. The finite-difference form relating ρ_j^1 to ρ_j^0 is

$$\rho_j^1 + \frac{\epsilon}{4}(\rho_{j+1}^1 - \rho_{j-1}^1) + \nu(\rho_{j+1}^1 - 2\rho_j^1 + \rho_{j-1}^1) = \rho_j^0 - \frac{\epsilon}{4}(\rho_{j+1}^0 - \rho_{j-1}^0) + \nu(\rho_{j+1}^0 - 2\rho_j^0 + \rho_{j-1}^0). \tag{A.26}$$

For $\nu = 0$, (A.26) reduces to the ordinary Euler-modified scheme. Clearly, this is implicit for $\{\rho_j^1\}$, requiring a full tridiagonal solution. Generalizations to variable grid and velocity are obvious but the implicit nature of the algorithm is a drawback.

$$A_{\mathrm{REV}} = \frac{[1 - i(\epsilon/2)\sin\beta - 2\nu(1 - \cos\beta)]}{[1 + i(\epsilon/2)\sin\beta - 2\nu(1 - \cos\beta)]}. \tag{A.27}$$

It follows immediately from (A.27) that

$$|A|^2_{\mathrm{REV}} = 1, \tag{A.28}$$

for all β, ϵ, and ν. The condition satisfied by the phoenical algorithm, $A = 1$ when $\epsilon = 0$, is automatically satisfied; hence, the form of the algorithm itself assures the zero residual damping property.

The phase properties are derived from

$$\tan(\tfrac{1}{2}kx_{\mathrm{REV}}) = \frac{(\epsilon/2)\sin\beta}{[1 - 2\nu(1 - \cos\beta)]}, \tag{A.29}$$

which gives

$$R_{\text{REV}} = \beta^2 \left[\nu - \frac{1}{6} - \frac{\epsilon^2}{12}\right] + \beta^4 \left[\frac{1}{120} - \frac{\nu}{4} + \nu^2 + \frac{\epsilon^2}{24} - \frac{\epsilon^2\nu}{4} + \frac{\epsilon^4}{80}\right] + O(\beta^6). \quad \text{(A.30)}$$

REFERENCES

1. J. P. BORIS AND D. L. BOOK, Flux-corrected transport. I. SHASTA, a transport algorithm that works, *J. Comp. Phys.* **11** (1973), 38 (referenced as FCT/I in text).
2. D. L. BOOK, J. P. BORIS, B. E. MCDONALD, AND C. E. WAGNER, SHASTA, a transport algorithm that works, DNA Sympos. Menlo Park, California, August 10–12, 1971; J. P. BORIS, A fluid transport algorithm that works, *in* "Proc. Seminar Course in Comput. as Lang. of Physics," International Center for Theoretical Physics, Trieste, Italy, August 2–20, 1971.
3. J. P. BORIS, SHAS2D, a fully compressible hydrodynamics code in two dimensions, U.S. Naval Research Laboratory, Memorandum Report No. 2542. December 1972.
4. D. L. BOOK, J. P. BORIS, AND K. HAIN, Flux-corrected transport. II. Generalizations of the method, *J. Comp. Phys.* **18** (1975), 248 (referenced as FCT/II in text).
5. D. V. ANDERSON, Axisymmetric multifluid simulation of high beta plasmas with anisotropic transport using a moving flux coordinate grid, *J. Comp. Phys.* **17** (1975), 246.
6. D. L. BOOK, EDWARD OTT, AND A. L. SULTON, The Rayleigh–Taylor instability in the shallow-water approximation, *Phys. Fluids* **17** (1974), 676.
7. D. L. BOOK, S. R. GOLDMAN, AND S. L. OSSAKOW, Interaction of a small *F*-region barium cloud with the neutral wind, *EOS* **56** (1974), 1154.
8. J. P. BORIS, J. H. GARDNER, AND S. ZALESAK, Atmosphere hydrocodes using FCT algorithms, NRL Memorandum Report No. 3081, 1975.
9. D. BOOK, J. BORIS, R. CLARK, D. COLUMBANT, J. GARDNER, P. LIEWER, W. MANHEIMER, AND S. ZALESAK, Flux-corrected transport strikes again, *in* "Proc. 7th Conf. on Numer. Simulation of Plasma," New York, June 2–4, 1975.
10. J. GARDNER AND R. A. CHEVALIER, The evolution of supernova remenants. II. Models of an Explosion in a plane, stratified medium, *Astrophys. J.* **192** (1974), 457.
11. P. C. LIEWER AND N. A. KRALL, Self-consistent approach to anomalous resistivity applied to theta pinch experiments, *Phys. Fluids* **16** (1973), 1953.
12. P. C. LIEWER, N. A. KRALL, AND C. E. WAGNER, Numerical studies of theta-pinch implosions in turbulent plasmas, *Bull. Amer. Phys. Soc.* **18** (1975), 1366.
13. M. WIDNER, T. WRIGHT, AND J. FREEMAN, *Bull. Amer. Phys. Soc.* **17** (1972), 1047; J. FREEMAN AND S. THOMPSON, *Proc. 7th Conf. on Numer. Simulation of Plasmas*, New York, June 2–4, 1975.
14. GINO MORETTI, Thoughts and afterthoughts about shock computations, Polytechnic Institute of Brooklyn, Report No. PIBAL 72-37, 1972.
15. M. E. EVANS AND F. H. HARLOW, The particle-in-cell method for hydrodynamic calculations, Los Alamos Sci. Lab. Report No. LA-2139, 1957.
16. S. A. ORSZAG, Numerical simulation of incompressible flows within simple boundaries. I. Galerkin (spectral) representation, *Stud. in Appl. Math.* **50** (1971), 293; S. A. ORSZAG AND M. ISRAEL, Numerical flow simulation by spectral methods, *in* "Proc. of Sympos. on Numer. Models of Ocean Circulations," Nat. Acad. Sci., Washington, D.C., 1972.

17. J. H. AHLBERG, E. N. NILSON, AND J. L. WALSH, "The Theory of Splines and Their Applications," Academic Press, New York and London, 1967.
18. R. D. RICHTMYER AND K. W. MORTON, "Difference Methods for Initial Value Problems," 2nd Edition, Sections 12.9, and 13.9. Interscience, New York, 1967.
19. N. A. PHILLIPS, An example of non-linear computational instability, *in* "Atmosphere and Sea in Motion," Rossby Memorial Volume (B. Bolin, Ed.), p. 501. Rockefeller Institute Press, New York, 1959.
20. A. GRAMMELTVEDT, A survey of finite-difference schemes for the primitive equation for a barotropic fluid, *Mon. Weather Rev.* **97** (1969), 384.
21. J. P. GERRITY, A note on the convectional stability of the two-step Lax–Wendroff form of the advection equation, *Mon. Weather Rev.* **100** (1972), 72.