

A High-Order WENO-Limited Finite-Volume Algorithm for Atmospheric Flow Using the ADER-Differential Transform Time Discretization

Matthew R Norman^{1*}

¹National Center for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, TN, 37831-6008, USA

Correspondence

P.O. BOX 2008, MS6008
Oak Ridge, TN, 37831-6008, USA
Email: normanmr@ornl.gov

Funding information

This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

A high-order-accurate Weighted Essentially Non-Oscillatory (WENO) limited Finite-Volume scheme is detailed for the compressible, non-hydrostatic, inviscid Euler equations using an Arbitrary DERivatives (ADER) time stepping scheme based on Differential Transforms (DTs). The scheme achieves high-order-accurate simulation with a large time step and densely clustered computations ideal for modern computing. After demonstrating expected results for literature-standard test cases, this study demonstrates the behavior of the WENO limiting and higher-order accuracy at up to ninth-order. It is shown that even without limiting, upwind fluxes render the scheme stable even with discontinuous flows. Further, it is shown that WENO limiting effectively removes oscillations and affects only the very smallest wavelengths (only modestly larger than $2\Delta x$). Finally, the improvement in resolution and resolved mixing is shown as the order of accuracy is increased for a turbulent flow.

KEYWORDS

WENO, ADER, Finite-Volume, High-Order

*Corresponding author

1 | INTRODUCTION

1.1 | Computational Constraints for Non-Hydrostatic Flows

The 3-D Euler equations form the backbone of many fluid dynamics applications, and the numerical scheme presented here is intended for stratified fluid flows as seen in the large-scale atmosphere. While the Euler equations themselves are simple, solving them in ways that are physically realistic can be quite challenging. One must conserve the conserved variable vector for longer-term simulations with climate time scales. There often occur various discontinuities in the flow arising from phenomena such as cold fronts, monsoons, and cloud convection; and these can lead to oscillations. Also, straightforward discretizations of these non-linear equations often exhibit accumulation of low-wavelength energy even without discontinuities, which can lead to instability [69].

There are further constraints on a potential numerical discretization apart from accuracy and stability concerns that deal with the computational hardware being used [48, 42, 46]. Increasingly, moving data to where computations can be performed costs significantly more than the computations themselves, and modern hardware strongly prefers to perform these computations in “blocks” (i.e., vectorization). The scheme presented here is working toward considering all of these constraints together to create a more optimal numerical discretization for atmospheric simulation. The following are desirable computational traits: a large time step, economical cost per time step, low frequency and volume in parallel data transfers, contiguous accesses to Dynamic Random Access Memory (DRAM), small local data requirements to improve cache efficiency, large amounts of work performed in cache without interrupting data transfers, and relatively little branching (if-then-else statements) in the code [48].

This study concerns only rectangular Cartesian grids for cloud-scale atmospheric simulation, which obviates the need to handle large aspect ratios with techniques such as Horizontally Explicit, Vertically Implicit (HEVI) integration schemes (e.g., [15, 68]). Here, one can rely on fully explicit simulation in all dimensions.

1.2 | Survey of Existing Discretizations

1.2.1 | Spatial Discretizations

Multi-Moment Schemes

Spatial discretizations used for the Euler equations (or any hyperbolic equation system) can be grouped into two main categories: single-moment and multi-moment. Multi-moment discretizations include all element-based techniques, the time-explicit variants largely belonging to the Galerkin class of methods. Key to the definition of multi-moment schemes is that they treat groups of Degrees Of Freedom (DOFs), usually called “elements”, all together at once. What effectively happens in all multi-moment schemes is that in each element, only local DOFs within the elements are used to recover intra-element variation of the solution. Then, each DOF in the element uses the *same* recovery when computing the update for the spatial discretization. Single-moment discretizations, by contrast, treat each DOF individually, usually using a “stencil” of information from surrounding DOFs to constrain the variation of the solution in the neighborhood of each individual DOF. There also exist multi-moment schemes with recoveries as well (e.g., [11]), and these generally share a hybrid of the properties of each.

Multi-moment schemes used for the Euler equations include modal and nodal Galerkin schemes. Modal schemes use basis expansions that include basis functions of differing levels of variation (e.g., [39]), while nodal schemes use basis functions that each have the same level of variation and can often be represented as point value estimates of the solution at various locations within the element (e.g., [14]). Multi-moment schemes can also be discontinuous or continuous, meaning that co-located solutions between adjacent elements are either forced to be continuous or are

allowed to be discontinuous (usually enforcing a flux term between elements) [17, 37]. There are mass-conserving approaches for both continuous and discontinuous element-based schemes. Some multi-moment schemes are based on a variational form of the PDE that uses test functions and integration by parts to remove spatial derivatives from flux terms, and some are based on the simpler weak (spatially integrated) form [71, 23, 44]. Often, weak variations have larger maximum stable explicit time steps than variational forms.

There are also finite-element and mixed finite-element simulations with discrete spatial operators that mimic various properties from the continuous equations (e.g., [8]). The discretization presented here does not have these properties, though it conserves the conserved variable vector to machine precision.

Single-Moment Schemes

Single-moment schemes are largely divided into two camps: finite-differences and finite-volumes. Finite-differences use the strong form of the PDE, leaving the derivative on the flux term and thus requiring the flux recovery to have a well-posed derivative. Finite-volumes use the weak (integrated, not variational) form of the PDE, removing the flux term spatial derivative via the Gauss divergence theorem, therefore no longer requiring a differentiable flux recovery. For this reason, finite-volumes are often more dissipative than finite-differences, though much of this property owes itself to the choice of how to compute the flux.

Finite-volumes, in turn, can be decomposed into two main camps based on how fluxes are computed: (1) direct flux computations; and (2) reconciled discontinuous fluxes. Directly reconstructed fluxes (e.g., [57]) add little dissipation to the FV scheme, and in these methods, cell-edge fluxes are directly reconstructed from surrounding information. Reconciled fluxes [62, 52] allow the scheme developer to introduce a varying range of dissipation. In these methods, one typically uses cell-centered recovery operators to provide two flux estimates per cell edge: one from the left cell and one from the right cell (thinking in 1-D). Then, one reconciles these fluxes using a numerical flux function that could be some approximation to the upwind Riemann state or some other perhaps less dissipative flux function that may also be valid for non-hyperbolic systems of PDEs [53, 25].

1.2.2 | Oscillation Limiting Considerations

As mentioned earlier, instabilities often develop in the Euler equations that lead to accumulation of energy at small wavelengths. Even if the simulation is stable without explicit dissipation, as it is with upwind FV schemes, oscillations can still develop, which can contaminate certain added physical approximations such as microphysics and chemistry [29]. Discontinuities need to be limited to remove oscillations while keeping the gradients as steep as possible. This is an important consideration because element-based schemes and single-moment schemes limit in different ways.

To the author's knowledge, every element-based scheme limits the DOFs on a per-element basis rather than individually, and there are varying ways to do this. There are variable second-order viscosity approaches [72], higher-order "hyper"-viscosity approaches [57, 10], optimization-based approaches that limit element extrema over a time step to that of the neighborhood [19], WENO-based limiters [51, 20], modal high-frequency limiters [38], and flux-correction approaches [34, 26]. Each of these treats all DOFs within each element at the same time rather than treating each DOF individually. Conceptually, however, treating each DOF individually allows more localized (thus less dissipative) treatments of discontinuities, lending a potential advantage to single-moment schemes. Further, if one did treat each DOF within an element individually, the computational complexity becomes something much closer to that of single-moment schemes, negating the speed benefit of an element-based approach discussed in the previous section. Also, limiting in multi-moment schemes invariably requires extra parallel communication.

With FV schemes in particular, since a halo has already been transferred for reconstruction, that same halo can also

be used for high-resolution limiting techniques that include Essentially Non-Oscillatory (ENO) [56], Weighted ENO (WENO) [33], Piecewise Parabolic Method (PPM) [7], and even non-polynomial recovery operators [70, 47]. Thus, with the same communication requirements, one can both limit and recover all at once.

Because a direct limiting comparison between FV and element-based techniques is beyond the scope of this article, we strongly caveat that the statement about potentially more accurate limiting for FV methods remains strictly conceptual in nature and must be further tested experimentally before it can be stated with greater confidence.

1.2.3 | Eulerian versus Lagrangian

There is also the consideration of whether to use an Eulerian or Lagrangian framework, and this choice has strong interactions with the temporal operator as well. While there exist Lagrangian operators for the full non-linear dynamics equations [49, 22], such characteristics-based approaches cannot be extended to a CFL value beyond the Eulerian stability limit without degrading accuracy when the eigenvalues experience strong variation [49], which they always do in realistic atmospheric simulation. Further, dimensional splitting must be employed to avoid the multi-dimensional nature of acoustic Mach cones (2-D) and ellipsoids (3-D). While in [35], the true multi-dimensional nature of acoustics is represented, it is rendered semi-discrete in practice (and therefore essentially Eulerian in nature). In most Lagrangian approaches to full Euler dynamics, only the slow (transport-based) dynamics are handled in a Lagrangian manner since those characteristics trace 1-D lines in 3-D space (in contrast to acoustics), and fast dynamics are handled in an Eulerian manner [54, 60]. Lagrangian schemes exist for single-moment and multi-moment schemes [67, 16] in non-conservative and conservative form [27].

This study chooses an Eulerian framework to keep slow and fast dynamics in the same reference and to allow cheaper limiting that needs only be applied once rather than separately for the Eulerian and Lagrangian updates for fast and slow dynamics, respectively. Having dynamics and tracer transport in the same framework also simplifies the coupling, which can become complex when one part is Lagrangian and the other is Eulerian [30]. Also, trajectory calculations are indirectly addressed in computer memory because it isn't known a priori what cell the trajectory lands in at a given time – leading to additional latency in data fetches in the interpolation as well as non-contiguous memory accesses. While in tracer transport, coefficients of dot-product-based interpolation can be pre-computed and re-used for many tracer quantities [28], this is not the case in non-linear dynamics, which has fewer variables.

1.2.4 | Grid Staggering

Grid staggering of the various variables in an atmospheric dynamics model has long been a hot topic of conversation, and they are defined on 2-D and 3-D grids by which variables live on cell centers, faces, edges, and corners (as well as what the directionality of those variables) [58, 2]. Certain aspects of staggering have also been generalized in differential form spaces in compatible finite-element techniques [40]. Staggering-related concepts can change conservation and stability properties of a numerical schemes significantly.

In the present study, a co-located grid (so-called “A”-grid) is used for sake of simplicity for several reasons. First, while staggering has been shown to improve accuracy for certain parts of the discretization, the improvement is at the scheme's numerical truncation. This study proposes high-order schemes where this benefit is not as pronounced, and has found that there is virtually no penalty to high-order accuracy in terms of runtime on accelerators because of their excellent compute throughput. Also, it is found in this study that a Finite-Volume method with upwind fluxes using the linear Godunov state are more than sufficient to damp out small wavelength energy, leaving the scheme reliably stable even at ninth-order accuracy without any explicit limiting or damping.

1.2.5 | Dimensional Splitting

Dimensional splitting is the act of treating each dimension separately, and it can be done in different ways. One method is to apply a traditional operator splitting technique such as Strang splitting, which computes x , y , and z -direction tendencies in succession, using updates from the previous direction before computing tendencies in the next [36]. Another approach is to compute tendencies in all directions completely separately using the same forcing, and couple them within each stage of a multi-stage operator, relying on the stages to generate cross terms automatically in the discretization [24]. There also exists splitting schemes that couple cheap non-conservative Lagrangian updates with more expensive flux-form updates (e.g. [31]).

Dimensional splitting for Finite-Volume methods is desirable computationally because it reduces the computational complexity of the scheme by a factor of $N^{(D-1)}$, where N is the order of accuracy of the scheme, and D is the number of spatial dimensions. Most splittings will suffice in an orthogonal coordinate system, but in non-orthogonal coordinates, a straightforward Strang splitting gives very poor accuracy [48, 5], though coupling dimensions through a multi-stage time integrator is still accurate [24]. However, for ADER-DT methods, there is no existing implementation that uses stages. For this reason, ADER-DT methods must be fully multi-dimensional in non-orthogonal coordinates as demonstrated in [48]. However, since the present study only considers Cartesian coordinates, a second-order-accurate Strang splitting is used here.

1.3 | Temporal Discretizations

1.3.1 | Time-Implicit Methods

The Euler equations, when linearized, can be decomposed into a set of waves propagating information at finite speed (see Appendix A), making the equations “hyperbolic.” Because of this, temporal discretizations that set up global dependence in the solution inherently allow very large and sometimes unlimited time steps. This is because they respect an infinite degree of information propagation within a time step. Time-implicit and Spectral methods both have this property. Time-implicit methods create global dependence by forcing the solution at the next time step to depend (at least partially) upon itself, setting up a global linear system to solve. Spectral methods do this by transforming the PDEs into modal space using spherical harmonics (a globally dependent operation).

Global dependence can improve the efficiency of the discretization for stiff systems where physically unimportant fast waves unduly limit stability. While much of the atmosphere can correctly be labeled as “stiff”, the *global* atmosphere is arguably not very stiff. Jet streak maxima can reach up to order 100m s^{-1} , while the speed of sound has a maximum near the surface of about 350m s^{-1} . This provides a reasonable global Mach number of about $1/4$, meaning there isn't much scale separation between the advective dynamics we wish to capture accurately and acoustic dynamics that are arguably physically insignificant in the *global* atmosphere. Note that this section is only considering horizontal wave propagation, not vertical wave propagation, for which global spatial scales typically require HEVI schemes.

This argument is relevant because global dependence, computationally speaking, implies global data communication via MPI or some other library. Time-implicit methods require many iterations of `MPI_AllReduce` each time step (to orthonormalize Krylov subspace vectors)[21], and spectral methods require a set of quasi-global block-wise `MPI_AllToAll` each time step [6] to re-arrange the pencil decomposition. At small scales, these are manageable, but at large scales the costs of these operations become prohibitive. With enough per-node workload, one can amortize the global data communication overhead. However, high-resolution climate simulation, which must simulate at $2,000\times$ realtime, has per-node workloads far too small to amortize global communication overheads. Consider that on the Oak Ridge Leadership Computing Facility's (OLCF's) Titan supercomputer, the Energy Exascale Earth System Model (E3SM)

spends roughly 90% of the dynamics runtime in *nearest-neighbor* MPI communication when run on 5,400 nodes with the Spectral Element method, easily among the most scalable dynamics implementation available (the author performed these performance evaluations). Global MPI communications would be significantly more costly.

For these reasons, this study uses a time-explicit temporal discretizations.

1.3.2 | Time-Explicit Methods

Time-explicit temporal discretizations do not set up global dependence within a time step. Therefore, the time step will be limited by stability ultimately due to the need to resolve finite propagation of information, though rarely is the linear analysis easy to explain physically for general methods. This concept was first introduced in [9], but the physical reasoning of a wave propagating between two grid points is not applicable in general. We must rely on the results of linear analysis instead to determine the maximum stable time step of a scheme, as seemingly minor changes to a spatial discretization can lead to dramatic changes in the resulting stability.

Time-explicit discretizations are generally categorized as multi-step, multi-stage, or fully-discrete [12, 49]. The semi-discrete schemes (e.g., multi-stage or multi-step) treat the spatially discretized (continuous in time) ODE with an ODE solver such as Runge-Kutta or Adams-Bashforth (or other varieties as well). Fully-discrete schemes operate by propagating known spatial information into the temporal domain, the high-order generalization of which will either be Lagrangian [59] or Arbitrary DERivatives (ADER) in nature [61, 62]. Lagrangian methods use characteristic trajectories of hyperbolic waves to link temporal and spatial information, and ADER methods use the PDE definition itself, which directly relates spatial and temporal variation through derivatives.

Multi-stage and multi-step methods must be Strong Stability Preserving (SSP) in order to avoid having the temporal discretization itself lead to new extrema [18]. As far as mainstream SSP RK integrators are concerned, if one considers higher-than-second-order integrators, the best effective CFL coefficient is about 1/3. The same is true for SSP multi-step integrators with the notable exception of a six-step third-order method with a CFL of 0.57. Multi-stage and multi-step SSP methods can get up to effective CFL values near 0.7 [4]. For this reason, we consider ADER schemes to have a potential advantage because they maintain non-oscillatory properties of the underlying spatial discretization out to a CFL value of 1, allowing a larger time step and thus less overall data communication. Note that in general ADER schemes are not fully monotone, though, which is a potential disadvantage in certain contexts. Efficiently implemented, ADER methods also have the desirable computational complexity of $O(N^{D+2})$ for the complete space-time operator [48].

1.4 | Summary of the Proposed Method

The main hypothesis being explored in this study is that high-order accuracy, when *properly limited*, can provide increasing resolution even for complex and discontinuous flows. There is already evidence showing improved solutions for very similar high-order methods for discontinuous transport in [48]. Here, we look at a chaotic non-linear context with the Euler equations. The goal is eventually to increase physical realism while also obtaining more appropriate computational properties in the discretization. For the same reasons as given in [48], this study chooses to use an arbitrarily high-order upwind Finite-Volume method with WENO limiting and the ADER-Differential Transform (ADER-DT) time discretization using dimensional splitting to efficiently discretize the compressible non-hydrostatic Euler equations over a Cartesian domain. The emphasis of this scheme is accuracy, clustered computation, and a large time step.

2 | MATHEMATICAL FORMULATION AND IMPLEMENTATION

2.1 | Inviscid Euler Equations

The 3-D compressible non-hydrostatic Euler equations, though inviscid, form the basis of atmospheric simulation and a good testbed for prospective numerical methods. The equation set describing this flow is:

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho \theta \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho u \theta \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ \rho v \theta \end{bmatrix} + \frac{\partial}{\partial z} \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p - \rho_H \\ \rho w \theta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -(\rho - \rho_H)g \\ 0 \end{bmatrix}$$

$$\rho_H = -\frac{1}{g} \frac{\partial p}{\partial z}$$

where ρ is density, u , v , and w are winds in the x -, y -, and z -directions, respectively, θ is potential temperature related to temperature, T , by $\theta = T (P_0/P)^{R_d/c_p}$, $P_0 = 10^5$ Pa, $g = 9.8 \text{ m s}^{-2}$ is acceleration due to gravity, $p = C_0 (\rho \theta)^\gamma$ is pressure, $C_0 = R_d^\gamma p_0^{-R_d/c_v}$, $R_d = 287 \text{ J kg}^{-1} \text{ K}^{-1}$, $\gamma = c_p/c_v$, $c_p = 1004 \text{ J kg}^{-1} \text{ K}^{-1}$, $c_v = 717 \text{ J kg}^{-1} \text{ K}^{-1}$, and $p_0 = 10^5$ Pa. The hydrostatic density and potential temperature are related to hydrostatic pressure through the following relation:

$$\frac{dp_H}{dz} = C_0 \frac{d(\rho_H \theta_H)^\gamma}{dz} = -\rho_H g$$

To cast this in terms of the state vector, it can be restated as:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} + \frac{\partial \mathbf{h}}{\partial z} = \mathbf{s} \quad (1)$$

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{bmatrix}; \quad \mathbf{f} = \begin{bmatrix} q_2 \\ q_2^2/q_1 + C_0 q_5^\gamma \\ q_2 q_3/q_1 \\ q_2 q_4/q_1 \\ q_2 q_5/q_1 \end{bmatrix}; \quad \mathbf{g} = \begin{bmatrix} q_3 \\ q_3 q_2/q_1 \\ q_3^2/q_1 + C_0 q_5^\gamma \\ q_3 q_4/q_1 \\ q_3 q_5/q_1 \end{bmatrix}$$

$$\mathbf{h} = \begin{bmatrix} q_4 \\ q_4 q_2/q_1 \\ q_4 q_3/q_1 \\ q_4^2/q_1 + C_0 (q_5^\gamma - q_{5,H}^\gamma) \\ q_4 q_5/q_1 \end{bmatrix}; \quad \mathbf{s} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -g(q_1 - q_{1,H}) \\ 0 \end{bmatrix}$$

where q_i denotes the i th component of \mathbf{q} .

2.2 | Spatial Discretization

This study uses a Finite-Volume spatial discretization, meaning the essential operation to the original PDE system is to integrate it over a cell domain and use the Gauss divergence theorem to remove the need for a differentiable flux function. The choice was made to reconstruct variation over cell domains and to sample multi-valued fluxes at each cell boundary, which are then reconciled using a Riemann solver. Even for flows with discontinuities (excluding shocks), when using upwind fluxes in a FV method, the resulting scheme is stable without the need for any additional dissipation (which will be demonstrated in the numerical results). This method uses no grid staggering.

2.2.1 | Dimensional Splitting

When run in a fully multi-dimensional manner, FV consumes N^D more operations than element-based operators such as SE or DG when considering an equal number of DOFs between the two, where N is the spatial order of accuracy. A spatial recovery operator (the most expensive component for a FV algorithm) costs DN^{D+1} operations, which is the same cost for a local update for SE and DG (assuming the mass matrix is rendered diagonal by continuous or discrete orthogonality or mass "lumping"). However, the element-based update need only be performed *once* for all of the N^D DOFs in the element, whereas the FV recovery operator must be completed for each DOF. This is why genuinely multi-dimensional FV algorithms are prohibitively expensive compared to their element-based counterparts.

However, considering computational complexity alone, FV is more appropriately compared when it is dimensionally split. For a non-orthogonal grid, a direct dimensional splitting can cause serious errors [48], but for an orthogonal grid, the errors introduced by splitting are significantly lower. Dimensional sweeps of 1-D FV reconstruction is only a factor of N more expensive than element-based reconstruction, whereas fully multi-dimensional FV reconstruction is N^D times more expensive. Further, when run dimensionally split, the FV method can run with a CFL value that is D times larger than the maximum stable CFL value of a fully multi-dimensional reconstruction. Therefore, this study uses the FV method in a dimensionally split manner.

A second-order-accurate alternating Strang splitting is used, which reverses the order of the dimensional solves each time step. If the update in a given direction can be cast as $q_{n+1} = RHS(q_n)$, where n is the time step index, then the update proceeds as follows:

$$q_n^* = RHS_x(q_n); \quad q_n^{**} = RHS_y(q_n^*); \quad q_{n+1} = RHS_z(q_n^{**})$$

$$q_{n+1}^* = RHS_z(q_{n+1}); \quad q_{n+1}^{**} = RHS_y(q_{n+1}^*); \quad q_{n+2} = RHS_x(q_{n+1}^{**})$$

For the dynamics solve, the equations in the x -, y -, and z -directions are: $\partial_t \mathbf{q} + \partial_x \mathbf{f} = 0$, $\partial_t \mathbf{q} + \partial_y \mathbf{g} = 0$, and $\partial_t \mathbf{q} + \partial_z \mathbf{h} = \mathbf{s}$, respectively. For sake of simplicity in the rest of this article, each will be described in general as

$$\partial_t \mathbf{q} + \partial_x \mathbf{f} = \mathbf{s} \tag{2}$$

keeping in mind that except for the vertical update of dynamics, $\mathbf{s} \equiv 0$.

2.2.2 | Finite-Volume Semi-Discretized Equations

Since the equations are dimensionally split, consider the general case of a 1-D PDE with a source term. For the Finite-Volume discretization, the 1-D equation (2) is integrated over a 1-D local cell domain, $\Omega_i \in [x_{i-1/2}, x_{i+1/2}]$, where $x_{i\pm 1/2} = x_i \pm \Delta x_i/2$, and Δx_i is the grid spacing for the cell of index i . This gives rise to the following semi-discretized equations:

$$\frac{\partial \bar{\mathbf{q}}_i}{\partial t} + \frac{1}{\Delta x_i} (\mathcal{F}_{i+1/2} - \mathcal{F}_{i-1/2}) = \frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} \tilde{\mathbf{s}}_i(x, t) dx$$

$$\mathcal{F}_{i-1/2}(t) = \mathcal{R}[\tilde{\mathbf{f}}_{i-1,j}(x_{i+1/2}, t), \tilde{\mathbf{f}}_{i,j}(x_{i-1/2}, t)]$$

$$\bar{\mathbf{q}}_i \equiv \frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{q}(x, t) dx$$

where $\tilde{\mathbf{f}}_i(x, t)$ and $\tilde{\mathbf{s}}_i(x, t)$ are spatial reconstructions of the vectors \mathbf{f} and \mathbf{s} , respectively, over the i th cell domain, Ω_i , (the time dimension is left continuous for now), and $\mathcal{R}(\mathbf{f}^-, \mathbf{f}^+)$ is a Riemann solver to reconcile the multi-valued flux at a cell interface due to samples from disparate reconstructions at the same location in space.

2.2.3 | Spatial Recovery Operator

N th-order-accurate FV methods recover intra-cell variation by projecting a stencil of N surrounding cell averages onto a polynomial of order $N - 1$. In the present case, and for reasons that will be more fully discussed, the stencil is rather projected onto N_T GLL points, where N_T is the temporal order of accuracy. Consider an arbitrary polynomial, $p(x) = \sum_{i=0}^{N-1} a_i x^i$. Also consider the following vectors of constraints applied to this arbitrary polynomial:

$$\mathbf{c}_{stencil,j} = (1/\Delta x_{i-h+j}) \int_{\Omega_{i-h+j}} p(x) dx \quad ; \quad \forall j \in \{0, \dots, N-1\}$$

$$\mathbf{c}_{GLL,j} = p(x_j + \xi_j/(2\Delta x_i)) \quad ; \quad \forall j \in \{1, \dots, N_T\}$$

where $h = (N - 1)/2$ is the "halo" size and ξ_j is the j th of N_T GLL points spanning the domain, $[-1, 1]$. From these constraints, one can pre-compute a non-square matrix operator, \mathcal{M}_{GLL} , that efficiently transforms N stencil averages into N_T GLL points as follows:

$$\mathcal{M}_{GLL} = \left(\frac{\partial \mathbf{c}_{GLL}}{\partial \mathbf{a}} \right) \left(\frac{\partial \mathbf{c}_{stencil}}{\partial \mathbf{a}} \right)^{-1}$$

For the vertical dimension, extra care is needed because it is less accurate to directly reconstruct the hydrostatic background state via polynomials. The vertical profiles of density and potential temperature are dominated by hydrostatic balance, upon which relevant motions result from relatively small perturbations. Hydrostasis is defined

by:

$$\frac{dp_H(z)}{dz} = -\rho_H(z)g \implies C_0 \frac{dq_{H,5}(z)}{dz} = -q_{H,1}(z)g$$

where the H subscript denotes the hydrostatically balanced profile that depends only on the vertical dimension. Therefore, before reconstruction, hydrostasis is removed from each cell average in the stencil follows:

$$\begin{bmatrix} \bar{q}'_1 \\ \bar{q}'_2 \\ \bar{q}'_3 \\ \bar{q}'_4 \\ \bar{q}'_5 \end{bmatrix} = \begin{bmatrix} \bar{q}_1 - \bar{q}_{H,1} \\ \bar{q}_2 / \bar{q}_{H,1} \\ \bar{q}_3 / \bar{q}_{H,1} \\ \bar{q}_4 / \bar{q}_{H,1} \\ \bar{q}_5 - \bar{q}_{H,5} \end{bmatrix}$$

Then, the perturbed vector, $\bar{\mathbf{q}}'$, is reconstructed in the vertical direction. Finally, once the perturbed stencil is projected onto N_T GLL points giving $\mathbf{q}_{GLL,j}$ for $j \in \{1, \dots, N_T\}$, the hydrostatic background (which has been pre-computed on each of the GLL points) is added back as:

$$\begin{bmatrix} q_{GLL,j,1} \\ q_{GLL,j,2} \\ q_{GLL,j,3} \\ q_{GLL,j,4} \\ q_{GLL,j,5} \end{bmatrix} = \begin{bmatrix} q'_{GLL,j,1} + q_{H,GLL,j,1} \\ q'_{GLL,j,2} q_{H,GLL,j,1} \\ q'_{GLL,j,3} q_{H,GLL,j,1} \\ q'_{GLL,j,4} q_{H,GLL,j,1} \\ q'_{GLL,j,5} + q_{H,GLL,5} \end{bmatrix}$$

where j is the j th of N_T GLL points in the domain Ω_i . The hydrostatic background state will be updated infrequently, and therefore, much higher accuracy can be used in computing the hydrostatic cell averages and associated GLL points within cells – thus the benefit to removing them from standard reconstructions.

2.2.4 | Linear Upwind Riemann Solver

For the dynamics, a characteristics-based flux vector splitting is used to compute an upwind Riemann state using linear characteristics from a locally “frozen” background state. Consider a PDE: $\partial_t \mathbf{q} + \partial_x \mathbf{f} = 0$ with a flux Jacobian $A = \partial \mathbf{f} / \partial \mathbf{q}$. Left-multiplying the equation by the flux Jacobian and assuming that it is locally constant in space and uniform in time gives: $\partial_t \mathbf{f} + A \partial_x \mathbf{f} = 0$. The flux Jacobian can then be diagonalized into $A = R \Lambda R^{-1}$, where R is a matrix whose columns are right-eigenvectors, and Λ is a diagonal matrix whose diagonal components are eigenvalues, λ_p . Using this diagonalization gives: $\partial_t \mathbf{f} + R \Lambda R^{-1} \partial_x \mathbf{f} = 0$. Left-multiplying by R^{-1} gives: $\partial_t \mathbf{w} + \Lambda \partial_x \mathbf{w} = 0$, where $\mathbf{w} = R^{-1} \mathbf{f}$ is the vector of five flux-based characteristic variables that each travel at a finite speed of λ_p . The diagonalizations for each direction are given in Appendix A.

To compute the locally frozen flux vector, the linear average at the interface is used: $\hat{\mathbf{q}} = (\mathbf{q}^- + \mathbf{q}^+) / 2$. Then, the locally frozen eigen-matrices, \hat{R} , $\hat{\Lambda}$, and $\hat{L} = \hat{R}^{-1}$ are built on the averaged state, $\hat{\mathbf{q}}$. Next, the flux-based characteristic

variable vector, \mathbf{w}_{upw} , is computed as:

$$w_{upw,j} = \begin{cases} \widehat{\ell}_j \cdot \mathbf{f}^- & \text{if } \lambda_j > \epsilon \\ \widehat{\ell}_j \cdot \mathbf{f}^+ & \text{if } \lambda_j < -\epsilon \\ \widehat{\ell}_j \cdot (\mathbf{f}^- + \mathbf{f}^+) / 2 & \text{otherwise} \end{cases}$$

where $\widehat{\ell}_j$ is the j th row of \widehat{L} , λ_j is the j th diagonal component of $\widehat{\Lambda}$, and ϵ is a small tolerance to handle characteristic speeds very close to zero. Finally, the upwind flux is obtained as: $\mathcal{R}(\mathbf{f}^-, \mathbf{f}^+) = \widehat{R}\mathbf{w}_{upw}$.

2.3 | Weighted Essentially Non-Oscillatory (WENO) Limiting

In this study, WENO limiting is applied to the dynamics variables. WENO limiting computes multiple polynomial reconstructions over the stencil for each cell. Then, it computes the variation of each reconstruction and gives the smoothest polynomials the largest weighting. In this manner, the resulting polynomial approximation is smoother near discontinuous regions yet recovers the original high-order-accurate polynomial in smoother regions. This study uses the same WENO method described in [48], which is a function-based approach that gives sharper resolution than traditional WENO interpolation. In this study, each variable has its own non-linear WENO weights (in contrast to sharing the same weights for all dynamics variables). A brief graphical demonstration of WENO limiting is given in Appendix B.

2.4 | ADER-DT Time Discretization

2.4.1 | The ADER Approach

The ADER discretization [61, 64, 65] can be thought of in one sense as a higher-order extension of the Lax-Wendroff [55] approach. Typical fluid dynamics PDEs relate spatial derivatives of the fluid state to the temporal derivative of the fluid state. Consider 1-D homogeneous transport with uniform wind: $q_t = -q_x$. If the first spatial derivative is known, the first temporal derivative is known as well. By differentiating successively to high-order derivatives in space and time, any order of temporal derivative can be computed if that same order of spatial derivative is also known: $q_{xt} = -q_{xx}$, $q_{tt} = -q_{xt}$, etc. This process is described in literature as the Cauchy-Kowalewski (C-K) procedure, and it demonstrates that the temporal order of accuracy of an ADER method matches that of the spatial order of accuracy. One advantage of ADER time stepping is that it inherently maintains non-oscillatory properties of a limited spatial reconstruction because temporal information is computed directly from the spatial information. While the C-K method is straightforward to implement with symbolic mathematical software such as Maple or Sagemath (even exported directly to C or Fortran), in practice, this method is very expensive – especially compared to other realizations of the ADER approach to time discretization.

Other improvements to the ADER methodology include the Galerkin-in-time realization [3, 11], which iterates the coefficients of time and space-time basis functions, given an initial knowledge of the spatial-only coefficients. These methods require fast transformations between modal and nodal space, and they also require iteratively solving non-linear algebraic equations that are different for different orders of accuracy and different geometries. This is an efficient way to implement an ADER scheme, especially compared to the C-K procedure, and it has been implemented for both Finite-Volume and Galerkin spatial operators.

2.4.2 | ADER with Differential Transforms

Another implementation called ADER-DT relies on Differential Transforms (DTs) to transform the PDE into a recurrence relation to generate higher-order time and space-time derivatives [45, 41, 43]. The temporal polynomial coefficients can be collapsed down to a time-average and inserted into the spatial operator as if it were semi-discrete, requiring no changes to the original semi-discretized operator. The downside to the implementation in [45, 41] is that the computational complexity is $O(N^{2(D+1)})$, which is incredibly expensive, compared to the expected efficient computational complexity of a space-time operator of $O(N^{D+2})$. In [48], however, the ADER-DT approach was significantly improved by only computing DTs in the temporal dimension on a set of GLL points in space. This study uses the improved implementation in [48] expanded to the Euler equations.

The k th Differential Transform of a function, f , is the k th term in the Taylor Series of f . The inverse transform is the Taylor Series itself.

$$F(k) = \frac{1}{k!} \frac{\partial^k f(x)}{\partial x^k}$$

$$f(x) = \sum_{k=0}^{N-1} F(k) x^k$$

Using these two definitions, the rules to transform various non-linear operators can be derived, and transform rules for all common operators can be found in [66].

First, the spatial reconstruction procedure transforms a stencil of N cell averages of the model state, \mathbf{q} , into N_T GLL points, where N and N_T are the spatial and temporal orders of accuracy, respectively. Next, the PDE system (2) is transformed into modal derivative space only in the temporal dimension using DTs, leaving the spatial dimension in nodal space on GLL points.

Temporal Differential Transforms of the Euler Equations

Since the DTs are only being performed in time for a given GLL point, the spatial dimensions are neglected for simplicity, and the transformed PDEs become in the x -, y -, and z -directions, respectively:

$$\mathbf{Q}(k_t + 1) = -\frac{1}{k_t + 1} \partial_x \mathbf{F}(k_t)$$

$$\mathbf{Q}(k_t + 1) = -\frac{1}{k_t + 1} \partial_y \mathbf{G}(k_t)$$

$$\mathbf{Q}(k_t + 1) = -\frac{1}{k_t + 1} [\partial_z \mathbf{H}(k_t) + \mathbf{S}(k_t)]$$

where a capital letter denotes the transformed function in time, and k_t denotes the (k_t) th-order DT in time, which is also the (k_t) th term in the temporal Taylor series.

Before computing the DTs of the flux and source terms, the following auxiliary functions are defined: $\phi_{ij}(t) =$

$q_i(t) q_j(t) / q_1(t)$ and $p^*(t) = q_5(t)^Y$. The symmetry $\phi_{ij} = \phi_{ji}$ (naturally mirrored in the temporal DTs) can be employed to reuse previously computed auxiliary functions instead of re-computing the same DT twice. Given these functions, the following are the temporal DTs of the flux and source vectors in the Euler equations.

$$\mathbf{F}(k_t) = \begin{bmatrix} Q_2(k_t) \\ \Phi_{22}(k_t) + P(k_t) \\ \Phi_{23}(k_t) \\ \Phi_{24}(k_t) \\ \Phi_{25}(k_t) \end{bmatrix}; \quad \mathbf{G}(k_t) = \begin{bmatrix} Q_3(k_t) \\ \Phi_{32}(k_t) \\ \Phi_{33}(k_t) + P(k_t) \\ \Phi_{34}(k_t) \\ \Phi_{35}(k_t) \end{bmatrix}; \quad \mathbf{H}(k_t) = \begin{bmatrix} Q_4(k_t) \\ \Phi_{42}(k_t) \\ \Phi_{43}(k_t) \\ \Phi_{44}(k_t) + P(k_t) \\ \Phi_{45}(k_t) \end{bmatrix}$$

$$\mathbf{S}(k_t) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -g Q_1(k_t) \\ 0 \end{bmatrix}$$

$$\Phi_{ij}(k_t) = \frac{1}{Q_1(0)} \sum_{r_t=0}^{k_t} [Q_i(r_t) Q_j(k_t - r_t) - Q_1(r_t) \Phi_{ij}(k_t - r_t)]; \quad \forall i \in \{2, 3, 4\}, j \in \{2, 3, 4, 5\}$$

$$P(k_t) = \frac{C_0}{2} P^*(k_t); \quad \forall k_t > 0$$

$$P^*(k_t) = \frac{1}{Q_5(0)} \left[\gamma P^*(0) Q_5(k_t) + \frac{1}{k_t} \sum_{r_t=0}^{k_t-1} [(k_t - r_t) (\gamma P^*(r_t) Q_5(k_t - r_t) - Q_5(r_t) P^*(k_t - r_t))] \right]$$

These functions are initialized as follows:

$$\Phi_{ij}(0) = Q_i(0) Q_j(0) / Q_1(0)$$

$$P(0) = C_0 P^*(0)$$

$$P^*(0) = Q_5(0)^Y$$

The summations used to compute the DTs for $\Phi_{ij}(k_t)$ and $P^*(k_t)$ include the actual values $\Phi_{ij}(k_t)$ and $P^*(k_t)$ themselves, even though they are not technically defined by that point. In these cases when a function's DT depends on itself, it is assumed to be zero in the summation. This can be implemented in practice by initializing each of these functions to zero before entering the loops to compute the DTs.

Implementation Specifics

Given the initial DTs, $\mathbf{Q}(0)$, which are the values of \mathbf{q} at a given spatial GLL point, one can compute the zeroth-order DTs of the flux and source terms. From there, a spatial derivative of the zeroth-order flux DTs must be computed, and then one can compute $\mathbf{Q}(1)$. Then, iteratively, higher-order DTs of the state, flux, and source vectors can be computed accordingly. All that is left is creating an efficient means of computing a spatial derivative, $\partial_x \mathbf{F}(k_t)$.

To compute the a single spatial derivative of $\mathbf{F}(k_t)$, an $N_T \times N_T$ matrix, \mathcal{M}_D , is created that transforms the N_T GLL points over the cell into N_T polynomial coefficients, differentiates the polynomial, and then casts the differentiated polynomial coefficients back into N_T GLL points across the cell. Since multiple, successive linear operators can be merged, this can be done with a single matrix-vector multiply. To compute the matrix, \mathcal{M}_D , let us first define an $(N_T - 1)$ th-order polynomial as $p(x) = \sum_{i=0}^{N_T-1} a_i x^i$, where the polynomial is defined over a local coordinate $x \in [-1/2, 1/2]$. Then, define two vectors of constraints:

$$a'_i = \begin{cases} i a_{i+1} & \text{if } 0 \leq i \leq N_T - 1 \\ 0 & \text{if } i = N_T \end{cases}$$

$$c_{GLL,i} = \frac{1}{\Delta x} p(\xi_i); \quad \forall i \in \{0, \dots, N_T - 1\}$$

where ξ_i is the i th GLL point on the domain $[-1/2, 1/2]$. The matrix that transforms GLL points to coefficients, differentiates those coefficients, and transforms them back into GLL points is defined as:

$$\mathcal{M}_D = \frac{\partial c_{GLL}}{\partial \mathbf{a}} \frac{\partial \mathbf{a}'}{\partial \mathbf{a}} \left(\frac{\partial c_{GLL}}{\partial \mathbf{a}} \right)^{-1}$$

This matrix is pre-computed using symbolic mathematical software before the simulation is run.

Algorithmic Simplifications

Traditionally, ADER methods perform a Riemann solve not only for the flux (using more traditional solvers such as HLLC [63]), but also for the temporal derivatives as well. This sets up high-order Derivative Riemann Problems that are typically solved using the locally constant linear upwind state. In this implementation, however, the algorithm is simplified by computing a single, traditional Riemann solve on the *time-averaged* state. This has two advantages over traditional ADER approaches: (1) only one Riemann solve needs to be performed per cell edge per time step, and (2) less data needs to be communicated between parallel nodes. Note that this simplification may be less accurate for problems with strong shocks, but for the weaker discontinuities seen in buoyancy-driven atmospheric flows, it sufficiently accurate.

2.5 | Boundary Conditions

In all test cases in this study, the vertical boundaries are assumed to be no-slip solid walls, and the horizontal boundaries are assumed to be periodic. For the vertical solid wall boundaries, hydrostasis is computed into the ghost cells. Then, for all variables except vertical wind, the wind values and potential temperature of the cell nearest the boundary is copied

into each of the ghost cells by multiplying the winds and potential temperature by hydrostatic density in the ghost cell. The vertical wind is set to zero in the ghost cells. Further, when GLL points of the state and flux are computed, the vertical component coinciding with the vertical boundaries is set to zero to enforce a no-flux boundary condition. Setting the vertical wind to zero in the vertical ghost cells acts to help the recovery operators near the vertical boundaries to approach zero naturally so that setting the actual boundary samples to zero does not create oscillations. The Straka density current test case is quite sensitive to boundary condition implementation, and it gave the expected answers with this approach.

2.6 | 2-D Option

Since there are a fair number of test cases in literature in only two spatial dimensions, the code can switch to 2-D mode with a few simple modifications. First, only one cell is used in the y -dimension, and the y location is assumed to be the middle of the y -direction domain. Next, the v -velocity is initialized to zero. Finally, the y -direction update in the dimensionally-split time stepping is not performed. Therefore, the state vector never feels the y -direction forcing, and v remains zero for all time.

2.7 | Parallel Implementation

This code uses domain decomposition in both horizontal dimensions and the Message Passing Interface (MPI) in order to distribute the work across multiple processes. For I/O, the parallel-netcdf [32] library is used. The parts of the code that require MPI data transfers are as follows:

- The computation of the maximum wind requires an `MPI_AllReduce` so that each process proceeds with the same time step. This is only performed during initialization because atmospheric flows do not have a large variance in wind speeds that would lead to significantly greater CFL values than are seen at initialization (not compared to acoustic wave speeds anyway).
- 1-D halo exchanges before tendency calculations in each horizontal direction are performed with point-to-point MPI (e.g., `MPI_Isend` and `MPI_Irecv`). $5C_z C_t h$ values are passed across each domain decomposition boundary, C_z is the number of vertical cells, C_t is the number of cells in the transverse direction, and $h = (N - 1) / 2$ is the number of halo cells with N being the spatial order of accuracy.
- 1-D exchanges of fluxes at cell interfaces coinciding with domain decomposition boundaries are performed with point-to-point MPI. $10C_z C_t$ values are passed along each domain decomposition boundary.
- 1-D exchanges of outgoing mass fluxes at interfaces coinciding with domain decomposition boundaries are performed with point-to-point MPI. $7C_z C_t$ values are passed along each domain decomposition boundary.

3 | NUMERICAL TEST CASES AND VALIDATION

This study uses literature standard test cases to validate the expected behavior of the non-hydrostatic model. While there are no exact test cases for this framework, there are both quantitative and qualitative aspects for each case that demonstrate expected answers. All experiments use an initial CFL value of 0.8 and use a constant time step throughout the simulation. In this section, only the fifth-order-accurate 2-D version of the model with WENO limiting is used to validate the model against expected literature results (e.g., [50] and references therein). The next section is used to

investigate the effects of WENO limiting and order of accuracy.

3.1 | Hydrostatic Initializations

To initialize a hydrostatic atmosphere, N -point Gauss-Legendre quadrature is used to initialize cell means for an N th-order-accurate method to ensure the atmosphere is initially balanced. For cell boundaries, exact estimates of the hydrostatic background state are sampled. Hydrostasis is a function of the vertical dimension only.

3.1.1 | Neutral Atmosphere

To initialize hydrostatic balance for a neutral atmosphere with constant potential temperature, θ_0 , it is easiest to obtain a vertical profile for Exner pressure, π , rather than pressure directly. Exner pressure is a function of pressure only, given by:

$$\pi = \left(\frac{p}{p_0} \right)^{R_d/c_p} \quad (3)$$

And its hydrostatic balance equation is given in terms of only potential temperature:

$$\frac{d\pi}{dz} = -\frac{g}{c_p\theta} \quad (4)$$

The first two test cases assume a constant potential temperature basic state. Therefore, the hydrostatically balanced Exner pressure profile is trivial:

$$\bar{\pi}(z) = \bar{\pi}_{sf c} - \frac{g z}{c_p \theta_0} \quad (5)$$

where it is assumed that $\bar{\pi}_{sf c} = 1$ meaning $p = p_0$ at the surface.

3.1.2 | Constant Brunt-Vaisala Frequency

To initialize a stable atmosphere, a constant Brunt-Vaisala frequency, N_0 , is specified. The Brunt-Vaisala frequency is given in terms of fractional vertical gradient of potential temperature:

$$\sqrt{\frac{g}{\theta} \frac{d\theta}{dz}} = N_0$$

Therefore,

$$\theta(z) = \theta_{sf c} e^{\frac{N_0^2}{g} z}$$

Plugging this into (4), one eventually obtains the following vertical profile:

$$\bar{\pi}(z) = \bar{\pi}_{sf c} - \frac{g^2}{c_p N_0^2} \left(\frac{\theta(z) - \theta_{sf c}}{\theta(z) \theta_{sf c}} \right) \quad (6)$$

The constants are set as follows: $\bar{\pi}_{sf_c} = 1$, $\theta_{sf_c} = 300$ K, and $N_0 = 10^{-2} \text{ s}^{-1}$.

Pressure can be obtained from (5) or (6) by the Exner pressure equation, (3). Then, density is determined from the pressure by the equation of state.

3.2 | Rising Thermal

The rising thermal test case perturbs the potential temperature of a neutrally stratified atmosphere as follows:

$$\theta(x, y, z) = \theta_H(z) + \widehat{\theta} \max(0, 1 - d)$$

$$d = \sqrt{\left(\frac{x - x_0}{x_R}\right)^2 + \left(\frac{y - y_0}{y_R}\right)^2 + \left(\frac{z - z_0}{z_R}\right)^2}$$

where $\widehat{\theta} = 2$ K is the amplitude of the perturbation, $(x_0, y_0, z_0) = (10 \text{ km}, 10 \text{ km}, 2 \text{ km})$ is the center of the thermal, and $x_R = y_R = z_R = 2$ km is the radius of the thermal. This test case is simulated on a domain of $[0, 20] \times [0, 20] \times [0, 10]$ km in the x -, y -, and z -directions for 1,000 seconds. Winds are initially at rest. For a 2-D domain, the y coordinate is always set to y_0 .

Plots of the solution are given in Figure 1 using a fifth-order-accurate WENO ADER-DT scheme with 100m grid spacing. The results fit well with literature precedent. Quantitative comparison is difficult since it's a non-linear flow regime wherein small initial condition differences and numerical differences eventually lead to large differences in the flow. However, the general structure matches other studies.

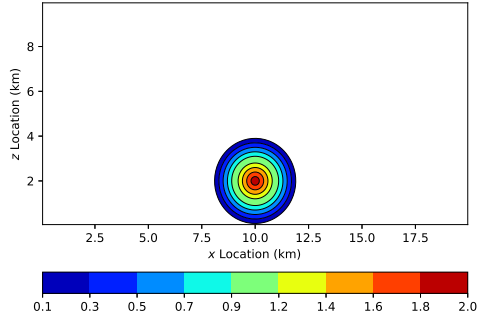
3.3 | Straka Density Current

The Straka density current test case perturbs the potential temperature of a neutrally stratified atmosphere as follows:

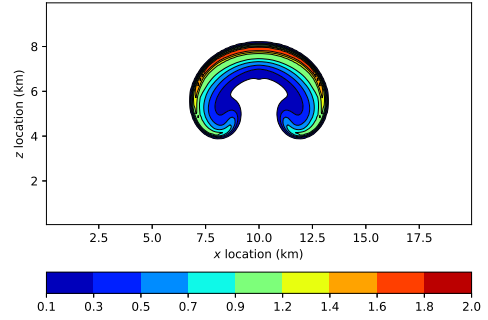
$$\theta(x, y, z) = \theta_H(z) + \widehat{\theta} \begin{cases} (\cos(\pi d) + 1)/2 & \text{if } d \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$d = \sqrt{\left(\frac{x - x_0}{x_R}\right)^2 + \left(\frac{y - y_0}{y_R}\right)^2 + \left(\frac{z - z_0}{z_R}\right)^2}$$

where $\widehat{\theta} = -15$ K is the amplitude of the perturbation, $(x_0, y_0, z_0) = (26.5 \text{ km}, 26.5 \text{ km}, 3 \text{ km})$ is the center of the cold bubble, and $x_R = y_R = 4$ km and $z_R = 2$ km are the radii of the cold bubble in the horizontal and vertical directions, respectively. This test case is simulated on a domain of $[0, 53] \times [0, 53] \times [0, 6.4]$ km in the x -, y -, and z -directions for 900 seconds. For a 2-D domain, the y coordinate is always set to y_0 . This test case is traditionally run with a dynamic



(a) Potential Temperature, initial



(b) Potential Temperature, 1,000 seconds

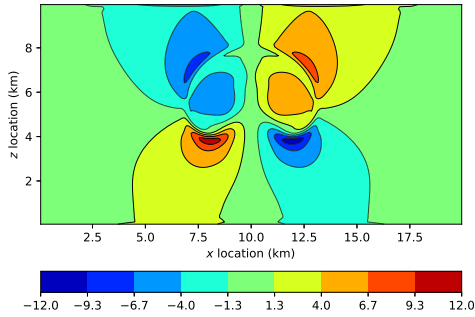
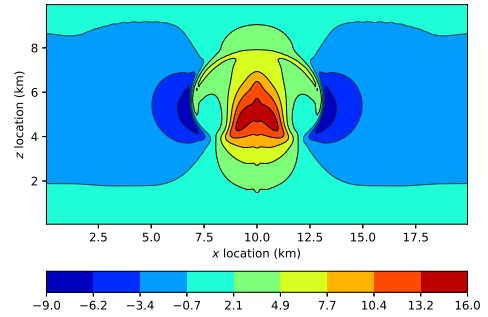
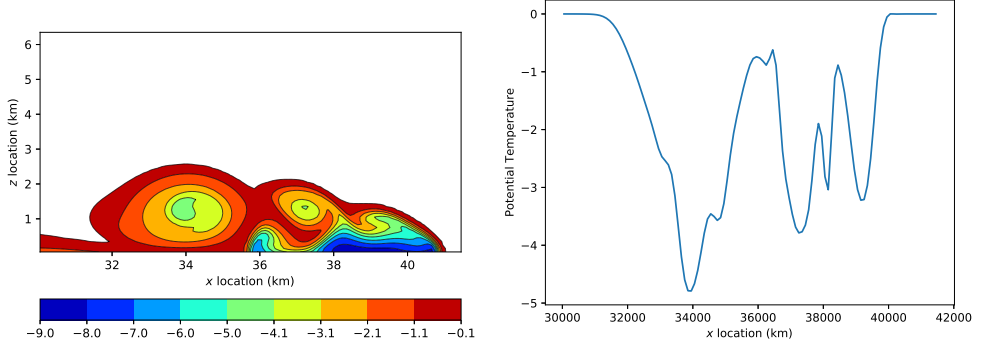
(c) u -wind, 1,000 seconds(d) w -wind, 1,000 seconds

FIGURE 1 Solution plots for the 2-D rising thermal test case with a fifth-order-accurate WENO ADER-DT scheme using 200×100 cells.



(a) Potential temperature contour plot

(b) Potential temperature at $z = 1,200$ m

FIGURE 2 Potential temperature after 900s for the 2-D viscous Straka density current test case with a fifth-order-accurate WENO ADER-DT scheme using 530×64 cells.

viscosity defined as:

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho u \\ \rho v \\ \rho w \\ \rho \theta \end{bmatrix} = \rho K \begin{bmatrix} \partial_{xx} u + \partial_{yy} u + \partial_{zz} u \\ \partial_{xx} v + \partial_{yy} v + \partial_{zz} v \\ \partial_{xx} w + \partial_{yy} w + \partial_{zz} w \\ \partial_{xx} \theta + \partial_{yy} \theta + \partial_{zz} \theta \end{bmatrix}$$

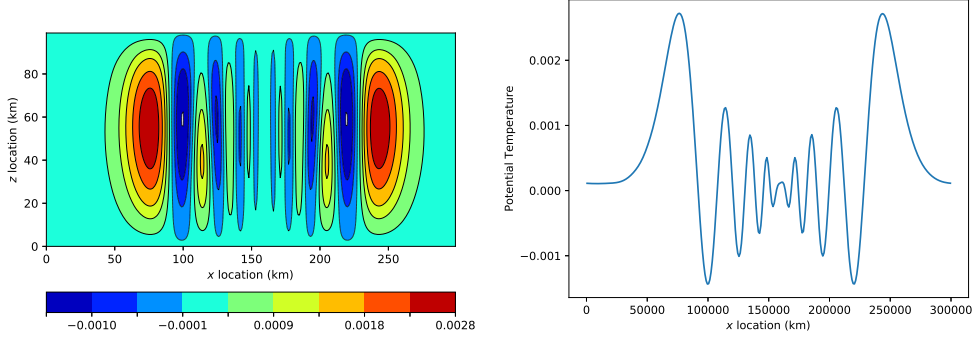
where $K = 75 \text{ m}^2 \text{ s}^{-1}$. However, it can be run without viscosity as well. Winds are initially at rest. The potential temperature is plotted in Figure 2 using a 100m grid spacing. The minimum and maximum potential temperature perturbations at 900s were -8.65K and 0.015K, respectively. These match literature expectations.

3.4 | Internal Gravity Waves

The internal gravity wave test case perturbs the potential temperature of a stably stratified atmosphere (with $N_0 = 10^{-2}$) as follows:

$$\theta(x, y, z) = \theta_H(z) + \hat{\theta} \frac{h_R^2 \sin(\pi z / z_T)}{(h_R^2 + (x - x_0)^2 + (y - y_0)^2)}$$

where $\hat{\theta} = 10^{-2} \text{ K}$ is the amplitude of the perturbation, $(x_0, y_0) = (100 \text{ km}, 100 \text{ km})$ is the center of the perturbation, and $h_R = 5 \text{ km}$ is the horizontal radius. This test case is simulated on a domain of $[0, 300] \times [0, 300] \times [0, 10] \text{ km}$ in the x -, y -, and z -directions for 3,000 seconds. For a 2-D domain, the y coordinate is always set to y_0 . The horizontal winds are initialized to $u = v = 20 \text{ m s}^{-1}$ (where v is set to zero in 2-D simulations). Typically, this test case is solved with the horizontal to vertical grid spacing ration set to 10:1. Plots of the potential temperature are given in Figure 3 after 3,000s using 100m and 1km grid spacing in the vertical and horizontal directions, respectively. Again, these solutions match literature precedent.



(a) Potential Temperature contour plot, 3,000s

(b) Potential Temperature at $z = 5$ km, 3,000s

FIGURE 3 Solution plots for the 2-D internal gravity wave test case after 3,000s with a fifth-order-accurate WENO ADER-DT scheme using 300×100 cells.

3.5 | Colliding Thermals

The colliding thermals test case perturbs the potential temperature of a neutrally stratified atmosphere as follows:

$$\theta(x, y, z) = \theta_H(z) + \hat{\theta} \begin{cases} \cos^2(\pi d_1) & \text{if } d_1 \leq \frac{\pi}{2} \\ -\cos^2(\pi d_2) & \text{if } d_2 \leq \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases}$$

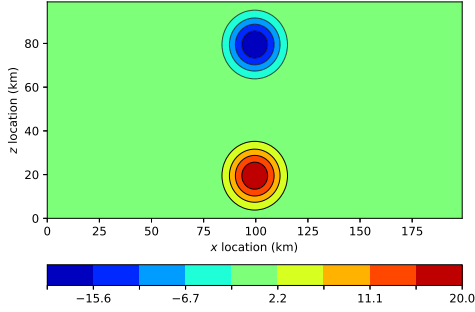
$$d_1 = \frac{\pi}{2} \sqrt{\left(\frac{x-x_1}{r}\right)^2 + \left(\frac{y-y_1}{r}\right)^2 + \left(\frac{z-z_1}{r}\right)^2}$$

$$d_2 = \frac{\pi}{2} \sqrt{\left(\frac{x-x_2}{r}\right)^2 + \left(\frac{y-y_2}{r}\right)^2 + \left(\frac{z-z_2}{r}\right)^2}$$

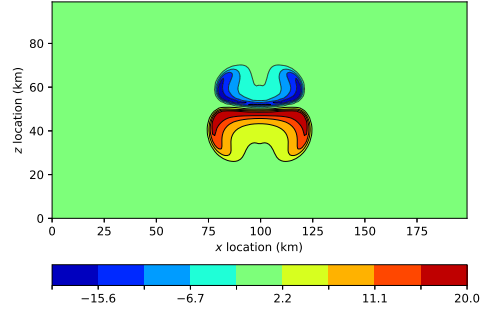
where $\hat{\theta} = 20$ K is the amplitude of the perturbation, $(x_1, y_1, z_1) = (10 \text{ km}, 10 \text{ km}, 2 \text{ km})$ is the center of the warm bubble, $(x_2, y_2, z_2) = (10 \text{ km}, 10 \text{ km}, 8 \text{ km})$ is the center of the cold bubble, and $r = 2$ km is the radius of the bubbles. This test case is simulated on a domain of $[0, 20] \times [0, 20] \times [0, 10]$ km in the x -, y -, and z -directions for 700 seconds. Winds are initially at rest. For a 2-D domain, the y coordinate is always set to y_0 . Plots of potential temperature after 0s, 200s, 400, and 700s are given in Figure 4 using 100m grid spacing. This is not a literature-standard test case, but it was added to generate oscillations and show detail in 2-D stratified turbulence visually.

4 | RESULTS AND DISCUSSION

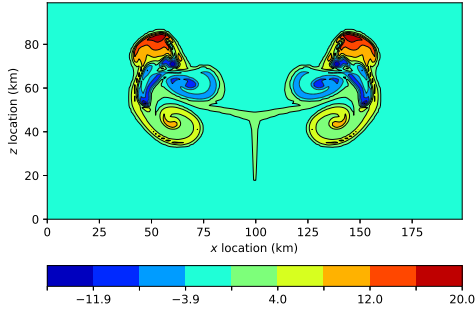
In this section, only odd orders of accuracy are used (i.e., 3rd, 5th, 7th, and 9th order). This is because a cell-centered approach naturally contains an odd number of cells to use for recovery. It would, similarly, be more natural for a



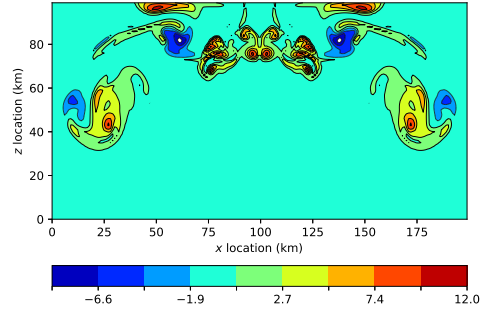
(a) Potential Temperature, initial



(b) Potential Temperature, 200s



(c) Potential Temperature, 400s



(d) Potential Temperature, 700s

FIGURE 4 Solution plots for the 2-D colliding thermals test case with a fifth-order-accurate WENO ADER-DT scheme using 200×100 cells.

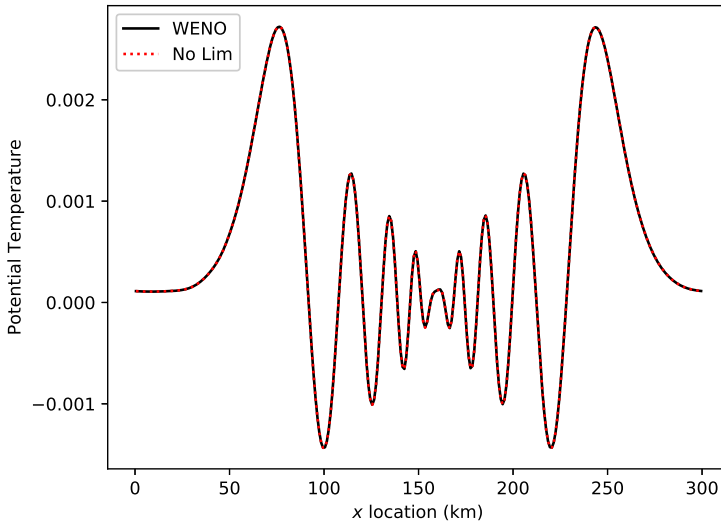


FIGURE 5 Plot of potential temperature at $z = 5$ km with and without WENO limiting for the internal gravity wave test case using 300×100 cells.

cell-edge-centered approach (for direct flux recovery) to use an even-ordered accuracy.

4.1 | WENO vs. No Limiter

Figure 5 shows a plot of potential temperature perturbation for the internal gravity waves test case at fifth-order accuracy with and without WENO limiting at $z = 5$ km. The overlapping lines demonstrate that for smooth flows, there is no effect when applying the WENO limiter, a designed property of WENO limiting.

Figures 6 and 7 show contour plots and line plots at $x = 10$ km of potential temperature perturbations with and without WENO limiting to show the effects of WENO limiting for flows that become discontinuous at the front of the colliding thermals. It slightly reduces the extrema and significantly reduces the visible $2\Delta x$ oscillations. Yet, the larger scale flow remains untouched.

Figure 8 shows the kinetic energy spectra for the collision test case after 700 seconds of simulation to demonstrate what happens to the resolved kinetic energy after applying WENO limiting. Clearly, only the KE spectra at $2\Delta x$ are affected by the limiting, while scales of $4\Delta x$ and higher are left untouched. In the non-limited simulation, it's clear that a small amount of $2\Delta x$ energy is accumulating in the simulation. It's not enough to generate instability, but it does generate oscillations. After WENO limiting, the KE spectra now tails off, which is the desired behavior for a stable simulation.

4.2 | Order of Accuracy

Figure 9 shows the vertical wind for the rising thermal test case over varying orders of accuracy from third- to ninth-order. Notice the increasing detail in the thermal center in particular. Of course, there is a limit to how much order of accuracy can help, particularly for smoother flows, since they are by definition already fairly well resolved.

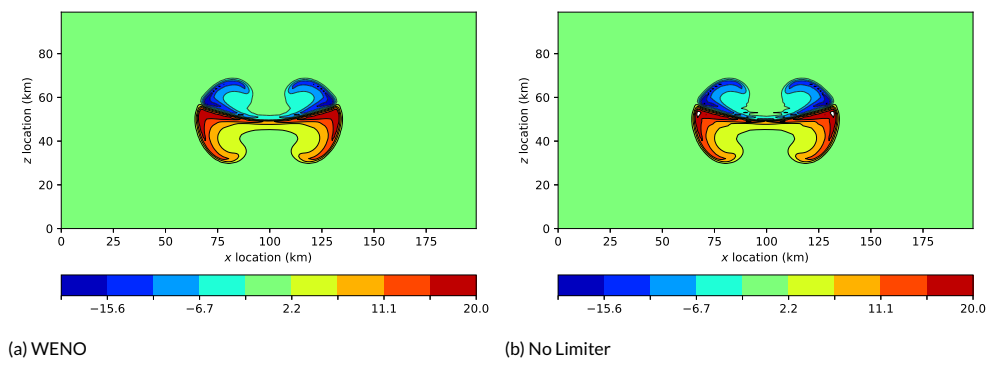


FIGURE 6 Plot of potential temperature after 250s with and without WENO limiting for the colliding thermals test case using 200×100 cells.

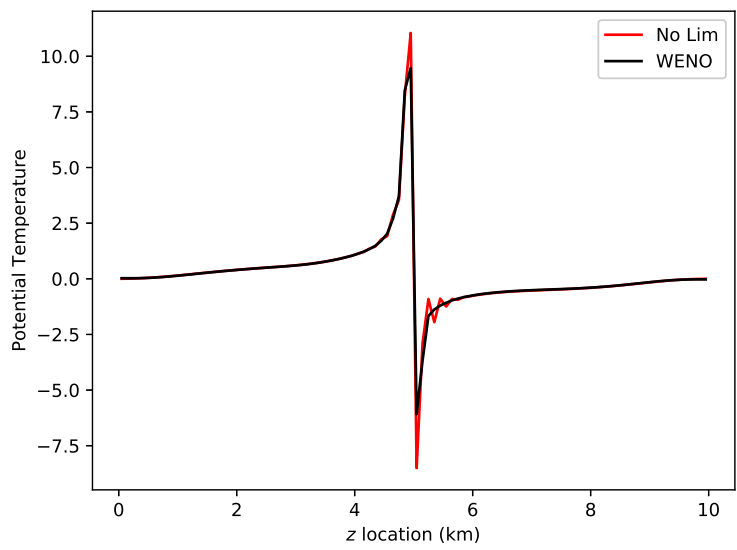


FIGURE 7 Plot of potential temperature at $x = 10$ km at 250 seconds with and without WENO limiting for the colliding thermals test case using 200×100 cells.

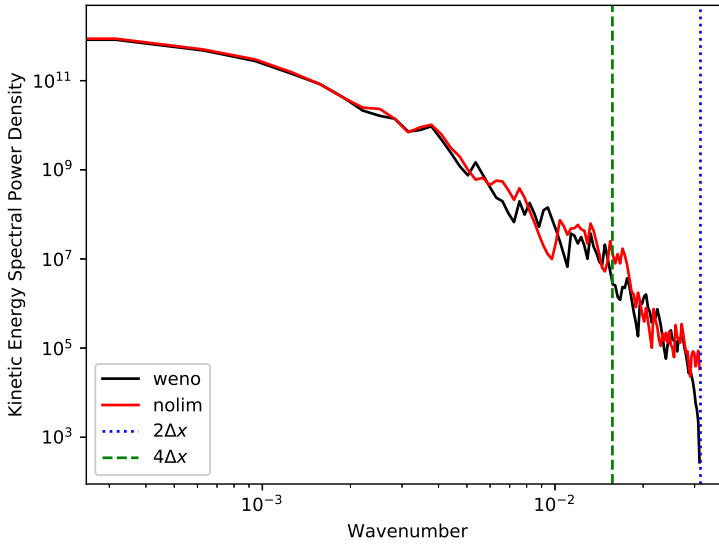


FIGURE 8 Plot of kinetic energy spectral power density with and without WENO limiting at 700 seconds for the colliding thermals test case using 200×100 cells with markers for $2\Delta x$ and $4\Delta x$ waves.

Figure 10 shows plots of potential temperature for the collision test case after 700 seconds of simulation. Again, notice the increasing amount of detail as the order of accuracy increases. Since this is a chaotic fluid that is now in a turbulent regime, the results are expected to diverge due to different numerical properties. In particular, a greater amount of detail and resolved mixing are apparent in the flow.

5 | CONCLUSIONS AND FUTURE WORK

Again, the hypothesis being explored in this study is that high-order accuracy, when *properly limited*, can provide increasing resolution even for complex and discontinuous flows. This is furthering the study in [48] that explores this hypothesis in the context of transport. A survey of existing techniques was given in Section 1. The mathematical discretization was described in detail in Section 2. Basic results of the new model at fifth-order accuracy to compare against literature-standard test cases was given in Section 3. An extension of studying the numerical method's behavior in relation to limiting and higher-order accuracy was given in Section 4.

It was found that this method gives the expected literature-standard results in common non-hydrostatic test cases. Further, it was discovered that the WENO limiting only changes the very smallest wavelengths present in the model at scales barely larger than $2\Delta x$. Without limiting, there are small $2\Delta x$ oscillations that result, causing accumulation of energy at the smallest scales. While this accumulation is not sufficient to make the model unstable, it is known that oscillations can cause unphysical behavior in physical parameterization schemes. It was shown that WENO limiting effectively removed these oscillations while leaving scales larger than $2\Delta x$ alone. Also, it was shown that going to high-order accuracy, even with WENO limiting, led to an increase in detail and resolved mixing, adding to the physical realism of the Euler equations.

While a detailed description of the code's C++ implementation in Kokkos [13] and analysis of this code's performance

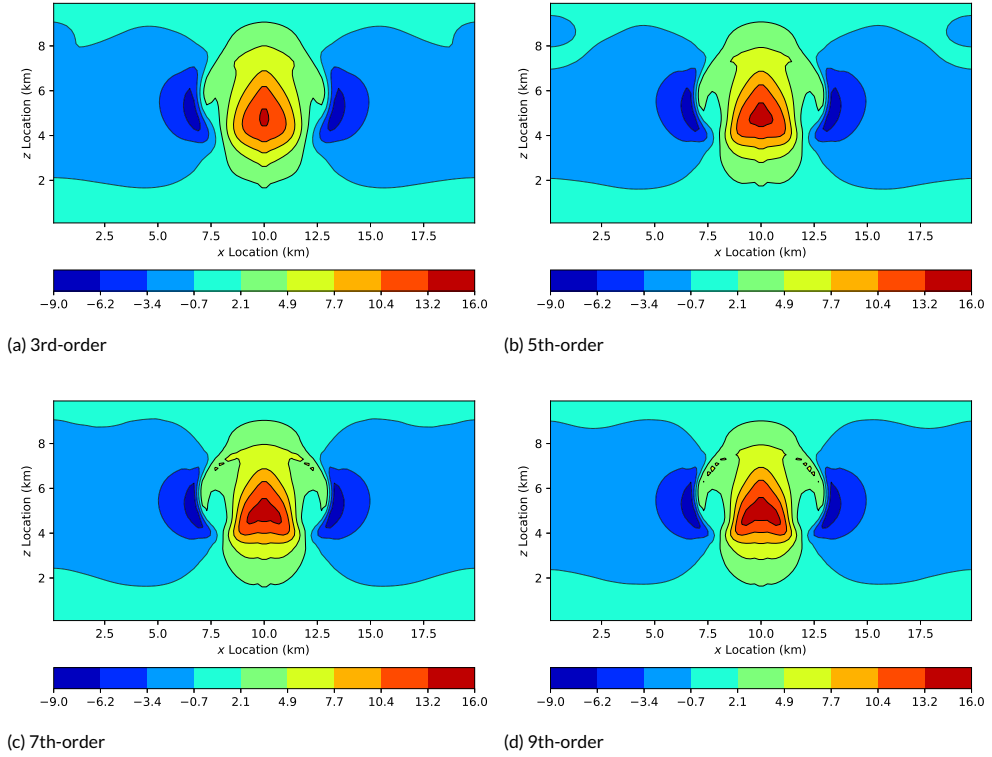
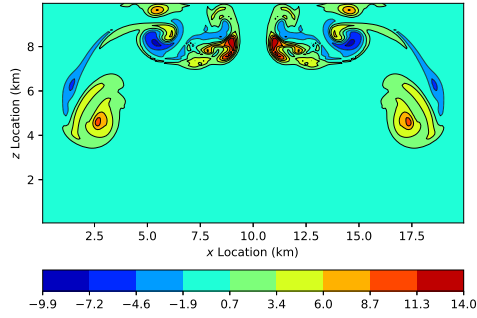
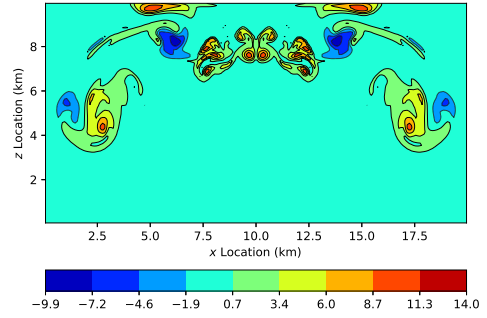


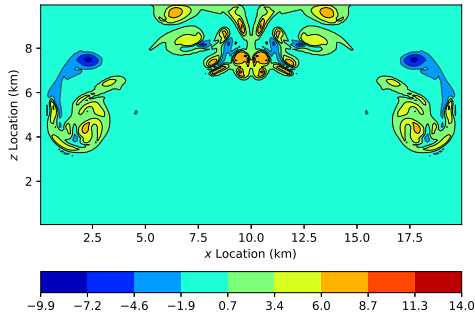
FIGURE 9 Plots of vertical wind for the rising thermal test case after 1,000 seconds of simulation with 100×50 cells at different orders of accuracy.



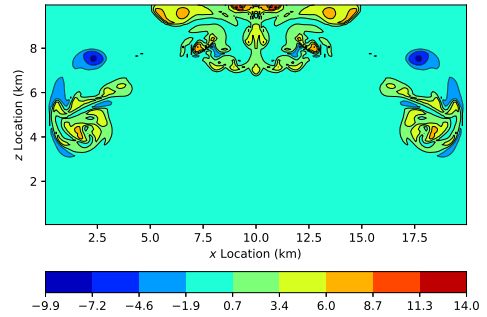
(a) 3rd-order



(b) 5th-order



(c) 7th-order



(d) 9th-order

FIGURE 10 Plots of potential temperature for the colliding thermals test case with 200×100 cells at different orders of accuracy.

on accelerators is left to separate publication, it is worth mentioning a few interesting computational findings. First, the WENO limiting was only marginally more expensive than simulation without limiting, even though it requires several times more computations. The reason is those computations use data that has already been transferred from main memory and can be performed easily within fast, on-chip register and cache reserves. Therefore, the limiting essentially comes for free. Further, ninth-order of accuracy was only marginally more expensive than third-order accuracy for the same reasons. While ninth-order requires about 10× more computations (without considering limiting), those computations are done locally inside register and cache reserves. Finally, on the Nvidia V100 GPU, with enough cells per GPU, the most expensive kernel in the ninth-order-accurate WENO-limited configuration ran at 80% of the peak possible single precision floating point operations per second (flop/s), which is close to the vaunted Top500 benchmark's throughput¹, albeit for only part of the algorithm. Therefore, high-order WENO-limited ADER-DT simulation also improves computational performance on accelerators.

Future directions planned for this research include investigating a multi-stage implementation of the ADER-DT algorithm in hopes of a simple dimensional splitting in non-orthogonal coordinates similar to [24]. The computational expense of an unsplit approach will also be investigated. Next, the inevitable expansion into global spatial scales will require implementing HEVI techniques, which have already been evaluated at length in various contexts [68, 15]. This will also require further investigation into how to couple a Diagonally Implicit Runge-Kutta (DIRK) scheme [1] with an ADER-DT explicit discretization, if the ADER approach continues to prove worthwhile in that context. This strongly interacts with the potential development of a multi-stage realization of ADER-DT to couple the implicit and explicit operators more closely.

APPENDIX A: EULER EQUATION CHARACTERISTICS

Here, the flux Jacobians and their diagonalizations in each coordinate direction are given, where $c_s = \sqrt{\gamma p/\rho}$ is the speed of sound. In each direction, the flux Jacobian is diagonalized into $R\Lambda L$, where $L = R^{-1}$.

$$\frac{\partial \mathbf{f}}{\partial \mathbf{q}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -u^2 & 2u & 0 & 0 & \frac{c_s^2}{\theta} \\ -uv & v & u & 0 & 0 \\ -uw & w & 0 & u & 0 \\ -\theta u & \theta & 0 & 0 & u \end{bmatrix} ; \quad \Lambda_x = \begin{bmatrix} -c_s + u & 0 & 0 & 0 & 0 \\ 0 & c_s + u & 0 & 0 & 0 \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & 0 & 0 & 0 & u \end{bmatrix}$$

$$R_x = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ -c_s + u & c_s + u & u & 0 & 0 \\ v & v & 0 & 1 & 0 \\ w & w & 0 & 0 & 1 \\ \theta & \theta & 0 & 0 & 0 \end{bmatrix} ; \quad L_x = \begin{bmatrix} \frac{u}{2c_s} & -\frac{1}{2c_s} & 0 & 0 & \frac{1}{2\theta} \\ -\frac{u}{2c_s} & \frac{1}{2c_s} & 0 & 0 & \frac{1}{2\theta} \\ 1 & 0 & 0 & 0 & -\frac{1}{\theta} \\ 0 & 0 & 1 & 0 & -\frac{v}{\theta} \\ 0 & 0 & 0 & 1 & -\frac{w}{\theta} \end{bmatrix}$$

¹<https://www.top500.org>

$$\frac{\partial \mathbf{g}}{\partial \mathbf{q}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ -uv & v & u & 0 & 0 \\ -v^2 & 0 & 2v & 0 & \frac{c_s^2}{\theta} \\ -vw & 0 & w & v & 0 \\ -\theta v & 0 & \theta & 0 & v \end{bmatrix}; \quad \Lambda_y = \begin{bmatrix} -c_s + v & 0 & 0 & 0 & 0 \\ 0 & c_s + v & 0 & 0 & 0 \\ 0 & 0 & v & 0 & 0 \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & 0 & 0 & v \end{bmatrix}$$

$$R_y = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ u & u & 0 & 1 & 0 \\ -c_s + v & c_s + v & v & 0 & 0 \\ w & w & 0 & 0 & 1 \\ \theta & \theta & 0 & 0 & 0 \end{bmatrix}; \quad L_y = \begin{bmatrix} \frac{v}{2c_s} & 0 & -\frac{1}{2c_s} & 0 & \frac{1}{2\theta} \\ -\frac{v}{2c_s} & 0 & \frac{1}{2c_s} & 0 & \frac{1}{2\theta} \\ 1 & 0 & 0 & 0 & -\frac{1}{\theta} \\ 0 & 1 & 0 & 0 & -\frac{u}{\theta} \\ 0 & 0 & 0 & 1 & -\frac{w}{\theta} \end{bmatrix}$$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{q}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ -uw & w & 0 & u & 0 \\ -vw & 0 & w & v & 0 \\ -w^2 & 0 & 0 & 2w & \frac{c_s^2}{\theta} \\ -\theta w & 0 & 0 & \theta & w \end{bmatrix}; \quad \Lambda_z = \begin{bmatrix} -c_s + w & 0 & 0 & 0 & 0 \\ 0 & c_s + w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & w & 0 \\ 0 & 0 & 0 & 0 & w \end{bmatrix}$$

$$R_z = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ u & u & 0 & 1 & 0 \\ v & v & 0 & 0 & 1 \\ -c_s + w & c_s + w & w & 0 & 0 \\ \theta & \theta & 0 & 0 & 0 \end{bmatrix}; \quad L_z = \begin{bmatrix} \frac{w}{2c_s} & 0 & 0 & -\frac{1}{2c_s} & \frac{1}{2\theta} \\ -\frac{w}{2c_s} & 0 & 0 & \frac{1}{2c_s} & \frac{1}{2\theta} \\ 1 & 0 & 0 & 0 & -\frac{1}{\theta} \\ 0 & 1 & 0 & 0 & -\frac{u}{\theta} \\ 0 & 0 & 1 & 0 & -\frac{v}{\theta} \end{bmatrix}$$

APPENDIX B: WENO LIMITING

WENO limiting is depicted visually in Figure 11 with a discontinuity at $x = 1/2$. In this figure, a fifth-order-accurate polynomial is formed using five cell averages, and three quadratic polynomials are formed as well. The resulting polynomial is less oscillatory than the high-order polynomial because the candidate polynomials weights are inversely proportional to their Total Variation, giving the smoothest polynomials the largest weights.

ACKNOWLEDGEMENTS

This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

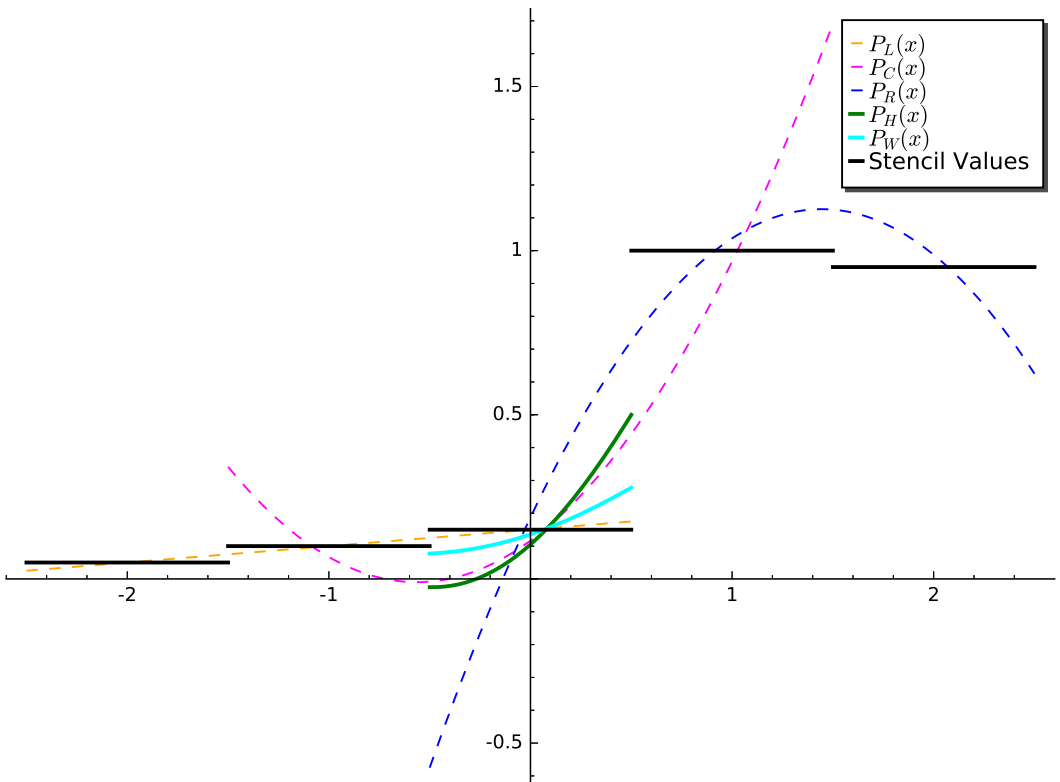


FIGURE 11 Example of WENO limiting, assuming a grid spacing of 1, where $P_L(x)$, $P_C(x)$, and $P_R(x)$ are the left-biased, centered, and right-biased lower-ordered polynomials, respectively; and $P_H(x)$ and $P_W(x)$ are the high-order and WENO polynomials over the center cell's domain. Thick black lines denote the stencil average values for this example. Note the domains of P_L , P_C , and P_R all overlap in the center cell.

ONLINE RESOURCES

The code used to generate the results presented in this paper is publicly available on github at <https://github.com/mrnorman/awflCloud>. It is affectionately named the [A]DER [W]ENO [F]inite-Vo[L]ume (AWFL) method.

REFERENCES

- [1] Alexander, R. (1977) Diagonally implicit runge-kutta methods for stiff ode's. *SIAM Journal on Numerical Analysis*, **14**, 1006–1021.
- [2] Arakawa, A. and Lamb, V. R. (1981) A potential enstrophy and energy conserving scheme for the shallow water equations. *Monthly Weather Review*, **109**, 18–36.
- [3] Balsara, D. S., Rumpf, T., Dumbser, M. and Munz, C.-D. (2009) Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics. *Journal of Computational Physics*, **228**, 2480–2516.
- [4] Bresten, C., Gottlieb, S., Grant, Z., Higgs, D., Ketcheson, D. and Németh, A. (2017) Explicit strong stability preserving multistep runge-kutta methods. *Mathematics of Computation*, **86**, 747–769.
- [5] Chen, Y., Weller, H., Pring, S. and Shaw, J. (2017) Comparison of dimensionally split and multi-dimensional atmospheric transport schemes for long time steps. *Quarterly Journal of the Royal Meteorological Society*, **143**, 2764–2779.
- [6] Christara, C., Ding, X. and Jackson, K. (2002) An efficient transposition algorithm for distributed memory computers. In *High Performance Computing Systems and Applications*, 349–370. Springer.
- [7] Colella, P. and Woodward, P. R. (1984) The piecewise parabolic method (ppm) for gas-dynamical simulations. *Journal of computational physics*, **54**, 174–201.
- [8] Cotter, C. J. and Shipton, J. (2012) Mixed finite elements for numerical weather prediction. *Journal of Computational Physics*, **231**, 7076–7091.
- [9] Courant, R., Friedrichs, K. and Lewy, H. (1967) On the partial difference equations of mathematical physics. *IBM journal of Research and Development*, **11**, 215–234.
- [10] Dennis, J. M., Edwards, J., Evans, K. J., Guba, O., Lauritzen, P. H., Mirin, A. A., St-Cyr, A., Taylor, M. A. and Worley, P. H. (2012) Cam-se: A scalable spectral element dynamical core for the community atmosphere model. *The International Journal of High Performance Computing Applications*, **26**, 74–89.
- [11] Dumbser, M., Balsara, D. S., Toro, E. F. and Munz, C.-D. (2008) A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. *Journal of Computational Physics*, **227**, 8209–8253.
- [12] Durran, D. R. (2010) *Numerical methods for fluid dynamics: With applications to geophysics*, vol. 32. Springer Science & Business Media.
- [13] Edwards, H. C., Trott, C. R. and Sunderland, D. (2014) Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. *Journal of Parallel and Distributed Computing*, **74**, 3202 – 3216. URL: <http://www.sciencedirect.com/science/article/pii/S0743731514001257>. Domain-Specific Languages and High-Level Frameworks for High-Performance Computing.
- [14] Fournier, A., Taylor, M. A. and Tribbia, J. J. (2004) The spectral element atmosphere model (SEAM): High-resolution parallel computation and localized resolution of regional dynamics. *Monthly Weather Review*, **132**, 726–748.
- [15] Gardner, D. J., Guerra, J. E., Hamon, F. P., Reynolds, D. R., Ullrich, P. A. and Woodward, C. S. (2018) Implicit-explicit (imex) runge-kutta methods for non-hydrostatic atmospheric models.

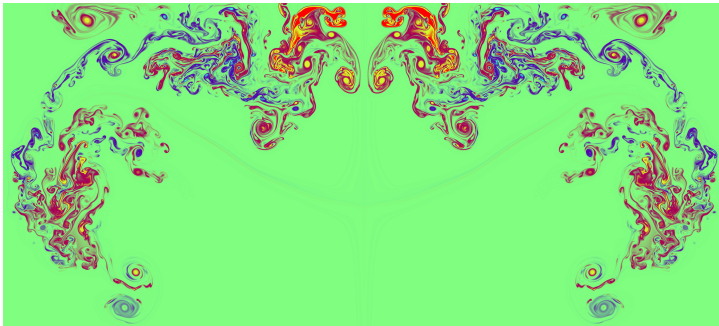
- [16] Giraldo, F., Perot, J. and Fischer, P. (2003) A spectral element semi-lagrangian (sesl) method for the spherical shallow water equations. *Journal of Computational Physics*, **190**, 623–650.
- [17] Giraldo, F. X., Hesthaven, J. S. and Warburton, T. (2002) Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations. *Journal of Computational Physics*, **181**, 499–525.
- [18] Gottlieb, S. (2005) On high order strong stability preserving Runge–Kutta and multi step time discretizations. *Journal of Scientific Computing*, **25**, 105–128.
- [19] Guba, O., Taylor, M. and St-Cyr, A. (2014) Optimization-based limiters for the spectral element method. *Journal of Computational Physics*, **267**, 176–195.
- [20] Guo, W., Nair, R. D. and Zhong, X. (2016) An efficient weno limiter for discontinuous galerkin transport scheme on the cubed sphere. *International Journal for Numerical Methods in Fluids*, **81**, 3–21.
- [21] Hicken, J. E. and Zingg, D. W. (2008) Parallel newton-krylov solver for the euler equations discretized using simultaneous approximation terms. *AIAA journal*, **46**, 2773–2786.
- [22] li, S. and Xiao, F. (2007) Cip/multi-moment finite volume method for euler equations: A semi-lagrangian characteristic formulation. *Journal of Computational Physics*, **222**, 849–871.
- [23] — (2009) High order multi-moment constrained finite volume method. Part I: Basic formulation. *Journal of Computational Physics*, **228**, 3669–3707.
- [24] Katta, K. K., Nair, R. D. and Kumar, V. (2015) High-Order Finite-Volume Transport on the Cubed Sphere: Comparison between 1d and 2d Reconstruction Schemes. *Monthly Weather Review*, **143**, 2937–2954. URL: <http://journals.ametsoc.org/doi/10.1175/MWR-D-13-00176.1>.
- [25] Kurganov, A. and Lin, C.-T. (2007) On the reduction of numerical dissipation in central-upwind schemes. *Commun. Comput. Phys*, **2**, 141–163.
- [26] Kuzmin, D., Löhner, R. and Turek, S. (2012) *Flux-corrected transport: principles, algorithms, and applications*. Springer.
- [27] Lauritzen, P. H., Nair, R. D. and Ullrich, P. A. (2010) A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid. *Journal of Computational Physics*, **229**, 1401–1424. URL: <http://linkinghub.elsevier.com/retrieve/pii/S002199910900597X>.
- [28] — (2010) A conservative semi-lagrangian multi-tracer transport scheme (cslam) on the cubed-sphere grid. *Journal of Computational Physics*, **229**, 1401–1424.
- [29] Lauritzen, P. H., Skamarock, W. C., Prather, M. J. and Taylor, M. A. (2012) A standard test case suite for two-dimensional linear transport on the sphere. *Geoscientific Model Development*, **5**.
- [30] Lauritzen, P. H., Taylor, M. A., Overfelt, J., Ullrich, P. A., Nair, R. D., Goldhaber, S. and Kelly, R. (2017) CAM-SE-CSLAM: Consistent Coupling of a Conservative Semi-Lagrangian Finite-Volume Method with Spectral Element Dynamics. *Monthly Weather Review*, **145**, 833–855. URL: <http://journals.ametsoc.org/doi/10.1175/MWR-D-16-0258.1>.
- [31] Leonard, B., Lock, A. and MacVean, M. (1996) Conservative explicit unrestricted-time-step multidimensional constancy-preserving advection schemes. *Monthly Weather Review*, **124**, 2588–2606.
- [32] Li, J., Liao, W.-k., Choudhary, A., Ross, R., Thakur, R., Gropp, W., Latham, R., Siegel, A., Gallagher, B. and Zingale, M. (2003) Parallel netcdf: A high-performance scientific i/o interface. In *SC'03: Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, 39–39. IEEE.
- [33] Liu, X.-D., Osher, S. and Chan, T. (1994) Weighted essentially non-oscillatory schemes. *Journal of computational physics*, **115**, 200–212.

- [34] Löhner, R., Morgan, K., Peraire, J. and Vahdati, M. (1987) Finite element flux-corrected transport (fem-fct) for the euler and navier–stokes equations. *International Journal for Numerical Methods in Fluids*, **7**, 1093–1109.
- [35] Lukáčová-Medvid'ová, M., Morton, K. and Warnecke, G. (2002) Finite volume evolution galerkin methods for euler equations of gas dynamics. *International journal for numerical methods in fluids*, **40**, 425–434.
- [36] MacNamara, S. and Strang, G. (2016) Operator splitting. In *Splitting Methods in Communication, Imaging, Science, and Engineering*, 95–114. Springer.
- [37] Marras, S., Kelly, J. F., Moragues, M., Müller, A., Kopera, M. A., Vázquez, M., Giraldo, F. X., Houzeaux, G. and Jorba, O. (2016) A review of element-based Galerkin methods for numerical weather prediction: Finite elements, spectral elements, and discontinuous Galerkin. *Archives of Computational Methods in Engineering*, **23**, 673–722.
- [38] Michoski, C., Dawson, C., Kubatko, E. J., Wirasaet, D., Brus, S. and Westerink, J. J. (2016) A comparison of artificial viscosity, limiters, and filters, for high order discontinuous galerkin solutions in nonlinear settings. *Journal of Scientific Computing*, **66**, 406–434.
- [39] Nair, R. D., Thomas, S. J. and Loft, R. D. (2005) A discontinuous Galerkin transport scheme on the cubed sphere. *Monthly Weather Review*, **133**, 814–828.
- [40] Natale, A., Shipton, J. and Cotter, C. J. (2016) Compatible finite element spaces for geophysical fluid dynamics. *Dynamics and Statistics of the Climate System*, **1**.
- [41] Norman, M. R. (2013) Algorithmic improvements for schemes using the ADER time discretization. *Journal of Computational Physics*, **243**, 176–178.
- [42] — (2013) Targeting atmospheric simulation algorithms for large, distributed-memory, gpu-accelerated computers. In *GPU Solutions to Multi-scale Problems in Science and Engineering*, 271–282. Springer, Berlin, Heidelberg.
- [43] — (2014) A WENO-limited, ADER-DT, finite-volume scheme for efficient, robust, and communication-avoiding multi-dimensional transport. *Journal of Computational Physics*, **274**, 1–18.
- [44] — (2015) Arbitrarily high-order-accurate, hermite weno limited, boundary-averaged multi-moment constrained finite-volume (ba-mcv) schemes for 1-d transport. *Procedia Computer Science*, **51**, 2688–2697.
- [45] Norman, M. R. and Finkel, H. (2012) Multi-moment ADER-Taylor methods for systems of conservation laws with source terms in one dimension. *Journal of Computational Physics*, **231**, 6622–6642.
- [46] Norman, M. R., Mametjanov, A. and Taylor, M. (2017) 9 exascale programming approaches for accelerated climate modeling for energy. *Exascale Scientific Applications: Scalability and Performance Portability*, 187.
- [47] Norman, M. R. and Nair, R. D. (2008) Inherently conservative nonpolynomial-based remapping schemes: Application to semi-lagrangian transport. *Monthly Weather Review*, **136**, 5044–5061.
- [48] — (2018) A positive-definite, weno-limited, high-order finite volume solver for 2-d transport on the cubed sphere using an ader time discretization. *Journal of Advances in Modeling Earth Systems*, **10**, 1587–1612.
- [49] Norman, M. R., Nair, R. D. and Semazzi, F. H. (2011) A low communication and large time step explicit finite-volume solver for non-hydrostatic atmospheric dynamics. *Journal of Computational Physics*, **230**, 1567–1584.
- [50] — (2011) A low communication and large time step explicit finite-volume solver for non-hydrostatic atmospheric dynamics. *Journal of Computational Physics*, **230**, 1567–1584. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0021999110006376>.
- [51] Qiu, J. and Shu, C.-W. (2004) Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one-dimensional case. *Journal of Computational Physics*, **193**, 115–135.

- [52] Rider, W. J. (1994) A review of approximate riemann solvers with godunov's method in lagrangian coordinates. *Computers & fluids*, **23**, 397–413.
- [53] Rider, W. J. and Lowrie, R. B. (2002) The use of classical lax–friedrichs riemann solvers with discontinuous galerkin methods. *International journal for numerical methods in fluids*, **40**, 479–486.
- [54] Robert, A. (1982) A semi-lagrangian and semi-implicit numerical integration scheme for the primitive meteorological equations. *Journal of the Meteorological Society of Japan. Ser. II*, **60**, 319–325.
- [55] Roe, P. L. (1984) Generalized formulation of TVD Lax-Wendroff schemes.
- [56] Shu, C.-W. and Osher, S. (1988) Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of computational physics*, **77**, 439–471.
- [57] Skamarock, W. C. and Klemp, J. B. (2008) A time-split nonhydrostatic atmospheric model for weather research and forecasting applications. *Journal of computational physics*, **227**, 3465–3485.
- [58] Skamarock, W. C., Klemp, J. B., Duda, M. G., Fowler, L. D., Park, S.-H. and Ringler, T. D. (2012) A multiscale nonhydrostatic atmospheric model using centroidal voronoi tessellations and c-grid staggering. *Monthly Weather Review*, **140**, 3090–3105.
- [59] Staniforth, A. and Côté, J. (1991) Semi-lagrangian integration schemes for atmospheric models—a review. *Monthly weather review*, **119**, 2206–2223.
- [60] Staniforth, A. and Wood, N. (2008) Aspects of the dynamical core of a nonhydrostatic, deep-atmosphere, unified weather and climate-prediction model. *Journal of Computational Physics*, **227**, 3445–3464.
- [61] Titarev, V. A. and Toro, E. F. (2002) ADER: Arbitrary high order Godunov approach. *Journal of Scientific Computing*, **17**, 609–618.
- [62] Toro, E. F. (2013) *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media.
- [63] Toro, E. F., Spruce, M. and Speares, W. (1994) Restoration of the contact surface in the hll-riemann solver. *Shock waves*, **4**, 25–34.
- [64] Toro, E. F. and Titarev, V. A. (2005) ADER schemes for scalar non-linear hyperbolic conservation laws with source terms in three-space dimensions. *Journal of Computational Physics*, **202**, 196–215.
- [65] — (2006) Derivative Riemann solvers for systems of conservation laws and ADER methods. *Journal of Computational Physics*, **212**, 150–165.
- [66] Tsukanov, I. and Hill, M. (2000) Fast forward automatic differentiation library (FADLIB). *User manual*.
- [67] Ullrich, P. A. and Norman, M. R. (2014) The flux-form semi-lagrangian spectral element (ff-slse) method for tracer transport. *Quarterly Journal of the Royal Meteorological Society*, **140**, 1069–1085.
- [68] Weller, H., Lock, S.-J. and Wood, N. (2013) Runge–kutta imex schemes for the horizontally explicit/vertically implicit (hevi) solution of wave equations. *Journal of Computational Physics*, **252**, 365–381.
- [69] Weller, H., Thuburn, J. and Cotter, C. J. (2012) Computational modes and grid imprinting on five quasi-uniform spherical c grids. *Monthly Weather Review*, **140**, 2734–2755.
- [70] Xiao, F., Yabe, T., Peng, X. and Kobayashi, H. (2002) Conservative and oscillation-less atmospheric transport schemes based on rational functions. *Journal of Geophysical Research: Atmospheres*, **107**.

- [71] Yabe, T., Xiao, F. and Utsumi, T. (2001) The constrained interpolation profile method for multiphase analysis. *Journal of Computational physics*, **169**, 556–593.
- [72] Yu, M. L., Giraldo, F. X., Peng, M. and Wang, Z. J. (2015) Localized artificial viscosity stabilization of discontinuous Galerkin methods for nonhydrostatic mesoscale atmospheric modeling. *Monthly Weather Review*, **143**, 4823–4845.

GRAPHICAL ABSTRACT



A high-order WENO-limited Finite-Volume algorithm for atmospheric flow using ADER-Differential Transform time stepping is presented. The scheme is validated against literature standard test cases. The scheme is shown to be stable without WENO limiting or any explicit damping even at ninth-order accuracy due to upwind fluxes. The effects of WENO limiting are demon-

strated, and the increased resolution of higher-order accuracy is shown. The scheme is also shown to be very computationally efficient on accelerators.