Journal of Computational Physics 274 (2014) 1–18

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A WENO-limited, ADER-DT, finite-volume scheme for efficient, robust, and communication-avoiding multi-dimensional transport

Matthew R. Norman

Oak Ridge National Laboratory, PO Box 2008 MS6016, Oak Ridge, TN 37831, United States

ARTICLE INFO

Article history:
Received 2 January 2014
Received in revised form 15 April 2014
Accepted 30 May 2014
Available online 5 June 2014

Keywords:
ADER
WENO
FCT
Differential transform
Finite-volume

ABSTRACT

The novel ADER-DT time discretization is applied to two-dimensional transport in a quadrature-free, WENO- and FCT-limited, Finite-Volume context. Emphasis is placed on (1) the serial and parallel computational properties of ADER-DT and this framework and (2) the flexibility of ADER-DT and this framework in efficiently balancing accuracy with other constraints important to transport applications. This study demonstrates a range of choices for the user when approaching their specific application while maintaining good parallel properties. In this method, genuine multi-dimensionality, single-step and single-stage time stepping, strict positivity, and a flexible range of limiting are all achieved with only one parallel synchronization and data exchange per time step. In terms of parallel data transfers per simulated time interval, this improves upon multi-stage time stepping and post-hoc filtering techniques such as hyperdiffusion. This method is evaluated with standard transport test cases over a range of limiting options to demonstrate quantitatively and qualitatively what a user should expect when employing this method in their application.

Published by Elsevier Inc.

1. Introduction

Transport is an operation performed by a number of applications. Though described by a particularly simple PDE, the constraints on the numerical solution are rigorous. Not only must the method be accurate and economic in runtime, it must also be positivity preserving, shape preserving, and correlation preserving [1], and mass conserving. Correlation preservation means that the pre-existing spatial correlation between two tracers often needs to be preserved as both are evolved. However, some of these constraints are opposed to one another, requiring some measure of trade-off. Accuracy and shape preservation, for instance, are usually opposed to one another in that shape preservation often over damps the solution and clips extrema in smooth regions of flow. Finding a way to balance these constraints on a spectrum could better help modelers create a more ideal fit for their specific application.

Atmospheric transport is a well-researched area of atmospheric modeling in general. There have been a number of semi-Lagrangian (SL), also called Arbitrary Lagrangian–Eulerian (ALE), methods implemented with varying novelties such as local conservation [2–5], flux-form formulation [6–8], various functional approximation models [9], deformed grid geometries [10,11], and methods of limiting [7,2,6,12,13,8]. There have also been Eulerian implementations of transport in atmospheric models [11,10,13]. For transport, SL methods often use CFL numbers that are limited only by accuracy, which itself is largely

E-mail address: normanmr@ornl.gov.

http://dx.doi.org/10.1016/j.jcp.2014.05.043
0021-9991/Published by Elsevier Inc.

due to trajectory errors. These CFL numbers can typically be much larger than the Eulerian maximum stable CFL. The larger the CFL number becomes, the larger the halo region for parallel data transfer becomes. For a more thorough review of SL methods in particular, please see [14,8] and references therein.

SL schemes have an advantage in terms of parallel efficiency because in a single stage, they can perform arbitrarily high-order-accurate time integration of transport, given the trajectories are sufficiently accurate. SL methods translate spatial information into temporal information through these trajectories. The present method, though in the Eulerian framework, shares some of the philosophy of SL methods because the ADER procedure also translates spatial information into temporal information. Like SL methods, ADER-DT methods perform arbitrarily high-order-accurate time integration in a single stage through the PDE definition itself and the use of Differential Transforms (DTs) [15,16]. Also like SL methods, ADER maintains non-oscillatory properties of an underlying reconstruction in the time integration. Unlike SL methods, however, trajectories do not need to be formed for ADER methods, but rather the temporal evolution is generated from within a given cell.

In Section 2, the numerical method is described in detail including the framework, time discretization, spatial reconstruction, limiting, and computational properties. In Section 3, the limited ADER-DT methods are evaluated using standard test problems with error norms and other quantitative and qualitative evaluations. Finally, conclusions are drawn in Section 4.

2. Multi-dimensional numerical framework

This study is concerned with the 2-D, Cartesian transport equation for the scalar mixing ratio, $\psi(x, y, t)$, of a passive tracer quantity. The quantities actually stored and evolved in a typical flux-form fluid model are density (ρ), momenta (ρu and ρv), and tracer mass ($\rho\psi$), where $u(x, y, t)$ and $v(x, y, t)$ are fluid velocities in the x- and y-dimensions, respectively. Therefore, the momenta are defined with bold face as $\mathbf{u} \equiv \rho u$ and $\mathbf{v} \equiv \rho v$ and the tracer mass as $q \equiv \rho\psi$. Thus, the transport equation is stated as:

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0 \tag{1}$$

where $f = q\mathbf{u}/\rho$ and $g = q\mathbf{v}/\rho$.

2.1. Finite-volume framework

The local cell domain is defined by $\Omega_{i,j} \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ where $x_{i\pm\frac{1}{2}} = x_i \pm \frac{1}{2}\Delta x_i$, $y_{j\pm\frac{1}{2}} = y_j \pm \frac{1}{2}\Delta y_j$, and Δx_i and Δy_j are the grid spacings in the x- and y-directions, respectively, for cell $\Omega_{i,j}$. To form the Finite-Volume (FV) evolution equation, (1) is integrated over a local space–time domain defined by $\Omega_{i,j} \times [t_n, t_{n+1}]$, where $t_{n+1} = t_n + \Delta t_n$, giving:

$$q_{i,j,n+1} = q_{i,j,n} - \frac{1}{\Delta x_i \Delta y_j}\left(\widehat{f}_{i+\frac{1}{2},j} - \widehat{f}_{i-\frac{1}{2},j} + \widehat{g}_{i,j+\frac{1}{2}} - \widehat{g}_{i,j-\frac{1}{2}}\right) \tag{2}$$

$$\widehat{f}_{i-\frac{1}{2},j} = \widehat{f}\left(\mathbf{u}_{i-1,j,+\frac{1}{2}}, \mathbf{u}_{i,j,-\frac{1}{2}}, f_{i-1,j,+\frac{1}{2}}, f_{i,j,-\frac{1}{2}}\right) \tag{3}$$

$$\widehat{g}_{i,j-\frac{1}{2}} = \widehat{g}\left(\mathbf{v}_{i,j-1,+\frac{1}{2}}, \mathbf{v}_{i,j,-\frac{1}{2}}, g_{i,j-1,+\frac{1}{2}}, g_{i,j,-\frac{1}{2}}\right) \tag{4}$$

$$f_{i,j,\pm\frac{1}{2}} = \int\limits_{t_n}^{t_{n+1}} \int\limits_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f_{i,j,n}(x_{i\pm\frac{1}{2}}, y, t)\,dy\,dt \tag{5}$$

$$g_{i,j,\pm\frac{1}{2}} = \int\limits_{t_n}^{t_{n+1}} \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} g_{i,j,n}(x, y_{j\pm\frac{1}{2}}, t)\,dx\,dt \tag{6}$$

$$q_{i,j,n} = \frac{1}{\Delta x_i \Delta y_j} \int\limits_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(x, y, t_n)\,dx\,dy \tag{7}$$

where $f_{i,j,n}(x, y, t)$ and $g_{i,j,n}(x, y, t)$ are local space–time polynomials of the flux terms, f and g, valid over the space–time domain, $\Omega_{i,j} \times [t_n, t_{n+1}]$. Eq. (7) exists solely for initialization purposes and is not actually integrated during a simulation. Thus, computing (3)–(6) is enough to complete a single time step. Eqs. (3)–(4), and hence $\widehat{f}$ and $\widehat{g}$, are the numerical flux functions used to reconcile discontinuous estimates at the positive and negative limits to each cell face. Note that a fully discrete (i.e., single-stage, single-step) form has been assumed by casting the time integral directly. The only new global quantity added to an existing fluid model in this framework is one time-level of the tracer mass, $q_{i,j,n}$, which is overwritten at each new time step. This is in contrast to semi-discrete methods like Runge–Kutta and Adams–Bashforth ODE integrators, which require at least one (usually more) additional copy of q.

Regarding mass conservation, any finite-volume scheme on a periodic domain that relies a single flux vector between adjacent cells is automatically locally and globally mass conserving. This is true regardless of how the single flux vector is computed or the accuracy of the scheme. Therefore, the methods implemented herein are all exactly mass conserving down to machine precision effects (in this case, a relative mass change of order 10^{-15} or less). We used simple summations of cell mass across the global domain at the beginning and end of the simulation to compute these measures of relative mass conservation.

2.2. Tensor reconstruction

For an Nth-order accurate transport scheme, it is assumed that one already knows $\mathbf{u}_{i,j,n}(x,y,t)$, $\mathbf{v}_{i,j,n}(x,y,t)$, and $\rho_{i,j,n}(x,y,t)$ such that mixed space–time derivatives can be obtained, valid at the space–time point (x_i, y_j, t_n) of total order up to $(N-1)$ with the jth-order mixed derivative being at least $(N-j)$th-order accurate. If the driving fluid model uses ADER-DT, this is already known. However, if another integration method is used for the driving fluid model, then one must perform an ADER-DT procedure on $\mathbf{u}$, $\mathbf{v}$, and ρ starting at the correct time level.

From here, it is also assumed that $q_{i,j,n}\ \forall i,j$ are available, according to the definition (7). To obtain all of the necessary derivatives, a genuinely multi-dimensional spatial polynomial, $q_{i,j,n}(x,y)$, is reconstructed from a surrounding stencil of $N \times N$ values: $q_{k,l,n}$ where $k \in \{i-s,\dots,i+s\}$, $l \in \{j-s,\dots,j+s\}$, and $s=(N-1)/2$. For a symmetric stencil, it makes sense to choose schemes of odd-ordered accuracy. Therefore, 3rd-, 5th-, and 7th-order accurate schemes are implemented. Implementation of the reconstruction is important. A straightforward approach, multiplying a pre-computed $N^2 \times N^2$ Vandermonde inverse by the $N \times N$ stencil values leads to a computational complexity of N^6. A common improvement to this used in many multi-dimensional models is the tensor approach detailed in [17] wherein one computes $(A_y \otimes I)(I \otimes A_x)\phi \equiv A_x \phi A_y^\top$ where A_x and A_y are the 1-D Vandermonde inverses in the x and y directions, respectively, and ϕ is a vector containing the $N \times N$ stencil values in a single array with the x-dimension index varying the fastest. This computation is only N^3 complexity.

2.3. The ADER-DT time discretization

Now that the spatial derivatives are computed at time t_n, the ADER-DT process has all the necessary components for initialization. Given an initial tensor of spatial DTs for q, one can iterate over the DT of the PDE being solved as follows:

$$Q(k_x,k_y,k_t+1) = -\frac{k_x+1}{k_t+1}F(k_x+1,k_y,k_t) - \frac{k_y+1}{k_t+1}G(k_x,k_y+1,k_t) \tag{8}$$

$$\begin{aligned}F(k_x,k_y,k_t) = \frac{1}{R(0,0,0)}\sum_{r_t=0}^{k_t}\sum_{r_y=0}^{k_y}\sum_{r_x=0}^{k_x} & Q(r_x,r_y,r_t)\mathbf{U}(k_x-r_x,k_y-r_y,k_t-r_t) \\ & - R(r_x,r_y,r_t)F(k_x-r_x,k_y-r_y,k_t-r_t)\end{aligned} \tag{9}$$

$$\begin{aligned}G(k_x,k_y,k_t) = \frac{1}{R(0,0,0)}\sum_{r_t=0}^{k_t}\sum_{r_y=0}^{k_y}\sum_{r_x=0}^{k_x} & Q(r_x,r_y,r_t)\mathbf{V}(k_x-r_x,k_y-r_y,k_t-r_t) \\ & - R(r_x,r_y,r_t)G(k_x-r_x,k_y-r_y,k_t-r_t)\end{aligned} \tag{10}$$

where the capital letter of a variable is considered to be its DT (the capital letter of ρ being R). The indices k_x, k_y, and k_t represent the order of differentiation in the x, y, and time dimensions, respectively. Any terms in (9) and (10) that would cause self-dependence should be omitted from the summation. This can be done conveniently by simply setting the arrays for F and G to zero upon initialization. Then $F(0,0,0)$ and $G(0,0,0)$ should be computed and stored before looping begins, and overwriting $F(0,0,0)$ and $G(0,0,0)$ during looping should be avoided with an if statement. Formally, the only DTs that will be used are F and G, and Q is only computed in order to get F and G. F and G are identically the coefficients for the space–time Taylor polynomials $f_{i,j,n}(x,y,t)$ and $g_{i,j,n}(x,y,t)$ because the inverse DT is the Taylor expansion. The ADER-DT procedure is performed once per cell per time step, forming a Taylor series about the cell centroid and time step beginning, which is valid over the cell and time step in question.

For help implementing the ADER-DT procedure and for a more thorough explanation and derivation of various DTs, please consult [15,16] and references therein.

2.3.1. Differences between ADER, ADER-CG, and ADER-DT

While there are a number of differences between traditional ADER [18], ADER-CG [19,20], and the ADER-DT scheme presented herein regarding framework and implementation, this section focuses on the generation of time and space–time information. The original ADER scheme used a symbolic-based Cauchy–Kowalewski procedure to generate time and space–time derivatives. Practically, for a non-trivial PDE, symbolic mathematical software must be used to generate these complex expressions directly into code. These symbolic expressions can increase to exponential computational complexity with increasing order, making them difficult to use for arbitrarily high-order accuracy in practice.

The ADER-CG method represents a big efficiency improvement to ADER in general as it reduces the complexity for general PDEs from exponential to polynomial. The ADER-CG method initializes a set of spatial-only nodal basis functions, establishes space–time bases, and then non-linearly iterates the time and space–time basis coefficients using a space–time variational form of the PDE. These space–time bases are polynomials, which can then be integrated. In the sense that both are *iterated* internally within a cell in a single-stage manner, ADER-DT is quite similar in nature to ADER-CG, but there are also some differences. First, for each new order of accuracy, for each new PDE system, and for each new grid, the ADER-CG must establish either (1) the locations of a new set of space–time nodes, (2) a new set of non-linear algebraic iterations, or both.

For the ADER-DT method, there are no node locations to establish (no internal grid at all, in fact) but only a set of spatial derivatives valid at the cell centroid and time step beginning. This is usually accomplished with a matrix–vector product against the existing moments (bases, stencils, or some combination). Also, the summations and code remain constant no matter what order of accuracy is used. Order of accuracy in generating time and space–time derivatives is increased merely by changing one line of code: the loop bounds of the outer summation expressions. Also, once DTs of a PDE system are coded, they remain the same for any grid or order of accuracy. Coding the DTs for a new PDE can be cumbersome, but additional spatial dimensions only involve adding new indices and new loops. Because ADER-DT provides a straightforward high-order space–time Taylor expansion of each PDE term as the final result, it seems to be a particularly *flexible* option regarding order, mesh, PDE system, and spatial operator.

2.4. Half-tensor space–time evolution

Space–time derivatives of total order larger than $N-1$ have little effect on the total error because their error is much smaller than the formal truncation of the scheme. This has been confirmed in practice with numerical experiments using both a quadrature-free ADER-DT framework and a Runge–Kutta time integration with quadrature. Therefore, it makes the most sense to ignore derivatives of total order higher than $N-1$ by throwing them away. The only step wherein derivatives of total order higher than $N-1$ are computed is the reconstruction of Section 2.2. When the ADER-DT procedure is performed, space–time derivatives of order higher than the formal truncation of the overall scheme are not computed. This leads to a triangular loop structure in both the inner and outer loops of the ADER-DT procedure, which is addressed computationally in Section 2.6.

Taking the total number of loop iterations for F and G to be proportional to runtime cost, the cost of computing a full tensor of DTs up to order $N-1$ in each spatial dimension and in time is proportional to $(N+1)^3N^3/2^3$. Yet, if derivatives of total order larger than $N-1$ are discarded (i.e., using only a half-tensor of space–time derivatives), then runtime cost is now proportional to $(N+5)!/(6!(N-1)!)$. For $N=3$, $N=5$, and $N=7$, computing a half tensor instead of a full tensor saves a factor of 7.7, 16.0, and 23.8, respectively, for the ADER-DT procedure. Also, when computing only a half-tensor of derivatives, the analytical integration of the flux space–time polynomials at cell boundaries is also significantly cheaper because there are many fewer terms in the space–time polynomials. A full tensor of derivatives has N^3 terms in the space–time polynomial, while the half-tensor only has $(N+2)!/(3!(N-1)!)$ terms.

It is infeasible for memory requirements and for caching efficiency to store the full N^3 tensor of data for q, $\mathbf{u}$, $\mathbf{v}$, and ρ. Therefore, the half-tensor is actually stored in a single-dimension array with the x-dimension varying the fastest and the time dimension varying the slowest. To efficiently index this single-dimension array, a small integer arithmetic function is used to map the x, y, and time indices to a single index. This function changes only changes if the order of accuracy changes. Algorithmically, once a full tensor of derivatives is produced by tensor-based reconstruction, those values are passed through a routine that immediately condenses them into the smaller, half-tensor-sized, single-dimension array. This storage condensing routine consumes very little runtime compared to the other routines, and its looping overhead is significantly reduced by the blocking technique of Section 2.6.

2.5. Analytical, quadrature-free integration

As a result of the ADER-DT procedure, $f_{i,j,n}(x,y,t)$ and $g_{i,j,n}(x,y,t)$ for all i and j are obtained as Nth-order accurate space–time polynomials valid over the space–time domain: $\Omega_{i,j} \times [t_n, t_{n+t}]$. Therefore, quadrature is not necessary, and analytical space–time integration can be performed. For each cell, after the fluxes are expanded as space–time polynomials, (5) and (6) are computed analytically with a single linear combination of polynomial coefficients. The linear combination of coefficients for space–time integration is precomputed because it is the same for each cell, and Sage symbolic mathematical software is used to compute these linear combinations and translate them directly to Fortran 90 code.

2.6. Blocking the reconstruction, ADER-DT, and integration

The ADER-DT procedure is the most expensive single portion of this algorithm, and the looping overhead is quite large compared to the small amount of computation performed at the innermost loop level. Because of this, a blocking option is implemented to improve the amount of Single Instruction Multiple Data (SIMD) level parallelism for improved on-chip performance. A small local array is created, which holds a block of $b_x \times b_y$ cells, and the fastest varying index loops over these indices. The stencils for this block are stored redundantly so that the reconstruction, ADER-DT, and integration procedures can all be performed simultaneously. This increases the memory usage locally, and it requires a packing procedure, but the packing takes up very little time relative to the other operations, and memory increase is limited to the size of the block.

Using this blocking has two benefits. First, it enables SIMD operations to work on Streaming SIMD Extension (SSE) enabled processors. Second, the looping overhead becomes significantly smaller in the ADER-DT procedure because relatively more computation is done at the inner-most level. For the present 2-D transport problems, a block size of 64 was found to be the most effective with 64-bit precision on an AMD bulldozer processor, but the user can tune this to their own processor. Ideally, if the problem size is large enough, one would set $b_x = 64$ and $b_y = 1$, which has the x-dimension varying the fastest. However, there is flexibility to keep the block size at 64 even when the number of cells is less than that in the x-dimensions. Also, it is wise to constrain the number of cells in each dimension to an integer multiple of the block size in that dimension so that no loop iterations are wasted. Since the upwind numerical flux and the update of $q_{i,j,n+1}$ are significantly less expensive than reconstruction, ADER-DT, and space–time integration of face fluxes, those portions are not blocked.

In the event of using a highly threaded acceleration device like Intel's MIC or an Nvidia or ATI GPU, the size of the block may be substantially increased at the expense of repeating the stencils in local memory. Doing this, the reconstruction, ADER-DT, and integration procedures can provide ample data-independent threading.

2.7. Time-averaged upwind numerical flux

An upwind numerical flux is used at cell faces to reconcile discontinuous time-averaged flux limits at either side of an interface. To perform the upwinding, space–time integrations of the momenta at cell faces are computed:

$$\mathbf{u}_{i,j,\pm\frac{1}{2}} = \int_{t_n}^{t_{n+1}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \mathbf{u}_{i,j,n}(x_{i\pm\frac{1}{2}}, y, t) dy dt \tag{11}$$

$$\mathbf{v}_{i,j,\pm\frac{1}{2}} = \int_{t_n}^{t_{n+1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{v}_{i,j,n}(x, y_{j\pm\frac{1}{2}}, t) dx dt \tag{12}$$

If more than one tracer is evolved, this wind averaging is done only once for all tracers per time step. Because density is a positive quantity and cannot change the sign of the wind, the momenta are used directly rather than dividing by density. The time-averaged upwind Riemann solver is defined as:

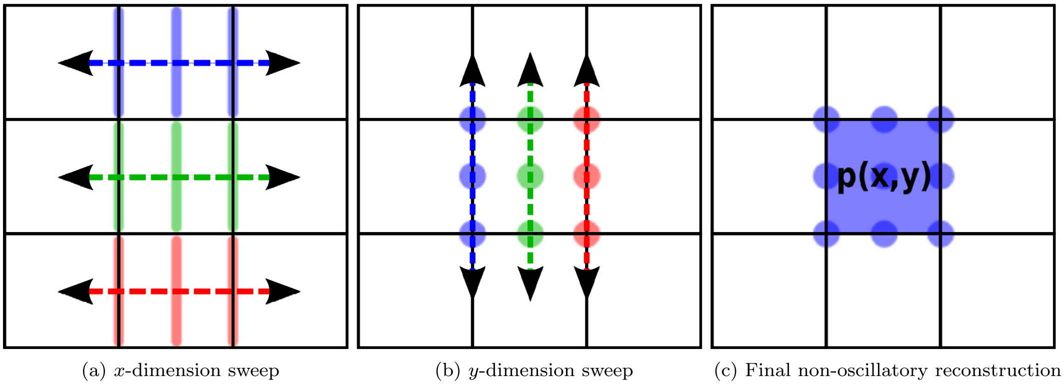
$$\widehat{f}\left(\mathbf{u}^-, \mathbf{u}^+, f^-, f^+\right) = \begin{cases} f^- & \text{if } \mathbf{u}^- + \mathbf{u}^+ > 0 \\ f^+ & \text{otherwise} \end{cases}$$

$$\widehat{g}\left(\mathbf{v}^-, \mathbf{v}^+, g^-, g^+\right) = \begin{cases} g^- & \text{if } \mathbf{v}^- + \mathbf{v}^+ > 0 \\ g^+ & \text{otherwise} \end{cases}$$

where the function inputs match the inputs of (3)–(4). Only one Riemann solver is applied per interface per time step per tracer. Also, for analytical wind values, there is no merit in using interface limits since they will be the same. However, when coupling to a general fluid model, especially one with limiters like WENO, these limits will not generally be the same. Since this codebase is to be coupled to a fluid model, the more general form is used to best understand cost in a real application.

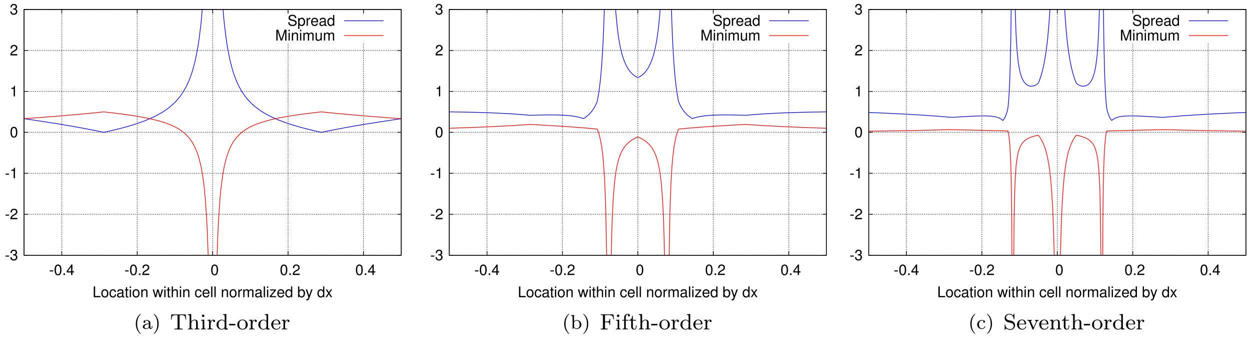
2.8. WENO limiting

Unlike most schemes that use WENO limiting, this scheme needs full multi-dimensional polynomials of Nth-order accuracy, not just point values at cell boundaries. Since WENO does not form these high-order polynomials directly, short of using an extremely high-order-accurate $(2N-1)$th-order method, this information is not immediately available. This study uses a slight tweak on the innovative sub-cell WENO technique of [21]. The implementation here will be described in detail since some simplifications are used and the computation of point values is slightly different when done in multiple dimensions. The WENO technique is applied in a series of 1-D sweeps to accomplish genuinely multi-dimensional limiting. As diagramed in Fig. 1(a), starting with the x-direction, for each row of cells for the stencil in question (i.e., $l \in [-s, s]$ where $s = (N-1)/2$ is the so-called "halo" size), a 1-D WENO procedure is performed using the stencil $q_{i+k,j+l,n}$ where $k \in [-s, s]$, to compute non-oscillatory line averages at a series of x-locations: $x_{i,\kappa} = x_{i-\frac{1}{2}} + \frac{\kappa}{N-1} \Delta x_i$ where $\kappa \in \{0, \ldots, N-1\} \backslash \{\frac{N-1}{2}\}$. These line averages are defined as:



(a) x-dimension sweep (b) y-dimension sweep (c) Final non-oscillatory reconstruction

Fig. 1. Schematics of the dimension-by-dimension sub-cell WENO procedure adapted from [21] for up to third-order accuracy.



(a) Third-order (b) Fifth-order (c) Seventh-order

Fig. 2. Plots of the spread of optimal WENO weights and the minimum optimal WENO weight as a function of the sampling location within a cell.

$$q_{i,j+l,n,\kappa} \equiv \frac{1}{\Delta y_j} \int\limits_{y_{j+l-\frac{1}{2}}}^{y_{j+l+\frac{1}{2}}} q(x_{i,\kappa}, y, t_n) dy \tag{13}$$

The relation (13) is not something explicitly integrated but rather defines what is obtained after the x-dimension 1-D WENO sweeps.

Typically, a WENO scheme is well-behaved at the cell boundaries, and this is where most point-wise WENO schemes compute the optimal weights. However, every WENO scheme (defined uniquely by the choice of lower-ordered and optimal-order polynomials) exhibits different behavior when computing optimal weights for general points *within* the cell being reconstructed. Fig. 2 shows plots of the spread (maximum minus minimum) of optimal weights along with the minimum optimal weight for third-, fifth-, and seventh-order accuracy. Each scheme has different points of singularity when computing optimal weights. For the third-, fifth-, and seventh-order schemes, uniform spacing of points (except the cell center) happens to land in well-behaved regions, but the reader should be careful of assuming this in general. Also, while the weights are fairly well behaved at the cell center for fifth-order accuracy, negative weights emerge that require treatment according to [22]. Thus, this study avoids the center point as with the third- and seventh-order schemes in order to both keep the WENO procedure consistent and avoid needing to treat negative weights. To obtain a line-averaged value at the cell center location, x_i, for each 1-D sweep, a polynomial is constrained by the other WENO-derived line averages and by the cell mean (which is inherently non-oscillatory) and then sampled at x_i. In practice, this is all precomputed such that the center value is simply a linear combination of the other point values and the cell mean.

Note that most of the computation lies in computing smoothness indicators and non-oscillatory weights, and the same weights are reused for each point x_k. Therefore, computing multiple point values does not significantly increase WENO expense. Also, the cost of computing WENO weights is further mitigated by computing these weights only once per direction. A column-averaged x-direction stencil is formed from the original stencil values:

$$\bar{q}_{i+k,j,n} = \frac{1}{N} \sum_{l=-s}^{s} q_{i+k,j+l,n} \tag{14}$$

Then, limited weight are computed only once per direction based on these averaged values with a similar treatment in the y-direction. This significantly reduces the cost of multi-dimensional WENO, and it does not appreciably affect the results, even for solid body rotation and deformational flows. The weight mapping technique of [23] is also used here to improve convergence to optimal weights within smooth regions of flow. Because this is performed only once per direction per time step, its cost is negligible. Also, by locally saving the original weights before mapping, one can perform a continuous transition in weight mapping for any positive real number. This gives the user a continuous tuning parameter for determining the smoothness of their results to match their application. Also, the exponent applied to smoothness indicators when computing limited weights can be any positive real number to provide yet another continuous tunable parameter for smoothness versus accuracy. The larger the smoothness indicator exponent and the smaller the weight mapping coefficient, the smoother the results.

Next, as diagramed in Fig. 1(b), for each column of line averages, a series of y-direction WENO procedures is performed in the same manner as the x-dimension sweep in order to obtain a 2-D grid of non-oscillatory point values located inside $\Omega_{i,j}$:

$$q_{i,j,n,\kappa,\lambda} \equiv q(x_{i,\kappa}, y_{j,\lambda}, t_n) \tag{15}$$

where $y_{j,\lambda} = y_{j-\frac{1}{2}} + \frac{\lambda}{N-1}\Delta y_j$ where $\lambda \in \{0, \ldots, N-1\}\backslash\{\frac{N-1}{2}\}$. As in the x-dimension sweep, the center point value at y_j is computed as a linear combination of the limited point values directly computed and the center line average of each stencil. Then, instead of performing tensor reconstruction based on an $N \times N$ stencil of cell averages, a tensor reconstruction based on the $N \times N$ grid of point values within the center cell is used as diagrammed in Fig. 1(c).

Each 1-D WENO procedure is of order N^2 complexity, and N WENO procedures are performed per sweep. Thus, the total complexity of 2-D WENO limiting is order N^3, which is the same as the tensor reconstruction operator. In practice, this technique is easy to extend to more spatial dimensions, and the total complexity of the overall WENO procedure is N^{D+1}, where D is the number of spatial dimensions, again matching the tensor reconstruction complexity. The blocking strategy is used here as well for improved on-chip performance.

When using a fully discrete time integration, there is an advantage when limiting with WENO in that the expense is only incurred once to limit the entire time step. Also, since the ADER-DT method analytically computes time derivatives from spatial derivatives using the PDE, if the underlying spatial derivatives are non-oscillatory, then so are the resulting time derivatives. ADER methods have shown this property in other studies as well [21,24].

2.9. FCT-enforced positivity

Non-oscillatory limiting is generally considered to be a strong requirement for most transport codes, especially when chemistry reactions are involved. However, another strong requirement for most transport codes is that the tracer mass not be allowed to become negative. There are post-hoc fixers for this, but local hole fillers are not guaranteed to succeed, depending on the size of the negativity violation, and they require additional parallel data transfers. Additionally, global hole fillers require global parallel data transfer, which is prohibitive in large parallel environments. Since non-oscillatory limiters are not strictly monotone, strict positivity is not enforced by WENO alone. The negative values are small and tunable, but they happen. Therefore, an FCT-based positivity filter is employed for strict positivity [25]. Define the outward-oriented flux divergence in cell $\Omega_{i,j}$ as:

$$\mathcal{F}_{i,j,out} = \frac{1}{\Delta x_i \Delta y_j}\left(\max(\widehat{f}_{i+\frac{1}{2},j}, 0) - \min(\widehat{f}_{i-\frac{1}{2},j}, 0) + \max(\widehat{g}_{i,j+\frac{1}{2}}, 0) - \min(\widehat{g}_{i,j-\frac{1}{2}}, 0)\right) \tag{16}$$

To keep $q_{i,j,n+1}$ positive, it just needs to be ensured that the mass leaving the cell is not larger than the mass in the cell: i.e., $q_{i,j,n} - \mathcal{F}_{i,j,out} \geq 0$. To do this, define a flux reduction factor $\phi_{i,j} = q_{i,j,n}/(\mathcal{F}_{i,j,out} + \epsilon)$ for each cell, where ϵ is a small number to avoid division by zero. Then, for each cell face, apply the upwind cell's flux reduction factor to that face's flux. After this, the update is unable to produce negative values. Note that this does not require an additional parallel data transfer as is the case with traditional FCT [26]. Also, this procedure can be easily adapted to apply an upper bound, for example if a tracer cannot exceed unity. If the negative values being disallowed by FCT are relatively large, this procedure can cause serious numerical artifacts to form. However, after applying a WENO limiter, the FCT corrections are quite small. Therefore, numerical artifacts are also small. After applying WENO and FCT, the present transport algorithm provides a robust, positive-definite solution with less damping than strictly monotone limiters.

2.10. Stability

To assess the numerical stability of the new methods in multiple dimensions, define a spatial harmonic function:

$$q_h(x, y) = e^{i(\frac{\kappa_x \pi}{\Delta x}x + \frac{\kappa_y \pi}{\Delta y}y)} \tag{17}$$

where the grid will admit wave numbers such that $k_x, k_y \in [0, 1]$. This harmonic function is used as an input to the previously described ADER-DT method (except for the non-linear WENO and FCT filters) applied to $\partial_t q_h + \partial_x q_h + \partial_y q_h = 0$. A stencil is initialized with $q_{h,i,j,n} = \frac{1}{\Delta x_i \Delta y_j}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q_h(x, y)dxdy$, where $i, j \in [-s-1, s+1]$ for $s = (N-1)/2$, and this stencil is used to apply the ADER-DT algorithm to the stencil's center cell averaged value. After applying the ADER-DT FV

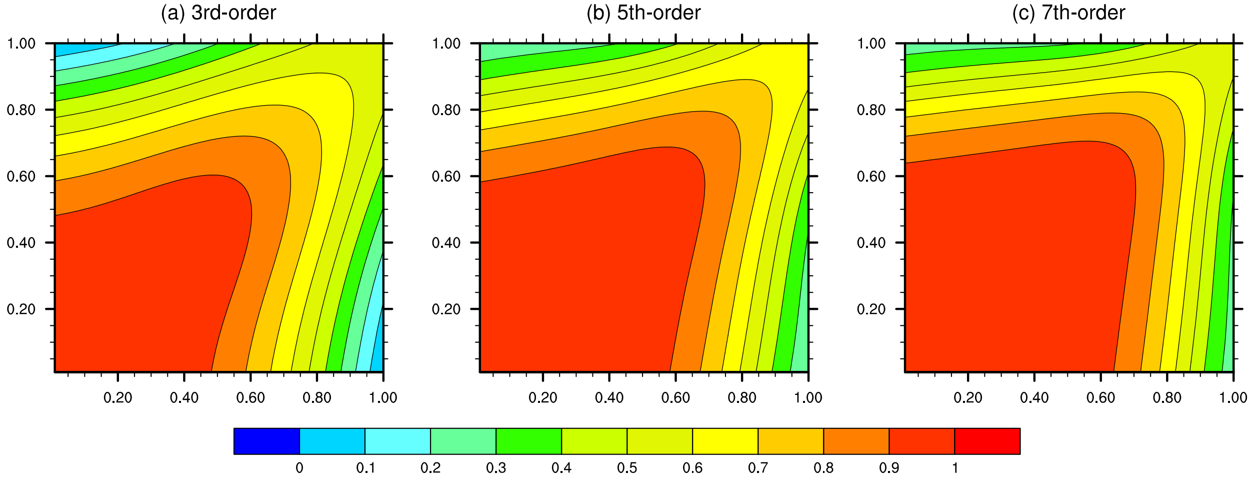


Fig. 3. Amplification factor for ADER-DT at the maximum stable CFL value. x-axis and y-axis are κ_x and κ_y, respectively, from (17).

algorithm, the amplification factor is defined as $A = q_{h,0,0,n+1}/q_{h,0,0,n}$. When $|A| \le 1$ for all k_x and k_y, the scheme is stable using that time step. Defining the CFL value as $u_{max}\Delta t/\Delta x$ (where $\Delta x = \Delta y$) and using a bisection search over time step, the maximum stable CFL value is calculated for the ADER-DT scheme at third-, fifth-, and seventh-order accuracies to be 0.500, 0.602, and 0.607, respectively. These numbers were the same when computed in 64-bit and 128-bit floating point precision. As expected, resolution power is visibly increased with increasing order as shown by Fig. 3.

2.11. Parallel efficiency of WENO-limited ADER-DT

There are aspects of this framework that lend themselves well to large parallel simulations wherein parallel data transfer costs are very high compared to computation within a node. First, the ADER-DT time discretization is single stage. This means there are no required data transfers within a time step no matter how high-order the accuracy is in time. Data is transferred only once at the time step beginning. With multi-stage time integrators such as Runge–Kutta, parallel data transfers are required in between each stage. Therefore, ADER-DT is communication-avoiding compared to multi-stage methods.

Also, regarding limiting, the use of WENO also has advantages compared to post-hoc limiting such as traditional FCT and/or hyperdiffusion. Each application of hyperdiffusion requires an additional data exchange beforehand, and for higher-order hyperdiffusion, either the halo size must increase (for FV methods) or the number of parallel transfers must increase (for Galerkin methods). For FCT in its traditional form (i.e., not the positivity-only form), one typically must first apply a hyperdiffusive operator to avoid terracing artifacts, and then one must solve for both a high-order and low-order solution, each of which require separate data transfer.

For WENO, this limiting is performed beforehand so that as the integration is performed, there are no oscillations, and it does not require an additional parallel data transfer. The reason is that an Nth-order accurate method resorts to the use of off-centered $((N+1)/2)$th-order-accurate polynomials in the presence of a discontinuity, which successfully limits oscillations without requiring additional parallel data transfers. Smooth regions of the flow are not damped by the WENO limiting and remain at high-order accuracy. With WENO and the positivity-only version of FCT, there is only one parallel data transfer required at the beginning of a time step with no other transfers required during the time step. Therefore, with one parallel data exchange, a full, arbitrarily high-order-accurate, limited, and positive-definite time step can be performed. This is advantageous in a large parallel computing environment.

3. Numerical experiments

For all tests except the consistency check, the density field will be $\rho(x, y, t) = 1$. Boundaries are periodic, and the spatial domain is $[0, 1] \times [0, 1]$. For the winds in the ADER-DT method, the space–time DTs for $\mathbf{u}$ and $\mathbf{v}$ are computed analytically at the beginning of each time step. Error norms are computed as:

$$L_1 = \frac{1}{n_x n_y} \sum_{i,j} |q_{a,i,j} - q_{n,i,j}|$$

$$L_2 = \sqrt{\frac{1}{n_x n_y} \sum_{i,j} (q_{a,i,j} - q_{n,i,j})^2}$$

Table 1
ADER-DT error norms and convergence rates for uniform advection of a 2-D sine wave function. CFL = 0.45 in all tests.

Order	# Cells	L_1	Conv.	L_2	Conv.	L_∞	Conv.
3	64	1.18E−04		1.32E−04		1.37E−04	
	128	1.48E−05	−3.00	1.65E−05	−3.00	1.71E−05	−3.00
	256	1.85E−06	−3.00	2.06E−06	−3.00	2.14E−06	−3.00
5	64	3.39E−07		3.82E−07		4.36E−07	
	128	1.06E−08	−5.00	1.20E−08	−5.00	1.36E−08	−5.00
	256	3.32E−10	−5.00	3.74E−10	−5.00	4.26E−10	−5.00
7	64	6.62E−10		7.47E−10		8.57E−10	
	128	5.19E−12	−7.00	5.85E−12	−7.00	6.71E−12	−7.00
	256	4.34E−14	−6.90	5.13E−14	−6.83	6.84E−14	−6.62

$$L_\infty = \max_{i,j} |q_{a,i,j} - q_{n,i,j}|$$

where q_a and q_n are the analytical and numerical solutions, respectively, and n_x and n_y are the number of grid points in the x- and y-directions, respectively.

3.1. *Uniform advection*

3.1.1. *Runge–Kutta quadrature-based method for comparison*

A Runge–Kutta (RK), quadrature-based method is included for this test to provide some comparison point for the ADER-DT method. A simple, low-storage, N-stage, Nth-order-accurate RK method described by Eq. (2.45) in [25] is used for a spatially Nth-order-accurate method. The same blocking, the same reconstruction, the same sub-cell WENO procedure, and the same half-tensor storage as the ADER-DT method is used, but fluxes are calculated for each stage of the RK method using spatial quadrature at the cell faces. For an Nth-order-accurate overall method, an $(N+1)/2$-point Gauss–Legendre quadrature rule is used to ensure fluxes are integrated at a sufficient order of accuracy to maintain the scheme's overall formal order of accuracy. Since density and winds are identically unity in space and time for this test case, analytical computation of these values does not come into play in this comparison. The author's best attempt was made to ensure equal efficiency in both RK and ADER-DT codebases, which both utilized 8-way threading and blocking. In fact, the blocking proved more beneficial for RK than for ADER-DT compared to no blocking. Each codebase shares much of the same code, including reconstruction, WENO-limiting, reconstruction condensing, and flux upwinding.

3.1.2. *Results*

A 2-D sine wave defined by $\psi(x,y,0) = [\sin(2\pi x)\sin(2\pi y)+1]/2$ is advected with a spatially uniform and temporally constant velocity of $u = v = 1\ \mathrm{m\,s^{-1}}$. Each model second represents one translation over the domain. Since $\mathbf{u}(x,y,t) = \mathbf{v}(x,y,t) = \rho(x,y,t) = 1$, the fluxes are analytically integrated in space for both the ADER-DT and RK models, meaning that the only appreciable source of difference in accuracy is due to the time stepping procedure.

Table 1 provides the error norms for the ADER-DT method undergoing cell refinement at third-, fifth-, and seventh-order accuracies to demonstrate expected convergence rates. The seventh-order refinement reaches machine precision limits before the full convergence at 256×256 cells. When computed with WENO and FCT enabled, the errors differed from default simulation by a relative magnitude of 10^{-6} or less, which is expected for smooth flows. Therefore, these numbers are not presented verbatim to avoid redundance. The WENO + FCT simulations did reach machine precision limits at numbers slightly larger than the default simulations, indicating slightly more cancellation in the seventh-order WENO procedure. This will not affect real simulations, which exhibit errors many orders of magnitude larger than machine precision.

Table 2 provides error norms for the RK time discretizations with spatial quadrature, and all differences in error (apart from machine precision effects) should be due to the time discretization because the spatial polynomials are integrated analytically in each case. At third-order accuracy, ADER-DT consistently produced 1.7× less error than RK in the L_1 norm and 3.4× less error than RK in the L_∞ norm. At fifth- and seventh-order accuracies, ADER-DT was 1.2× lower in error the L_1 norm and 2.3× lower in error in the L_∞ norm. The improvements in temporal discretization error are the greatest for the largest model errors, which are the peaks and crests of the 2-D sine wave in this case.

For runtimes, CPU time and wall time timers were used, and they differed by less than 5% in all cases (see Table 3). Thus, only wall times are presented to avoid redundancy. 8-way threading was used with an OpenMP implementation with the previously mentioned blocking for cache efficiency, SIMD utilization, and reduced looping overheads for all schemes on an AMD bulldozer chip. The paired cores (i.e., 16-way threading) were not used so as to reduce thread resource contention and particularly runtime variability, for a more robust and consistent comparison between RK and ADER-DT runtimes.

Without applying WENO and FCT limiting, the ADER-DT runtime improvements over RK reduce with increasing order due to the increasing cost of the ADER-DT procedure itself, which has a computational complexity of N^6 for an Nth-order-accurate scheme in two spatial dimensions plus time. With limiting, however, since ADER-DT requires only one WENO limiting per time step, it becomes increasingly cheaper in comparison to RK as order increases. Also notable is that WENO-limiting becomes relatively less expensive for ADER-DT as order increases because (1) it is only required once per time step

Table 2
RK with quadrature error norms and convergence rates for uniform advection of a 2-D sine wave function. CFL = 0.45 in all tests.

Order	# Cells	L_1	Conv.	L_2	Conv.	L_∞	Conv.
3	64	1.96E−04		2.35E−04		4.60E−04	
	128	2.45E−05	−3.00	2.94E−05	−3.00	5.77E−05	−3.00
	256	3.07E−06	−3.00	3.68E−06	−3.00	7.22E−06	−3.00
5	64	4.03E−07		4.97E−07		9.95E−07	
	128	1.26E−08	−5.00	1.56E−08	−5.00	3.11E−08	−5.00
	256	3.95E−10	−5.00	4.86E−10	−5.00	9.73E−10	−5.00
7	64	7.93E−10		9.78E−10		1.96E−09	
	128	6.21E−12	−7.00	7.66E−12	−7.00	1.53E−11	−7.00
	256	5.22E−14	−6.90	6.40E−14	−6.90	1.30E−13	−6.88

Table 3
Wall times in seconds for one model second of uniform transport of a 2-D wine wave with 256 cells with and without WENO + FCT limiting for ADER-DT and RK time discretizations. “Lim” denotes use of WENO and FCT limiting.

Order	ADER		RK		(RK)/(ADER)		(Lim)/(Default)	
	Default	Lim	Default	Lim	Default	Lim	ADER	RK
3	2.36	5.22	4.83	13.29	2.04×	2.55×	2.21×	2.75×
5	11.13	18.09	18.66	52.94	1.68×	2.93×	1.63×	2.84×
7	36.60	54.41	50.53	172.87	1.38×	3.18×	1.49×	3.42×

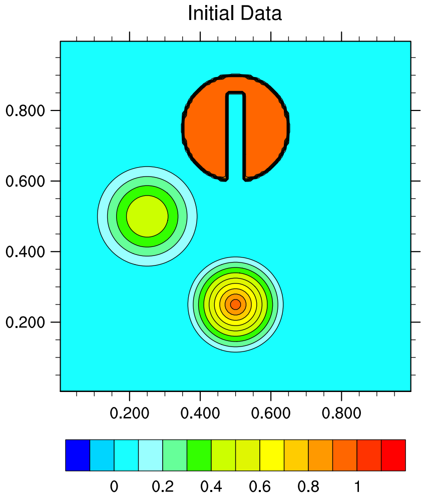


Fig. 4. Plot of initial data for SBR tests.

and (2) the ADER-DT procedure becomes increasingly expensive with increasing order. It becomes relatively more expensive for RK with increasing order because the constant of computational complexity is larger for WENO than it is for 2-D spatial reconstruction. Thus, the biggest utility of ADER-DT efficiency is realized in a limited context.

This is important because WENO is regarded by many as being prohibitively expensive. But if the overhead of WENO in the seventh-order accurate case is only 50%, and if it doesn't require any additional parallel communications, then one could argue it is advantageous to other methods of limiting such as FCT and hyperdiffusion, which incur the same or greater overhead and require additional parallel communications.

3.2. Solid body rotation (SBR)

The SBR tests are used to perform a comprehensive analysis of the effectiveness of WENO limiting within this ADER-DT quadrature-free FV framework at varying orders of accuracy. For solid body rotation, the winds are initialized as $u = -\omega y$ and $v = \omega x$, where $\omega = 2\pi$. The tracer field is initialized, as shown in Fig. 4, with a linear cone (ψ_1), a cosine bell (ψ_2), and a slotted cylinder (ψ_3). They are defined as follows:

$$\psi_1(x, y, 0) = \max(1 - r_1, 0)$$

$$\psi_2(x, y, 0) = \begin{cases} \frac{1}{4}(\cos(\pi r_2) + 1) & \text{if } r_2 \le 1 \\ 0 & \text{otherwise} \end{cases}$$

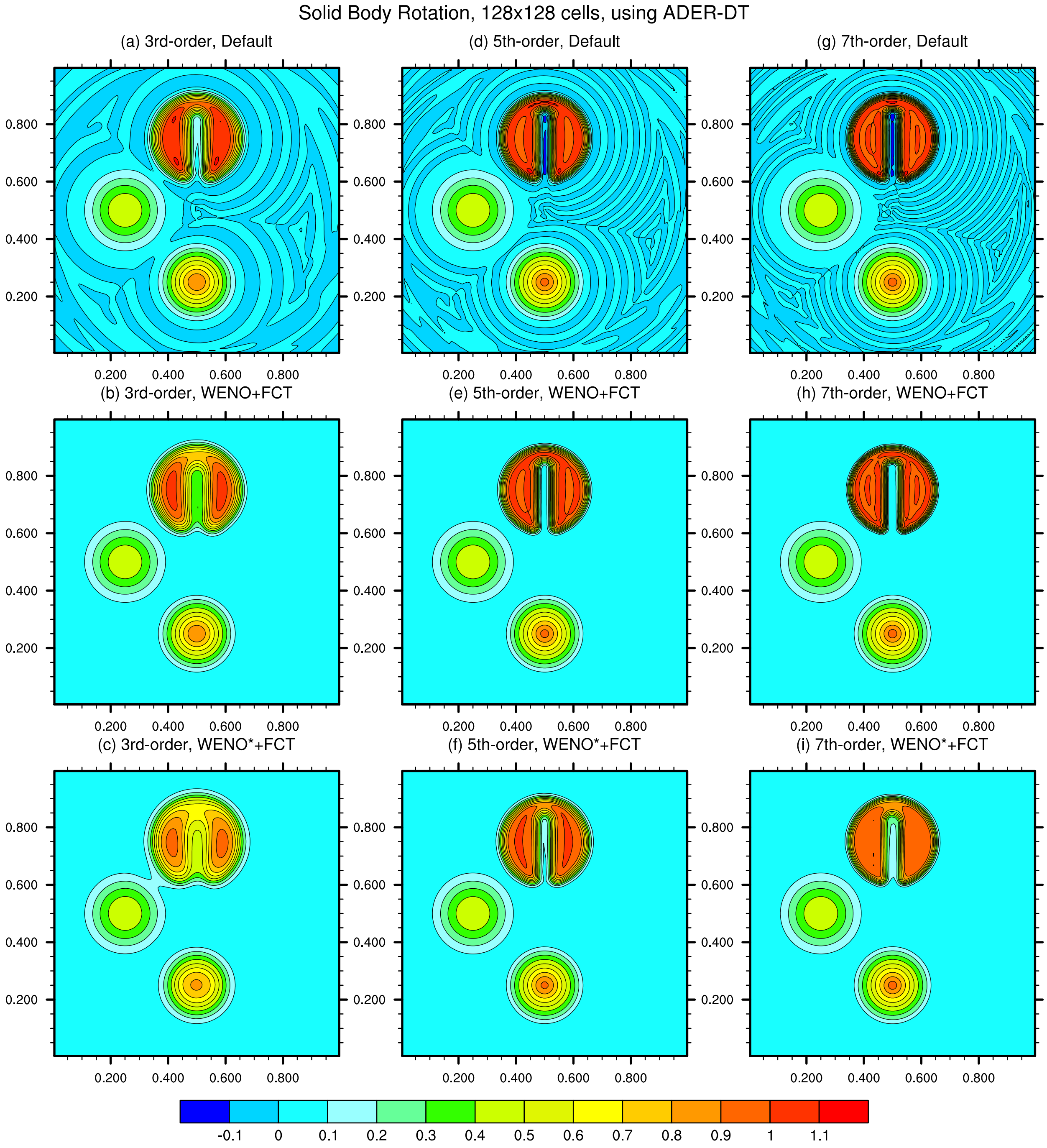


Fig. 5. Plots after one SBR revolution (1 s) with 128 × 128 cells and CFL = 0.45.

$$\psi_3(x,y,0)=\begin{cases}1 & \text{if } r_3\le 1 \text{ and } |x-x_{C,3}|\ge \frac{x_{R,3}}{6} \text{ or } y-y_{C,3}\ge \frac{y_{R,3}}{1.5}\\ 0 & \text{otherwise}\end{cases}$$

$$r_i=\sqrt{\left(\frac{x-x_{C,i}}{x_{R,i}}\right)^2+\left(\frac{y-y_{C,i}}{y_{R,i}}\right)^2},\quad i\in\{1,2,3\}$$

where $x_{C,1}=0.50$, $y_{C,1}=0.25$, $x_{R,1}=y_{R,1}=0.15$, $x_{C,2}=0.25$, $y_{C,2}=0.50$, $x_{R,2}=y_{R,2}=0.2$, $x_{C,3}=0.50$, $y_{C,3}=0.75$, and $x_{R,3}=y_{R,3}=0.15$. Each model second represents a full revolution around the domain.

In Fig. 5, results are plotted after one revolution (one model second) using the default scheme, a WENO scheme with a smoothness indicator exponent of two and one application of weight mapping (this is considered the default WENO

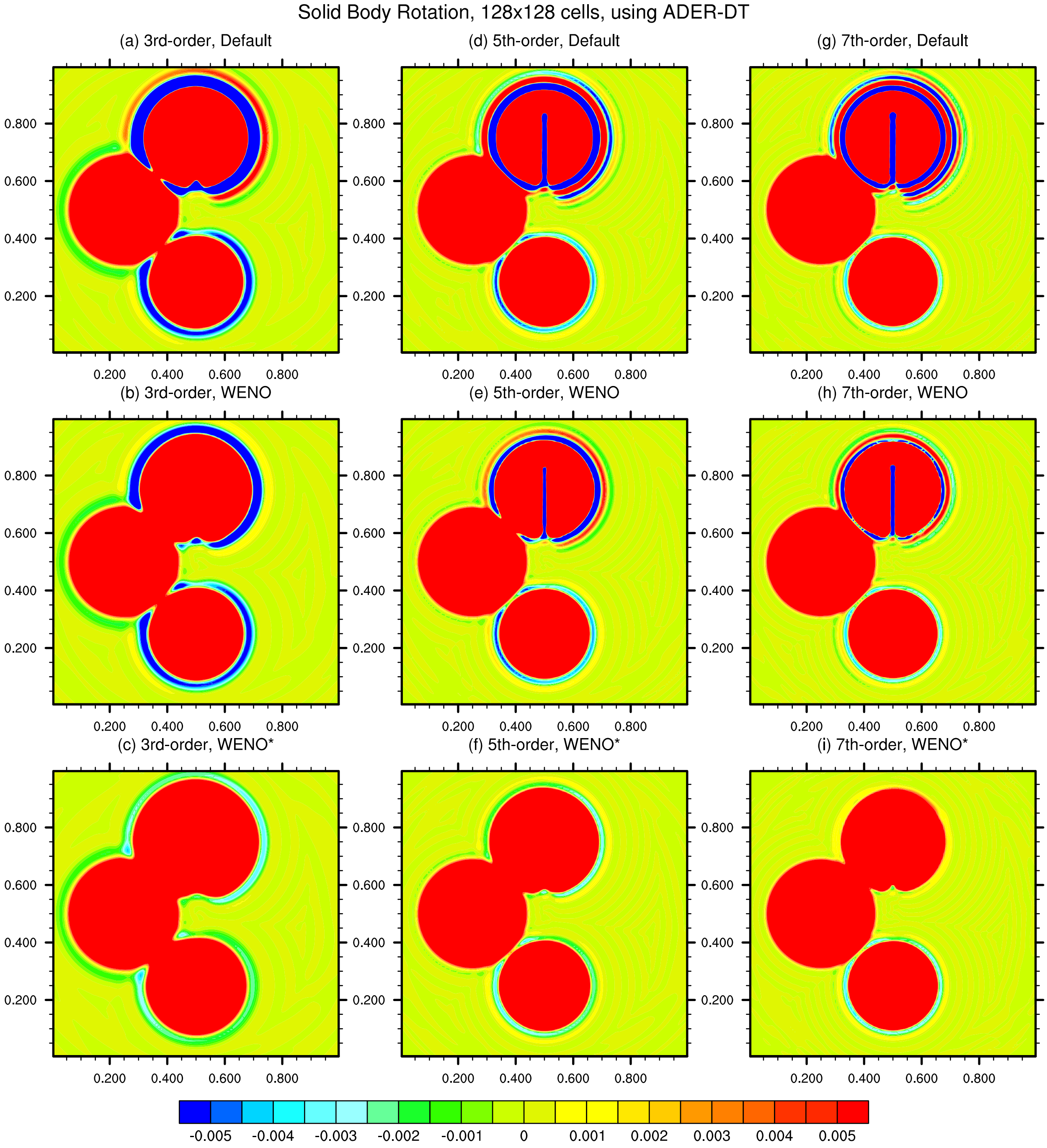


Fig. 6. Same as Fig. 5, except the color bar is changed to highlight oscillations to show the effect of WENO limiting without the positivity filter.

scheme), and a "WENO*" scheme that uses a smoothness indicator exponent of four and no weight mapping to represent a much smoother version of WENO. Visually, one can see the progression from very oscillatory to less and less oscillatory. The WENO* scheme demonstrates the fact that in raising the exponent of smoothness indicators, one approaches the ENO scheme wherein only one of the stencils is used each time. The WENO* scheme is highly diffusive, and overshoots are reduced to order 10^{-3} relative to the size of the discontinuity in question.

For Fig. 6, the FCT positivity filter is removed and the color bar is zoomed in toward small magnitudes about the zero contour to visually show oscillations, particularly around the slotted cylinder. Here, one can see how WENO reduces the radius of noise contamination around the slotted cylinder, and how WENO* quite effectively eliminates the noise, demon-

Table 4
Error norms using 128 × 128 cells after 1 revolution of SBR with all of the data and each shape individually as well. "WENO*" denotes WENO run with a smoothness indicator exponent of 4 and no weight mapping.

Order	Data	Default			WENO + FCT			WENO* + FCT		
		L_1	L_2	L_∞	L_1	L_2	L_∞	L_1	L_2	L_∞
3	All	1.84E−02	6.33E−02	5.70E−01	2.15E−02	7.72E−02	6.57E−01	3.12E−02	9.62E−02	6.77E−01
	ψ_1	8.71E−04	3.39E−03	6.90E−02	6.27E−04	2.98E−03	7.54E−02	1.75E−03	7.00E−03	1.48E−01
	ψ_2	1.69E−04	4.84E−04	5.70E−03	5.47E−05	1.83E−04	2.05E−03	2.81E−04	8.48E−04	1.66E−02
	ψ_3	1.74E−02	6.32E−02	5.70E−01	2.07E−02	7.71E−02	6.57E−01	2.91E−02	9.58E−02	6.77E−01
5	All	1.21E−02	4.72E−02	4.47E−01	1.11E−02	5.06E−02	4.74E−01	1.55E−02	6.16E−02	5.41E−01
	ψ_1	3.61E−04	1.47E−03	2.90E−02	2.82E−04	1.34E−03	2.92E−02	3.96E−04	1.76E−03	3.99E−02
	ψ_2	3.09E−05	1.06E−04	1.46E−03	1.75E−05	6.99E−05	1.12E−03	3.12E−05	1.06E−04	1.48E−03
	ψ_3	1.18E−02	4.72E−02	4.47E−01	1.08E−02	5.05E−02	4.74E−01	1.51E−02	6.15E−02	5.41E−01
7	All	9.69E−03	3.85E−02	3.80E−01	8.19E−03	3.99E−02	3.97E−01	1.23E−02	5.26E−02	5.41E−01
	ψ_1	2.31E−04	9.45E−04	1.66E−02	2.07E−04	9.85E−04	1.66E−02	2.46E−04	1.09E−03	2.29E−02
	ψ_2	1.51E−05	5.21E−05	7.77E−04	1.31E−05	5.48E−05	1.02E−03	1.53E−05	5.26E−05	7.89E−04
	ψ_3	9.46E−03	3.85E−02	3.80E−01	7.90E−03	3.98E−02	3.97E−01	1.21E−02	5.26E−02	5.41E−01

Table 5
Global minimum and maximum values using 128 × 128 cells after 1 revolution of SBR evolving all data. "WENO*" denotes WENO run with a smoothness indicator exponent of 4 and no weight mapping.

Order	Data	Default	WENO	WENO + FCT	WENO*
3	min	−0.059	−0.013	0.000	−0.004
	max	1.114	1.033	1.033	1.000
5	min	−0.138	−0.062	0.000	−0.004
	max	1.142	1.056	1.058	1.007
7	min	−0.188	−0.101	0.000	−0.005
	max	1.137	1.059	1.060	1.000

strating that ADER-DT with WENO limiting is capable of producing results that are close to shape preserving for applications that require that level of shape preservation.

Table 4 provides error norms for all of the shapes together and each shape separately to help the reader see how the accuracy is influenced by these choices in WENO and WENO* limiting. By increasing the smoothness indicator exponent and taking away the weight mapping, the WENO weights are much slower to converge to optimal values. This effect is clear in the error norms as accuracy can degrade quite a bit for the smoother functions. This study's main purpose is to demonstrate the range of limiting versus accuracy that this framework can provide, depending on the balance of accuracy and shape preservation a user desires for their particular transport application. In Table 5, the minimum and maximum values are explicitly given when using no limiting and when using WENO and WENO* limiting to show more clearly how much limiting is being performed. The WENO* limiting, even at seventh-order accuracy admits overshoots of only order 10^{-3} relative to the magnitude of the discontinuity.

3.3. Deformational flow

The deformational flow test cases are used to assess the performance of ADER-DT on preserving filaments as well as the effect of FCT, WENO, and WENO* limiting in this context. For deformational flow, the winds are initialized as $u(x,y,t) = \sin^2(\pi x)\sin(2\pi y)\cos(\pi t/T)$ and $v(x,y,t) = -\sin(2\pi x)\sin^2(\pi y)\cos(\pi t/T)$ where $T = 5$ is the final simulation time. The tracer field is initialized as

$$\psi_4(x,y,0) = \left(\cos(\pi r) + 1\right)^2/4$$

where $r = \min(1, 4\sqrt{(x-x_C)^2 + (y-y_C)^2})$ and $x_C = y_C = 0.25$.

Fig. 7 shows the result of deformational flow after forming the filament and returning to normal. Table 6 provides the error norms and minimum and maximum values as well. Higher-order accuracy is effective in better resolving the filament as expected, but also, the effects of limiting decrease with increasing order. Notably, at fifth- and seventh-order accuracies, some of the error norms actually improve with limiting.

3.4. Non-linear correlation preservation

A very important constraint for reactive transport applications that use passively transported tracers in reaction physics routines separately is that existing correlations between tracers be preserved as best as possible [1,27]. To test this, a new tracer, $\psi_5(x,y,0) = 0.9\psi_4(x,y,0) + 0.1$, is introduced as well as a non-linearly correlated tracer, $\psi_6(x,y,0) =$

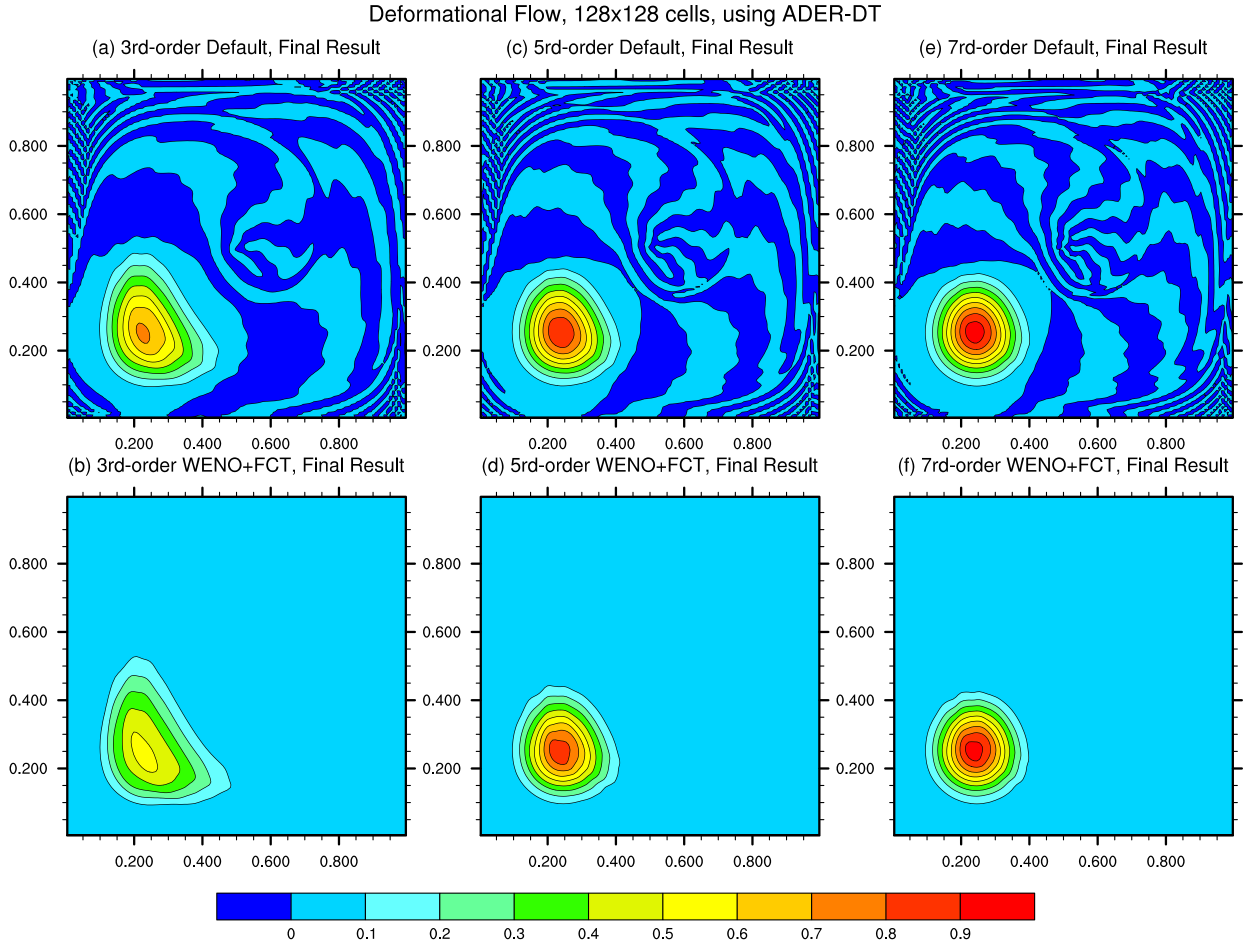


Fig. 7. Plots of final result of deformational flow tests at $t = 5$ s.

Table 6
Error norms using 128×128 cells after 5 seconds of deformational flow. Also included are the minimum and maximum values at $t = 5$ s.

Order	Default					WENO + FCT				
	L_1	L_2	L_∞	min	max	L_1	L_2	L_∞	min	max
3	1.37E−02	4.32E−02	3.16E−01	−5.47E−02	7.24E−01	1.79E−02	6.12E−02	5.00E−01	0.00E+00	5.36E−01
5	6.50E−03	2.18E−02	1.78E−01	−5.21E−02	8.96E−01	5.55E−03	2.20E−02	1.81E−01	0.00E+00	8.80E−01
7	4.64E−03	1.57E−02	1.32E−01	−3.64E−02	9.61E−01	3.87E−03	1.52E−02	1.17E−01	0.00E+00	9.58E−01

$-0.8[\psi_5(x,y,0)]^2 + 0.9$. Both ψ_4 and ψ_5 are evolved separately using deformational flow, and scatter plots of ψ_4 and ψ_5 after evolving with deformational flow are given in Fig. 8 using 128×128 cells and in Fig. 9 using 256×256 cells.

A perfect method would preserve the initial correlation exactly, giving a plot similar to Fig. 9(g). Anything inside the black outline in these figures represents physically realizable mixing for these correlated tracers. Values outside the black outline are considered "unmixing." Values outside the domain $[0.1, 1] \times [0.1, 0.9]$ are considered overshoots and are often considered poisonous to many reaction routines such as chemistry reactions. Since the minimum value for both tracers is 0.1, the FCT filter does not cut off undershoots in these tests, and the only active limiter is WENO.

Correlation plots for two different resolutions are shown to demonstrate that the scheme does converge to the right answer. When the correlations are not resolved, however, each profile is diffused in such a way that the correlations often exhibit some amount of unmixing. This unmixing is expected in terms of numerics because the correlated profile, ψ_5, is steeper and therefore damped more by WENO limiting than the original profile, particularly at larger wave numbers, which are proxied by the lower values of ψ_6. However, with WENO, the overshoots and undershoots are substantially reduced, and with WENO*, they are nearly eliminated altogether, demonstrating that the range of limiting provided by this framework should be sufficient for transport applications.

These plots provide a unique visible demonstration of the resolving power of the different schemes in terms of WENO damping and order of accuracy. The further down and to the right the scatter diagram extends, the better the scheme

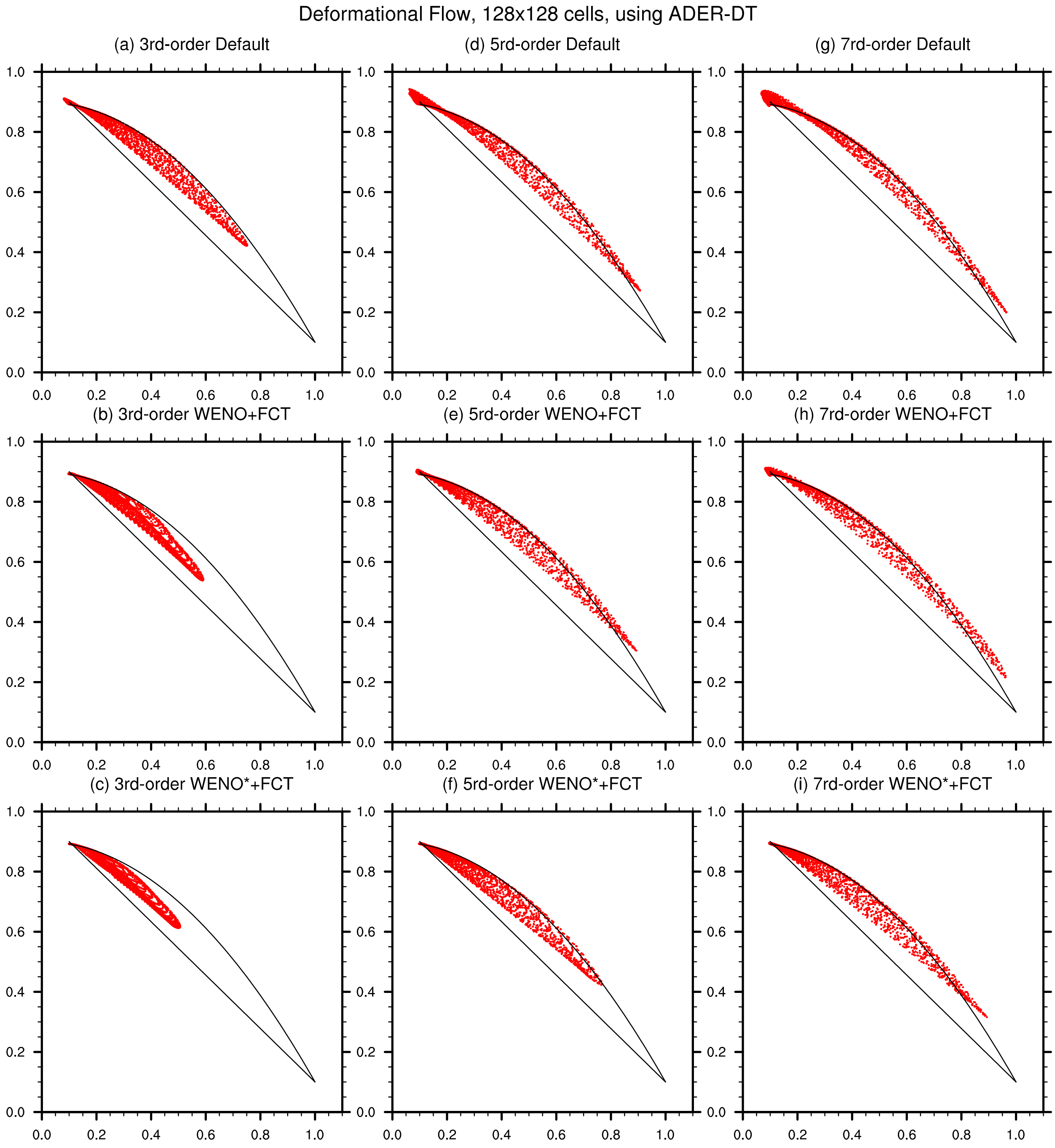


Fig. 8. Scatter plots of ψ_5 and ψ_6 where both are evolved with the same deformational flow using 128 × 128 cells with CFL = 0.45. The x-axis and y-axis are ψ_5 ad ψ_6 values, respectively, for each point on the grid.

is at resolving the smooth extrema in ψ_4 and ψ_5. Clearly, the unmixing is isolated to the less resolved portions of the tracer profiles. Also note the severe damping effects that WENO and WENO* can have at third-order accuracy and how this effect diminishes relatively as the order increases. This shows visually the price paid for eliminating overshoots in terms of accuracy. It may be beneficial for applications to allow some amount of overshooting in exchange for improved resolution of the various tracer profiles. Because of the FCT limiter, none of the undershoots will result in a negative value.

As an example of potential balancing of accuracy and overshoots, consider Fig. 8(h), in which the amount of overshooting is still large, especially compared to third- and fifth-order accuracies in the WENO scheme. Perhaps a slightly larger amount of damping via less weight mapping or a larger smoothness indicator exponent would be desirable in this case.

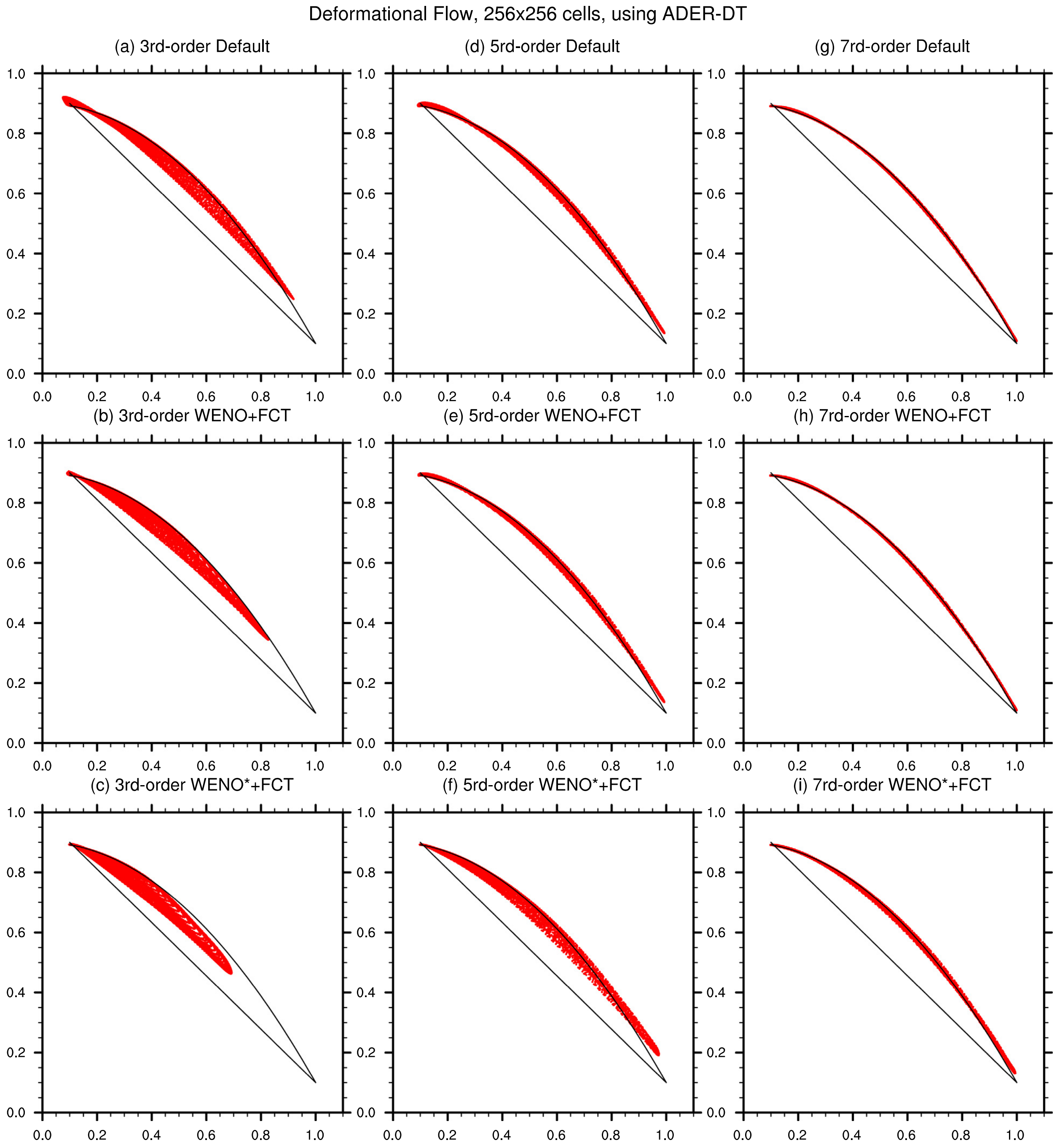


Fig. 9. Same as Fig. 8, except using 256 × 256 cells.

3.5. *Consistency check*

For transport, consistency requires that an initially uniform tracer field in the presence of a non-divergent velocity field must remain uniform for all time. To test this, the density field is initialized to include a slotted cylinder identical by defining: $\rho(x, y, 0) = \psi_3(x, y, 0) + 1$ from Section 3.2. It is important to initialize the density field as positive and non-zero because zero densities are unphysical for most fluid models. A spatially uniform tracer field, $\psi(x, y, 0) = 1$, is used, and solid body rotation is used for velocities. Momentum was initialized via the DT of the product of density and velocity so that it is accurate to the order of the given scheme. After 1.1 revolutions, the L_∞ norm comparing the final tracer field to the initial tracer field was 1.7×10^{-16}, confirming that the scheme is indeed consistent, even for a discontinuous background density field. This is with WENO limiting and FCT positivity turned on, but this is expected since there is no variation for WENO and FCT to respond to.

This test was also run using deformational winds. With a central flux function, e.g., $(f^- + f^+)/2$, the scheme maintains consistency to machine precision just as before, with a maximum L_∞ error of 1.9×10^{-16}. Therefore, ADER-DT itself is able to maintain consistency. However, when an upwind flux function is used with deformational winds, the integration around the cell uses two different Taylor Series expansions to update the cell mean, and this introduces a numerical divergence with a magnitude of the scheme's numerical truncation order. This is true for an upwind flux no matter what is used in the rest of the space–time operator. The numerically introduced divergence from the upwind flux, being at numerical truncation order, does converge to zero with p- and h-refinement. For instance, at third-order accuracy with 64×64 cells, the L_∞ error for the upwind flux function is 8.6×10^{-5}. Yet, at seventh-order accuracy with 160×160 cells, the L_∞ error for the upwind flux function is 1.4×10^{-14}.

4. Conclusions

The WENO- and FCT-limited ADER-DT method within the FV framework for 2-D transport has been described. Therein, constraints for the numerical solution to 2-D transport were given, and it was described how the present method responds to each constraint. Consistency is maintained by using momenta and density from the driving fluid model. Essential shape preservation is enabled to a tunable degree via WENO limiting with the flexibility of two continuous parameters to control smoothness. Accuracy is addressed by limiting only where necessary with WENO and via the use of ADER-DT, which nonlinearly couples all PDE terms to high-order accuracy over a time step. Positivity is enforced with a quick FCT limiting to the fluxes to ensure the mass leaving a cell does not exceed the mass in that cell. Serial runtime is addressed by evolving only a half-tensor of space–time derivatives, by reconstructing efficiently, by reusing WENO weights in each dimension, and by blocking the reconstruction, WENO-limiting, ADER-DT, and analytical integration procedures. Parallel efficiency is addressed by providing a fully-limited, positivity-preserving, arbitrarily high-order-accurate time step with only one required parallel data transfer.

These properties were demonstrated and evaluated with standard 2-D transport test cases. Using uniform transport of a 2-D sine wave, it was found that ADER-DT converges as expected for third-, fifth-, and seventh-order accuracies. Also, ADER-DT performed more accurately than a comparable RK method with less runtime. The solid body rotation test case was used as the primary avenue for analyzing oscillations because of the severity of the discontinuous slotted cylinder. Quantitative bounds and qualitative plots showed the successive reduction of oscillations when using WENO as well as the potential for essential shape preservation when using very smooth parameters with WENO. Thus, a range of limiting options was demonstrated, something tunable to the application in question to balance accuracy and shape preservation. Handling of filaments was demonstrated with deformational flow tests, wherein it was shown how the effect of WENO limiting becomes relatively less severe with increasing order without sacrificing the successful limiting of oscillations. Finally, correlation preservation was evaluated, and again, the range of limiting options with WENO and the ability to virtually eliminate overshoots and undershoots was demonstrated. Finally, consistency to machine precision was verified by using a discontinuous, non-uniform density field with a uniform tracer mixing ratio field.

Thus, this WENO-limited ADER-DT method in the FV framework appears well-suited for consideration in transport applications. Particularly, this study highlights the parallel properties of this scheme in that a fully-limited, positivity-preserving, arbitrarily high-order-accurate time step can be performed with only one required parallel data transfer. In contrast to traditional FCT and hyperdiffusion, this is an appealing quality.

Acknowledgements

The author wishes to thank Dr. Duane Rosenburg of Oak Ridge National Laboratory for essential help regarding efficient multi-dimensional tensor-based reconstruction.

The author also wishes to acknowledge Sage open-source mathematical software system, which is a free, GPL-licensed, open source codebase mainly written in Python. Sage has become an essential part of an efficient workflow in generating these arbitrarily high-order-accurate methods.

This research used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

[1] P.H. Lauritzen, P.A. Ullrich, R.D. Nair, Atmospheric transport schemes: desirable properties and a semi-Lagrangian view on finite-volume discretizations, in: Numerical Techniques for Global Atmospheric Models, Springer, 2011, pp. 185–250.

[2] P.H. Lauritzen, R.D. Nair, P.A. Ullrich, A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid, J. Comput. Phys. 229 (5) (2010) 1401–1424.

[3] R.D. Nair, J.S. Scroggs, F.H. Semazzi, Efficient conservative global transport schemes for climate and atmospheric chemistry models, Mon. Weather Rev. 130 (8) (2002) 2059–2073.

[4] R.D. Nair, B. Machenhauer, The mass-conservative cell-integrated semi-Lagrangian advection scheme on the sphere, Mon. Weather Rev. 130 (3) (2002) 649–667.

[5] M. Zerroukat, T. Allen, A three-dimensional monotone and conservative semi-Lagrangian scheme (SLICE-3D) for transport problems, Q. J. R. Meteorol. Soc. 138 (667) (2012) 1640–1651.

[6] S.-J. Lin, R.B. Rood, Multidimensional flux-form semi-Lagrangian transport schemes, Mon. Weather Rev. 124 (9) (1996) 2046–2070.

[7] L.M. Harris, P.H. Lauritzen, R. Mittal, A flux-form version of the conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed sphere grid, J. Comput. Phys. 230 (4) (2011) 1215–1237.

[8] P.A. Ullrich, M.R. Norman, The flux-form semi-Lagrangian spectral element (FF-SLSE) method for tracer transport, Q. J. R. Meteorol. Soc. 140 (680) (2014) 1069–1085, http://dx.doi.org/10.1002/qj.2184.

[9] M.R. Norman, R.D. Nair, Inherently conservative nonpolynomial-based remapping schemes: application to semi-Lagrangian transport, Mon. Weather Rev. 136 (12) (2008) 5044–5061.

[10] D.M. Hall, R.D. Nair, Discontinuous Galerkin transport on the spherical Yin–Yang overset mesh, Mon. Weather Rev. 141 (1) (2013) 264–282.

[11] R.D. Nair, S.J. Thomas, R.D. Loft, A discontinuous Galerkin transport scheme on the cubed sphere, Mon. Weather Rev. 133 (4) (2005) 814–828.

[12] M. Zerroukat, N. Wood, A. Staniforth, SLICE-S: a semi-Lagrangian inherently conserving and efficient scheme for transport problems on the sphere, Q. J. R. Meteorol. Soc. 130 (602) (2004) 2649–2664.

[13] Y. Zhang, R.D. Nair, A nonoscillatory discontinuous Galerkin transport scheme on the cubed sphere, Mon. Weather Rev. 140 (9) (2012) 3106–3126.

[14] A. Staniforth, J. Côté, Semi-Lagrangian integration schemes for atmospheric models—a review, Mon. Weather Rev. 119 (9) (1991) 2206–2223.

[15] M.R. Norman, H. Finkel, Multi-moment ADER–Taylor methods for systems of conservation laws with source terms in one dimension, J. Comput. Phys. 231 (20) (2012) 6622–6642.

[16] M.R. Norman, Algorithmic improvements for schemes using the ADER time discretization, J. Comput. Phys. 243 (15) (2013) 176–178.

[17] M.O. Deville, P.F. Fischer, E.H. Mund, High-order Methods for Incompressible Fluid Flow, vol. 9, Cambridge University Press, 2002.

[18] E.F. Toro, V.A. Titarev, ADER schemes for scalar non-linear hyperbolic conservation laws with source terms in three-space dimensions, J. Comput. Phys. 202 (1) (2005) 196–215.

[19] M. Dumbser, D.S. Balsara, E.F. Toro, C.-D. Munz, A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes, J. Comput. Phys. 227 (18) (2008) 8209–8253.

[20] D.S. Balsara, C. Meyer, M. Dumbser, H. Du, 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.

[21] J.-B. Cheng, E.F. Toro, S. Jiang, W. Tang, A sub-cell WENO reconstruction method for spatial derivatives in the ADER scheme, J. Comput. Phys. 251 (2013) 53–80.

[22] J. Shi, C. Hu, C.-W. Shu, A technique of treating negative weights in WENO schemes, J. Comput. Phys. 175 (1) (2002) 108–127.

[23] A.K. Henrick, T.D. Aslam, J.M. Powers, Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points, J. Comput. Phys. 207 (2) (2005) 542–567.

[24] V.A. Titarev, E.F. Toro, ADER schemes for three-dimensional non-linear hyperbolic systems, J. Comput. Phys. 204 (2) (2005) 715–736.

[25] D.R. Durran, Numerical Methods for Fluid Dynamics: With Applications to Geophysics, vol. 32, Springer, 2010.

[26] D. Kuzmin, R. Lohner, S. Turek, Flux Corrected Transport, 2nd edition, Springer, 2012.

[27] P.H. Lauritzen, J. Thuburn, Evaluating advection/transport schemes using interrelated tracers, scatter plots and numerical mixing diagnostics, Q. J. R. Meteorol. Soc. 138 (665) (2012) 906–918.