

*Communications in  
Applied  
Mathematics and  
Computational  
Science*

**A SINGLE-STAGE FLUX-CORRECTED  
TRANSPORT ALGORITHM FOR HIGH-ORDER  
FINITE-VOLUME METHODS**

CHRISTOPHER CHAPLIN AND PHILLIP COLELLA

vol. 12 no. 1 2017



# A SINGLE-STAGE FLUX-CORRECTED TRANSPORT ALGORITHM FOR HIGH-ORDER FINITE-VOLUME METHODS

CHRISTOPHER CHAPLIN AND PHILLIP COLELLA

We present a new limiter method for solving the advection equation using a high-order, finite-volume discretization. The limiter is based on the flux-corrected transport algorithm. We modify the classical algorithm by introducing a new computation for solution bounds at smooth extrema, as well as improving the preconstraint on the high-order fluxes. We compute the high-order fluxes via a method-of-lines approach with fourth-order Runge–Kutta as the time integrator. For computing low-order fluxes, we select the corner-transport upwind method due to its improved stability over donor-cell upwind. Several spatial differencing schemes are investigated for the high-order flux computation, including centered-difference and upwind schemes. We show that the upwind schemes perform well on account of the dissipation of high-wavenumber components. The new limiter method retains high-order accuracy for smooth solutions and accurately captures fronts in discontinuous solutions. Further, we need only apply the limiter once per complete time step.

## 1. Introduction

We wish to solve hyperbolic conservation laws of the form

$$\frac{\partial U}{\partial t} + \nabla \cdot (\vec{F}(U)) = 0, \quad (1)$$

where  $U$  represents a vector of conserved values and  $\vec{F} = (F^1 \dots F^D)$  the corresponding  $D$ -dimensional fluxes. The discrete solution of these equations at a given time  $t^{n+1}$  and spatial location  $i$  is given by

$$\langle U \rangle_i^{n+1} = \langle U \rangle_i^n - \frac{\Delta t}{h} \sum_{d=1}^D [(F^d)_{i+e^d/2}^{n+1/2} - (F^d)_{i-e^d/2}^{n+1/2}], \quad (2)$$

where  $\langle U \rangle_i^n$  approximates the average of  $U$  over a rectangular Cartesian control volume at time  $t^n$  and  $(F^d)^{n+1/2}$  approximates the average of  $\vec{F}(U)$  from time  $t^n$  to  $t^{n+1}$  over the faces of the same control volume. The parameters  $\Delta t$  and  $h$  represent

---

*MSC2010:* 65M08.

*Keywords:* finite-volume method, high order, advection, limiter.

the time-step size and grid spacing, respectively. Methods for accurately computing the fluxes,  $(\mathbf{F}^d)^{n+1/2}$ , to obtain high-order accuracy for smooth solutions are well understood. However, these high-order methods must be modified to selectively introduce dissipation in the presence of discontinuities or underresolved gradients. These modification methods are called limiter schemes. Modern limiter schemes seek to achieve high-order accuracy for smooth solutions regardless of complexity and to represent discontinuities well.

Many of the original second-order limiter schemes, such as monotonic upstream-centered schemes for conservation laws (MUSCL) [25], total variation diminishing (TVD) [11], piecewise parabolic method (PPM) [7], and flux-corrected transport (FCT) [2], are still used in some form today. But these original schemes struggled to achieve all of the aforementioned goals, particularly high-order accuracy for solutions that are both complicated and smooth. The standard and weighted essentially nonoscillatory (ENO/WENO) schemes were developed to address these issues for TVD [12; 17; 22]. An extremum-preserving limiter has been added to PPM [6]. For FCT, a nonclipping limiter was developed [28]. Finite-element methods, in particular discontinuous Galerkin (DG) methods, have also been gaining popularity for these problems. DG methods use a menagerie of technologies to handle discontinuities [3]. Recent efforts have been made to combine aspects of WENO limiting and the DG discretization [22]. FCT was also extended to finite-element discretizations [18; 19], and many of the recent improvements to the algorithm have been on these discretizations [16; 13; 15; 14]. Most of the limiter methods mentioned so far use a semidiscrete, method-of-lines formulation, wherein a standard ordinary differential equation (ODE) integrator is used to advance the solution after the spatial differencing scheme has been applied. There are fully discrete methods as well, such as arbitrary derivative in space and time (ADER) [24; 23]. They offer an alternative to the method-of-lines approaches [1; 9].

The starting point for our approach is a method-of-lines, finite-volume formulation. In all semidiscrete implementations mentioned above, the limiter algorithm is applied every time a high-order flux evaluation occurs. This requires the limiter to be applied several times during a single time-update procedure. For this study we chose to only apply limiting once per time update, after the total high-order flux was generated. This choice is motivated by both current and future performance considerations [10; 26]. Moving limiting to a postprocessing procedure allows for smaller stencils and less parallel communication. Limiters that are designed to preserve high-order accuracy at smooth extrema typically make use of second- or higher-order spatial derivative information to determine where the solution is smooth enough to not require limiting. Computing these derivatives at each stage in high-order ODE integration scheme requires a larger stencil than the standard high-order flux stencil, at least up until ninth order. The other half of this is that,

typically because the limiter procedure is complicated and has a wide stencil, ghost cells are synchronized at each stage computation. If one can fit all the data required to complete the entire time update in local memory, then synchronization must only occur once per time advance. These two hindrances, namely wider stencils and repeated synchronization barriers, motivated the departure from the stage-by-stage limiting approach.

We elected to use a version of flux-corrected transport (FCT) for our limiting scheme [2; 28; 15]. FCT introduces dissipation through a nonlinear hybridization of a high-order flux with a dissipative, low-order flux. To compute the high-order flux, we used a method-of-lines approach with fourth-order Runge–Kutta (RK4) as the time-integration scheme. For the spatial derivatives we looked at a family of methods based on high-order centered- and one-point-upwinded linear finite-volume interpolations. The low-order scheme was the corner-transport upwind (CTU) method [4; 21]. We elected to use CTU for two reasons: CTU permits the use of larger time steps than the standard donor-cell method, and CTU can be constructed in such a way as to preserve positivity in the solution. In addition to using these schemes for the high- and low-order fluxes, we modified the FCT algorithm in three important ways. First we included an extremum-preserving bound computation based on the approach used in [6; 20] for interpolation-based limiting. We also designed a more restrictive condition on applying the typical precondition for the high-order fluxes. Furthermore, we extended the product rule to sixth-order accuracy. All of these features were required in order to maintain high-order accuracy for complicated smooth solutions.

For this study, we restricted our attention to the scalar advection equation. This allowed us to explore design space for this novel single-stage limiter in a simple setting, but one that is still relatively unforgiving. These advective terms appear directly in real applications including transport of scalars in the atmosphere, Vlasov equations in phase space, and combustion.

**Advection equation.** We will consider the linear advection equation in the form

$$\frac{\partial q}{\partial t} + \nabla \cdot (q \vec{u}) = 0, \quad (3)$$

$$\nabla \cdot \vec{u} = 0, \quad (4)$$

on a  $D$ -dimensional square domain  $\Omega = [0, 1]^D$ . In this case  $\vec{u}$  is an advective velocity and  $q$  is a scalar field. The partial differential equation above can also be written as

$$\frac{dq}{dt} = 0, \quad \frac{d\vec{x}}{dt} = \vec{u}. \quad (5)$$

Provided that an initial condition is specified ( $q_0 = q(\vec{x}(t_0), t_0)$ ), this system of ordinary differential equations yields a unique solution for any  $q(\vec{x}(t), t)$  and  $\vec{x}(t)$ .

The solution arrived at by integrating the equations is that  $q$  is constant along characteristic curves defined by  $\vec{x}(t)$ . Even though there is a simple solution to this equation, the analysis is still quite useful since there is no diffusion or entropy condition built into the equation: any numerical errors introduced are propagated through the domain.

**Finite-volume discretization.** Our approach is to use a finite-volume method to discretize the physical domain into a union of control volumes

$$V_i = \left[ \left( i - \frac{1}{2} \right) h, \left( i + \frac{1}{2} \right) h \right], \quad \mathbf{i} \in \mathbb{Z}^D, \quad (6)$$

where  $h$  is the grid spacing and  $\mathbf{i}$  is a  $D$ -dimensional index denoting location. The origin in the physical domain occurs at the point  $\left( i - \frac{1}{2} \right) h$  when  $\mathbf{i} = \mathbf{0}$ .

Values of the conserved scalar quantity  $q$  are stored as cell averages  $\langle q \rangle$  over each  $V_i$ , and the fluxes  $\mathbf{F}^d = q\mathbf{u}^d$  are stored as averages  $\langle \mathbf{F}^d \rangle_{i \pm e^d/2}$  over the surface faces  $A_d^\pm$  of each cell:

$$\langle q \rangle_i(t) = \frac{1}{h^D} \int_{V_i} q(\mathbf{x}, t) d\mathbf{x}, \quad (7)$$

$$\langle \mathbf{F}^d \rangle_{i \pm e^d/2}(t) = \frac{1}{h^{D-1}} \int_{A_d^\pm} \mathbf{F}^d(\mathbf{x}, t) d\mathbf{x}. \quad (8)$$

Applying the finite-volume discretization (6) to (3) yields a semidiscrete system of ordinary differential equations (ODEs) in time

$$\frac{d\langle q \rangle_i}{dt} = -\frac{1}{h^D} \int_{V_i} (\nabla \cdot (\vec{\mathbf{F}})) d\mathbf{x}. \quad (9)$$

The divergence theorem is then applied to (9):

$$\frac{d\langle q \rangle_i}{dt} = -D \cdot \langle \vec{\mathbf{F}} \rangle(t), \quad (10)$$

$$= -\frac{1}{h} \sum_{d=1}^D [\langle \mathbf{F}^d \rangle_{i+e^d/2} - \langle \mathbf{F}^d \rangle_{i-e^d/2}]. \quad (11)$$

The integration of the above system with respect to time from  $t^n$  to  $t^{n+1}$  produces the solution

$$\langle q \rangle_i^{n+1} = \langle q \rangle_i^n - \frac{\Delta t}{h} \sum_{d=1}^D [\langle \mathbf{F}^d \rangle_{i+e^d/2}^{n+1/2} - \langle \mathbf{F}^d \rangle_{i-e^d/2}^{n+1/2}], \quad (12)$$

$$\langle \mathbf{F}^d \rangle_{i \pm e^d/2}^{n+1/2} = \frac{1}{\Delta t} \int_{t^n}^{t^n + \Delta t} \langle \mathbf{F}^d \rangle_{i \pm e^d/2}(t) dt. \quad (13)$$

The resulting challenge is to accurately compute  $\langle \mathbf{F}^d \rangle_{i \pm e^d/2}^{n+1/2}$ . It is important to note that no approximations have been made at this point: (12) and (13) are exact

relationships. However, to obtain a full discrete approximation, we need quadrature rules for the surface fluxes in (11) and for the time-averaged fluxes in (13). The quadrature rules for computing these fluxes are defined following ideas from [20]. In that work, the high-order quadratures were computed using a method-of-lines approach. The surface fluxes were computed using a high-order centered-difference method, and the temporal integration was computed using the classic RK4 method. We retained the use of RK4 in this study and investigated several high-order methods for computing the surface fluxes.

**Hybridization.** Returning to the flux description in (13), we may now define the hybridization

$$\langle \mathbf{F}^d \rangle_{i+e^d/2}^{n+1/2} = (\eta_{i+e^d/2}) \langle \mathbf{F}_H^d \rangle_{i+e^d/2} + (1 - \eta_{i+e^d/2}) \langle \mathbf{F}_L^d \rangle_{i+e^d/2}, \quad (14)$$

where the subscripts  $H$  and  $L$  refer to the high-order and low-order fluxes and  $\eta_{i+e^d/2}$  is the hybridization coefficient.

In the following sections of the paper we will describe the design choices and procedures for computing the high-order flux, the low-order flux, and the hybridization coefficient.

## 2. High-order flux computation

We compute the high-order fluxes using the method of lines. Two schemes must be chosen: a scheme for integrating the solution in time and a scheme for computing the spatial derivatives. High-order accuracy requires that both schemes be high-order accurate.

**High-order temporal integration scheme.** We use the RK4 scheme to advance the solution. Returning to the system of ODEs (10),

$$\frac{d\langle q \rangle}{dt} = -D \cdot \langle \vec{\mathbf{F}} \rangle(t),$$

we want to integrate  $\langle q \rangle$  from  $t^n$  to  $t^{n+1}$ . RK4 is a fourth-order integration scheme that consists of computing a linear combination of stage-update variables  $k_s$ . The updates are defined as

$$\langle q \rangle^0 = \langle q \rangle(t^n), \quad k_1 = -D \cdot \langle \vec{\mathbf{F}}(\langle q \rangle^0) \rangle \Delta t, \quad (15)$$

$$\langle q \rangle^1 = \langle q \rangle^0 + \frac{1}{2}k_1, \quad k_2 = -D \cdot \langle \vec{\mathbf{F}}(\langle q \rangle^1) \rangle \Delta t, \quad (16)$$

$$\langle q \rangle^2 = \langle q \rangle^0 + \frac{1}{2}k_2, \quad k_3 = -D \cdot \langle \vec{\mathbf{F}}(\langle q \rangle^2) \rangle \Delta t, \quad (17)$$

$$\langle q \rangle^3 = \langle q \rangle^0 + k_3, \quad k_4 = -D \cdot \langle \vec{\mathbf{F}}(\langle q \rangle^3) \rangle \Delta t. \quad (18)$$

Each update variable  $k_s$  requires computing stage fluxes  $\langle \mathbf{F}^d \rangle_{i \pm e^d/2}^s = \langle q \mathbf{u}^d \rangle_{i \pm e^d/2}^s$ . The stage fluxes are functions of the stage values  $\langle q \rangle_i^s$  and  $\langle \mathbf{u}^d \rangle_i^s$  exclusively, and the procedure for computing the fluxes will be described in the next section.

To perform the RK4 integration, we compute the appropriate linear combination of stage updates

$$\langle q \rangle(t^n + \Delta t) = \langle q \rangle(t^n) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) + \mathcal{O}(h^5). \quad (19)$$

Using the conservation notation, this RK4 integration can also be described by

$$\langle q \rangle_i^{n+1} = \langle q \rangle_i^n - \frac{\Delta t}{h} \sum_{d=1}^D [\langle \mathbf{F}_H^d \rangle_{i+e^d/2} - \langle \mathbf{F}_H^d \rangle_{i-e^d/2}], \quad (20)$$

$$\langle \mathbf{F}_H^d \rangle_{i \pm e^d/2} = \frac{1}{6} [\langle \mathbf{F}^d \rangle_{i \pm e^d/2}^{(0)} + 2\langle \mathbf{F}^d \rangle_{i \pm e^d/2}^{(1)} + 2\langle \mathbf{F}^d \rangle_{i \pm e^d/2}^{(2)} + \langle \mathbf{F}^d \rangle_{i \pm e^d/2}^{(3)}]. \quad (21)$$

**High-order spatial difference schemes.** We use high-order finite-difference methods to approximate the surface fluxes associated with the spatial derivatives. The fluxes  $\langle q \mathbf{u}^d \rangle_{i \pm e^d/2}$  for the spatial derivatives are functions only of the cell-averaged  $\langle q \rangle_i$  and  $\langle \mathbf{u}^d \rangle_i$  at any time. Several methods were explored in this study for computing  $\langle q \rangle_{i \pm e^d/2}$ , including high-order centered-difference schemes and upwind schemes. The advantage of the upwind methods is that they have greater diffusion especially in regimes where the phase error begins to rise. The upwind methods only require a small additional computation, and the stability of similar-order centered-difference and upwind methods is almost identical. Although not investigated in this study, high-order centered-difference schemes with hyperdiffusive fluxes [28] offer a possible alternative to the upwind ones used here.

The interpolation formulae corresponding to the spatial differencing schemes used are presented below. For compactness, the following notation will be used:

$$\langle q \rangle_{i+e^d/2}^n = \sum_{s=-S}^S a_s \langle q \rangle_{i+s e^d}^n, \quad (22)$$

where  $S$  is the width of the stencil and  $a_s$  are the coefficients. The odd-ordered methods use the full range of coefficients, whereas the even-ordered methods have no coefficient at  $s = -S$ .

- Fourth-order centered difference ( $S = 2$ ):

$$\{a_s : s = -S + 1, \dots, S\} = \frac{1}{12} \{-1, 7, 7, -1\}. \quad (23)$$

- Fifth-order upwind ( $S = 2$ ):

$$\{a_s : s = -S, \dots, S\} = \frac{1}{60} \{2, -13, 47, 27, -3\}. \quad (24)$$

- Sixth-order centered difference ( $S = 3$ ):

$$\{a_s : s = -S + 1, \dots, S\} = \frac{1}{60}\{1, -8, 37, 37, -8, 1\}. \quad (25)$$

- Seventh-order upwind ( $S = 3$ ):

$$\{a_s : s = -S, \dots, S\} = \frac{1}{420}\{-3, 25, -101, 319, 214, -38, 4\}. \quad (26)$$

- Ninth-order upwind ( $S = 4$ ):

$$\{a_s : s = -S, \dots, S\} = \frac{1}{2520}\{4, -41, 199, -641, 1879, 1375, -305, 55, -5\}. \quad (27)$$

**Product rule.** To complete the flux computation, we must compute the average of the product of the scalar variable and the velocity ( $\langle q\mathbf{u}^d \rangle_{i+e^d/2}$ ). The 2D product rules for second-, fourth-, and sixth-order accuracy are

$$\langle q\mathbf{u}^d \rangle_{i+e^d/2} = \langle q \rangle_{i+e^d/2} \langle \mathbf{u}^d \rangle_{i+e^d/2} + \mathcal{O}(h^2), \quad (28)$$

$$\langle q\mathbf{u}^d \rangle_{i+e^d/2} = \langle q \rangle_{i+e^d/2} \langle \mathbf{u}^d \rangle_{i+e^d/2} + \frac{1}{12}h^2 \sum_{d' \neq d} \frac{\partial q}{\partial x_{d'}} \frac{\partial \mathbf{u}^d}{\partial x_{d'}} + \mathcal{O}(h^4), \quad (29)$$

$$\begin{aligned} \langle q\mathbf{u}^d \rangle_{i+e^d/2} &= \langle q \rangle_{i+e^d/2} \langle \mathbf{u}^d \rangle_{i+e^d/2} + \frac{1}{12}h^2 \sum_{d' \neq d} \left( \frac{\partial q}{\partial x_{d'}} \frac{\partial \mathbf{u}^d}{\partial x_{d'}} \right) \\ &+ \frac{1}{1440}h^4 \sum_{d' \neq d} \left( 3 \frac{\partial^3 q}{\partial x_{d'}^3} \frac{\partial \mathbf{u}^d}{\partial x_{d'}} + 3 \frac{\partial^3 \mathbf{u}^d}{\partial x_{d'}^3} \frac{\partial q}{\partial x_{d'}} + 2 \frac{\partial^2 \mathbf{u}^d}{\partial x_{d'}^2} \frac{\partial^2 q}{\partial x_{d'}^2} \right) + \mathcal{O}(h^6). \quad (30) \end{aligned}$$

The possible sources of error in the product formulae above are computing the averages  $\langle q \rangle_{i+e^d/2}$  and  $\langle \mathbf{u}^d \rangle_{i+e^d/2}$  and computing the partial-derivative sums. We have already discussed several methods and their accuracy for computing  $\langle q \rangle_{i+e^d/2}$ . The velocity fields are analytic for advection, so  $\langle \mathbf{u}^d \rangle_{i+e^d/2}$  introduces no error. The derivative terms in the summations above were computed exclusively using centered-difference approximations of appropriate accuracy. For example, the derivatives in the fourth-order-accurate product formula were computed using a second-order centered difference. The derivatives in the sixth-order formula were computed to fourth order (for the term multiplied by  $h^2$ ) and to second order (for the term multiplied by  $h^4$ ).

We note that the product rule has no contribution in a 1D problem or in any multidimensional problem with constant velocity. To obtain an arbitrary  $\mathcal{O}(h^N)$ -accurate solution for a multidimensional problem with varying velocity, we need to ensure that the product rule along with the time integrator and the spatial differencing scheme are all at least  $\mathcal{O}(h^N)$ . In this study, our overall solution accuracy was constrained by the use of RK4 for integration. However, lower spatial discretization errors are produced using the sixth-order product rule, in place of the fourth-order rule, with fifth- and higher-order-accurate spatial differencing schemes.

Method	Stability constraint
4th center	$\sigma \lesssim 2.06/D$
5th upwind	$\sigma \lesssim 1.73/D$
6th center	$\sigma \lesssim 1.78/D$
7th upwind	$\sigma \lesssim 1.69/D$
9th upwind	$\sigma \lesssim 1.60/D$

**Table 1.** Stability of methods for varying spatial difference operators and dimensionality  $D$ .

**Stability.** We compute the stability for each high-order scheme to determine the allowable time-step size following the procedure in [5]. Stability for the method of lines requires the eigenvalues of the right-hand side to lie within the stability region of the time integrator. These eigenvalues are computed by diagonalizing the semidiscrete system (10). For advection the eigenvalues are defined as the product of the velocity and the spatial derivative operator:

$$\frac{d\langle q \rangle}{dt} = \lambda \langle q \rangle, \quad (31)$$

$$\lambda \langle q \rangle = -\vec{u} \frac{\partial}{\partial \mathbf{x}} \langle q \rangle. \quad (32)$$

The particular eigenvalues for each spatial differencing scheme will be presented later.

These eigenvalues must lie within the stability region of the time integrator. The stability region for RK4 is well known and can be described by its characteristic polynomial

$$P(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4, \quad (33)$$

where  $z = \Delta t \lambda$ . Stability for this problem requires that  $|P(z)| \leq 1$ . The resulting stability constraints for each spatial differencing scheme are presented in Table 1, where  $\sigma = |u|\Delta t/h$ .

Along with stability, the phase error and dissipation were computed (Figure 1). The dissipation was defined as  $(1 - |g|)$ , where

$$|g| = \sqrt{\text{Re}(g)^2 + \text{Im}(g)^2}, \quad (34)$$

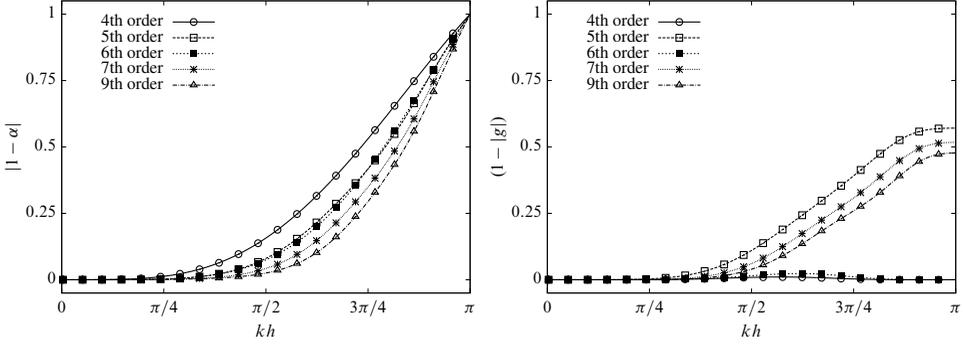
$$\text{Re}(g) = (1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4) - \frac{1}{2}y^2(1 + x + \frac{1}{2}x^2) + \frac{1}{24}y^4, \quad (35)$$

$$\text{Im}(g) = y(1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3) - \frac{1}{6}y^3(1 + x), \quad (36)$$

and  $z = x + iy$ . The normalized phase error,  $|1 - \alpha|$ , is defined using

$$\alpha = \frac{\alpha(\beta)}{|u|} = -\frac{1}{\sigma\beta} \frac{\text{Im}(g)}{\text{Re}(g)}, \quad (37)$$

where  $\beta = 2\pi kh$  for  $k = 0, \pm 1, \pm 2, \dots, \pm N/2$ .



**Figure 1.** Normalized phase error (left) and dissipation (right) for the high-order methods ( $\sigma = 0.8$ ).

*Spatial differencing eigenvalues.* The eigenvalues for each of the different high-order spatial differencing schemes are presented below. In each of the eigenvalue descriptions,  $\beta_d$  may range from  $-\pi$  to  $\pi$  and is defined as  $2\pi k_d h$  with  $k_d = 0, \pm 1, \pm 2, \dots, \pm N/2$ .

- Fourth-order centered difference:

$$\lambda_4 = \frac{i}{12h} \sum_{d=1}^D u^d [16 \sin(\beta_d) - 2 \sin(2\beta_d)]. \quad (38)$$

- Fifth-order upwind:

$$\lambda_5 = \frac{1}{60h} \sum_{d=1}^D u^d [(-2 \cos(3\beta_d) + 12 \cos(2\beta_d) - 30 \cos(\beta_d) + 20) + i(2 \sin(3\beta_d) - 18 \sin(2\beta_d) + 90 \sin(\beta_d))]. \quad (39)$$

- Sixth-order centered difference:

$$\lambda_6 = \frac{i}{60h} \sum_{d=1}^D u^d [2 \sin(3\beta_d) - 18 \sin(2\beta_d) + 90 \sin(\beta_d)]. \quad (40)$$

- Seventh-order upwind:

$$\lambda_7 = \frac{1}{420h} \sum_{d=1}^D u^d [(3 \cos(4\beta_d) - 24 \cos(3\beta_d) + 84 \cos(2\beta_d) - 168 \cos(\beta_d) + 105) + i(-3 \sin(4\beta_d) + 32 \sin(3\beta_d) - 168 \sin(2\beta_d) + 672 \sin(\beta_d))]. \quad (41)$$

- Ninth-order upwind:

$$\lambda_9 = \frac{1}{2520h} \sum_{d=1}^D u^d [(-4 \cos(5\beta_d) + 40 \cos(4\beta_d) - 180 \cos(3\beta_d) + 480 \cos(2\beta_d) - 840 \cos(\beta_d) + 504) + i(4 \sin(5\beta_d) - 50 \sin(4\beta_d) + 300 \sin(3\beta_d) - 1200 \sin(2\beta_d) + 4200 \sin(\beta_d))]. \quad (42)$$

### 3. Low-order flux computation

The low-order fluxes are computed using the CTU method [4; 21]. CTU is a first-order time-advancement scheme. The method is desirable over the simpler donor-cell upwind method because its stability is independent of dimensionality. However, this increased stability comes with a price. Instead of a single flux being defined by a single upwind value, the CTU flux is dependent upon a set of upwinded values. These values are determined by tracing the characteristic paths from the nodes that define the flux surface. This process involves an increasing number of Riemann solves as the dimensionality of the problem increases. In the 1D case, CTU is identical to donor-cell upwind.

### 4. Computing the hybridization coefficient

We compute the hybridization coefficient  $\eta$  using a modified multidimensional flux-corrected transport (FCT) algorithm. Note that the time superscript notation ( $n$ ) for fluxes is dropped for the remainder of the paper, but it is implied. Our algorithm is based upon the method described first in [28]. Here is the generic FCT procedure:

- (1) Compute the high-order fluxes  $\langle \mathbf{F}_H^d \rangle_{i \pm e^d / 2}$  over the cell volume  $V_i$ .
- (2) Compute the low-order fluxes  $\langle \mathbf{F}_L^d \rangle_{i \pm e^d / 2}$  and the corresponding low-order update

$$\langle q \rangle_i^{td} = \langle q \rangle_i^n - \frac{\Delta t}{h} \sum_{d=1}^D [\langle \mathbf{F}_L^d \rangle_{i+e^d/2} - \langle \mathbf{F}_L^d \rangle_{i-e^d/2}]. \quad (43)$$

- (3) Compute the antidiffusive fluxes

$$\langle A^d \rangle_{i \pm e^d / 2} = \langle \mathbf{F}_H^d \rangle_{i \pm e^d / 2} - \langle \mathbf{F}_L^d \rangle_{i \pm e^d / 2}. \quad (44)$$

- (4) Limit the antidiffusive fluxes:

$$\langle A_\eta^d \rangle_{i \pm e^d / 2} = \eta_{i \pm e^d / 2}^d \langle A^d \rangle_{i \pm e^d / 2}, \quad 0 \leq \eta_{i \pm e^d / 2}^d \leq 1. \quad (45)$$

(5) Update the solution with the limited antidiffusive fluxes:

$$\langle q \rangle_i^{n+1} = \langle q \rangle_i^{td} - \frac{\Delta t}{h} \sum_{d=1}^D [\langle A_\eta^d \rangle_{i+e^d/2} - \langle A_\eta^d \rangle_{i-e^d/2}]. \quad (46)$$

**Limiting the antidiffusive flux.** The primary challenge in the above formulation is computing the hybridization coefficients ( $\eta_{i \pm e^d/2}$ ). Following the procedure in [28], we compute the coefficients in the following manner.

Preconstrain the high-order fluxes  $\langle F_H \rangle_{i \pm e^d/2}$ . This is a prelimiting step that in effect sets  $\langle A^d \rangle_{i \pm e^d/2}$  to zero when it would otherwise admit diffusion and flatten the solution profile.

Compute the sum ( $P_i^\pm$ ) of all the antidiffusive fluxes into and out of the cell and a measure of the allowable flux into or out of the cell ( $Q_i^\pm$ ):

$$P_i^+ = \sum_{d=1}^D [\max(\langle A^d \rangle_{i-e^d/2}, 0) - \min(\langle A^d \rangle_{i+e^d/2}, 0)], \quad (47)$$

$$Q_i^+ = ((q_{\max})_i - \langle q \rangle_i^{td}) \frac{h}{\Delta t}, \quad (48)$$

$$P_i^- = \sum_{d=1}^D [\max(\langle A^d \rangle_{i+e^d/2}, 0) - \min(\langle A^d \rangle_{i-e^d/2}, 0)], \quad (49)$$

$$Q_i^- = (\langle q \rangle_i^{td} - (q_{\min})_i) \frac{h}{\Delta t}. \quad (50)$$

Compute the least upper bounds

$$R_i^+ = \begin{cases} \min(1.0, Q_i^+ / P_i^+) & \text{if } P_i^+ > 0.0, \\ 0.0 & \text{otherwise,} \end{cases} \quad (51)$$

$$R_i^- = \begin{cases} \min(1.0, Q_i^- / P_i^-) & \text{if } P_i^- > 0.0, \\ 0.0 & \text{otherwise.} \end{cases} \quad (52)$$

Select the hybridization coefficient with the most restrictive upper bound:

$$\eta_{i+e^d/2} = \begin{cases} \min(R_{i+e^d}^+, R_i^-) & \text{if } \langle A^d \rangle_{i+e^d/2} > 0.0, \\ \min(R_i^+, R_{i+e^d}^-) & \text{if } \langle A^d \rangle_{i+e^d/2} \leq 0.0. \end{cases} \quad (53)$$

In the above description the user is provided with two design choices: precondition for the high-order flux and method of computing the solution bounds  $(q_{\max})_i$  and  $(q_{\min})_i$ .

**Computing the solution bounds.** Compute initial estimates of the solution bounds,  $(q_{\max})_i$  and  $(q_{\min})_i$ . First, compute the bounded solutions in a rectangular stencil ( $B_i$ ) that is  $[2s_i + 1]^D$  cells in size, where  $s_i$  is the stencil size. Following convention the stencil size was fixed to be one cell.

After the stencil is determined, four bounds are computed: max based on  $\langle q \rangle^n$ , min based on  $\langle q \rangle^n$ , max based on  $\langle q \rangle^{td}$ , and min based on  $\langle q \rangle^{td}$ :

$$(q_{\max})_i^n = \max(B_i(\langle q \rangle^n)), \quad (54)$$

$$(q_{\min})_i^n = \min(B_i(\langle q \rangle^n)), \quad (55)$$

$$(q_{\max})_i^{td} = \max(B_i(\langle q \rangle^{td})), \quad (56)$$

$$(q_{\min})_i^{td} = \min(B_i(\langle q \rangle^{td})). \quad (57)$$

Then select the upper and lower bounds of the two estimates:

$$(q_{\max})_i = \max((q_{\max})_i^n, (q_{\max})_i^{td}), \quad (58)$$

$$(q_{\min})_i = \min((q_{\min})_i^n, (q_{\min})_i^{td}). \quad (59)$$

*Accurate solution bounds at smooth extrema.* For the vast majority of cells within the domain, the previous bound computation is sufficiently accurate. However, computing bounds at extrema is more complicated. Ideally the bounds need to keep the solution monotonic and positive, but the bounds should also not “clip” the solution. There are a few different methods for avoiding clipping, and we use a geometric construction that is only applied at smoothly varying extrema. It is based on the ideas in [6].

The first task is to detect a smooth extremum. The smooth-extremum criterion in 1D is

$$(\text{ext}^d)_i = \min[(dq)_i \cdot (dq)_{i+e^d}, (dq)_{i-e^d} \cdot (dq)_{i+2e^d}] \leq 0.0, \quad 1.25 \cdot (dq_{\text{tot}})_i < (tv)_i, \quad (60)$$

where

$$(dq)_i = \langle q \rangle_i^{td} - \langle q \rangle_{i-e^d}^{td}, \quad (61)$$

$$(dq_{\text{tot}})_i = |\langle q \rangle_{i+2e^d}^{td} - \langle q \rangle_{i-2e^d}^{td}|, \quad (62)$$

$$(tv)_i = |(dq)_{i+2e^d}| + |(dq)_{i+e^d}| + |(dq)_i| + |(dq)_{i-e^d}|. \quad (63)$$

This criterion has two parts. First, check for a sign change in the first derivative. The sign change will indicate either an extremum or a discontinuity in the solution. Second, ensure that the solution locally is not a perturbation of a discontinuity.

For a smooth multidimensional extremum, either  $(\text{ext}^d)_i$  must be true in all dimensions or it must be true for some  $d$  and the solution must remain constant along the dimensions in which  $(\text{ext}^d)_i$  is not true. We use this criterion to determine if the solution is constant:

$$\max(|(q_{\max}^d)_i^{td} - \langle q \rangle_i^{td}|, |(q_{\min}^d)_i^{td} - \langle q \rangle_i^{td}|) \leq 10^{-14}, \quad (64)$$

where

$$(q_{\max}^d)_i^{td} = \max(\langle q \rangle_{i-e^d}^{td}, \langle q \rangle_i^{td}, \langle q \rangle_{i+e^d}^{td}), \quad (65)$$

$$(q_{\min}^d)_i^{td} = \min(\langle q \rangle_{i-e^d}^{td}, \langle q \rangle_i^{td}, \langle q \rangle_{i+e^d}^{td}). \quad (66)$$

Once we have determined that the solution at  $V_i$  is at a smooth extremum, we compute new values of  $(q_{\max})_i$  and  $(q_{\min})_i$ . The first step is to construct a parabolic function from the local values of  $\langle q \rangle^n$ :

$$q^d(x) = \frac{1}{2}(d2q)_i^n x^2 + \frac{1}{2}(\langle q \rangle_{i+e^d}^n - \langle q \rangle_{i-e^d}^n)x + \langle q \rangle_i^n, \quad (67)$$

where

$$(d2q)_i^n = \langle q \rangle_{i+e^d}^n + \langle q \rangle_{i-e^d}^n - 2\langle q \rangle_i^n. \quad (68)$$

The location of the vertex ( $x_c$ ) is given by the ratio  $-b/2a$ , where  $a$  and  $b$  are the quadratic and linear coefficients from (67):

$$x_c = -\frac{\langle q \rangle_{i+e^d}^n - \langle q \rangle_{i-e^d}^n}{2(d2q)_i^n} \quad (69)$$

and  $-0.5 \leq x_c \leq 0.5$ . Then, we evaluate the quadratic at the vertex to find the extremum value as well as deconvolve to get an estimate of the point value:

$$(q_{\text{ext}}^d)_i = \frac{1}{2}(d2q)_i^n x_c^2 + \frac{1}{2}(\langle q \rangle_{i+e^d}^n - \langle q \rangle_{i-e^d}^n)x_c + \langle q \rangle_i^n - \frac{1}{24}(d2q)_i^n. \quad (70)$$

We select the largest  $(q_{\text{ext}}^d)_i$  or smallest  $(q_{\text{ext}}^d)_i$  depending on the sign of the second derivative:

$$(q_{\text{ext}})_i = \begin{cases} \max_d((q_{\text{ext}}^d)_i, (q_{\max})_i) & \text{if } \text{sgn}((d2q)_i^n) \leq 0.0, \\ \min_d((q_{\text{ext}}^d)_i, (q_{\min})_i) & \text{otherwise.} \end{cases} \quad (71)$$

Finally, we compute the appropriate extremum bound by augmenting the solution value at the previous time by a scaled difference between the extremum value and the solution value

$$(q_{\max})_i = \begin{cases} \langle q \rangle_i^n + 2.0[(q_{\text{ext}})_i - \langle q \rangle_i^n] & \text{if } \text{sgn}((d2q)_i^n) \leq 0.0, \\ (q_{\max})_i & \text{otherwise,} \end{cases} \quad (72)$$

$$(q_{\min})_i = \begin{cases} \langle q \rangle_i^n + 2.0[(q_{\text{ext}})_i - \langle q \rangle_i^n] & \text{if } \text{sgn}((d2q)_i^n) > 0.0, \\ (q_{\min})_i & \text{otherwise.} \end{cases} \quad (73)$$

*Updating  $R_i^\pm$  at extrema.* We flag the extrema at which the Laplacian is changing sign. In most cases this flag should not be activated. However, if the Laplacian does change sign at a smooth extremum, we turn the limiter on so that the low-order flux is chosen. This is a protective measure we have included in the algorithm.

We compute the  $D$ -dimensional approximation to the Laplacian over a three-point stencil

$$\Delta q_i = \sum_{d=1}^D \frac{\partial^2 q_i}{\partial x_d^2} \approx \sum_{d=1}^D \frac{(d2q)_i^n}{h^2}. \quad (74)$$

If  $\Delta q$  changes sign anywhere in the three-point vicinity of  $\mathbf{i}$ , then we flag that cell  $\mathbf{i}$ . We then update the least upper bound multiplier at the flagged cells:

$$R_i^\pm = 0 \quad \text{if } \mathbf{i} \text{ flagged.} \quad (75)$$

**Preconstraining the high-order flux.** We prestrain the high-order flux where the corresponding antidiffusive flux would admit diffusion and flatten the solution profile. In practice, the value of the antidiffusive flux is edited instead of the high-order flux directly. Following [28] we set the antidiffusive flux to zero in these regions.

The baseline condition for applying the preconstraint is

$$\langle A^d \rangle_{i+e^d/2} (\langle q \rangle_{i+e^d}^{td} - \langle q \rangle_i^{td}) \leq 0.0. \quad (76)$$

However, this condition was not sufficient for our algorithm. The condition was occasionally satisfied at smooth areas in the solution. This manifested itself as a drop in convergence rate. We noticed that the preconstraint was mainly being applied near steep gradients and discontinuities. Moving forward, we only want to apply the preconstraint at discontinuities.

We added the following requirements to make sure we only apply this condition away from smooth areas:

$$\min[(d2q)_{i+e^d}^n \cdot (d2q)_i^n, (d2q)_i^n \cdot (d2q)_{i-e^d}^n, (d2q)_{i+e^d}^n \cdot (d2q)_{i+2e^d}^n] < 0.0, \quad (77)$$

$$|\langle A^d \rangle_{i+e^d/2}| \leq \frac{|(\mathbf{u}^d)_{i+e^d/2}| h}{2} (1 - \sigma_{i+e^d/2}) \frac{|(d2q)_i + (d2q)_{i+e^d}|}{2}, \quad (78)$$

where  $\sigma_{i+e^d/2} = |(\mathbf{u}^d)_{i+e^d/2}| \Delta t / h$ .

The first constraint above (77) attempts to detect a discontinuity in the solution. However, there are smooth multidimensional solutions in which the second derivative naturally changes sign. The second constraint (78) seeks to preclude this case. The term on the right-hand side of the inequality (78) is the  $d$ -directional dissipation term, scaled by the cell size, in the modified equation analysis of CTU applied to the advection equation:

$$\frac{\partial q}{\partial t} + \sum_{d=1}^D \left( \mathbf{u}^d \frac{\partial q}{\partial x_d} \right) = \sum_{d=1}^D \left( \frac{\mathbf{u}^d h}{2} (1 - \sigma_d) \frac{\partial^2 q}{\partial x_d^2} \right) + \mathcal{O}(h^2). \quad (79)$$

We are interested in comparing the magnitude of this dissipative term to the antidiffusive flux. The magnitude of the dissipative term is large in the neighborhood

of discontinuities. If this magnitude is large relative to the antidiffusive term, then we assume we are near a discontinuity and allow the preconstraint.

*Steepening.* In the previous subsection, we stated that the antidiffusive flux is set to zero in the regions where it would flatten the profile. As recognized in [8; 15], we may alternatively reverse the sign of and scale the antidiffusive flux. This process, known as steepening, seems to produce even sharper solution profiles at fronts. However, steepening has led to robustness issues in the past, including producing overshoots in the solution if the scaling factor is too large. For this reason we elected to keep the scaling factor at zero for the vast majority of the tests in this study.

If steepening is deemed necessary, we make the following changes to the algorithm: make the smoothness check more stringent and scale the antidiffusive flux instead of zeroing it. We found that scaling the right-hand side of (78) by 0.5 worked well to ensure the solution was discontinuous. In addition, we found that the best scaling factor for the tests in this study was 2.0.

## 5. Results

Results in one and two dimensions are presented. A total of four initial conditions were investigated. Of the four, one initial condition was smooth and the others contained a discontinuity. For the two-dimensional tests, we used a constant diagonal velocity field and a solid-body-rotation velocity field:

$$\mathbf{u} = [1, 1], \quad (80)$$

$$\mathbf{u} = 2\pi[y-0.5, 0.5-x]. \quad (81)$$

The center for the constant-velocity initial condition was in the middle of the domain, whereas it was offset by 0.25 of the grid height for the solid-body-rotation examples:

$$\mathbf{x}_c^{\text{const}} = (0.5, 0.5),$$

$$\mathbf{x}_c^{\text{solid}} = (0.5, 0.75).$$

**Initial conditions.** The smooth initial condition was constructed as a power of cosines

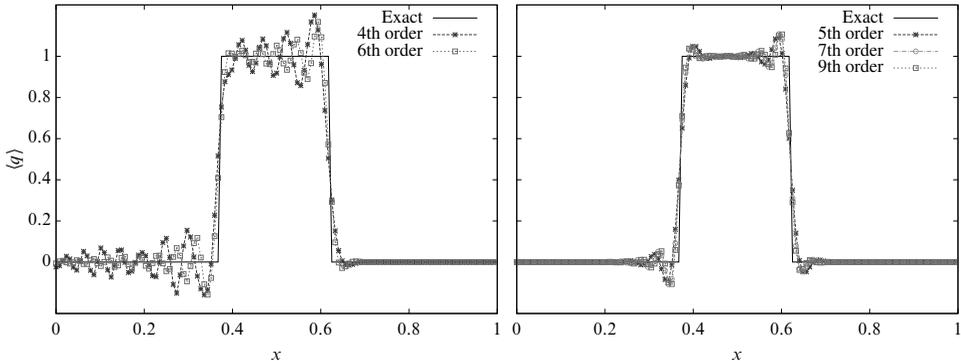
$$q_i(t_0) = \begin{cases} \cos^8\left(\frac{\pi}{2}(R/R_0)\right) & \text{if } R \leq R_0, \\ 0 & \text{otherwise,} \end{cases} \quad (82)$$

with

$$R = \sqrt{(\mathbf{x}_i - \mathbf{x}_c)^2}, \quad \mathbf{x}_i \in [0, 1], \quad R_0 = 0.15.$$

Three different discontinuous initial conditions were investigated. The first was a square and is described as

$$q_i(t_0) = \begin{cases} 1 & \text{if } |x_i^D - x_c^D| \leq 0.15 \text{ for each } D, \\ 0 & \text{otherwise.} \end{cases} \quad (83)$$



**Figure 2.** Centered (left) versus upwind (right) difference methods without limiting ( $\sigma = 0.8$ ,  $t = 1.0$ , and  $N = 128$ ).

The next is a semiellipse

$$q_i(t_0) = \begin{cases} \sqrt{1.0 - (R/R_0)^2} & \text{if } R \leq R_0, \\ 0 & \text{otherwise,} \end{cases} \quad \text{with } R_0 = 0.25. \quad (84)$$

The last test case is the classic slotted cylinder in two dimensions

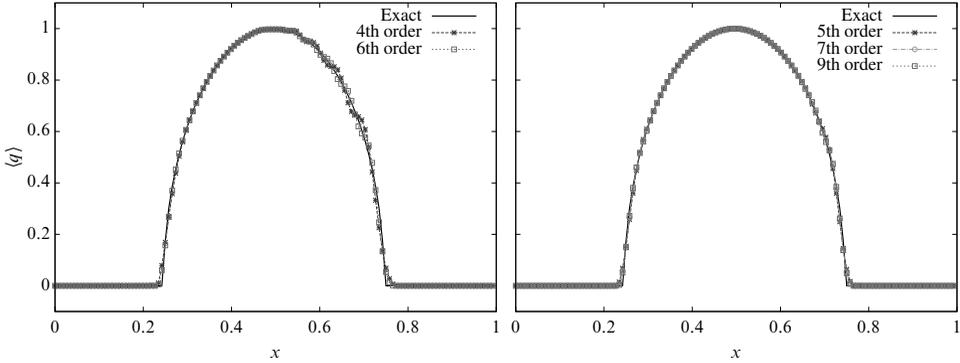
$$q_i(t_0) = \begin{cases} 1 & \text{if } |x_i^0 - x_c^0| \geq 0.025 \text{ or } x_i^1 \geq 0.85 \text{ (provided } R \leq 0.15), \\ 0 & \text{otherwise,} \end{cases} \quad (85)$$

with  $R = \sqrt{(x_i - x_c^{\text{solid}})^2}$ .

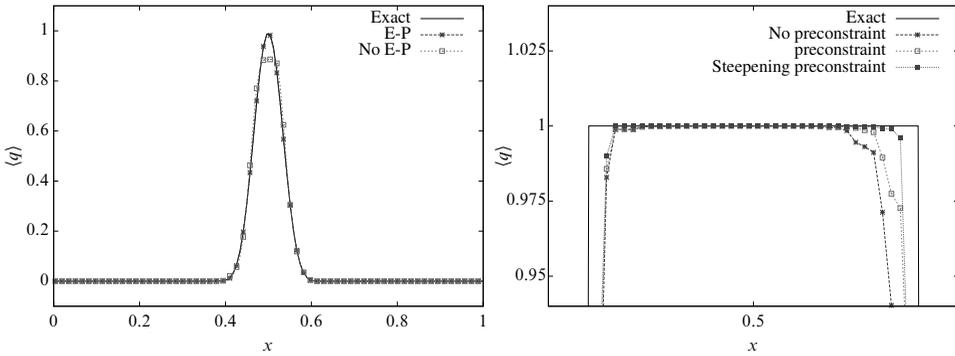
**Effectiveness of design features.** We seek to demonstrate the need for each of the design choices made in the algorithm. The first feature is upwind methods for high-order fluxes. To show the effectiveness of the upwind methods, we examined the performance of the high-order fluxes with no limiting (Figure 2) on the square initial condition (83) in 1D. In the presence of a discontinuity, upwind methods produced much smaller magnitude oscillations than centered-difference methods. This outcome is consistent with the amplitude and phase error analysis presented earlier. When the limiter is turned on, the oscillations are clipped for both types of fluxes but the dispersive errors remain in the centered-difference solutions. Figure 3 shows this remaining dispersive error on the semiellipse initial condition (84).

The second design feature is the extremum-preserving limiter. We simulated advection with the smooth cosine initial condition (82) in 2D. This initial condition has a smoothly varying extremum in the middle of the domain. Figure 4, left, shows the excessive diffusion at this extremum that results from not using the extremum-preserving limiter. Not only is there excess diffusion, but also the convergence rate of the method suffers.

The precondition is another important feature of this algorithm. Figure 4, right, shows the effectiveness of the precondition on the square initial condition running



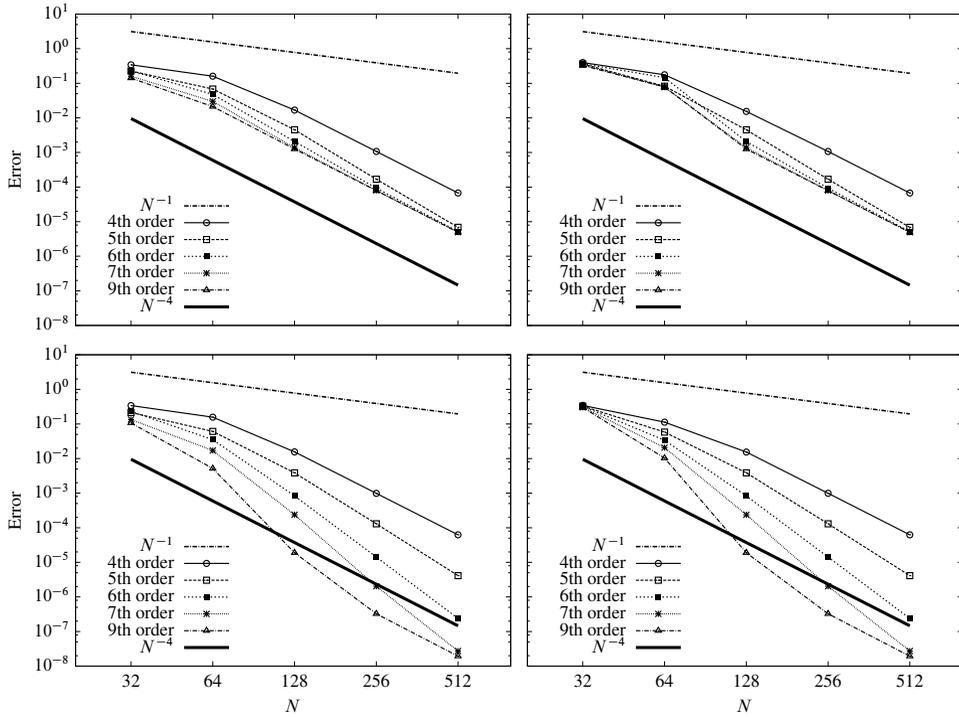
**Figure 3.** Centered (left) versus upwind (right) difference methods with limiting ( $\sigma = 0.8$ ,  $t = 1.0$ , and  $N = 128$ ).



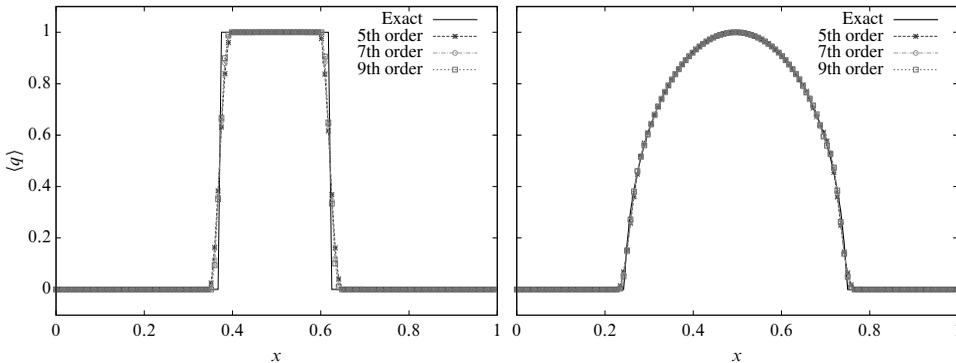
**Figure 4.** Effectiveness of extremum-preserving limiter at  $t = 5.0$  (left) and preconstraint and steepener in 2D at  $t = 1.0$  (right) ( $\sigma = 0.8$  and  $N = 128$ ).

right at the stability limit of the method in 2D. The preconstraint sharpens the solution profile near the front. The steepening preconstraint improves the solution profile even further. However, for most of the tests in this study, our preconstraint produced identical results with and without steepening. The remaining results do not include steepening.

**One-dimensional tests.** The first requirement for the limiter method is that it reduces to the high-order scheme away from discontinuities. All of the high-order schemes running at a large CFL number ( $\sigma = 0.8$ ) achieved similar errors for smooth solutions in 1D (Figure 5). At this CFL number, the rate of convergence for each method was 4.0. We also computed the errors running at a lower CFL number ( $\sigma = 0.2$ ). The error-reduction rate for each scheme at the low CFL number roughly matched the order of the spatial differencing scheme for the grid sizes displayed here. These results demonstrate that the limiter is not being activated in smooth regions, and hence, the first requirement is met.

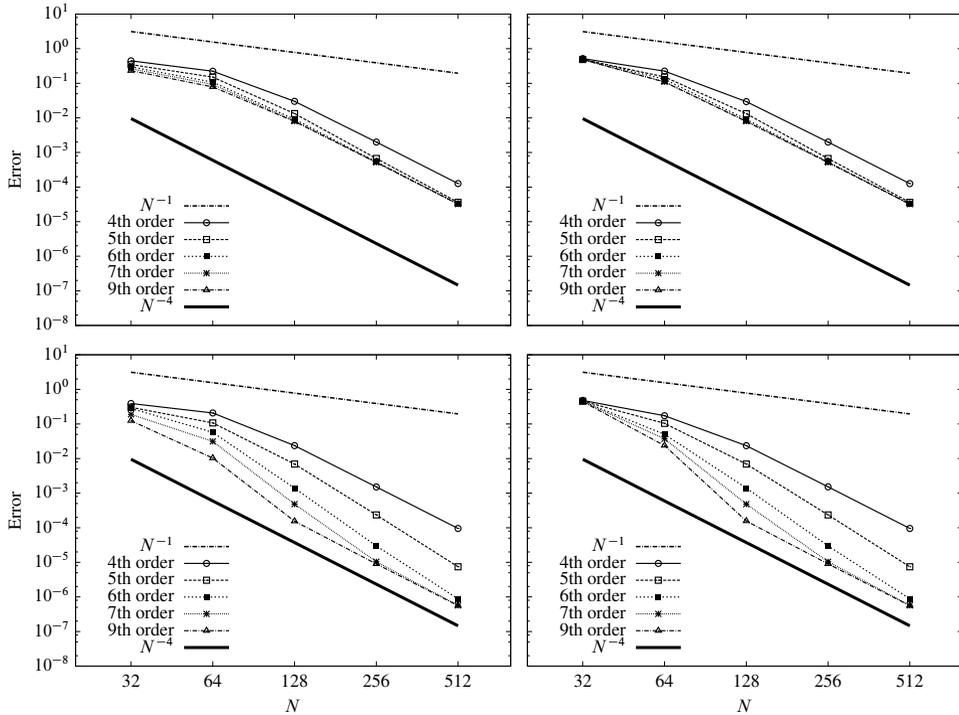


**Figure 5.**  $L_\infty$  errors in 1D. Left column: error without limiter. Right column: error with limiter. Top row:  $\sigma = 0.8$ . Bottom row:  $\sigma = 0.2$ .



**Figure 6.** Discontinuous solutions with limiting and square (left) and semiellipse (right) initial conditions ( $\sigma = 0.8$ ,  $t = 1.0$ , and  $N = 128$ ).

The second requirement is that the limiter method accurately represent discontinuities. Figure 6 shows how the limiter performs on the square and semiellipse initial conditions in 1D. For both cases, the limiter method accurately captures the front and keeps the solution bounded.

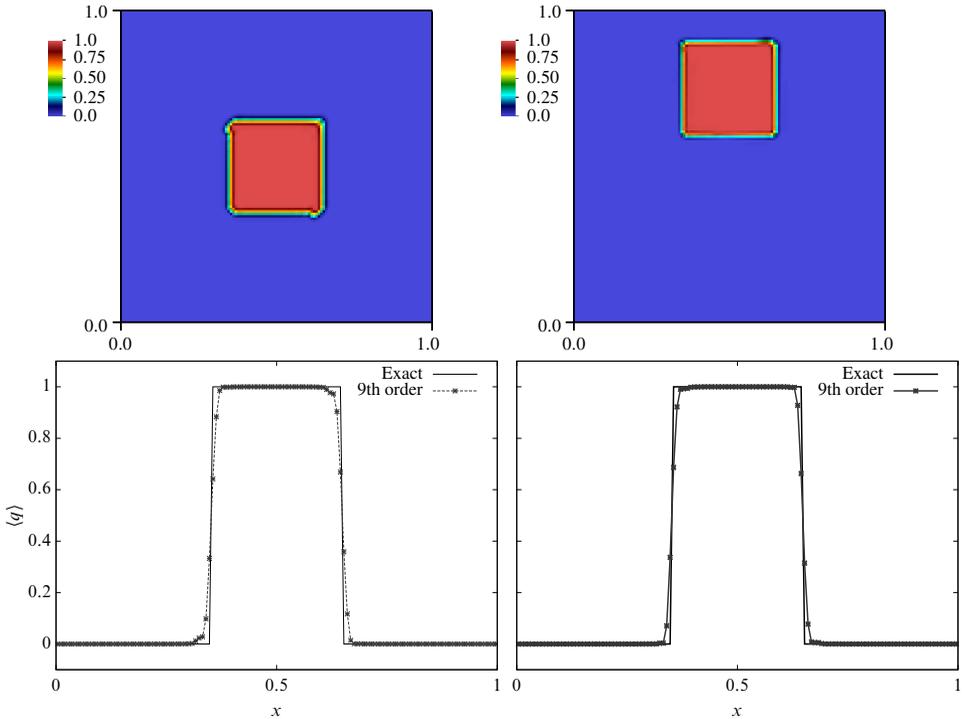


**Figure 7.**  $L_\infty$  errors in 2D. Left column: without limiter. Right column: with limiter. Top row: constant velocity. Bottom row: rotation velocity.

**Two-dimensional tests.** The errors for the smooth initial condition test in two dimensions are reported with and without the limiter for both velocity fields (Figure 7). As in the one-dimensional case, the high-order solution accuracy requirement is met. Interestingly, for the solid-body-rotation solution, the error reduction is greater than fourth order for many of the spatial differencing schemes. Also the ninth-order scheme is still convergent right at its theoretical stability limit ( $\sigma \approx 0.8$ ).

The limiter also performs quite well at representing discontinuities in two dimensions. Various solution plots for discontinuous initial conditions are presented (Figures 8–10). All of the two-dimensional plots were generated using the ninth-order scheme in space, running at  $\sigma = 0.8$ . The square solution under constant velocity has few, if any, ripples and is nicely bounded (Figure 8). There is some distortion of the corners, particularly at the top-left and bottom-right. The square solution under solid-body rotation looks similar to the constant-velocity solution, and the corner issue is mitigated.

The semiellipse solution is likewise well resolved (Figure 9). As with the one-dimensional case there are some dispersive errors on the leading edge, but they are small. The solutions under both velocity fields are accurately bounded. The



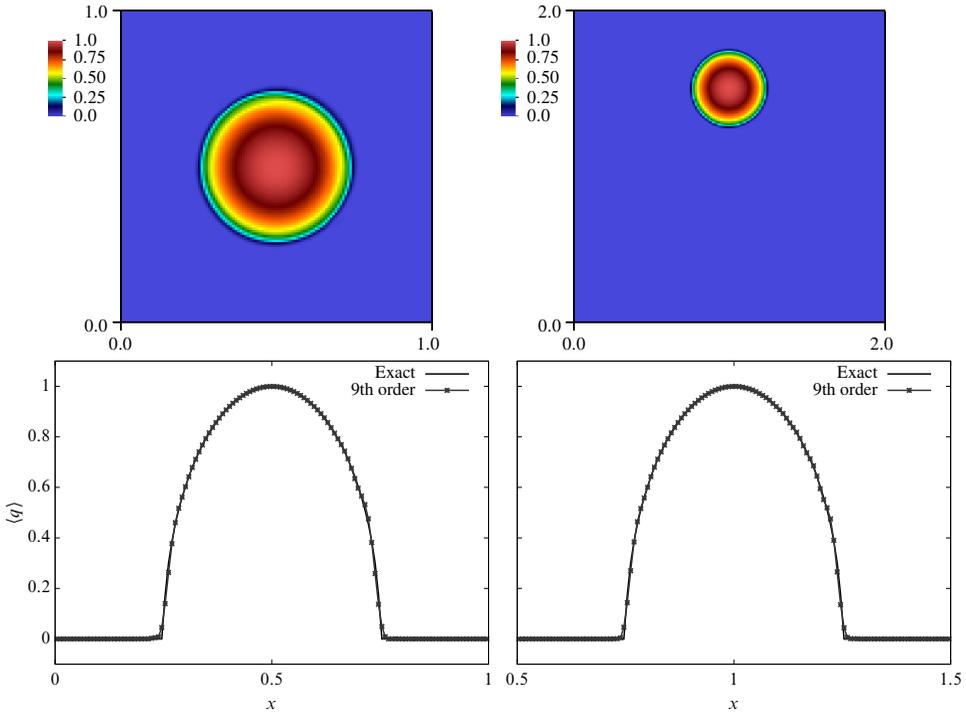
**Figure 8.** Square solutions ( $\sigma = 0.8$ ,  $t = 1.0$ , and  $N = 128$ ). Top left: constant velocity. Top right: solid-body rotation. Bottom row: centerline comparison.

semiellipse solution under solid-body rotation was centered at  $\mathbf{x}_c^{\text{solid}} = (1.0, 1.5)$  to keep the edge of the condition away from the domain boundary. The domain was also expanded to  $x_i \in [0, 2]$ .

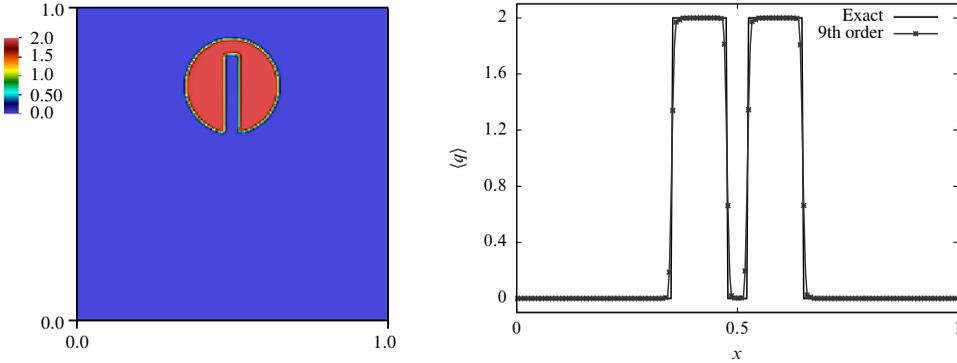
The final test was the slotted cylinder (Figure 10). The limiter method keeps the solution bounded and resolves the fronts quite nicely. At lower grid resolutions, the slot can fill in and the bounds may not be enforced. But as the grid is refined, both of these issues are resolved.

## 6. Conclusions

We presented a new flux limiter based upon FCT that retains high-order accuracy for smooth solutions and captures fronts well. Our algorithm presented here uses CTU for low-order fluxes, upwind schemes for high-order fluxes, and RK4 for time integration. Our additions to the previous FCT method included a new computation for the extrema, an expanded precondition on the high-order fluxes, and a sixth-order-accurate finite-volume product rule. Furthermore, the limiter was only applied once per each time advance.



**Figure 9.** Semiellipse solutions ( $\sigma = 0.8$ ,  $t = 1.0$ , and  $N = 128$ ). Top left: constant velocity. Top right: solid-body rotation. Bottom row: centerline comparison.



**Figure 10.** Slotted cylinder under solid-body rotation ( $\sigma = 0.8$ ,  $t = 1.0$ , and  $N = 256$ ). Left: solution. Right: centerline comparison.

The convergence rates for the smooth initial-condition tests were extremely similar in all three of the standard norms:  $L_1$ ,  $L_2$ , and  $L_\infty$ . On account of this only the  $L_\infty$  errors were displayed. Each high-order spatial discretization achieved fourth-order accuracy, at a minimum, for the smooth initial condition. In theory, we could also have achieved a higher overall order of accuracy with a more accurate

temporal integration scheme. Fourth-order accuracy was expected since we used a fourth-order-accurate time integrator (RK4) and a relatively large CFL number (0.8) for the majority of the test cases presented. At this CFL number, the error from the temporal discretization is much larger than the spatial discretization. To confirm this we also ran simulations at a low CFL number (0.2). The algorithm achieved higher convergence rates at this lower CFL number. We concluded that the errors from the spatial differencing schemes dominate the solution when running at low CFL numbers. The solid-body-rotation test also produced higher-order convergence rates than expected. This is likely explained by the fact that the solution is being advected at a range of CFL numbers tending toward zero as the solution approaches the center of the domain. Another interesting feature is the large difference in errors between the various spatial differencing schemes for the solid-body-rotation example. At the higher CFL number there are nearly two orders of magnitude difference in the max-norm error between the fourth- and ninth-order schemes. For the same initial condition running at the lower CFL number, there are over three orders of magnitude difference between these max-norm errors.

Potential extensions for this work are applying the limiter to systems of hyperbolic conservation laws, developing new high-order upwind methods with corner coupling, and further improving the precondition on the high-order fluxes. Applying the limiter to hyperbolic systems is the most direct extension of this work. Implementing this limiter for a compressible gas dynamics solver would be a good starting point. Following [15] we would not apply the limiter directly to the conserved variables but rather to the characteristic variables. The equations that govern the characteristic variables in 1D look like a system of decoupled advection equations, and the ideas presented here have clear applicability. Characteristic decomposition and limiting become more difficult in multiple dimensions, but there is a road map to follow. There are two general approaches available: either compute the fluxes and limiter along each direction in a split manner, or compute the fluxes in a multidimensional manner and limit the directional fluxes independently or sequentially. In previous studies [15] there was no discernible difference between the two approaches for the test cases analyzed, but it is important to consider both in general.

High-order, corner-coupled upwind methods for use with general multistage time integrators could remove the dimensional dependence of the stability. However, no upwind method of this nature currently exists. The precondition on the high-order fluxes is another area where additional study could pay off. In this work we found that the precondition affected a delicate balance between effectively representing discontinuities and retaining high-order accuracy in smooth yet complex areas. It was relatively simple to achieve one or the other. Ensuring both required testing many versions of the precondition. Introducing a steepening coefficient with the precondition improved discontinuity representation, but additional work must be

done to ensure robustness. The primary robustness concern with steepening is avoiding overshoot in the solution; however, our restrictions of only applying the precondition near fronts as well as limiting smooth extrema at which the Laplacian changes sign should reduce the risk of overshoot. We could also use other antidiffusive approaches for handling these types of contact discontinuities [27]. On the other hand, several canonical hyperbolic systems of equations have steepening mechanisms built into the physics, so steepening may only be useful in certain limited contexts.

### Acknowledgements

This research was supported at the Lawrence Berkeley National Laboratory by the Office of Advanced Scientific Computing Research of the U.S. Department of Energy under Contract Number DE-AC02-05CH11231.

### References

- [1] D. S. Balsara, C. Meyer, M. Dumbser, H. Du, and Z. Xu, *Efficient implementation of ADER schemes for Euler and magnetohydrodynamical flows on structured meshes — speed comparisons with Runge–Kutta methods*, *J. Comput. Phys.* **235** (2013), 934–969.
- [2] J. P. Boris and D. L. Book, *Flux-corrected transport, I: SHASTA, a fluid transport algorithm that works*, *J. Comput. Phys.* **11** (1973), no. 1, 38–69.
- [3] E. Casoni, J. Peraire, and A. Huerta, *One-dimensional shock-capturing for high-order discontinuous Galerkin methods*, *Internat. J. Numer. Methods Fluids* **71** (2013), no. 6, 737–755.
- [4] P. Colella, *Multidimensional upwind methods for hyperbolic conservation laws*, *J. Comput. Phys.* **87** (1990), no. 1, 171–200.
- [5] P. Colella, M. R. Dorr, J. A. F. Hittinger, and D. F. Martin, *High-order, finite-volume methods in mapped coordinates*, *J. Comput. Phys.* **230** (2011), no. 8, 2952–2976.
- [6] P. Colella and M. D. Sekora, *A limiter for PPM that preserves accuracy at smooth extrema*, *J. Comput. Phys.* **227** (2008), no. 15, 7069–7076.
- [7] P. Colella and P. R. Woodward, *The Piecewise Parabolic Method (PPM) for gas-dynamical simulations*, *J. Comput. Phys.* **54** (1984), no. 1, 174–201.
- [8] C. R. DeVore, *An improved limiter for multidimensional flux-corrected transport*, technical report NRL/MR/6440-98-8330, Naval Research Laboratory, Washington, DC, 1998.
- [9] M. Dumbser, O. Zanotti, A. Hidalgo, and D. S. Balsara, *ADER-WENO finite volume schemes with space-time adaptive mesh refinement*, *J. Comput. Phys.* **248** (2013), 257–286.
- [10] S. H. Fuller and L. I. Millett (eds.), *The future of computing performance: game over or next level?*, National Academies, Washington, DC, 2011.
- [11] A. Harten, *High resolution schemes for hyperbolic conservation laws*, *J. Comput. Phys.* **49** (1983), no. 3, 357–393.
- [12] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, *Uniformly high-order accurate essentially nonoscillatory schemes, III*, *J. Comput. Phys.* **71** (1987), no. 2, 231–303.
- [13] D. Kuzmin, *A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods*, *J. Comput. Appl. Math.* **233** (2010), no. 12, 3077–3085.

- [14] ———, *Hierarchical slope limiting in explicit and implicit discontinuous Galerkin methods*, J. Comput. Phys. **257B** (2014), 1140–1162.
- [15] D. Kuzmin, R. Löhner, and S. Turek (eds.), *Flux-corrected transport: principles, algorithms, and applications*, 2nd ed., Springer, Dordrecht, 2012.
- [16] D. Kuzmin, M. Möller, and S. Turek, *High-resolution FEM-FCT schemes for multidimensional conservation laws*, Comput. Methods Appl. Mech. Engrg. **193** (2004), no. 45–47, 4915–4946.
- [17] X.-D. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, J. Comput. Phys. **115** (1994), no. 1, 200–212.
- [18] R. Löhner, K. Morgan, J. Peraire, and M. Vahdati, *Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier–Stokes equations*, Internat. J. Numer. Methods Fluids **7** (1987), no. 10, 1093–1109.
- [19] R. Löhner, K. Morgan, M. Vahdati, J. P. Boris, and D. L. Book, *FEM-FCT: combining unstructured grids with high resolution*, Comm. Appl. Numer. Methods **4** (1988), no. 6, 717–729.
- [20] P. McCorquodale and P. Colella, *A high-order finite-volume method for conservation laws on locally refined grids*, Commun. Appl. Math. Comput. Sci. **6** (2011), no. 1, 1–25.
- [21] J. Saltzman, *An unsplit 3D upwind method for hyperbolic conservation laws*, J. Comput. Phys. **115** (1994), no. 1, 153–168.
- [22] C.-W. Shu, *High order weighted essentially nonoscillatory schemes for convection dominated problems*, SIAM Rev. **51** (2009), no. 1, 82–126.
- [23] V. A. Titarev and E. F. Toro, *ADER schemes for three-dimensional non-linear hyperbolic systems*, J. Comput. Phys. **204** (2005), no. 2, 715–736.
- [24] E. F. Toro and V. A. Titarev, *Solution of the generalized Riemann problem for advection-reaction equations*, R. Soc. Lond. Proc. A **458** (2002), no. 2018, 271–281.
- [25] B. van Leer, *Towards the ultimate conservative difference scheme, V: A second-order sequel to Godunov’s method*, J. Comput. Phys. **32** (1979), no. 1, 101–136.
- [26] S. Williams, A. Waterman, and D. Patterson, *Roofline: an insightful visual performance model for multicore architectures*, Commun. ACM **52** (2009), no. 4, 65–76.
- [27] Z. Xu and C.-W. Shu, *Anti-diffusive flux corrections for high order finite difference WENO schemes*, J. Comput. Phys. **205** (2005), no. 2, 458–485.
- [28] S. T. Zalesak, *Fully multidimensional flux-corrected transport algorithms for fluids*, J. Comput. Phys. **31** (1979), no. 3, 335–362.

Received May 4, 2015. Revised January 13, 2017.

CHRISTOPHER CHAPLIN: CChaplin@lbl.gov  
*Applied Numerical Algorithms Group, Computational Research Division,  
 Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 50A-3111, Berkeley, CA 94720,  
 United States*

PHILLIP COLELLA: PColella@lbl.gov  
*Applied Numerical Algorithms Group, Computational Research Division,  
 Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 50A-3111, Berkeley, CA 94720,  
 United States*

# Communications in Applied Mathematics and Computational Science

msp.org/camcos

## EDITORS

### MANAGING EDITOR

John B. Bell  
Lawrence Berkeley National Laboratory, USA  
jbbell@lbl.gov

### BOARD OF EDITORS

Marsha Berger	New York University berger@cs.nyu.edu	Ahmed Ghoniem	Massachusetts Inst. of Technology, USA ghoniem@mit.edu
Alexandre Chorin	University of California, Berkeley, USA chorin@math.berkeley.edu	Raz Kupferman	The Hebrew University, Israel raz@math.huji.ac.il
Phil Colella	Lawrence Berkeley Nat. Lab., USA pcolella@lbl.gov	Randall J. LeVeque	University of Washington, USA rjl@amath.washington.edu
Peter Constantin	University of Chicago, USA const@cs.uchicago.edu	Mitchell Luskin	University of Minnesota, USA luskin@umn.edu
Maksymilian Dryja	Warsaw University, Poland maksymilian.dryja@acn.waw.pl	Yvon Maday	Université Pierre et Marie Curie, France maday@ann.jussieu.fr
M. Gregory Forest	University of North Carolina, USA forest@amath.unc.edu	James Sethian	University of California, Berkeley, USA sethian@math.berkeley.edu
Leslie Greengard	New York University, USA greengard@cims.nyu.edu	Juan Luis Vázquez	Universidad Autónoma de Madrid, Spain juanluis.vazquez@uam.es
Rupert Klein	Freie Universität Berlin, Germany rupert.klein@pik-potsdam.de	Alfio Quarteroni	Ecole Polytech. Féd. Lausanne, Switzerland alfio.quarteroni@epfl.ch
Nigel Goldenfeld	University of Illinois, USA nigel@uiuc.edu	Eitan Tadmor	University of Maryland, USA etadmor@cscamm.umd.edu
		Denis Talay	INRIA, France denis.talay@inria.fr

## PRODUCTION

production@msp.org

Silvio Levy, Scientific Editor

---

See inside back cover or [msp.org/camcos](http://msp.org/camcos) for submission instructions.

The subscription price for 2017 is US \$100/year for the electronic version, and \$150/year (+\$15, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscriber address should be sent to MSP.

Communications in Applied Mathematics and Computational Science (ISSN 2157-5452 electronic, 1559-3940 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840, is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

---

CAMCoS peer review and production are managed by EditFLOW® from MSP.

PUBLISHED BY

 **mathematical sciences publishers**  
nonprofit scientific publishing

<http://msp.org/>

© 2017 Mathematical Sciences Publishers

# *Communications in Applied Mathematics and Computational Science*

vol. 12

no. 1

2017

---

- A single-stage flux-corrected transport algorithm for high-order finite-volume methods 1  
CHRISTOPHER CHAPLIN and PHILLIP COLELLA
- Achieving algorithmic resilience for temporal integration through spectral deferred corrections 25  
RAY W. GROUT, HEMANTH KOLLA, MICHAEL L. MINION and JOHN B. BELL
- A fourth-order Cartesian grid embedded boundary method for Poisson's equation 51  
DHARSHI DEVENDRAN, DANIEL T. GRAVES, HANS JOHANSEN and TERRY LIGOCKI
- A central-upwind geometry-preserving method for hyperbolic conservation laws on the sphere 81  
ABDELAZIZ BELJADID and PHILIPPE G. LEFLOCH
- Time-parallel gravitational collapse simulation 109  
ANDREAS KREIENBUEHL, PIETRO BENEDUSI, DANIEL RUPRECHT and ROLF KRAUSE



1559-3940(2017)12:1;1-G