AIAA Journal
Vol. 54, No. 4, April 2016

Well-Balanced, Conservative Finite Difference Algorithm for Atmospheric Flows

Debojyoti Ghosh* and Emil M. Constantinescu†
Argonne National Laboratory, Argonne, Illinois 60439

DOI: 10.2514/1.J054580

The numerical simulation of meso-, convective-, and microscale atmospheric flows requires the solution of the Euler or the Navier–Stokes equations. Nonhydrostatic weather prediction algorithms often solve the equations in terms of derived quantities such as Exner pressure and potential temperature (and are thus not conservative) and/or as perturbations to the hydrostatically balanced equilibrium state. This paper presents a well-balanced, conservative finite difference formulation for the Euler equations with a gravitational source term, where the governing equations are solved as conservation laws for mass, momentum, and energy. Preservation of the hydrostatic balance to machine precision by the discretized equations is essential because atmospheric phenomena are often small perturbations to this balance. The proposed algorithm uses the weighted essentially nonoscillatory and compact-reconstruction weighted essentially nonoscillatory schemes for spatial discretization that yields high-order accurate solutions for smooth flows and is essentially nonoscillatory across strong gradients; however, the well-balanced formulation may be used with other conservative finite difference methods. The performance of the algorithm is demonstrated on test problems as well as benchmark atmospheric flow problems, and the results are verified with those in the literature.

Nomenclature

$\mathcal{A}$	=	flux Jacobian matrix
a	=	Runge–Kutta coefficients for stage calculation
b	=	Runge–Kutta coefficients for step completion
c	=	optimal weights for weighted essentially nonoscillatory scheme
$\mathcal{D}$	=	discretization operator
e	=	energy per unit mass, J/m^3
e^*	=	nondimensional energy per unit mass
$\boldsymbol{F}, \boldsymbol{G}$	=	flux vectors along x, y
$\boldsymbol{f}$	=	one-dimensional flux function (vector)
f	=	one-dimensional flux function (scalar)
$\boldsymbol{g}$	=	gravitational force (per unit mass) vector, m/s^2
$\boldsymbol{g}^*$	=	nondimensional gravitational force (per unit mass) vector
g	=	gravitational force per unit mass, m/s^2
i, j	=	grid indices
$\boldsymbol{L}$	=	right-hand-side operator for semidiscrete ordinary differential equation
n	=	index for time integration
p	=	pressure, N/m^2
p^*	=	nondimensional pressure
$\boldsymbol{Q}$	=	Runge–Kutta stage values of $\boldsymbol{q}$
$\boldsymbol{q}$	=	state vector
R	=	universal gas constant, J/kg · K
$\mathcal{R}$	=	interpolation operator
r	=	order of spatial discretization scheme
$\boldsymbol{s}$	=	source term
T	=	temperature, K
t	=	time, s
t^*	=	nondimensional time
$\boldsymbol{u}$	=	velocity vector, m/s
$\boldsymbol{u}^*$	=	nondimensional velocity vector
u, v	=	flow velocity components in x, y, m/s
X, X^{-1}	=	matrices with right and left eigenvectors as columns and rows, respectively
$\boldsymbol{x}$	=	spatial position vector, m
$\boldsymbol{x}^*$	=	nondimensional spatial position vector
x, y	=	spatial position
α	=	nonconvex weighted essentially nonoscillatory weights
β	=	smoothness indicators for weighted essentially nonoscillatory discretization
γ	=	specific heat ratio
ϵ	=	parameter for weighted essentially nonoscillatory discretization
θ	=	potential temperature, K
κ	=	dissipation factor due to gravitational field
Λ	=	diagonal matrix of eigenvalues
ν	=	dissipation factor in Rusanov's upwinding
π	=	Exner pressure
ρ	=	density, kg/m^3
ρ^*	=	nondimensional density
ϱ	=	density variation function for stratified atmosphere
σ	=	interpolation coefficients
τ	=	parameter for weighted essentially nonoscillatory discretization
ϕ	=	scalar function
φ	=	pressure variation function for stratified atmosphere
ω	=	nonlinear weighted essentially nonoscillatory weights

I. Introduction

RECENT decades have seen the development of several numerical algorithms for the accurate simulation of atmospheric flows. Hydrostatic models [1,2] or simplified models that remove acoustic waves [3–5] are often used for flows with large horizontal scales (such as planetary simulations). Nonhydrostatic effects are significant when simulating meso-, convective-, and microscale atmospheric phenomena; thus, the solution to the compressible Euler equations is required [6–8]. Various formulations of the Euler equations have been proposed in the literature and used for operational weather prediction software [9,10]. Algorithms that solve the Euler equations in terms of derived quantities that are relevant to atmospheric flows [11–15] (such as Exner pressure and potential temperature) do not conserve mass, momentum, and energy, even though the numerical discretization may be conservative. Several algorithms [8,16–20] solve the governing

Presented as Paper 2015-2889 at the 7th AIAA Atmospheric and Space Environments Conference, Dallas, TX, 22–26 June 2015; received 30 June 2015; revision received 12 October 2015; accepted for publication 24 November 2015; published online 12 January 2016.

*Postdoctoral Appointee, Mathematics and Computer Science Division; ghosh@mcs.anl.gov. Member AIAA.

†Computational Mathematician, Mathematics and Computer Science Division; emconsta@mcs.anl.gov.

equations in terms of mass and momentum conservation and use the assumption of adiabaticity to simplify the energy conservation equation to the conservation of potential temperature [21]. Recent efforts solve the Euler equations as conservation laws for mass, momentum, and energy [9,22–24], with no additional assumptions. This form of the governing equations is identical to that used in simulating compressible aerodynamic flows [25,26]. Thus, a conservative discretization conserves the mass, momentum, and energy to machine precision, and the true viscous terms may be specified if needed. This approach is followed here.

Several operational algorithms to simulate atmospheric flows [12,16,27] are based on the finite difference method; however, they have been criticized for their low spectral resolution (due to low-order spatial discretization) and the lack of scalability [19]. These drawbacks have been addressed to some extent by high-order finite volume [8,17,20,22,24] and finite element or spectral-element [9,10,19] methods. Often, atmospheric phenomena are characterized by strong gradients, and stable solutions are obtained by using a total variation diminishing/bounded discretization (such as slope-limited methods [17,20,24]), applying a filter [9], or adding artificial diffusion [8,9]. On the other hand, the weighted essentially nonoscillatory (WENO) schemes [28,29] use solution-dependent interpolation stencils to yield high-order accurate nonoscillatory solutions and have been applied successfully to several application areas [30] including compressible and incompressible fluid dynamics. The compact-reconstruction WENO (CRWENO) schemes [31] apply the solution-dependent stencil selection to compact finite difference schemes [32] and thus have higher spectral resolution than the standard WENO schemes. The CRWENO schemes have been applied to turbulent flows [33] and aerodynamic flows [34] where the resolution of small length scales is crucial. Although the CRWENO schemes require the solution to banded systems of equations at every time-integration step or stage, a scalable implementation of the CRWENO scheme [35] demonstrated its performance for massively parallel simulations. The accuracy, spectral resolution, and scalability of the WENO and CRWENO schemes make them well suited for the simulation of atmospheric flows.

The Euler equations, with the addition of gravitational and Coriolis forces as source terms, govern the dynamics of atmospheric flows and constitute a hyperbolic balance law. This paper focuses on meso-, convective-, and microscale flows, and the Coriolis forces are neglected. Balance laws admit steady states where the flux derivatives are balanced by the source terms. In the context of atmospheric flows, steady states are in hydrostatic balance, where the pressure gradient is counteracted by the gravitational body force. Numerical methods must be able to preserve such steady states on a finite grid to machine precision. Atmospheric phenomena are often small perturbations to the hydrostatic balance, and thus errors in balancing the discretized pressure gradient with the gravitational forces have the potential to overwhelm the flow. One way to ensure the preservation of this balance is by subtracting the hydrostatically balanced quantities from the flow variables and expressing the equations in terms of the perturbations [8,9,20]. Alternatively, well-balanced discretization methods for the governing equations can be formulated that preserve the hydrostatic balance on a finite grid. Balanced finite volume methods have been proposed [36,37] and applied to the Euler equations with gravitational source terms [17,22,24]. A well-balanced, conservative finite difference formulation for the shallow water equations was introduced [38] and extended to general balance laws [39] as well as finite volume and discontinuous Galerkin discretizations [40]. This formulation was applied to the Euler equations with gravitational source terms [41]; however, it was derived only for the special case of an isothermal equilibrium, and other balanced equilibria or flow problems relevant to atmospheric flows were not considered.

This paper presents a high-order, well-balanced finite difference algorithm for atmospheric flows. The governing equations are solved as conservation laws for mass, momentum, and energy. The balanced formulation of Xing and Shu [39,41] for conservative finite difference methods is extended to a more general form of the hydrostatic balance that includes, as specific cases, the isothermal equilibrium [41], as well as other examples of stratified atmosphere encountered in the literature. The fifth-order WENO and CRWENO schemes are used in this study; however, the balanced formulation may be used with other discretization schemes expressed in the conservative finite difference form. In the absence of gravitational forces, the proposed algorithm reduces to the standard finite difference discretization of the Euler equations. One of the motivations for this approach is to develop a unified numerical framework for both aerodynamic and atmospheric flows. Explicit, multistage Runge–Kutta methods are used for time integration; in the future, efficient semi-implicit methods [10,18] will be explored. The ability of the algorithm to maintain the hydrostatically balanced equilibrium to machine precision is demonstrated. The algorithm is verified by solving benchmark atmospheric flow problems, and the results are compared with those obtained with two operational weather prediction solvers: Weather Research and Forecasting (WRF) [16] and Nonhydrostatic Unified Model of the Atmosphere (NUMA) [9].

The outline of the paper is as follows. Section II describes the governing equations. The numerical method, including the well-balanced formulation, is described in Sec. III. The algorithm is verified and results for benchmark flow problems are presented in Sec. IV. The Appendix contains three specific examples of the general well-balanced formulation.

II. Governing Equations

The dynamics of atmospheric flows are governed by the Navier–Stokes equations [26], with the addition of gravitational and Coriolis forces as source terms. The effects of viscosity are insignificant, and the inviscid Euler equations [25] are solved. Various equation sets have been used in the literature [9,10]. The algorithm proposed here solves the Euler equations stated as the conservation of mass, momentum, and energy. Mesoscale flows are considered, and Coriolis forces are neglected. The governing equations are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{1}$$

$$\frac{\partial (\rho \boldsymbol{u})}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u} + p \mathcal{I}_d) = -\rho \boldsymbol{g} \tag{2}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot (e + p)\boldsymbol{u} = -\rho \boldsymbol{g} \cdot \boldsymbol{u} \tag{3}$$

where ρ is the density, $\boldsymbol{u}$ is the velocity vector, p is the pressure, and $\boldsymbol{g}$ is the gravitational force vector (per unit mass). $\mathcal{I}_d$ denotes the identity matrix of size d, where d is the number of space dimensions. The energy is given by

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho \boldsymbol{u} \cdot \boldsymbol{u} \tag{4}$$

where γ is the specific heat ratio. The equation of state relates the pressure, density, and temperature as $p = \rho R T$, where R is the universal gas constant and T is the temperature. Two additional quantities of interest in atmospheric flows are the Exner pressure π and the potential temperature θ, defined as

$$\pi = \left(\frac{p}{p_0}\right)^{\frac{\gamma-1}{\gamma}}, \qquad \text{and} \quad \theta = \frac{T}{\pi} \tag{5}$$

respectively. The pressure at a reference altitude is denoted by p_0. Equations (1–3) may be nondimensionalized as follows:

$$\boldsymbol{x}^* = \frac{\boldsymbol{x}}{L_\infty}, \qquad \boldsymbol{u}^* = \frac{\boldsymbol{u}}{a_\infty}, \qquad t^* = \frac{t}{L_\infty / a_\infty},$$
$$\rho^* = \frac{\rho}{\rho_\infty}, \qquad p^* = \frac{p}{\rho a_\infty^2}, \qquad \boldsymbol{g}^* = \frac{\boldsymbol{g}}{a_\infty^2 / L_\infty} \tag{6}$$

where the subscript ∞ denotes reference quantities, the superscript $*$ denotes nondimensionalized variables, L_∞ is a characteristic length scale, and $a = \sqrt{\gamma p/\rho}$ is the speed of sound. The governing equations, expressed in terms of the nondimensionalized variables, are

$$\frac{\partial \rho^*}{\partial t^*} + \nabla \cdot (\rho^* \boldsymbol{u}^*) = 0 \tag{7}$$

$$\frac{\partial (\rho^* \boldsymbol{u}^*)}{\partial t^*} + \nabla \cdot (\rho^* \boldsymbol{u}^* \otimes \boldsymbol{u}^* + p^* \mathcal{I}_d) = -\rho^* \boldsymbol{g}^* \tag{8}$$

$$\frac{\partial e^*}{\partial t^*} + \nabla \cdot (e^* + p^*)\boldsymbol{u}^* = -\rho^* \boldsymbol{g}^* \cdot \boldsymbol{u}^* \tag{9}$$

where the ∇ operator now denotes derivatives with respect to the nondimensionalized vector $\boldsymbol{x}^*$. Equations (7–9) are identical in form to Eqs. (1–3). Thus, the same equations are used to solve both nondimensional and dimensional problems. Subsequent discussions describing the numerical discretization do not distinguish between the dimensional and nondimensional equations, and the superscript $*$ is omitted for convenience.

III. Numerical Method

This paper considers two-dimensional flows with gravity acting along the y dimension; however, the numerical methodology and the well-balanced formulation can be trivially extended to three-dimensional flows. Equations (1–3) constitute a hyperbolic system of partial differential equations and are discretized by a conservative finite difference method. The governing equations can be expressed as a hyperbolic conservation law:

$$\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{F}(\boldsymbol{q})}{\partial x} + \frac{\partial \boldsymbol{G}(\boldsymbol{q})}{\partial y} = \boldsymbol{s}(\boldsymbol{q}) \tag{10}$$

where the state vector, the flux vectors along x and y, and the source terms are

$$\boldsymbol{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (e+p)u \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (e+p)v \end{bmatrix},$$
$$\boldsymbol{s} = \begin{bmatrix} 0 \\ 0 \\ -\rho g \\ -\rho v g \end{bmatrix} \tag{11}$$

The Cartesian velocity components are u and v, and g is the gravitational force (per unit mass). Figure 1 shows part of a two-dimensional Cartesian grid around the grid point (i, j) whose spatial coordinates are $(x_i = i\Delta x,\ y_j = j\Delta y)$ along with the neighboring grid points and the cell interfaces. A conservative spatial discretization [42,43] of Eq. (10) on this grid yields a semidiscrete ordinary differential equation in time,

$$\frac{d\boldsymbol{q}_{ij}}{dt} + \frac{1}{\Delta x}[\hat{\boldsymbol{f}}_{i+1/2,j} - \hat{\boldsymbol{f}}_{i-1/2,j}] + \frac{1}{\Delta y}[\hat{\boldsymbol{g}}_{i,j+1/2} - \hat{\boldsymbol{g}}_{i,j-1/2}] = \boldsymbol{s}_{ij} \tag{12}$$

where $\boldsymbol{q}_{ij} = \boldsymbol{q}(x_i, y_j)$ is the cell-centered solution, $(x_i = i\Delta x, y_j = j\Delta y)$ are the spatial coordinates of a grid point, and i, j denote the grid indices. The numerical approximation to the flux function at the cell interfaces $\hat{\boldsymbol{f}}_{i+1/2,j} = \hat{\boldsymbol{f}}(x_{i\pm1/2,j})$, $\hat{\boldsymbol{g}}_{i,j+1/2} = \hat{\boldsymbol{g}}(y_{i,j\pm1/2})$ satisfies

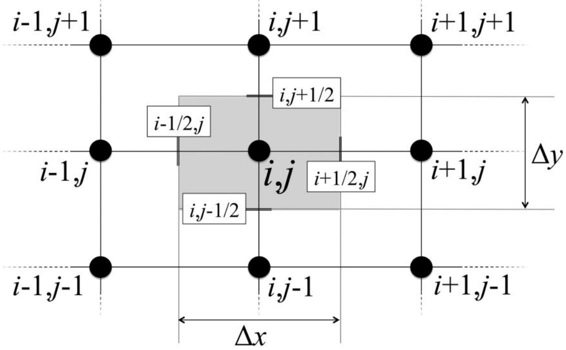


Fig. 1 Illustration of a two-dimensional Cartesian grid with the grid points (cell centers) and cell interfaces on which Eq. (10) is discretized with a conservative finite difference formulation. The gray shaded region represents a discrete cell.

$$\left.\frac{\partial \boldsymbol{F}}{\partial x}\right|_{x_i,y_j} = \frac{1}{\Delta x}[\hat{\boldsymbol{f}}(x_{i+1/2,j}, t) - \hat{\boldsymbol{f}}(x_{i-1/2,j}, t)] + O(\Delta x^r) \tag{13}$$

$$\left.\frac{\partial \boldsymbol{G}}{\partial y}\right|_{x_i,y_j} = \frac{1}{\Delta y}[\hat{\boldsymbol{g}}(y_{i,j+1/2}, t) - \hat{\boldsymbol{g}}(y_{i,j-1/2}, t)] + O(\Delta y^r) \tag{14}$$

for an rth-order spatial discretization method, and thus $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{g}}$ are the primitives of $\boldsymbol{F}$ and $\boldsymbol{G}$, respectively. The discretized source term $\boldsymbol{s}_{ij}$ in Eq. (12) is expressed as its cell-centered value, and this naive treatment does not preserve the hydrostatic equilibrium [41] except in the limit $\Delta x \to 0$. The treatment of the source term for a well-balanced formulation is discussed in subsequent sections.

Equation (12) can be rewritten as

$$\frac{d\tilde{\boldsymbol{q}}}{dt} = \boldsymbol{L}(\tilde{\boldsymbol{q}}) \tag{15}$$

where $\tilde{\boldsymbol{q}} = [\boldsymbol{q}_{ij}; 1 \le i \le N_i, 1 \le j \le N_j]$ is the entire solution vector on a grid with $N_i \times N_j$ points, and $\boldsymbol{L}$ denotes the right-hand-side operator comprising the discretized flux and source terms. Equation (15) is integrated in time with multistage explicit Runge–Kutta schemes, expressed as follows:

$$\boldsymbol{Q}^{(s)} = \tilde{\boldsymbol{q}}^n + \Delta t \sum_{t=1}^{s-1} a_{\mathrm{st}} \boldsymbol{L}(\boldsymbol{Q}^{(t)}), \qquad s = 1, \cdots, S \tag{16}$$

$$\tilde{\boldsymbol{q}}^{n+1} = \tilde{\boldsymbol{q}}^n + \Delta t \sum_{s=1}^{S} b_s \boldsymbol{L}(\boldsymbol{Q}^{(s)}) \tag{17}$$

where S is the number of stages, $\boldsymbol{Q}^{(s)}$ is the sth-stage value, a_{st} and b_s are the coefficients of the Butcher table [44], and the superscripts n and $n+1$ denote the time levels $t^n = n\Delta t$ and $t^{n+1} = (n+1)\Delta t$, respectively. The strong-stability-preserving third-order Runge–Kutta (SSPRK3) [45] and the classical fourth-order Runge–Kutta (RK4) schemes are used in this study; their Butcher tables are given by

$$\begin{array}{c|ccc} 0 & 0 & & \\ 1 & 1 & 0 & \\ 1/2 & 1/4 & 1/4 & 0 \\ \hline & 1/6 & 1/6 & 2/3 \end{array}, \quad \text{and} \quad \begin{array}{c|cccc} 0 & 0 & & & \\ 1/2 & 1/2 & 0 & & \\ 1/2 & 0 & 1/2 & 0 & \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array} \tag{18}$$

respectively.

A. Reconstruction

The reconstruction step computes the numerical flux primitives $\hat{\boldsymbol{f}}_{i\pm 1/2,j}$ and $\hat{\boldsymbol{g}}_{i,j\pm 1/2}$ in Eq. (12) from the cell-centered values of the flux functions $\boldsymbol{F}(\boldsymbol{q})$ and $\boldsymbol{G}(\boldsymbol{q})$, respectively. This section describes the approximation of a scalar, one-dimensional function primitive at the cell interface $\hat{f}_{j+1/2} = \hat{f}(x_{j+1/2})$ from the cell-centered values of the function $f_j = f(x_j)$. It is extended to multiple dimensions by carrying out a one-dimensional reconstruction at the interface along the grid line normal to that interface. Vector quantities are reconstructed in a componentwise manner where the reconstruction process described later is applied to each component of the vector independently. The fifth-order WENO [29] and CRWENO [31] schemes are used in this study.

The fifth-order WENO scheme (WENO5) is constructed by considering three third-order accurate interpolation schemes for the numerical flux $\hat{f}_{j+1/2}$:

$$\hat{f}^1_{j+1/2} = \frac{1}{3}f_{j-2} - \frac{7}{6}f_{j-1} + \frac{11}{6}f_j, \qquad c_1 = \frac{1}{10} \tag{19}$$

$$\hat{f}^2_{j+1/2} = -\frac{1}{6}f_{j-1} + \frac{5}{6}f_j + \frac{1}{3}f_{j+1}, \qquad c_2 = \frac{6}{10} \tag{20}$$

$$\hat{f}^3_{j+1/2} = \frac{1}{3}f_j + \frac{5}{6}f_{j+1} - \frac{1}{6}f_{j+2}, \qquad c_3 = \frac{3}{10} \tag{21}$$

where c_k, $k = 1, 2, 3$ are the optimal coefficients. Multiplying Eqs. (19–21) by the corresponding c_k and taking the sum yields a fifth-order interpolation scheme,

$$\hat{f}_{j+1/2} = \frac{1}{30}f_{j-2} - \frac{13}{60}f_{j-1} + \frac{47}{60}f_j + \frac{27}{60}f_{j+1} - \frac{1}{20}f_{j+2} \tag{22}$$

The optimal coefficients c_k are replaced by nonlinear weights (ω_k, $k = 1, 2, 3$), and the WENO5 scheme is the weighted sum of Eqs. (19–21) with these nonlinear weights:

$$\begin{aligned}\hat{f}_{j+1/2} = {}& \frac{\omega_1}{3}f_{j-2} - \frac{1}{6}(7\omega_1 + \omega_2)f_{j-1} + \frac{1}{6}(11\omega_1 + 5\omega_2 + 2\omega_3)f_j \\ & + \frac{1}{6}(2\omega_2 + 5\omega_3)f_{j+1} - \frac{\omega_3}{6}f_{j+2}\end{aligned} \tag{23}$$

The weights are evaluated based on the smoothness of the solution [46],

$$\omega_k = \frac{\alpha_k}{\sum_k \alpha_k}; \qquad \alpha_k = c_k\left[1 + \left(\frac{\tau}{\epsilon + \beta_k}\right)^2\right] \tag{24}$$

where

$$\tau = (f_{j-2} - 4f_{j-1} + 6f_j - 4f_{j+1} + f_{j+2})^2 \tag{25}$$

The parameter $\epsilon = 10^{-6}$ prevents division by zero, and the smoothness indicators (β_k) are given by

$$\beta_1 = \frac{13}{12}(f_{j-2} - 2f_{j-1} + f_j)^2 + \frac{1}{4}(f_{j-2} - 4f_{j-1} + 3f_j)^2 \tag{26}$$

$$\beta_2 = \frac{13}{12}(f_{j-1} - 2f_j + f_{j+1})^2 + \frac{1}{4}(f_{j-1} - f_{j+1})^2 \tag{27}$$

and $$\beta_3 = \frac{13}{12}(f_j - 2f_{j+1} + f_{j+2})^2 + \frac{1}{4}(3f_j - 4f_{j+1} + f_{j+2})^2 \tag{28}$$

Other definitions for the nonlinear weights exist in the literature [47–49] as well as a comparison of the nonlinear properties of the WENO5 scheme with these weights [33]. When the solution is smooth, the nonlinear weights converge to the optimal coefficients ($\omega_k \to c_k$), and Eq. (23) reduces to Eq. (22). The scheme is fifth-order accurate for such solutions. Across and near discontinuities, the weights corresponding to the stencil containing the discontinuity approach zero, and Eq. (23) represents an interpolation scheme with its stencil biased away from the discontinuity. Nonoscillatory solutions are thus obtained.

The fifth-order CRWENO scheme (CRWENO5) [31] is similarly constructed by considering three third-order accurate compact interpolation schemes [32] for the numerical flux $\hat{f}_{j+1/2}$:

$$\frac{2}{3}\hat{f}^1_{j-1/2} + \frac{1}{3}\hat{f}^1_{j+1/2} = \frac{1}{6}(f_{j-1} + 5f_j); \qquad c_1 = \frac{2}{10} \tag{29}$$

$$\frac{1}{3}\hat{f}^2_{j-1/2} + \frac{2}{3}\hat{f}^2_{j+1/2} = \frac{1}{6}(5f_j + f_{j+1}); \qquad c_2 = \frac{5}{10} \tag{30}$$

$$\frac{2}{3}\hat{f}^3_{j+1/2} + \frac{1}{3}\hat{f}^3_{j+3/2} = \frac{1}{6}(f_j + 5f_{j+1}); \qquad c_3 = \frac{3}{10} \tag{31}$$

where c_k, $k = 1, 2, 3$ are the optimal coefficients. A fifth-order compact scheme is obtained by multiplying Eqs. (29–31) by their corresponding optimal coefficient c_k and adding

$$\frac{3}{10}\hat{f}_{j-1/2} + \frac{6}{10}\hat{f}_{j+1/2} + \frac{1}{10}\hat{f}_{j+3/2} = \frac{1}{30}f_{j-1} + \frac{19}{30}f_j + \frac{1}{3}f_{j+1} \tag{32}$$

The CRWENO5 scheme is constructed by replacing the optimal coefficients c_k by nonlinear weights ω_k. It can be expressed as

$$\begin{aligned}&\left(\frac{2}{3}\omega_1 + \frac{1}{3}\omega_2\right)\hat{f}_{j-1/2} + \left[\frac{1}{3}\omega_1 + \frac{2}{3}(\omega_2 + \omega_3)\right]\hat{f}_{j+1/2} \\ &\quad + \frac{1}{3}\omega_3\hat{f}_{j+3/2} \\ &\quad = \frac{\omega_1}{6}f_{j-1} + \frac{5(\omega_1 + \omega_2) + \omega_3}{6}f_j + \frac{\omega_2 + 5\omega_3}{6}f_{j+1}\end{aligned} \tag{33}$$

The weights ω_k are computed by Eqs. (24) and (26–28). The resulting scheme, given by Eq. (33), is fifth-order accurate when the solution is smooth ($\omega_k \to c_k$) and reduces to Eq. (32). Across and near discontinuities, the weights corresponding to the stencils containing the discontinuity approach zero, and a biased (away from the discontinuity) compact scheme is obtained. Equation (33) results in a tridiagonal system of equations that must be solved at each time-integration step or stage; however, the additional expense is justified by the higher accuracy and spectral resolution of the compact scheme [31,33,34]. An efficient and scalable implementation of the CRWENO5 scheme was recently proposed in [35,50] and is used in this study.

The solution of a hyperbolic system is composed of waves propagating at their characteristic speeds along their characteristic directions, and thus the final flux at the interface is an appropriate combination of the left- and right-biased fluxes. The description of the preceding WENO5 and CRWENO5 schemes considered a left-biased reconstruction of the scalar flux; the corresponding right-biased reconstruction can be obtained by reflecting the expressions around the interface $j + 1/2$. The upwinding schemes of Roe [51,52] and Rusanov [52,53] are used in this study. Roe's scheme is expressed as

$$\hat{\boldsymbol{f}}_{j+1/2} = \frac{1}{2}[\hat{\boldsymbol{f}}^L_{j+1/2} + \hat{\boldsymbol{f}}^R_{j+1/2} - |\mathcal{A}(\hat{\boldsymbol{q}}^L_{j+1/2}, \hat{\boldsymbol{q}}^R_{j+1/2})|(\hat{\boldsymbol{q}}^R_{j+1/2} - \hat{\boldsymbol{q}}^L_{j+1/2})] \tag{34}$$

where

$$|\mathcal{A}(\hat{\boldsymbol{q}}^L_{j+1/2}, \hat{\boldsymbol{q}}^R_{j+1/2})| = X_{j+1/2}|\Lambda_{j+1/2}|X^{-1}_{j+1/2} \tag{35}$$

The eigenvalues Λ and the eigenvectors X, X^{-1} are evaluated at the interface from the Roe-averaged flow variables. The superscripts L and R indicate the left- and right-biased interpolations, respectively. Rusanov's scheme is given by

$$\hat{\boldsymbol{f}}_{j+1/2} = \frac{1}{2}[\hat{\boldsymbol{f}}^L_{j+1/2} + \hat{\boldsymbol{f}}^R_{j+1/2} - (\max_{j,j+1} \nu)(\hat{\boldsymbol{q}}^R_{j+1/2} - \hat{\boldsymbol{q}}^L_{j+1/2})] \tag{36}$$

The dissipation factor is $\nu = a + |\boldsymbol{u}|$, where a is the speed of sound, and $\boldsymbol{u}$ is the flow velocity. We note that $\hat{\boldsymbol{q}}^{L,R}_{j+1/2}$ in Eqs. (34) and (36) are the left- and right-biased interface values for $\boldsymbol{q}$ that are reconstructed in the same manner as $\hat{\boldsymbol{f}}^{L,R}_{j+1/2}$.

B. Well-Balanced Formulation

A hyperbolic balance law, such as Eq. (10), admits steady-state solutions where the flux derivative is exactly balanced by the source term. For atmospheric flows, the gravitational force on the fluid is balanced by the pressure gradient, resulting in the hydrostatic balance. The numerical algorithm must preserve this balance to machine precision because errors have the potential to overwhelm physically relevant perturbations to the balance. In this section, a well-balanced formulation of the finite difference discretization of Eq. (10) is presented; this formulation reduces to the balanced discretization previously proposed [41] for the specific case of an isothermal hydrostatic balance. Although the formulation is described for two-dimensional flows with gravity acting along the y dimension, it can be easily extended to three dimensions and for domains where the gravity may not be aligned with a specific dimension.

Steady atmospheric flow in hydrostatic balance can be expressed in the following general form:

$$u = \text{constant}, \qquad v = 0, \qquad \rho = \rho_0 \varrho(y), \qquad p = p_0 \varphi(y) \tag{37}$$

where the subscript 0 indicates the flow variables at a reference altitude, and ϱ, φ are scalar functions. The flow quantities are a function of y only because the gravitational force is assumed to act along the y direction. The pressure and density at the reference altitude are related by the equation of state $p_0 = \rho_0 R T_0$. At equilibrium, Eq. (10) reduces to

$$\frac{dp}{dy} = -\rho g \tag{38}$$

Therefore, by substituting Eq. (37) in Eq. (38) and considering the equation of state, the functions $\varrho(y)$ and $\varphi(y)$ satisfy

$$RT_0[\varrho(y)]^{-1}\varphi'(y) = -g \tag{39}$$

where $\varphi'(y)$ is the y derivative of $\varphi(y)$. The necessity of a well-balanced algorithm can be explained as follows. Let a general, linear finite difference approximation to the derivative of an arbitrary function $\phi(y)$ be expressed as

$$\left.\frac{\partial \phi}{\partial y}\right|_{y=y_j} \approx \mathcal{D}[\phi] \equiv \frac{1}{\Delta y} \sum_{k=-m}^{n} \sigma^{\mathcal{D}}_k \phi_{j+k} \tag{40}$$

where m and n are integers defining the stencil $[j-m, j-m+1, \cdot \cdot \cdot, j+n-1, j+n]$ of the finite difference operator $\mathcal{D}$, and $\sigma^{\mathcal{D}}_k$ are the stencil coefficients. With this notation, the discretized form of Eq. (38) at a grid point can be written as

$$\mathcal{D}[p]_j = -(\rho g)_j \tag{41}$$

where the subscript j indicates the corresponding terms evaluated at the jth grid point. If $\mathcal{D}$ is a consistent finite difference operator, Eq. (41) is exactly satisfied as $\Delta x \to 0$. On a finite grid with $\Delta x \neq 0$, however, the error in satisfying Eq. (41) is nonzero and is related to the spatial discretization error of the finite difference operator $\mathcal{D}$.

A well-balanced algorithm must satisfy Eq. (38) in its discretized form on a finite grid ($\Delta x \neq 0$) as well; thus, the discretized flux derivative must exactly balance the discretized source term. The first step modifies Eq. (10) as

$$\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{F}(\boldsymbol{q})}{\partial x} + \frac{\partial \boldsymbol{G}(\boldsymbol{q})}{\partial y} = \boldsymbol{s}^*(\boldsymbol{q}, y) \tag{42}$$

where $\boldsymbol{s}^* = [0, 0, \rho R T_0[\varrho(y)]^{-1}\varphi'(y), \rho v R T_0[\varrho(y)]^{-1}\varphi'(y)]^T$. The relationship between $\varrho(y)$ and $\varphi(y)$, given by Eq. (39), ensures that Eq. (42) is consistent with Eq. (10). The source terms are thus rendered in a form similar to that of the flux term [39,41]. With this modification, Eq. (38) can be rewritten as

$$\frac{dp}{dy} = \rho R T_0[\varrho(y)]^{-1}\varphi'(y) \tag{43}$$

It can then be discretized by using the notation in Eq. (40) to yield

$$\mathcal{D}_{\boldsymbol{G}}[p] - \rho R T_0\{\varrho(y)\}^{-1}\mathcal{D}_{\boldsymbol{s}^*}[\varphi(y)] = 0 \tag{44}$$

where $\mathcal{D}_{\boldsymbol{G}}$ and $\mathcal{D}_{\boldsymbol{s}^*}$ are the finite difference operators used to approximate the y derivatives of the flux function $\boldsymbol{G}$ and the source term $\boldsymbol{s}^*$, respectively. Although Eq. (44) holds true for $\Delta x \to 0$ if $\mathcal{D}_{\boldsymbol{G}}$ and $\mathcal{D}_{\boldsymbol{s}^*}$ are both consistent finite difference operators, it is not exactly satisfied for $\Delta x \neq 0$ with the error being related to the spatial discretization errors of $\mathcal{D}_{\boldsymbol{G}}$ and $\mathcal{D}_{\boldsymbol{s}^*}$. However, Eq. (44) is exactly satisfied for $\Delta x \neq 0$ if

$$\mathcal{D}_{\boldsymbol{G}} = \mathcal{D}_{\boldsymbol{s}^*} = \mathcal{D} \tag{45}$$

Substituting Eq. (45) and exploiting the linearity of $\mathcal{D}$, the left-hand side of Eq. (44) reduces to

$$\begin{aligned} \mathcal{D}[p - \rho R T_0\{\varrho(y)\}^{-1}\varphi(y)] &= \mathcal{D}[p_0\varphi(y) - \rho_0\varrho(y) R T_0\{\varrho(y)\}^{-1}\varphi(y)] \\ &= 0 \end{aligned} \tag{46}$$

The term $\rho R T_0\{\varrho(y)\}^{-1} = p_0$ in Eq. (44) is constant, and hence it can be moved inside the discretized derivative operator $\mathcal{D}$. Equation (45) implies that a linear finite difference algorithm to solve Eq. (42) is well balanced (preserves hydrostatically balanced steady states to machine precision) if the flux derivative, and the modified source terms are discretized by the same linear operator.

The WENO5 and CRWENO5 schemes are nonlinear finite difference operators because their coefficients are solution-dependent. The standard procedure to compute the discretized flux derivative along y with these schemes can be summarized as follows:

$$\hat{\boldsymbol{G}}^{L,R}_{j+1/2} = \mathcal{R}^{L,R}_{\boldsymbol{G}}[\boldsymbol{G}] \equiv \sum_{k=-m}^{n} \hat{\sigma}_k(\omega)\boldsymbol{G}_{j+k} \tag{47}$$

$$\hat{\boldsymbol{G}}_{j+1/2} = \frac{1}{2}[\hat{\boldsymbol{G}}^{L}_{j+1/2} + \hat{\boldsymbol{G}}^{R}_{j+1/2} + |\mathcal{A}_{j+1/2}|(\hat{\boldsymbol{q}}^{L}_{j+1/2} - \hat{\boldsymbol{q}}^{R}_{j+1/2})] \tag{48}$$

$$\text{or } \hat{\boldsymbol{G}}_{j+1/2} = \frac{1}{2}[\hat{\boldsymbol{G}}^{L}_{j+1/2} + \hat{\boldsymbol{G}}^{R}_{j+1/2} + (\max_{j,j+1} \nu)(\hat{\boldsymbol{q}}^{L}_{j+1/2} - \hat{\boldsymbol{q}}^{R}_{j+1/2})] \tag{49}$$

$$\left.\frac{\partial \boldsymbol{G}}{\partial y}\right|_{y=y_j} \approx \frac{1}{\Delta y}[\hat{\boldsymbol{G}}_{j+1/2} - \hat{\boldsymbol{G}}_{j-1/2}] \tag{50}$$

where j is the grid index along the y coordinate (the index along the x coordinate is suppressed for convenience of notation), $\mathcal{R}^{L,R}_{\boldsymbol{G}}$ are the reconstruction operators with m and n defining the stencil bounds, and the coefficients for the stencil points $\hat{\sigma}_k$ are functions of the solution-dependent nonlinear weights ω. Equations (23) and (33), representing the WENO5 and CRWENO5 schemes, can be represented through this operator. The subscript $\boldsymbol{G}$ denotes that the nonlinear weights ω are computed based on $\boldsymbol{G}(\boldsymbol{q})$. The superscripts L and R denote the left- and right-biased operators, respectively. The additional steps needed to construct a well-balanced algorithm are now described.

The derivative of $\varphi(y)$ in the source term of Eq. (42) is discretized in the same manner as the flux derivative, summarized as follows:

$$\hat{\boldsymbol{\varphi}}^{L,R}_{j+1/2} = \mathcal{R}^{L,R}_{\boldsymbol{G}}[\boldsymbol{\varphi}] \equiv \sum_{k=-m}^{n} \hat{\sigma}_k \boldsymbol{\varphi}_{j+k} \tag{51}$$

$$\hat{\boldsymbol{\varphi}}_{j+1/2} = \frac{1}{2}[\hat{\boldsymbol{\varphi}}^{L}_{j+1/2} + \hat{\boldsymbol{\varphi}}^{R}_{j+1/2}] \tag{52}$$

$$\left.\frac{\partial \boldsymbol{\varphi}}{\partial y}\right|_{y=y_j} \approx \frac{1}{\Delta y}[\hat{\boldsymbol{\varphi}}_{j+1/2} - \hat{\boldsymbol{\varphi}}_{j-1/2}] \tag{53}$$

where the vector $\boldsymbol{\varphi}$ is simply given by $\boldsymbol{\varphi} = [0, 0, \varphi(y), \varphi(y)]^T$. The remaining terms in the source $\boldsymbol{s}^*$ are evaluated at the cell center j. The interpolation operators in Eqs. (47) and (51) are both $\mathcal{R}_{\boldsymbol{G}}$; the interface values of both $\boldsymbol{G}$ and $\boldsymbol{\varphi}$ are computed with the same interpolation operator. This is achieved in the WENO5 and CRWENO5 schemes by calculating the weights based on the smoothness of the flux function $\boldsymbol{G}(\boldsymbol{q})$ and using these weights to compute the interface values of both $\boldsymbol{G}$ and $\boldsymbol{\varphi}$ at a given time-integration step or stage.

The final step in the construction of a well-balanced method is the suitable modification of the dissipation term in the upwinding step. The Roe and Rusanov schemes, given by Eqs. (48) and (49), are modified as follows.

Roe:

$$\hat{\boldsymbol{G}}_{j+1/2} = \frac{1}{2}[\hat{\boldsymbol{G}}^{L}_{j+1/2} + \hat{\boldsymbol{G}}^{R}_{j+1/2} + \kappa|\mathcal{A}_{j+1/2}|(\hat{\boldsymbol{q}}^{*,L}_{j+1/2} - \hat{\boldsymbol{q}}^{*,R}_{j+1/2})] \tag{54}$$

Rusanov:

$$\hat{\boldsymbol{G}}_{j+1/2} = \frac{1}{2}[\hat{\boldsymbol{G}}^{L}_{j+1/2} + \hat{\boldsymbol{G}}^{R}_{j+1/2} + \kappa(\max_{j,j+1} \nu)(\hat{\boldsymbol{q}}^{*,L}_{j+1/2} - \hat{\boldsymbol{q}}^{*,R}_{j+1/2})] \tag{55}$$

where $\kappa = \max_{j,j+1}\varphi(y)$, and $\hat{\boldsymbol{q}}^{*,L}_{j+1/2}$ and $\hat{\boldsymbol{q}}^{*,R}_{j+1/2}$ are respectively the left- and right-biased interpolation (at the interface) of a modified state vector $\boldsymbol{q}^* = [\varrho\{\rho(y)\}^{-1}, \rho u\{\varrho(y)\}^{-1}, \rho v\{\varrho(y)\}^{-1}, e^*]^T$. The modified energy e^* is given by

$$e^* = \frac{p\{\varphi(y)\}^{-1}}{\gamma - 1} + \frac{1}{2}\rho\{\varrho(y)\}^{-1}(u^2 + v^2) \tag{56}$$

At steady state, $\boldsymbol{q}^*$ is a constant, and the dissipation term in Eqs. (54) and (55) is zero with this modification ($\hat{\boldsymbol{q}}^{*,L}_{j+1/2} = \hat{\boldsymbol{q}}^{*,R}_{j+1/2}$).

Discretization of the flux derivatives in Eq. (42) by Eq. (47), Eq. (54), or Eq. (55) and Eq. (50) as well as evaluation of the source term as Eqs. (51–53) results in the following discretized form of the steady-state equation [Eq. (43)] at grid point j:

$$p_0\left[\frac{\hat{\varphi}_{j+1/2} - \hat{\varphi}_{j-1/2}}{\Delta y}\right] = \rho_j R T_0\{\varrho(y_j)\}^{-1}\left[\frac{\hat{\varphi}_{j+1/2} - \hat{\varphi}_{j-1/2}}{\Delta y}\right] \tag{57}$$

The interface approximation of the scalar function $\varphi(y)$ is denoted by $\hat{\varphi}_{j+1/2}$. It is evaluated on the left-hand side through Eq. (47) and Eq. (54) or Eq. (55) and is evaluated on the right-hand side through Eqs. (51) and (52). Equation (57) is exactly satisfied if the operator $\mathcal{R}_{\boldsymbol{G}}$ is linear. Although $\mathcal{R}_{\boldsymbol{G}}$ represents nonlinear finite difference operators, given by Eqs. (23) and (33), the nonlinearity of these schemes arises from the solution-dependent weights ω_k. Within a time-integration step or stage, these weights are computed and fixed, and the operator $\mathcal{R}_{\boldsymbol{G}}$ is essentially linear. Therefore, Eq. (57) is exactly satisfied.

C. Summary

The steps to construct a well-balanced conservative finite difference algorithm are summarized as follows.

1) The governing equations are modified as Eq. (42).

2) The flux derivatives are computed through Eq. (47), Eq. (54) or Eq. (55), and Eq. (50).

3) The derivatives in the modified source term are computed through Eqs. (51–53); the remaining terms are evaluated at the cell centers.

The resulting algorithm preserves a hydrostatically balanced steady state to machine precision. The modified procedure to compute the flux derivative is applied to both the x and y dimensions; in the absence of gravitational forces along a particular dimension, for example $g = 0 \Rightarrow \varrho(y) = \varphi(y) = 1$, it reduces to the standard flux computation given by Eqs. (47–50). Three examples of the steady state [Eq. (37)] that occur in atmospheric flow problems are presented in the Appendix as well as the resulting expressions for the modified source term in Eq. (42) and modified solution in Eqs. (54) and (55).

IV. Verification and Results

This section demonstrates the performance of the numerical algorithm and verifies the computed results with those in the literature. The ability of the algorithm to preserve the hydrostatic balance to machine precision and accurately capture small perturbations is demonstrated. Comparisons are made with a naive discretization of the source term to show the necessity for the well-balanced formulation. Further, benchmark atmospheric flow problems are solved, and the results obtained agree with those obtained with solvers based on other forms of the governing equations and using different discretization techniques. Both dimensional and nondimensional problems are considered; the descriptions of the former have the relevant units specified. One-dimensional problems in y are solved with the two-dimensional code by specifying an arbitrary domain size in the x dimension; the number of grid points in x is taken as the number of ghost points required to implement the boundary treatment, and periodic boundary conditions are applied along this dimension.

A. Well-Balanced Tests

A one-dimensional problem is initially considered to demonstrate the need for a well-balanced discretization. The initial flow is static and in hydrostatic balance. An isothermal equilibrium is considered with a sinusoidal gravitational field potential [41,54]

$$\phi = -\frac{1}{2\pi}\ \sin(2\pi y) \tag{58}$$

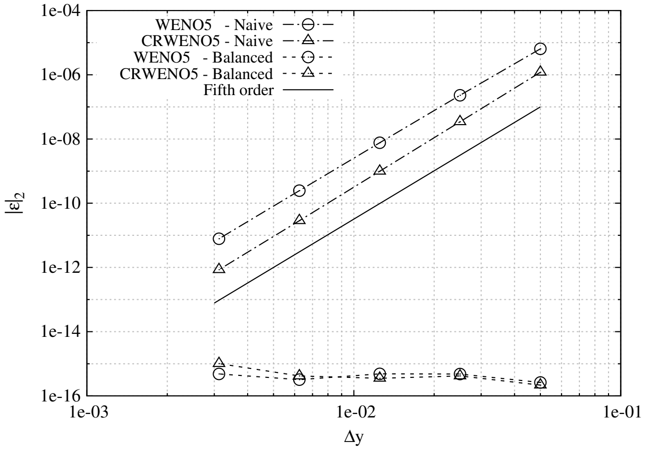


Fig. 2 Relative error (L_2 norm) with respect to the initial solution for a steady-state problem with naive and balanced implementations of the algorithm.

specified on a periodic domain $0 \le y < 1$. The gravitational force in Eq. (10) is $g(y) = \phi_y$ (where the subscript denotes the derivative), and the steady state is given by

$$\rho = \exp(-\phi), \qquad p = \exp(-\phi) \tag{59}$$

The specific heat ratio is $\gamma = 1.4$, and the solutions are obtained at a final time of 1.0. This problem is solved with the balanced discretization as well as with a naive discretization that can be described by Eq. (12), where the source term is discretized as $\boldsymbol{s}_{ij} = [0, 0, -(\rho g)_{ij}, -(\rho v g)_{ij}]^T$, and the flux terms are discretized by the standard WENO5 and CRWENO5 schemes. The Roe scheme is used for upwinding: Eq. (34) for the naive algorithm, and Eq. (54) for the balanced algorithm. Because the problem is steady, the error in the final solution is defined as

$$\|\boldsymbol{\epsilon}\|_{(\cdot)} = \frac{\|\boldsymbol{q}(\boldsymbol{x}, t_f) - \boldsymbol{q}(\boldsymbol{x}, 0)\|_{(\cdot)}}{\|\boldsymbol{q}(\boldsymbol{x}, 0)\|_{(\cdot)}} \tag{60}$$

where t_f denotes the final time. Figure 2 shows the L_2 norm of the error as a function of the grid resolution Δy for the solutions obtained with the balanced and naive implementations. The RK4 method is used for time integration with a time step of 10^{-3} for the cases. The naive implementation results in a nonzero error for all the grid sizes, which shows that it is unable to preserve the hydrostatically balanced equilibrium. In addition, the error in preserving the equilibrium shows fifth-order convergence as the grid is refined for both the WENO5 and CRWENO5 schemes. The errors for the CRWENO5 schemes are an order of magnitude lower than those of the WENO5 scheme at all grid resolutions. These observations are consistent with the numerical properties of these schemes [31]. Therefore, the error in preserving the hydrostatic balance is indeed the spatial discretization error of the algorithm, as discussed regarding Eq. (41). The balanced implementation preserves the steady state to machine precision for both the WENO5 and CRWENO5 schemes.

The ability of the proposed formulation to preserve two examples of hydrostatic balance encountered in atmospheric flows is tested. These correspond to the second and third cases in the Appendix; the specific flow conditions are described later. The initial solution is specified as the balanced steady-state flow. The specific heat ratio is $\gamma = 1.4$ in all the examples.

1. Case 1

The first case corresponds to a stratified atmosphere with constant potential temperature θ (example 2 in the Appendix). The initial solution is specified by Eq. (A4) with $u = v = 0\ \mathrm{m/s^{-1}}$, $R = 287.058\ \mathrm{J/kg \cdot K}$, $T = 300\ \mathrm{K}$, $p_0 = 10^5\ \mathrm{N/m^2}$, and $g = 9.8\ \mathrm{m/s^2}$. The domain is $1000 \times 1000\ \mathrm{m^2}$ discretized by 51×51 points. Inviscid wall conditions are specified on all boundaries. This case represents the hydrostatic equilibrium for the two-dimensional rising thermal bubble problem [9]. A time step of 0.02 s is taken, and the solution is evolved until a final time of 1000 s with the RK4 method.

2. Case 2

The second case corresponds to a stratified atmosphere with a specified Brunt–Väisälä frequency $\mathcal{N}$ (example 3 in the Appendix). The initial solution is specified by Eq. (A11) with $\mathcal{N} = 0.01/\mathrm{s}$, $u = 20\ \mathrm{m/s}$, $v = 0\ \mathrm{m/s}$, $R = 287.058\ \mathrm{J/kg \cdot K}$, $T = 300\ \mathrm{K}$, $p_0 = 10^5\ \mathrm{N/m^2}$, and $g = 9.8\ \mathrm{m/s^2}$. The domain is $300{,}000 \times 10{,}000\ \mathrm{m^2}$ discretized by 1200×50 points. Periodic boundaries are specified along x, and inviscid wall boundaries are specified along y. This case represents the hydrostatic equilibrium for the inertia-gravity wave problem [9]. A time step of 0.25 s is taken, and the solution is evolved until a final time of 3000 s with the SSPRK3 method.

The two cases are solved with the balanced formulation as well as a naive discretization of the source term. The Roe scheme is used for upwinding in all the cases. Table 1 shows the L_1, L_2, and L_∞ norms of the error defined by Eq. (60). When the balanced formulation is used, the errors are zero to machine precision for both the CRWENO5 and WENO5 schemes in all the cases. However, the naive discretization of the source term results in a significant error in preserving the steady state. Thus, these tests demonstrate that the balanced algorithm is necessary to accurately preserve the hydrostatic balance.

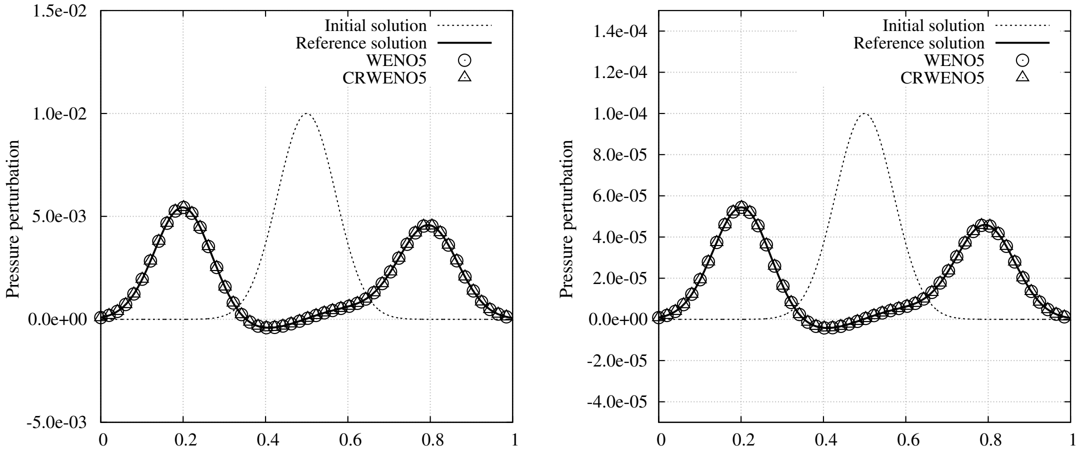
A one-dimensional problem [41] is used to test the accurate simulation of small perturbations to the hydrostatic balance. The initial solution represents a pressure perturbation to an isothermal hydrostatic equilibrium with a constant gravitational field of unit magnitude,

$$\begin{aligned}&\rho(y,0) = \exp(-y),\\ &p(y,0) = \exp(-y) + \eta \exp[-100(y-0.5)^2], \qquad u(y,0) = 0\end{aligned} \tag{61}$$

on a unit domain $y \in [0, 1]$ with extrapolative boundary conditions. The specific heat ratio is $\gamma = 1.4$. Solutions are obtained at a final time of 0.25 with the RK4 method, and a time step of 0.0025 [corresponding to a Courant–Friedrichs–Lewy (CFL) number of ~0.6]. Figure 3 shows the initial and final pressure perturbations $p(y,t) - \exp(-y)$, obtained with the WENO5 and CRWENO5 schemes on a grid with 200 points. The reference solutions are obtained with the CRWENO5 scheme on a grid with 2000 points. Two values of the perturbation strength η are considered: 10^{-2} and 10^{-4}. The computed solutions agree well with the reference solutions as well as with results reported in the literature [41]. These results demonstrate that the algorithm is able to accurately capture both strong and weak perturbations to the balanced steady state.

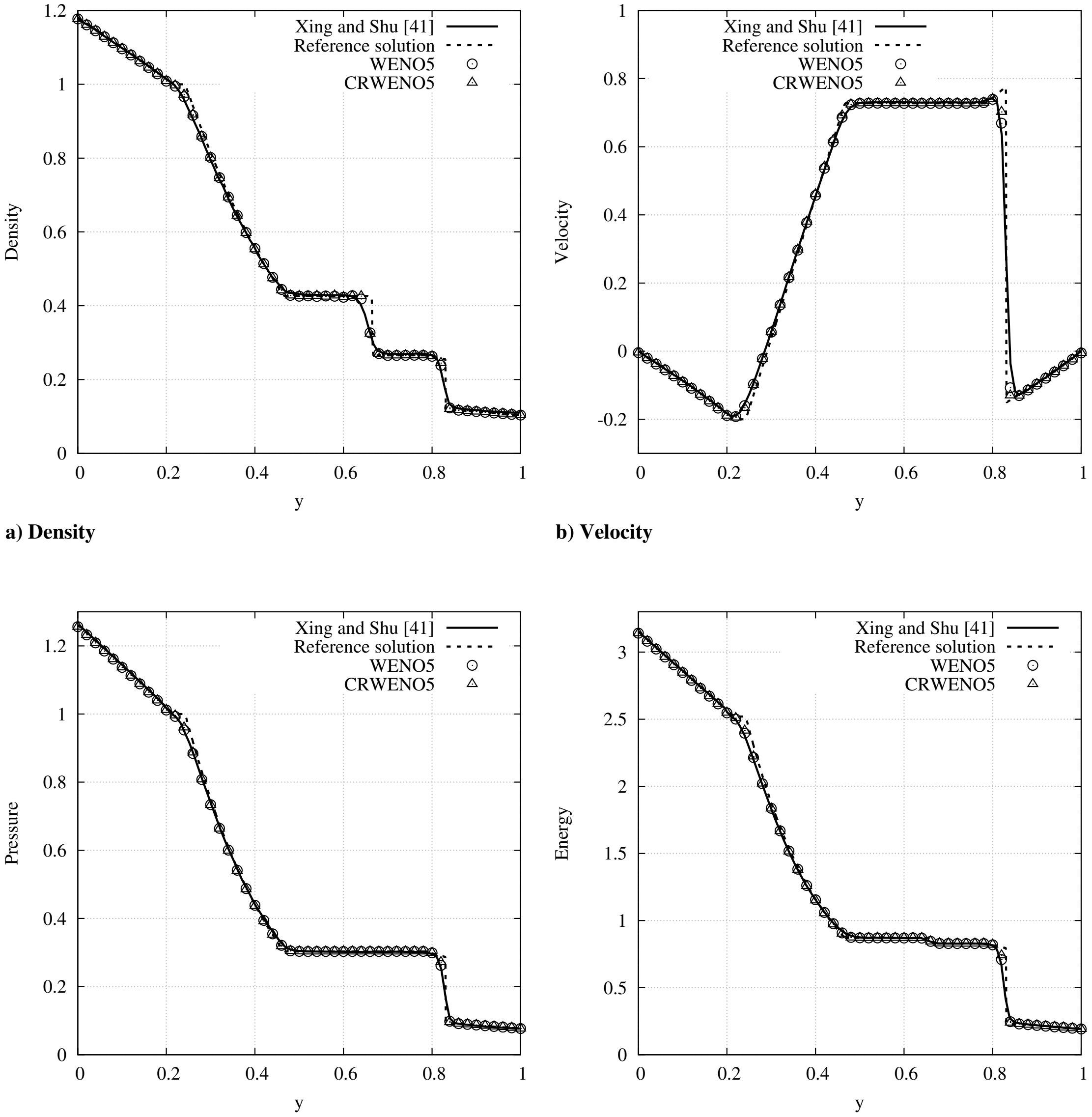
Table 1 Relative error with respect to the initial solution

		WENO5			CRWENO5		
Case	Algorithm	$\|\boldsymbol{\epsilon}\|_1$	$\|\boldsymbol{\epsilon}\|_2$	$\|\boldsymbol{\epsilon}\|_\infty$	$\|\boldsymbol{\epsilon}\|_1$	$\|\boldsymbol{\epsilon}\|_2$	$\|\boldsymbol{\epsilon}\|_\infty$
1	Balanced	6.02E − 15	7.11E − 15	1.31E − 14	1.50E − 14	1.53E − 14	2.09E − 14
1	Naive	3.75E − 02	4.39E − 02	7.34E − 02	2.86E − 02	3.37E − 02	6.17E − 02
2	Balanced	3.63E − 15	4.35E − 15	8.15E − 15	1.58E − 14	1.83E − 14	6.11E − 14
2	Naive	1.62E − 02	1.60E − 02	1.72E − 02	1.89E − 02	1.86E − 02	1.99E − 02



a) $\eta = 10^{-2}$ **b)** $\eta = 10^{-4}$

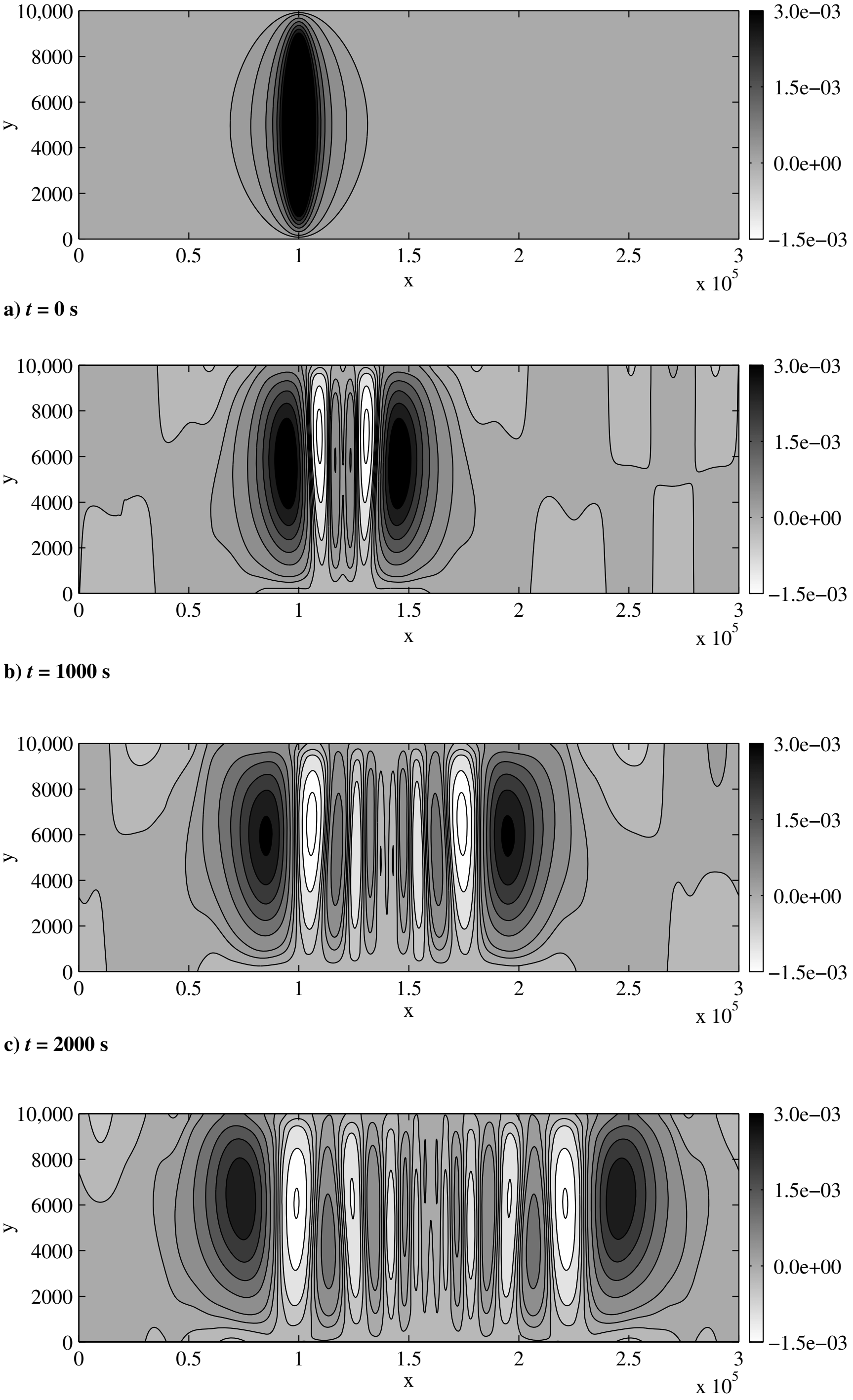
Fig. 3 Pressure perturbation at final time 0.25 on a grid with 200 points (every fourth grid point is shown).



a) Density **b) Velocity**

c) Pressure **d) Energy**

Fig. 4 Solution to the modified Sod's shock tube problem obtained on a grid with 101 points (every second point is shown).



a) *t* = 0 s

b) *t* = 1000 s

c) *t* = 2000 s

d) *t* = 3000 s

Fig. 5 Inertia-gravity waves: potential temperature perturbation contours for the solution obtained with the CRWENO5 scheme on a grid with 1200 × 50 points.

B. Sod's Shock Tube with Gravitational Forcing

Sod's shock tube test [55] is a benchmark one-dimensional Riemann problem. A modified test case with gravitational forcing [41] is solved. The initial solution is given by

$$(\rho, v, p) = \begin{cases} (1, 0, 1) & y < 0.5 \\ (0.125, 0, 0.1) & y \geq 0.5 \end{cases} \tag{62}$$

on a unit domain $y \in [0, 1]$ discretized by a grid with 101 points. The specific heat ratio is $\gamma = 1.4$. Reflective boundary conditions are applied at both ends of the domain. The flow is subjected to a gravitational field $g = 1$. Solutions are obtained at a final time of 0.2 with the SSPRK3 method and a time step of 0.002 (corresponding to a CFL of ∼0.4). The Roe upwinding scheme is used. Figure 4 shows the solutions obtained with the WENO5 and CRWENO5 schemes; the reference solution is obtained with the CRWENO5 scheme on a grid with 2001 points. The computed solutions agree well with the reference solution and results in the literature [41].

C. Inertia-Gravity Waves

The inertia-gravity wave [9,56] is a two-dimensional benchmark for atmospheric models that involves the evolution of a potential

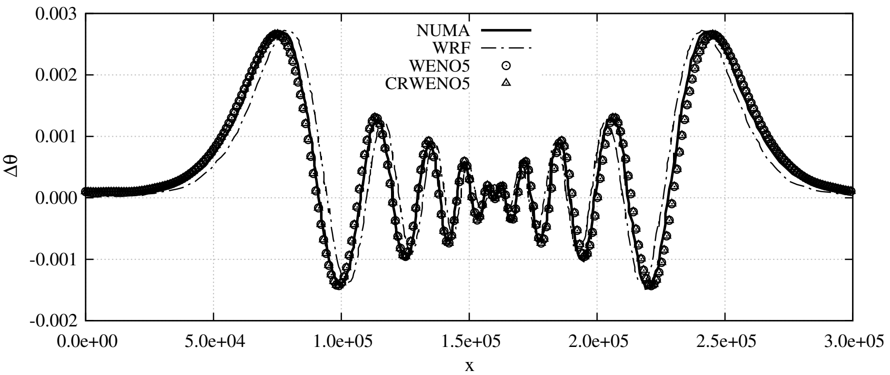


Fig. 6 Inertia-gravity waves: cross-sectional potential temperature for the solution obtained at 3000 s on a grid with 1200 × 50 points.

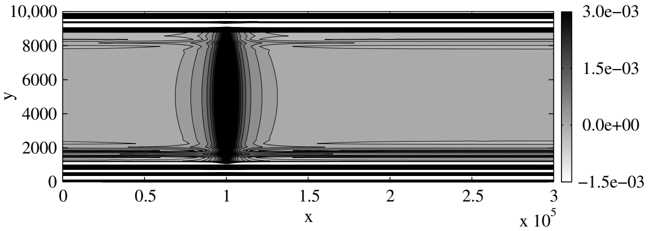


Fig. 7 Inertia-gravity waves: potential temperature perturbation contours with a naive treatment of the source term at final time 1 s on a grid with 1200 × 50 points.

temperature perturbation. The domain consists of a channel with dimensions 300,000 m × 10,000 m. Periodic boundary conditions are applied on the left ($x = 0$ m) and right ($x = 300{,}000$ m) boundaries, whereas inviscid wall boundary conditions are applied on the bottom ($y = 0$ m) and top ($y = 10{,}000$ m) boundaries. The initial flow is a perturbation added to a stratified atmosphere in hydrostatic balance (example 3 in the Appendix). The Brunt–Väisälä frequency is specified as $\mathcal{N} = 0.0\ 1/\mathrm{s}$, the gravitational force per unit mass is $9.8\ \mathrm{m/s^2}$, and the horizontal flow velocity is $u = 20\ \mathrm{m/s}$ throughout the domain. The reference pressure and temperature at $y = 0$ m are $10^5\ \mathrm{N/m^2}$ and 300 K, respectively. The perturbation is added to the potential temperature, specified as

$$\Delta\theta(x, y, t = 0) = \theta_c \sin\left(\frac{\pi_c y}{h_c}\right)\left[1 + \left(\frac{x - x_c}{a_c}\right)^2\right]^{-1} \tag{63}$$

where $\theta_c = 0.01$ K is the perturbation strength, $h_c = 10{,}000$ m is the height of the domain, $a_c = 5000$ m is the perturbation half-width, $x_c = 100{,}000$ m is the horizontal location of the perturbation, and $\pi_c \approx 3.141592654$ is the Archimedes (trigonometric) constant. The evolution of the perturbation is simulated until a final time of 3000 s.

Solutions are obtained with the WENO5 and CRWENO5 schemes on a grid with 1200 × 50 points that results in a resolution of 250 m in x and 200 m in y. The SSPRK3 method is used for time integration with a time step of 0.25 s (corresponding to a CFL of ~0.4). The Rusanov scheme is used for upwinding. Figure 5 shows the potential temperature perturbation $\Delta\theta$ for the initial, intermediate, and final solutions. The results agree well with those in the literature [17,20,24,56]. The initial perturbation is centered at $x = 10{,}000$ m, whereas the flow features in the final solution are centered at $x = 160{,}000$ m; this translation is expected because of the mean horizontal velocity of $20\ \mathrm{m/s}$.

Figure 6 shows the cross-sectional potential temperature perturbation at an altitude of $y = 5000$ m. The solutions obtained with the WENO5 and CRWENO5 schemes are compared with two reference solutions: NUMA and WRF. NUMA [9] refers to a spectral-element solver, and the solution is obtained with 10th-order polynomials and 250 m effective grid resolution. WRF [16] uses a finite difference discretization, and the solution is obtained with a fifth-order upwind scheme in x and a third-order upwind scheme in y [17]. Good agreement is observed with NUMA, whereas there is a slight difference in the perturbation propagation speed with WRF. Figure 7 shows the potential temperature contours of the solution after four time steps (final time of 1 s), where a naive treatment of the source term is used (the cell-centered values of the density and velocity are used to compute it). The horizontal contours near the top and the bottom boundaries are the errors resulting from the hydrostatic imbalance, and they overwhelm the solution. This result demonstrates the need for the well-balanced formulation.

D. Rising Thermal Bubble

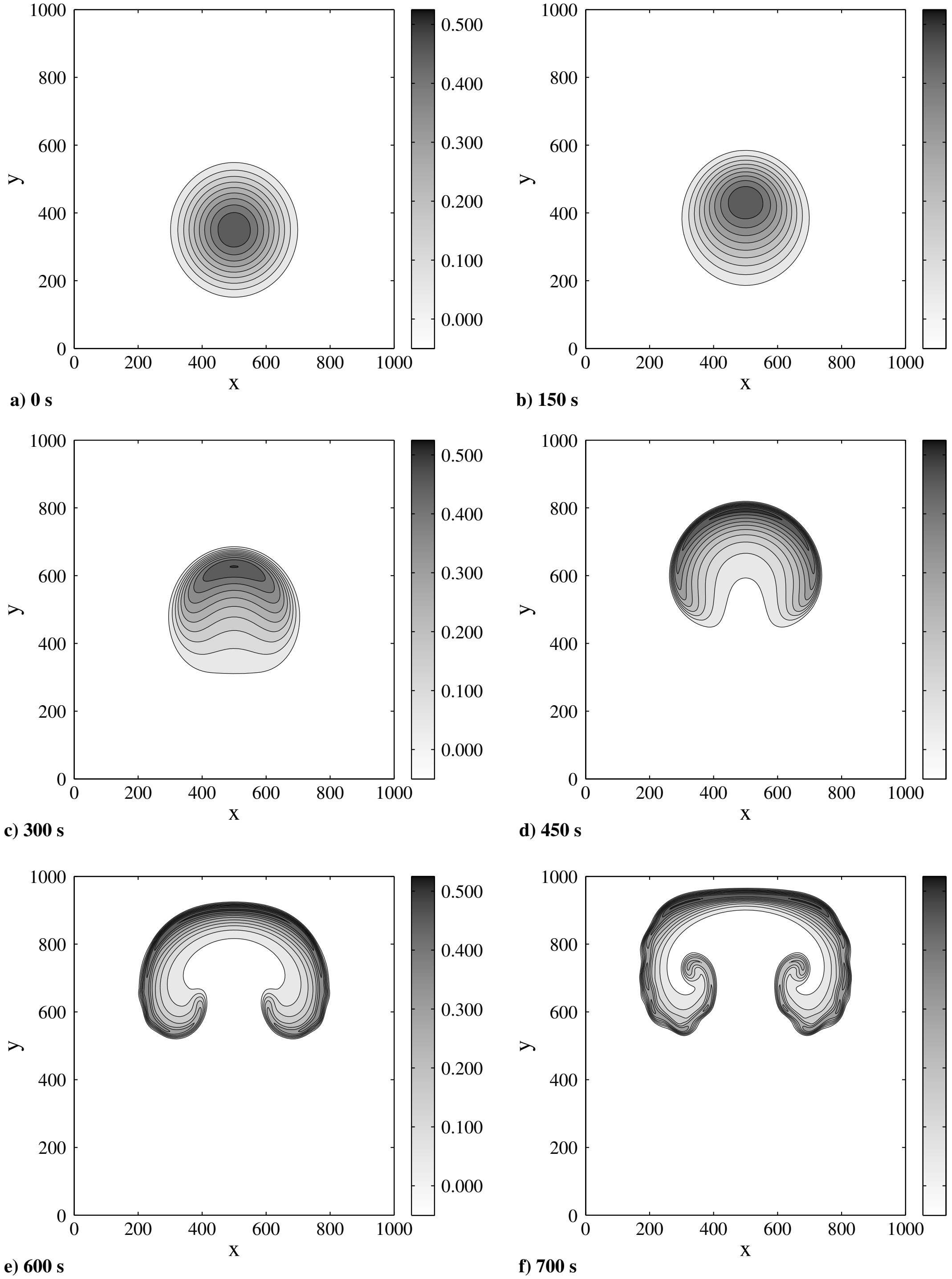
The two-dimensional rising thermal bubble [9] is another benchmark for atmospheric flows that simulates the dynamics of a warm bubble. The square domain of dimensions 1000 m × 1000 m is specified with inviscid wall boundary conditions on all sides. The initial solution is a stratified atmosphere in hydrostatic balance corresponding to example 2 in the Appendix. The constant potential temperature (and thus the reference temperature at $y = 0$ m) is 300 K, and the reference pressure is $10^5\ \mathrm{N/m^2}$. The ambient flow is at rest and experiences a constant gravitational force per unit mass of $9.8\ \mathrm{m/s^2}$. The warm bubble is added as a potential temperature perturbation specified as

$$\Delta\theta(x, y, t = 0) = \begin{cases} 0 & r > r_c \\ \frac{\theta_c}{2}\left[1 + \cos\left(\frac{\pi_c r}{r_c}\right)\right] & r \le r_c \end{cases}, \qquad r = \sqrt{(x - x_c)^2 + (z - z_c)^2} \tag{64}$$

where $\theta_c = 0.5$ K is the perturbation strength, $(x_c, y_c) = (500, 350)$ m is the initial location at which the bubble is centered, $r_c = 250$ m is the radius of the bubble, and π_c is the trigonometric constant. The flow is simulated to a final time of 700 s.

Figure 8 shows the solution obtained on a grid with 401^2 points, corresponding to a resolution of 2.5 m. The Rusanov scheme is used for upwinding. The RK4 method is used for time integration with a time step of 0.005 s (corresponding to a maximum CFL of ~0.7). The potential temperature perturbation $\Delta\theta$ is shown at 0 (initial bubble), 250, 500, and 700 s. The warm bubble rises as a result of buoyancy. The temperature differential within the bubble causes velocity gradients that shear and deform the bubble to a mushroomlike cloud. At the top boundary, the deformed bubble interacts with the inviscid wall to form a thin layer of warm air, whereas the trailing edges roll up due to the local temperature difference. The solution agrees with the inviscid results reported in the literature [8,9].

The solutions obtained by the proposed algorithm are compared with those obtained with the two-dimensional version of NUMA [9] with the same order of accuracy and grid resolution. NUMA solves



a) 0 s

b) 150 s

c) 300 s

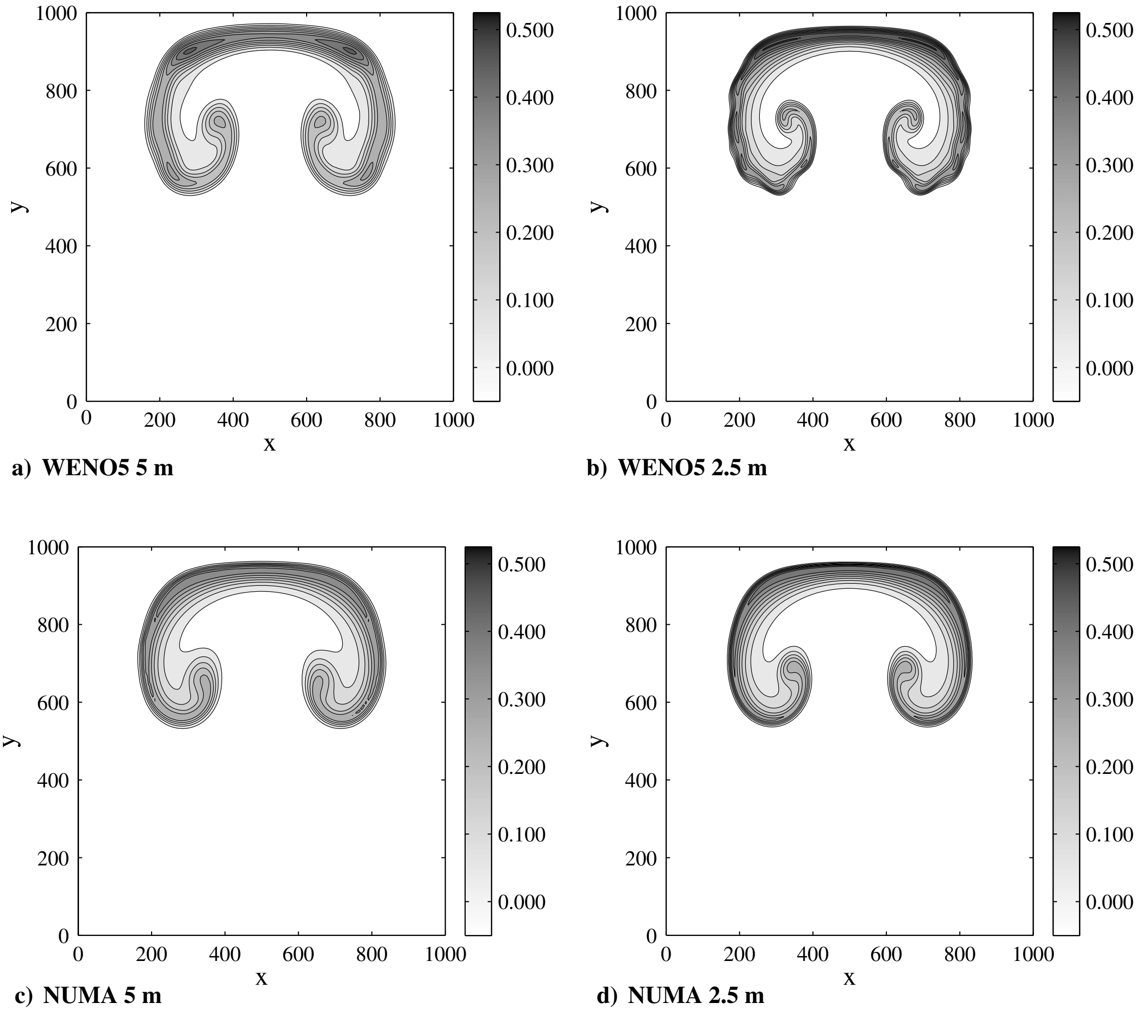
d) 450 s

e) 600 s

f) 700 s

Fig. 8 Rising thermal bubble: potential temperature perturbation $\Delta\theta$ contours for the solution obtained with the WENO5 scheme on a grid with 401×401 points (2.5 m resolution).

the governing equations in terms of mass, momentum, and potential temperature discretized by the continuous and discontinuous Galerkin methods with spectral elements. Figures 9a and 9b show the final solution at 700 s obtained with the WENO5 scheme on grids with 201^2 and 401^2 points (5 and 2.5 m grid resolutions), respectively. Figures 9c and 9d show the solutions obtained with NUMA using the continuous Galerkin discretization. The domain is discretized with 40^2 and 80^2 elements with fifth-order polynomials, respectively, resulting in effective grid resolutions of 5 and 2.5 m. These solutions are obtained at a CFL of ~0.7 with the RK4 time-integration method. Good agreement is observed for the overall flow at both grid resolutions. Figure 10 shows the cross-sectional potential temperature perturbation for the solutions shown in Fig. 9 along y ($x = 500$ m) and x ($y = 720$ m). Although the solutions obtained with NUMA exhibit smooth flow features, the WENO5 solutions predict stronger gradients that result in qualitatively different flow features at small length scales. As an example, WENO5 predicts a stronger roll-up of the trailing edges at 2.5 m grid resolution at $600 \leq 700 \leq 720$ m, as observed in Fig. 10b. This difference is due to the treatment of subgrid-length scales. The NUMA solver stabilizes the solution through a residual-based dynamic subgrid-scale model [57,58] designed for large-eddy simulation, whereas the WENO5 solution to the inviscid Euler equations relies on the linear and nonlinear numerical diffusion to stabilize the unresolved scales. Incorporation of a subgrid-scale model in the current solver will be explored in the future.



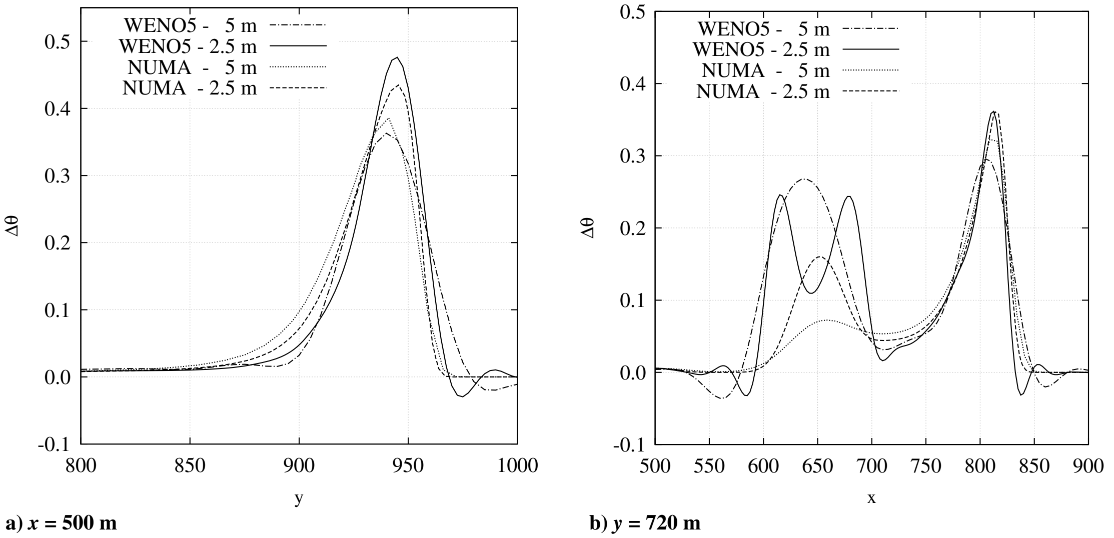
a) WENO5 5 m

b) WENO5 2.5 m

c) NUMA 5 m

d) NUMA 2.5 m

Fig. 9 Rising thermal bubble: comparison of potential temperature perturbation $\Delta\theta$ contours at 700 s for the solution obtained with the WENO5 scheme and NUMA [9] for two grid resolutions.



a) x = 500 m

b) y = 720 m

Fig. 10 Rising thermal bubble: cross-sectional potential temperature perturbation $\Delta\theta$ for final time 700 s.

V. Conclusions

A well-balanced conservative finite difference algorithm is proposed in this paper for the numerical simulation of atmospheric flows. The governing equations (inviscid Euler equations) are solved as conservation laws for mass, momentum, and energy, with no additional assumptions; thus, they are of the same form as those that are used in the computational aerodynamics community. The discretization of the hyperbolic flux and the treatment of the source term are modified so that the overall algorithm preserves hydrostatically balanced equilibria to machine precision; this formulation is an extension of the previous work by Xing and Shu [41]. In the absence of gravitational forces, the algorithm reduces to the standard finite difference discretization of the Euler equations. The fifth-order WENO and the CRWENO schemes are used in this paper for spatial discretization; however, the well-balanced formulation can be used with any conservative finite difference methods. The ability of the proposed algorithm to preserve a hydrostatically balanced equilibrium is demonstrated on examples of stratified atmosphere. We show that the balanced formulation preserves such steady states to machine precision, whereas a naive treatment of the source term does not. Benchmark flow problems are solved to verify the algorithm. The solutions are compared with those obtained with operational, state-of-the-art atmospheric flow solvers (WRF and NUMA), and good agreement is observed. Future work will focus on incorporating semi-implicit and implicit time-integration methods as well as a subgrid-scale model similar to the models used in established atmospheric flow solvers.

Appendix A: Examples of Hydrostatically Balanced Equilibrium States

Three examples of the well-balanced finite difference formulation are presented. These examples are representative of the hydrostatic equilibrium encountered in benchmark atmospheric flow problems.

AI. Example 1

The first example is the isothermal steady state. The resulting formulation is identical to the previously proposed well-balanced WENO scheme [41]. The steady state can be derived by assuming the temperature $T = T_0$ to be a constant and applying the hydrostatic balance Eq. (38). It is given by

$$\begin{aligned} &u = \text{constant}, \qquad v = 0, \\ &\rho = \rho_0 \exp\left(-\frac{gy}{RT}\right), \qquad p = p_0 \exp\left(-\frac{gy}{RT}\right) \end{aligned} \tag{A1}$$

where the reference density ρ_0 and pressure p_0 are related by the equation of state. Thus, the functions $\varrho(y)$ and $\varphi(y)$ in Eq. (37) are

$$\varrho(y) = \exp\left(-\frac{gy}{RT}\right), \qquad \varphi(y) = \exp\left(-\frac{gy}{RT}\right) \tag{A2}$$

The modified source term in Eq. (42) and the modified solution in Eqs. (54) and (55) are

$$\boldsymbol{s}^* = \begin{bmatrix} 0 \\ 0 \\ \rho R T_0 \exp\left(\frac{gy}{RT}\right)\left\{\exp\left(-\frac{gy}{RT}\right)\right\}_y \\ \rho v R T_0 \exp\left(\frac{gy}{RT}\right)\left\{\exp\left(-\frac{gy}{RT}\right)\right\}_y \end{bmatrix}, \quad \text{and}$$

$$\boldsymbol{q}^* = \begin{bmatrix} \rho \exp\left(\frac{gy}{RT}\right) \\ \rho u \exp\left(\frac{gy}{RT}\right) \\ \rho v \exp\left(\frac{gy}{RT}\right) \\ e \exp\left(\frac{gy}{RT}\right) \end{bmatrix} \tag{A3}$$

AII. Example 2

The second example is a hydrostatic balance that is frequently encountered in atmospheric flows of practical relevance [9,10,17]. The steady state is derived by specifying a stratified atmosphere with constant potential temperature $\theta = T_0$. The hydrostatic balance is thus expressed as

$$\begin{aligned} \frac{\gamma R}{\gamma - 1}\theta\frac{\mathrm{d}\pi}{\mathrm{d}y} = -g \Rightarrow \rho &= \rho_0\left[1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right]^{1/(\gamma-1)}, \\ p &= p_0\left[1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right]^{\gamma/(\gamma-1)}, \\ u &= \text{constant}, \qquad v = 0 \end{aligned} \tag{A4}$$

where π is the Exner pressure (see Sec. II). Thus, the functions $\varrho(y)$ and $\varphi(y)$ in Eq. (37) are

$$\begin{aligned} \varrho(y) &= \left[1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right]^{1/(\gamma-1)}, \\ \varphi(y) &= \left[1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right]^{\gamma/(\gamma-1)} \end{aligned} \tag{A5}$$

and the modified source term in Eq. (42) as well as the modified solution in Eqs. (54) and (55) are

$$\boldsymbol{s}^* = \begin{bmatrix} 0 \\ 0 \\ \rho R T_0\left\{1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right\}^{-1/(\gamma-1)}\left\{\left(1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right)^{\gamma/(\gamma-1)}\right\}_y \\ \rho v R T_0\left\{1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right\}^{-1/(\gamma-1)}\left\{\left(1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right)^{\gamma/(\gamma-1)}\right\}_y \end{bmatrix} \tag{A6}$$

and

$$\boldsymbol{q}^* = \begin{bmatrix} \rho\left\{1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right\}^{-1/(\gamma-1)} \\ \rho u\left\{1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right\}^{-1/(\gamma-1)} \\ \rho v\left\{1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right\}^{-1/(\gamma-1)} \\ e^* \end{bmatrix} \tag{A7}$$

where

$$\begin{aligned} e^* = &\frac{p}{\gamma - 1}\left[1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right]^{-\gamma/(\gamma-1)} \\ &+ \frac{1}{2}\rho\left[1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right]^{-1/(\gamma-1)}(u^2 + v^2) \end{aligned} \tag{A8}$$

AIII. Example 3

The third example is a stratified atmosphere with a specified Brunt–Väisälä frequency $\mathcal{N}$ [9,56]:

$$\mathcal{N}^2 = g\frac{d}{dy}(\log\theta) \Rightarrow \theta = T_0 \exp\left(\frac{\mathcal{N}^2}{g}y\right) \tag{A9}$$

Assuming hydrostatic balance, the Exner pressure is given by

$$\pi = 1 + \frac{(\gamma-1)g^2}{\gamma R T_0 \mathcal{N}^2}\left[\exp\left(-\frac{\mathcal{N}^2}{g}y\right) - 1\right] \tag{A10}$$

and the steady-state flow variables are

$$p = p_0\left[1 + \frac{(\gamma-1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g}y\right) - 1\right\}\right]^{\gamma/(\gamma-1)} \tag{A11}$$

$$\rho = \rho_0 \exp\left(-\frac{\mathcal{N}^2}{g} y\right)\left[1 + \frac{(\gamma - 1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g} y\right) - 1\right\}\right]^{1/(\gamma-1)} \tag{A12}$$

$$u = \text{constant}, \qquad v = 0 \tag{A13}$$

Thus, the functions $\varrho(y)$ and $\varphi(y)$ in Eq. (37) are

$$\varrho(y) = \exp\left(-\frac{\mathcal{N}^2}{g} y\right)\left[1 + \frac{(\gamma - 1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g} y\right) - 1\right\}\right]^{1/(\gamma-1)} \tag{A14}$$

$$\varphi(y) = \left[1 + \frac{(\gamma - 1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g} y\right) - 1\right\}\right]^{\gamma/(\gamma-1)} \tag{A15}$$

The modified source term in Eq. (42) and the modified solution in Eqs. (54,55) are

$$\boldsymbol{s}^* = \begin{bmatrix} 0 \\ 0 \\ \rho R T_0 \exp\left(\frac{\mathcal{N}^2}{g} y\right)\left[1 + \frac{(\gamma-1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g} y\right) - 1\right\}\right]^{-1/(\gamma-1)}\left\{\left(1 + \frac{(\gamma-1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g} y\right) - 1\right\}\right)^{\gamma/(\gamma-1)}\right\}_y \\ \rho v R T_0 \exp\left(\frac{\mathcal{N}^2}{g} y\right)\left[1 + \frac{(\gamma-1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g} y\right) - 1\right\}\right]^{-1/(\gamma-1)}\left\{\left(1 + \frac{(\gamma-1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g} y\right) - 1\right\}\right)^{\gamma/(\gamma-1)}\right\}_y \end{bmatrix} \tag{A16}$$

and

$$\boldsymbol{q}^* = \begin{bmatrix} \rho \exp\left(\frac{\mathcal{N}^2}{g} y\right)\left[1 + \frac{(\gamma-1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g} y\right) - 1\right\}\right]^{-1/(\gamma-1)} \\ \rho u \exp\left(\frac{\mathcal{N}^2}{g} y\right)\left[1 + \frac{(\gamma-1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g} y\right) - 1\right\}\right]^{-1/(\gamma-1)} \\ \rho v \exp\left(\frac{\mathcal{N}^2}{g} y\right)\left[1 + \frac{(\gamma-1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g} y\right) - 1\right\}\right]^{-1/(\gamma-1)} \\ e^* \end{bmatrix} \tag{A17}$$

where

$$e^* = \frac{p}{\gamma - 1}\left[1 + \frac{(\gamma - 1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g} y\right) - 1\right\}\right]^{-\gamma/(\gamma-1)} \tag{A18}$$

$$+ \frac{1}{2}\rho \exp\left(\frac{\mathcal{N}^2}{g} y\right)\left[1 + \frac{(\gamma - 1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g} y\right) - 1\right\}\right]^{-1/(\gamma-1)} \times (u^2 + v^2) \tag{A19}$$

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, under contract DE-AC02-06CH11357. We thank Francis X. Giraldo (Naval Postgraduate School) for giving us access to the Nonhydrostatic Unified Model of the Atmosphere code, helping us with generating the benchmark solutions, and providing valuable feedback on our work. We also thank Simone Marras (Stanford University) for his comments and insights on this paper.

References

[1] Taylor, M. A., Edwards, J., and Cyr, A. S., "Petascale Atmospheric Models for the Community Climate System Model: New Developments and Evaluation of Scalable Dynamical Cores," *Journal of Physics: Conference Series*, Vol. 125, No. 1, 2008, Paper 012023. doi:10.1088/1742-6596/125/1/012023

[2] Lin, S.-J., "A 'Vertically Lagrangian' Finite-Volume Dynamical Core for Global Models," *Monthly Weather Review*, Vol. 132, No. 10, 2004, pp. 2293–2307. doi:10.1175/1520-0493(2004)132<2293:AVLFDC>2.0.CO;2

[3] Ogura, Y., and Phillips, N., "Scale Analysis of Deep and Shallow Convection in the Atmosphere," *Journal of the Atmospheric Sciences*, Vol. 19, No. 2, 1962, pp. 173–179. doi:10.1175/1520-0469(1962)019<0173:SAODAS>2.0.CO;2

[4] Durran, D., "Improving the Anelastic Approximation," *Journal of the Atmospheric Sciences*, Vol. 46, No. 11, 1989, pp. 1453–1461. doi:10.1175/1520-0469(1989)046<1453:ITAA>2.0.CO;2

[5] Arakawa, A., and Konor, C., "Unification of the Anelastic and Quasi-Hydrostatic Systems of Equations," *Monthly Weather Review*, Vol. 137, No. 2, 2009, pp. 710–726. doi:10.1175/2008MWR2520.1

[6] Davies, T., Staniforth, A., Wood, N., and Thuburn, J., "Validity of Anelastic and Other Equation Sets as Inferred from Normal-Mode Analysis," *Quarterly Journal of the Royal Meteorological Society*, Vol. 129, No. 593, 2003, pp. 2761–2775. doi:10.1256/qj.02.1951

[7] Klein, R., Achatz, U., Bresch, D., Knio, O., and Smolarkiewicz, P., "Regime of Validity of Soundproof Atmospheric Flow Models," *Journal of the Atmospheric Sciences*, Vol. 67, No. 10, 2010, pp. 3226–3237. doi:10.1175/2010JAS3490.1

[8] Ullrich, P., and Jablonowski, C., "Operator-Split Runge–Kutta–Rosenbrock Methods for Nonhydrostatic Atmospheric Models," *Monthly Weather Review*, Vol. 140, No. 4, 2012, pp. 1257–1284. doi:10.1175/MWR-D-10-05073.1

[9] Giraldo, F. X., and Restelli, M., "A Study of Spectral Element and Discontinuous Galerkin Methods for the Navier–Stokes Equations in Nonhydrostatic Mesoscale Atmospheric Modeling: Equation Sets and Test Cases," *Journal of Computational Physics*, Vol. 227, No. 8, 2008, pp. 3849–3877. doi:10.1016/j.jcp.2007.12.009

[10] Giraldo, F. X., Restelli, M., and Läuter, M., "Semi-Implicit Formulations of the Navier–Stokes Equations: Application to Nonhydrostatic Atmospheric Modeling," *SIAM Journal on Scientific Computing*, Vol. 32, No. 6, 2010, pp. 3394–3425. doi:10.1137/090775889

[11] Grell, G. A., Dudhia, J., and Stauffer, D., "A Description of the Fifth-Generation Penn State/NCAR Mesoscale Model (MM5)," National Center for Atmospheric Research, NCAR TN-398, Boulder, CO, 1994. doi:10.5065/D60Z716B

[12] Hodur, R., "The Naval Research Laboratory's Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS)," *Monthly Weather Review*, Vol. 125, No. 7, 1997, pp. 1414–1430. doi:10.1175/1520-0493(1997)125<1414:TNRLSC>2.0.CO;2

[13] Xue, M., Droegemeier, K. K., and Wong, V., "The Advanced Regional Prediction System (ARPS)—A Multi-Scale Nonhydrostatic Atmospheric Simulation and Prediction Model. Part 1: Model Dynamics and Verification," *Meteorology and Atmospheric Physics*, Vol. 75, Nos. 3–4, 2000, pp. 161–193. doi:10.1007/s007030070003

[14] Janjic, Z., "A Nonhydrostatic Model Based on a New Approach," *Meteorology and Atmospheric Physics*, Vol. 82, Nos. 1–4, 2003,

pp. 271–285.
doi:10.1007/s00703-001-0587-6

[15] Gassmann, A., "An Improved Two-Time-Level Split-Explicit Integration Scheme for Non-Hydrostatic Compressible Models," *Meteorology and Atmospheric Physics*, Vol. 88, Nos. 1–2, 2005, pp. 23–38.
doi:10.1007/s00703-003-0053-8

[16] Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Barker, D. M., Wang, W., and Powers, J. G., "A Description of the Advanced Research WRF Version 2," National Center for Atmospheric Research NCAR/TN-468, Boulder, Colorado, 2005.

[17] Ahmad, N., and Lindeman, J., "Euler Solutions Using Flux-Based Wave Decomposition," *International Journal for Numerical Methods in Fluids*, Vol. 54, No. 1, 2007, pp. 47–72.
doi:10.1002/(ISSN)1097-0363

[18] Giraldo, F. X., Kelly, J. F., and Constantinescu, E., "Implicit-Explicit Formulations of a Three-Dimensional Nonhydrostatic Unified Model of the Atmosphere (NUMA)," *SIAM Journal on Scientific Computing*, Vol. 35, No. 5, 2013, pp. B1162–B1194.
doi:10.1137/120876034

[19] Kelly, J. F., and Giraldo, F. X., "Continuous and Discontinuous Galerkin Methods for a Scalable Three-Dimensional Nonhydrostatic Atmospheric Model: Limited-Area Mode," *Journal of Computational Physics*, Vol. 231, No. 24, 2012, pp. 7988–8008.
doi:10.1016/j.jcp.2012.04.042

[20] Yang, C., and Cai, X., "A Scalable Fully Implicit Compressible Euler Solver for Mesoscale Nonhydrostatic Simulation of Atmospheric Flows," *SIAM Journal on Scientific Computing*, Vol. 36, No. 5, 2014, pp. S23–S47.
doi:10.1137/130919167

[21] Das, P., "A Non-Archimedean Approach to the Equations of Convection Dynamics," *Journal of the Atmospheric Sciences*, Vol. 36, No. 11, 1979, pp. 2183–2190.
doi:10.1175/1520-0469(1979)036<2183:ANAATT>2.0.CO;2

[22] Botta, N., Klein, R., Langenberg, S., and Lützenkirchen, S., "Well Balanced Finite Volume Methods for Nearly Hydrostatic Flows," *Journal of Computational Physics*, Vol. 196, No. 2, 2004, pp. 539–565.
doi:10.1016/j.jcp.2003.11.008

[23] Ahmad, N., Bacon, D., Sarma, A., Koračin, D., Vellore, R., Boybeyi, Z., and Lindeman, J., "Simulations of Non-Hydrostatic Atmosphere Using Conservation Laws Package," *45th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2007-0084, Jan. 2007,
doi:10.2514/6.2007-84

[24] Ahmad, N., and Proctor, F., "The High-Resolution Wave-Propagation Method Applied to Meso- and Micro-Scale Flows," *50th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2012-0430, Jan. 2012,
doi: 10.2514/6.2012-430

[25] Laney, C. B., *Computational Gasdynamics*, Cambridge Univ. Press, New York, 1998, pp. 5–20.

[26] Hirsch, C., *Numerical Computation of Internal and External Flows*, Vols. 1–2, The Fundamentals of Computational Fluid Dynamics, Elsevier Science, New York, 2007, pp. 597–602.

[27] Davies, T., Cullen, M. J. P., Malcolm, A. J., Mawson, M. H., Staniforth, A., White, A. A., and Wood, N., "A New Dynamical Core for the Met Office's Global and Regional Modelling of the Atmosphere," *Quarterly Journal of the Royal Meteorological Society*, Vol. 131, No. 608, 2005, pp. 1759–1782.
doi:10.1256/qj.04.101

[28] Liu, X.-D., Osher, S., and Chan, T., "Weighted Essentially Non-Oscillatory Schemes," *Journal of Computational Physics*, Vol. 115, No. 1, 1994, pp. 200–212.
doi:10.1006/jcph.1994.1187

[29] Jiang, G.-S., and Shu, C.-W., "Efficient Implementation of Weighted ENO Schemes," *Journal of Computational Physics*, Vol. 126, No. 1, 1996, pp. 202–228.
doi:10.1006/jcph.1996.0130

[30] Shu, C.-W., "High Order Weighted Essentially Nonoscillatory Schemes for Convection Dominated Problems," *SIAM Review*, Vol. 51, No. 1, 2009, pp. 82–126.
doi:10.1137/070679065

[31] Ghosh, D., and Baeder, J. D., "Compact Reconstruction Schemes with Weighted ENO Limiting for Hyperbolic Conservation Laws," *SIAM Journal on Scientific Computing*, Vol. 34, No. 3, 2012, pp. 1678–1706.
doi:10.1137/110857659

[32] Lele, S. K., "Compact Finite Difference Schemes with Spectral-Like Resolution," *Journal of Computational Physics*, Vol. 103, No. 1, 1992, pp. 16–42.
doi:10.1016/0021-9991(92)90324-R

[33] Ghosh, D., and Baeder, J. D., "Weighted Non-Linear Compact Schemes for the Direct Numerical Simulation of Compressible, Turbulent Flows," *Journal of Scientific Computing*, Vol. 61, No. 1, 2014, pp. 61–89.
doi:10.1007/s10915-014-9818-0

[34] Ghosh, D., Medida, S., and Baeder, J. D., "Application of Compact-Reconstruction Weighted Essentially Nonoscillatory Schemes to Compressible Aerodynamic Flows," *AIAA Journal*, Vol. 52, No. 9, 2014, pp. 1858–1870.
doi:10.2514/1.J052654

[35] Ghosh, D., Constantinescu, E. M., and Brown, J., "Efficient Implementation of Nonlinear Compact Schemes on Massively Parallel Platforms," *SIAM Journal on Scientific Computing*, Vol. 37, No. 3, 2015, pp. C354–C383.
doi:10.1137/140989261

[36] LeVeque, R. J., "Balancing Source Terms and Flux Gradients in High-Resolution Godunov Methods: The Quasi-Steady Wave-Propagation Algorithm," *Journal of Computational Physics*, Vol. 146, No. 1, 1998, pp. 346–365.
doi:10.1006/jcph.1998.6058

[37] Bale, D., LeVeque, R., Mitran, S., and Rossmanith, J., "A Wave Propagation Method for Conservation Laws and Balance Laws with Spatially Varying Flux Functions," *SIAM Journal on Scientific Computing*, Vol. 24, No. 3, 2003, pp. 955–978.
doi:10.1137/S106482750139738X

[38] Xing, Y., and Shu, C.-W., "High Order Finite Difference WENO Schemes with the Exact Conservation Property for the Shallow Water Equations," *Journal of Computational Physics*, Vol. 208, No. 1, 2005, pp. 206–227.
doi:10.1016/j.jcp.2005.02.006

[39] Xing, Y., and Shu, C.-W., "High-Order Well-Balanced Finite Difference WENO Schemes for a Class of Hyperbolic Systems with Source Terms," *Journal of Scientific Computing*, Vol. 27, Nos. 1–3, 2006, pp. 477–494.
doi:10.1007/s10915-005-9027-y

[40] Xing, Y., and Shu, C.-W., "High Order Well-Balanced Finite Volume WENO Schemes and Discontinuous Galerkin Methods for a Class of Hyperbolic Systems with Source Terms," *Journal of Computational Physics*, Vol. 214, No. 2, 2006, pp. 567–598.
doi:10.1016/j.jcp.2005.10.005

[41] Xing, Y., and Shu, C.-W., "High Order Well-Balanced WENO Scheme for the Gas Dynamics Equations Under Gravitational Fields," *Journal of Scientific Computing*, Vol. 54, Nos. 2–3, 2013, pp. 645–662.
doi:10.1007/s10915-012-9585-8

[42] Shu, C.-W., and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes," *Journal of Computational Physics*, Vol. 77, No. 2, 1988, pp. 439–471.
doi:10.1016/0021-9991(88)90177-5

[43] Shu, C.-W., and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes, 2," *Journal of Computational Physics*, Vol. 83, No. 1, 1989, pp. 32–78.
doi:10.1016/0021-9991(89)90222-2

[44] Butcher, J., *Numerical Methods for Ordinary Differential Equations*, Wiley, New York, 2003, pp. 94–95.

[45] Gottlieb, S., Ketcheson, D. I., and Shu, C.-W., "High Order Strong Stability Preserving Time Discretizations," *Journal of Scientific Computing*, Vol. 38, No. 3, 2009, pp. 251–289.
doi:10.1007/s10915-008-9239-z

[46] Yamaleev, N. K., and Carpenter, M. H., "A Systematic Methodology for Constructing High-Order Energy Stable WENO Schemes," *Journal of Computational Physics*, Vol. 228, No. 11, 2009, pp. 4248–4272.
doi:10.1016/j.jcp.2009.03.002

[47] Henrick, A. K., Aslam, T. D., and Powers, J. M., "Mapped Weighted Essentially Non-Oscillatory Schemes: Achieving Optimal Order near Critical Points," *Journal of Computational Physics*, Vol. 207, No. 2, 2005, pp. 542–567.
doi:10.1016/j.jcp.2005.01.023

[48] Borges, R., Carmona, M., Costa, B., and Don, W. S., "An Improved Weighted Essentially Non-Oscillatory Scheme for Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 227, No. 6, 2008, pp. 3191–3211.
doi:10.1016/j.jcp.2007.11.038

[49] Castro, M., Costa, B., and Don, W. S., "High Order Weighted Essentially Non-Oscillatory WENO-Z Schemes for Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 230, No. 5, 2011, pp. 1766–1792.
doi:10.1016/j.jcp.2010.11.028

[50] Ghosh, D., Constantinescu, E. M., and Brown, J., "Scalable Nonlinear Compact Schemes," Argonne National Lab. TR-ANL/MCS-TM-340, Argonne, IL, 2014.

[51] Roe, P. L., “Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes,” *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.
doi:10.1016/0021-9991(81)90128-5

[52] LeVeque, R., Finite Volume Methods for Hyperbolic Problems, *Cambridge Texts in Applied Mathematics*, Cambridge Univ. Press, New York, 2002, pp. 227–251.

[53] Rusanov, V. V., “The Calculation of the Interaction of Non-Stationary Shock Waves and Obstacles,” *USSR Computational Mathematics and Mathematical Physics*, Vol. 1, No. 2, 1962, pp. 304–320.
doi:10.1016/0041-5553(62)90062-9

[54] Slyz, A., and Prendergast, K., “Time-Independent Gravitational Fields in the BGK Scheme for Hydrodynamics,” *Astronomy and Astrophysics Supplement Series*, Vol. 139, No. 1, 1999, pp. 199–217.
doi:10.1051/aas:1999389

[55] Sod, G. A., “A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws,” *Journal of Computational Physics*, Vol. 27, No. 1, 1978, pp. 1–31.
doi:10.1016/0021-9991(78)90023-2

[56] Skamarock, W. C., and Klemp, J. B., “Efficiency and Accuracy of the Klemp–Wilhelmson Time-Splitting Technique,” *Monthly Weather Review*, Vol. 122, No. 11, 1994, pp. 2623–2630.
doi:10.1175/1520-0493(1994)122<2623:EAAOTK>2.0.CO;2

[57] Nazarov, M., and Hoffman, J., “Residual-Based Artificial Viscosity for Simulation of Turbulent Compressible Flow Using Adaptive Finite Element Methods,” *International Journal for Numerical Methods in Fluids*, Vol. 71, No. 3, 2013, pp. 339–357.
doi:10.1002/fld.v71.3

[58] Marras, S., Nazarov, M., and Giraldo, F. X., “Stabilized High-Order Galerkin Methods Based on a Parameter-Free Dynamic SGS Model for LES,” *Journal of Computational Physics*, Vol. 301, No. C, 2015, pp. 77–101.
doi:10.1016/j.jcp.2015.07.034