A Multimoment Constrained Finite-Volume Model for Nonhydrostatic Atmospheric Dynamics

XINGLIANG LI
State Key Laboratory of Severe Weather, Chinese Academy of Meteorological Sciences, and Center of Numerical Weather Prediction, China Meteorological Administration, Beijing, China

CHUNGANG CHEN
School of Human Settlement and Civil Engineering, Xi’an Jiaotong University, Xi’an, Shaanxi, China

XUESHUN SHEN
State Key Laboratory of Severe Weather, Chinese Academy of Meteorological Sciences, and Center of Numerical Weather Prediction, China Meteorological Administration, Beijing, China

FENG XIAO
Department of Energy Sciences, Tokyo Institute of Technology, Yokohama, Japan

(Manuscript received 9 May 2012, in final form 2 September 2012)

ABSTRACT

The two-dimensional nonhydrostatic compressible dynamical core for the atmosphere has been developed by using a new nodal-type high-order conservative method, the so-called multimoment constrained finite-volume (MCV) method. Different from the conventional finite-volume method, the predicted variables (unknowns) in an MCV scheme are the values at the solution points distributed within each mesh cell. The time evolution equations to update the unknown point values are derived from a set of constraint conditions based on the multimoment concept, where the constraint on the volume-integrated average (VIA) for each mesh cell is cast into a flux form and thus guarantees rigorously the numerical conservation. Two important features make the MCV method particularly attractive as an accurate and practical numerical framework for atmospheric and oceanic modeling. 1) The predicted variables are the nodal values at the solution points that can be flexibly located within a mesh cell (equidistant solution points are used in the present model). It is computationally efficient and provides great convenience in dealing with complex geometry and source terms. 2) High-order and physically consistent formulations can be built by choosing proper constraints in view of not only numerical accuracy and efficiency but also underlying physics. In this paper the authors present a dynamical core that uses the third- and the fourth-order MCV schemes. They have verified the numerical outputs of both schemes by widely used standard benchmark tests and obtained competitive results. The present numerical core provides a promising and practical framework for further development of nonhydrostatic compressible atmospheric models.

1. Introduction

The nonhydrostatic effects are found to be very important in the meso- and synoptic-scale atmospheric dynamics and have been included in regional models for several decades. Among others, some examples of regional nonhydrostatic dynamical models are the fifth-generation Pennsylvania State University–National Center for Atmospheric Research (PSU–NCAR) Mesoscale Model (MM5; Dudhia 1993), Environment Canada’s mesoscale compressible model, version 2 (MC2; Benoit et al. 1997), the U.S. Navy Coupled Ocean–Atmosphere Mesoscale Prediction System (Hodur 1997), the German Weather Service’s local model (LM; Doms and Schattler 1999), the University of Oklahoma’s Advanced Regional Prediction System (ARPS; Xue et al. 2000), the NCAR Weather Research and Forecasting...
Model (WRF; Skamarock and Klemp 2008), and the China Meteorological Administration’s Global/Regional Assimilation and Prediction Systems (GRAPES; Chen et al. 2008). It is observed as a noticeable trend that finer grid resolutions are available for global models due to the rapid progress in computer hardware, and mesoscale dynamics can be solved on a global base, which requires the global models to be built in the nonhydrostatic framework as well (Putman and Lin 2007; Satoh et al. 2008; Walko and Avisser 2008; Gassmann 2011; Ullrich and Jablonowski 2012a).

Another requirement for an atmospheric model to be able to efficiently use next-generation large-scale supercomputers, which usually consist of tens to hundreds of thousands of processors with distributed-memory nodes, is that the dynamical core should be highly scalable for large-scale parallel processing. Some recent studies demonstrate that the dynamical cores based on high-order schemes with local spectral reconstructions are superior in parallel scalability and overcome the barrier that prevents the spherical-harmonic spectral-transform-based method and finite-volume method from efficient implementations on supercomputers toward exaflops computation (Dennis et al. 2005, 2012). Researches have been so far carried out in such a direction by using the spectral element (SE) method (Thomas and Loft 2000; Iskandarani et al. 2002; Giraldo and Rosmond 2004; Fournier et al. 2004; Taylor and Fournier 2010) or the discontinuous Galerkin (DG) method (Levy et al. 2007; Giraldo and Restelli 2008; Restelli and Giraldo 2009; Nair et al. 2009; Blaise and St-Cyr 2012). The latter can be then further divided into modal type and nodal type (Hesthaven and Warburton 2008), and the nodal type is getting an increase in popularity because of its computational efficiency. As pointed by these authors, the high-order accuracy, geometric flexibility, and the scalability on massively parallel computer systems make this class of local reconstruction-based methods quite worthy of further exploration as the potential numerical frameworks of new-generation atmospheric and oceanic models. Meanwhile, it is also noted that the existing high-order spectral convergence methods are more computationally expensive due to for example the numerical quadrature in the Galerkin formulation.

Another type of high-order schemes can be devised by using the multimoment concept (Yabe and Aoki 1991; Yabe et al. 2001; Xiao 2004; Xiao et al. 2006; Ii et al. 2005; Xiao and Ii 2007; Ii and Xiao 2007; Chen and Xiao 2008), where we make use of different kinds of discrete quantities, so-called moments in out context, to describe the physical field, such as the pointwise value, derivatives, and volume-integrated value. These moments are locally defined over each mesh cell, which allows us to build high-order local reconstructions. Different moments can be then updated by different formulations that may have different forms but should be consistent with the original conservation law. For example, the point value can be updated by a pointwise Riemann solver or semi-implicit semi-Lagrangian (SISL) solver, while the volume-integrated average value is computed by a finite-volume formulation to ensure the rigorous numerical conservation. In our previous studies, two global multimoment finite-volume shallow-water models have been reported using either a pointwise Riemann solver (Chen and Xiao 2008) or the SISL method (Li et al. 2008). In these schemes, the moments (i.e., cell-integrated average), which point to value and derivative, are directly treated as the prognostic variables.

A more efficient alternative to the multimoment finite-volume formulation is to define the unknowns [or the degrees of freedom (DOFs)] as the values at the points located within each grid cell and to use the time evolution equations of different moments as the constraint conditions to derive the governing equations for updating the unknown point values. The resulting scheme is so-called multimoment constrained finite-volume (MCV) method (Ii and Xiao 2009) where the moments are not directly used as the predicted unknowns (DOFs), but the constraint conditions. The numerical conservation is exactly guaranteed through a constraint on the cell-integrated average, which is cast in a finite-volume formulation of flux form. In the present multimoment constrained method, all predicted unknowns are the nodal values at the solution points and the volume integration is not involved for a conservation law, which makes the numerical formulation very efficient, especially when the physical source term and metric term are included in the governing equations. Our experiences also show that an MCV scheme can use a larger Courant–Friedrichs–Lewy (CFL) number for computational stability compared to other high-order schemes of the same order, and the location of the solution points can be determined in a more flexible manner. We have recently implemented the MCV method to develop global shallow-water models on an icosahedral grid (Ii and Xiao 2010), hexagonal geodesic grid (Chen et al. 2012), and the Yin–Yang grid (Li et al. 2012), and demonstrated its potential as the fundamental numerics for a new type of dynamical cores.

In this paper, we implement for first time the third- and fourth-order MCV method to construct the nonhydrostatic compressible atmospheric dynamical core. The basic numerical formulations are straightforwardly derived by applying the numerical schemes presented in Ii and Xiao (2009) to the dynamic equations of the compressible and nonhydrostatic atmosphere. The remaining paper consists of the following parts. Section 2 describes the 2D compressible governing equations and
their alternative form related to the hyperbolicity. The dynamical core based on the third- and fourth-order MCV schemes is presented in section 3. In section 4, the numerical results of some standard benchmark tests are reported to evaluate the presented model in both accuracy and efficiency. Finally, we end the paper with some conclusions and future goals in section 5.

2. The governing equations for compressible nonhydrostatic dynamical core

We consider a 2D compressible nonhydrostatic atmospheric model and write its governing equations as the Euler conservation laws with the effect of gravity as

\[
\begin{align*}
\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho w)}{\partial z} & = 0, \\
\frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho u^2 + p)}{\partial x} + \frac{\partial (\rho uw)}{\partial z} & = 0, \\
\frac{\partial (\rho w)}{\partial t} + \frac{\partial (\rho wu)}{\partial x} + \frac{\partial (\rho w^2 + p)}{\partial z} & = -\rho g, \\
\frac{\partial (\rho \theta)}{\partial t} + \frac{\partial (\rho u \theta)}{\partial x} + \frac{\partial (\rho w \theta)}{\partial z} & = 0,
\end{align*}
\]

where \( \rho \) is the density, \( u = (u, w)^T \) the vector wind in the Cartesian coordinate, \( p \) is the pressure, and \( \theta \) is the potential temperature. Since the potential temperature is related to the air temperature \( T \) and pressure \( p \) by \( \theta = T(p_0/p)^{\gamma R_d/c_p} \), the equation of state is expressed by \( p = C_0(\rho \theta)^{\gamma} \), which exactly closes the equation set where \( C_0 \) is constant given by \( C_0 = R_d p_0^{-\gamma} \). In the above expressions, the constants are given as \( \gamma = c_p/c_v = 1.4, \ R_d = 287 \text{ J kg}^{-1} \text{ K}^{-1}, \ c_p = 1004.5 \text{ J kg}^{-1} \text{ K}^{-1}, \ c_v = 717.5 \text{ J kg}^{-1} \text{ K}^{-1}, \) and \( p_0 = 10^5 \text{ Pa} \).

a. Splitting of reference state

As commonly applied in atmospheric models, the thermodynamic variables are split into a reference state and deviations. The reference state satisfies the stratification balance [i.e., the hydrostatic relation in the vertical direction \( (z) \)]. The thermodynamic variables are then written as

\[
\begin{align*}
\rho(x, t) & = \bar{\rho}(z) + \rho'(x, t), \\
p(x, t) & = \bar{p}(z) + p'(x, t), \\
(\rho \theta)(x, t) & = \bar{\rho}(z) \theta(z) + (\rho \theta)'(x, t),
\end{align*}
\]

where the reference pressure \( \bar{p}(z) \) and density \( \bar{\rho}(z) \) are in local hydrostatic balance:

\[
\frac{\partial \bar{\rho}}{\partial z} = -\bar{p} g.
\]

After the splitting of the thermodynamic variables, Eqs. (1)–(4) become

\[
\begin{align*}
\frac{\partial \rho'}{\partial t} + \frac{\partial (\rho u')}{\partial x} + \frac{\partial (\rho w')}{\partial z} & = 0, \\
\frac{\partial (\rho u')}{\partial t} + \frac{\partial (\rho u'^2 + p')}{\partial x} + \frac{\partial (\rho u'w')}{\partial z} & = 0, \\
\frac{\partial (\rho w')}{\partial t} + \frac{\partial (\rho w'u')}{\partial x} + \frac{\partial (\rho w'^2 + p')}{\partial z} & = -\rho' g, \\
\frac{\partial (\rho \theta')}{\partial t} + \frac{\partial (\rho u' \theta)}{\partial x} + \frac{\partial (\rho w' \theta)}{\partial z} & = 0,
\end{align*}
\]

where \( p' = \epsilon_0(\rho \theta)' \) and \( \epsilon_0 = \gamma C_0(\bar{\rho} \bar{\theta})^{-\gamma} \), which is a linearized approximation of the equation of state.

The horizontal mean fields of the thermodynamic variables in the reference state are in hydrostatic balance and make no contribution to the dynamical processes. The deviations of the thermodynamic variables from their mean fields play roles in the dynamic processes. Subtracting the reference state from the governing equations and using the deviations of thermodynamic variables as the computational variables make the system more suitable for numerical procedures, and thus can be interpreted as a preconditioner that effectively enhances the computational stability and efficiency.

b. Governing equations with the effects of topography

In the presence of topography, the height-based terrain-following coordinate introduced by Gal-Chen and Somerville (1975) is utilized to map the physical space \((x, z)\) into the computational domain \((x, \zeta)\) via the transformation relationship \( \zeta = \xi(x, z) \).

Put in the component form in the transformed coordinate system, the governing equations read as

\[
\begin{align*}
\frac{\partial \rho'}{\partial t} + \frac{1}{\sqrt{G}} \left[ \frac{\partial (\sqrt{G} \rho u')}{\partial x} + \frac{\partial (\sqrt{G} \rho w')}{\partial \zeta} \right] & = 0, \\
\frac{\partial (\rho u')}{\partial t} + \frac{1}{\sqrt{G}} \left[ \frac{\partial (\sqrt{G} \rho u'^2 + \sqrt{G} p')}{\partial x} \\
+ \frac{\partial (\sqrt{G} \rho u'w' + \sqrt{G} G^{13} \rho')}{\partial \zeta} \right] & = 0,
\end{align*}
\]
\[
\frac{\partial (pw)}{\partial t} + \frac{1}{\sqrt{G}} \left[ \frac{\partial (\sqrt{G} p w)}{\partial x} + \frac{\partial (\sqrt{G} p w + p')}{\partial \zeta} \right] = -\rho' g, \tag{15}
\]

\[
\frac{\partial (\rho b)}{\partial t} + \frac{1}{\sqrt{G}} \left[ \frac{\partial (\sqrt{G} \rho bu)}{\partial x} + \frac{\partial (\sqrt{G} \rho bu + \rho b')}{\partial \zeta} \right] = 0, \tag{16}
\]

where \( \sqrt{G} = \frac{\partial z}{\partial x} \) is the Jacobian of the transformation and \( G^{13} = \frac{\partial z}{\partial x} \) is the contravariant metric, and \( \dot{w} = \frac{d \zeta}{dt} \) is the velocity component in the transformed coordinate. Here, the following relations are utilized by the chain rule (Clark 1977):

\[
\frac{\partial \phi}{\partial x} \bigg|_{\zeta=\text{constant}} = \frac{1}{\sqrt{G}} \left[ \frac{\partial (\sqrt{G} \phi)}{\partial x} \bigg|_{\zeta=\text{constant}} + \frac{\partial (\sqrt{G} G^{13} \phi)}{\partial \zeta} \right], \tag{17}
\]

\[
\frac{\partial \phi}{\partial \zeta} = \frac{1}{\sqrt{G}} \frac{\partial \phi}{\partial \zeta}. \tag{18}
\]

The quantity \( \phi \) denotes an arbitrary field variable. The relation between the vertical components of velocity in \( z \) coordinate and \( \zeta \) coordinate is

\[
\dot{w} = \frac{d \zeta}{dt} = \frac{1}{\sqrt{G}} (w + \sqrt{G} G^{13} u). \tag{19}
\]

It is noted that the governing Eqs. (13)–(16) in the absence of topography will reduce to Eqs. (9)–(12).

c. Flux Jacobian matrix

The left-hand sides [i.e., the homogeneous part of governing Eqs. (13)–(16)], are of purely hyperbolic type, which have a complete set of real eigenvalues and eigenvectors. We recast Eqs. (13)–(16) into the following vector form:

\[
\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial \zeta} = \mathbf{s}(\mathbf{q}), \tag{20}
\]

where \( \mathbf{q} = [\sqrt{G} p', \sqrt{G} p u, \sqrt{G} p w, \sqrt{G} (\rho \theta)]^T \) are the conservative state variables; \( \mathbf{f} = (\sqrt{G} p w u, \sqrt{G} p w u + \sqrt{G} p') \), \( \sqrt{G} p w u, \sqrt{G} p w u + \sqrt{G} G^{13} p', \sqrt{G} p w + \sqrt{G} G^{13} p' \) denote the vectors of the flux functions in \( x \) and \( \zeta \) directions, and \( \mathbf{s}(\mathbf{q}) = (0, 0, -\sqrt{G} p' g, 0)^T \) are the source terms.

We further decompose the homogeneous part of Eq. (20) as

\[
\frac{\partial \mathbf{q}}{\partial t} + A \frac{\partial \mathbf{q}}{\partial x} + B \frac{\partial \mathbf{q}}{\partial \zeta} = 0, \tag{21}
\]

where \( A \) and \( B \) are the Jacobian matrices of flux functions \( \mathbf{f} \) and \( \mathbf{g} \), and which are computed, respectively, by

\[
A = \frac{\partial \mathbf{f}}{\partial \mathbf{q}} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-\rho^2 & 2 & 0 & \alpha_0 \\
-\rho w & w & u & 0 \\
-\alpha_0 & \theta & 0 & u
\end{bmatrix}, \tag{22}
\]

and

\[
B = \frac{\partial \mathbf{g}}{\partial \mathbf{q}} = \begin{bmatrix}
0 & 0 & 1 & 0 \\
-\rho w & w & u & G^{13} \alpha_0 \\
-w^2 & 0 & 2w & \alpha_0 \\
-w \theta & \theta & 0 & w
\end{bmatrix}. \tag{23}
\]

The hyperbolicity assures that the Jacobians can be diagonalized by the corresponding eigenvectors and eigenvalues (i.e., \( A = R_x L_x L^{-1}_x \) and \( B = R_z L_z L^{-1}_z \)). Some algebraic manipulations yield \( L_x = \text{diag}(u, u - a_u, u + a_u) \) and \( L_z = \text{diag}(w, w - a_w, w + a_w) \), which are the characteristic speeds of the system. The transformation matrices \( R_x \) and \( R_z \) are constructed from the right eigenvectors of \( A \) and \( B \), while \( L_x \) and \( L_z \) are the left eigenvectors, which are given as follows for completeness:
where \( a_u = \sqrt{\gamma \theta p / \rho} \) and \( a_w = \sqrt{\gamma \theta p / (\rho \theta \sqrt{G})} \) are the components of sound speed in \( x \) and \( \zeta \) directions, respectively. In the following numerical formulations, we make use of the approximate Riemann solver, which requires the eigenvalues obtained above.

### 3. Multimoment constrained formulations

#### a. Spatial discretization

As mentioned before, the height-based terrain-following coordinate originally proposed by Gal-Chen and Somerville (1975) is used to map the physical domain \((x, z)\) to the computational domain \((x, \zeta)\) through a transformation (Schär et al. 2002):

\[
\begin{align*}
\zeta(x) &= z_s(x) \frac{\sinh[(z_T - \xi)/s]}{\sinh(z_T/s)},
\end{align*}
\]

where \( z_s(x) \) is the elevation of topography, \( z_T \) is the altitude of the model top level, and \( s \) is the scale height.

We partition the computational domain \( \Omega \) into non-overlapping cells \((C_{ij})\) as a structured mesh where the mesh elements are numbered by \( i \) and \( j \), respectively, in \( x \) and \( z \) directions, such that

\[
\Omega = \bigcup_{i,j=1}^{I,J} C_{ij},
\]

where the mesh cell is defined as \( C_{ij} = [x_{i-1/2}, x_{i+1/2}] \otimes [\zeta_{j-1/2}, \zeta_{j+1/2}] \), and \( I \) and \( J \) are the total numbers of mesh cells in the \( x \) and \( \zeta \) directions.

As shown in Fig. 1, the unknown DOFs are defined as points along the line segment \( x_1(=x_{i-1/2}) \) to \( x_L(=x_{i+1/2}) \).

![Diagram](image-url)
scheme is \([I \times (L - 1) + 1] \times [J \times (L - 1) + 1]\). Our experience shows that in an MCV scheme the numerical result is not sensitive to the location of the solution points. No significant difference is observed in the solutions between the Gauss-point and the equidistance-point configurations. We use the equidistance-point configuration including the cell boundary points in the present model, which is simpler and easy to implement for real applications. In this case, the point value at a boundary point is shared by the adjacent cells and the physical field is \(C^0\) continuous. For simplicity, we drop the cell indices \(i\) and \(j\) from now on and focus our discussions on the local control volume where the solution points are equally spaced over \([x_{i-1/2}, x_{i+1/2}]\) and \([\zeta_{j-1/2}, \zeta_{j+1/2}]\), respectively, by

\[
x_1(=x_{i-1/2}), x_2, \ldots, x_L(=x_{i+1/2}),
\]

\[
\zeta_1(=\zeta_{j-1/2}), \zeta_2, \ldots, \zeta_L(=\zeta_{j+1/2}).
\]

Shown in Ii and Xiao (2009), a fully multidimensional MCV formulation in structured mesh can be derived by directly implementing the one-dimensional formulation along each direction. Without losing generality, we present how to develop the spatial discretization of an MCV scheme for a scalar conservation law,

\[
\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} = 0,
\tag{28}
\]

in the \(x\) direction along the \(m\)th line segment \(x_L x_L \times \zeta_m\), where \(q\) and \(f\) represent any component of the conservative variables \(q\) and the corresponding component of the flux function \(f\), respectively. The same procedure applies in the \(\zeta\) direction along the \(l\)th line segment \(\zeta_L x_L \times x_L\) by only replacing \(x\) with \(\zeta\) and flux component \(f\) with \(g\) for the \(\zeta\) direction. The numerical details presented below can be straightforwardly extended to the system conservation laws in Eq. (20).

To build a high-order MCV scheme over line segment \(x_L x_L \times \zeta_m\), we define the moments as the line-integrated average value (LIA moment), the point value (PV moment), and the derivative value (DV moment) of the field variable \(q\) as the following:

\[
q_m(x, t) = \int_{x_{i-1/2}}^{x_{i+1/2}} q(x, t) \, dx,
\tag{29}
\]

\[
q_{\text{cpm}}(t) = q(x_{\text{cpm}}, t),
\tag{30}
\]

\[
\delta^k_q q_{\text{cpm}}(t) = \frac{\partial^k}{\partial x^k} q(x_{\text{cpm}}, t): \quad \text{with} \quad k = 1, 2, \ldots
\tag{31}
\]

where \(\Delta x_i = x_{i+1/2} - x_{i-1/2}\) and \(x_{\text{cpm}}\) represents a constraint point within or at the two ends of line segment \(x_L x_L \times \zeta_m\) where constraints in terms of multimoments are imposed. The constraint points can, but do not have to, coincide with the solution points. Shown later, the constraint points of the third-order scheme are the two ends of \(x_L x_L \times \zeta_m\), while for the fourth-order scheme an additional constraint is put at the center of the segment.

1) LOCAL RECONSTRUCTION

Given the values of field variable \(\phi_m, l = 1, 2, \ldots, L\), at the solution points \(x_1\) to \(x_L\), a one-dimensional Lagrange interpolation polynomial of degree \((L-1)\) is constructed as

\[
\Psi(\phi : x)_m = \sum_{l=1}^{L} (B_l \phi_{lm}),
\tag{32}
\]

where the Lagrange basis function is

\[
B_l = \prod_{p=1, p\neq l}^{L} \frac{x - x_p}{(x_l - x_p)}.
\tag{33}
\]

The \(\phi\) denotes either the conservative state variables \(q\) or the flux component of \(f\) and \(g\).

2) MULTIMOMENT CONSTRAINTS

As addressed by Ii and Xiao (2009), high-order schemes can be constructed by properly choosing multimoment constraints derived from the governing equation in Eq. (28). We summarize the constraint conditions used in this paper as follows:

C1) The LIA moment over line segment \(x_L x_L \times \zeta_m\), which is cast in a finite-volume formulation,

\[
\frac{d}{dt}[q_m(t)] = -\frac{1}{\Delta x_i} (\hat{f}_{Lm} - \hat{f}_{1m}),
\tag{34}
\]

where \(\hat{f}_{1m}\) and \(\hat{f}_{Lm}\) are the numerical flux at the two ends of segment \(x_L x_L \times \zeta_m\). Because in the present schemes the segment ends are also the solution points where the solution is readily updated, numerical fluxes can be immediately computed. For example, the flux component \(\rho u\) of the continuity equation at the left boundary point can be directly computed by \(\hat{f}_{1m} = \rho_{1m} u_{1m}\), where \(\rho_{1m}\) and \(u_{1m}\) are the solutions at the left boundary point of the cell, which are updated every time step and shared by the two neighboring cells. The finite-volume formulation in Eq. (34) assures the rigorous conservation in numerical solution.
C2) Being other constraints, the PV-moment values at the two ends of segment \(\tau_{ixL} \times \xi_m\) are predicted by a collocation formulation of Eq. (28),
\[
\frac{d}{dt}[q_{1m}(t)] = -\partial_x \hat{f}_{1m} \quad \text{and} \quad \frac{d}{dt}[q_{Lm}(t)] = -\partial_x \hat{f}_{Lm},
\]

(35)
where \(\partial_x \hat{f}_{1m}\) and \(\partial_x \hat{f}_{Lm}\) are the derivatives of the flux function, which need to be computed from the approximate Riemann solver discussed later.

C3) In the fourth-order scheme, an extra constraint condition is imposed on the first-order DV moment at the segment center. (Note that we denote \(x_{cpm}\) as the constraint point at the line segment center by \(x_{cm}\) and the first-order derivative \(q_x\) by \(q\), hereafter.) Its evolution equation is obtained by differentiating the governing equation in Eq. (28) with respect to \(x\) and collocate the resultant equation at cell center, which yields
\[
\frac{d}{dt}[q_{x,cm}(t)] = -\partial_x^2 \hat{f}_{cm},
\]

(36)
where the second-order derivative value of the flux function \(\partial_x^2 \hat{f}_{cm}\) at the segment center is computed from the fourth-order approximation:
\[
(\partial_x^2 \hat{f})_{cm} = -\frac{16f_{cm} + 8\hat{f}_{1m} + 8\hat{f}_{4m}}{\Delta x^2} + \frac{(\partial_x \hat{f})_{4m} - (\partial_x \hat{f})_{1m}}{\Delta x},
\]

(37)
and the line segment center value \(f_{cm}\) is obtained from the Lagrange interpolation in Eq. (32).

3) Third-order MCV scheme

For the third-order MCV scheme (\(L = 3\)), three solution points over line segment \(\tau_{ixL} \times \xi_m\) are located at \(x_1 = x_{i-1/2}\), \(x_2 = [x_{i-1/2} + x_{i+1/2}]/2\), and \(x_3 = x_{i+1/2}\) (note that we omit the subscript \(m\) here), where the solutions \(q_{lm}\), \(l = 1, 2, 3\), are computed. From the constraints C1 and C2 as well as the Lagrange interpolation in Eq. (32), we get the following semidiscrete equations to update the unknowns for \(\tau_{ixL} \times \xi_m\) from the numerical flux and its first-order derivative:
\[
\begin{bmatrix}
\frac{d}{dt}(q_{1m}) \\
\frac{d}{dt}(q_{2m}) \\
\frac{d}{dt}(q_{3m})
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & -1 & 0 \\
3/2 & 3 & 1 & 1 \\
0 & 0 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
\hat{f}_{1m} \\
\hat{f}_{3m} \\
(\partial_x \hat{f})_{1m} \\
(\partial_x \hat{f})_{3m}
\end{bmatrix},
\]

(38)
The evolution equation of \(q_{2m}\) is derived by replacing the LIA moment in Eq. (34) with the nodal solutions \(q_{1m}, q_{2m}, \text{ and } q_{3m}\), using the LIA moment definition in Eq. (29) and the Lagrange interpolation in Eq. (32).

Making use of the following notations:
\[
\mathbf{M}_3^{(x)} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 3/2 & 3 & 1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{F}_3^{(x)} = \begin{bmatrix} \hat{f}_{1m} \\ \hat{f}_{3m} \\ (\partial_x \hat{f})_{1m} \\ (\partial_x \hat{f})_{3m} \end{bmatrix}
\]

(39)
and denoting the entries of matrix \(\mathbf{M}_3^{(x)}\) by \(M_{3\alpha\beta}^{(x)}\) and components of \(\mathbf{F}_3^{(x)}\) by \(F_{3\beta m}\), we rewrite Eq. (38) into a component form:
\[
\frac{d}{dt}(q_{lm}) = \sum_{\beta=1}^{4} M_{3\alpha\beta}^{(x)} F_{3\beta m}, \quad \text{for } l, m = 1, 2, 3.
\]

(40)
The semidiscrete formulation for updating the conservative variables in the 2D system in Eq. (20) is then obtained as
\[
\frac{d}{dt}(q_{lm}) = \sum_{\beta=1}^{4} M_{3\alpha\beta}^{(x)} F_{3\beta m} + \sum_{\beta=1}^{4} M_{3\alpha\beta}^{(x)} G_{3\beta m} + S(q_{lm}),
\]

(41)
for \(l, m = 1, 2, 3\).

The terms \(M_{3\alpha\beta}^{(x)}\) and \(G_{3\beta m}\) are the entries of the following matrix and vector generated from the spatial discretization in \(\xi\) direction:
\[
\mathbf{M}_3^{(x)} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 3/2 & 3 & 1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{G}_3^{(x)} = \begin{bmatrix} \hat{g}_{l1} \\ \hat{g}_{l3} \\ (\partial_x \hat{g})_{11} \\ (\partial_x \hat{g})_{33} \end{bmatrix}
\]

(42)
and \(S(q_{lm})\) represents the pointwise evaluated source term.

4) Fourth-order MCV scheme

In the fourth-order scheme (\(L = 4\)), the unknowns \(q_{lm}\), \(l = 1, 2, 3, 4\), over line segment \(\tau_{ixL} \times \xi_m\), are defined at four solution points located at \(x_1 = x_{i-1/2}\), \(x_2 = x_{i-1/2} + \Delta x/3\), \(x_3 = x_{i-1/2} + 2\Delta x/3\), and \(x_4 = x_{i+1/2}\). Constraint C3 is included in addition to C1 and C2 in the third-order scheme as discussed above. By
analogous manipulation, we obtained the semidiscrete formulation for Eq. (28) over $x_{L} \times \xi_{m}$ as

$$
\frac{d}{dt}(q_{lm}) = \mathbf{M}_{4}^{(c)} \mathbf{F}_{4}^{(c)} ,
$$

(43)

where

$$
\mathbf{M}_{4}^{(c)} = \begin{bmatrix}
0 & 0 & -1 & 0 & 0 \\
4 & 4 & 4 & 4 & 4 \\
\frac{3\Delta x}{3\Delta x_{j}} & \frac{3\Delta x}{3\Delta x_{j}} & \frac{3\Delta x}{3\Delta x_{j}} & \frac{3\Delta x}{3\Delta x_{j}} \\
4 & 4 & 4 & 4 & 4 \\
0 & 0 & 0 & -1 & 0
\end{bmatrix}
$$

and

$$
\mathbf{F}_{4}^{(c)} = \begin{bmatrix}
\hat{f}_{1m} \\
\hat{f}_{2m} \\
(\partial_{x}\hat{f})_{1m} \\
(\partial_{x}\hat{f})_{2m}
\end{bmatrix}.
$$

(44)

Denoting the entries of matrix $\mathbf{M}_{4}^{(c)}$ by $M_{4\alpha\beta}^{(c)}$ and components of $\mathbf{F}_{4}^{(c)}$ by $F_{4\beta m}$, we get the componentwise expression as

$$
\frac{d}{dt}(q_{lm}) = \sum_{\beta=1}^{5} M_{4\alpha\beta}^{(c)} F_{4\beta m} , \text{ for } l = 1, 2, 3, 4.
$$

(45)

Similarly, the fourth-order semidiscrete formulation for the 2D system in Eq. (20) is obtained as

$$
\frac{d}{dt}(q_{lm}) = \sum_{\beta=1}^{5} M_{4\alpha\beta}^{(c)} F_{4\beta m} + \sum_{\beta=1}^{5} M_{4\alpha\beta}^{(c)} G_{4\beta m} + S(q_{lm}),
$$

(46)

for $l, m = 1, 2, 3, 4$.

The $M_{4\alpha\beta}^{(c)}$ and $G_{4\beta m}$ are the entries of the following matrix and vector generated from the spatial discretization in $\xi$ direction:

$$
M_{4}^{(c)} = \begin{bmatrix}
0 & 0 & -1 & 0 & 0 \\
4 & 4 & 4 & 4 & 4 \\
\frac{3\Delta \xi}{3\Delta \xi_{j}} & \frac{3\Delta \xi}{3\Delta \xi_{j}} & \frac{3\Delta \xi}{3\Delta \xi_{j}} & \frac{3\Delta \xi}{3\Delta \xi_{j}} \\
4 & 4 & 4 & 4 & 4 \\
0 & 0 & 0 & -1 & 0
\end{bmatrix}
$$

and

$$
G_{4}^{(c)} = \begin{bmatrix}
\tilde{g}_{l1} \\
\tilde{g}_{l4} \\
(\partial_{\xi}\tilde{g})_{l1} \\
(\partial_{\xi}\tilde{g})_{l4} \\
(\partial_{\xi}^{2}\tilde{g})_{cm}
\end{bmatrix}.
$$

(47)

b. Approximate Riemann solver

In the above discussions we are left with the need to find the numerical flux and its derivative at the cell boundaries. As part of the solution points are at the cell boundaries, the numerical fluxes $f_{1m}$ and $f_{5m}$, as well as $g_{1}$ and $g_{4}$, are continuous at cell boundaries, and can be immediately calculated from the values of the state variables at the boundary solution points. The derivatives of the flux functions, at the cell boundaries, such as $(\partial_{x}f)_{1m}$, $(\partial_{x}f)_{5m}$, $(\partial_{\xi}g)_{1}$, and $(\partial_{\xi}g)_{4}$, however, are not continuous, which require them to be evaluated by using the hyperbolicity of the system. The numerical solver for Riemann problem in terms of derivatives or high-order variations, known as derivative or generalized Riemann problem, was suggested in Toro et al. (2001) and Titarev and Toro (2002) to numerically evaluate the derivatives of the flux functions. We follow the same approach and use the numerical approximate solver for the derivative Riemann problems.

Considering that the Riemann problem can be solved dimensionwise in structured grids, we discuss below the formulation in $x$ direction over $x_{L} \times \xi_{m}$ and its neighboring cells. We denote the boundary point $x_{bpm}$ for either $x_{1}$ or $x_{5}$, where the piecewise reconstructed polynomial in Eq. (32) may lead to two different values of the derivatives of the state variable on the left and right sides of $x_{bpm}$ denoted by $\partial_{x}q_{bpm}^{-}$ and $\partial_{x}q_{bpm}^{+}$, respectively. The corresponding values of the derivatives of flux function are denoted by $\partial_{x}f_{bpm}^{-}$ and $\partial_{x}f_{bpm}^{+}$. The Riemann solvers to find the derivative of the flux function can be generally written as

$$
(\partial_{x}\hat{f})_{bpm} = \text{Riemann}(\partial_{x}f_{bpm}^{-}, \partial_{x}f_{bpm}^{+}),
$$

(48)
which determines numerical derivative of the flux function from the left- and right-hand-side values obtained by the local reconstructions at the cell level conducted in the neighboring cells of boundary point \(x_{bpm}\). For computational efficiency, a local Lax–Friedrich (LLF) approximate Riemann solver (Shu and Osher 1988) is adopted in this paper, which reads as

\[
\hat{\phi}_{bpm} = \frac{1}{2} \left[ \phi_{bpm}^{+} + \phi_{bpm}^{-} - |\lambda| \max (\phi_{bpm}^{+} - \phi_{bpm}^{-}) \right],
\]

(49)

where \(\lambda|\max\) is the local maximum of the eigenvalues of the Jacobian matrix \(A\), which is directly evaluated from the point values at \(x_{bpm}\).

The left- and right-hand-side values required in the above expressions are computed from the cellwise Lagrange interpolations Eq. (32) by

\[
\phi_{bpm}^{+} = \frac{d}{dx} [\Psi^{+}(\phi : x)]_{bpm}, \quad (50)
\]

\[
\phi_{bpm}^{-} = \frac{d}{dx} [\Psi^{-}(\phi : x)]_{bpm}, \quad (51)
\]

\[
\phi_{bpm}^{+} = \frac{d}{dx} [\Psi^{+}(\phi : x)]_{bpm}, \quad (52)
\]

\[
\phi_{bpm}^{-} = \frac{d}{dx} [\Psi^{-}(\phi : x)]_{bpm}, \quad (53)
\]

where \(\Psi^{-}(\phi : x)\) denotes the interpolation function over the left-hand-side cell and \(\Psi^{+}(\phi : x)\) the right-hand-side cell of boundary point \(x_{bpm}\). The numerical procedure described above applies to the \(\zeta\) direction as well.

c. Numerical formulation for dynamic viscosity

Although we start with an inviscid form of the governing equations in Eq. (1), the computation of dynamical viscosity term, \(V \cdot (\mu \nabla \phi_{lm})\), will be involved when the physical viscosity, as in the Navier–Stokes equations, has an effect. It is always the case that some numerical viscosity might be necessary as a filter for high-order schemes where the numerical dissipation is suppressed and noises of small scales do not diminish. In such a case, viscous or dissipative terms are added to the right-hand side of the momentum equations and the energy equation for a given viscosity coefficient \(\mu\). Here \(\phi_{lm}\) denotes the components of the wind \((u, w)\) or the potential temperature \(\theta\).

With the local reconstructed polynomials of degree 2 in MCV3 and degree 3 in MCV4, the numerical formula for the second-order derivative can be directly obtained. We give the formula for \((\phi_{bpm}^{+})_{lm}\) in \(x\) direction, for example,

\[
(\phi_{bpm}^{+})_{lm} = \frac{4\phi_{1m} - 8\phi_{2m} + 4\phi_{3m}}{\Delta x^2}, \quad I = 1, 2, 3,
\]

(54)

for MCV3 and

\[
(\phi_{bpm}^{+})_{1m} = \frac{18\phi_{1m} - 45\phi_{2m} + 36\phi_{3m} - 9\phi_{4m}}{\Delta x^2},
\]

(55)

\[
(\phi_{bpm}^{+})_{2m} = \frac{9\phi_{1m} - 18\phi_{2m} + 9\phi_{3m}}{\Delta x^2},
\]

(56)

\[
(\phi_{bpm}^{+})_{3m} = \frac{9\phi_{2m} - 18\phi_{3m} + 9\phi_{4m}}{\Delta x^2},
\]

(57)

\[
(\phi_{bpm}^{+})_{4m} = \frac{-9\phi_{1m} + 36\phi_{2m} - 45\phi_{3m} + 18\phi_{4m}}{\Delta x^2},
\]

(58)

for MCV4. The same applies to the \(\zeta\) direction.

It should be noted that in the formulation of the MCV schemes (at least those used in the present study) the cell boundary values are shared by the neighboring cells, which are uniquely updated by the derivative Riemann solver. The cell boundary values connect the neighboring cells and maintain at least the \(C^0\) continuity for the physical field, which allows the above formula for second-order derivative to work properly without other special treatment, like the local discontinuous Galerkin (LDG) method (Cockburn and Shu 1998).

As mentioned before, the LLF Riemann solver in terms of the derivative of the flux function is utilized in MCV method, which stabilizes the computation through an effective numerical viscosity. However, as observed in other high-order schemes, extra dissipation is required in some cases as will be reported in section 4.

d. Boundary conditions

We used different boundary conditions for the test cases presented in this paper. The no-flux condition is used along the bottom boundary. According to different test cases, periodical, no-flux and nonreflecting conditions are used, respectively, for the lateral boundaries. We use either no-flux or nonreflecting condition for the top boundary. Next, we describe how the no-flux and nonreflecting boundary conditions are implemented in the MCV schemes. Some representative procedures under other frameworks can be also found in the literature (Duran and Klemp 1983; Giraldo and Restelli 2008; Ullrich and Jablonowski 2012b).

1) NO-FLUX BOUNDARY CONDITIONS

The no-flux boundary condition requires the velocity field to satisfy the following relation:

\[
u \cdot n = 0,
\]
where \( \mathbf{n} \) is the normal direction of the boundary and \( \mathbf{u} = (u, w)^T \) is the wind field. It can be also interpreted as the slip boundary condition, and is used in all test cases in this paper.

We assume that the cells of \( j = 1 \) are boundary cells \( C_{i1} \) and their bottoms coincide with grid line \( \xi_{1/2} \). One layer of ghost cells \( C_{i0} \) is placed beneath \( \xi_{1/2} \). Given the values at the solution points of boundary cell \( C_{i1} \), \( q_{i1m}, m = 1, 2, \ldots, M \) with \( M = 3 \) for the third-order scheme and \( M = 4 \) for the fourth-order scheme, the above no-flux boundary condition can be realized by mapping the boundary cell values to the corresponding points in the ghost cell as follows:

\[
\begin{align*}
(p')_{i0l} &= (p')_{i1l}(M+1-m), \\
(pu)_{i0l} &= (pu)_{i1l}(M+1-m), \\
(pw)_{i0l} &= -(pw)_{i1l}(M+1-m), \\
(w)_{i0l} &= -(w)_{i1l}(M+1-m), \\
[(\rho\theta)^f]_{i0l} &= [(\rho\theta)^f]_{i1l}(M+1-m).
\end{align*}
\]

The above mapping ensures that there is not fluid motion across the bottom boundary. The values in the ghost cells are updated at every time step.

2) Nonreflecting Boundary Conditions

The nonreflecting boundary conditions are mimicked by a sponge layer along the lateral boundaries or top boundary, where damping terms are added to the momentum and potential temperature evolution equations as

\[
\frac{\partial \mathbf{q}}{\partial t} = [\text{governing equation terms}] - \tau (\mathbf{q} - \mathbf{q}_b),
\]

where \( \tau \) is the relaxation coefficients and \( \mathbf{q}_b \) represent some specified background fields such as the mean velocity or the reference state of the thermal-dynamic variables. The strength of the Rayleigh damping \( \tau \) varies over a finite interval in the interior of computational domain. Following the existing works (Giraldo and Restelli 2008; Ullrich and Jablonowski 2012b), the relaxation coefficients is defined as

\[
\tau(s) = \begin{cases} 
0, & \text{if } s < s_0 - s_T, \\
\tau_0 \left[ s - (s_0 - s_T) \right]^4, & \text{otherwise},
\end{cases}
\]

where \( s \in (x, \xi) \) indicates a location in computational domain, and \( s_0 \) means the location of the boundary. The term \( s_T \) denotes the thickness of the absorbing layer and \( \tau_0 \) is the inverse time scale of the damping. Different values for the maximum coefficients are adopted when two absorbing boundary layers, such as the upper boundary and the lateral boundary, overlap (Ullrich and Jablonowski 2012b).

e. Time Integration

Once the spatial discretization is accomplished, the semidiscrete formulations are cast in a set of ordinary differential equations with respect to time as Eq. (41) for the third-order scheme and Eq. (46) for the fourth-order scheme. We use the third-order TVD Runge–Kutta method (Shu 1988) for the time integration. Denoting the right-hand side of Eqs. (41) or (46) by \( \mathcal{L}(q_{lm}) \), we summarize the semidiscrete formulations as

\[
\frac{dq_{lm}}{dt} = \mathcal{L}(q_{lm}).
\]

Given the values \( q_{lm}^n \) at step \( n \), we use the following multistep updating to obtain the values \( q_{lm}^{n+1} \) at step \( n + 1 \):

\[
\begin{align*}
q_{lm}^{(1)} &= q_{lm}^n + \Delta t \mathcal{L}(q_{lm}^n), \\
q_{lm}^{(2)} &= \frac{3}{4} q_{lm}^n + \frac{1}{4} q_{lm}^{(1)} + \frac{1}{4} \Delta t \mathcal{L}(q_{lm}^{(1)}), \\
q_{lm}^{n+1} &= \frac{1}{3} q_{lm}^n + \frac{2}{3} q_{lm}^{(2)} + \frac{2}{3} \Delta t \mathcal{L}(q_{lm}^{(2)}).
\end{align*}
\]

It should be noted that a multistep time integration scheme might not be the final choice in the context of parallel processing. Although the local reconstruction used in an MCV scheme effectively suppresses data communication, using a time marching procedure with fewer substeps will definitely improve the parallel efficiency.

As a matter of fact, we have implemented other time integration approaches under the framework of the multimoment method, such as the semi-Lagrangian method based on characteristics (Li and Xiao 2007) and the conservative semi-implicit semi-Lagrangian method (Li et al. 2008), which can be implemented to the MCV schemes in a straightforward manner. For further exploration, we do not see substantial difficulties to use other alternatives for time integration (e.g., Klemp et al. 2007; Restelli and Giraldo 2009; Ullrich and Jablonowski 2012b; Dumbser et al. 2008), for the MCV formulations.

f. Some remarks

Before ending this section, we want to make some brief remarks on the important features of the MCV schemes.
The largest allowable CFL number for computational stability for 1D advection test was estimated in Li and Xiao (2009), which reveals that the MCV schemes can use larger CFL number in comparison with other high-order schemes of the same order reported in Zhang and Shu (2005). For example, the largest stable CFL number for the third-order MCV scheme with a third-order Runge–Kutta method is 0.4 in terms of the cell width, while those are 0.145 for the discontinuous Galerkin method (Cockburn and Shu 1989) and 0.209 for the spectral volume method (Wang 2002).

We also measured the stable CFL conditions of the atmospheric core in this paper by examining the largest stable CFL number defined by max$_{i\in I}((U + a)_i/\Delta d_i)$ with $U = \sqrt{u^2 + w^2}$ and $\Delta d_i = \Delta z/(L - 1) = \Delta z/(L - 1)$. The third-order TVD Runge–Kutta time integration scheme is used. The largest stable CFL number for the core using the third-order scheme is 0.52, while that for the fourth-order scheme is 0.5. It shows that the stable CFL number remains nearly unchanged (i.e., the time stepping is linearly in proportion to the interval of the neighboring solution points), which to some extent is appealing for the use of uniformly located solution points.

4. Numerical results

We present in this section some standard and widely used benchmark tests including the rising convective thermal bubble, density current, internal gravity waves, and Schär mountain test in the nonhydrostatic atmospheric scope to validate the present compressible nonhydrostatic atmospheric dynamical core in comparison with other existing ones.

For all test cases, the initial hydrostatic state is specified in terms of Exner pressure $\Pi$ and potential temperature $\theta$,

$$\frac{d\Pi}{dz} = -\frac{g}{c_p} \theta,$$

where $\Pi = (p/p_0)^{R/\rho c_p}$.

In the test cases presented in this paper, we specify the basic state by either a constant potential temperature or a constant Brunt–Väisälä frequency, that is,

$$\bar{\theta}(z) = \theta_0,$$

or

$$\bar{\theta}(z) = \theta_0 \exp\left(\frac{N_0^2}{g} z\right),$$

where $N_0^2 = gd\ln\bar{\theta}/dz$ and $\theta_0$ is constant. Thus, the conversion from $\bar{\Pi}$ and $\bar{T} = \bar{\theta}/\Pi$ to $\bar{p}$ is as follows:

$$\bar{p} = \frac{p_0}{R_d T} \frac{\bar{T}}{\Pi}.$$

Unauthenticated | Downloaded 08/12/22 06:29 PM UTC
A rising convective thermal bubble

A rising convective thermal in neutral atmosphere is widely used to validate the ability of numerical models for simulating the atmospheric motions due to thermodynamic effects (e.g., Carpenter et al. 1990; Wicker and Skamarock 1998; Ahmad and Lindeman 2007; Norman et al. 2011). A thermal bubble warmer than the ambient air rises up while deforming as a result of the shearing motion incurred by the wind field.
gradients until it forms a mushroom cloud. Without an analytical solution, the numerical schemes are generally evaluated in a qualitative manner. The air mass is initially at rest and in hydrostatic balance based on a uniform potential temperature $\theta(z) = 300$ K. Then a perturbation in potential temperature is added to the hydrostatic background field so that the potential temperature has the initial distribution as

$$\theta(x, z) = \theta(z) + \Delta \theta \max(0, 1 - r/R),$$

where $r = \sqrt{(x-x_0)^2 + (z-z_0)^2}$ and $R$ is the radius of the bubble. In this test case, $\Delta \theta = 2$ K, $R = 2000$ m, and $(x_0, z_0) = (10000$ m, 2000 m). The simulation is run for 1000 s on a domain of $[0, 20] \times [0, 10]$ km. The boundary conditions are no-flux along all boundaries.

---

**FIG. 3.** As in Fig. 2, but using a 200-m grid spacing with the MCV4 scheme and the time step of 0.05 s.

(a) Potential temperature perturbation

(b) Horizontal wind

(c) Vertical wind
Figure 2 shows the contours of potential temperature perturbations, horizontal wind, and vertical wind computed by the third-order MCV scheme with an equi-distance grid spacing of 125 m. As shown in Fig. 2a, sharp gradient of potential temperature perturbations are created in the upper part of the thermal bubble since the center of convective thermal bubble rises faster than anywhere else inside the bubble because of the distribution of the potential temperature perturbation. An explicit dissipation filter in Eq. (54) is used with a viscosity coefficient of $\mu = 10 \, m^2 \, s^{-1}$. The dissipation filter effectively eliminates the noises (oscillations) of small scale in both potential temperature and velocity fields. The numerical dissipation in the present model is adaptively controllable, which is different from other methods where numerical dissipation is embedded in

---

**Fig. 4.** As in Fig. 2, but for with a 62.5-m grid resolution. The time step is 0.0375 s for this grid spacing. The explicit numerical filter was not used in this simulation.
the reconstruction procedure, such as Ahmad and Lindeman (2007) and Norman et al. (2011). A proper limiting projection to enforce the monotonicity when implementing an MCV scheme for atmospheric modeling should be another interesting topic worthy of further exploration.

The fine structures in the numerical solutions are resolved. It is observed that the symmetric structures of both potential temperature and wind field as in Fig. 2 are perfectly reproduced similar to previous studies such as Ahmad and Lindeman (2007) and Norman et al. (2011), although different contour intervals are used here.

Figure 3 shows the numerical results of the fourth-order MCV scheme on a coarser grid spacing of 200 m, which corresponds to the results given in Fig. 2 with regard of the total number of DOFs. The numerical dissipation filter in Eq. (55) was used in this example with $\mu = 10 \text{ m}^2 \text{ s}^{-1}$. Again, the symmetric structures of
the physical fields are perfectly simulated. As shown in Fig. 3, the structures in both potential temperature and velocity fields look more intensive compared to those from the third-order MCV scheme.

We have also run simulations without any dissipation filter using different grid resolutions. We doubly refined the grid spacing for the third-order MCV scheme to 62.5 m and the fourth-order MCV scheme to 100 m, respectively, which requires approximately the same number of DOFs for both schemes. As shown in Figs. 4 and 5, refining the grid resolution effectively reduces the numerical oscillations of small scales. This observation agrees well with the common understanding that a refined grid can resolve better structures of smaller scale, which are more likely to remain in a less dissipative model of higher-order convergence. It reveals that the simulation results on a low-resolution grid adequately reproduce the converged solutions in the present model.

In a qualitative manner, the time histories of the maximum of potential temperature perturbations and vertical velocity in a series of grid refinement experiments without the explicit dissipation filter are plotted in Fig. 6. Simulations were run with gradually refined resolutions for the third-order and fourth-order MCV schemes where the finest grid spacings are 62.5 m for MCV3 and 100 m for MCV4 schemes. In the results of high-resolution simulations, the convexities of vertical velocity during [600, 900] s are observed, and the maximum vertical velocity appears around 800 s. It agrees well with other existing studies (e.g., Ahmad and Lindeman 2007; Norman et al. 2011).

**b. Density current**

The density current benchmark suggested by Straka et al. (1993) is often used to evaluate dynamical cores of atmospheric models. The evolution of a density flow generated by a cold bubble in a neutrally stratified

---

**FIG. 6.** The time evolution of the maximum (a),(c) potential temperature perturbation and (b),(d) vertical wind for the convective thermal test case over a range of grid spacings: (top) 62.5–500 m and (bottom) 100–800 m. The x axis is in s and the y axis is in K and m s$^{-1}$ for vertical velocity. The explicit numerical filter was not used in these simulations.
atmosphere is simulated in this test. The cold bubble descends to the ground and spreads out in the horizontal direction, forming three Kelvin–Helmholtz shear instability rotors along the cold front surface. The background field in the initial condition is in a hydrostatic balance, which is similar to the convective thermal bubble test discussed above where a constant potential temperature of \( \theta(z) = 300 \text{ K} \) is specified. Then the following potential temperature perturbation is added to the constant potential temperature background:

\[
\theta(x, z) = \begin{cases} 
\theta(z), & \text{if } r > 1 \\
\theta(z) + \Delta \theta [\cos(\pi r) + 1], & \text{if } r \leq 1
\end{cases}
\]

where \( \Delta \theta = -15 \text{ K}, (x_0, z_0) = (0, 3000 \text{ m}), \) and \((x_r, z_r) = (4000, 2000 \text{ m})\). The simulation is run for 900 s on a domain of \([-26.5, 26.5] \text{ km} \times [0, 6.4] \text{ km}\). Note that the potential temperature perturbation is adopted similar to Giraldo and Restelli (2008), which is a little different

**FIG. 7.** Potential temperature perturbation after 900 s using (a) 400-, (b) 200-, (c) 100-, (d) 50-, and (e) 25-m grid spacings with the MCV3 scheme and the TVD RK3 time integrator. The contour values are between −9 and 0 with an interval of 1. The x and y axes are in km and potential temperature perturbations are in K, respectively. The time step is 0.24 s for the 400-m grid spacing and is linearly decreased as the grid spacing is gradually refined.

**FIG. 8.** As in Fig. 7, but using (a) 600-, (b) 300-, (c) 150-, (d) 75-, and (e) 37.5-m grid spacings and with the MCV4 scheme. The time step is 0.2 s for the 600-m grid spacing and linearly decreases as the grid spacing is gradually decreased.
from the original temperature perturbation defined by Straka et al. (1993).

The boundary conditions for all four boundaries are no flux. Being a requirement of the physical process, dynamic viscosity and diffusion are added to the momentum and potential temperature equations with the dissipation coefficient of $\mu = 75 \text{ m}^2 \text{s}^{-1}$ in this test case. The second-order derivative terms are computed by the numerical formulation discussed in section 3c.

Figure 7 indicates the potential temperature perturbation contours after 900 s for 400-, 200-, 100-, 50-, and 25-m grid spacings using the third-order MCV scheme. It is observed that two of the three Kelvin–Helmholtz rotors are reproduced at the very coarse resolution of 400 m. The second rotor comes forth when the grid resolution becomes finer than 200 m. The solutions on grids finer than 100 m converge with the vortical structures adequately simulated.

Figure 8 shows the potential temperature perturbation contours after 900 s for 600-, 300-, 150-, 75-, and 37.5-m grid spacings computed by the fourth-order MCV scheme. With the same number in the DOFs of third-order MCV scheme on a 400-m grid, the fourth-order MCV method on a coarser grid of 600 m reproduced the three rotors. The numerical solutions converge when the grid spacing is finer than 300 m, and the solutions on 150- and 75-m grids are visually identical to the converged results shown as those of the 25-m grid for MCV3 and the 37.5-m grid for MCV4. It reveals that a higher-order scheme converges with the numerical solution more rapidly with the same DOFs.

The profiles of the potential temperature perturbation along the horizontal direction at the height of $z = 1200 \text{ m}$ are plotted in Fig. 9 for both third-order and fourth-order MCV schemes. Figure 9a shows the results from the highest grid resolutions for two schemes: 25 and 37.5 m for the MCV3 and MCV4 schemes, respectively. Both solutions are well converged and no noticeable difference is seen between them. It is observed that three valleys exist, which correspond to the three distinct rotors shown in Figs. 7 and 8. Compared with the results of other high-order numerical methods such as the spectral element and discontinuous Galerkin, our results agree well with that of DG3 (see Fig. 8a in Giraldo and Restelli 2008) with competitive performance. Figure 9b shows the numerical results from different grid resolutions computed by the third-order MCV scheme. It is indicated that at coarse resolution such as 400 and 200 m the potential temperature profiles vary significantly. However, there are not much noticeable changes for the potential temperature profiles on high-resolution grids finer than 100 m. With the equivalent

![Fig. 9. Profiles of potential temperature perturbation after 900 s along 1200-m height using MCV schemes: (a) high-resolution simulations of MCV3 (with 25-m grid spacing) and MCV4 (with 37.5-m grid spacing) schemes. The total number of the DOFs is equivalent to a conventional finite-volume scheme with 12.5-m grid spacing; (b) MCV3 and (c) MCV4 each with five different resolutions. The x axis is in km and the y axis is in K.](image-url)
numbers of DOFs for different grid resolutions, the numerical results by the fourth-order MCV scheme are shown in Fig. 9c, which reveals an adequate convergence when the grid resolution is finer than 300 m. Again it indicates a better convergence of the fourth-order method, which is competitive with the existing high-order schemes, for example, the DG method as shown in Fig. 8b of Giraldo and Restelli (2008).

Table 1 shows the comparisons of the front location, maximum–minimum of potential temperature, and density perturbations between the numerical results from the third- and fourth-order schemes using the same fine resolution in terms of equivalent number of DOFs. It is evident that the two MCV schemes with high-resolution grids give the converged solution. We have obtained −9.06-K minimum potential temperature perturbation in comparison with that of −9.08 K in DG3 and SE3 [see Table 5 in Giraldo and Restelli (2008)] where the same initial conditions of potential temperature perturbation are utilized.

c. Internal gravity waves

We tested the numerical core with the internal gravity waves (IGWs) problem. This test involves the evolution of a potential temperature perturbation in a channel having a periodicity in the horizontal direction. The initial conditions used in this paper are identical to those of Skamarock and Klemp (1994). The atmosphere is

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Front location (m)</th>
<th>$\rho'_\text{max}$</th>
<th>$\rho'_\text{min}$</th>
<th>$\theta'_\text{max}$</th>
<th>$\theta'_\text{min}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Third</td>
<td>14 883</td>
<td>0.036</td>
<td>−0.0019</td>
<td>8.52 × 10⁻¹⁵</td>
<td>−9.06</td>
</tr>
<tr>
<td>Fourth</td>
<td>14 896</td>
<td>0.036</td>
<td>−0.0019</td>
<td>1.19 × 10⁻⁷</td>
<td>−9.06</td>
</tr>
</tbody>
</table>

Fig. 10. Potential temperature perturbation after 3000 s using (a) 100-m grid spacing for MCV3 scheme and (b) 250-m grid spacing for MCV4 scheme with the TVD RK3 time integrator. An aspect ratio of grid spacing, $\Delta x = 10\Delta z$, is used for this internal gravity wave test. The time step is 0.1 s for the MCV3 scheme and 0.15 s for the MCV4 scheme. The contour values are between −0.0015 and 0.003 with an interval of 0.0005. The x and y axes are in km and the contours of potential temperature perturbation are in K.
initially hydrostatic with a constant Brunt–Väisälä frequency of $N_0 = 10^{-2}$ s$^{-1}$ to admit IGWs. A potential temperature perturbation is added to the basic potential temperature field as follows:

$$\theta(x, z) = \bar{\theta}(z) + \Delta \theta \frac{\sin(\frac{\pi z}{H})}{1 + (x - x_0)^2/a^2},$$

(69)

where $H = 10$ km, $\Delta \theta = 0.01$ K, $x_0 = 100$ km, $a = 5$ km, and $\bar{\theta}(z)$ is given by Eq. (65) with the constant temperature $\theta_0 = 300$ K. The initial state of the atmosphere is assumed to ride on a constant mean flow of $\bar{u} = 20$ m s$^{-1}$ for adventing the entire IGW train. The simulation is run for 3000 s on a domain of $[0, 300]$ km $\times$ $[0, 10]$ km. No-flux boundary conditions are used along the bottom and top boundaries while periodic boundary conditions are used along the lateral boundaries.

Table 2 shows the maximum and minimum of vertical velocities and potential temperature perturbations for third- and fourth-order MCV schemes after 3000 s. It can be seen that the third- and fourth-order schemes obtained almost the same numerical outputs though different grid spacings are used here. Specifically, as mentioned by Giraldo and Restelli (2008), the ranges of potential temperature perturbations are $\theta' \in [-1.49 \times 10^{-3}, 2.82 \times 10^{-3}]$ in the numerical results of the model based on the technique of flux-based wave decomposition in Ahmad and Lindeman (2007) and $\theta' \in [-1.51 \times 10^{-3}, 2.78 \times 10^{-3}]$ from models based on the spectral element and discontinuous Galerkin in Giraldo and Restelli (2008), while our results are $\theta' \in$

![Fig. 11. Potential temperature perturbations along the line $z = 5$ km for the internal gravity wave test after 3000 s with different grid resolutions. The cell aspect ratio is $\Delta x = 10\Delta z$. The x axis is in km and the y axis is in K.](image-url)
as shown in Table 2. It is observed that these numerical results agree well with each other.

Figure 11 gives the profiles of potential temperature perturbations along the horizontal cross section \( z = 5 \) km after 3000 using the three-order and fourth-order MCV schemes. It is observed that the profiles are perfectly symmetric about the position \( x = 16 \) km. Compared with the results of the model using fifth-order weighted essentially nonoscillatory (WENO; Norman et al. 2011), the numerical solution of competitive quality are obtained by utilizing the equivalent DOF resolution with the third-order MCV scheme, as shown in Fig. 11a. The fourth-order MCV scheme, with a little increase in the equivalent DOF resolution, results in significantly improved numerical outputs as shown in Fig. 11b. In addition, the fourth-order MCV scheme also has competitive numerical results with the same grid spacing of 250 m compared to those using the schemes of spectral element and discontinuous Galerkin methods [see Fig. 2b in Giraldo and Restelli (2008)].

d. Schär mountain

We computed the Schär mountain test (Schär et al. 2002) to evaluate the capability of the model to deal with the effects of complex terrain. A particular topography of a five-peak mountain ridge is defined by

\[
zs(x) = h_0 \exp \left[ -\left( \frac{x}{a_0} \right)^2 \right] \cos \left( \frac{\pi x}{\lambda_0} \right),
\]

(70)

where \( h_0 = 250 \) m, \( a_0 = 5000 \) m, and \( \lambda_0 = 4000 \) m. A plot of the mountain profile is shown in Fig. 12. As mentioned above, the hybrid vertical coordinate (Schär et al...
(2002) is adopted. Following the configuration setup of Schär et al. (2002), \( s = 3 \) km is specified in Eq. (26).

The basic state of the atmosphere has a constant mean flow of \( \vec{u} = 10 \) m s\(^{-1}\) and a uniform stratification with a constant Brunt–Väisälä frequency of \( N_0 = 0.01 \) s\(^{-1}\). The reference potential temperature is given by Eq. (65) with \( \theta_0 = 280 \) K.

As described in Schär et al. (2002), gravity waves of different scales are generated by the mountain in the lower atmosphere. The larger-scale hydrostatic waves propagate vertically through the whole domain, while the smaller-scale waves decay rapidly with height due to the nonhydrostatic effects. The mountain wave consists of different scales and has more complex structures in the low atmosphere. For the Schär mountain test, the semi-analytic solutions based on linear theory can be found by a Fourier transformation (Smith 1979, 1980).

The computational domain is \([-25000, 25000] \times [0, 21000] \) m with the grid spacings of \( \Delta x = 250 \) m in the \( x \) direction and \( \Delta \zeta = 210 \) m in the \( \zeta \) direction. No-flux boundary conditions are used for the bottom boundary, and nonreflecting boundary conditions are imposed by placing sponge layers in the regions of \( \zeta \approx 9000 \) m for the top boundary and \( |x| \approx 15000 \) m for the lateral outflow boundary.

Figures 13 and 14 show the simulated results of horizontal and vertical wind speeds after 10 h by MCV3 and MCV4 schemes, respectively, against the semi-analytical solution. In comparison with the results of other numerical models (Klemp et al. 2003; Giraldo and Restelli 2008; Ullrich and Jablonowski 2012b; Simmaro and Hortal 2012), the results of the present model look very competitive.

For further comparison, the root-mean-square (RMS) errors against the semi-analytical solution are examined and compared with the results reported in Giraldo and Restelli (2008). Table 3 gives the RMS errors for different physical variables (\( II, u, w, \theta \)) in our simulations. Compared with the outputs of high-order numerical methods of local reconstructions, such as the SE method and the DG method reported in Giraldo and Restelli (2008, see Table 6 therein), the RMS errors of our results are smaller in this test case. Moreover, other smoother vertical coordinates such as Schär et al. (2002) and Klemp (2011) can be adopted in future to improve the accuracy. It is also observed that although the MCV4 scheme gives less diffusive results compared to the
MCV3 scheme, but it does not show a superiority in numerical accuracy in this particular case. Leaving the lucid explanation to further investigation, we think that part of the reason may be due to the boundary conditions and the vertical coordinate transformation currently used, which might not be necessarily favorable for the high-order schemes.

5. Conclusions

We have presented a 2D nonhydrostatic compressible atmospheric core by using the third- and fourth-order multimoment constrained finite-volume (MCV) schemes. The unknown DOFs are the point values defined at equally spaced solution points within each mesh cell. High-order reconstructions are conducted at the cell level through the point values by 1D Lagrange interpolations in structured grids. The time evolution equations for updating the unknowns are derived from a set of constraint conditions in terms of multimoments such as volume-integrated average (VIA), point value (PV), and spatial derivative values (DV). There is no any numerical quadrature explicitly involved in the present formulations, and the solution points can be flexibly located within each mesh element, which makes the resulting models very efficient and particularly attractive when dealing with the source terms of physical processes in atmospheric models.

The numerical results of widely used standard benchmark tests including the topographic effects show that the present dynamical core can produce numerical solutions of good quality comparable to other high-order schemes as expected. The numerical formulations based on multimoment constraints, as presented in this paper, are exactly conservative and have significant advantages in algorithmic simplicity, flexibility, and computational efficiency.

As the first implementation of MCV schemes in developing nonhydrostatic compressible dynamical cores, we have very promising results from the present study. Further research will be continued to develop dynamical cores using the same methodology for the global atmosphere where the spherical geometry is another key issue, which has been extensively investigated in our previous researches with promising outputs as well.

Acknowledgments. This work is supported by the National Key Technology R&D Program of China (Grant 2012BAC22B01), the National Natural Science Foundation of China (Grants 10902116 and 40805045), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant 41221064), and special research of public service sectors (meteorological) of China (Grant GYHY201006013). We gratefully acknowledge Professor F. X. Giraldo for providing the semi-analytic solution of the Schärf mountain test for comparison. We also thank the anonymous reviewers for their constructive suggestions.

REFERENCES


