The Design of Flux-Corrected Transport (FCT) Algorithms for Structured Grids

Steven T. Zalesak

Abstract A given flux-corrected transport (FCT) algorithm consists of three components: (1) a high order algorithm to which it reduces in smooth parts of the flow; (2) a low order algorithm to which it reduces in parts of the flow devoid of smoothness; and (3) a flux limiter which calculates the weights assigned to the high and low order fluxes in various regions of the flow field. One way of optimizing an FCT algorithm is to optimize each of these three components individually. We present some of the ideas that have been developed over the past 30 years toward this end. These include the use of very high order spatial operators in the design of the high order fluxes, non-clipping flux limiters, the appropriate choice of constraint variables in the critical flux-limiting step, and the implementation of a "failsafe" flux-limiting strategy. This chapter confines itself to the design of FCT algorithms for structured grids, using a finite volume formalism, for this is the area with which the present author is most familiar. The reader will find excellent material on the design of FCT algorithms for unstructured grids, using both finite volume and finite element formalisms, in the chapters by Professors Löhner, Baum, Kuzmin, Turek, and Möller in the present volume.

1 Introduction: Modern Front-Capturing Methods

We are interested in systems of conservation laws of the form

$$\frac{\partial \mathbf{q}(\mathbf{x}, t)}{\partial t} + \nabla \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}, t) = 0 \tag{1}$$

where $\mathbf{q}(\mathbf{x}, t)$ and $\mathbf{f}(\mathbf{q}, \mathbf{x}, t)$ are vector functions of the independent variables $\mathbf{x}$ and t, which we henceforth refer to as space and time respectively. Examples of such equations include the Navier-Stokes equations, the equations of magnetohydrodynamics (MHD), the Vlasov equation, and passively-driven convection.

This work was supported by the U.S. Department of Energy.

S.T. Zalesak (✉)
Plasma Physics Division, Naval Research Laboratory, Washington, DC 20375, USA
e-mail: steven.zalesak.ctr@this.nrl.navy.mil

D. Kuzmin et al. (eds.), *Flux-Corrected Transport*, Scientific Computation,
DOI 10.1007/978-94-007-4038-9_2,

It is well known that differentiable solutions to (1) may cease to exist after a finite time t_s, even if the initial conditions $\mathbf{q}(\mathbf{x}, 0)$ are smooth. After t_s, only the integral or "weak" form of (1) will have solutions, and these will contain discontinuities in $\mathbf{q}$ and/or one or more of its derivatives. We will term such discontinuities "fronts" for the purpose of this chapter. This situation is addressed by the Lax-Wendroff Theorem, which states that if one's numerical approximation to Eq. (1) is in "flux" or "conservation" form, and the numerical solution converges everywhere but on a set of measure zero to some solution, then that solution is a weak solution to Eq. (1). Thus the great majority of methods designed to treat fronts in the context of Eq. (1) are in conservation form, i.e., a form consisting of numerical fluxes connecting adjacent grid points, these fluxes being used to advance the numerical solution in time.

Using conservation form does not by itself give the desired result however, since one still needs to compute a convergent solution. In general, numerical methods not designed to deal with fronts will not produce the desired convergence in their presence, often producing numerical oscillations that degrade the solution severely. It is precisely this situation that prompted Von Neumann and Richtmyer to add an explicit artificial dissipation term to Eq. (1), the idea being to smooth the fronts to the point where they are resolved on the grid as narrow but smooth features, thereby producing the desired convergence. This is the fundamental idea underlying nearly all conservative numerical methods designed to handle fronts, including the FCT algorithms we address here. (We are excluding, of course, methods that treat fronts as moving internal boundaries, the class of methods known as "front-tracking methods." We are also excluding random choice methods [3, 4, 8].) For the purposes of this chapter, we shall refer to methods which attempt to smooth a front into a narrow but smooth transition as "front-capturing methods."

Over the past 30 years a host of algorithms, known variously as "modern" front-capturing methods or "high resolution methods," have been developed in an attempt to perform calculations more accurately and more efficiently than with the more traditional explicit artificial dissipation approach. The first of these methods was flux-corrected transport (FCT) [1, 2], but there are now a large number of others. What distinguishes the "modern" front-capturing methods from their predecessors is their attempt to constrain the numerical fluxes, grid point by grid point and timestep by timestep, in such a way as to avoid the production of unphysical values in the solution vector $\mathbf{q}$ in and near the fronts, and at the same time treat the regions in space and time in which $\mathbf{q}$ is smooth as accurately as possible. Clearly the success of these methods depends critically on an accurate criterion for determining what constitutes an unphysical value for $\mathbf{q}$, one of the primary topics of this chapter.

One way of stating the design philosophy of these methods, and the one we shall embrace in this chapter, is as follows:

When the numerics fails, substitute the physics.

Clearly the designers of such algorithms must possess a knowledge of the physics being addressed if they are to be successful.

These modern front-capturing methods may be thought of as consisting of three parts:

1. an algorithm to which they reduce in regions of time and space where $\mathbf{q}$ is smooth;
2. an algorithm to which they reduce at fronts; and
3. a mechanism for weighting each of the above algorithms at each grid point and timestep.

Obviously the accuracy of a given modern front-capturing method may depend strongly on the choices made in each of its three parts. In the FCT algorithms we shall consider here, these three parts correspond to the high-order fluxes, the low-order fluxes, and the flux limiter respectively, terms we shall define shortly. Our experience is that FCT algorithms are capable of solving most problems involving fronts with both robustness and accuracy, as long as certain design principles are adhered to. Toward that end, this chapter shall present to the reader a collection of design principles that we have found to be of value in the creation of an FCT algorithm for a given situation. In general, they involve optimizing one's choice of each of the above three components of the algorithm. The reader will not be surprised to learn that a knowledge of the physics problem being addressed is an essential part of the design criteria.

In Sect. 2, we give a formal definition of FCT, first for the special case of one spatial dimension, and then for multidimensions. In Sect. 3 we give six design criteria that collectively define what we mean by a "properly designed" FCT algorithm. In Sect. 4 we give examples of the kind of performance one can expect from a properly designed FCT algorithm, using the scalar linear advection problem. For this problem, the "physics" that must be incorporated into the algorithm is simple and intuitive, and accurate and robust algorithms are easy to construct. In Sect. 5 we move on to nonlinear systems of equations, using the Euler equations as an example. Here the physics is not trivial as it was in the case of linear advection, and we find that blindly applying the methods that worked well for advection can be disappointing. However, when we transform the problem into a set of variables for which we have a legitimate set of physical constraints that can be imposed, we recover the kind of performance that we saw in the linear advection case. In Sect. 6 we treat both passively-driven convection and compressible gas dynamics in two space dimensions, and again have to face and solve the question of physically appropriate constraints. Finally in Sect. 7 we give our conclusions.

2 Flux-Corrected Transport (FCT) Defined

As we mentioned in the previous section, the great majority of methods designed to treat fronts in the context of Eq. (1), are in conservation form, i.e., a form consisting of numerical fluxes connecting adjacent grid points, these fluxes being used to advance the numerical solution in time. In FCT, at every timestep and at every flux point, these fluxes are computed twice, once using an algorithm guaranteed not to generate unphysical values (the "low order fluxes"), and once using an algorithm that is formally of high accuracy in the smooth portions of the solution (the "high

order fluxes"). FCT then constructs the net fluxes for the timestep as weighted averages of these two candidate fluxes. The weighting is performed in a manner which ensures that the high order fluxes are used to the greatest extent possible without introducing unphysical values into the solution. The procedure is referred to as "flux-correction" or "flux limiting" for reasons which will become clear shortly.

From the above description, it should be clear that one may easily define an FCT algorithm on any structured or unstructured grid in any number of spatial dimensions, as long as one can define a numerical technique for which the difference between a low order time advancement operator and its higher order counterpart can be written as an array of fluxes between adjacent grid points. Rather than attempt to give a definition at that level of generality, we will give formal definitions for the cases of one spatial dimension, and for two spatial dimensions on a structured grid. From these two examples it should be clear how to construct an FCT algorithm in any number of dimensions, and on any grid, structured or unstructured.

2.1 FCT in One Spatial Dimension

In one spatial dimension, Eq. (1) takes the simpler form

$$\frac{\partial q(x,t)}{\partial t} + \frac{\partial f(q,x,t)}{\partial x} = 0. \tag{2}$$

A simple example of such a system of equations is the system describing one-dimensional ideal inviscid fluid flow, also known as the Euler equations:

$$q = \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix}; \quad f = \begin{pmatrix} \rho u \\ \rho u u + P \\ \rho u E + P u \end{pmatrix} \tag{3}$$

where ρ, u, P, and E are the fluid density, velocity, pressure, and specific total energy respectively.

We say that a discrete approximation to Eq. (2) is in conservation or "flux" form when it can be written in the form

$$q_i^{n+1} = q_i^n - \Delta x_i^{-1}[F_{i+(1/2)} - F_{i-(1/2)}]. \tag{4}$$

Here q and f are defined on the spatial grid points x_i and temporal grid points t^n, and Δx_i is the cell width associated with cell i. The $F_{i+(1/2)}$ are called numerical fluxes, and are functions of f and q at one or more of the time levels t^n. The functional dependence of F on f and q *defines* the particular discrete approximation.

As mentioned above, FCT constructs the net flux $F_{i+(1/2)}$ point by point and timestep by timestep (nonlinearly) as the weighted average of two fluxes, one produced by a "high order" method and the other by a "low order" method. The formal procedure introduced in [13] is as follows:

1. Compute $F^L_{i+(1/2)}$, the "low order fluxes," using a method guaranteed not to generate unphysical values in the solution for the problem at hand.

2. Compute $F^H_{i+(1/2)}$, the "high order fluxes" using a method chosen to be accurate in smooth regions for the problem at hand.
3. *Define* the "antidiffusive fluxes" [2]

$$A_{i+(1/2)} \equiv F^H_{i+(1/2)} - F^L_{i+(1/2)}. \tag{5}$$

4. Compute the time advanced low order ("transported and diffused" [2]) solution:

$$q_i^{td} = q_i^n - \Delta x_i^{-1}\left[F^L_{i+(1/2)} - F^L_{i-(1/2)}\right]. \tag{6}$$

5. *Limit* the antidiffusive fluxes in a manner such that q_i^{n+1} as computed in step 6 below does not take on nonphysical values:

$$A^C_{i+(1/2)} = C_{i+(1/2)} A_{i+(1/2)}, \quad 0 \le C_{i+(1/2)} \le 1. \tag{7}$$

6. Apply the limited antidiffusive fluxes:

$$q_i^{n+1} = q_i^{td} - \Delta x_i^{-1}\left[A^C_{i+(1/2)} - A^C_{i-(1/2)}\right].$$

The critical step in the above is step 5, the flux limiting step. In the absence of step 5 (i.e., $A^C_{i+(1/2)} = A_{i+(1/2)}$), q_i^{n+1} would simply be the time-advanced high order solution.

2.2 Multidimensional Flux-Corrected Transport

Let us see how the procedure above might be implemented in multidimensions. An obvious choice would be to use an operator-splitting technique, splitting along spatial dimensions, when it can be shown that the equations allow such a technique to be used without serious error. Indeed, such a procedure may even be preferable from programming and time-step considerations. However, there are many problems for which such splitting produces unacceptable numerical results, among which are incompressible or nearly incompressible flow fields. The technique is straightforward and shall not be discussed here. Instead, let us now consider the two-dimensional system of conservation laws

$$q(\mathbf{x}, t)_t + f(q, \mathbf{x}, t)_x + g(q, \mathbf{x}, t)_y = 0. \tag{8}$$

A simple example of such a system of equations, and one we consider later, is the system describing two-dimensional ideal inviscid fluid flow, also known as the two-dimensional Euler equations:

$$q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}; \quad f = \begin{pmatrix} \rho u \\ \rho uu + P \\ \rho uv \\ \rho uE + Pu \end{pmatrix}; \quad g = \begin{pmatrix} \rho v \\ \rho vu \\ \rho vv + P \\ \rho vE + Pv \end{pmatrix} \tag{9}$$

where ρ, u, v, P, and E are the fluid density, x velocity, y velocity, pressure, and specific total energy respectively.

If we work on a finite volume coordinate-aligned mesh, we can define our two-dimensional FCT algorithm thus:

$$q_{ij}^{n+1} = q_{ij}^{n} - \Delta V_{ij}^{-1}[F_{i+(1/2),j} - F_{i-(1/2),j} + G_{i,j+(1/2)} - G_{i,j-(1/2)}] \qquad (10)$$

where ΔV_{ij} is the volume of cell ij.

Now there are two sets of transportive fluxes F and G, and the FCT algorithm proceeds as before:

1. Compute $F^{L}_{i+(1/2),j}$ and $G^{L}_{i,j+(1/2)}$, the "low order fluxes," using a method guaranteed not to generate unphysical values in the solution for the problem at hand.
2. Compute $F^{H}_{i+(1/2),j}$ and $G^{H}_{i,j+(1/2)}$, the "high order fluxes," using a method chosen to be accurate in smooth regions for the problem at hand.
3. *Define* the "antidiffusive fluxes" [2]

$$A_{i+(1/2),j} \equiv F^{H}_{i+(1/2),j} - F^{L}_{i+(1/2),j},$$
$$A_{i,j+(1/2)} \equiv G^{H}_{i,j+(1/2)} - G^{L}_{i,j+(1/2)}.$$

4. Compute the time advanced low order ("transported and diffused" [2]) solution:

$$q_{ij}^{td} = q_{ij}^{n} - \Delta V_{ij}^{-1}\left[F^{L}_{i+(1/2),j} - F^{L}_{i-(1/2),j} + G^{L}_{i,j+(1/2)} - G^{L}_{i,j-(1/2)}\right].$$

5. *Limit* the antidiffusive fluxes in a manner such that q_{ij}^{n+1} as computed in step 6 below does not take on nonphysical values:

$$A^{C}_{i+(1/2),j} = C_{i+(1/2),j} A_{i+(1/2),j}, \quad 0 \le C_{i+(1/2),j} \le 1,$$
$$A^{C}_{i,j+(1/2)} = C_{i,j+(1/2)} A_{i,j+(1/2)}, \quad 0 \le C_{i,j+(1/2)} \le 1.$$

6. Apply the limited antidiffusive fluxes:

$$q_{ij}^{n+1} = q_{ij}^{td} - \Delta V_{ij}^{-1}\left[A^{C}_{i+(1/2),j} - A^{C}_{i-(1/2),j} + A^{C}_{i,j+(1/2)} - A^{C}_{i,j-(1/2)}\right].$$

As can be easily seen, implementation of FCT in multidimensions is straightforward, with the possible exception of Step 5, the flux limiter, which will be addressed in a later section.

3 Design Criteria for FCT Algorithms

Here we give, with only modest detail, six criteria that we believe are necessary for the construction of properly designed (robust but accurate) FCT algorithms. They are:

1. The resolving power of the high order fluxes should be as high as is practical. The term "resolving power" will be defined precisely below.
2. The high order fluxes should have a dissipative component which adapts itself to the resolving power of the nondissipative component.
3. The high order fluxes should be "pre-constrained" with respect to physically appropriate bounds before being input to the flux limiter.

4. The low order flux must be dissipative enough to guarantee that unphysical values in the solution cannot be generated, but should otherwise be as accurate as is practical.
5. The flux limiter should accommodate as flexible a specification of solution bounds as possible, and should utilize constraints that have a strong physical basis.
6. The flux limiter should have a simple fail-safe feature that reduces all fluxes into a grid point to their low order values when the normal flux limiter machinery fails.

We will treat each one of these in turn.

3.1 High Order Fluxes with Very High Resolving Power

The primary result of [14] was that there is a significant advantage in using fluxes derived from very high order, spatially-centered finite difference operators (fourth order or higher) for the "high order fluxes" in FCT algorithms. That conclusion was based on some analysis showing a strong empirical relationship between order and resolving power (a term which we define below) for centered finite difference operators, some heuristic reasoning as to how FCT works in practice, and several one-dimensional test calculations dominated by linear advection test problems. We review that work in this section.

For analysis, Eq. (2) is usually reduced to a scalar conservation law and linearized:

$$\frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} = 0 \tag{11}$$

where u is a constant. This is the linear advection equation with advection speed u. Its solution is simply

$$q(x,t) = q(x - ut, 0). \tag{12}$$

That is, the profile is simply translated right or left with velocity u and no change in shape. In Fourier space, this takes the form of each Fourier mode moving with a phase velocity u, with no change in amplitude. Thus all numerical errors associated with a given numerical algorithm can be quantified by a specification of phase velocity error per timestep and amplitude error per timestep as a function of the wavenumber k, and as a function of the discretization step in space and time Δx and Δt respectively.

In [14] we examined a particular algorithm for solving Eq. (11) on a uniform mesh, that of using a leapfrog discretization in time and centered finite differences of arbitrary order in space. This choice allowed us to ignore the amplitude errors entirely, since for the leapfrog discretization these errors vanish for all k and for all Δx and Δt satisfying the Courant condition $\epsilon \equiv |u|\Delta t/\Delta x < 1$. Further noting that the total phase error was the algebraic sum of that induced by the temporal and

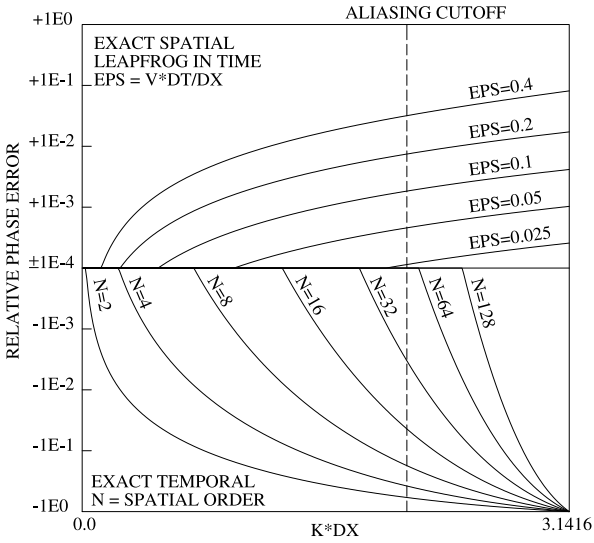

Fig. 1 Plot of relative phase error versus the normalized wavenumber $k\Delta x$ for the leapfrog time-marching scheme and analytic spatial derivatives (*top*), and for centered finite difference schemes of order N and analytic temporal derivatives (*bottom*). Figure taken from Ref. [14]

spatial discretizations separately, as long as both were small, we were able to reduce our algorithm analysis to a single plot, reproduced here in Fig. 1.

The striking aspect of Fig. 1 is the marked effect of the order N of the centered spatial finite difference operator, as well as the timestep Δt, on the resolving power of the algorithm. By "resolving power" we mean, loosely, the ability of an algorithm to maintain low phase errors over a large part of k-space. In more precise terms, we define the resolving power $k_r(E)$ of a given combination of N and Δt to be the largest wavenumber k for which all phase errors are smaller than a pre-specified value E. Thus Fig. 1 tells us that for a given E, we can resolve more and more of k-space if, on a given grid, we simply increase the spatial order N of our centered finite difference operator, decreasing Δt appropriately as we do so. The primary conclusion of [14] was that not only is this statement true for smooth functions, but that it is true in the presence of fronts also, as long as one is treating the fronts with a front-capturing algorithm such as FCT. Nothing since 1981 has dissuaded us from that view, and thus we present it as the first of our FCT design criteria. Computational examples presented later in this chapter will hopefully provide the reader more evidence of its correctness.

3.2 High Order Fluxes with an Adaptive Dissipation Component

Looking at the bottom portion of Fig. 1, we see that the arguments of the last section become less and less convincing as one moves to the extreme right portion of the plot. Phase error in this portion of the plot becomes increasing resistant to reduction by simply increasing N. Indeed, at the Nyquist frequency $k\Delta x = \pi$, the phase error is -100% regardless of how large we make N or how small we make Δt. Thus for

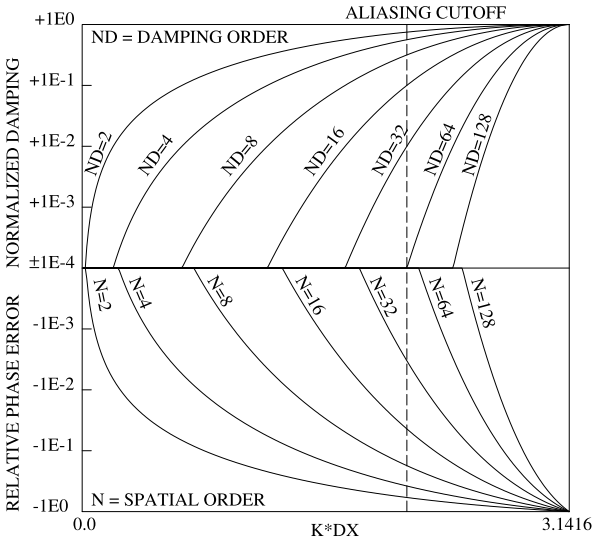

Fig. 2 *Top*: Plot of the damping induced by a centered finite difference approximation to the N_Dth derivative versus the normalized wavenumber $k\Delta x$, normalized so that the Nyquist mode is completely eliminated. *Bottom*: Same as bottom of Fig. 1: Plot of relative phase error versus the normalized wavenumber $k\Delta x$ for centered finite difference schemes of order N and analytic temporal derivatives

any given N, there will be some portion of k-space which is resolved poorly. Logic dictates that these Fourier modes be damped rather than carried with an erroneous speed, and that any given Fourier mode be damped nearly entirely by the time it is 180 degrees out of phase with the analytic result. Clearly, from Fig. 1, the functional dependence of this dissipation on k must itself depend on N if we are to achieve this result without damping the modes which are actually being carried accurately.

We have found that one can form such dissipative fluxes from the centered finite difference forms of $d^{N_D}q/dx^{N_D}$ where $N_D \leq N+2$. One then simply adds these fluxes to FCT's "high order" fluxes. Indeed, the early work of Kreiss and Oliger [9] contained calculations of the linear advection of a triangular wave using $N_D = N = 4$ and leapfrog time differencing, with results that, although not oscillation free, were considerably better than without the dissipation. These results are what prompted us to use such forms to construct FCT algorithms, and we have found them to be of sufficient value to include them among our design criteria.

In the top portion of Fig. 2 we plot the damping induced by centered finite difference approximations to $d^{N_D}q/dx^{N_D}$ versus $k\Delta x$, normalized so that the Nyquist mode is completely eliminated. In the bottom portion, we simply reproduce the bottom portion of Fig. 1. Focusing our attention on pairs of lines for which $N = N_D$, we see a rather remarkable match between phase error amplitude and dissipation amplitude as a function of $k\Delta x$. That is, for such pairs, as the relative phase error increases, so does the damping, with approximately the same functional dependence on $k\Delta x$. This is in accordance with our expressed desire to induce damping in proportion to the relative phase error. Thus, for the computational examples in this chapter, we have chosen to use $N_D = N$. An equally good case can be made for the choice $N_D = N+2$, since this will leave the overall order of the algorithm intact. Indeed we have made that choice ourselves in some contexts. The specific construction of these operators in flux form is given in the appropriate later sections.

3.3 Imposition of Physically-Motivated Constraints on the High Order Fluxes Before the Flux Limiting Step

In general, the numerical algorithms used to construct high order numerical fluxes from the cell-centered values of q assume by necessity a degree of smoothness to q. Thus near fronts it is not unusual, especially for spatial orders higher than 2, to find that these fluxes violate physically-motivated bounds on their values. One could, of course, take the position that the flux limiter itself will ensure that these fluxes are prevented from producing values of q in the next time step that violate appropriate bounds for q, and thus that these unphysical values for the high order fluxes should be allowed to stand when computing the antidiffusive fluxes A. Nonetheless, it is wise to attempt to address this problem at its source, rather than shift the burden to the flux limiter at a later stage, and simply not allow the high order fluxes to take on values that are clearly outside the bounds of possibility. This design principle is not truly new, although not generally expressed as we have above. The prime example is the default behavior of the original Boris-Book flux limiter [2], which the reader will meet shortly. As explained in [13], this flux limiter sets the antidiffusive flux A to zero in virtually all cases where the antidiffusive flux has the same direction as the gradient of q^{td}, i.e., where A is actually diffusive, and in most cases would not actually cause the adjacent values of q to take on unphysical values. Although it is difficult to make rigorous statements in this context, in the great majority of the cases for which this flux-canceling machinery is active, and for which one can place physically-motivated upper and lower bounds on the value of the flux, the high order flux itself can be shown to be outside those bounds, without resort to arguments about its effect of the subsequent values of q. Another example of constraints imposed on the high order fluxes prior to the primary monotonicity machinery is to be found in the PPM algorithm [5], wherein candidate point values of q at cell interfaces are computed, given its cell averages. Rather than simply calculate these "high order" point values of q in a straightforward way, Colella and Woodward use a multistep algorithm that utilizes MUSCL slope limiting in a way that guarantees that the candidate "high order" interface value of q is bounded by the corresponding cell averages at the adjacent grid points. The motivation and the effects of this "pre-limiting" is similar to that of the Boris-Book limiter.

If one can place rigorous physically-motivated bounds on the high order flux, this step can be quite simple, as well as quite effective. We will see an example of such a situation when we construct a non-clipping flux limiter for advection in the next section. However, in general it is difficult to find such rigorous bounds, for at least two reasons. The first is that the flux $\mathbf{f}$ lives in a space one dimension lower than the corresponding q. For example, in three spatial dimensions in a finite volume context, q represents a volume average over the cell while the fluxes $\mathbf{f}$ are area averages at cell faces. The second is that in general $\mathbf{f}$ is a nonlinear function of q. These two combine to make it quite difficult to reliably place bounds on the high order fluxes. Thus we often use the reasoning implicit in the Boris-Book limiter: if the antidiffusive flux is actually diffusive, then there must be something wrong with the high order flux, and we set that antidiffusive flux to zero before limiting. We are

not happy with the lack of rigor associated with that choice, but empirically we have found that this is the correct action to take in most cases.

3.4 Low Order Fluxes That Guarantee That Physical Bounds Are Not Violated

The prime requirement of FCT's low order flux is that it be guaranteed to produce a solution free of unphysical grid point values. If this cannot be guaranteed, then there is no hope of guaranteeing that property in the final FCT solution, even with fail-safe limiting (item 6). One way to satisfy this requirement is to simply incorporate enough numerical dissipation in the algorithm, but overkill here is to be avoided (see below). Another way is to use first-order upwind methods or kinetics-based methods such as the beam scheme of Sanders and Prendergast. When properly chosen, these are probably the best choice, for their inherent dissipation is usually close to the minimum required to meet the prime requirement. However, be aware that many such schemes using "approximate" Riemann solvers cannot meet the prime requirement. It should be obvious that, within the prime requirement, the low order algorithm should be as accurate as is practical. In particular, any low order flux with more dissipation than is necessary to meet the prime requirement is harmful, putting an extra burden on the flux limiter, which is arguably the weakest link of the algorithm. Thus, for example, one would not use a Lax-Friedrichs flux for an advection problem, since the less diffusive donor cell flux already satisfies the prime requirement. As another example, if we are solving a two-dimensional advection problem, and we must choose between two donor cell algorithms, the first of which allows corner transport, and the second of which does not, we would choose the former as long as it satisfied the prime requirement.

3.5 Flexible Flux Limiters That Utilize Constraints with a Strong Physical Basis

It is useful to consider a flux limiter as being comprised of two components:

1. A physics component which specifies physically-motivated upper and lower bounds on grid point values in the next time step; and
2. An algorithmic machinery component for enforcing the above bounds.

It is clear that the algorithmic machinery component must have sufficient flexibility to accommodate the needs of the physics component. Thus, in our view, a good flux limiter must possess both a robust and accurate physics component and a flexible algorithmic machinery component. There exist several flux limiters which are of extremely simple form, expressible in as little as one line of Fortran. The original flux limiter of Boris and Book, and the ones typically used in algorithms

that describe themselves as "TVD" are prime examples. The simplicity of these flux limiters is due to an extremely simple physics component and a very inflexible algorithmic machinery component: they make rather strong assumptions about what constitutes proper upper and lower bounds on the solution at a given grid point at a given timestep, and their algorithmic machinery is capable of accommodating those strong assumptions and little more. These assumptions may be overly restrictive, as they are in the case of the "clipping" phenomenon, or overly loose, as in the case of the "terracing" phenomenon. Thus we strongly encourage the reader to derive the above flux limiters for himself or herself. This exercise will reveal the assumptions implicit in these limiters, and allow an assessment as to their appropriateness for the problem at hand. If they are not appropriate, the reader may wish to consider a more flexible flux limiter, albeit one probably more complex and longer that one line of Fortran. We give an example of a more flexible flux limiter later. In our view the most difficult issue in the design of flux limiters is the physics component. Most commonly the antidiffusive fluxes are those which update conservative variables directly, and hence the default choice for most FCT algorithms has been to constrain the conservative variables using the default bounds built into the Boris-Book flux limiter. But this can often be a bad choice. Even if we were to circumvent the default bounds by using more flexible algorithmic machinery, it can often be extremely difficult to determine the appropriate bounds for the conserved variables. Using gas dynamics as an example, one would be hard pressed to specify the physically appropriate bounds on mass, momentum, and energy per unit volume at a given grid point at a given timestep, even if he or she were given the complete time history of the computed solution up to that point. We know that the physics allows the formation of new extrema in all three of these quantities. Thus looking at the adjacent grid point values at the previous time step, or even at the values of the time-advanced low order solution (FCT's default), can lead to bounds on the solution that are not physically appropriate. When we discuss the construction of FCT algorithms for gas dynamics in Sect. 5, we will put forth the hypothesis that much more reliable constraints are to be obtained by performing the flux limiting step in characteristic variables rather than conservative variables.

3.6 Failsafe Flux Limiters

We suppose that this topic comes under the general heading of "dirty laundry," but it cannot be ignored in any objective discussion of front-capturing algorithms. If one is attempting to solve difficult problems, our experience is that, no matter how carefully one tries to design algorithms that are consistent both with numerical analysis and with the physics problem one is attempting to solve, there will be situations in which at least one grid point at least one time step takes on values that are outside the bounds of physical possibility. For FCT and similar algorithms, these will usually be variables that one is not directly constraining. For example, if one is performing flux limiting on the conserved variables (not recommended here, but done often by

many users of FCT), the density by construction can never become negative, but the internal energy can do so, since it is not directly constrained by the limiting process. What should one do in this case? As another example that we ourselves will face later in this chapter in the Woodward-Colella double shock tube problem, even when we limit with respect to what in our view are the most physically appropriate variables, the characteristic variables, we can occasionally generate unphysical grid point values. The characteristic variables are, after all, a linearization, and a linearization can be inaccurate at large jumps. Thus we could generate negative internal energies in that case also (or, in theory, even negative densities!). What action should we take when this happens?

Prudence, if nothing else, would dictate that the algorithm make provisions for such a case. Assuming we do not wish to simply terminate the calculation, we desire a solution which is as consistent with the design philosophy of the algorithm as possible and, most important of all, is explicitly stated.

For FCT, we believe that there is an obvious solution consistent with FCT's design, and with the numerical analysis goal of being at least first order accurate, and that is the one we choose here: For any such offending grid point, we iteratively drive all the fluxes into or out of that grid point toward their low order values, until the offense is eliminated. Thus it is especially important that FCT's low order scheme be guaranteed to be free of unphysical values! We use an especially simple algorithm here, which we describe in a later section.

4 FCT Algorithms for One Dimensional Linear Advection

We wish to give the reader an idea of the kind of performance one can expect of an FCT algorithm for the simplest of scalar conservation laws, linear advection. That is, we have Eq. (2) with $f = qu$ and u a constant. We use a uniform spatial mesh of cell size Δx. We utilize a method of lines approach, choosing our spatial and temporal discretization independently. All temporal discretizations we shall use (e.g., modified Euler, explicit Runge-Kutta, leapfrog, leapfrog-trapezoidal) involve one or more leapfrog-like substeps of the following form:

$$q_i^{n+1} = q_i^n - \Delta x_i^{-1}[F_{i+(1/2)} - F_{i-(1/2)}]. \tag{13}$$

Here t^{n+1} and t^n are substep time levels associated with a particular substep, with associated timestep $\Delta t^{n+1/2}$. The fluxes F are functions of f at one or more of the time levels, not necessarily t^{n+1} and t^n. The timestep $\Delta t^{n+1/2}$ has been absorbed into the definition of the fluxes. This leapfrog-like substep will be used as the fundamental building block for any time discretization we use. Thus we can describe our treatment for all temporal discretizations by describing our treatment of this substep.

Our low order flux for advection is given by the first order upwind scheme:

$$F_{i+(1/2)}^L = \left[\frac{1}{2}\left(f_{i+1}^n + f_i^n\right) - \frac{1}{2}|u|\left(q_{i+1}^n - q_i^n\right)\right]\Delta t^{n+1/2}. \tag{14}$$

The high order fluxes are given by the formulae in the Appendix of [13]. As an example, the fourth order flux is given by:

$$F^{H4}_{i+(1/2)} = \left[\frac{7}{12}\left(f^a_{i+1} + f^a_i\right) - \frac{1}{12}\left(f^a_{i+2} + f^a_{i-1}\right)\right]\Delta t^{n+1/2} \tag{15}$$

where the time level t^a is meant to denote whatever time level or average of time levels is required by the particular substep of the particular time discretization chosen.

The high order dissipative fluxes of order N_D, which are added to the above high order fluxes, are simply the flux form representation of $\partial^{N_D} q/\partial x^{N_D}$, normalized to damp the Nyquist mode completely in one timestep at a Courant number of unity. As an example, the order 4 dissipative flux is given by:

$$F^{D4}_{i+(1/2)} = -|u|\left[\frac{3}{16}\left(q^n_{i+1} - q^n_i\right) - \frac{1}{16}\left(q^n_{i+2} - q^n_{i-1}\right)\right]\Delta t^{n+1/2}. \tag{16}$$

Thus far we have dealt with only three of our six FCT design criteria, the design of the high and low order fluxes. The other three are the pre-constraint of the high order fluxes, the construction of the flux limiter, and the failsafe limiter. A failsafe limiter is not needed here, since we are directly constraining the only variable of interest. For the moment, we will choose a simple default for the remaining two criteria, the original Boris-Book limiter:

$$A^C_{i+(1/2)} = S\max\left(0, \min\left(|A_{i+(1/2)}|, S\left(q^{td}_{i+2} - q^{td}_{i+1}\right)\Delta x, S\left(q^{td}_i - q^{td}_{i-1}\right)\Delta x\right)\right)$$
$$\text{where } S \equiv \text{sign}(1, A_{i+(1/2)}). \tag{17}$$

This simple formula implicitly determines our choices for the remaining two design criteria. These choices turn out to be reasonable for this advection problem, at least away from extrema. However, as we will shortly see, we can improve on FCT's performance at extrema by addressing these remaining two criteria explicitly.

All of the tests in this section use the classic explicit fourth order Runge-Kutta time discretization, each substep of which is treated in the manner described above.

4.1 Tests of FCT Advection Algorithms on Three Classic Test Problems

In [15] we compared a number of advection algorithms on three test problems chosen from the open literature: the square wave test of Boris and Book [2], the Gaussian of Forester [7], and the semi-ellipse of McDonald [10]. The first test consists of a square wave 20 cells wide to be advected 800 time steps at a Courant number of 0.2. The second test consists of a Gaussian of half width 2 cells to be advected 600 time steps at a Courant number of 0.1. The third and final test consists of a semi-ellipse of radius 15 cells to be advected 600 time steps at a Courant number

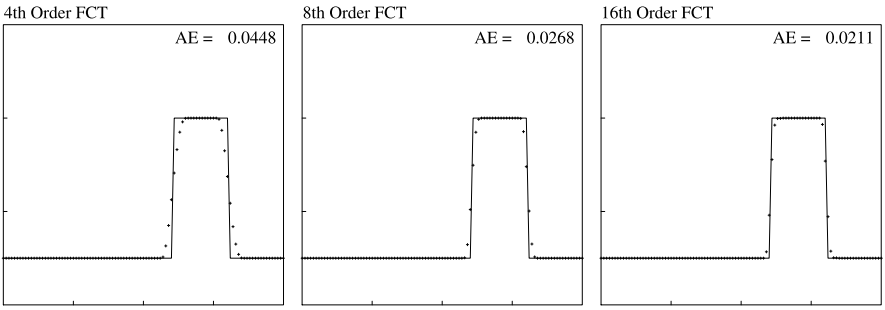

Fig. 3 Results for the Boris-Book square wave using FCT algorithms with high order fluxes of fourth, eighth, and sixteenth order. The analytic solution is shown as a *solid line*, while the computed solution is shown as *discrete data points*. The L_1 error is denoted "AE" in the plots, for consistency with the original plots of Boris and Book. Note the marked improvement with resolving power

of 0.1. We will use those same test problems here to demonstrate various aspects of the FCT algorithms we have just described.

In Fig. 3, we examine the effect that we had observed in our earlier work [14]. Running the square wave problem, we vary only the order of the high order flux from fourth to eighth to sixteenth, and see a marked increase in the resolution of the discontinuities. Our interpretation of this effect was, and continues to be, that since the discontinuity is the result of the superposition of a large number of Fourier modes, with precise phase relationships being critical, increasing the resolving power, i.e., the percentage of k-space for which the phase speed is accurate, makes it possible for the flux limiter to introduce less and less dissipation to prevent unphysical values, thus yielding more accurate results.

In Fig. 4, we show the same sequence of algorithms, but for the semi-ellipse of McDonald. Again we see an increase of performance with resolving power. However, this problem is prone to the "terracing" phenomenon, some hints of which can be seen at the right edge of the semi-ellipse. To show the value of the dissipative component of the high order flux, we show the same problem with the same set of algorithms in Fig. 5, but with the dissipative component eliminated. Although not as dramatic as the effect of increasing the resolving power of the high order flux, the dissipative flux clearly is of value in preventing the occurrence of errors that are not detected by the flux limiter. What is happening here is that dispersive oscillations are being shed by the leading (right) edge of the semi-ellipse. As they propagate into the semi-ellipse they are not detected as oscillations because they are hidden by the large gradient in the right side of the ellipse. As they get closer to the center of the ellipse, they try to take the form of true extrema, at which point they are prevented from doing so by the flux limiter. The damage is already done, however. The effect of the dissipative component in the high order flux is to damp the modes moving with the wrong phase velocity before the fact. Calculations like these, as well as analytic arguments, are the reason we believe that some high order dissipation should be present in most calculations, whether one is using front capturing techniques or not.

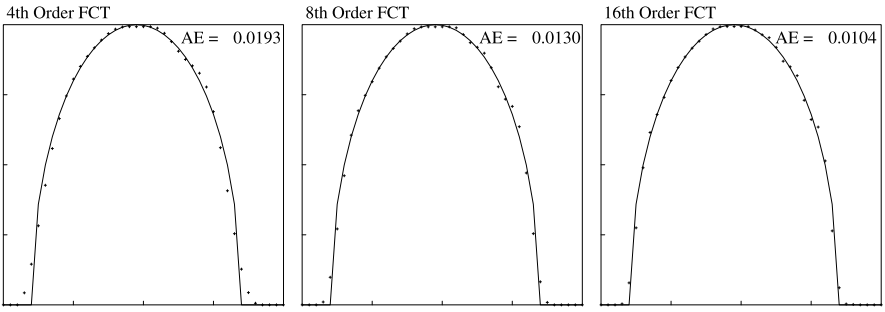

Fig. 4 Results for the semi-ellipse of McDonald using FCT algorithms with high order fluxes of fourth, eighth, and sixteenth order. The analytic solution is shown as a *solid line*, while the computed solution is shown as *discrete data points*. The L_1 error is denoted "AE" in the plots. Note the improvement, albeit modest, with increased resolving power. Although mitigated significantly by the high order dissipation, clear hints of the terracing phenomenon are still visible. Compare to Fig. 5

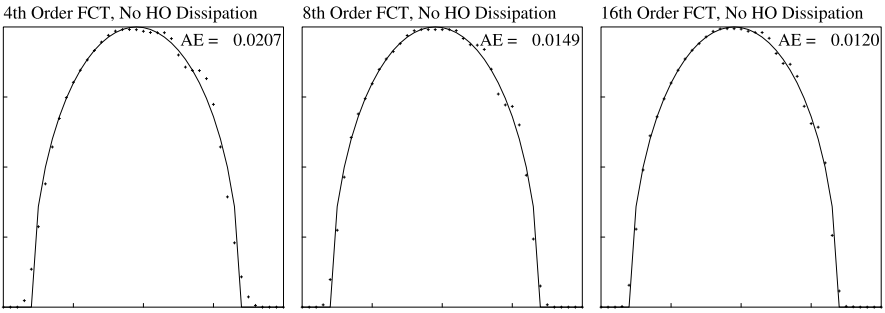

Fig. 5 Same as Fig. 4, but with the dissipative component of the high order fluxes removed. Note the "terracing" phenomenon on the right edge of the semi-ellipse, which is the result of dispersive waves being ignored by the flux limiter until they attempt to become extrema. Compare to Fig. 4

The previous set of calculations was an example of one way a flux limiter can fail. In that case, the flux limiter failed to perceive and prevent an error because its definition of an error was the creation of new extrema in q. Note that the algorithmic machinery component of the flux limiter did not fail, but rather its physics component. In this case its "physics" criterion for what constituted an error was too weak. In the next set of calculations, we see an example where exactly the same criterion is too strong, preventing the formation of an extremum when it is physically allowable. In Fig. 6 we show the same sequence of algorithms, but for the Gaussian of Forester. Although we again see the same pattern of increased performance with increased resolving power, we also see the well-known "clipping" problem. Here, as the true peak of the Gaussian passes between grid point centers, the true grid point extrema value should increase and decrease in an oscillatory fashion. However, the flux limiter used here does not allow for that possibility, treating all attempts to accentuate an extremum during a time step as an error to be prevented. The problem

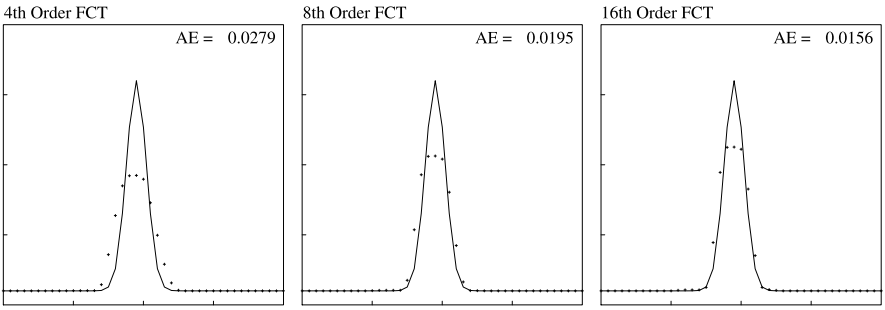

Fig. 6 Results for the Gaussian of Forester using FCT algorithms with high order fluxes of fourth, eighth, and sixteenth order. The analytic solution is shown as a *solid line*, while the computed solution is shown as *discrete data points*. The L_1 error is denoted "AE" in the plots. Again we see improvement with resolving power, but the errors are dominated by the "clipping" phenomenon

can be addressed by using a more flexible limiter and a better estimate of the allowable upper and lower bounds on the solution, as we show in the next subsection.

4.2 An Alternative to the Boris-Book Flux Limiter

In [13], we described a new flux-limiting algorithm for FCT. Although developed primarily to allow the construction of fully multidimensional FCT algorithms, that flux limiter also allowed a much more flexible specification of upper and lower bounds on the solution than did the original Boris-Book limiter Eq. (17). In particular, it allowed the construction of flux limiters which do not clip physical extrema. We describe that algorithm in one spatial dimension in this section, and then use it to construct a non-clipping flux limiter for one dimensional advection. In Sect. 5 we describe and use the algorithm in two spatial dimensions.

In words, the alternative flux limiter constrains the solution by first computing two independent sets of provisional coefficients $C_{i+(1/2)}$ for each antidiffusive flux, one to enforce the user-supplied upper bounds on the solution, and the other to enforce the user-supplied lower bounds. Both bounds are satisfied simply by choosing the final coefficients to be the minimum of the two provisional coefficients.

The upper bounds constraint is computed by dividing Q_i^+, the maximum allowable net flux into a cell, by P_i^+, the sum of all those fluxes whose effect is to increase the value of q_i. That fraction, bounded by 0 and 1, is provisionally assigned to the $C_{i+(1/2)}$ of each of those fluxes. A similar procedure is undertaken for the lower bounds constraint, and still another provisional value of $C_{i+(1/2)}$ assigned to each of the fluxes whose effect is to decrease the value of q_i. The net $C_{i+(1/2)}$ is simply the minimum of the two temporary values. From the above description it should be clear that this limiter is unambiguously defined for any number of spatial dimensions and for both structured and unstructured meshes, as long as the difference between the low and high order components can be written as fluxes flowing between adjacent cells. In one spatial dimension, the procedure is as follows:

1. Compute, for each grid point i, physically-motivated upper and lower bounds on the solution in the next timestep, q_i^{max} and q_i^{min} respectively. This step is both flexible and critical, requiring intimate knowledge of the science underlying one's equation. It is important that q_i^{td} already satisfy these bounds.
2. For the upper bound, compute P, Q, and their ratio R at each grid point:

$$P_i^+ = \max(A_{i-(1/2)}, 0) - \min(A_{i+(1/2)}, 0), \tag{18}$$

$$Q_i^+ = \left(q_i^{max} - q_i^{td}\right)\Delta x_i, \tag{19}$$

$$R_i^+ = \min\left(1, Q_i^+ / P_i^+\right), \ P_i^+ > 0, \quad 0 \text{ otherwise.} \tag{20}$$

3. For the lower bound, compute P, Q, and their ratio R at each grid point:

$$P_i^- = \max(A_{i+(1/2)}, 0) - \min(A_{i-(1/2)}, 0), \tag{21}$$

$$Q_i^- = \left(q_i^{td} - q_i^{min}\right)\Delta x_i, \tag{22}$$

$$R_i^- = \min\left(1, Q_i^- / P_i^-\right), \ P_i^- > 0, \quad 0 \text{ otherwise.} \tag{23}$$

4. Compute $C_{i+(1/2)}$ by taking a minimum:

$$C_{i+(1/2)} = \begin{cases} \min(R_{i+1}^+, R_i^-) & \text{when } A_{i+(1/2)} > 0, \\ \min(R_i^+, R_{i+1}^-) & \text{when } A_{i+(1/2)} \le 0. \end{cases} \tag{24}$$

Note that in the above we do not specify the equivalent of Eq. (14) in [13], which, as we explained in the previous section, can be thought of as a method for pre-constraining the high order fluxes prior to the flux limiting step. In the case of linear advection in one dimension, we have a much more robust way of pre-limiting those fluxes, as we will see below.

4.3 A Non-clipping Version of the Alternative Flux Limiter

Let us now specify our non-clipping flux limiter for one-dimensional linear advection. To do so we need to define an algorithm for computing q_i^{min} and q_i^{max} above. We also need to address our third criterion and specify an algorithm for pre-constraining our high order fluxes. We shall use a similar approach for both. In Fig. 7, we show a technique we shall use to reconstruct extrema between grid points, for use both in specifying q_i^{min} and q_i^{max} and in pre-constraining our high order fluxes. On each interval $[x_i, x_{i+1}]$ we define $q_{i+(1/2)}^{peak}$ to be the value of q at the intersection of the lines formed by connecting the point (x_{i-1}, q_{i-1}) with (x_i, q_i) and the point (x_{i+1}, q_{i+1}) with (x_{i+2}, q_{i+2}). If the x coordinate of this intersection lies between x_i and x_{i+1}, then we consider this $q_{i+(1/2)}^{peak}$ to be a physically legitimate value for q on the interval $[x_i, x_{i+1}]$.

Let us now define the upper and lower bounds for q on the interval $[x_i, x_{i+1}]$ to be

$$q_{i+(1/2)}^{max} = \max\left(q_i, q_{i+1}, q_{i+(1/2)}^{peak}\right), \tag{25}$$

$$q_{i+(1/2)}^{min} = \min\left(q_i, q_{i+1}, q_{i+(1/2)}^{peak}\right) \tag{26}$$

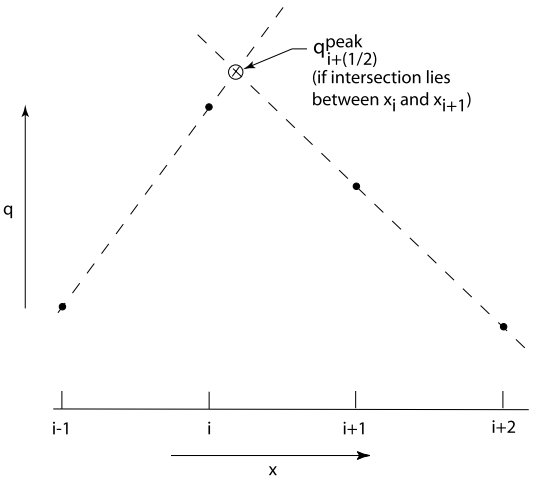

Fig. 7 A possible scheme for extracting information about extrema which exist between grid points at a given point in time. An extremum is assumed to exist between grid points i and $i+1$ if the intersection of the right and left extrapolation of q has an x coordinate between x_i and x_{i+1}. The q coordinate of this intersection in then used both to compute q^{max} and q^{min}, and to pre-constrain the high order flux $F^H_{i+(1/2)}$ (see text)

where all quantities are evaluated at time level n.

We are now in a position to introduce the physics of the problem into the flux limiter. Given that this is an advection problem, and that we choose our Courant number to be less than unity, we know that q_i^{n+1} must be bounded by $q^{min}_{i-(1/2)}$ and $q^{max}_{i-(1/2)}$ if $u > 0$, and by $q^{min}_{i+(1/2)}$ and $q^{max}_{i+(1/2)}$ if $u < 0$. Thus we take

$$q_i^{min} = \begin{cases} \min(q_i^{td}, q^{min}_{i-(1/2)}) & \text{when } u > 0, \\ \min(q_i^{td}, q^{min}_{i+(1/2)}) & \text{when } u \le 0, \end{cases} \tag{27}$$

$$q_i^{max} = \begin{cases} \max(q_i^{td}, q^{max}_{i-(1/2)}) & \text{when } u > 0, \\ \max(q_i^{td}, q^{max}_{i+(1/2)}) & \text{when } u \le 0. \end{cases} \tag{28}$$

Finally we specify our pre-constraint condition on the high order fluxes. Again we use the physics of the problem. Since this is advection, the physical fluxes $F_{i+(1/2)}$ must be bounded by $uq^{max}_{i+(1/2)}$ and $uq^{min}_{i+(1/2)}$. Thus we define $F^{max}_{i+(1/2)} \equiv \max(uq^{max}_{i+(1/2)}, uq^{min}_{i+(1/2)})$ and $F^{min}_{i+(1/2)} \equiv \min(uq^{max}_{i+(1/2)}, uq^{min}_{i+(1/2)})$, and after computing the unconstrained high order fluxes $F^H_{i+(1/2)}$, we constrain them thus:

$$F^H_{i+(1/2)} = \min\bigl(F^{max}_{i+(1/2)}, \max\bigl(F^{min}_{i+(1/2)}, F^H_{i+(1/2)}\bigr)\bigr). \tag{29}$$

The results of using the above pre-constraint condition and flux limiter are shown in Fig. 8. For sufficiently high resolving power, the clipping phenomenon has been virtually eliminated. We believe that this demonstrates the advantage of using one's knowledge of the physics of the problem to design FCT and other front-capturing algorithms, rather than accepting their default behavior.

Before leaving this section, let us try to design the "ultimate" high order FCT scheme, and see how it performs on the three test problems we have examined in this section. It was shown by Fornberg that the asymptotic limit of an Nth order finite difference scheme on a periodic domain as N goes to infinity is in fact just the

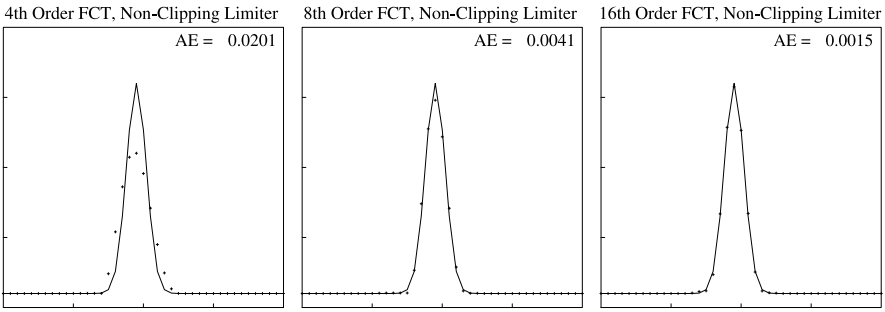

Fig. 8 Same as Fig. 6, but using the non-clipping flux limiter and the pre-constraint of the high order fluxes described in the text. Note the marked increase of accuracy with increased resolving power. Also note that the clipping can be virtually eliminated as long as one has sufficient resolving power

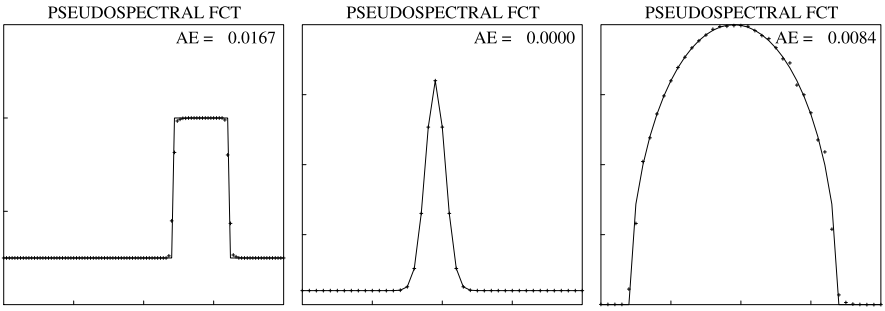

Fig. 9 Performance of a pseudospectral FCT algorithm on all three of the test problems used in this section. We have used the non-clipping flux limiter and the pre-constraint of the high order fluxes described in the text. These are the best results we have been able to produce for these problems using front-capturing algorithms whose high-order components are stable

pseudospectral approximation using Fourier modes as basis functions. Thus we will take our high order fluxes to be those which reproduce the pseudospectral discretization. The results of using these pseudospectral fluxes and the non-clipping limiter and pre-constraint algorithm described above are shown in Fig. 9. These are the best results we have been able to produce for these problems using front-capturing algorithms whose high-order components are stable. (Using unstable schemes as the high order component of front-capturing algorithms can produce extremely sharp square waves, but can severely distort the Gaussian and semi-ellipse. Examples are Superbee, Ultrabee, ACM, and the contact detection algorithm in PPM.)

The above examples provide a springboard for the next section, where we address a nonlinear system of equations, and the construction of our FCT algorithms will become more complex. The path we will choose to success will be the same, however: We will incorporate as much knowledge of the physics as possible into the design of the algorithm.

5 FCT Algorithms for One Dimensional Nonlinear Systems of Conservation Laws

We now consider Eq. (2), where f is a fully nonlinear function of q. The example we shall use is that of the Euler equations (3). If we go down our list of FCT design criteria, we find that many of the optimal choices are the same as those for linear advection, or are obvious generalizations thereof. However, the proper construction of the flux limiter, both with regard to its physics component and with regard to its algorithmic machinery, is less than obvious. Hence we will focus primarily on the construction of the flux limiter in this section.

We saw in the last section that the results of an FCT calculation can be sensitive to the choice of the flux limiter. But we also learned that modest modifications of the basic FCT machinery allowed us to produce results which are quite good. Thus it is worth looking at both the advection equation and at the advection flux limiters for the purpose of determining how we wish to proceed for systems of hyperbolic conservation laws.

The simplicity of the advection equation allows us to make some rather strong statements about the allowable bounds for q in the next timestep. In particular, we know that for any Courant number less than unity, the value of q_i^{n+1} is bounded by the values of q^n on the interval $[x_{i-1}, x_{i+1}]$. This fact allows us to construct reasonably precise upper and lower bounds on the solution. The bounds for q_i^{n+1} used by the Boris and Book flux limiter Eq. (17) are simply the maximum and minimum of $(q_{i-1}^{td}, q_i^{td}, q_{i+1}^{td})$ respectively. While this could certainly be refined, we saw in the last section that this algorithm produces reasonable results if one is willing to tolerate the clipping of extrema. The reason for this, we believe, is that the built-it physics component of the limiter is reasonably close to one physically appropriate to the advection problem, again except near extrema.

5.1 Hyperbolic Systems of Conservation Laws: The Case for Characteristic Variables

Let us now consider systems of hyperbolic conservation laws, using the Euler equations as an example. If we choose to deal completely with the conserved variables q, what sort of statements can we make about upper and lower bounds on q in the next timestep? In contrast to the case for advection, we are at a loss. In fact we know for certain that q_i^{n+1} is not necessarily bounded by q^n anywhere in the vicinity of grid point i. The default route taken by most FCT algorithms is to simply use what worked for advection, and take the upper and lower bounds for q^{n+1} to be the maximum and minimum of $(q_{i-1}^{td}, q_i^{td}, q_{i+1}^{td})$ respectively. While certainly better than the disastrous choice of using the maximum and minimum of $(q_{i-1}^{n}, q_i^{n}, q_{i+1}^{n})$, in contrast to the case for advection, brand new extrema will be a common occurrence. The default choice would allow strong suppression of these new extrema by the combination of the flux limiter and the dissipation in the low order fluxes. Using the

non-clipping flux limiter described in the last section would not help either, again because we have no basis by which to determine the magnitude that a new extremum could attain before being declared unphysical. These difficulties are reflected in the difficulty that FCT has, in our experience, in attaining the same kind of clean results for systems of equations that are relatively easy to come by for scalar equations, when using the default strategy of flux limiting with respect to the conserved variables directly. This is because we have strong and simple statements that we can make about the upper and lower bounds in q in the next time step for the scalar case that we just don't have in the case of systems.

However, there is a set of variables in which a one-dimensional hyperbolic system looks exactly like an advection equation, a set of uncoupled advection equations to be precise, and that is the set of characteristic variables. These variables are not global variables, but rather the result of locally linearizing the equations. Briefly, what we will do in what follows is take the entire flux limiting problem, consisting of the low order solution q^{td} and the set of "antidiffusive fluxes" A, and transform them both into a set of variables in which the same flux limiting problem looks like a set of uncoupled linear advection flux limiting problems. We then limit the fluxes using constraints physically and mathematically appropriate to an advection problem, and then transform the limited fluxes back into conserved variables, where they will be applied to q^{td} to produce the new solution q^{n+1}. This will produce results that in our view are far superior to those produced using the conserved variables directly.

Let us specialize our one-dimensional system of conservation laws Eq. (2) to the case of a flux function f which is solely a function of q (the usual case), and write it in the more compact notation

$$q_t + f(q)_x = 0 \tag{30}$$

where the subscript denotes partial differentiation. We can then further rewrite Eq. (30) in the following form:

$$q_t + A(q)q_x = 0 \tag{31}$$

where A is the $m \times m$ Jacobian matrix $\partial f/\partial q$, and m is the number of conservation laws in Eq. (30). It is not clear that Eq. (31) is any improvement over Eq. (30), since we have lost our explicit conservation form, and in general most entries of A are nonzero. If our goal is to find an appropriate set of constraints on the values of q for the purpose of flux limiting, we apparently have made no progress. But for most hyperbolic systems it is possible to find a new set of variables q' defined by a transformation matrix $T^{-1} = T^{-1}(q)$ such that Eq. (31) takes the form of a series of m decoupled advection equations:

$$T^{-1}q_t + T^{-1}A(q)TT^{-1}q_x = 0, \tag{32}$$

$$q'_t + \Lambda q'_x = 0 \tag{33}$$

where $q' = T^{-1}q$ and $\Lambda = T^{-1}A(q)T$ is a diagonal matrix with diagonal elements λ_j, $1 \le j \le m$.

Specifically, we will have m scalar equations of the form

$$\frac{\partial q_j'}{\partial t} + \lambda_j \frac{\partial q_j'}{\partial x} = 0. \tag{34}$$

Our problem has now been reduced to performing the flux limiting step for m independent advection equations, something that we know how to do well, precisely because we have very good information on how to constrain the solution, as we demonstrated in the last section. We note that the characteristic variables q' are not the conserved quantities, and that we wish to construct our FCT algorithm such that the conserved variables q are updated in flux form. Thus it will be important to transform the fluxes themselves between the two spaces, not just the solution vectors.

To construct our characteristic variable (CV) flux limiter, we first look at the basic Boris-Book limiter, Eq. (17)

$$A^C_{i+(1/2)} = S \max\big(0, \min\big(|A_{i+(1/2)}|, S\big(q^{td}_{i+2} - q^{td}_{i+1}\big)\Delta x, S\big(q^{td}_{i} - q^{td}_{i-1}\big)\Delta x\big)\big). \tag{35}$$

This one-line formula provides one possible answer to the following question, which we term the "Flux Limiting Problem" (FLP) for advection: Given the time-advanced low order solution q^{td} and perhaps other auxiliary solution vectors, and given a set of antidiffusive fluxes A, what is a set of corrected antidiffusive fluxes A^C that are as close to A as possible, and that will constrain q^{n+1} to lie within the bounds appropriate to the advection problem? The FLP requires at least two inputs q^{td} and A, and asks for one output A^C. A look at Eq. (35), however, will convince the reader that q^{td} itself is not really needed, but rather only its first differences at flux evaluation points $i + (1/2)$. This is not atypical. All flux limiting algorithms of which we are aware have the property of depending only on local variations in q, not on the values of q themselves. This observation makes the construction of a CV limiter particularly simple.

5.2 A Characteristic Variable Implementation of the Boris-Book Flux Limiter

To be concrete here, we present a version of the CV flux limiter using the Boris-Book limiter as a building block. Using other limiters as building blocks may require some modification which will hopefully be obvious to the reader.

Given a hyperbolic system of conservation laws of length m with components j, $1 \le j \le m$ in one spatial dimension, a low-order solution vector q^{td} with components denoted $q^{td(j)}$, $1 \le j \le m$ defined on grid points x_i, and a vector of antidiffusive fluxes A with components denoted $A^{(j)}$, $1 \le j \le m$ defined at flux points $x_{i+(1/2)}$, the following steps define a characteristic variable-based implementation of the Boris-Book flux limiter.

1. Calculate some appropriate average $q^{td(j)}_{i+(1/2)}$, $\forall j, i$.

2. From $q^{td}_{i+(1/2)}$ calculate $T^{-1}_{i+(1/2)}$ and $T_{i+(1/2)}$, $\forall i$.
3. Set $D_{i+(1/2)} = T^{-1}_{i+(1/2)}(q^{td}_{i+1} - q^{td}_i)$, $\forall i$.
4. Set $B_{i+(1/2)} = T^{-1}_{i+(1/2)} A_{i+(1/2)}$, $\forall i$.
5. Set $B^{C(j)}_{i-(1/2)} = S\max(0, \min(|B^{(j)}_{i+(1/2)}|, SD^{(j)}_{i+(3/2)}\Delta x, SD^{(j)}_{i-(1/2)}\Delta x))$, $\forall j, i$, where $S = \text{sign}(1, B^{(j)}_{i+(1/2)})$.
6. Set $A^C_{i+(1/2)} = T_{i+(1/2)} B^C_{i+(1/2)}$, $\forall i$.

The notation above uses the superscript (j) on quantities only when it is necessary to emphasize that each component of the vector is being manipulated separately. Otherwise when the superscript is not present, a vector operation of length m is assumed.

5.3 Computational Examples: The One Dimensional Euler Equations

The equations of interest are

$$w = \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix}; \qquad f = \begin{pmatrix} \rho u \\ \rho u u + P \\ \rho u E + P u \end{pmatrix} \tag{36}$$

where ρ, u, P, and E are the fluid density, velocity, pressure, and specific total energy respectively. We will assume an ideal gas equation of state

$$P = (\gamma - 1)\left(\rho E - \frac{1}{2}\rho u^2\right). \tag{37}$$

The matrices T and T^{-1} that we shall need are found by first setting

$$|A - \lambda I| = 0$$

and solving for the eigenvalues λ_j of A. Then for each of these eigenvalues the right and left eigenvectors are found. T is the matrix whose columns are the right eigenvectors of A. T^{-1} is the matrix whose rows are the corresponding left eigenvectors of A. These matrices are well known for this system. They are

$$T = \begin{bmatrix} 1 & 1 & 1 \\ u - c & u & u + c \\ H - uc & \frac{1}{2}u^2 & H + uc \end{bmatrix},$$

$$T^{-1} = \begin{bmatrix} \frac{1}{2}(\frac{\gamma-1}{2}M^2 + \frac{u}{c}) & -\frac{1}{2c} - \frac{(\gamma-1)u}{2c^2} & \frac{\gamma-1}{2c^2} \\ 1 - \frac{\gamma-1}{2}M^2 & \frac{(\gamma-1)u}{c^2} & -\frac{\gamma-1}{c^2} \\ \frac{1}{2}(\frac{\gamma-1}{2}M^2 - \frac{u}{c}) & \frac{1}{2c} - \frac{(\gamma-1)u}{2c^2} & \frac{\gamma-1}{2c^2} \end{bmatrix}$$

where $M^2 \equiv u^2/c^2$, $c^2 = \gamma P/\rho$, and $H = c^2/(\gamma - 1) + u^2/2$ is the stagnation enthalpy.

For our low order flux for the Euler equations we choose the Rusanov scheme:

$$F^L_{i+(1/2)} = \left[\frac{1}{2}\left(f^n_{i+1} + f^n_i\right) - \frac{1}{4}(Q_i + Q_{i+1})\left(q^n_{i+1} - q^n_i\right)\right]\Delta t^{n+1/2} \tag{38}$$

where Q_i is the maximum characteristic speed at i:

$$Q_i = |u_i| + c_i. \tag{39}$$

The high order fluxes are as given before for advection. As an example, the fourth order flux is given by:

$$F^{H4}_{i+(1/2)} = \left[\frac{7}{12}\left(f^a_{i+1} + f^a_i\right) - \frac{1}{12}\left(f^a_{i+2} + f^a_{i-1}\right)\right]\Delta t^{n+1/2} \tag{40}$$

where the time level t^a is meant to denote whatever time level or average of time levels is required by the particular substep of the particular time discretization chosen.

The high order dissipative fluxes are modified versions of those used for advection, with the advection speed u replaced by the maximum characteristic speed Q. As an example, the order 4 dissipative flux is given by:

$$F^{D4}_{i+(1/2)} = -\frac{1}{2}(Q_i + Q_{i+1})\left[\frac{3}{16}\left(q^n_{i+1} - q^n_i\right) - \frac{1}{16}\left(q^n_{i+2} - q^n_{i-1}\right)\right]\Delta t^{n+1/2}. \tag{41}$$

The flux limiter, when we are not using the CV limiter described above, is again given by the original Boris-Book limiter:

$$A^C_{i+(1/2)} = S\max\left(0, \min\left(|A_{i+(1/2)}|, S\left(q^{td}_{i+2} - q^{td}_{i+1}\right)\Delta x, S\left(q^{td}_i - q^{td}_{i-1}\right)\Delta x\right)\right)$$
$$\text{where } S \equiv \text{sign}(1, A_{i+(1/2)}). \tag{42}$$

All of the tests in this section use a modified Euler time discretization, each substep of which is treated in the manner described in Sect. 4.

Our failsafe limiter is the simplest imaginable: If, after flux limiting, either the density or the pressure in a cell is negative, all the fluxes into that cell are set to their low order values, and the grid point values recalculated. Clearly there is much room for a more precise failsafe mechanism, but this one has proved adequate for the problems presented here.

Now that we have described the algorithms we will be using in this section, we show some results using standard test problems. The first is the shock tube problem due to Sod [11]. The initial conditions consist of a single discontinuity in density (8 : 1) and pressure (10 : 1), with both $\gamma = 1.4$ gases at rest. All of our results plot the analytic solution as a solid line, and the computed grid point values as data points, using the temperature field, which we have found to be the field most sensitive to numerical error. In Fig. 10 we show the results of our CV FCT algorithms for $N = N_D = 4$, 8, and 16. From left to right in each of the three plots, the reader will recognize the shock wave, the contact discontinuity, and the rarefaction fan associated with this problem. Note the marked increase in the accuracy of the contact discontinuity as the resolving power of the high order fluxes increases, similar to our experiences with advection. For comparison, in Fig. 11 we show the same

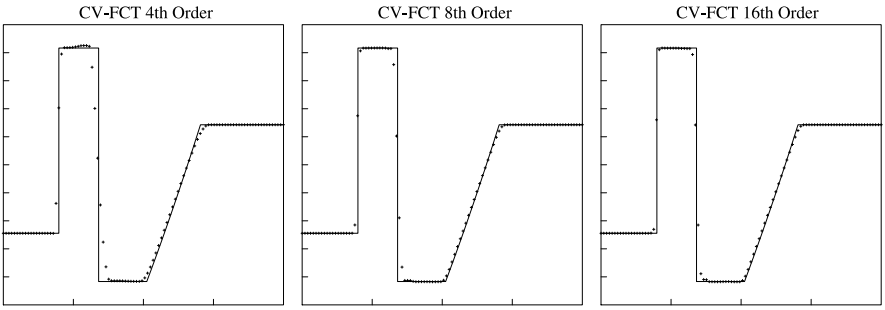

Fig. 10 Results for the temperature field, Sod shock tube problem using CV FCT algorithms with $N = N_D = 4$, 8, and 16

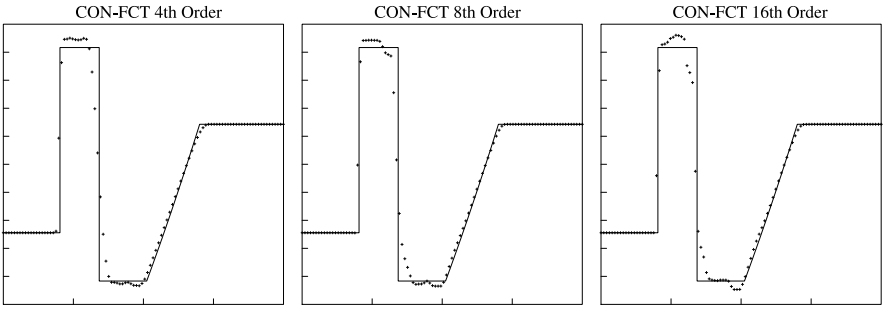

Fig. 11 Same as Fig. 10, but using a flux limiter which limits only with respect to conserved variables

three calculations, but using the more conventional FCT flux limiter which limits the fluxes based solely on the conserved variables. Note the marked superiority of the characteristic variable-based CV flux limiter.

The next test problem is the double shock tube of Woodward and Colella [12]. This problem involves the complex interaction of very strong waves of all types, and is considerably more difficult than the Sod problem. Here we show the performance of our CV FCT algorithms on three grids, of size 200, 400, and 800 grid points, testing both absolute performance and convergence. In Fig. 12, we show the density field at $t = 0.2$ using a grid of 200 points, and using our CV FCT algorithms with $N = N_D = 4$, 8, and 16. As with the advection tests, and with the Sod test problem, we see increased accuracy with resolving power, but all calculations suffer from lack of resolution.

Figures 13 and 14 show the same calculations with 400 and 800 grid points respectively. We see increased accuracy with resolving power, as well as with grid refinement. Note that the two shock waves are resolved over 1–2 grid points regardless of the resolving power, but that the accuracy (sharpness in this case) of the three contact discontinuities increases markedly with increased resolving power at all refinement levels.

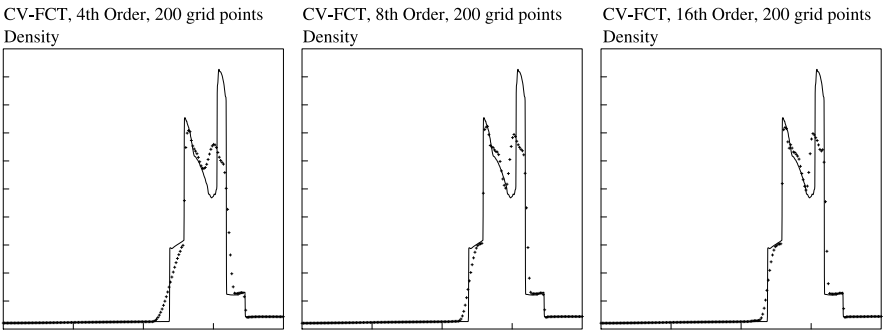

Fig. 12 Results for the density field for the 200-point Woodward-Colella double shock tube problem using CV FCT algorithms with $N = N_D = 4$, 8, and 16

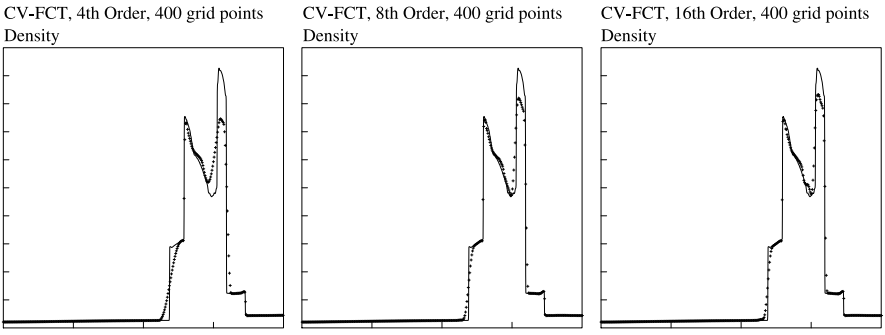

Fig. 13 Same as Fig. 12, but using a grid of 400 points

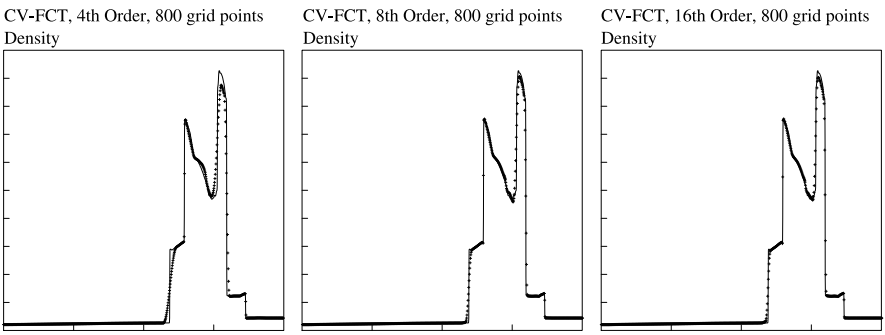

Fig. 14 Same as Fig. 12, but using a grid of 800 points

5.4 Using Characteristic Variables in Other FCT Components

Thus far, we have dealt with the use of characteristic variables only in the flux limiter, but there are two other FCT components that could conceivably benefit from their use: the low order fluxes, and the dissipative component of the high order fluxes. The treatment of both is quite similar, so we will discuss them together. Recall that our low order flux for advection was given by the first order upwind method:

$$F^L_{i+(1/2)} = \left[\frac{1}{2}\left(f^n_{i+1} + f^n_i\right) - \frac{1}{2}|u|\left(q^n_{i+1} - q^n_i\right)\right]\Delta t^{n+1/2}. \tag{43}$$

It can be proven that for any flux of the above form, the coefficient $|u|/2$ used above is the smallest that will guarantee that the flux will maintain the monotonicity of a monotone profile. Thus this flux is in some sense the optimum low order flux for advection.

By contrast, our low order flux for the Euler equations was the Rusanov flux:

$$F^L_{i+(1/2)} = \left[\frac{1}{2}\left(f^n_{i+1} + f^n_i\right) - \frac{1}{4}(Q_i + Q_{i+1})\left(q^n_{i+1} - q^n_i\right)\right]\Delta t^{n+1/2} \tag{44}$$

where Q_i is the maximum characteristic speed at i:

$$Q_i = |u_i| + c_i. \tag{45}$$

This flux is *not* the optimum low order flux for the Euler equations, and is in general considerably more dissipative than necessary to guarantee that unphysical solutions cannot be generated. Rather, the Godunov flux is the optimal choice. This flux requires the solution of the full nonlinear Riemann problem at each flux point. However, a good approximation to the Godunov flux $F^G_{i+(1/2)}$ is obtained by doing exactly what we did to limit fluxes: We transform the entire "low order flux" problem into characteristic variables, where the system is of the form of m uncoupled advection problems, compute first order upwind fluxes in those variables, and then transform the fluxes back to conserved variables:

1. Calculate some appropriate average $q^{n(j)}_{i+(1/2)}, \forall j, i$.
2. From $q^n_{i+(1/2)}$ calculate $T^{-1}_{i+(1/2)}$ and $T_{i+(1/2)}, \forall i$.
3. Set $D_{i+(1/2)} = T^{-1}_{i+(1/2)}(q^n_{i+1} - q^n_i), \forall i$.
4. Set $D^{(j)}_{i+(1/2)} = -\frac{|\lambda^{(j)}_{i+(1/2)}|}{2} D^{(j)}_{i+(1/2)}, \forall i, j$.
5. Set $F^L_{i+(1/2)} = [\frac{1}{2}(f^n_{i+1} + f^n_i) + T_{i+(1/2)} D_{i+(1/2)}]\Delta t^{n+1/2}$.

In principle, this would be a much better choice for our low order flux than the Rusanov flux we have chosen, because it would be less dissipative. However, we cannot forget that the primary property that we want of our low order flux is its guaranteed freedom from unphysical behavior. Anyone familiar with the modern literature on approximate Riemann solvers will recognize the above as one of the popular ways of constructing them. He or she will also know that such approximate solvers are in general devoid of the guarantees that we need. Thus we leave this

promising topic for future exploration, and move on to the related topic of optimizing the adaptive dissipation in the high order flux.

Recall that our order 4 dissipative flux for advection was given by:

$$F^{D4}_{i+(1/2)} = -|u|\left[\frac{3}{16}\left(q^n_{i+1} - q^n_i\right) - \frac{1}{16}\left(q^n_{i+2} - q^n_{i-1}\right)\right]\Delta t^{n+1/2} \tag{46}$$

while the corresponding flux for the Euler equations was

$$F^{D4}_{i+(1/2)} = -\frac{1}{2}(Q_i + Q_{i+1})\left[\frac{3}{16}\left(q^n_{i+1} - q^n_i\right) - \frac{1}{16}\left(q^n_{i+2} - q^n_{i-1}\right)\right]\Delta t^{n+1/2} \tag{47}$$

where Q_i is again the maximum characteristic speed at i:

$$Q_i = |u_i| + c_i. \tag{48}$$

Comparing the above pair of equations with the preceding pair, we see that the order 4 dissipative flux for the Euler equations suffers from the same flaw as does the Rusanov flux: in general it will provide more dissipation than is needed. The most extreme example of this is that of very low Mach number flow in which advected waves would be subject to a dissipation proportional to c, while the waves themselves were moving with a velocity of $v \ll c$. A way of addressing this is, again, to use the characteristic variables, dissipating each of the component waves in proportion to its own wave speed.

To give a concrete example of the procedure, let us first rewrite Eq. (47):

$$F^{D4}_{i+(1/2)} = \frac{1}{32}(Q_i + Q_{i+1})[\Delta q_{i-(1/2)} - 2\Delta q_{i+(1/2)} + \Delta q_{i+(3/2)}]\Delta t^{n+1/2}$$
$$\text{where } \Delta q_{i+(1/2)} \equiv q^n_{i+1} - q^n_i. \tag{49}$$

Our CV dissipative flux of order 4 would then be computed as follows:

1. Calculate some appropriate average $q^{n(j)}_{i+(1/2)}, \forall j, i.$
2. From $q^n_{i+(1/2)}$ calculate $T^{-1}_{i+(1/2)}$ and $T_{i+(1/2)}, \forall i.$
3. Set $\Delta_{i+(1/2)} = T^{-1}_{i+(1/2)}(q^n_{i+1} - q^n_i), \forall i.$
4. Set $D^{(j)}_{i+(1/2)} = \frac{|\lambda^{(j)}_{i+(1/2)}|}{16}(\Delta^{(j)}_{i-(1/2)} - 2\Delta^{(j)}_{i+(1/2)} + \Delta^{(j)}_{i+(3/2)}), \forall j, i.$
5. Set $F^{D4}_{i+(1/2)} = [T_{i+(1/2)} D_{i+(1/2)}]\Delta t^{n+1/2}.$

Other adaptive dissipative fluxes can be computed in a similar manner. Rerunning all of our previous CV limiter calculations with this CV adaptive dissipation, we find that only the $N = 4$ calculations display any significant differences. We show only those here. In Fig. 15 we compare two calculations, both using CV limiting, for the Sod shock tube problem. The left panel is the same as that of the left panel in Fig. 10, using the adaptive dissipation given by Eq. (47), while the right panel instead uses the CV adaptive dissipation given by the above algorithm. Note a significant increase in the sharpness of the contact discontinuity, without any other adverse effects. This is, of course, what we hoped we would achieve.

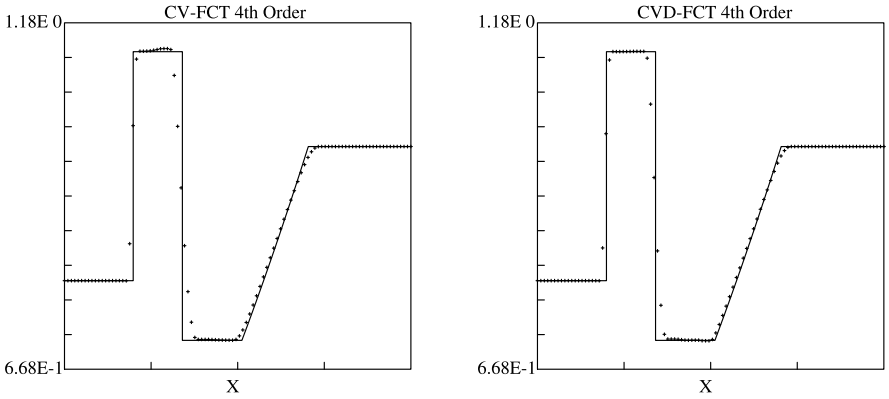

Fig. 15 Sod shock tube problem: Comparison of the $N = N_d = 4$ CV FCT algorithm using the conventional adaptive dissipation given by Eq. (47) (*left*), and the CV-based adaptive dissipation described in the text (*right*)

6 Flux-Corrected Transport in Multidimensions

As we have stated before, there is a large class of problems for which an operator splitting strategy, using sequences of one-dimensional time-advancement operators, can be successful. We are assuming here, however, that we are interested in pursuing a more fully multidimensional approach wherein the results are independent, or as independent as possible, of any apparent ordering of one-dimensional operators. Of the three components of an FCT algorithm, only the flux limiter normally presents any difficulty in this regard. Indeed, much of [13] was devoted to defining a fully multidimensional flux limiter, which we present below.

6.1 A Fully Multidimensional Flux Limiter

The alternative flux limiting algorithm presented in Sect. 4.2 generalizes trivially to any number of spatial dimensions, and in fact to unstructured as well as the structured meshes we consider here. For the sake of completeness we present the algorithm for the structured coordinate-aligned two dimensional mesh referred to in Eq. (10).

Referring to Fig. 16, we seek to limit the antidiffusive fluxes $A_{i+(1/2),j}$ and $A_{i,j+(1/2)}$ by finding coefficients $C_{i+(1/2),j}$ and $C_{i,j+(1/2)}$ such that

$$A^C_{i+(1/2),j} = C_{i+(1/2),j} A_{i+(1/2),j}, \quad 0 \le C_{i+(1/2),j} \le 1,$$
$$A^C_{i,j+(1/2)} = C_{i,j+(1/2)} A_{i,j+(1/2)}, \quad 0 \le C_{i,j+(1/2)} \le 1$$

and such that $A^C_{i+(1/2),j}$, $A^C_{i-(1/2),j}$, $A^C_{i,j+(1/2)}$, and $A^C_{i,j-(1/2)}$ acting in concert shall not cause

$$q^{n+1}_{ij} = q^{td}_{ij} - \Delta V^{-1}_{ij}\left[A^C_{i+(1/2),j} - A^C_{i-(1/2),j} + A^C_{i,j+(1/2)} - A^C_{i,j-(1/2)}\right]$$

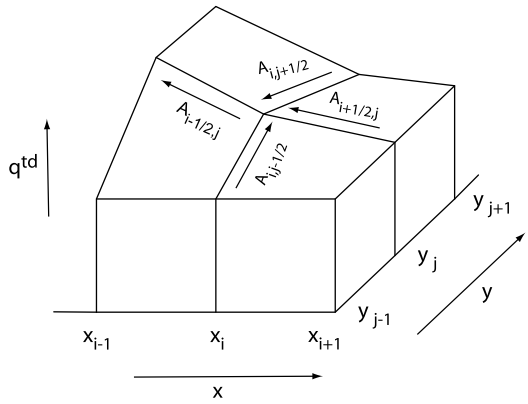

Fig. 16 Schematic of the flux limiting problem in two dimensions

to exceed some maximum value q_{ij}^{max} or fall below some minimum value q_{ij}^{min}. The procedure is completely analogous to that given in Sect. 2:

1. Compute, for each grid point ij, physically-motivated upper and lower bounds on the solution in the next timestep, q_{ij}^{max} and q_{ij}^{min} respectively.
2. For the upper bound, compute P, Q, and their ratio R at each grid point:

$$P_{ij}^{+} = \max(A_{i-(1/2),j}, 0) - \min(A_{i+(1/2),j}, 0) \tag{50}$$

$$+ \max(A_{i,j-(1/2)}, 0) - \min(A_{i,j+(1/2)}, 0), \tag{51}$$

$$Q_{ij}^{+} = \left(q_{ij}^{max} - q_{ij}^{td}\right)\Delta V_{ij}, \tag{52}$$

$$R_{ij}^{+} = \min\left(1, Q_{ij}^{+}/P_{ij}^{+}\right), \ P_{ij}^{+} > 0, \quad 0 \text{ otherwise.} \tag{53}$$

3. For the lower bound, compute P, Q, and their ratio R at each grid point:

$$P_{ij}^{-} = \max(A_{i+(1/2),j}, 0) - \min(A_{i-(1/2),j}, 0) \tag{54}$$

$$+ \max(A_{i,j+(1/2)}, 0) - \min(A_{i,j-(1/2)}, 0), \tag{55}$$

$$Q_{ij}^{-} = \left(q_{ij}^{td} - q_{ij}^{min}\right)\Delta V_{ij}, \tag{56}$$

$$R_{ij}^{-} = \min\left(1, Q_{ij}^{-}/P_{ij}^{-}\right), \ P_{ij}^{-} > 0, \quad 0 \text{ otherwise.} \tag{57}$$

4. Compute $C_{i+(1/2),j}$ and $C_{i,j+(1/2)}$ by taking a minimum:

$$C_{i+(1/2),j} = \begin{cases} \min(R_{i+1,j}^{+}, R_{ij}^{-}) & \text{when } A_{i+(1/2),j} > 0, \\ \min(R_{ij}^{+}, R_{i+1,j}^{-}) & \text{when } A_{i+(1/2),j} \le 0, \end{cases} \tag{58}$$

$$C_{i,j+(1/2)} = \begin{cases} \min(R_{i,j+1}^{+}, R_{ij}^{-}) & \text{when } A_{i,j+(1/2)} > 0, \\ \min(R_{ij}^{+}, R_{i,j+1}^{-}) & \text{when } A_{i,j+(1/2)} \le 0. \end{cases} \tag{59}$$

Again note that in the above we do not specify the equivalent of Eq. (14) in [13]. As we have stated, we do not consider that equation to be part of the flux limiter proper, but rather an algorithm for pre-constraining the high order fluxes. Nonetheless, for fully multidimensional problems we have yet to find anything better, and

we use an abbreviated form of that equation to pre-constrain the high order fluxes in the multidimensional advection problems that follow:

$$\begin{aligned} A_{i+(1/2),j} &= 0 \quad \text{if } A_{i+(1/2),j}\left(q^{td}_{i+1,j} - q^{td}_{ij}\right) \le 0, \\ A_{i,j+(1/2)} &= 0 \quad \text{if } A_{i,j+(1/2)}\left(q^{td}_{i,j+1} - q^{td}_{ij}\right) \le 0. \end{aligned} \tag{60}$$

With our fully multidimensional flux limiter in hand, along with our algorithm for pre-constraining the high order fluxes Eq. (60), let us consider two multidimensional problems: passively-driven convection in two dimensions, and compressible gas dynamics in two dimensions.

6.2 FCT Algorithms for Two-Dimensional Passively-Driven Convection

We shall be interested in solving Eq. (8) for the special case where $q(x, y)$ is a scalar and where

$$f = qu, \tag{61}$$

$$g = qv. \tag{62}$$

Here $u(x, y)$ and $v(x, y)$ are convection velocity components in the x and y directions respectively. They are assumed to be specified either globally or at the very least at cell boundaries. Thus our equation is

$$q_t + (qu)_x + (qv)_y = 0. \tag{63}$$

Our first order of business is to specify high and low order fluxes. Since $u_{i+(1/2),j}$ and $v_{i,j+(1/2)}$ are specified at cell faces, our job reduces to specifying $q_{i+(1/2),j}$ and $q_{i,j+(1/2)}$ at cell faces, and then multiplying them by the appropriate cell face velocity. The low order fluxes, the high order fluxes, and the high order dissipation components are all straightforward generalizations of the fluxes we used in one dimensional linear advection.

For our low order fluxes, we choose a two-dimensional donor cell algorithm:

$$q^L_{i+(1/2),j} = \left\{ \begin{array}{ll} q_{ij} & \text{when } u_{i+(1/2),j} > 0 \\ q_{i+1,j} & \text{when } u_{i+(1/2),j} \le 0 \end{array} \right\}, \tag{64}$$

$$q^L_{i,j+(1/2)} = \left\{ \begin{array}{ll} q_{ij} & \text{when } v_{i,j+(1/2)} > 0 \\ q_{i,j+1} & \text{when } v_{i,j+(1/2)} \le 0 \end{array} \right\}, \tag{65}$$

$$F^L_{i+(1/2),j} = q^L_{i+(1/2),j} u_{i+(1/2),j} S_{i+(1/2),j} \Delta t^{n+1/2}, \tag{66}$$

$$G^L_{i,j+(1/2)} = q^L_{i,j+(1/2)} v_{i,j+(1/2)} S_{i,j+(1/2)} \Delta t^{n+1/2} \tag{67}$$

where $S_{i+(1/2),j}$ and $S_{i,j+(1/2)}$ are the areas of the x and y cell faces respectively. We note that the above "four-flux" donor cell algorithm does not account for corner transport in a single step. While we do not describe them here, variants of the above

do allow corner transport and at the same time satisfy the prime requirement of preventing unphysical values of q. Thus these variants are the preferred low order fluxes for this problem, and are the ones we use here.

The high order fluxes are again computed using the formulae in the Appendix of [13]. As an example, the fourth order fluxes are given by:

$$q^{H4}_{i+(1/2),j} = \frac{7}{12}\left(q^a_{i+1,j} + q^a_{ij}\right) - \frac{1}{12}\left(q^a_{i+2,j} + q^a_{i-1,j}\right), \tag{68}$$

$$q^{H4}_{i,j+(1/2)} = \frac{7}{12}\left(q^a_{i,j+1} + q^a_{ij}\right) - \frac{1}{12}\left(q^a_{i,j+2} + q^a_{i,j-1}\right), \tag{69}$$

$$F^{H4}_{i+(1/2),j} = q^{H4}_{i+(1/2),j} u_{i+(1/2),j} S_{i+(1/2),j} \Delta t^{n+1/2}, \tag{70}$$

$$G^{H4}_{i,j+(1/2)} = q^{H4}_{i,j+(1/2)} v_{i,j+(1/2)} S_{i,j+(1/2)} \Delta t^{n+1/2} \tag{71}$$

where the time level t^a is meant to denote whatever time level or average of time levels is required by the particular substep of the particular time discretization chosen.

The high order dissipative fluxes of order N_D, which are added to the above high order fluxes, again follow very closely to their one-dimensional counterparts. As an example, the order 4 dissipative fluxes are given by:

$$\begin{aligned} F^{D4}_{i+(1/2),j} = &-|u_{i+(1/2),j}|\left[\frac{3}{16}\left(q^n_{i+1,j} - q^n_{ij}\right) - \frac{1}{16}\left(q^n_{i+2,j} - q^n_{i-1,j}\right)\right] \\ &\times S_{i+(1/2),j} \Delta t^{n+1/2}, \end{aligned} \tag{72}$$

$$\begin{aligned} F^{D4}_{i,j+(1/2)} = &-|v_{i,j+(1/2)}|\left[\frac{3}{16}\left(q^n_{i,j+1} - q^n_{ij}\right) - \frac{1}{16}\left(q^n_{i,j+2} - q^n_{i,j-1}\right)\right] \\ &\times S_{i,j+(1/2)} \Delta t^{n+1/2}. \end{aligned} \tag{73}$$

The pre-constraint of the high order fluxes is given by Eq. (60). Since we will be limiting directly on the variable q, there is no need for a fail-safe procedure.

For our flux limiter, we choose the multidimensional limiter given above, with q^{max}_{ij} and q^{min}_{ij} specified thus:

$$\begin{aligned} q^+_{ij} &= \max\left(q^n_{ij}, q^{td}_{ij}\right), \\ q^{max}_{ij} &= \max\left(q^+_{i-1,j}, q^+_{i,j}, q^+_{i+1,j}, q^+_{i,j-1}, q^+_{i,j+1}\right), \\ q^-_{ij} &= \min\left(q^n_{ij}, q^{td}_{ij}\right), \\ q^{min}_{ij} &= \min\left(q^-_{i-1,j}, q^-_{i,j}, q^-_{i+1,j}, q^-_{i,j-1}, q^-_{i,j+1}\right). \end{aligned} \tag{74}$$

For our test problem we choose the solid body rotation problem given in [13]. We have Eq. (63) with $u = -\Omega(y - y_0)$ and $v = \Omega(x - x_0)$, where Ω is a constant angular velocity, and (x_0, y_0) is the axis of rotation. The computational grid is 100×100 cells, $\Delta x = \Delta y$, with counterclockwise rotation taking place about grid point $(50, 50)$. Centered at grid point $(50, 75)$ is a cylinder of radius 15 grid points, through which a slot has been cut of width 5 grid points. The time step and rotation

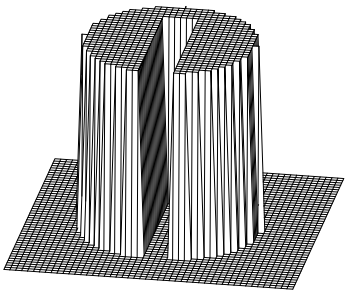

Fig. 17 Initial condition. Grid points inside the slotted cylinder have $q = 3.0$. All others have $q = 1.0$. Only the central 50×50 array of grid points around the analytic center of the distribution is shown

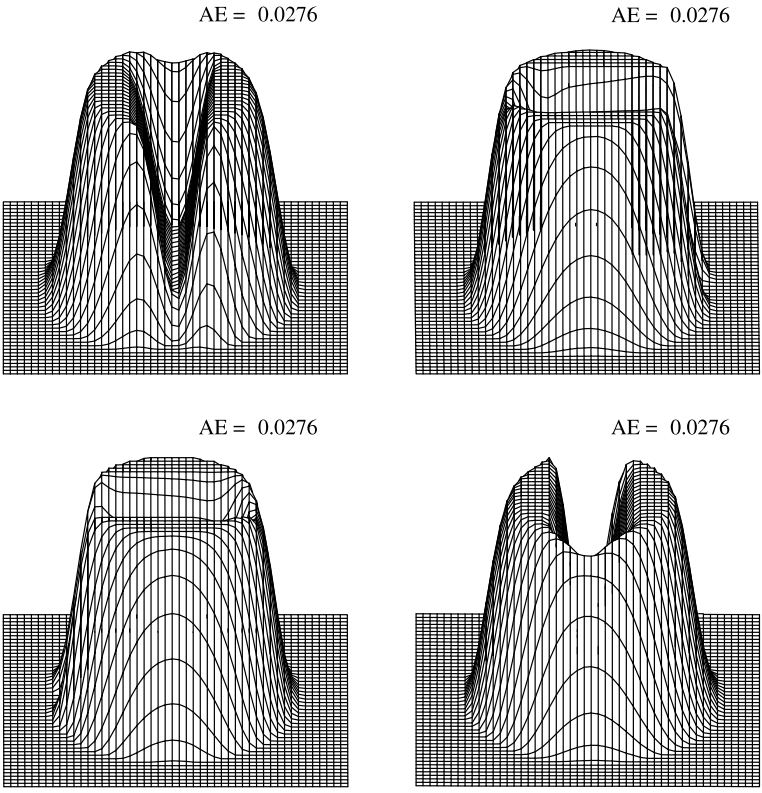

Fig. 18 Results after one revolution with $N = N_D = 4$

speed are such that 1256 time steps will effect one complete revolution of the cylinder about the central point. A perspective view of the initial conditions is shown in Fig. 17. In this and following figures, only the central 50×50 array of grid points around the analytic center of the distribution is shown.

In Fig. 18 we show the results after one revolution of the cylinder about the axis, using $N = N_D = 4$. We show the profile from four different angles, with the L_1 error denoted by "AE." Overall, the FCT algorithm has performed well. Nowhere on the

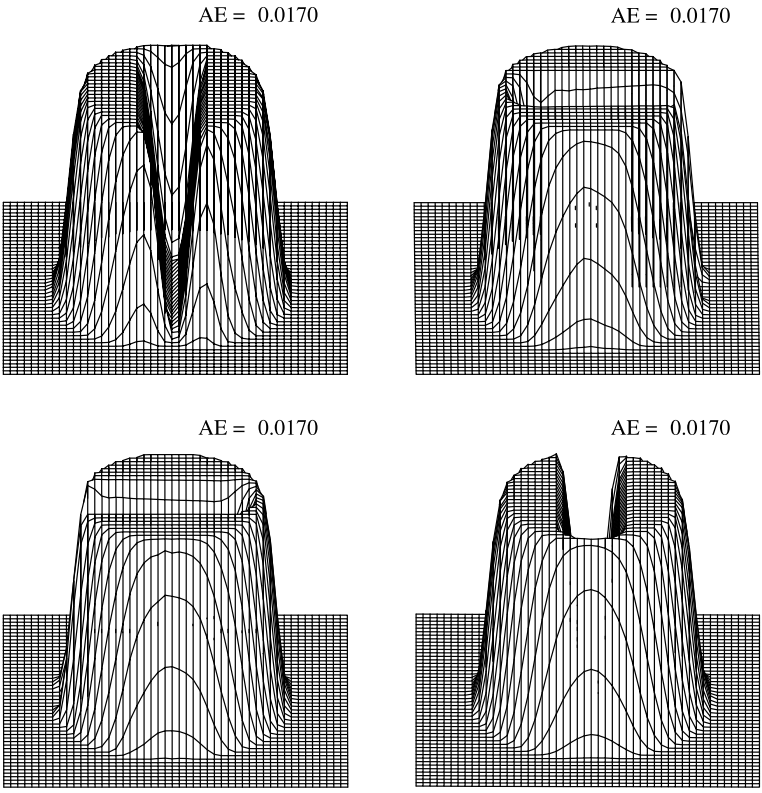

Fig. 19 Results after one revolution with $N = N_D = 8$

grid are there values of q outside the bounds of the analytic result, bounds the high order algorithm would have violated in the very first timestep. Yet the numerical diffusion, although certainly present, is far less than that which would have been generated by the low order algorithm. The top of the cylinder has remained flat and free of oscillations, and kept its original value of 3.0. The flat area outside the cylinder has also remained flat and free of oscillations, and kept its original value of 1.0. The L_1 error is 0.0276. The profile is a bit more diffuse than the fourth order calculation shown in [13]. This can be attributed to the fact that we include a fourth order dissipation term in the high order flux in the present calculation, and did not do so in [13].

In Sect. 4, we found that by increasing the resolving power of the high order fluxes, we could improve the performance of the corresponding FCT algorithm for one-dimensional advection. Let us see if that pattern plays out in multidimensional advection as well. In Fig. 19 we show the results after one revolution of the cylinder about the axis, using $N = N_D = 8$. The L_1 error is 0.0170. The results are clearly quite a bit better than the $N = N_D = 4$ calculation. There is far less erosion in the slot, and the bridge connecting the two halves of the cylinder has maintained its integrity. In Fig. 20 we show the results after one revolution of the cylinder about

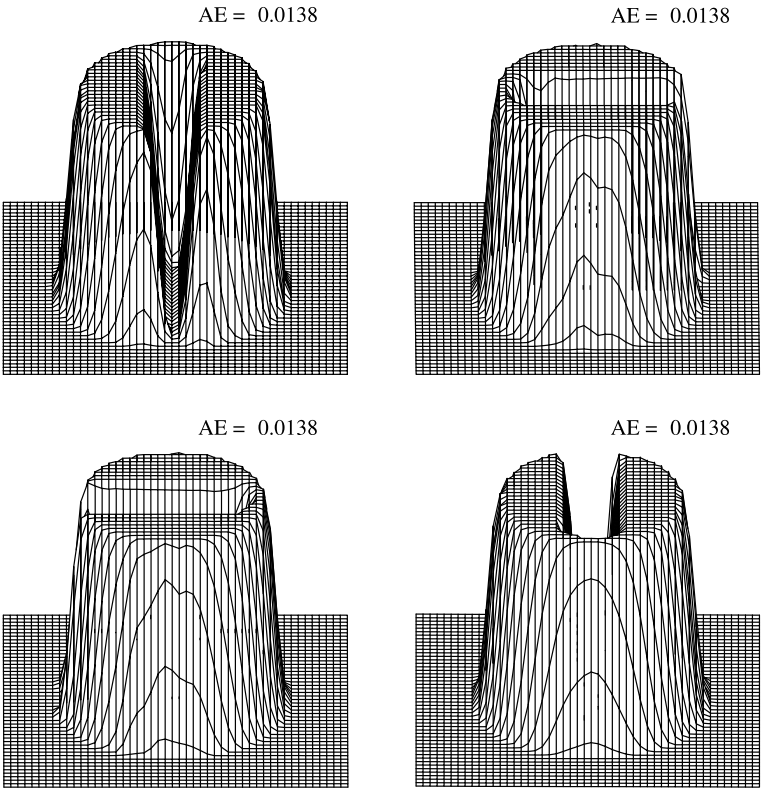

Fig. 20 Results after one revolution with $N = N_D = 16$

the axis, using $N = N_D = 16$. The L_1 error is 0.0138. Again we see a marked improvement with increasing resolving power in the high order flux.

A careful look at Fig. 20 will reveal an aspect of the multidimensional limiter used with the bounds given in Eq. (74) that has been noted both in [13] and more recently by DeVore [6]: Even though this combination of limiter and upper and lower bounds does prevent the occurrence of maxima and minima beyond those bounds, this property is not synonymous with the enforcement of monotonicity in any given coordinate direction. Note in particular the breaking of one-dimensional monotonicity along the front upper portion of the cylinder in the lower left panel of Fig. 20. Such breaking of monotonicity is often, but not always, caused by the development of dispersive ripples due to high order fluxes in one direction which are not seen as errors by the multidimensional limiter due to the presence of a steep gradient in a transverse direction. To address this issue, both [13] and [6] recommended adding a "pre-limiting" step before the multidimensional flux limiter, consisting of a call to the Boris-Book flux limiter for each of the one-dimensional fluxes. That is, prior to the multidimensional flux limiter, $A_{i+(1/2),j}$ is limited with respect to q^{td} in the x-direction, and $A_{i,j+(1/2)}$ is limited with respect to q^{td} in the y-direction, using the Boris-Book limiter. In Fig. 21 we show the results of applying that technique

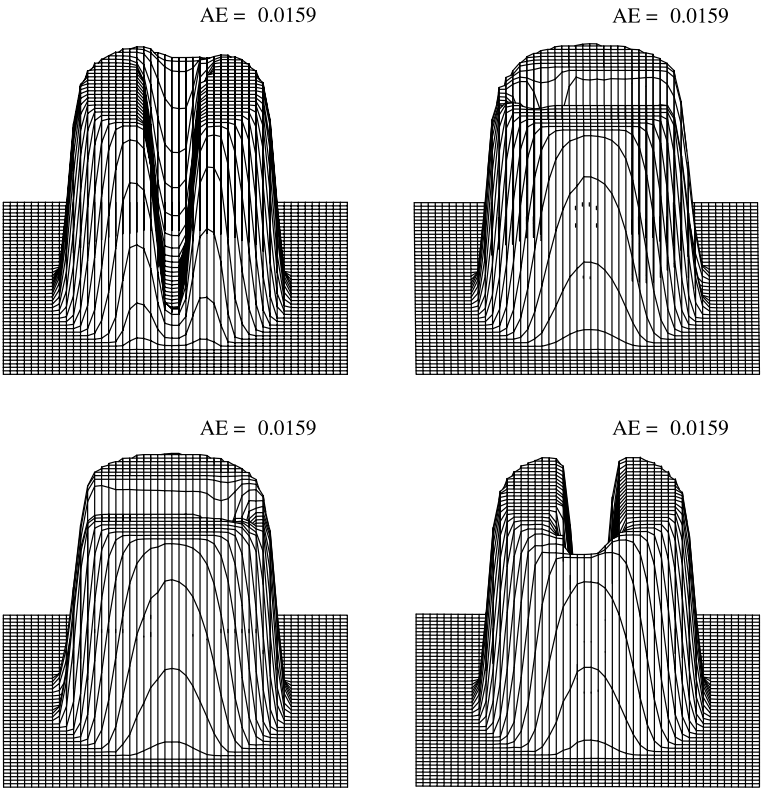

Fig. 21 $N = N_D = 16$ with the Boris-Book pre-limiter

to the $N = N_D = 16$ calculation shown previously in Fig. 20. Although many of the regions of broken monotonicity have been eliminated, the overall solution has been degraded. Significant erosion of the slot and the bridge have taken place, and we now have an L_1 error of 0.0159. This degradation is due primarily to the fact that peaked profiles naturally occur along the outer portions of the cylinder, both initially and as the profile moves and diffuses slightly. These peaked profiles are subsequently "clipped" by the Boris-Book limiter, giving us worse results than if we had not invoked the pre-limiter at all, at least for this problem.

A solution to the above dilemma is to pre-limit using a one-dimensional limiter as above, but to do so using a limiter which does not clip extrema, rather than the Boris-Book limiter. In Fig. 22 we show the results of using a slightly modified version of the non-clipping one-dimensional flux limiter described in Sect. 4.3 to "pre-limit" the $N = N_D = 16$ calculation shown previously in Fig. 20. We see that not only have many of the regions of broken monotonicity vanished, but the overall solution has improved, with an L_1 error of 0.0137. Thus if pre-limiting is deemed advisable, our recommendation is to use non-clipping limiters rather than the Boris-Book limiter to accomplish that task.

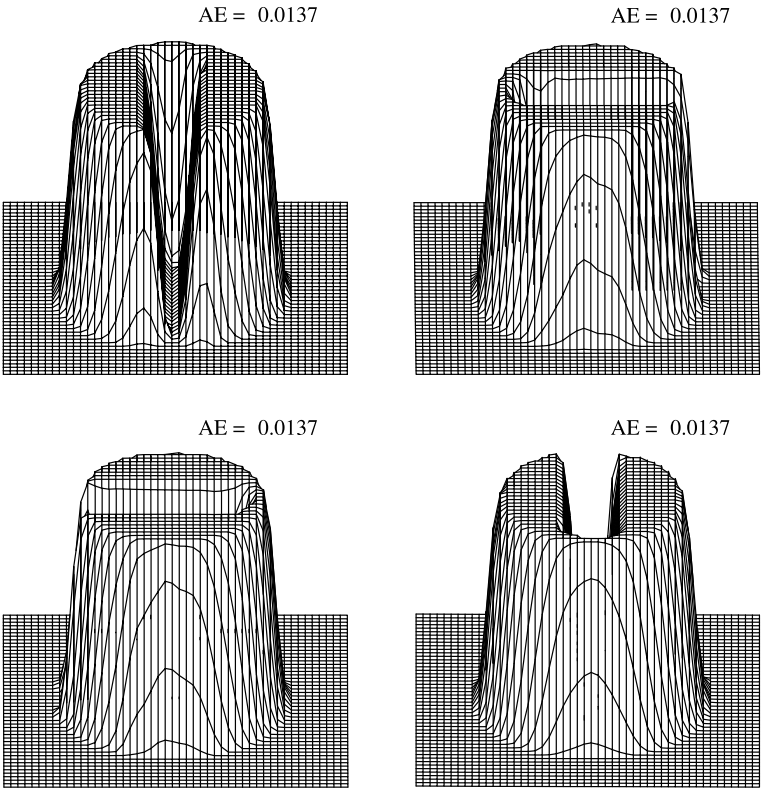

Fig. 22 $N = N_D = 16$ with a non-clipping pre-limiter

6.3 FCT Algorithms for Two-Dimensional Compressible Gas Dynamics

We are interested in solving the equations of two-dimensional compressible inviscid fluid flow Eq. (9). Recall that when we studied the corresponding one-dimensional system, we found a distinct advantage to limiting with respect to the characteristic variables rather than the conserved variables. We also found that we could use the Boris-Book limiter with fairly good success, indicating that clipping was not a serious problem, at least when one uses characteristic variables. Here we will try to build on that success.

We immediately face an apparent problem, however. The characteristic variables are only rigorously defined for one spatial dimension, i.e., it is not possible to simultaneously diagonalize both f and g with the same similarity transformation (for gas dynamics). It is clear, then, that if we wish to limit with respect to characteristic variables, we can only perform flux limiting in one direction at a time. We shall use the characteristic form of the Boris-Book limiter that we developed in Sect. 5.2 for

this task, using it in such a manner as to preserve as much full multidimensionality as possible in the algorithm.

The flux limiter we shall use is exactly as described in Sect. 5.2, except that we shall require similarity transformations appropriate for the full set of four conserved variables. The ones we actually use here are those appropriate for three-dimensional gas dynamics, with five conserved variables, with the third component of momentum set to zero. The matrices T and T^{-1} in the x direction are given by

$$T = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ u-c & 0 & 0 & u & u+c \\ v & 1 & 0 & v & v \\ w & 0 & 1 & w & w \\ H-uc & v & w & \frac{1}{2}q^2 & H+uc \end{bmatrix}, \tag{75}$$

$$T^{-1} = \begin{bmatrix} \frac{1}{2}(\frac{\gamma-1}{2}M^2+\frac{u}{c}) & -\frac{1}{2c}-\frac{(\gamma-1)u}{2c^2} & -\frac{(\gamma-1)v}{2c^2} & -\frac{(\gamma-1)w}{2c^2} & \frac{\gamma-1}{2c^2} \\ -v & 0 & 1 & 0 & 0 \\ -w & 0 & 0 & 1 & 0 \\ 1-\frac{\gamma-1}{2}M^2 & \frac{(\gamma-1)u}{c^2} & \frac{(\gamma-1)v}{c^2} & \frac{(\gamma-1)w}{c^2} & -\frac{\gamma-1}{c^2} \\ \frac{1}{2}(\frac{\gamma-1}{2}M^2-\frac{u}{c}) & \frac{1}{2c}-\frac{(\gamma-1)u}{2c^2} & -\frac{(\gamma-1)v}{2c^2} & -\frac{(\gamma-1)w}{2c^2} & \frac{\gamma-1}{2c^2} \end{bmatrix} \tag{76}$$

where $q^2 \equiv u^2+v^2+w^2$, $M^2 \equiv q^2/c^2$, H is the stagnation enthalpy, and u, v, and w are the x, y, and z components of velocity respectively. For the y direction, a corresponding set of transformation matrices is used.

The easiest solution is, of course, to simply use directional operator splitting. To demonstrate that such a technique is viable, in Fig. 23 we show a calculation using a directionally split version of the $N=8$, $N_D=8$ CV FCT algorithm given here to solve the Mach reflection problem given by Woodward and Colella [12]. The problem consists of a Mach 10 shock reflecting from a 30 degree wedge. The three resolutions used in [12] are shown, corresponding to meshes of 120×30, 240×60, and 480×120 from top to bottom. We invite the reader to compare the results to those obtained elsewhere. In Fig. 24 we show the same calculation, but using the conventional non-CV limiter which limits only on the conserved variables. The morphology of the jet along the bottom wall disagrees both with experimental data and with other published numerical calculations. We conclude that the CV limiter used in Fig. 23 is by far the better choice. We also conclude that, for this particular test problem, directional splitting is satisfactory.

Of course one would prefer not to use directional splitting, since one cannot be sure in advance that the particular physics problem of interest will yield satisfactory results when such splitting is employed. Thus we would prefer not to use full-blown directional splitting, and yet the variables which we desire to use for flux limiting would seem to require that the limiting step itself be directionally split. Is there a way to satisfy both requirements? Is there some way to be "fully multidimensional" and also use characteristic variables?

One way to define the term "fully multidimensional algorithm" is to demand that the results be independent of any choice of ordering that may be present in an algorithm. Another, perhaps just as satisfactory, is to demand that same independence,

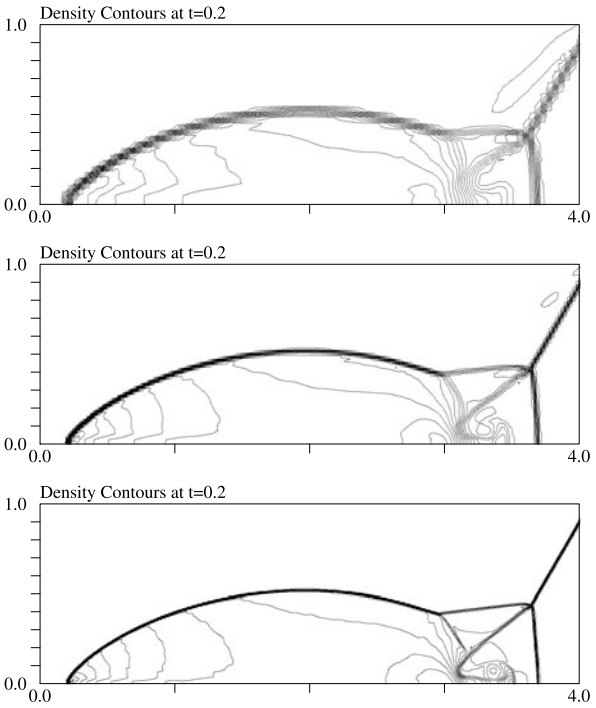

Fig. 23 Isodensity contours for the Woodward-Colella ramp problem at $t = 0.2$ using a CV limiter with $N = N_D = 8$, and directional splitting. From top to bottom, the displayed grids are 120×30, 240×60, and 480×120

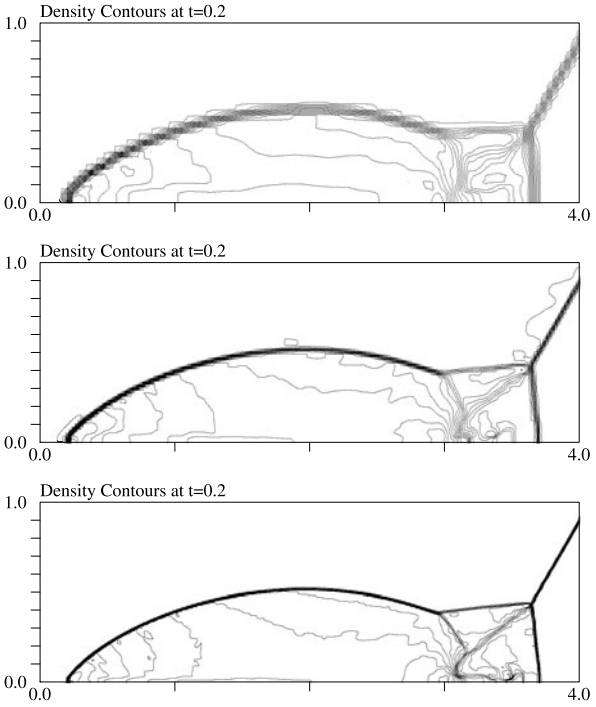

Fig. 24 Same as Fig. 23, but using the conventional flux limiter which limits only on the conserved variables. The morphology of the jet along the bottom wall disagrees both with experimental data and with other published numerical calculations. We conclude that the CV limiter used in Fig. 23 is by far the better choice

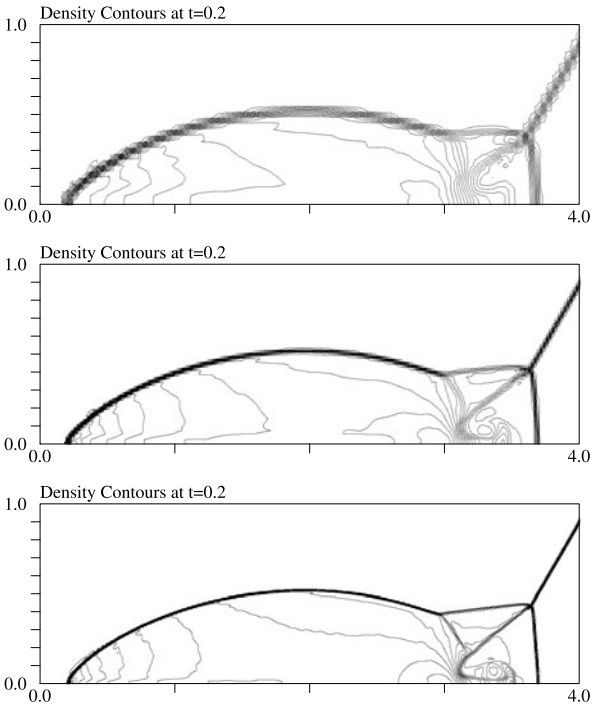

Fig. 25 Same as Fig. 23, but using a CV limiter independently on fully multidimensional antidiffusive fluxes

except for the flux limiting step itself. We use one variant of each here, which we describe in compact form:

1. Compute all high and low order fluxes fully multidimensionally.
2. Either

   - limit the x-, y-, and z-directed fluxes independently; or
   - limit the x-, y-, and z-directed fluxes sequentially, updating solution values between steps.

The first choice increases the risk that the failsafe limiter will be brought into play, but is truly multidimensional. The second is less likely to generate the need for the failsafe limiter, but is multidimensional only in the second sense above. In Fig. 25, we show the results of the Woodward-Colella Mach reflection problem using a CV limiter and limiting the x- and y-directed fluxes independently. In Fig. 26, we show the results using a CV limiter and limiting the x- and y-directed fluxes sequentially. Both are seen to perform quite well, albeit with results that are virtually indistinguishable from those in Fig. 23. Clearly this test problem, although it is the standard test problem for multidimensional compressible flow, is not one for which one needs a fully multidimensional algorithm to achieve accurate solutions. Nonetheless, we would recommend either of the two multidimensional approaches over the fully split one, as a means of avoiding the splitting errors that may occur when simulating more general flows.

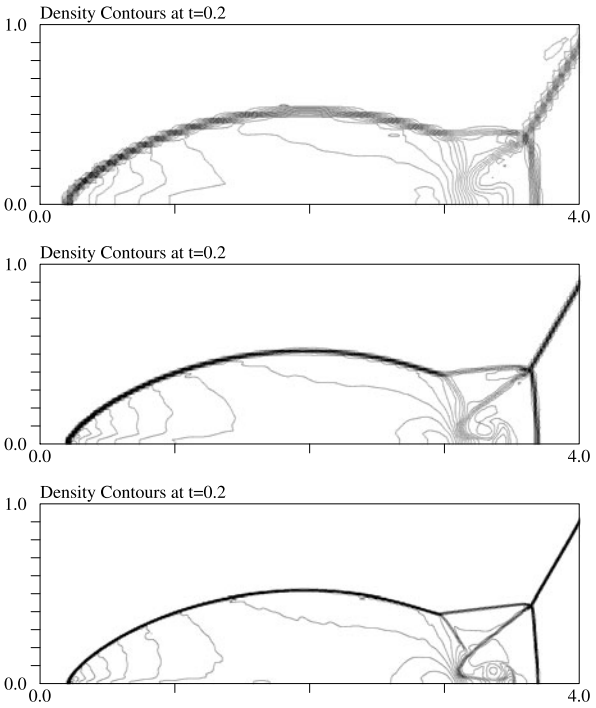

Fig. 26 Same as Fig. 23, but using a CV limiter sequentially on fully multidimensional antidiffusive fluxes

7 Conclusions

We have tried to give the reader a distillation of the design principles for building FCT algorithms, and front-capturing algorithms in general, that we have gleaned from experience over the past several decades. If there is a common thread to all of them it is this: the scientists who use such algorithms must have both input to and knowledge of their design. There may come a day when we no longer hold to this view, when the design of such algorithms can be left to expert numerical analysts alone, but that day has not yet arrived.

References

1. Boris, J.P.: A fluid transport algorithm that works. In: Computing as a Language of Physics, International Atomic Energy Agency, pp. 171–189 (1971)
2. Boris, J.P., Book, D.L.: Flux-Corrected Transport I: SHASTA, a fluid-transport algorithm that works. J. Comput. Phys. **11**, 38–69 (1973)
3. Chorin, A.J.: Random choice solution of hyperbolic systems. J. Comput. Phys. **22**, 517–536 (1976)
4. Chorin, A.J.: Random choice methods with application to reacting gas flow. J. Comput. Phys. **25**, 252–272 (1977)
5. Colella, P., Woodward, P.R.: The Piecewise-Parabolic Method (PPM) for gas-dynamical simulations. J. Comput. Phys. **54**, 174–201 (1984)

6. DeVore, C.R.: An improved limiter for multidimensional flux-corrected transport. NRL Memorandum Report 6440-98-8330, Naval Research Laboratory, Washington, DC (1998)
7. Forester, C.K.: Higher order monotonic convective difference schemes. J. Comput. Phys. **23**, 1–22 (1977)
8. Glimm, J.: Solution in the large for nonlinear hyperbolic systems of equations. Commun. Pure Appl. Math. **18**, 697–715 (1955)
9. Kreiss, H.-O., Oliger, J.: Comparison of accurate methods for the integration of hyperbolic equations. Tellus **24**, 199 (1972)
10. McDonald, B.E.: Flux-corrected pseudospectral method for scalar hyperbolic conservation laws. J. Comput. Phys. **82**, 413 (1989)
11. Sod, G.A.: A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. J. Comput. Phys. **27**, 1–31 (1978)
12. Woodward, P.R., Colella, P.: The numerical simulation of two-dimensional flow with strong shocks. J. Comput. Phys. **54**, 115–173 (1984)
13. Zalesak, S.T.: Fully multidimensional Flux-Corrected Transport algorithms for fluids. J. Comput. Phys. **31**, 335–362 (1979)
14. Zalesak, S.T.: Very high order and pseudospectral Flux-Corrected Transport (FCT) algorithms for conservation laws. In: Vichnevetsky, R., Stepleman, R.S. (eds.) Advances in Computer Methods for Partial Differential Equations IV, IMACS, Rutgers University, pp. 126–134 (1981)
15. Zalesak, S.T.: A preliminary comparison of modern shock-capturing schemes: Linear advection. In: Vichnevetsky, R., Stepleman, R.S. (eds.) Advances in Computer Methods for Partial Differential Equations VI, IMACS, Rutgers University, pp. 15–22 (1987)