A Fourth-Order Solution-Adaptive CENO Scheme for Space-Physics Flows on Three-Dimensional Multi-Block Cubed-Sphere Grids

Lucian Ivan[1], Hans De Sterck[1], and Clinton P. T. Groth[3]

[1] *Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1*
[2] *University of Toronto Institute for Aerospace Studies, Toronto, Ontario, Canada M3H 5T6*

Email: *lucian.ivan@uwaterloo.ca*

Abstract

A high-order central essentially non-oscillatory (CENO) finite-volume scheme in combination with a block-based adaptive mesh refinement (AMR) algorithm is proposed for solution of hyperbolic conservation laws on three-dimensional cubed-sphere grids. In particular, the fluid flows of interest are governed by the compressible form of Euler and ideal magnetohydrodynamics (MHD) equations and pertain to space-physics applications. The adaptive cubed-sphere simulation framework represents a flexible design based on a genuine multi-block implementation, leading to high-order accuracy, flux calculation, adaptivity and parallelism that are consistent throughout the domain, including at the boundaries between the sectors of the cubed-sphere grid. The CENO scheme is based on a hybrid solution reconstruction procedure that provides high-order accuracy in smooth regions, even for smooth extrema, and non-oscillatory transitions at discontinuities. Furthermore, the CENO formulation is naturally uniformly high-order on the whole cubed-sphere grid including at sector boundaries and corners. The scheme is applied in combination with a divergence correction technique to enforce the solenoidal condition for the magnetic field. A fourth-order quadrature rule is derived and employed in transferring the solution content between levels of coarse- and fine-grained mesh resolution so as to preserve the high-order character of the solution while minimizing the computational effort and inter-processor communication. Several numerical results are presented and discussed to demonstrate the accuracy and current capabilities of the three-dimensional, high-order, solution-adaptive CENO computational framework for multi-block cubed-sphere grids.

1 Scope of Current Work

High-order accurate and efficient computational methods are highly desirable in many fields of computational physics, especially in the study of problems characterized by a wide range of temporal and length scales on which the interesting physics occurs, such as the global magnetohydrodynamics (MHD) modelling of space-physics problems. To obtain accurate and more affordable solutions of such complex flows, the usage of both high-order discretizations and adaptive mesh refinement (AMR) is often required.

Adaptive mesh refinement is an effective approach for coping with the computational cost of large-scale numerical simulations, such as those encountered in space physics. For geometries involving spherical objects, which is usually the case in space physics flows, second-order schemes in combination with various AMR strategies have been successfully developed and widely used on Cartesian and latitude-longitude spherical (or spherical-polar) grids to enhance local solution features [13–15, 20, 21, 29, 33, 36–38]. Moreover, AMR strategies for discretizations of spherical domains based on cubed-sphere grids [31], which recently emerged as an attractive alternative to spherical-polar grids, have also been proposed in both two-dimensions (2D) by St-Cyr *et al.* [32] and, later on, in three-dimensions (3D) by Ivan *et al.* [18]. An important characteristic of the latter approach is that rather than mapping the cubed-sphere grid to a Cartesian computational domain (a typical approach for 2D spheres [8, 32]), the employed strategy is based on a multi-block approach with unstructured root connectivity, previously proposed by Gao and Groth [12], thereby providing a multi-block AMR finite-volume framework that is more general and not restricted

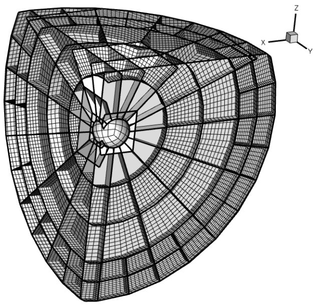


Figure 1: Complex cut into an adapted cubed-sphere grid showing block boundaries and associated meshes.

only to cubed-sphere topologies. An example of an adapted, isotropically-refined, cubed-sphere grid that has been obtained with the AMR framework described in [18] is shown in Fig. 1. Very recently, Williamschen and Groth [41] developed a second-order anisotropic refinement strategy for multi-block hexahedral grids and applied it to cubed-sphere grids as well.

Despite significant advances in new discretization schemes of various types for 3D hyperbolic conservation laws, including finite difference methods (e.g., [26]), discontinuous Galerkin methods (e.g., [6, 24, 40]), finite-volume methods (e.g. [1, 2, 8, 22]), and combinations of these approaches (e.g., [10]), most advanced MHD frameworks for parallel space-physics simulations are only second-order accurate [21,36,38].

Recently, Ivan *et al.* [16, 19] have extended the high-order central essentially non-oscillatory (CENO) finite-volume schemes [17, 34] to 3D geometries with structured hexahedral grids, with particular emphasis on cubed-sphere mesh topologies. Although the formulation is general and suitable for solution accuracies of different orders, the emphasis was on providing fourth-order accuracy. The scope of our current work is to further advance the computational framework by combining the 3D CENO approach [16, 19] with the block-based AMR strategy described in [18], and provide a first assessment on the predictive capabilities of the proposed 3D high-order AMR framework for solutions of hyperbolic conservation laws. The overarching objective of the research is to develop a flexible, scalable and efficient 3D high-order framework that can be applied in combination with (an)isotropic mesh refinement to various space-physics flows.

A first technical challenge in the application of CENO scheme to cubed-sphere grids is the high-order treatment of hexahedral cells with non-planar cell faces, that are present in cubed-sphere discretizations. A second major technical challenge that must be addressed is how to deal with reconstruction stencils at degenerate block edges and corners. Near these regions the numerical scheme must adapt to deviations from the regular mesh topology and find efficiently the right neighbours to be included in the supporting reconstruction stencil. A third challenge is the formulation of efficient algorithms to transfer solution content accurately between blocks of different grid resolution. Consistent algorithms have been developed to address all these challenges efficiently, thereby allowing the application of the CENO scheme to multi-block cubed-sphere grids and the carrying out of large-scale simulations on massively-parallel computing clusters with distributed memory architecture. The remainder of the paper provides a brief description of the proposed high-order solution-adaptive CENO algorithm along with a selected summary of numerical results.

2 Fourth-Order CENO Method on 3D Cubed-Sphere Grids

Herein, nonlinear conservation laws of the form

$$\partial_t \mathbf{U} + \vec{\nabla} \cdot \vec{\mathbf{F}} = \mathbf{S} + \mathbf{Q} \qquad (1)$$

are considered, where $\mathbf{U}$ is the vector of conserved variables, $\vec{\mathbf{F}}$ is the flux dyad, and $\mathbf{S}$ and $\mathbf{Q}$ are numerical and physical source terms that may arise for certain equation sets and application problems. In particular, this paper discusses the development and application of the fourth-order adaptive CENO method on cubed-sphere grids for the cases of the MHD and Euler equations. For MHD using the generalized Lagrange multiplier (GLM) approach to control divergence errors, as in [9,34], $\mathbf{U}$ is given by $\mathbf{U} = \left[\ \rho,\ \rho\vec{V},\ \vec{B},\ \rho e,\ \psi\ \right]^{\mathrm{T}}$, where ρ is the gas density, $\vec{V} = (u, v, w)$ is the velocity, $\vec{B} = (B_x, B_y, B_z)$ is the magnetic field, ρe is the total energy and ψ is the generalized Lagrange multiplier employed to control errors in the divergence of the magnetic field. Here, the total energy is given by $\rho e = p/(\gamma-1) + \rho V^2/2 + B^2/2$, where V and B are the magnitudes of the velocity and magnetic field vectors, respectively, and γ is the ratio of specific heats. The flux dyad, $\vec{\mathbf{F}}$, is given by

$$\vec{\mathbf{F}} = \left[\begin{array}{c} \rho\vec{V} \\ \rho\vec{V}\vec{V} + \left(p + \dfrac{\vec{B}\cdot\vec{B}}{2}\right)\mathbb{I} - \vec{B}\vec{B} \\ \vec{V}\vec{B} - \vec{B}\vec{V} + \psi\,\mathbb{I} \\ \left(\rho e + p + \dfrac{\vec{B}\cdot\vec{B}}{2}\right)\vec{V} - (\vec{V}\cdot\vec{B})\vec{B} \\ c_h^2\vec{B} \end{array} \right], \qquad (2)$$

where $\mathbb{I}$ is the 3×3 identity matrix. The numerical source term employed to control magnetic field divergence [9, 34] is $\mathbf{S} = \left[\ 0,\ \vec{0},\ \vec{0},\ 0,\ -\frac{c_h^2}{c_p^2}\psi\ \right]^{\mathrm{T}}$. The parameters c_h and c_p are chosen as in [34], following [9]. When $\vec{B}$ and ψ are set to zero, the equations reduce to the Euler equations of gas dynamics.

We formulate the finite-volume method for hexahedral cells in a 3D structured grid block with indices ijk. Thus, the semi-discretization for the temporal evolution of the cell average $\overline{\mathbf{U}}_{ijk}$ in cell ijk is derived from

$$\frac{d\overline{\mathbf{U}}_{ijk}}{dt} = \frac{1}{V_{ijk}}\left[-\oiint_{\partial\mathcal{V}_{ijk}} \vec{\mathbf{F}}\cdot\vec{n}\,\mathrm{da} + \iiint_{\mathcal{V}_{ijk}} (\mathbf{S}+\mathbf{Q})\,\mathrm{dv}\right], \tag{3}$$

where V_{ijk} is the volume of cell $\mathcal{V}_{ijk}$, $\vec{n}$ is the unit outward normal of the cell surface $\partial\mathcal{V}_{ijk}$, and da and dv are surface and volume elements, respectively. Approximating the integrals in Eq. 3 numerically gives

$$\frac{d\overline{\mathbf{U}}_{ijk}}{dt} = -\frac{1}{V_{ijk}}\sum_{f=1}^{6}\sum_{m=1}^{N_g}\left(\tilde{\omega}\vec{\mathbf{F}}_{\mathrm{num}}\cdot\vec{n}\right)_{ijk,f,m} + \overline{\mathbf{S}}_{ijk} + \overline{\mathbf{Q}}_{ijk}, \tag{4}$$

where f is used to indicate the six interfaces of the hexahedral cell, and m indicates the Gauss quadrature points over each interface. Here, $\vec{\mathbf{F}}_{\mathrm{num}}$ is the numerical flux function, $\overline{\mathbf{S}}_{ijk}$ and $\overline{\mathbf{Q}}_{ijk}$ are cell-averaged source terms. High-order integration based on Gaussian quadratures for hexahedral cells is discussed in Sect. 2.2. The right-hand side of Eq. 4 denotes the residual vector in cell ijk, $\mathbf{R}_{ijk}(\overline{\mathbf{U}})$, which depends on the set of cell averages $\overline{\mathbf{U}}$ for a given tessellation.

2.1 High-order CENO Reconstruction

The 2D CENO method [16, 17, 34] achieves high-order accuracy by employing K-exact polynomial reconstruction [3] of the solution field in each cell using the cell averages in a reconstruction stencil formed with neighbouring cells. Once determined, the K-exact polynomials provides sufficiently accurate intercellular solution approximations to compute an accurate numerical flux at each Gauss quadrature point. In our 3D extension, a K-exact polynomial of the form

$$u^K_{ijk}(\vec{X}) = \sum_{\substack{p_1=0\\(p_1+p_2+p_3\leq K)}}^{K}\sum_{p_2=0}^{K}\sum_{p_3=0}^{K}\left(x^{p_1}y^{p_2}z^{p_3}\right)_{ijk} D^K_{p_1p_2p_3} \tag{5}$$

is computed for each solution variable in cell ijk, where K is the degree of the polynomial, $\vec{X}=(x,y,z)$ is the coordinate vector,

$$\left(x^{p_1}y^{p_2}z^{p_3}\right)_{ijk} \equiv \left(x-\bar{x}_{ijk}\right)^{p_1}\left(y-\bar{y}_{ijk}\right)^{p_2}\left(z-\bar{z}_{ijk}\right)^{p_3} \tag{6}$$

is the monomial of powers (p_1,p_2,p_3), $(\bar{x}_{ijk},\bar{y}_{ijk},\bar{z}_{ijk})$ are the coordinates of the centroid of cell ijk, and the

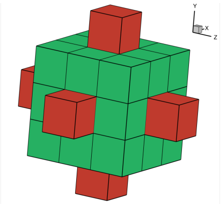


Figure 2: Central reconstruction stencil of 33 cells.

$D^K_{p_1p_2p_3}$ are the polynomial coefficients that are determined such that $u^K_{ijk}(\vec{X})$ matches cell averages in the reconstruction stencil with high accuracy. There are $\mathcal{N}_\mathrm{D} = (K+1)(K+2)(K+3)/6$ polynomial coefficients $D^K_{p_1p_2p_3}$, and they can be computed such that a function $u_{\mathrm{exact}}(\vec{X})$ is represented with accuracy

$$u^K_{ijk}(\vec{X}) - u_{\mathrm{exact}}(\vec{X}) = O(\Delta x^{K+1}), \tag{7}$$

while a polynomial with degree K can be reconstructed exactly [3]. In this paper, cubic reconstruction ($K=3$) is used to obtain a fourth-order accurate numerical scheme, in which case $\mathcal{N}_\mathrm{D}=20$. We choose the coefficients such that the cell average of the reconstruction exactly equals the cell average of variable u in cell ijk,

$$\bar{u}_{ijk} = \frac{1}{V_{ijk}}\iiint_{\mathcal{V}_{ijk}} u^K_{ijk}(\vec{X})\,\mathrm{dv}, \tag{8}$$

and for all cells $\gamma\delta\zeta$ in the reconstruction stencil of cell ijk we also desire that the cell averages of the reconstruction equal the cell averages $\bar{u}_{\gamma\delta\zeta}$:

$$\frac{1}{V_{\gamma\delta\zeta}}\iiint_{\mathcal{V}_{\gamma\delta\zeta}} u^K_{ijk}(\vec{X})\,\mathrm{dv} - \bar{u}_{\gamma\delta\zeta} = 0. \tag{9}$$

As in [3, 16, 17, 34], we use overdetermined reconstruction stencils, and impose Eqs. (9) in the least-squares sense, while Eq. (8) is imposed exactly. Note that the use of overdetermined stencils is especially attractive in the case of cubed-sphere grids, where the grid topology is unstructured near root-block edges and smaller numbers of neighbour cells are available than in regular Cartesian topology. This work uses the reconstruction stencil with 33 cells shown in Fig. 2. Reconstructions of orders higher than cubic can also be computed by enlarging the supporting stencil. Thus, a quartic reconstruction ($K=4$) can be obtain with symmetric stencils having 57 and 81 cells (see [19] for details and analysis).

Equation (8) can be enforced analytically by replacing u^K_{ijk} with Eq. (5) and expressing the first coefficient, D^K_{000}, as a function of the other $M=\mathcal{N}_\mathrm{D}-1$ polynomial unknowns and the volume average values of $\left(x^{p_1}y^{p_2}z^{p_3}\right)_{ijk}$ over $\mathcal{V}_{ijk}$. The latter volume-weighted integrals represent the geometric moments

$\left(\overline{x^{p_1}y^{p_2}z^{p_3}}\right)_{ijk}$ of powers (p_1,p_2,p_3). Substituting u^K_{ijk} from Eq. (5) in Eq. (9) and replacing D^K_{000} an overdetermined linear system of the form

$$\mathbb{L}_{N_n\times M}\mathbf{D}_{M\times 1}-\mathbf{B}_{N_n\times 1}=\mathbf{0}_{N_n\times 1} \tag{10}$$

for the M polynomial unknowns is obtained. Each row in Eq. 10 represents Eq. (9) written explicitly for a particular neighbouring cell out of the N_n number of neighbours in the reconstruction stencil. Thus, the coefficient matrix $\mathbb{L}$ depends only on mesh geometry, $\mathbf{D}$ is the matrix of unknown polynomial coefficients and entries in $\mathbf{B}$ matrix are solution differences between neighbouring cells. Inverse square geometric weighting is applied to the linear system Eq. 10 [16].

The solution of the overdetermined linear system Eq. 10 can be obtained using QR factorization or by multiplication with the pseudo-inverse of $\mathbb{L}$ [16, 23]. In each time step, the constrained least-squares reconstruction problem is solved for each cell and for each primitive variable. Matrix $\mathbb{L}$ depends completely on geometry and is the same for all least-squares problems in a given cell ijk and for all time steps, so its inverse can be precomputed and reused to provide computational speedup. Column scaling can be applied to improve conditioning of the system [16].

A specific technical difficulty in extending the 2D high-order CENO finite-volume method to 3D grids composed of general hexahedral cells with nonplanar surfaces, namely, the high-order accurate computation of the surface and volume integrals arising in the derivation of the scheme, is discussed in Sect. 2.2.

In order to control spurious oscillations at discontinuities, we use the CENO monotonicity procedure that was introduced by Ivan and Groth [17] for the 2D Euler equations, and has since been extended to the Navier-Stokes equations [16] and MHD [34] in 2D. The CENO procedure switches between an unlimited high-order accurate reconstruction (piecewise cubic in this paper, leading to a fourth-order accurate scheme) and a limited piecewise-linear reconstruction (second-order accurate), with the switching based on the smoothness indicator introduced in [17]. The smoothness indicator is computed in each cell for every reconstructed variable to determine whether the flow is locally smooth and well-resolved. For cells containing non-smooth or under-resolved solution content, the unlimited K-exact reconstruction is switched to limited piecewise linear reconstruction. It should be emphasized that local switching to lower order is only performed for those reconstructed variables that are deemed non-smooth or under-resolved.

Limiting for the second-order method is performed using the procedure developed by Park et al. [28] specifically for multiple dimensions in conjunction with the slope limiter function of [39]. The limiting is applied at the corners of the hexahedral cells and not at the quadrature points for the faces. Apart from monotonicity, this limiting strategy offers significant computational advantages by reducing, at least by a factor of three, the number of reconstruction evaluations required for the calculation of the limiter.

Full details on the CENO smoothness indicator and switching mechanism are described in the 2D context in [16, 17], and the generalization of the relevant formulas to 3D is presented in [19].

2.2 Application of High-Order CENO Method to 3D Cubed-Sphere Grids

A specific technical difficulty in obtaining high-order accuracy on 3D cubed-sphere grids is to properly handle the nonplanar cell faces of the cubed-sphere grid cells. We use a trilinear representation [11, 35, 42] for the cell domain which can be exploited to compute sufficiently accurate the volume and surface integrals arising in the formulation of the high-order scheme. Given a hexahedral cell in physical space a trilinear mapping from a reference unit cube can be defined as $\vec{X}(p,q,s)=\vec{A}+\vec{B}p+\vec{C}q+\vec{D}s+\vec{E}pq+\vec{F}ps+\vec{G}qs+\vec{H}pqs$, where p,q,s are the coordinates in the reference domain and the vector coefficients, $\vec{A}-\vec{H}$, are determined such that the vertices in physical space of the hexahedral cell are obtained for $(p,q,s)\in\{0,1\}\times\{0,1\}\times\{0,1\}$ (see, e.g., [42] for details).

To evaluate the volumetric integral $I=\iiint_{\mathcal{V}_{ijk}} g(\vec{X})\,\mathrm{dv}$, with $g(\vec{X})$ continuous smooth function, the variables and integration domain are changed to those of the reference cube by making use of the trilinear transformation, $\vec{X}=\vec{X}(p,q,s)$, and its transformation Jacobian determinant, $\det\mathbf{J}$, [42]. Thus, the volumetric integral, I, is calculated in the canonical space (p,q,s) as

$$\begin{aligned} I&=\int_0^1\int_0^1\int_0^1 g(\vec{X}(p,q,s))\det\mathbf{J}\,\mathrm{dp}\,\mathrm{dq}\,\mathrm{ds}\\ &\simeq\sum_{m=1}^{N_v} g\left(\vec{X}(p_m,q_m,s_m)\right)(\det\mathbf{J})_m\,\omega_m=\sum_{m=1}^{N_v} g(\vec{X}_m)\,\tilde{\omega}_m\,, \end{aligned} \tag{11}$$

where N_v is the number of volumetric Gauss points. For example, a fifth-order Gaussian integration rule uses $N_v=27$ volumetric Gauss points (i.e., a $3\times3\times3$ product rule [11]), and it is sufficiently accurate to integrate exactly polynomials up to order five. The basic abscissae and weights generated for a $[0,1]$ domain for a given accuracy order can be determined as described in [30]. Note that the Gaussian abscissae $\vec{X}_m$ and weights $\tilde{\omega}_m = (\det \mathbf{J})_m \, \omega_m$ are needed for the initial computation of volumetric integrals of various quantities but need not be stored during the actual simulation, unless motivated by other requirements such as the integration of time-dependent volumetric source terms. In case of AMR grids, we have also made use of these volumetric Gaussian points to transfer efficiently the solution content with high-order accuracy at mesh resolution changes, as described in more details in Sect. 3.

Similarly, a surface integral along a surface with constant p-coordinate, $\mathcal{A}_p$, is evaluated as follows:

$$I_p = \iint_{\mathcal{A}_p} g(\vec{X})\,\mathrm{da} = \int_0^1\!\!\int_0^1 g(\vec{X}(p,q,s))\,\det \mathbf{J}_p \,\mathrm{d}q\,\mathrm{d}s$$
$$\simeq \sum_{m=1}^{N_g} g(\vec{X}_m)\,\tilde{\omega}_m\,, \qquad (12)$$

and the expressions for integrals along surfaces $\mathcal{A}_q$ and $\mathcal{A}_s$ follow by cyclic permutation. Here, N_g is the number of Gauss points on the surfaces, which for a fourth-order scheme is $N_g = 4$ on each cell face. The $\vec{X}_m$ and $\tilde{\omega}_m = (\det \mathbf{J}_p)_m \, \omega_m$ for faces are stored in our framework to increase the computational performance.

A second major technical challenge that must be addressed in applying high-order CENO to cubed-sphere grids, as well as other degenerate grid topologies, is how to deal with reconstruction stencils at degenerate block edges and corners, i.e., block edges and corners where the number of neighbouring blocks is smaller than in a grid with Cartesian topology, and where fewer ghost cells are available. Near these regions the numerical scheme must adapt to deviations from the regular mesh topology and find efficiently the right neighbours to be included in the supporting reconstruction stencil. This challenge is addressed in a way that is general for a variety of stencils and block configurations by requiring consistency in a double sense: first, the number of cells dropped in the degenerate stencil needs to be as small as possible compared to the base stencil while maintaining the same stencil structure in terms of connectivity to neighbouring cells; and second, degenerate stencils generated for ghost cells need to be identical to the degenerate stencils generated for the corresponding physical cells in neighbouring blocks. It is important to formulate a general strategy that meets these requirements in the complex degenerate 3D topology with missing ghost cells. We have proposed a general rotation mechanism to automatically build consistent high-order stencils at these degenerate locations on the cubed-sphere grid, the detailed description of which is provided in [19], and this mechanism is used herein. The proposed mechanism is sufficiently general to also handle the case of a cubed-sphere grid with an additional grid block filling the interior [5, 7], where blocks may have corners where three degenerate edges meet.

3 HIGH-ORDER CENO WITH AMR

The high-order 3D CENO scheme is implemented in a parallel, solution-adaptive, block-based AMR simulation framework originally formulated by Groth and coworkers for second-order, finite-volume schemes on 3D hexahedral multi-block structured grids [12, 27]. The AMR scheme has been extended and optimized for cubed-sphere grid topologies in [18]. In the current work, the extension to high-order accuracy is considered by following closely the approach proposed in 2D by Ivan and Groth [16, 17], and which is generalized here to adaptive 3D hexahedral multi-block structured grids. As in the lower-order approach, a flexible block-based hierarchical data structure is used in combination with the CENO scheme to facilitate automatic solution-directed mesh adaptation on 3D cubed-sphere grids. The parallel implementation is achieved via domain decomposition. Solution information is shared between adjacent blocks having common interfaces using additional layers of overlapping “ghost” cells. The high-order variant requires inter-block communication of high-order solution content and high-order solution transfer between AMR grids, which is achieved in this work by deriving appropriate multigrid-type restriction and prolongation operators.

In the case of refinement, each of the eight octants of a parent block becomes a new block having the same number of cells as the parent, thereby doubling the cell resolution in the region of interest. The additional mesh nodes in the new fine blocks are inserted in the middle between the nodes inherited from the parent, except at the spherical boundaries where the new nodes are forced to conform with the geometry. This process can be reversed in regions that are deemed over-resolved and eight children are coarsened into a single parent block, the mesh nodes of which are obtained by eliminating every other interior node from the children blocks. Local refinement and coarsening of the mesh is accomplished by refining and coarsening appropriate solution blocks, as directed according to smoothness-based refinement criteria [16,17]. Refer to [12,18,27] for further details regarding the mechanics of mesh adaption on 3D cubed-sphere grids.

We focus now on the restriction and prolongation operators used to evaluate the high-order solution content on AMR grids. A coarse cell has mean solution data calculated from the corresponding eight fine cells it overlaps with in a way that satisfies conservation [4, 16, 25]. In particular, each mean solution variable of the coarse (parent) grid cell is determined by summing the corresponding eight mean solution variables of the fine grid cells weighted by the fraction of the fine grid cell volume to the coarse grid cell volume. The high-order prolongation from coarse to fine cells is accomplished by direct integration of the coarse-cell polynomial reconstructions, determined in terms of primitive variables here, over the domain of each newly created fine cell. However, the computational cost associated with integrating the polynomial reconstructions can be high, especially for hexahedral cells described with trilinear mapping, and unless carefully addressed, it may affect the parallel performance of the algorithm as only a fraction of the solution blocks execute this operation. Next paragraph briefly describes how we address this issue. Finally, the average conserved solutions of fine cells are determined so as to conserve the solution content of the parent, as in [16].

To integrate efficiently the polynomial reconstructions of the coarse-cell over the domain of fine cells, we propose to use a quadrature rule similar to Eq. 11, in which the integrant values at the Gaussian integration points of the coarse cell are re-used with different weights for each fine cell. The weights for each octant have been derived in our work based on basic weights generated for a $[0,1]$ domain. Thus, considering the abscissae of a basic Gaussian rule with n_g points, we determine the weights individually for each of the two subinterval, $[0,0.5]$ and $[0.5,1]$, by writing sufficient conditions to integrate polynomials up to order n_g-1 exactly. For example, to obtain a quadrature rule to integrate exactly cubic polynomials on basic subintervals, we evaluate the integrant on $[0,1]$ at the Gaussian abscissae corresponding to $n_g=4$, and then use the appropriate weights for each subinterval. Obviously, the number of integrant evaluations, four in this case, is the same as that required for optimal Gaussian quadrature on each subinterval. However, computational advantages are gained when our strategy is applied to general hexahedral cells, and when avoiding the cost associated with computing the data required by Eq. 11 for eight trilinear mappings is factored in. Thus, the fourth-order integration rule in this work uses only the 64 points, i.e., $4\times4\times4$ rule, of the coarse cell.

4 Numerical Results

To demonstrate current capabilities of the proposed solution-adaptive 3D CENO method, we discuss the 3D iso-density MHD vortex advection problem from [26]. This is a smooth time-dependent 3D test problem with an exact solution, consisting of a magnetized vortex structure in force equilibrium that is advected by a uniform flow field with a background velocity that equals $(1,1,2)$. Our numerical studies include both fixed and dynamically-refined meshes.

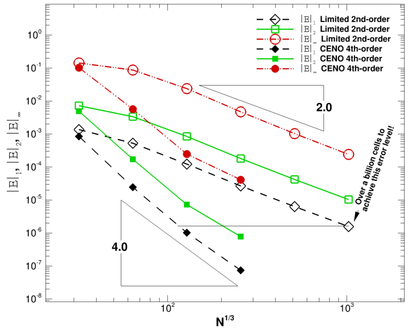
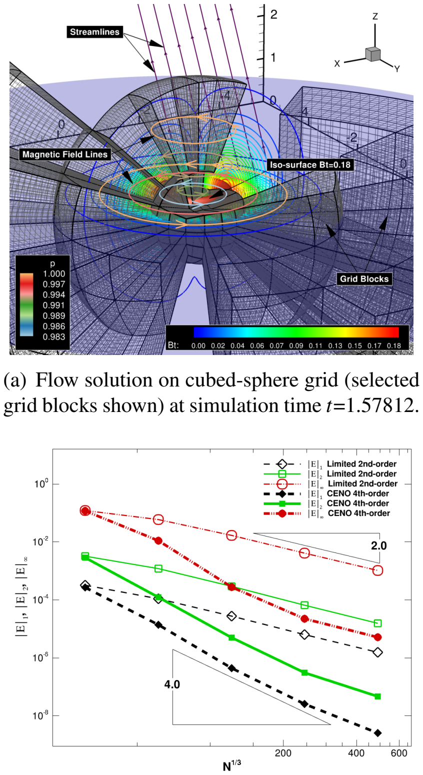


Figure 3: Error norms versus grid size for the iso-density vortex problem solved in a periodic box.

We first simulate this flow problem on a periodic Cartesian box with x, y, and z varying between [-5,5]. Initially the vortex centre is positioned at $(0,0,0)$ and the simulation runs until time t=10. We use $\mathcal{S}_C = 1500$ for the smoothness indicator cut-off value, use the HLLE flux function, and use $\gamma = 5/3$. Figure 3 demonstrates fourth-order convergence. The initial grid consists of 64 blocks with $8\times8\times8$ cells, corresponding to a total of 32,768 cells. The largest high-order simulation used 512 blocks with $8\times8\times8$ cells, corresponding to a total of 16,777,216 cells, while the largest low-order simulation used 4,096 blocks with $64\times64\times64$ cells corresponding to 1,073,741,824 cells, on 4,096 CPU cores. Figure 3 also illustrates the computational gains that can be made by using a fourth-order accurate numerical method as compared to a second-order method: for the L_1 error level indicated in the figure by the horizontal line, the (standard) second-order method requires 1,073,741,824 computational cells, while the fourth-order method would require 1,507,188 cells. In terms of the total execution time, the second-order method requires a total execution time (in core hours) that is greater by a factor of about 700 than the fourth-order method to achieve the same error, and this becomes larger as the required accuracy increases. While in general efficiency gains will depend on the problem and the problem size, this example clearly demonstrates the potential efficiency gains that can be offered by the high-order 3D CENO scheme.

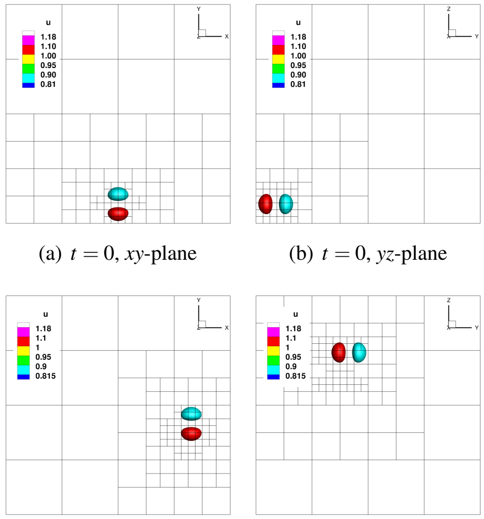
The simulation of 3D iso-density MHD vortex advection on a cubed-sphere grid with an additional root block filling the interior of the inner sphere is now considered. The radius of the outer and inner spheres are $R = 9$ and $R = 1$, respectively, and the inner sphere is



(a) Flow solution on cubed-sphere grid (selected grid blocks shown) at simulation time t=1.57812.

(b) Error norms versus grid size.

Figure 4: Iso-density vortex problem solved on a cubed-sphere grid with an additional seventh root block filling the interior sphere.



(a) $t=0$, xy-plane

(b) $t=0$, yz-plane

(c) $t=6$, xy-plane

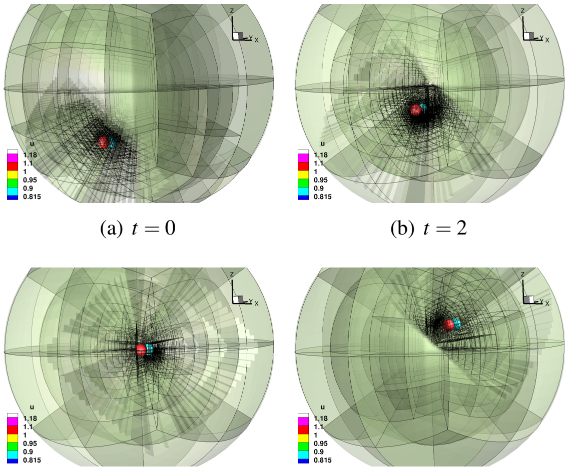
(d) $t=6$, yz-plane

Figure 5: Initial ($t=0$) and final ($t=6$) adapted Cartesian grids for solving the iso-density vortex problem.

filled by a seventh root block with Cartesian topology. In this problem the 3D vortex is initially located outside the inner sphere (centred at (-2, -2.5, -3)), and is then translated with velocity (1,1,2) over the cubed-sphere grid, passing through the inner sphere. The simulation is performed until $t = 3$, when the centre of the vortex is located at (1, 0.5, 3). Figure 4(a) shows the flow solution on the cubed-sphere grid at $t = 1.57812$, when the vortex passes through a corner of the seventh root block. The initial grid consists of 7 blocks with $16\times16\times16$ cells, corresponding to a total of 28,672 cells The final grid uses 3,584 blocks of $32\times32\times32$ cells corresponding to a total of 117,440,512 cells on 1,792 CPU cores. Figure 4(b) demonstrates that this simulation achieves fourth-order accuracy for a time-dependent flow on the cubed-sphere grid, and as such confirms the validity of our approaches for achieving high-order accuracy on grids with nonplanar cell surfaces and using a rotation mechanism to determine degenerate stencils at block boundaries and corners with unstructured grid topology. The performance gains in this case are comparable to those achieved on Cartesian meshes.

The capability of the high-order scheme is now examined in conjunction with AMR. In this case, the iso-density vortex problem is again considered, but the translation of the vortex is carried out in a larger computational domain than in the case of fixed grids. Thus, this test examines the capability of the high-order AMR framework to resolve small-scale solution features with dynamically refined AMR meshes while minimizing grid requirements. In this work, the AMR is directed based on the solution smoothness indicator for the components of the velocity vector.

The first AMR simulation is performed in the Cartesian box with x, y, and z varying between [-8.5,8.5] and with the vortex center initially located at $(0,-7,-7)$. The simulation runs until time t=6, when the vortex center is located at (6,-1,5). The initial mesh consists of 8 blocks with $8\times8\times8$ cells and 4,096 cells in total. Several iterations of AMR are used initially to adapt the mesh so as to capture the initial condition accurately. Thus, the starting mesh for the dynamic simulation has 428 blocks and 219,136 cells, and is shown in Fig. 5(a)-(b). The same figure depicts the iso-surfaces of x-direction velocity $u = 0.9$ and $u = 1.1$, which are also plotted on the final adapted grid with 561 blocks and 287,232 cells in Fig. 5(c)-(d). Over the course of the simulation 596 explicit time steps are perfomed and the AMR is applied for every 50 steps. Note that no attempt has been made in this work to optimize the parameters driving the mesh adaption, which will be addressed in future research along with a quantitative assessment of the computational performance. The results in Fig. 5 demonstrate the ability of the high-order AMR algorithm to adapt dynamically



(a) $t = 0$ (b) $t = 2$

(c) $t = 3.5$ (d) $t = 5$

Figure 6: Sequence of adapted cubed-sphere grids for solving the iso-density vortex problem.

the computational grid to maintain the desired resolution for small-scale solution features.

Finally, the capabilities of the fourth-order solution-adaptive CENO method are demonstrated for adaptive cubed-sphere grids. In this case, the computational domain is the cubed-sphere grid with the inner and outer spheres of radius $R = 1$ and $R = 17$, respectively. The iso-density vortex is positioned at (0,-7,-7) initially, and the final time is $t = 5$. Similarly, the translational velocity is (1,1,2). Grid blocks with $8 \times 8 \times 10$ cells have been used in this study. The initial cubed-sphere grid contains 18 blocks arranged in three radial layers, each of which with six cubed-sphere root blocks. The adapted grid shown in Fig. 6(a) contains 1,334 blocks and 853,760 cells, and has been obtained after six AMR iterations based on the initial condition. Dynamic AMR has been applied every 50 steps for a total of 50 times. Figures 6(b)-6(d) depict the sequence of adaptively refined AMR mesh corresponding to $t = 2$, $t = 3.5$ and $t = 5$. The former mesh has 2,195 blocks and 1,404,800 cells, whereas both latter meshes consist of 886 blocks and 567,040 cells. As in the previous study, the same iso-surfaces are depicted in Fig. 6. Obviously, the results show that high-order CENO with dynamic AMR has the potential to provide significant computational savings on cubed-sphere grids by reducing grid requirements to resolve solutions.

Concluding Remarks

A high-order 3D CENO finite-volume scheme with AMR has been proposed for hyperbolic conservation laws on cubed-sphere grids. Future research will involve further studies and more complex applications.

References

[1] D. S. Balsara. *J. Comput. Phys.*, 231(22):7504 – 7517, 2012.

[2] D. S. Balsara, T. Rumpf, M. Dumbser, and C.-D. Munz. *J. Comput. Phys.*, 228:2480–2516, 2009.

[3] T. J. Barth. Paper 93-0668, AIAA, January 1993.

[4] M. J. Berger and P. Colella. *J. Comput. Phys.*, 82:67–84, 1989.

[5] D. A. Calhoun, C. Helzel, and R. J. LeVeque. *SIAM Review*, 50:723–752, 2008.

[6] B. Cockburn and C.-W. Shu. *J. Comput. Phys.*, 141(2):199–224, 1998.

[7] P. Colella, M. Dorr, J. Hittinger, P. McCorquodale, and D. F. Martin. Paper, ASTRONUM-2008, 2008.

[8] P. Colella, M. Dorr, J. Hittinger, and D. Martin. *J. Comput. Phys.*, 230(8):2952 – 2976, 2011.

[9] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, and M. Wesenberg. *J. Comput. Phys.*, 175(2):645 – 673, 2002.

[10] M. Dumbser, D. S. Balsara, E. F. Toro, and C.-D. Munz. *J. Comput. Phys.*, 227(18):8209–8253, 2008.

[11] C. A. Felippa. *Eng. Comput.*, 21(8):867–890, 2004.

[12] X. Gao and C. P. T. Groth. *J. Comput. Phys.*, 229:3250–3275, 2010.

[13] T. I. Gombosi, K. G. Powell, D. L. D. Zeeuw, C. R. Clauer, K. C. Hansen, W. B. Manchester, A. J. Ridley, I. I. Roussev, I. V. Sokolov, Q. F. Stout, and G. Tóth. *Comp. Sci. & Eng.*, 6(2):14–35, 2004.

[14] C. P. T. Groth, D. L. De Zeeuw, T. I. Gombosi, and K. G. Powell. *J. Geophys. Res.*, 105(A11):25,053–25,078, 2000.

[15] C. P. T. Groth, D. L. D. Zeeuw, K. G. Powell, T. I. Gombosi, and Q. F. Stout. Paper 99-3273, AIAA, June 1999.

[16] L. Ivan and C. P. Groth. *J. Comput. Phys.*, 257, Part A(0):830 – 862, 2014.

[17] L. Ivan and C. P. T. Groth. Paper 2007-4323, AIAA, June 2007.

[18] L. Ivan, H. D. Sterck, S. A. Northrup, and C. P. Groth. *J. Comput. Phys.*, 255(0):205 – 227, 2013.

[19] L. Ivan, H. D. Sterck, A. Susanto, and C. P. T. Groth. *J. Comput. Phys.*, Submitted, 2014.

[20] C. Jablonowski, M. Herzog, J. E. Penner, R. C. Oehmke, Q. F. Stout, B. van Leer, and K. G. Powell. *Mon. Wea. Rev.*, 134:3691–3713, 2006.

[21] R. Keppens, Z. Meliani, A. J. van Marle, P. Delmont, A. Vlasis, and B. van der Holst. *J. Comput. Phys.*, 231(3):718–744, 2012.

[22] J. Kleimann, A. Kopp, H. Fichtner, R. Grauer, and K. Germaschewski. *Comput. Phys. Commun.*, 158:47–56, 2004.

[23] C. Lawson and R. Hanson. Prentice-Hall, INC, 1974.

[24] H. Luo, Y. Xia, S. Li, R. Nourgaliev, and C. Cai. *J. Comput. Phys.*, 231(16):5489–5503, 2012.

[25] D. J. Mavriplis. Report 95-27, ICASE, April 1995.

[26] A. Mignone, P. Tzeferacos, and G. Bodo. *J. Comput. Phys.*, 229:5896–5920, 2010.

[27] S. A. Northrup and C. P. T. Groth. In *Proceedings of the 6th U.S. National Combustion Meeting*, May 2009.

[28] J. S. Park, S.-H. Yoon, and C. Kim. *J. Comput. Phys.*, 229:788–812, 2010.

[29] K. G. Powell, P. L. Roe, T. J. Linde, T. I. Gombosi, and D. L. De Zeeuw. *J. Comput. Phys.*, 154:284–309, 1999.

[30] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Cambridge University Press, New York, 2007.

[31] C. Ronchi, R. Iacono, and P. S. Paolucci. *J. Comput. Phys.*, 124:93–114, 1996.

[32] A. St-Cyr, C. Jablonowski, J. M. Dennis, H. M. Tufo, and S. J. Thomas. *Mon. Weather Rev.*, 136:1898–1922, 2008.

[33] Q. F. Stout, D. L. De Zeeuw, T. I. Gombosi, C. P. T. Groth, H. G. Marshall, and K. G. Powell. In *Proceedings of SC97, San Jose, California, U.S.A., November 12–15, 1997*, 1997.

[34] A. Susanto, L. Ivan, H. D. Sterck, and C. Groth. *J. Comput. Phys.*, 250(0):141 – 164, 2013.

[35] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. North-Holland, New York, 1985.

[36] G. Tóth, B. van der Holst, I. V. Sokolov, D. L. D. Zeeuw, T. I. Gombosi, F. Fang, W. B. Manchester, X. Meng, D. Najib, K. G. Powell, Q. F. Stout, A. Glocer, Y.-J. Ma, and M. Opher. *J. Comput. Phys.*, 231(3):870 – 903, 2012. doi:10.1016/j.jcp.2011.02.006.

[37] B. van der Holst, W. M. IV, I. Sokolov, G. Tth, T. Gombosi, D. DeZeeuw, and O. Cohen. *APJ*, 693(2):1178, 2009.

[38] B. van der Holst and R. Keppens. *J. Comput. Phys.*, 226(1):925 – 946, 2007.

[39] V. Venkatakrishnan. Paper 93-0880, AIAA, January 1993.

[40] T. C. Warburton and G. E. Karniadakis. *J. Comput. Phys.*, 152:608–641, 1999.

[41] M. J. Williamschen and C. P. T. Groth. Paper 2013-2442, AIAA, June 2013.

[42] O. Zienkiewicz and R. Taylor. Butterworth-Heinemann, Fifth edition, 2000.