Arbitrarily high-order (weighted) essentially non-oscillatory finite difference schemes for anelastic flows on staggered meshes

C. Parés-Pulido and S. Mishra and K. Pressel

Arbitrarily high-order (weighted) essentially non-oscillatory finite difference schemes for anelastic flows on staggered meshes

Carlos Parés-Pulido [*], Kyle G. Pressel [†] and Siddhartha Mishra [‡]

May 31, 2019

Abstract

We propose a WENO finite difference scheme to approximate anelastic flows, and scalars advected by them, on staggered grids. In contrast to existing WENO schemes on staggered grids, the proposed scheme is designed to be arbitrarily high-order accurate as it judiciously combines ENO interpolations of velocities with WENO reconstructions of spatial derivatives. A set of numerical experiments are presented to demonstrate the increase in accuracy and robustness with the proposed scheme, when compared to existing WENO schemes and state-of-the-art central finite difference schemes.

1 Introduction

In numerical modeling of fluid systems that are characterized by a low Mach number, it is often advantageous to introduce approximations to the fully compressible equations that eliminate acoustic waves. Doing so allows the explicit time integration of the governing equations to take much longer time steps in comparison to a similar integration of the compressible equations. This is because numerical stability in the soundproofed system does not depend on the phase velocity of acoustic waves. Various approximations to the compressible equations that eliminate acoustic waves have been developed including the incompressible, Boussinesq, and pseudo-incompressible approximations. The choice of which approximation is used is determined by the properties of the system being studied.

In atmospheric science, the anelastic system of equations is widely used as the basis for limited area models because many atmospheric flows are buoyancy driven and the stratification of the atmosphere makes it natural to assume that vertical gradients of density are much larger than horizontal gradients. In the anelastic system, the momentum, entropy, and continuity equations are given by

$$\frac{\partial U}{\partial t} + \frac{1}{\rho_0} \nabla \cdot (\rho_0 U \otimes U) = -\nabla \left(\frac{p'}{\rho_0} \right) + b e_3, \tag{1.1}$$

$$\frac{\partial s}{\partial t} + \frac{1}{\rho_0} \nabla \cdot (\rho_0 U s) = 0 \tag{1.2}$$

$$\nabla \cdot (\rho_0 U) = 0 \tag{1.3}$$

respectively. Here $U = (u_1, u_2, u_3) \in \mathbb{R}^3$ is the fluid velocity, ρ_0 is a horizontally homogeneous reference state density, $p' = p - p_0$ is the dynamic pressure, b is the buoyancy, $e_3 = (0, 0, 1)$, and s is the specific entropy defined as in [13]. Following [10], we define ρ_0 to be consistent with an isentropic and hydrostatic state and where the buoyancy $b = g (\rho/\rho_0 - 1)$ couples the momentum to thermodynamics, and ρ is determined from the equation of state.

---

[*]Seminar for Applied Mathematics, ETH Zürich, Rämistrasse 101, Zürich, Switzerland.
[†]Pacific Northwest National Laboratory, P.O. Box 999, Richland, WA 99352, USA
[‡]Seminar for Applied Mathematics, ETH Zürich, Rämistrasse 101, Zürich, Switzerland.

Given their importance in applications, a large variety of numerical methods for approximating anelastic (incompressible) flows are available. In applications where complex domain geometries are rarely encountered, finite difference schemes are a popular discretization framework as they can account for more general boundary conditions than spectral methods [3].

The most straightforward finite difference discretizations of (1.1) and (1.3) are on *collocated* meshes, where one evolves point values of the velocity and pressure at cell centers. However, it is well known that this procedure can lead to what is termed as *velocity-pressure decoupling*, [9] and references therein. Consequently, nonphysical checkerboard modes are obtained as solutions to the (discrete) elliptic equation that one has to solve in order to compute the pressure from the continuity equation (1.3).

A possible remedy for these nonphysical numerical artifacts is the use of *staggered meshes*. In this framework, each velocity component is discretized on the center of the underlying normal cell edge. The pressure (and any passively or actively advected scalars) is discretized at cell centers. The spatial derivatives in (1.1) can then be discretized by (high-order) central differences. A large number of such *central schemes* exist and are employed in many atmospheric codes.

In many applications, the Reynolds number of the underlying flow is very high and description of the dynamics requires computing contributions across a large range of scales. Given the infeasibility of direct numerical simulation (DNS) in such problems, one resorts to an additional turbulence model for (1.1). *Large eddy simulations* (LES) are a very attractive framework for turbulence modeling. In LES, a certain range of scales are numerically resolved and remaining *sub-grid scales* (SGS) are modeled by a suitable closure model; a detailed discussion can be found in [11, 15]. Many closure models are based on adding eddy diffusivity in terms of grid-scale gradients ([20, 7], further references in [12]).

As the leading order truncation terms for central finite difference schemes are dispersive in nature, the dominant source of numerical dissipation is delegated to the SGS terms. Hence, central finite difference schemes are widely used in the context of LES for anelastic flows. However, in many recent papers, it has been observed that central difference schemes together with SGS terms may not provide enough numerical diffusion to enable a stable computation. For instance, in recent papers [14], the authors demonstrated that the lack of numerical dissipation and the generation of dispersive oscillations by central finite difference schemes led to very inaccurate computation of stratocumulus clouds. In particular, there was not enough dissipation in the mixing zone, where sharp gradients (at grid scale) are encountered which lead to excessive SGS scalar fluxes. Given this well-known inadequacy of central finite difference schemes in resolving sharp grid scale gradients, these schemes can be very inaccurate in the LES of a significant class of turbulent flows. Moreover, a crucial constraint of any numerical simulation of anelastic flows is to retain positivity and bound preservation for advected scalars. It is well-known that central difference schemes can yield spurious oscillations that may lead to a breakdown of positivity of advected scalars.

An alternative discretization framework for such flows can be provided by (*Weighted-*) *essentially non-oscillatory* or WENO schemes. These schemes were first developed in [6, 17, 16] in the context of robust approximation of compressible flows which are characterized by discontinuities such as shock waves. In this discretization framework, either cell averages or point values are reconstructed from a piecewise polynomial interpolation, based on a clever stencil selection or stencil weighing procedure that tracks the direction of smoothness of the underlying functions. Consequently, a precise amount of numerical diffusion is added in the vicinity of sharp gradients (at grid scale) to stabilize computations. Moreover, the WENO procedure yields schemes with arbitrarily high order of accuracy.

WENO schemes for anelastic flows on collocated grids were developed in [2] and references therein. WENO schemes for anelastic flows on staggered grids have been described recently in [13, 14] and references therein. In these articles, the authors adapted the WENO procedure to approximate anelastic flows on staggered grids and used these schemes for a large eddy simulation of stratocumulus clouds. The results were very satisfactory and a significant gain in accuracy (with respect to observations) was obtained over central finite difference schemes. However a detailed analysis of the WENO schemes of [13, 14] reveals that although arbitrarily high-order WENO interpolations are employed, the full scheme is at most *second-order* accurate (see section 2 for details).

As very high order of accuracy is deemed essential for resolving flows with significant range of small scales

such as in LES [4], our aim in this article is to design WENO schemes with these properties. In particular, we design finite difference schemes that are

- Able to resolve sharp gradients without oscillations.

- (Formally) arbitrarily high-order accurate for the momentum equation (1.1) of anelastic flows.

- Formally arbitrarily high-order accurate for scalars, passively or actively advected by the flow.

We design these schemes by judiciously combining a WENO procedure with an additional essentially non-oscillatory (ENO) interpolation step to provide point values of fields. The arbitrarily high-order scheme for scalars is based on a non-conservative formulation.

The rest of the paper is organized as follows. In section 2, we describe the existing WENO schemes of [13, 14] on a staggered grid and prove that they are at most second-order accurate. The novel WENO scheme of this paper is presented in section 3 and numerical experiments are presented in section 4. In section 5, we present the arbitrarily high-order schemes for scalars, advected by the anelastic flow.

2   Existing WENO schemes on staggered grids.

For simplicity of notation and exposition, we consider here the two-dimensional version of the momentum equation (1.1). The corresponding velocity field is denoted as $U = (u_1, u_2)$; for convenience of notation, we will also use $U = (u, v)$.

2.1   The Arakawa C-grid

The computational domain is the rectangle $(x_{min}, x_{max}) \times (y_{min}, y_{max}) \subset \mathbb{R}^2$. This rectangle is partitioned into a grid of $N_x \times N_y$ cells $C_{ij} = [x_i, x_{i+1}) \times [y_j, y_{j+1})$, $\forall (i, j) \in \{0, 1, \ldots, N_x\} \times \{0, 1, \ldots, N_y\}$, with $x_i = x_{min} + i\Delta x$, $y_j = y_{min} + j\Delta y$ and $\Delta x = \frac{x_{max} - x_{min}}{N_x}$, $\Delta y = \frac{y_{max} - y_{min}}{N_y}$. The grid cell centers are $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$ with $x_{i+\frac{1}{2}} = x_{min} + (i + \frac{1}{2})\Delta x$, $y_{j+\frac{1}{2}} = y_{min} + (j + \frac{1}{2})\Delta y$.

On this grid, we consider what is called an Arakawa-C [1] staggering of variables on the computational grid, depicted in Figure 1. In this configuration, scalars (see section 5) lie at grid cell centers (denoted $\phi_{i+\frac{1}{2}, j+\frac{1}{2}} := \phi(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$); pressure is diagnostically computed at those points as well. Conversely, velocities lie at the center of the cell interfaces normal to their direction; specifically, u velocities are located at $(x_i, y_{j+\frac{1}{2}})$, and v at $(x_{i+\frac{1}{2}}, y_j)$. We denote these as $u_{i,j+\frac{1}{2}} := u(x_i, y_{j+\frac{1}{2}})$, $v_{i+\frac{1}{2}, j} := v(x_{i+\frac{1}{2}}, y_j)$.

Our aim is to design a WENO scheme on this grid to approximate the anelastic equations of motion. We start by providing a very brief description of the WENO framework for the sake of completeness.

2.2   An overview of the ENO and WENO methodology

Given a grid and the cell averages (resp. point values) of a function on the grid, ENO and WENO methods provide a reconstruction (resp. interpolation) of the function at a set of points on the grid in terms of piecewise polynomials. See [6, 16, 17] and references therein.

For simplicity, we consider a function $w : [x_L, x_R] \to \mathbb{R}$ on a 1D grid with cells $C_j := [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$ and aim to reconstruct piecewise polynomials from cell averages, with formal order of accuracy $k \in \mathbb{N}$. To this end, the key idea behind ENO is a careful choice of the stencil; i.e. the set of k cells $S^{(r)}(j) = \{C_{j-r}, \ldots, C_{j-r+k-1}\}$, $r \in \{0, \ldots, k-1\}$, from which information is taken. This choice is based on the smoothness of the function: out of the available stencils, the interpolation corresponds to one where the function is the *most regular*. This measurement of smoothness can be performed with Newton's divided differences. Denoting the chosen stencil by r, and denoting by $\bar{w}_j$ the cell average of w in cell C_j, the reconstruction has the form:

$$w_{j+\frac{1}{2}}^{(r)} := \sum_{i=0}^{k-1} c_{ri} \bar{w}_{i-r+j} = w(x_{j+\frac{1}{2}}^-) + O(\Delta x^k),$$
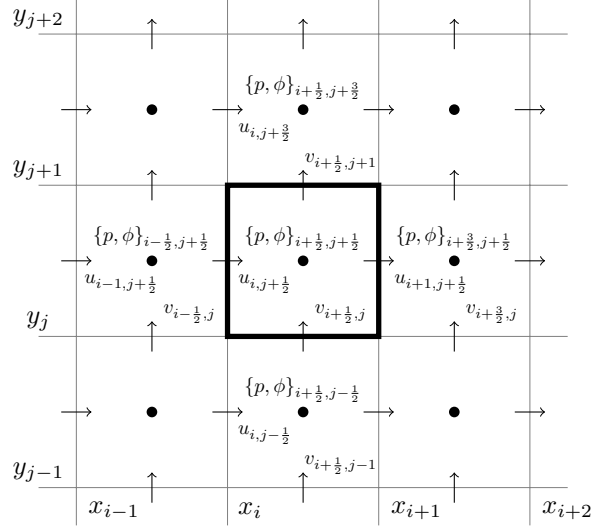
3

Figure 1: The staggered Arakawa-C grid in 2D. Velocities (u, v) are stored at cell interfaces, scalars ϕ at cell centers. Pressure p' is computed at cell centers as well.

where the constants c_{ri} are reconstruction weights, depending on stencil r, order k of the interpolation, and on the grid, but not on the underlying function w.

Even if information from $2k-1$ cells is used in the ENO method, the order of accuracy is limited to k. This redundancy is alleviated in the WENO method, in which the reconstruction is based on a convex combination of all candidate ENO reconstructions. The weights in this combination are very small for approximations coming from stencils with discontinuities, and we can still guarantee order of accuracy $2k-1$ when the function is sufficiently smooth in all stencils. To be more specific, the reconstruction at the interface of cell C_j has the form

$$w_{j+\frac{1}{2}} := \sum_{r=0}^{k-1} \omega_r w_{j+\frac{1}{2}}^{(r)} = w(x_{j+\frac{1}{2}}^-) + O(\Delta x^{2k-1}),$$

where $\omega_r \propto \frac{d_r}{(\epsilon + \beta_r)^p}$ is normalized so that $\sum_r \omega_r = 1$. Weights d_r are chosen to produce high-order reconstructions when w is smooth, and β_r is a smoothness indicator, such that $\beta_r \approx 0$ if v is smooth in $S^{(r)}(j)$, and $\beta_r = O(1)$ otherwise. Following [16], we set ϵ to be 10^{-10} for our experiments, and $p = 2$.

As we work with a finite difference framework, a priori it is unclear if the reconstruction procedure outlined above can be applied in the current context. However, we are only interested in a high-order approximation of the spatial derivatives in (1.1), and not in a very accurate approximation of point values. Therefore, we can use the novel trick of Shu [16] by which he converts reconstruction of cell averages into approximation of first derivatives. For the sake of completeness, we present the following proposition,

Proposition 1. Given the point values of a smooth function f on the cell centers of a uniform one-dimensional grid with step Δx, we can perform a WENO reconstruction to obtain *numerical fluxes* $\hat{f}$ such that the spatial derivative of f is approximated to a high-order of accuracy, i.e,

$$\frac{1}{\Delta x}(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}) = f'(x_i) + O(\Delta x^{k-1}) \tag{2.1}$$

Proof. We follow Shu [16] and seek a function h, depending on Δx, implicitly defined such that its sliding cell average is f. That is,

$$f(x) = \frac{1}{\Delta x} \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(s)ds \tag{2.2}$$

4

A straightforward application of the fundamental theorem of calculus yields,

$$f'(x) = \frac{1}{\Delta x}\left(h\left(x+\frac{\Delta x}{2}\right) - h\left(x-\frac{\Delta x}{2}\right)\right) \tag{2.3}$$

On the other hand, the cell averages of h verify that:

$$\bar{h}_i := \frac{1}{\Delta x}\int_{x_i-\frac{\Delta x}{2}}^{x_i+\frac{\Delta x}{2}} h(s)ds \stackrel{(2.2)}{=} f(x_i)$$

Now, we can apply standard WENO reconstruction, as outlined above, to the cell averages $\bar{h}_i$ to obtain

$$\hat{h}_{i+\frac{1}{2}} = h\left(x_i+\frac{\Delta x}{2}\right) + O(\Delta x^k)$$

And finally, inserting in (2.3),

$$f'(x_i) = \frac{1}{\Delta x}\left(\hat{h}_{i+\frac{1}{2}} - \hat{h}_{i-\frac{1}{2}}\right) + O(\Delta x^{k-1})$$

So it suffices to let $\hat{f}_{i+\frac{1}{2}} := \hat{h}_{i+\frac{1}{2}}$ and conclude the proof. $\qquad\square$

2.2.1 Further notation for WENO reconstructions.

Let $f: \mathbb{R} \to \mathbb{R}$ be a function, and let

$$\bar{f}_i := \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} f(x)dx,$$

$i \in I$ its cell averages. Fix $k \in \mathbb{N}$, we term $W_{-\frac{1}{2}}, W_{+\frac{1}{2}}: \mathbb{R}^{2k-1} \to \mathbb{R}$ as the WENO-$(2k-1)$ reconstruction operators that provide approximations to the point values at the left and right cell edge respectively, which if f is sufficiently regular are high-order accurate; i.e.:

$$W_{+\frac{1}{2}}(\bar{f}_{i-(k-1)}, \bar{f}_{i-(k-2)}, \ldots, \bar{f}_i, \ldots, \bar{f}_{i+(k-1)}) = f(x_{i+\frac{1}{2}}^-) + O(\Delta x^{2k-1})$$

$$W_{-\frac{1}{2}}(\bar{f}_{i-(k-1)}, \bar{f}_{i-(k-2)}, \ldots, \bar{f}_i, \ldots, \bar{f}_{i+(k-1)}) = f(x_{i-\frac{1}{2}}^+) + O(\Delta x^{2k-1})$$

2.3 The WENO scheme used by Pressel et al. [13, 14]

.

We start by describing our discretization of the advection terms in the momentum equation (1.1). To this end we assume uniform reference density and dynamic pressure ρ_0, $p' \equiv 1$ and buoyancy $b \equiv 0$ in (1.1) and write it down in two space dimensions as

$$\begin{aligned}
\frac{\partial u}{\partial t} &= -\frac{\partial}{\partial x}(u^2) - \frac{\partial}{\partial y}(uv),\\
\frac{\partial v}{\partial t} &= -\frac{\partial}{\partial x}(uv) - \frac{\partial}{\partial y}(v^2).
\end{aligned} \tag{2.4}$$

In particular, we will focus on a WENO discretization of the first equation in (2.4). Following [13], a semi-discrete WENO scheme for it can be written as

$$\frac{\partial}{\partial t}u(x_i, y_{j+\frac{1}{2}}, t) = -\frac{1}{\Delta x}\left(F_{i+\frac{1}{2},j+\frac{1}{2}}^{u,x}(t) - F_{i-\frac{1}{2},j+\frac{1}{2}}^{u,x}(t)\right) - \frac{1}{\Delta y}\left(F_{i,j+1}^{u,y}(t) - F_{i,j}^{u,y}(t)\right). \tag{2.5}$$
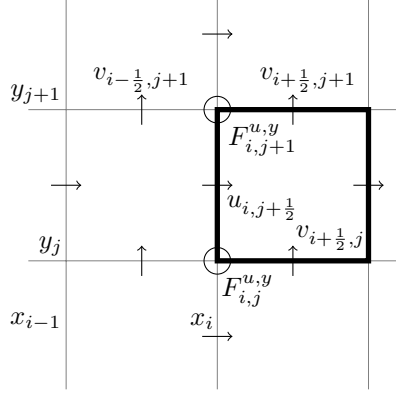
Figure 2: Reconstruction of $\frac{\partial}{\partial y}(uv)$ on a staggered grid. Circles denote points where fluxes need to be computed.

Flux reconstructions F (for the sake of clarity, we omit the time dependence throughout) are given by a generic reconstruction operator $\widehat{(\cdot)}$, to be described in the sequel:

$$F^{u,x}_{i+\frac{1}{2},j+\frac{1}{2}} = \widehat{u^2}_{i+\frac{1}{2},j+\frac{1}{2}}, \quad F^{u,y}_{i,j} = \widehat{uv}_{i,j}.$$

The reconstruction $F^{u,x}_{i+\frac{1}{2},j+\frac{1}{2}}$ corresponds to the function u^2 along the x-direction. It can be computed directly by performing a WENO reconstruction from the point values $u^2_{i,j+\frac{1}{2}}$ in the x-direction.

On the other hand, the reconstruction $F^{u,y}_{i,j}$ corresponds to approximating a y-spatial derivative of the function uv. This is problematic as the values of the velocities u and v are *staggered* and are not defined at the same locations. In particular, u is defined at points $(x_i, y_{j+\frac{1}{2}})$ whereas v is defined at points $(x_{i+\frac{1}{2}}, y_j)$. This situation is depicted in Figure 2. Thus, we have to perform an interpolation procedure to be able to obtain point values for the function uv, which can then provide the input to a WENO reconstruction procedure for calculating $F^{u,y}_{i,j}$ in (2.5).

One possible approach to reconstructing the uv momentum flux, that has been used in applications (e.g. Pressel et al. [13][*], Skamarock et al. [19]), is to use a high-order one-dimensional central interpolation to reconstruct the advecting velocity v, and WENO interpolation to reconstruct the velocity u. This procedure can be summarized as:

Algorithm 1.
Goal: Given $k \in \mathbb{N}$, find a reconstruction of the numerical flux, $F^{u,y}_{i,j+1}$, such that, formally, $\forall i, j$:

$$\frac{1}{\Delta y}(F^{u,y}_{i,j+1} - F^{u,y}_{i,j}) = \frac{\partial}{\partial y}(uv)(x_i, y_{j+\frac{1}{2}}) + O(\Delta x^{2k-1} + \Delta y^{2k-1}) \tag{2.6}$$

1. Using 1D central interpolation on the horizontal, symmetric, $2k$-point stencil $\{v_{i-\frac{1}{2}+l,\,j+1}\}^k_{l=-k+1}$, compute

$$\tilde{v}_{i,j+1} = v(x_i, y_{j+1}) + O(\Delta x^{2k})$$

2. Compute the two possible biased, vertical $(2k-1)$-WENO reconstructions of u at (x_i, y_{j+1}):

$$\hat{u}^-_{i,j+1} := W_{+\frac{1}{2}}\left(u_{i,j+\frac{1}{2}+(-k+1)}, u_{i,j+\frac{1}{2}+(-k+2)}, \ldots, u_{i,j+\frac{1}{2}+k-1}\right)$$

$$\hat{u}^+_{i,j+1} := W_{-\frac{1}{2}}\left(u_{i,j+\frac{1}{2}+(-k+2)}, u_{i,j+\frac{1}{2}+(-k+3)}, \ldots, u_{i,j+\frac{1}{2}+k}\right)$$

---

[*]In [13], this idea of interpolating the advecting velocity is also used for the reconstruction of the $(u_i)^2$ fluxes

3. Upwind with the sign of $\tilde{v}_{i,j+1}$:

$$\hat{u}_{i,j+1} = \begin{cases} \hat{u}^+_{i,j+1} & \text{if } \tilde{v}_{i,j+1} < 0 \\ \hat{u}^-_{i,j+1} & \text{if } \tilde{v}_{i,j+1} \geq 0 \end{cases}$$

4. Set $F^{u,y}_{i,j+1} := \tilde{v}_{i,j+1}\hat{u}_{i,j+1}$

This completes the description of the scheme (2.5). We observe that one can discretize the second equation in (2.4) in an analogous fashion. In particular, the WENO reconstruction of $\partial_y(v^2)$ is straightforward and Algorithm 1 can be readily modified to yield a WENO reconstruction of $\partial_x(uv)$.

2.3.1   Order of Accuracy

In this section, we will investigate the (formal) order of accuracy of the WENO scheme described in [13]. This order of accuracy is measured in terms of the truncation error of the scheme. We have the following theorem,

Theorem 1. *For u, v sufficiently smooth and for any $2 \leq k \in \mathbb{N}$, Algorithm 1 produces an at most second order approximation to $\frac{\partial}{\partial y}(uv)(x_i, y_{j+\frac{1}{2}})$ in (2.4). Hence, the WENO schemes used by[13] are at most second order accurate, and equation (2.6) does not hold for them.*

Proof. We assume $\Delta x \propto \Delta y$, and so $O(\Delta x) = O(\Delta y)$. Following Algorithm 1, the approximation to $\partial_y(uv)(x_i, y_{j+\frac{1}{2}})$ is:

$$D^{u,y}_{i,j+\frac{1}{2}} := \frac{1}{\Delta y}\left(\tilde{v}_{i,j+1}\hat{u}_{i,j+1} - \tilde{v}_{i,j}\hat{u}_{i,j}\right)$$

where $\hat{u}$ are the reconstructions of u such that, by (2.1),

$$\frac{1}{\Delta y}(\hat{u}_{i,j+1} - \hat{u}_{i,j}) = \partial_y u(x_i, y_{j+\frac{1}{2}}) + O(\Delta y^{2k-1}),$$

and $\tilde{v}_{i,m}$ is the horizontally interpolated value for v at (x_i, y_m), $m \in \{j, j+1\}$, as per step 1 in Algorithm 1:

$$\tilde{v}_{i,m} = v(x_i, y_m) + O(\Delta x^{2k})$$

We will show that $D^{u,y}_{i,j+\frac{1}{2}} = \partial_y(uv) + O(\Delta y^2)$.

Denote by $v_{i,j+\frac{1}{2}} := v(x_i, y_{j+\frac{1}{2}})$ the exact point value of v. Adding and subtracting $v_{i,j+\frac{1}{2}}(\hat{u}_{i,j+1} + \hat{u}_{i,j})/\Delta y$ and rearranging:

$$D^{u,y}_{i,j+\frac{1}{2}} = \frac{\tilde{v}_{i,j+1} - v_{i,j+\frac{1}{2}}}{\Delta y}\hat{u}_{i,j+1} + \frac{v_{i,j+\frac{1}{2}} - \tilde{v}_{i,j}}{\Delta y}\hat{u}_{i,j} + v_{i,j+\frac{1}{2}}\frac{\hat{u}_{i,j+1} - \hat{u}_{i,j}}{\Delta y} \tag{2.7}$$

We observe that, by (2.1), the last term in the right hand side of (2.7) is a high-order $O(\Delta x^{2k-1})$ approximation to $(v\,\partial_y u)(x_i, y_{j+\frac{1}{2}})$.

Next, we will show that the other terms in (2.7), which approximate $u\,\partial_y v$, produce a *lower-order* ap-

proximation. Using Taylor expansions and (2.1),

$$\frac{\tilde{v}_{i,j+1} - v_{i,j+\frac{1}{2}}}{\Delta y}\hat{u}_{i,j+1} + \frac{v_{i,j+\frac{1}{2}} - \tilde{v}_{i,j}}{\Delta y}\hat{u}_{i,j} =$$

$$= \frac{v_{i,j+1} - v_{i,j+\frac{1}{2}}}{\Delta y}\hat{u}_{i,j+1} + \frac{v_{i,j+\frac{1}{2}} - v_{i,j}}{\Delta y}\hat{u}_{i,j} + O(\Delta y^{2k-1})$$

$$= \left(\frac{\partial_y v(x_i, y_{j+\frac{1}{2}})}{2} + \partial_{yy} v(x_i, y_{j+\frac{1}{2}})\frac{\Delta y}{8} + O(\Delta y^2)\right)\hat{u}_{i,j+1}$$

$$+ \left(\frac{\partial_y v(x_i, y_{j+\frac{1}{2}})}{2} - \partial_{yy} v(x_i, y_{j+\frac{1}{2}})\frac{\Delta y}{8} + O(\Delta y^2)\right)\hat{u}_{i,j} + O(\Delta y^{2k-1})$$

$$= \partial_y v(x_i, y_{j+\frac{1}{2}})\frac{\hat{u}_{i,j+1} + \hat{u}_{i,j}}{2} + \frac{\partial_{yy} v(x_i, y_{j+\frac{1}{2}})}{8}\frac{\hat{u}_{i,j+1} - \hat{u}_{i,j}}{\Delta y}\Delta y^2 + O(\Delta y^2)$$

$$= \partial_y v(x_i, y_{j+\frac{1}{2}})\frac{\hat{u}_{i,j+1} + \hat{u}_{i,j}}{2} + \frac{(\partial_{yy} v\, \partial_y u)(x_i, y_{j+\frac{1}{2}})}{8}\Delta y^2 + O(\Delta y^2)$$

$$= \partial_y v(x_i, y_{j+\frac{1}{2}})\frac{\hat{u}_{i,j+1} + \hat{u}_{i,j}}{2} + O(\Delta y^2)$$

For the first term, considering the 1D function $f_i(\cdot) = u(x_i, \cdot)$, and naming $h_i(\cdot)$ its 1D WENO reconstruction function as in Proposition 1, we have that:

$$\frac{\hat{u}_{i,j+1} + \hat{u}_{i,j}}{2} = \frac{1}{\Delta y}\Delta y\frac{h_i(y_{j+1}) + h_i(y_j)}{2} + O(\Delta y^{2k-1}) \tag{2.8}$$

$$= \frac{1}{\Delta y}\left(\int_{y_j}^{y_{j+1}} h_i(t)\,dt + O(\Delta y^3)\right) + O(\Delta y^{2k-1}) \tag{2.9}$$

$$= u(x_i, y_{j+\frac{1}{2}}) + O(\Delta y^2) \tag{2.10}$$

Where in the second step we have used the accuracy of the trapezoidal quadrature rule. Finally,

$$\frac{\hat{u}_{i,j+1} + \hat{u}_{i,j}}{2}\partial_y v(x_i, y_{j+\frac{1}{2}}) = (u\,\partial_y v)(x_i, y_{j+\frac{1}{2}}) + O(\Delta y^2)$$

Summarizing,

$$\frac{1}{\Delta y}\left(\tilde{v}_{i,j+1}\hat{u}_{i,j+1} - \tilde{v}_{i,j}\hat{u}_{i,j}\right) = \left((v\,\partial_y u)(x_i, y_{j+\frac{1}{2}}) + O(\Delta y^{2k-1})\right) + \left((u\,\partial_y v)(x_i, y_{j+\frac{1}{2}}) + O(\Delta y^2)\right)$$

$$= \partial_y(uv) + O(\Delta y^2)$$

$\square$

An analogous treatment of the WENO discretization of the $\partial_x(uv)$ term in the second equation of (2.4) also yields an at most second order accuracy of this term.

Remark 1. The surprising aspect of Theorem 1 is the fact that the formal order of accuracy of the WENO scheme is at most 2, even if the underlying interpolating polynomials are of arbitrarily high order. As the above proof shows, this can be attributed to the errors that accumulate while interpolating values on the staggered grid. A similar and unexpected loss of convergence is also found for some popular central schemes for anelastic flows, for instance the Wicker-Skamarock scheme presented in [19]), see Appendix A for details.

3 An arbitrarily high-order WENO scheme on staggered grids

Given the fact that existing WENO schemes for discretizing the anelastic equations (1.1) are at most second order accurate, we will present an arbitrarily high-order WENO scheme on such grids. Again for the simplicity of notation and exposition, we will focus on the two-dimensional problem while remarking that the extension to three space dimensions is straightforward.

3.1 Interpolation

As identified in the previous section, the reason for the loss of accuracy of existing WENO schemes is the error accumulated while interpolating across the staggered grid. In particular, one-dimensional (central) interpolations cause a loss of order of accuracy. A genuinely multi-dimensional interpolation can potentially restore design order of accuracy. Similar ideas were explored in Ghosh and Baeder [5] and Zhang and Jackson [23]. In both these papers, the authors consider a non-conservative formulation of the momentum equations that necessitates a genuinely multi-dimensional (central) interpolation. Here, we will retain the conservative version of the momentum equations and design a two-dimensional ENO interpolation, in order to be consistent with the non-oscillatory character of the overall scheme. This increases the computational cost of the scheme, albeit the overall cost is still in the same order as the WENO reconstructions.

We illustrate the algorithm by describing the discretization of the *problematic* $\partial_y(uv)$ term in the first equation of (2.4). Recalling the issue from section 2 and as depicted in Figure 3, we have to obtain values of uv at a set of points in order to provide a WENO reconstruction of the derivative $\partial_y(uv)$. In particular, we need to find approximations to the values of v at the points $(x_i, y_{j+\frac{1}{2}})$ where values of u are defined. There are two possible choices, as depicted in Figure 3. Either we perform a one-dimensional ENO interpolation of v in the y-direction to define the point values $v_{i+\frac{1}{2},j+\frac{1}{2}}$ (at the cell centers) and then perform a one-dimensional ENO interpolation of $v_{i+\frac{1}{2},j+\frac{1}{2}}$ along the x-direction to obtain the values of v at the points $(x_i, y_{j+\frac{1}{2}})$ (this procedure is shown by solid curves in Figure 3) or we perform a one-dimensional ENO interpolation of v in the x-direction to define the point values $v_{i,j}$ and then perform a one-dimensional ENO interpolation of $v_{i,j}$ along the y-direction to obtain the values of v at the points $(x_i, y_{j+\frac{1}{2}})$ (this procedure is shown by dashed curves in Figure 3). In the sequel, we will require the values of the velocity field at cell centers. Therefore, we do the former and summarize the interpolation algorithm below,

Algorithm 2.

Goal: Given point values of $v_{i+\frac{1}{2},j} \approx v(x_{i+\frac{1}{2}}, y_j)$, $\forall i, j$, find non-oscillatory, order $2k-1$ interpolations of v, at the locations $(x_i, y_{j+\frac{1}{2}})$.

1. For each cell, use 1D vertical ENO interpolation to find an approximation to v at the cell center, $\tilde{v}_{i+\frac{1}{2},j+\frac{1}{2}}$, based on the stencil of maximal smoothness of $2k-1$ cells contained in $\{v_{i+\frac{1}{2},j+r}\}_{r=-(2k-3)}^{2k-2}$

$$\tilde{v}_{i+\frac{1}{2},j+\frac{1}{2}} = v(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) + O(\Delta y^{2k-1})$$

2. For each cell, use 1D horizontal ENO interpolation to find an approximation to v at the location of u, denote it $\tilde{v}_{i,j+\frac{1}{2}}$, based on the stencil of maximal smoothness of $2k-1$ cells contained in $\{\tilde{v}_{i+\frac{1}{2}+r,j+\frac{1}{2}}\}_{r=-(2k-2)}^{2k-3}$

$$\tilde{v}_{i,j+\frac{1}{2}} = v(x_i, y_{j+\frac{1}{2}}) + O(\Delta x^{2k-1} + \Delta y^{2k-1})$$

Remark 2. The above algorithm can be readily extended to discretize the $\partial_x(uv)$ term in the second equation of (2.4). This procedure applies almost verbatim to the 3D scenario, the only difference being that a higher number of interpolations, along each of three directions, are required.

Remark 3. In Algorithm 2, we choose to apply 1D ENO interpolation, rather than WENO; this choice is made for stability. Near sharp gradients, the scheme displays some sensitivity to the reconstruction stencil. Experimentally, we have found that the best results are produced with ENO, with a stencil that contains at least one node to each side of the point where interpolation is required. The stencils described above already enforce this requirement. E.g. for step 1, all stencils of $2k-1$ cells contain $v_{i+\frac{1}{2},j}$ and $v_{i+\frac{1}{2},j+1}$
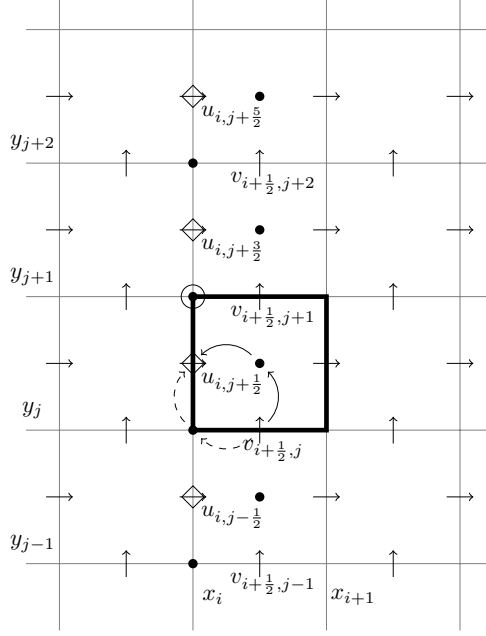
9

Figure 3: 2D interpolation for $\frac{\partial}{\partial y}(uv)$ on a staggered mesh. The two candidate 2-step 1D interpolations are depicted with solid and dashed curved arrows.
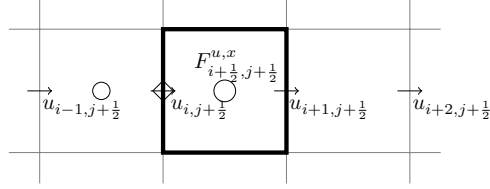

Figure 4: Reconstruction of $\frac{\partial}{\partial x}u^2$ on a staggered grid. The diamond marks the location of u in the highlighted cell; numerical fluxes must be calculated at the circles.

3.2 Numerical fluxes

In this section, we will describe the numerical fluxes for discretizing the advective terms in (1.1). As the preceding discussion indicates, we need to differentiate between two sets of terms, namely $\frac{\partial}{\partial x_i}(\rho_0 u_i^2)$, and $\frac{\partial}{\partial x_j}(\rho_0 u_i u_j)$ for $i \neq j$.

3.2.1 Terms of the form $\frac{\partial}{\partial x_i}(\rho_0 u_i^2)$

For illustration, we describe the flux for $\frac{\partial}{\partial x}(\rho_0 u^2)$ in two space dimensions. This configuration is displayed in Figure 4, where the diamond marks the position of u in the highlighted cell, and the circles are at the points where fluxes need to be reconstructed.

Algorithm 3.
 Goal: Find a numerical flux $F^{u,x}$ such that, formally, $\forall i, j$:

$$\frac{F^{u,x}_{i+\frac{1}{2},j+\frac{1}{2}} - F^{u,x}_{i-\frac{1}{2},j+\frac{1}{2}}}{\Delta x} = \frac{\partial(\rho_0 u^2)}{\partial x}(x_i, y_{j+\frac{1}{2}}) + O(\Delta x^{2k-1} + \Delta y^{2k-1})$$

1. Compute the two biased WENO reconstructions of the flux function $f(x) = \left(\rho_0 u^2\right)(x, y_{j+\frac{1}{2}})$ at $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$. Denoting $f_l := \rho_0(x_l, y_{j+\frac{1}{2}})\, u^2_{l,j+\frac{1}{2}}$, this is:

$$\hat{f}^-_{i+\frac{1}{2},j+\frac{1}{2}} = W_{+\frac{1}{2}}\left(f_{i-(k-1)}, f_{i-(k-2)}, \ldots, f_{i+(k-1)}\right)$$

$$\hat{f}^+_{i+\frac{1}{2},j+\frac{1}{2}} = W_{-\frac{1}{2}}\left(f_{i-(k-2)}, f_{i-(k-1)}, \ldots, f_{i+k}\right)$$

2. Choose a consistent numerical flux function $G(u^-, u^+)$, and set

$$F^{u,x}_{i+\frac{1}{2},j+\frac{1}{2}} := G(\hat{f}^-_{i+\frac{1}{2},j+\frac{1}{2}}, \hat{f}^+_{i+\frac{1}{2},j+\frac{1}{2}})$$

Remark 4. In the numerical experiments in the sequel, G is taken to be the upwind flux:

$$G(u_l, u_r) = \begin{cases} u_l & \text{if } a_{i+\frac{1}{2},j+\frac{1}{2}} \geq 0 \\ u_r & \text{if } a_{i+\frac{1}{2},j+\frac{1}{2}} < 0 \end{cases}; \qquad a_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{u_{i,j+\frac{1}{2}} + u_{i+1,j+\frac{1}{2}}}{2}.$$

Other fluxes, e.g. Lax-Friedrichs, Rusanov, etc. can also be considered.

3.2.2 Terms of the form $\frac{\partial}{\partial x_j}(\rho_0 u_i u_j)$, with $i \neq j$

Again for the purpose of illustration, we focus on the two-dimensional discretization of the term $\frac{\partial}{\partial y}(\rho_0 uv)$. As discussed before, this computation requires interpolations of v at the points where u lies; i.e. this algorithm assumes the availability in each cell of interpolations

$$\tilde{v}_{i,j+\frac{1}{2}+m} := v(x_i, y_{j+\frac{1}{2}+m}) + O(\Delta x^{2k-1} + \Delta y^{2k-1}), \quad \text{for } m \in \{-(k-1), \ldots, k\},$$

which can be computed with Algorithm 2. This is depicted in Figure 5, where the diamond marks the position of v in the highlighted cell; and the circles are at the locations where fluxes need to be computed.

With said interpolations, we now have high-order approximations to the values of (uv) at the desired points (marked with black dots in Fig. 5). Thus, for $F^{u,y}_{i,j+1}$, the procedure would be:

Algorithm 4.
Goal: Find a numerical flux $F^{u,y}_{i,j+1}$ such that, formally, $\forall i, j$:

$$\frac{F^{u,y}_{i,j+1} - F^{u,y}_{i,j}}{\Delta y} = \frac{\partial(\rho_0 uv)}{\partial y}(x_i, y_{j+\frac{1}{2}}) + O(\Delta x^{2k-1} + \Delta y^{2k-1})$$

1. Compute the two biased WENO reconstructions of the flux function $g(y) = (\rho_0 uv)(x_i, y)$ at (x_i, y_{j+1}). Denoting $g_l := \rho_0(x_i, y_{l+\frac{1}{2}}) u_{i,l+\frac{1}{2}} \tilde{v}_{i,l+\frac{1}{2}}$, with $\tilde{v}_{i,l+\frac{1}{2}}$ the high-order interpolations of Algorithm 2, results in

$$\hat{g}^-_{i,j+1} = W_{+\frac{1}{2}}\left(g_{j-(k-1)}, g_{j-(k-2)}, \ldots, g_{j+(k-1)}\right)$$

$$\hat{g}^+_{i,j+1} = W_{-\frac{1}{2}}\left(g_{j-(k-2)}, g_{j-(k-1)}, \ldots, g_{j+k}\right)$$

2. Choose a numerical flux function $G(u^-, u^+)$, and set

$$F^{u,y}_{i,j+1} := G(\hat{g}^-_{i,j+1}, \hat{g}^+_{i,j+1}).$$

Again, one can use the upwind flux function and upwind with the value of $\tilde{v}_{i,j+\frac{1}{2}}$ or use a Lax-Friedrichs or Rusanov flux.
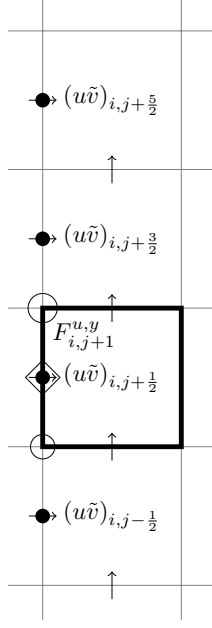
11

Figure 5: Reconstruction of $\frac{\partial}{\partial y}(uv)$ on a staggered grid. The diamond marks the location of u in the highlighted cell; numerical fluxes must be calculated at the circles. Interpolations of v are required at the points marked with dots.

3.3 The advection scheme

We collect all the above ingredients and write the advection scheme as:

Algorithm 5.
Goal: Given $i \in \{1, \ldots, d\}$, obtain a high-order, essentially non-oscillatory approximation, on an Arakawa-C grid (Fig. 1), to the spatial derivatives in:

$$\frac{\partial}{\partial t} u_i = -\frac{1}{\rho_0} \sum_{j=1}^{d} \frac{\partial}{\partial x_j} (\rho_0 u_i u_j)$$

1. Using Algorithm 2, $\forall j \in \{1, \ldots, d\}$, $j \neq i$, obtain high-order approximations to u_j at the location of u_i.

2. For $j \in \{1, \ldots, d\}$:

    (a) If $j = i$, find a numerical flux F^{u_i, x_i} for $\frac{\partial}{\partial x_i}(\rho_0 u_i^2)$ using Algorithm 3.

    (b) Otherwise, find a numerical flux F^{u_i, x_j} for $\frac{\partial}{\partial x_j}(\rho_0 u_i u_j)$ using Algorithm 4.

3. Find the spatial differences of the computed numerical fluxes F^{u_i, x_k} in the direction of x_k, $k \in \{1, \ldots, d\}$, and divide by ρ_0 to complete the approximation. As an example, the semi-discrete scheme for discretizing (2.4) is

$$
\begin{aligned}
\frac{\partial}{\partial t} u_{i,j+\frac{1}{2}} &= -\frac{1}{\rho_0(x_i, y_{j+\frac{1}{2}})} \left(\frac{F^{u,x}_{i+\frac{1}{2},j+\frac{1}{2}} - F^{u,x}_{i-\frac{1}{2},j+\frac{1}{2}}}{\Delta x} + \frac{F^{u,y}_{i,j+1} - F^{u,y}_{i,j}}{\Delta y} \right) \\
\frac{\partial}{\partial t} v_{i+\frac{1}{2},j} &= -\frac{1}{\rho_0(x_{i+\frac{1}{2}}, y_j)} \left(\frac{F^{u,y}_{i+1,j} - F^{u,y}_{i,j}}{\Delta x} + \frac{F^{v,y}_{i+\frac{1}{2},j+\frac{1}{2}} - F^{u,y}_{i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y} \right)
\end{aligned}
\tag{3.1}
$$

12

Lemma 1. *For sufficiently smooth U, ρ_0, Algorithm 5 produces arbitrarily high-order accurate approxima-
tions to the spatial derivatives of the fluxes in (1.1), i.e, for any $k \in \mathbb{N}$, the truncation error in discretizing
the advective parts of (1.1) is $O(\Delta x^{2k-1} + \Delta y^{2k-1})$.*

The proof follows in a straightforward manner from the construction and for the accuracy of the WENO
method, see Proposition 1.

3.4 The complete scheme

In this section, we will provide an overview of the discretizations of other terms in the anelastic equations.

3.4.1 Source terms

Source terms in the equations (e.g. buoyancy b in the equation of momentum (1.1), or any other physical
source terms) are easy to deal with in this context. Since buoyancy has a closed form in the anelastic
formulation (see section 5.2.1), this reduces to computing the source terms at the appropriate locations and
adding them to the tendencies, i.e, discretizations of the advective terms, computed in Algorithm 5.

3.4.2 Pressure

For the resolution of the pressure gradient in (1.1), and the continuity equation (1.3), the well-known idea
of Leray projection into the space of divergence-free fields can be used. The divergence operator is applied
to (1.1) yielding,

$$\frac{\partial^2}{\partial t \partial x_i} \rho_0 u_i = -\frac{\partial^2}{\partial x_i x_j}(\rho_0 u_i u_j) + \frac{\partial}{\partial x_3}(\rho_0 b e_3) - \frac{\partial}{\partial x_i}\rho_0 \frac{\partial}{\partial x_i}\frac{p'}{\rho_0}$$

where we use Einstein's notation of implied summation over repeated indices. Discretizing the time derivative
with forward Euler method with time step Δt, we can rewrite it as:

$$\left(\frac{\partial}{\partial x_i}\rho_0 u_i\right)^{n+1} = \left(\frac{\partial}{\partial x_i}\rho_0 u_i\right)^n + \Delta t \left(\frac{\partial^2}{\partial x_i x_j}(\rho_0 u_i u_j) + \frac{\partial}{\partial x_3}(\rho_0 b e_3) - \frac{\partial}{\partial x_i}\rho_0 \frac{\partial}{\partial x_i}\frac{p'}{\rho_0}\right)$$

Setting the left hand side to zero in order to enforce null divergence, and rearranging, results in

$$\frac{\partial}{\partial x_i}\rho_0 \frac{\partial}{\partial x_i}\frac{p'\Delta t}{\rho_0} = \frac{\partial}{\partial x_i}\rho_0 \left(u_i^n + \frac{1}{\rho_0}\frac{\partial}{\partial x_j}(\rho_0 u_i u_j) + b e_3\right)$$

We observe that the above is a Poisson-type equation for the function $\frac{p'\Delta t}{\rho_0}$, and the right hand side is
the result of evolving $(u_i)^n$ by forward Euler with the spatial tendencies computed so far.

In the directions in which the domain is periodic (e.g. for horizontal surfaces), this Poisson equation can
be efficiently solved with a spectral method. For the vertical direction, however, periodicity does not, in
general, hold (as $\rho_0(z_{bottom}) \neq \rho_0(z_{top})$). In this case, we need to apply a finite difference method such as
the one used in Pressel et al. [13].

Once pressure p' is computed, its gradient needs to be removed from the computed tendencies as per eq.
(1.1).

3.4.3 Time evolution

So far, we have a semi-discrete scheme of the form:

$$\frac{d}{dt}U_{i,j}(t) = L(t, U_{i,j}(t))$$

where L is the operator that collects all spatial discretizations. A time marching algorithm is necessary to
complete our discretization. Given our focus on obtaining a non-oscillatory discretization, we choose the

popular strong stability preserving, third order Runge-Kutta method (SSP-RK3), described e.g. by Shu in [17] for discretizing ODE $\varphi'(t) = L(t, \varphi(t))$ with

$$\varphi^{(1)} = \varphi_n + \Delta t (L(t_n, \varphi_n))$$
$$\varphi^{(2)} = \frac{3}{4}\varphi_n + \frac{1}{4}\left(\varphi^{(1)} + \Delta t \left(L(t_n + \Delta t, \varphi^{(1)})\right)\right)$$
$$\varphi_{n+1} = \frac{1}{3}\varphi_n + \frac{2}{3}\left(\varphi^{(2)} + \Delta t \left(L(t_n + \frac{\Delta t}{2}, \varphi^{(2)})\right)\right)$$

3.4.4 Layout of the full scheme

The complete scheme is realized in the following algorithmic form,

Algorithm 6.
 Goal: Find numerical approximations to the anelastic Euler equations (1.1)-(1.3) on an Arakawa-C grid.
   For each time step, $t = t^n$:

1. Compute Δt^n such that a CFL condition is verified

2. For each Runge-Kutta substep:

    (a) Compute the advective tendencies for momentum (Algorithm 5)

    (b) Add source terms

    (c) Perform pressure correction

3. Compute total SSP-RK3 tendencies and update velocities.

4. $t^{n+1} = t^n + \Delta t^n$

4   Numerical experiments

In this section, we will present a suite of numerical experiments in order to compare WENO schemes, such as the one described in [13] and the one proposed here, to each other and to well-known central schemes such as the Wicker-Skamarock scheme [19] and the Morinishi scheme [8]. See Appendix A for a detailed description of these schemes.

4.1   Discontinuous vortex patch

The first numerical experiment is designed to test the ability of a finite difference scheme to approximate sharp gradients. We consider only the advective part of the momentum equations (2.4) on the two-dimensional domain $D = [0, 2\pi]^2$ with periodic boundary conditions. The initial datum is

$$u_0(x, y) = \begin{cases} -\frac{1}{2}(y - \pi) & \text{if } (x, y) \in \Delta \\ 0 & \text{otherwise} \end{cases}, \qquad v_0(x, y) = \begin{cases} \frac{1}{2}(x - \pi) & \text{if } (x, y) \in \Delta \\ 0 & \text{otherwise} \end{cases}$$

Here, $\Delta = \{(x, y) \in \mathbb{R}^2, (x - \pi)^2 + (y - \pi)^2 < \frac{\pi}{2}\}$. Thus, initial velocity is discontinuous and the initial vorticity is confined to the region Δ. All schemes are tested on a uniform Cartesian grid of 128^2 points. The time step is chosen to be consistent with a CFL number of 0.1.

We approximate (2.4) with the sixth-order scheme of Morinishi et al [8], see also Appendix A, on this grid and display the results at time $T = 1$ in Figure 6a. We observe from Figure 6a that the computed velocity with this high-order central scheme is already very oscillatory around the discontinuity at this time. These spurious oscillations build up in time and the scheme blows up by time $T = 2$. Similar behavior

was also seen in the case of the Wicker-Skamarock scheme, [19] and Appendix A, and for different orders of the central schemes. The numerical results are expected as the central scheme will become unstable around discontinuities.

We compute solutions for this test case with the fifth order versions of the WENO scheme of [13] and the WENO scheme proposed in this paper, i.e. Algorithm 6; and display the results in Figures 6b and 6c. The results show that both schemes produce very stable approximations, even at the time $T = 5$. In particular, the vortex patch is very well confined to the disk. There are minor differences between the WENO scheme of [13] that we denote by Algorithm 1 and the scheme proposed here (Algorithm 6). In particular, although both schemes produce small amplitude oscillations at the discontinuity, the WENO scheme proposed here resolves the discontinuity with oscillations of significantly smaller amplitude.

Summarizing, both WENO schemes are quite good at resolving discontinuities or sharp gradients. This is in sharp contrast to central finite difference scheme which can blow up at discontinuities due to spurious oscillations.

(a) 6th order Morinishi, $T = 1$ (b) Algorithm 1), $T = 5$ (c) Algorithm 5), $T = 5$
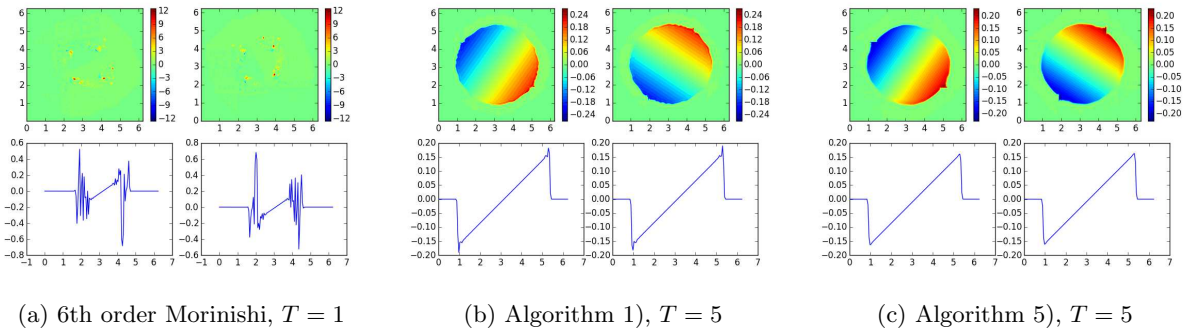
Figure 6: Simulations of discontinuous vortex patch problem. In each figure, left plots represent u (2D, and 1D cut along $y = \pi$), right ones v (2D, and 1D cut along $x = \pi$).

4.2 Taylor vortex

In this experiment, we fix the domain $[-8, 8]^2$ with periodic boundary conditions and solve the anelastic equations (1.1), (1.3) with buoyancy $b = 0$ and $\rho_0 \equiv 1$, i.e. incompressible Euler, and the following initial condition,

$$u_0(x, y) = -ye^{\frac{1}{2}(1-x^2-y^2)} + 8 \tag{4.1}$$

$$v_0(x, y) = xe^{\frac{1}{2}(1-x^2-y^2)} \tag{4.2}$$

As no exact solution is available, we compute a reference solution on a very fine grid of 8748×8748 points, and an appropriately small time step, using a sixth-order Morinishi scheme of [8]. We will compute approximations with the WENO scheme of [13] (Algorithm 1) and the WENO scheme proposed here (Algorithm 6) and compute corresponding errors with respect to the reference solution.

To guarantee that the leading error term in the advection scheme stems from the spatial discretization, we require $\Delta x^q > \Delta t^3$ (as we use SSP-RK3). For this reason, we limit these tests to $q \in \{3, 5\}$ (i.e. $k = \{2, 3\}$ in Algorithms 1 and 6), and we take a fixed Δt with $\Delta t^3 < \Delta x^5$. For the resolutions considered, $\Delta t = 0.0001$ suffices. We run our tests up to $T = 0.01$ (i.e. 100 time steps). For both the original and new schemes, we have compared third and fifth order approximations, on grids where $n_x = n_y$. We show here the discrete L^1 error for u with respect to the reference solution.

The numerical error results in Table 1 are consistent with our theoretical findings. The new scheme converges at a rate that is dictated by the order of the underlying approximations whereas the WENO scheme used by [13] is restricted to at most second order accuracy. Interestingly, the version of Algorithm

	Advection scheme (Alg. 1)				High-order formulation (Alg. 5)			
n_x	u (3rd)	EOC	u (5th)	EOC	u (3rd)	EOC	u (5th)	EOC
12	0.0469	–	0.0424	–	0.0529	–	0.0465	–
36	0.0267	0.51	0.00132	3.15	0.0069	1.66	0.00167	3.03
108	0.00802	1.09	8.41e-5	2.51	0.000419	2.55	7.45e-6	4.93
324	0.00284	0.94	9.89e-6	1.94	2.67e-5	2.51	3.89e-8	4.78
972	0.000939	1.01	1.06e-6	2.03	1.39e-5	2.69	1.88e-10	4.85

Table 1: Discrete L^1 numerical error for experiment 4.2 (Taylor vortex)

1 using third order interpolations shows only first order convergence. This is on account of the choice of the stencil in step 1 of Algorithm 1: instead of taking $2k$ points, in the description of the scheme in [13], the $(2k-2)$-point symmetric stencil was chosen. Revisiting the proof of Theorem 1, it is easy to see that taking a $(2k-2)$-nd order accurate approximation, the order of convergence is $\min\{2, 2k-3\}$; with $k = 2$ (i.e. WENO3) this only produces first-order accuracy.

4.3   Double shear layer

In the last numerical experiment of this section, we present a well-known example that has been used before, for instance in Zhang and Jackson [23], Ghosh and Baeder [5].

For $(x, y) \in [0, 2\pi) \times [0, 2\pi)$ with periodic boundary conditions, we solve the anelastic equations (1.1), (1.3), with initial data,

$$u_0(x, y) = \begin{cases} \tanh\left[\frac{1}{\rho}\left(y - \frac{\pi}{2}\right)\right] & \text{if } y < \pi \\ \tanh\left[\frac{1}{\rho}\left(\frac{3\pi}{2} - y\right)\right] & \text{if } y \geq \pi \end{cases}, \qquad v_0(x, y) = \delta \sin(x)$$

Following [5], we set $\rho = \frac{\pi}{15}$, $\delta = 0.05$, and approximate the problem on a grid of 256×256 points, using an underlying WENO5 scheme and a SSP-RK3 time marching method. We take a CFL number of 0.2. For the WENO scheme proposed here (Algorithm 6), reconstructions are performed with an ENO5 method. Figure 7 shows a numerical approximation of the vorticity $(-\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})$ at the cell centers, at three different times. There are very minor differences between the two WENO schemes, with the scheme proposed here adding slightly less dissipation. Both schemes approximate the solution very well, even at late times. This is in marked contrast to the behavior of well-known central schemes such as the schemes of [8] and [19]. This contrast can be seen from Figure 8 where the vorticity, computed with sixth-order versions of both central schemes are shown. In contrast to the WENO schemes, the central schemes seem to become unstable at later times. This example clearly demonstrates the superiority of WENO type schemes in long time integration of anelastic equations.

# 5   An arbitrarily high-order WENO scheme for advecting scalars in anelastic flows

Advection of scalars by an anelastic flow is modeled as:

$$\frac{\partial \phi}{\partial t} + \frac{1}{\rho_0} \sum_{i=1}^{d} \frac{\partial(\rho_0 u_i \phi)}{\partial x_i} = \Phi. \tag{5.1}$$

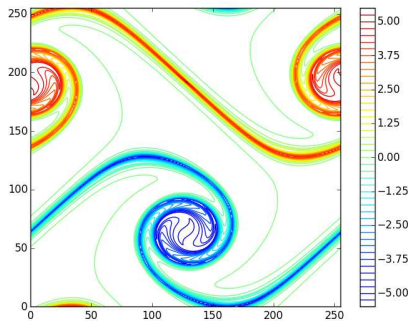Here, $\phi : \mathbb{R}^d \to \mathbb{R}$ is the scalar that is advected by the flow field $U = \{u_i\}_{1 \leq i \leq d}$, that solves the anelastic equations (1.1), (1.3), and Φ represents sources and sinks for the scalar. Our aim is to discretize the scalar
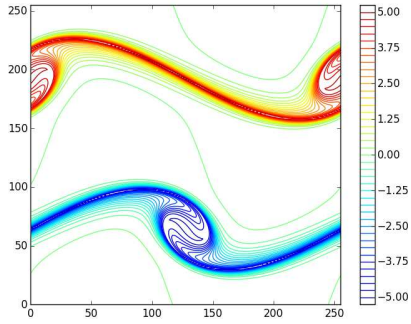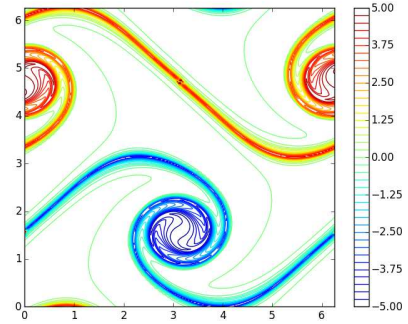
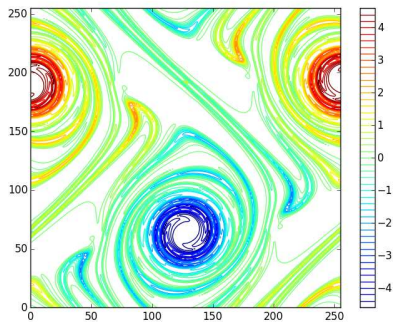(a) Vorticity at $t = 5$, Algorithm 1
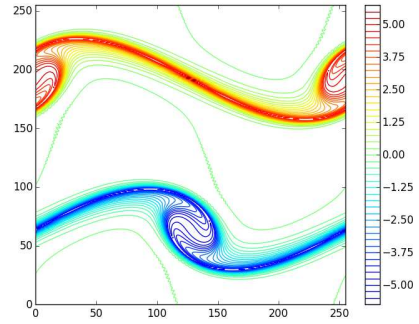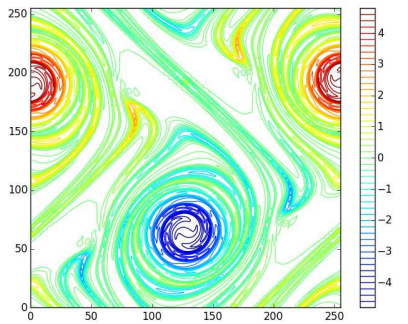
(b) Vorticity at $t = 5$, Algorithm 5

(c) Vorticity at $t = 7$, Algorithm 1

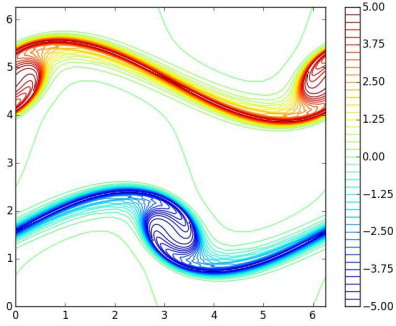(d) Vorticity at $t = 7$, Algorithm 5
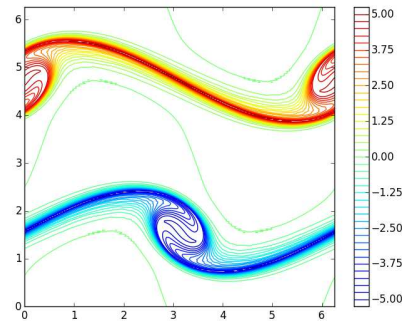
(e) Vorticity at $t = 14$, Algorithm 1
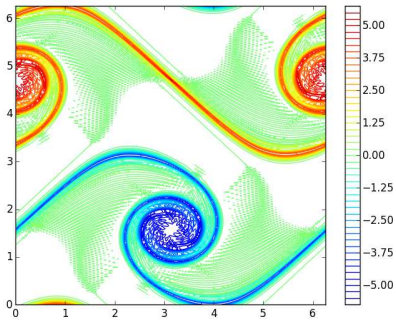
(f) Vorticity at $t = 14$, Algorithm 5

Figure 7: Vorticity for problem 4.3. We compare Algorithm 1 and 5 for the advection of momentum, otherwise using the same pressure solver, time-stepping, etc.
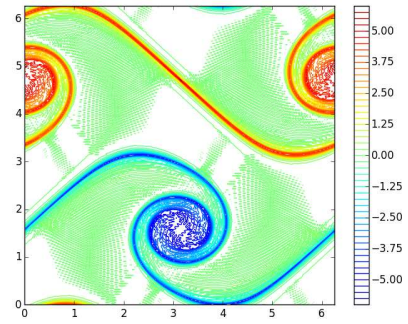
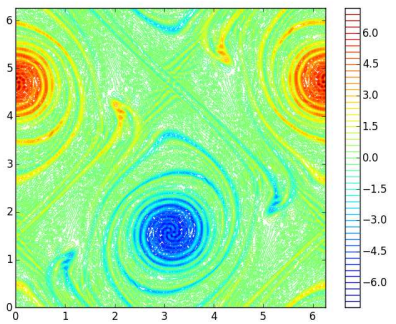(a) Vorticity at $t = 5$, Morinishi-6

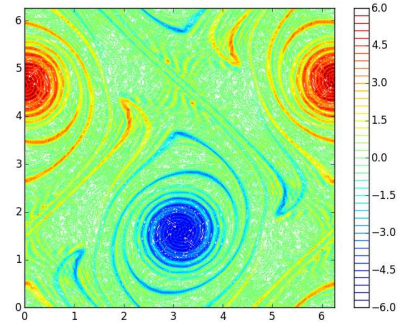(b) Vorticity at $t = 5$, Wicker-Skamarock-6

(c) Vorticity at $t = 7$, Morinishi-6

(d) Vorticity at $t = 7$, Wicker-Skamarock-6

(e) Vorticity at $t = 14$, Morinishi-6

(f) Vorticity at $t = 14$, Wicker-Skamarock-6

Figure 8: Comparison of the central schemes in Appendix A: 6th order Morinishi and Wicker-Skamarock, at the same time instants as in Fig. 7.
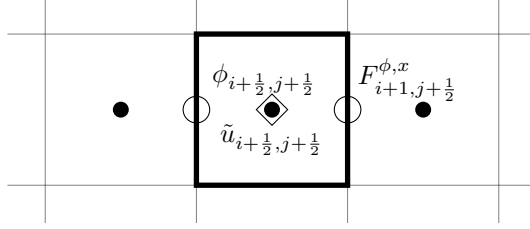
Figure 9: Reconstruction of $\frac{\partial}{\partial x}(u\phi)$ on a staggered grid. The diamond marks the location of ϕ in the highlighted cell; numerical fluxes must be calculated at the circles. Interpolations of u are required at the points marked with dots.

advection equation (5.1) on the Arakawa-C staggered grid. As pointed out in section 2 and shown in Figure 9, point values of the scalar ϕ are stored at cell centers, i.e $\phi_{i+\frac{1}{2},j+\frac{1}{2}} \approx \phi(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$.

In [13], a WENO scheme for the advection of scalars is presented, with spatial derivatives being discretized in a manner completely analogous to Algorithm 1. Reiterating the argument of Theorem 1, one can readily show that the resulting scheme is at most second-order accurate, even if very high-order interpolating polynomials are utilized within the WENO procedure. Our aim is to design an arbitrarily high-order accurate WENO scheme.

5.1 The high-order scheme

We start by using the divergence constraint and recasting the scalar advection equation (5.1) in the *non-conservative form*,

$$\frac{\partial\phi}{\partial t} + \frac{1}{\rho_0}\sum_{i=1}^{3}\frac{\partial(\rho_0 u_i\phi)}{\partial x_i} = \Phi \qquad \overset{\nabla\cdot(\rho_0 U)=0}{\Longleftrightarrow} \qquad \frac{\partial\phi}{\partial t} + \sum_{i=1}^{3}u_i\frac{\partial\phi}{\partial x_i} = \Phi \tag{5.2}$$

We choose to discretize this non-conservative form of scalar advection on account of observed better stability of this formulation, as well as for slightly lower computational costs.

The basis of our WENO discretization of (5.2) is to adapt ideas presented in section 3 to the present setting. For simplicity, we consider the problem in two space dimensions and assume a constant density ρ_0. Then, (5.2) reduces to

$$\phi_t + u\partial_x\phi + v\partial_y\phi = \Phi. \tag{5.3}$$

Here, $U = (u, v)$ is the velocity that solves (1.1), (1.3) in two space dimensions. A semi-discrete finite difference scheme (on the Arakawa-C grid) for discretizing (5.3) is given by

$$\frac{d}{dt}\phi_{i+\frac{1}{2},j+\frac{1}{2}}(t) + D^{\phi,x}_{i+\frac{1}{2},j+\frac{1}{2}} + D^{\phi,y}_{i+\frac{1}{2},j+\frac{1}{2}} = \Phi_{i+\frac{1}{2},j+\frac{1}{2}}. \tag{5.4}$$

Here, $\Phi_{i+\frac{1}{2},j+\frac{1}{2}} = \Phi(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$ is a point evaluation of the source term in (5.3) and

$$D^{\phi,x}_{i+\frac{1}{2},j+\frac{1}{2}} \approx (u\partial_x\phi)(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}), \quad D^{\phi,y}_{i+\frac{1}{2},j+\frac{1}{2}} \approx (v\partial_y\phi)(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}), \tag{5.5}$$

are discretizations of the non-conservative products of velocities and spatial derivatives of the scalar. Thus, the scheme (5.4) needs a routine to calculate spatial derivatives in a non-oscillatory manner and to very high order of accuracy. We follow ideas presented in section 3 and propose the following algorithm for defining $D^{\phi,x}_{i+\frac{1}{2},j+\frac{1}{2}}$. The definition of $D^{\phi,y}_{i+\frac{1}{2},j+\frac{1}{2}}$ is analogous.

Algorithm 7.
   Goal: Find $D^{\phi,x}_{i+\frac{1}{2},j+\frac{1}{2}}$ such that:

$$D_{i+\frac{1}{2},j+\frac{1}{2}}^{\phi,x} = \left(u \frac{\partial \phi}{\partial x} \right) (x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) + O(\Delta x^{2k-1} + \Delta y^{2k-1}), \quad 1 \leq k \in \mathbb{N}$$

1. Compute high-order approximations to the velocity field u at cell centers,

$$\tilde{u}_{i+\frac{1}{2},j+\frac{1}{2}} = u(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) + O(\Delta x^{2k-1})$$

2. Compute the two biased WENO-$(2k-1)$ reconstructions of ϕ along x at $(x_{i+1}, y_{j+\frac{1}{2}})$:

$$\hat{\phi}_{i+1,j+\frac{1}{2}}^{-} = W_{+\frac{1}{2}}(\phi_{i+\frac{1}{2}-(k-1),j+\frac{1}{2}}, \phi_{i+\frac{1}{2}-(k-2),j+\frac{1}{2}}, \ldots, \phi_{i+\frac{1}{2}+(k-1),j+\frac{1}{2}})$$

$$\hat{\phi}_{i+1,j+\frac{1}{2}}^{+} = W_{-\frac{1}{2}}(\phi_{i+\frac{1}{2}-(k-2),j+\frac{1}{2}}, \phi_{i+\frac{1}{2}-(k-1),j+\frac{1}{2}}, \ldots, \phi_{i+\frac{1}{2}+k,j+\frac{1}{2}})$$

3. Choose a numerical flux function $G(u^-, u^+)$ (for instance upwind with respect to $\tilde{u}_{i+\frac{1}{2},j+\frac{1}{2}}$), and set

$$\hat{\phi}_{i+1,j+\frac{1}{2}} := G(\hat{\phi}_{i+1,j+\frac{1}{2}}^{-}, \hat{\phi}_{i+1,j+\frac{1}{2}}^{+})$$

4. Define

$$D_{i+\frac{1}{2},j+\frac{1}{2}}^{\phi,x} = \tilde{u}_{i+\frac{1}{2},j+\frac{1}{2}} \frac{\hat{\phi}_{i+1,j+\frac{1}{2}} - \hat{\phi}_{i,j+\frac{1}{2}}}{\Delta x}$$

Remark 5. For the first step of Algorithm 7 we require high-order interpolations of the advecting velocity at cell centers. We recall from section 3 that these values have been already computed as an intermediate step of the interpolation procedure (Algorithm 2).

We can define $D_{i+\frac{1}{2},j+\frac{1}{2}}^{\phi,y}$ by readily adapting Algorithm 7 and using the reconstructed values of v at cell centers. The scheme (5.4) can be easily extended to include variable densities and to discretize (5.2) in three space dimensions. A straightforward truncation error analysis, together with the arbitrarily high-order nature of WENO interpolations, demonstrates that the resulting scheme is as accurate as the order of the underlying WENO reconstructions.

5.2 Numerical experiments

5.2.1 Passive scalar

We design this experiment to test the order of accuracy of the WENO schemes. In order do so, we will *manufacture* an exact solution of (5.3) in the two-dimensional periodic domain $[0, 2\pi]^2$ by setting,

$$u(x, y, t) = -\cos(t) \sin(x) \sin(2y)$$
$$v(x, y, t) = \cos(t) \cos(x) \sin(y)^2$$
$$\phi(x, y, t) = 2 + \cos(x) \sin(y) \cos(t),$$

and computing the source term Φ analytically in order to satisfy (5.3), as well as source terms for velocity so that eq. (1.1) holds. We observe that the velocity field (u, v) defined above is divergence free and satisfies the two-dimensional form of the anelastic equations with a suitable pressure term.

We compare the approximations computed for $T = 1$ with the exact solution, measuring error in L^1 norm. We choose Δt small enough to ensure that the leading error term comes from spatial discretization. Table 2 shows the numerical error for u (as the values for v are very similar) and ϕ. In this table, we compare the errors with the WENO scheme of [13] (denoted by Algorithm 1) and the WENO scheme proposed here (Algorithm 6-7). We see from the table that that there is a loss of order of convergence with the previously existing WENO scheme and it is limited to at most second order accuracy even when interpolation polynomials of a higher degree are used. On the other hand, the proposed scheme recovers the design order of accuracy and has an error that is three to four orders of magnitude smaller than the existing WENO schemes.

20

<div align="center">

L^1 error (velocity and scalar)

</div>

n_x	(Alg. 1)				(Alg. 6-7)			
	u (5th)	EOC	ϕ (5th)	EOC	u (5th)	EOC	ϕ (5th)	EOC
16	0.103	–	0.395	–	0.127	–	0.563	–
32	0.0367	1.489	0.126	1.651	0.01	3.661	0.0497	3.503
64	0.0104	1.823	0.0341	1.882	0.00042	4.578	0.00198	4.651
128	0.00269	1.947	0.00895	1.931	1.47e-05	4.84	7.68e-05	4.685
256	0.000688	1.968	0.00231	1.957	4.85e-07	4.916	2.62e-06	4.872
512	0.000174	1.983	0.000586	1.977	1.57e-08	4.949	8.54e-08	4.942

Table 2: Discrete L^1 numerical error for experiment 5.2.1 (Passive scalar)

5.2.2 Active scalar: entropy and buoyancy

Next, we consider a more realistic example of an *active scalar*, transported by the anelastic flow. The scalar represents s, the dry entropy. This, in turn, induces buoyancy (cf. the momentum equation (1.1)), which will drive the evolution, following the equations:

$$b(x, y, z) = g \frac{T(x, y, z) - T_0(z)}{T_0(z)} \tag{5.6}$$

$$T(x, y, z) = \exp\left(\frac{s(x, y, z)/\rho_0(z) - \tilde{s} + R_d \log(p_0(z)/\tilde{p})}{c_{pd}} \right), \tag{5.7}$$

where T is air temperature, variables with subindex 0 stand for values in the reference state, tildes denote standard surface values ($\tilde{p} = 10^5$ Pa, $\tilde{s} = 6864.8$ J kg^{-1}K^{-1}), $g = 9.8$ m/s^2, $R_d = 287.1$ J kg^{-1}K^{-1} and $c_{pd} = 1004$ J kg^{-1}K^{-1}. A detailed derivation of expressions (5.6) and (5.7) can be found in [13], for the more general case of moist air.

The test case is taken from Straka et al. [21]. The domain is $D = [-25600, 25600] \times [0, 6400]$. Here, the boundaries at $z = \{0, 6400\}$ are no-flow, with the boundaries in x still periodic. We begin by a cold perturbation of the reference state. Let $L : \mathbb{R}^2 \longrightarrow \mathbb{R}^+$,

$$L(x, z) = \left(\left(\frac{x}{4000} \right)^2 + \left(\frac{z - 3000}{2000} \right)^2 \right)^{\frac{1}{2}}$$

The perturbation in temperature from the reference atmospheric state is

$$\Delta T(x, z) = -7.5 \left(\cos(\min\{L(x, z), 1\} \, \pi) + 1 \right)$$

The evolution of the bubble is computed over a 512×512 grid, and results, computed with the fifth order versions of Algorithm 5 for momentum and Algorithm 7 for scalars, are presented in Figure 10. Due to the symmetry of the problem, plots show only the subdomain $[0, 22400] \times [0, 4000]$ (comprising 192×320 mesh points). The plot shows values for entropy temperature,

$$\theta(x, z) := 300 \exp\left(\frac{s(x, z) - \tilde{s}}{c_{pd}} \right).$$

Variations in the total amount of entropy, during the full simulation, remain in the order of 0.01%. This suggests that the non-conservative character of the scheme has a minor effect, if at all, on the results. Moreover, the figure also shows that the proposed high-order WENO scheme of this paper approximates this realistic bubble evolution very well.
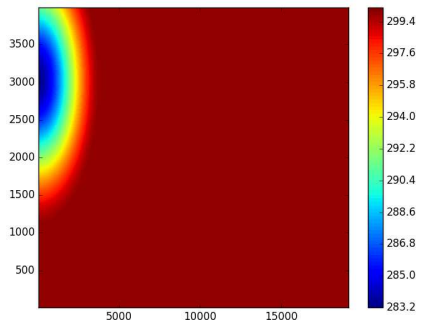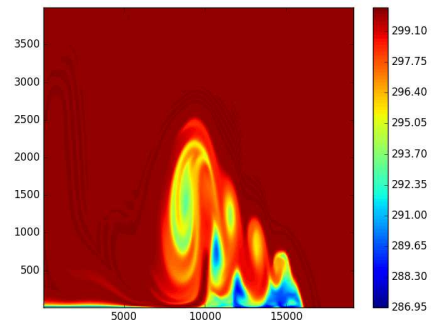
(a) $T = 0$
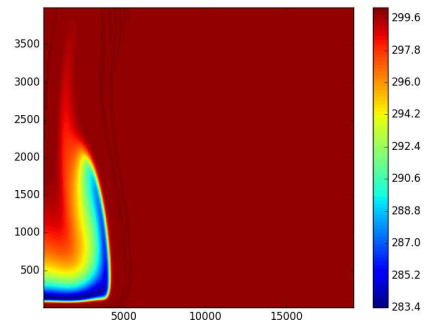
(b) $T = 300$
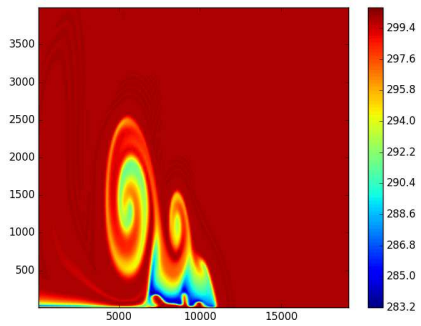
(c) $T = 600$

(d) $T = 900$

Figure 10: Entropy temperature θ for test case 5.2.2 in a 2D vertical domain (y axis represents height)

6 Summary

We consider the anelastic equations (1.1), (1.3) and passive or active scalars, advected by the anelastic flow (5.1). These equations arise frequently in atmospheric sciences and other related fields. Although (high-order) central finite difference schemes have been extensively used to discretize these equations, they are deficient at resolving sharp gradients, such as in large eddy simulations (LES) of clouds, or at ensuring positivity or bound preservation for scalars. Consequently, WENO type schemes, that are well-suited for simulation of flows with sharp gradients, have been designed in recent years to approximate the anelastic flow equations.

We show that these WENO schemes on the underlying staggered grid, such as the scheme described in [13], suffer from a loss in the design order of convergence due to the staggering of the velocity components on cell edges. In particular, the scheme is restricted to at most second order of accuracy, irrespective of the underlying (high-order) piecewise polynomial reconstructions. Given this deficiency, we propose a novel scheme that is guaranteed to be arbitrarily high-order accurate. This scheme is based on a novel ENO interpolation of the velocity components to a set of collocated points and the subsequent use of a WENO reconstruction of the spatial derivatives. This combination ensures that the overall scheme has a order of accuracy which is consistent with the underlying WENO method. The advection scheme is complemented with standard pressure solvers and SSP Runge-Kutta methods. Moreover, we design a WENO scheme that discretizes scalars advected by the anelastic flow (5.1). This scheme is based on a non-conservative formulation and reuses the ENO interpolated velocities at cell centers. It is also shown to be arbitrarily high-order accurate.

We have presented various numerical experiments comparing different schemes and summarize the results below,

- Both the proposed WENO scheme and the WENO scheme of [13] are very robust at resolving discontinuities and sharp gradients. We noticed a slight increase in robustness with our proposed scheme. This robustness of WENO schemes is in marked contrast to the failure of central difference schemes at resolving sharp gradients on account of the generation of spurious oscillations.

- The proposed WENO scheme was verified to possess an order of convergence that it is consistent with the design order, i.e the order of accuracy dictated by the underlying order of piecewise polynomials. This should be contrasted with existing WENO scheme of [13] and even the central difference schemes of [19], where the observed order of convergence was at most two. We demonstrate this gain in accuracy with both velocity fields as well as scalars.

- Both WENO schemes were observed to be particularly suitable for problems with long time integration, when compared to central schemes.

Based on the above observations, it is reasonable to argue that WENO schemes are considerably superior when approximating anelastic flows containing sharp gradients. Since such flows are ubiquitous, WENO schemes constitute a natural discretization framework for anelastic flows. The main advantage of the proposed scheme over existing WENO schemes is the increased order of accuracy. This results in errors that are several orders of magnitude lower than those generated by existing WENO schemes on staggered grids. On the other hand, the proposed scheme is more computationally expensive as additional ENO interpolations have to be performed. However, the additional cost is of the same order as the cost for the WENO reconstructions in the first place. Hence, this additional cost can be justified if higher order of accuracy is desired for the given problem, particularly on reasonably coarse grids.

Acknowledgments

Appendices

A Central schemes on a staggered grid

In this appendix we review existing, popular central schemes for the advection of scalars on a staggered grid.

A.1 Wicker-Skamarock schemes

The discussion of the Wicker-Skamarock schemes will be done here for 1D grids. To generalize to higher dimensions it suffices to repeat the procedure independently for the flux in every direction.

The scheme discussed in this subsection is presented in Wicker and Skamarock [22] as follows: for $\frac{\partial}{\partial t}\phi + \frac{\partial}{\partial x}(u\phi) = 0$, we use a discretization of the type:

$$\frac{\phi_{i+\frac{1}{2}}^{n+1} - \phi_{i+\frac{1}{2}}^{n}}{\Delta t} = -\frac{F_{i+1}^{n} - F_{i}^{n}}{\Delta x} \tag{A.1}$$

where several possibilities (with different orders of accuracy) are proposed for the spatial discretization, e.g.:

$$F_i^{4th} := \frac{u_i}{12}[7(\phi_{i+\frac{1}{2}} + \phi_{i-\frac{1}{2}}) - (\phi_{i+\frac{3}{2}} + \phi_{i-\frac{3}{2}})] \tag{A.2}$$

$$F_i^{6th} = \frac{u_i}{60}[37(\phi_{i+\frac{1}{2}} + \phi_{i-\frac{1}{2}}) - 8(\phi_{i+\frac{3}{2}} + \phi_{i-\frac{3}{2}}) + (\phi_{i+\frac{5}{2}} + \phi_{i-\frac{5}{2}})] \tag{A.3}$$

A.1.1 Order of convergence for the Wicker-Skamarock scheme

In [22], the above expressions for the flux are shown to produce high-order accurate results for constant advecting velocities. However, in a general case where velocities are not constant, they are limited to second-order accuracy, as acknowledged in Skamarock and Klemp [18]. As an example, we prove it here for F^{4th}.

Theorem 2. *For u, ϕ sufficiently smooth, and taking $F = F^{4th}$ as in* (A.2), *the central scheme* (A.1) *is at most second order accurate.*

Proof. Let us write the fluxes as:

$$F_i^{4th} = u_i Q_i, \quad F_{i+1}^{4th} = u_{i+1} Q_{i+1}, \qquad \text{with } Q_i := \frac{1}{12}\left(7(\phi_{i+\frac{1}{2}} + \phi_{i-\frac{1}{2}}) - (\phi_{i+\frac{3}{2}} + \phi_{i-\frac{3}{2}})\right)$$

By Taylor expansion, we know that:

$$u_i = u(x_{i+\frac{1}{2}}) - u'(x_{i+\frac{1}{2}})\frac{\Delta x}{2} + \frac{u''(x_{i+\frac{1}{2}})}{2}\frac{\Delta x^2}{4} + O(\Delta x^3)$$

$$u_{i+1} = u(x_{i+\frac{1}{2}}) + u'(x_{i+\frac{1}{2}})\frac{\Delta x}{2} + \frac{u''(x_{i+\frac{1}{2}})}{2}\frac{\Delta x^2}{4} + O(\Delta x^3)$$

Then:

$$(F^{4th})_{i+1}^n - (F^{4th})_i^n = \left(u(x_{i+\frac{1}{2}}) + \frac{u''(x_{i+\frac{1}{2}})\Delta x^2}{8}\right)(Q_{i+1} - Q_i) + \frac{u'(x_{i+\frac{1}{2}})\Delta x}{2}(Q_{i+1} + Q_i) + O(\Delta x^3) \tag{A.4}$$

Using Taylor expansions of $\{\phi_{i-\frac{3}{2}}, \phi_{i-\frac{1}{2}}, \ldots, \phi_{i+\frac{5}{2}}\}$, immediately:

$$Q_{i+1} - Q_i = \phi'(x_{i+\frac{1}{2}})\Delta x - \frac{\phi^{(5)}(x_{i+\frac{1}{2}})\Delta x^5}{30} + O(\Delta x^7)$$

$$Q_{i+1} + Q_i = 2\phi(x_{i+\frac{1}{2}}) + \frac{\phi''(x_{i+\frac{1}{2}})\Delta x^2}{6} + O(\Delta x^4)$$

Replacing these values in equation (A.4), and discarding error terms Δx^k for $k \geq 3$, some computations give

$$\frac{(F^{4th})_{i+1}^n - (F^{4th})_i^n}{\Delta x} = (u\phi)'(x_{i+\frac{1}{2}}) + O(\Delta x^2)$$

$\square$

Remark 6. It is easy to check that all error terms of order Δx^2 have a derivative of u as a factor, whereas the first error term with no derivatives of u is $\frac{u\phi^{(5)}}{30}\Delta x^4$. Thus the design order is obtained for spatially constant advecting velocity.

A.2 Morinishi schemes

A second popular set of conservative approximations to the fluxes is described by Morinishi et al. [8]. They can quite easily be described in full 3D generality, although some notation is required to write them in a convenient manner:

$$\bar{\varphi}^{n,x}(x_i, y_j, z_k) := \frac{\varphi(x_i + \frac{n}{2}\Delta x, y_j, z_k) + \varphi(x_i - \frac{n}{2}\Delta x, y_j, z_k)}{2} \tag{A.5}$$

$$\frac{\delta_n \varphi}{\delta_n x}(x_i, y_j, z_k) := \frac{\varphi(x_i + \frac{n}{2}\Delta x, y_j, z_k) - \varphi(x_i - \frac{n}{2}\Delta x, y_j, z_k)}{n\Delta x} \tag{A.6}$$

The corresponding definitions for (A.5) (interpolation) and (A.6) (approximation to the derivative) in y and z are analogous. With these, the schemes can be written as follows.

A.2.1 4th order Morinishi scheme

$$\frac{\partial}{\partial t}(u_i) = \frac{9}{8}\sum_{j=1}^{3}\frac{\delta_1}{\delta_1 x_j}\left[\left(\frac{9}{8}\bar{u}_j^{1,x_i} - \frac{1}{8}\bar{u}_j^{3,x_i}\right)\bar{u}_i^{1,x_j}\right]$$

$$- \frac{1}{8}\sum_{j=1}^{3}\frac{\delta_3}{\delta_3 x_j}\left[\left(\frac{9}{8}\bar{u}_j^{1,x_i} - \frac{1}{8}\bar{u}_j^{3,x_i}\right)\bar{u}_i^{3,x_j}\right] + O(\Delta x^4)$$

A.2.2 6th order Morinishi scheme

$$\frac{\partial}{\partial t}(u_i) = \frac{150}{128}\sum_{j=1}^{3}\frac{\delta_1}{\delta_1 x_j}\left[\left(\frac{150}{128}\bar{u}_j^{1,x_i} - \frac{25}{128}\bar{u}_j^{3,x_i} + \frac{3}{128}\bar{u}_j^{5,x_i}\right)\bar{u}_i^{1,x_j}\right]$$

$$- \frac{25}{128}\sum_{j=1}^{3}\frac{\delta_1}{\delta_1 x_j}\left[\left(\frac{150}{128}\bar{u}_j^{1,x_i} - \frac{25}{128}\bar{u}_j^{3,x_i} + \frac{3}{128}\bar{u}_j^{5,x_i}\right)\bar{u}_i^{3,x_j}\right]$$

$$+ \frac{3}{128}\sum_{j=1}^{3}\frac{\delta_1}{\delta_1 x_j}\left[\left(\frac{150}{128}\bar{u}_j^{1,x_i} - \frac{25}{128}\bar{u}_j^{3,x_i} + \frac{3}{128}\bar{u}_j^{5,x_i}\right)\bar{u}_i^{5,x_j}\right] + O(\Delta x^6)$$

References

[1] A. ARAKAWA AND V. R. LAMB, *Computational design of the basic dynamical processes of the UCLA general circulation model*, in Methods in Computational Physics: Advances in Research and Applications, J. Chang, ed., vol. 17, Academic Press, New York, San Francisco, London, 1977, pp. 173–265.

[2] D. S. BALSARA AND C.-W. SHU, *Monotonicity Preserving Weighted Essentially Non-oscillatory Schemes with Increasingly High Order of Accuracy*, Journal of Computational Physics, 160 (2000), pp. 405–452.

[3] Y. M. H. D. GOTTLIEB AND S. ORSZAG, *Theory and application of spectral methods*, SIAM, 1984.

[4] S. GHOSAL, *An Analysis of Numerical Errors in Large-Eddy Simulations of Turbulence*, Journal of Computational Physics, 125 (1996), pp. 187–206.

[5] D. GHOSH AND J. D. BAEDER, *High-Order Accurate Incompressible Navier-Stokes Algorithm for Vortex-Ring Interactions with Solid Wall*, AIAA Journal, 50 (2012), pp. 2408–2422.

[6] A. HARTEN, B. ENGQUIST, S. OSHER, AND S. R. CHAKRAVARTHY, *Uniformly high order accurate essentially non-oscillatory schemes, III*, Journal of Computational Physics, 71 (1987), pp. 231–303.

[7] K. LILLY, *The representation of small-scale turbulence in numerical simulation experiments*, Proc. IBM Scientific Computing Symp. on Environmental Sciences (Yorktown Heights, New York), IBM form no. 320-1951 (1966).

[8] Y. MORINISHI, T. LUND, O. VASILYEV, AND P. MOIN, *Fully Conservative Higher Order Finite Difference Schemes for Incompressible Flow*, Journal of Computational Physics, 143 (1998), pp. 90–124.

[9] S. PATANKAR, *Numerical Heat Transfer and Fluid Flow*, Series in computational methods in mechanics and thermal sciences, Taylor & Francis, 1980.

[10] O. PAULUIS, *Thermodynamic consistency of the anelastic approximation for a moist atmosphere*, Journal of the Atmospheric Sciences, 65 (2008), pp. 2719–2729.

[11] S. B. POPE, *Turbulent flows*, Cambridge university press, 2000.

[12] ——, *Ten questions concerning the large-eddy simulation of turbulent flows*, New journal of Physics, 6 (2004), p. 35.

[13] K. G. PRESSEL, C. M. KAUL, T. SCHNEIDER, Z. TAN, AND S. MISHRA, *Large-eddy simulation in an anelastic framework with closed water and entropy balances*, Journal of Advances in Modeling Earth Systems, 7 (2015), pp. 1425–1456.

[14] K. G. PRESSEL, S. MISHRA, T. SCHNEIDER, C. M. KAUL, AND Z. TAN, *Numerics and Subgrid-Scale Modeling in Large Eddy Simulations of Stratocumulus Clouds*, Journal of Advances in Modeling Earth Systems, 9 (2017), pp. 1342–1365.

[15] P. SAGAUT, *Large eddy simulation for incompressible flows: an introduction*, Springer Science & Business Media, 2006.

[16] C.-W. SHU, *Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws*, ICASE Report, (1997), pp. 1–78.

[17] C.-W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii*, Journal of Computational Physics, 83 (1989), pp. 32 – 78.

[18] W. C. SKAMAROCK AND J. B. KLEMP, *A time-split nonhydrostatic atmospheric model for weather research and forecasting applications*, Journal of Computational Physics, 227 (2008), pp. 3465–3485.

[19] W. C. Skamarock, J. B. Klemp, J. Dudhia, D. O. Gill, M. Barker, K. G. Duda, X. Y. Huang, W. Wang, and J. G. Powers, *A description of the Advanced Research WRF Version 3*, tech. rep., National Center for Atmospheric Research, 2008.

[20] J. Smagorinsky, *General Circulation Experiments with the Primitive Equations*, Monthly Weather Review, 91 (1963), p. 99.

[21] J. M. Straka, R. B. Wilhelmson, L. J. Wicker, J. R. Anderson, and K. K. Droegemeier, *Numerical solutions of a non-linear density current: A benchmark solution and comparisons*, International Journal for Numerical Methods in Fluids, 17 (1993), pp. 1–22.

[22] L. J. Wicker and W. C. Skamarock, *Time-Splitting Methods for Elastic Models Using Forward Time Schemes*, Monthly Weather Review, 130 (2002), pp. 2088–2097.

[23] J. Zhang and T. L. Jackson, *A high-order incompressible flow solver with WENO*, Journal of Computational Physics, 228 (2009), pp. 2426–2442.