Hyperbolic Conservation Laws on Three-Dimensional Cubed-Sphere Grids: A Parallel Solution-Adaptive Simulation Framework

Lucian Ivan[*,a], Hans De Sterck[a], Scott A. Northrup[b], Clinton P. T. Groth[b]

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

Abstract

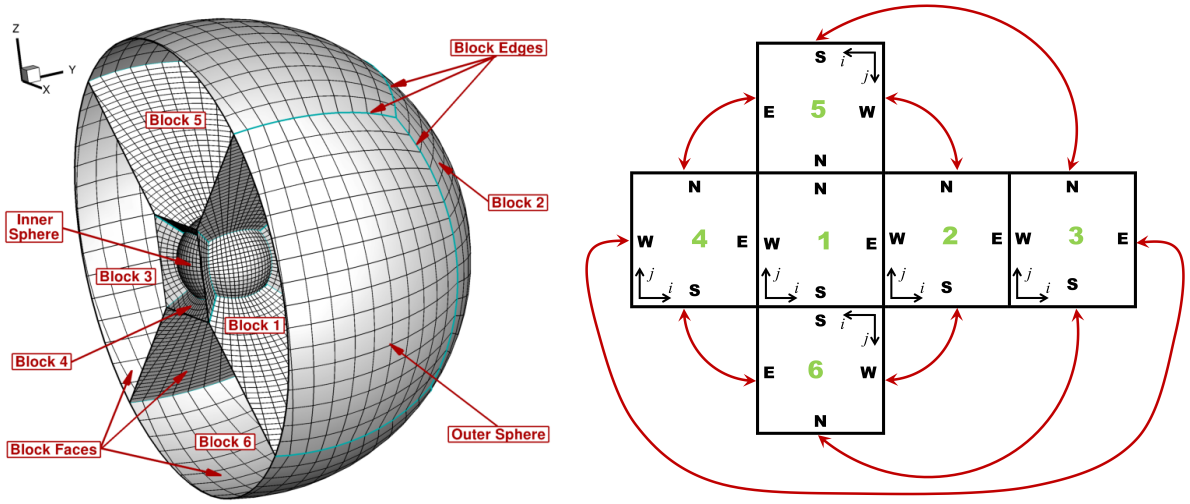
An accurate, efficient and scalable parallel, cubed-sphere grid numerical framework is described for solution of hyperbolic conservation laws in domains between two concentric spheres. The particular conservation laws considered in this work are the well-known Euler and ideal magnetohydrodynamics (MHD) equations. Our main contribution compared to existing cubed-sphere-based algorithms lies in the design of a cubed-sphere framework that is based on a genuine and consistent multi-block implementation, leading to flux calculations, adaptivity and parallelism that are fully transparent to the boundaries between the six sectors of the cubed-sphere grid. This results in the first fully adaptive three-dimensional cubed-sphere grid framework, with excellent parallel scalability on thousands of computing cores. Crucial elements of the proposed approach are: unstructured connectivity of the six grid root blocks that correspond to the six sectors of the cubed-sphere grid, multi-dimensional k-exact reconstruction that automatically takes into account information from neighbouring cells isotropically and is able to automatically handle varying stencil size, and adaptive division of the solution blocks into smaller blocks of varying spatial resolution that are all treated exactly equally for inter-block communication, flux calculation, adaptivity and parallelization. A second-order Godunov-type finite-volume scheme is employed to discretize the governing equations on the multi-block hexahedral mesh defining the cubed-sphere grid. An overview of the algorithm parallelization and of the flexible block-based adaptive mesh refinement (AMR) procedure used for performing automatic solution-directed mesh adaptation on the three-dimensional cubed-sphere grid is provided. A new fully supersonic axi-symmetric MHD test problem is presented that has an exact solution in a domain between two concentric spheres, and can thus be used to test the order of convergence of MHD solvers in this type of domains. Numerical results for several problems, including systematic grid convergence studies, MHD bow-shock flows, and global modelling of solar wind flow are discussed to demonstrate the accuracy and efficiency of the proposed solution procedure, along with measurements of parallel computing scalability up to thousands of computing cores.

Key words:
Magnetohydrodynamics (MHD), Cubed-sphere grid, Space-physics flows, Numerical approximation, Adaptive mesh refinement (AMR).
PACS: 47.11.Df, 47.40.-x, 52.65.Kj, 96.50.Ci

1. Introduction

High-performance computational methods for numerically solving conservation laws in domains between two concentric spheres are desirable in many fields of computational physics. Despite representing very specialized geometries with relatively low complexity, spherical shell domains are, for obvious reasons, ubiquitously present in computations of global physical processes associated with geophysical and celestial bodies, and the intervening space. Consequently, the discretization of spherical shell geometries is required, at least as a first approximation, in fields as

[*]Corresponding author
Email addresses: livan@uwaterloo.ca (Lucian Ivan), hdesterck@uwaterloo.ca (Hans De Sterck), northrup@utias.utoronto.ca (Scott A. Northrup), groth@utias.utoronto.ca (Clinton P. T. Groth)



(a) Cross-section of the cubed-sphere grid

(b) Illustration of connectivity among root blocks

Figure 1: Three-dimensional cubed-sphere grid with six root blocks (corresponding to the six sectors of the grid) and depiction of inter-block connectivity. In our approach, the root blocks can be refined in a block-adaptive way. In panel (b), the block faces are denoted with the initials of the cardinal directions: North (N), East (E), South (S) and West (W).

diverse as space physics, astrophysics, climate and weather modelling, and geophysics, and the choice of the grid is important in formulating an efficient and accurate numerical method for solving partial differential equations on such domains. For example, for space-physics problems, one of the main numerical challenges is provided by the presence of a wide variety of temporal and spatial scales on which interesting plasma physics phenomena occur throughout the vast domains associated with the large-scale space-weather environment. Numerical solutions of the equations arising in the modelling of these complex flows are computationally intensive and feasible only on massively parallel computers [1–4]. Therefore, scalable high-performance algorithms capable of efficiently resolving the solution features of these flows and of reducing the time required to obtain numerical solutions of these problems are actively researched.

In recent years, cubed-sphere grids have gained increasing popularity for simulating fluid flow in domains between concentric spheres, first in the area of climate and weather modelling [5–9], but more recently also in areas like astrophysics [10, 11]. Cubed-sphere grids (see Fig. 1) are attractive because they offer a nearly uniform covering of the spherical surface, while also providing logically Cartesian grids in each of the six sectors (or panels) of the grid, which can be exploited for efficient implementation. Cubed-sphere grids, however, also pose important numerical challenges due to the grid irregularity and non-trivial connectivity along the boundaries and at the corners of the six sectors of the grid. For these reasons, progress in terms of uniform second-order accuracy, adaptivity, and parallelization has been slow.

This paper proposes a new parallel and adaptive cubed-sphere simulation framework for solving conservation laws. This computational framework is the first fully-adaptive three-dimensional (3D) cubed-sphere grid implementation, it is uniformly second-order accurate and it has excellent parallel scalability on thousands of computing cores. These properties have been obtained by basing the framework on a genuine and consistent multi-block approach, with block-based adaptivity, unstructured root-block connectivity, and least-squares based reconstruction that allows for variable stencil sizes in the upwind finite-volume formulation. This approach results in uniform second-order accuracy, solution adaptivity and parallelization that are fully transparent to the boundaries between the six sectors of the cubed-sphere grid. The resulting scalable, parallel adaptive cubed-sphere grid simulation framework is applied in this work to non-conducting compressible gaseous flow problems and to magnetohydrodynamics (MHD) flows from space physics. Note that the multi-block AMR scheme used here is based on the previous work of Gao and Groth [12, 13] for reacting flows, and has been extended and optimized for cubed-sphere grids in this work. As such, the multi-block AMR finite-volume framework is not restricted to cubed-sphere topologies and more general grid geometries can be

treated with the proposed method.

This paper is organized as follows. The next section provides additional background on cubed-sphere grids, followed by a short section on the governing equations of interest. In Section 4, a detailed overview of the proposed block-adaptive cubed-sphere grid simulation framework for conservation laws is provided. In Section 5, the accuracy of the numerical procedure is demonstrated by comparing with analytical solutions and with highly-accurate one-dimensional (1D) transonic flow solutions. A new axi-symmetric MHD test problem is presented that has an exact solution in a domain between two concentric spheres, and can thus be used to test the order of convergence of MHD solvers in this type of domains. Additionally, numerical results are provided to illustrate the capabilities of the adaptive mesh refinement (AMR) algorithm for magnetized and non-magnetized space-plasma problems, such as spherically-symmetric transonic winds and supersonic bow-shock flows past a sphere. Finally, a more realistic space-physics problem is considered by developing a global MHD model similar to the one proposed by Groth *et al.* [2] for the time-averaged or quasi-steady solar wind at minimum solar activity.

2. Background

The development of meshing techniques for spheres and spherical shell domains has been historically linked to the formulation of efficient and accurate numerical methods in the fields of meteorology and atmospheric dynamics for solving the equations of atmospheric motion on the Earth's surface. Although they are a natural choice, spherical-polar grids raise numerical difficulties associated with the presence of pole singularities and consequently, may negatively impact the accuracy and performance of the numerical procedure [7, 11]. Alternatively, a potentially more efficient implementation can be obtained based on cubed-sphere grids [5]. In a two-dimensional (2D) cubed-sphere grid, the surface of a sphere is represented free of any strong singularities as the projection of a cube onto the spherical surface, which results in six adjoining grid sectors (or panels) that seamlessly cover the whole sphere (see Fig. 1 for a 3D cubed-sphere grid). Although eight weak singularities are created at the projection of the cube corners, the resulting cubed-sphere grid is still far more uniform than a polar grid in terms of shape and size of the cells. The roots of these discretization ideas can be found in the early work of Phillips [14] who proposed the use of three partially overlapping grids to geometrically decompose the spherical surface. Phillips [14] used an interpolation-based method to couple the three conformal coordinate systems, a procedure discarded by Sadourny [15] on grounds that it would make impossible the design of a globally conservative scheme. Alternatively, Sadourny [15] proposed to use central non-conformal projections to map the whole sphere onto a regular polyhedron (e.g., cube, tetrahedral, octahedron, icosahedral) and thus, to decompose the spherical surface into several non-overlapping regions that would require communication only along shared interface edges. Thus, the conservation laws valid on the sphere are mapped and then discretized over a regular grid constructed within each face of the selected polyhedron. To obtain numerical conservation, Sadourny opted for a coupling scheme based on the use of one-sided differencing formulae at the polyhedral face boundaries.

The cubic-gnomonic projection proposed by Sadourny to obtain a cubed-sphere grid by projecting a cube with Cartesian grid faces [15], represented a good compromise among computational efficiency, grid uniformity, and flexibility in designing accurate schemes. Unfortunately, despite providing a framework to solve conservation laws on quasi-uniform grids on the sphere, Sadourny's new solution approach suffered from a decay of accuracy at the internal boundaries (i.e., the boundaries between the six sectors), which seems to have prevented the widespread use of the method for about two decades. More recently, the method has been revived by Ronchi *et al.* [5] who realized that the cubic-gnomonic projection can be used to develop efficient and accurate parallel numerical algorithms for solving *any* conservation law on spherical surfaces by improving Sadourny's procedure to treat the singular sector edges. Crucial to the computational efficiency in the procedure proposed by Ronchi *et al.* was the coupling of the six sectors of the grid using a one-dimensional interpolation technique which exploits the symmetric nature of the cubed-sphere mesh in order to reduce the number of coupling operations at inter-sector boundaries and recover the accuracy of the interior scheme. Note that the cubed-sphere grids used in this paper are also generated using the cubic-gnomonic projection [5, 7, 15]. In particular, the angularly equidistant mapping described in [5] is used to generate the initial six blocks of the grids.

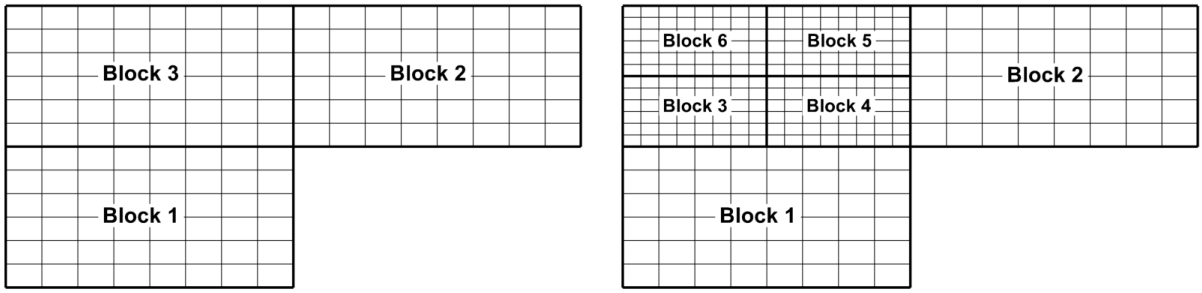
In the same fashion, a three-dimensional cubed-sphere grid can be obtained by overlaying a sequence of concentric 2D spherical shell grids in the radial direction and forming six three-dimensional blocks, each of which is enclosed by the union of four radial and two spherical faces (see Fig. 1). In contrast to the 2D counterpart, which requires

the definition of a curved coordinate system for each of the six cubed-sphere sectors, the 3D cubed-sphere grid in principle allows the use of a unique coordinate system (e.g., Cartesian) to discretize the governing conservation laws everywhere in the physical domain which makes unnecessary the usage of a covariant transformation [5, 9, 15, 16] to map vector fields from the curved coordinate system to the Cartesian system of a computational domain. However, in this approach the handling of the cubed-sphere discretization in the 3D physical space requires the numerical scheme to be applied on general hexahedral elements instead of on Cartesian cells. Note also that, due to the large discrepancies between horizontal and vertical scales, 3D weather and climate models are formulated as layered models, and thus will still require the transformation of information between different coordinate systems of the cubed-sphere grid. However, the space physics flows we target are suitable for a fully-coupled 3D approach, which means that the complication of having to deal with multiple coordinate systems is not required and can be completely avoided.

Other discretization approaches of 3D spherical shell geometries are also possible either based on regular Cartesian grids with embedded approximately-spherical boundaries [2, 17] or based on completely unstructured mesh connectivity. Examples of these approaches are Cartesian cut-cell methods [18] and geodesic grids generated with different tessellation elements such as the icosahedron [19]. Of course, the cubed-sphere grid itself could also be implemented as an unstructured mesh of hexahedral elements instead of a structured grid approach. However, the cubed-sphere discretization implemented using structured grids has an obvious advantage over other methods in that it not only allows the straightforward generation of quasi-uniform tessellations of the sphere, but it also avoids numerical complexities and possible inefficiencies raised by unstructured discretization techniques, such as determining computational cell connectivity and irregular access to data in memory. Hence, the logically Cartesian representation of the data structure within each sector of the cubed-sphere grid allows for efficient implementation, and optimizations can be performed based on proximate data residing in cache.

Cubed-sphere grids have already been successfully considered for applications to weather and climate modelling [6, 7, 20] as well as to computational astrophysics [10, 11]. Recently, Ullrich *et al.* [9] have proposed a 2D fourth-order finite-volume formulation and applied it to shallow-water equations and atmospheric modelling on the sphere. Yang *et al.* [21] have developed a fully implicit Newton-Krylov finite-volume algorithm for 2D shallow-water equations on the cubed sphere and carried out parallel performance studies with up to 2,048 processing cores. Although local enhancement of flow features has already been widely considered in the context of *h*-adaptation on 2D latitude-longitude spherical grids by, e.g., Jablonowski *et al.* [22], and more recently on 2D cubed-sphere grids by St-Cyr *et al.* [8], to our knowledge, a fully 3D adaptive mesh refinement procedure on cubed-sphere grids has not been described in the literature before, and the present study represents the first application of a block-based AMR algorithm to 3D cubed-sphere grids. 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 flows [2, 17, 23–26]. Computational grids that automatically adapt to the solution of the governing equations are effective in treating problems with disparate length scales, providing the required spatial resolution while minimizing memory and storage requirements. A recent review of the impact of AMR on space-physics simulations, numerical astrophysics and computational cosmology is given by Norman [27].

An important aspect in the development of numerical algorithms on cubed-sphere grids is related to the treatment of interfaces between adjacent grid sectors, across which grid lines have slope discontinuities [5, 9, 11, 15, 16]. If special care is not taken, the accuracy of the numerical discretizations that are based on one-dimensional approaches may be significantly degraded at these interfaces, and in traditional implementations of cubed-sphere grid codes the special treatment required at these interfaces for maintaining adequate accuracy poses difficulties for transparent implementation of AMR and parallelization. The large-scale parallel domain partitioning of 3D cubed-sphere meshes is typically obtained by generating cuts in the radial direction, thereby increasing the number of partitioning blocks that can be farmed to different processors to $6 \times N_c$, where N_c is the number of radial blocks in each sector. For 2D spherical geometries Putman and Lin [7] and Putman [28] used a discretization with a number of patches per grid sector. Several approaches have been proposed in the literature for coupling the solution blocks, including direct information transfer from interior cells of neighbouring blocks to layers of overlapping "ghost" (or halo) cells, one-dimensional interpolation known as *cascade interpolation* which makes use of some particular features of the cubed-sphere grid [5], as well as a one-sided reconstruction followed by high-order Gaussian quadrature to determine the average solution data within each ghost element [9]. It should be obvious that within the framework of Godunov-type finite-volume methods the particular choice of interface treatment is ultimately related to how accurate and robust the



(a) A multi-block grid with three blocks of equal resolution.

(b) The same domain as shown in (a), but now with a multi-block grid that has seven blocks of different resolution. This grid can be obtained by block-adaptive refinement of the grid in (a).

Figure 2: Examples of two-dimensional multi-block grids.

reconstruction procedure is to the presence of grid irregularities, in particular also at interfaces on sector boundaries.

In this paper, we present a novel approach for designing 3D cubed-sphere grid simulation frameworks that is based on a genuine multi-block implementation and follows the previous work of Gao and Groth [12, 13] for reacting flows. The approach requires significant initial investment in developing a general and sophisticated adaptive multi-block implementation, with the added complexity of unstructured root-block connectivity, but once this infrastructure is in place an adaptive cubed-sphere simulation framework that is uniformly accurate and highly scalable can be derived naturally. Multi-block grids (see Fig. 2) are grids composed of a number of adjacent logically Cartesian blocks, and are often used in engineering applications because of their improved flexibility in meshing complex geometries, while maintaining a logically Cartesian grid topology in each block, which is beneficial for efficient implementation of numerical schemes. Multi-block frameworks are often capable of handling blocks with unequal resolution, as illustrated in Fig. 2(b). Multi-block computational frameworks are normally implemented with carefully designed transparent mechanisms for flux calculations between neighbouring blocks (which may have different resolution), with automatic mechanisms to keep track of neighbouring blocks, and with transparent mechanisms for adaptation and parallelization. Such frameworks are not so common in computational physics applications, where complex geometries are often not required. We will demonstrate, however, that a consistent, genuine multi-block implementation offers very important advantages for developing 3D cubed-sphere-based algorithms, and this approach will be used to implement the 3D cubed-sphere hyperbolic conservation law simulation framework for the space physics applications presented in this paper. One of the primary non-standard technical elements that has permitted developing the proposed simulation framework for cubed-sphere grids is the use of unstructured connectivity between root blocks. This can be seen by noting in Fig. 1(a) that at the projection of the cube corners (i.e., where the block edges intersect) three root blocks meet, and as a result, the root-block connectivity is as depicted in Fig. 1(b). The connectivity of the root blocks is termed unstructured because it cannot be mapped to a logically Cartesian mesh. The multi-block AMR scheme proposed earlier by Gao and Groth for reacting flows [12, 13] allows for unstructured root-block connectivity and is therefore well-suited for handling the cubed-sphere grid topology. A second important ingredient in our approach is the use of a reconstruction mechanism to obtain second-order accuracy that is multi-dimensional and can handle stencils of various sizes automatically, which is important at sector corners. In particular, the least-squares-based multi-dimensional limited k-exact reconstruction proposed by Barth [29] is used. With the important mechanisms of unstructured block connectivity and flexible second-order accurate reconstruction in place, we then develop dynamic block-based adaptivity and parallelization that is transparent to the boundaries between the six grid sectors of the cubed-sphere grid. While the unstructured root-block connectivity and multi-dimensional reconstruction come with additional costs, as will be demonstrated, the benefits are significant.

3. Governing Equations

The solution of hyperbolic systems of conservation laws which may be expressed in conservation (or divergence) form

$$\partial_t \mathbf{U} + \vec{\nabla} \cdot \vec{\mathbf{F}} = \mathbf{S} + \mathbf{Q}\,, \tag{1}$$

is considered here, where $\mathbf{U}$ is the conserved variable solution vector, $\vec{\mathbf{F}}$ is the system flux dyad, and $\mathbf{S}$ and $\mathbf{Q}$ are volumetric source terms. The numerical solution of two prototypical hyperbolic systems will be considered: the three-dimensional forms of the Euler and ideal MHD equations for inviscid compressible non-conducting and conducting ideal gases, respectively. The conserved variable solution vector, $\mathbf{U}$, for these cases has the form

$$\mathbf{U} = \left[\begin{array}{cccc} \rho, & \rho\vec{V}, & \vec{B}, & \rho e \end{array} \right]^T , \tag{2}$$

where ρ is the gas density, $\vec{V} = (V_x, V_y, V_z)$ is the velocity, $\vec{B} = (B_x, B_y, B_z)$ is the magnetic field and ρe is the total energy. The flux dyad, $\vec{\mathbf{F}}$, is given by

$$\vec{\mathbf{F}} = \left[\begin{array}{c} \rho\vec{V} \\ \rho\vec{V}\vec{V} + (p + \dfrac{\vec{B}\cdot\vec{B}}{2})\vec{\vec{I}} - \vec{B}\vec{B} \\ \vec{V}\vec{B} - \vec{B}\vec{V} \\ (\rho e + p + \dfrac{\vec{B}\cdot\vec{B}}{2})\vec{V} - (\vec{V}\cdot\vec{B})\vec{B} \end{array} \right]. \tag{3}$$

In Eqs. (2) and (3) the specific total plasma energy is $e = p/(\rho(\gamma-1)) + V^2/2 + B^2/(2\rho)$, where p is the molecular pressure, V is the magnitude of the fluid velocity, and B is the magnitude of the magnetic field. The term $p_B = \frac{\vec{B}\cdot\vec{B}}{2}$ in Eq. (3) is known as the magnetic pressure.

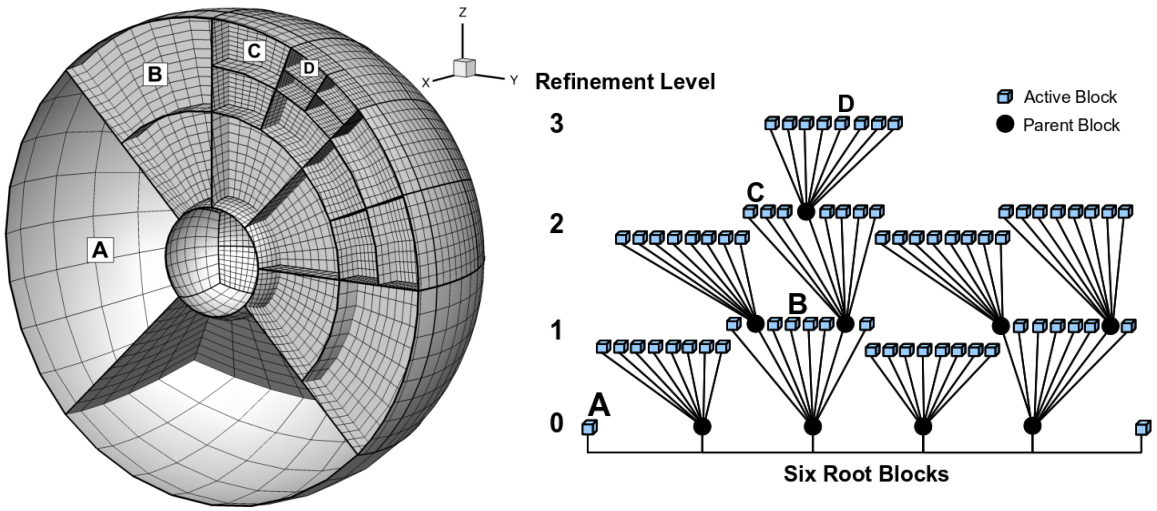
The so-called Powell source term, $\mathbf{S}$, is given by

$$\mathbf{S} = -\left[\begin{array}{c} 0 \\ \vec{B} \\ \vec{V} \\ \vec{V}\cdot\vec{B} \end{array} \right] \nabla\cdot\vec{B} = \mathcal{S}\,\nabla\cdot\vec{B}\,, \tag{4}$$

the role of which is not only to make the MHD equations Galilean invariant and symmetrizable, as proved by Godunov [30], but also to provide a numerically stable way of controlling the errors in the divergence free condition, $\nabla\cdot\vec{B} = 0$, a technique first advocated by Powell and co-workers [23]. Note that our framework can also be used with other mechanisms to control $\nabla\cdot\vec{B}$, for example, the generalized Lagrange multiplier mechanism of [31]. However, these alternative approaches are not considered here.

Note that Eqs. (1)-(4) represent the non-dimensional scaled form of the MHD equations following from the non-dimensionalization described by, e.g., Powell *et al.* [23] and by Groth *et al.* [2] for the solar wind. The ideal gas equation of state $p = \rho RT$ is assumed, where T is the gas temperature and $R = 1/\gamma$ is the gas constant. For a polytropic gas (thermally and calorically perfect), the ratio of plasma specific heats, γ, is a constant, and the specific heats are given by $C_v = 1/(\gamma-1)$ and $C_p = \gamma/(\gamma-1)$. Unless specified otherwise, di-atomic gases are used throughout this paper, which corresponds to $\gamma = 7/5 = 1.4$. The aforementioned equations can be simplified by setting the magnetic field, $\vec{B}$, equal to zero, in which case the conservation laws of the non-magnetized plasma reduce to the Euler equations of gas dynamics in non-dimensional form.

The column vector, $\mathbf{Q}$, appearing in Eq. 1 is used here to represent different volumetric sources arising from the physical modelling of the space-physics problem studied. Thus, the volumetric source term $\mathbf{Q} = \mathbf{Q}_G + \mathbf{Q}_H + \mathbf{Q}_O$ accounts for sources associated with external gravitational fields, $\mathbf{Q}_G$, effects of coronal heating processes and heat and radiation transfer, $\mathbf{Q}_H$, as well as other effects, $\mathbf{Q}_O$. Note that the source term $\mathbf{Q}_H$ is used only for modelling the time-averaged solar wind (see Sect. 5.2.3 for details) whereas the term $\mathbf{Q}_O$ is used to construct an analytical solution to the MHD equations by the method of manufactured solutions [32], as described in Sect. 5.2.1.



(a) Cross-section of a cubed-sphere grid with three refinement levels.

(b) Depiction of octree data structure used to keep track of connectivity among solution blocks.

Figure 3: Illustration of three-dimensional block-based adapted cubed-sphere grids showing the blocks and block boundaries.

4. Parallel Adaptive Second-Order-Accurate Simulation Framework on Cubed-Sphere Grids

4.1. Block-Structured Adaptive Mesh Refinement Approach

A fair variety of adaptive mesh refinement techniques have emerged in the literature, which can broadly be classified into cell-based, patch-based, block-based and hybrid block-based techniques. A recent review and discussion of all these strategies is provided by Gao and Groth in [13]. Among these AMR strategies, block-based AMR approaches lend themselves readily to efficient parallelization on distributed-memory clusters. This feature made the block-based AMR technique attractive for application to different configurations (e.g., on Cartesian, curvilinear and/or body-fitted meshes in conjunction with schemes of various orders of accuracy). AMR has been applied to a variety of engineering and scientific problems by Quirk [33], Berger and Colella [34–36], Gombosi and co-workers [2, 3, 23, 37, 38], Groth and co-workers [13, 39–42], Keppens *et al.* [43] and Jablonowski *et al.* [22], among many others.

The block-based AMR strategy considered in this work follows the approach of Gao and Groth [12, 13] and mesh adaptation is accomplished by dividing and coarsening appropriate solution blocks. To illustrate the general setting for our approach, an example of a 3D block-structured cubed-sphere mesh containing blocks of different resolution levels is depicted in Fig. 3 along with the corresponding hierarchical octree data structure. A refinement step divides a "parent" block into eight "children", and the reverse happens for coarsening. In general, each block has an equal number of cells for algorithm simplicity and computational efficiency (load balancing). Furthermore, for accuracy reasons and reduced algorithm complexity, the maximum resolution change between adjacent blocks is limited to two. In the adaptation process a grid hierarchy gets generated from a set of initial (i.e., root) blocks in the form of a sequence of nested grids that can be conveniently tracked with a hierarchical block-tree data structure, as the one depicted in Fig. 3(b). Note that only the blocks at the highest hierarchical level at a given time are actively used in the solution procedure of the PDE system and therefore, most of the simulation data stored in memory is associated with these active blocks, yet enough connectivity information for the parent blocks needs to be kept stored so as to be able to establish block connectivity among blocks that are neighbours in physical space and to exactly recover the "ancestors" during mesh coarsening.

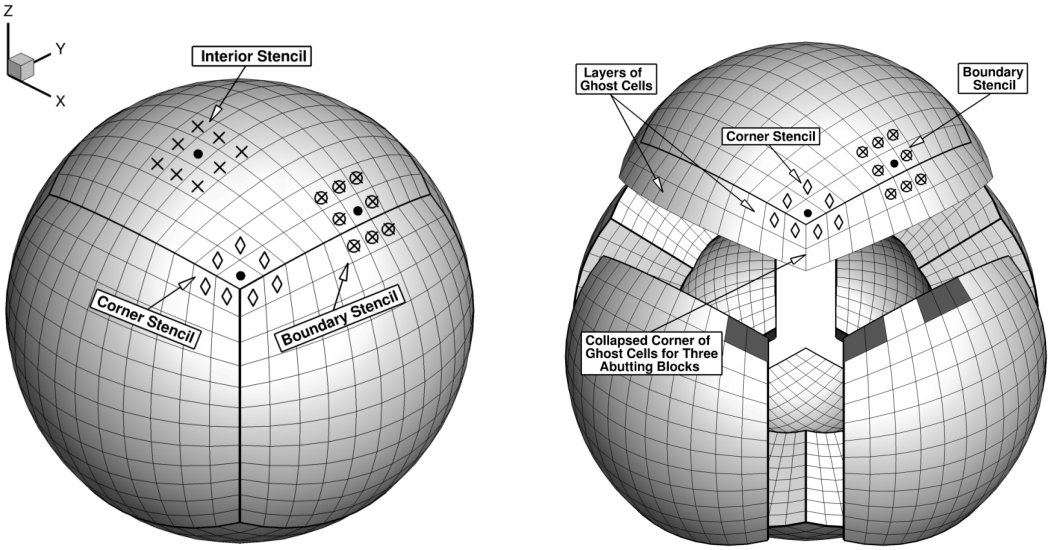
The octree data structure illustrated in Fig. 3(b) contains multiple "roots" and multiple "trees" that keep track of the connectivity and the mesh resolution change between solution blocks. The blocks of the initial mesh represent the roots of the forest. Traversal of the multi-tree structure by recursively visiting the parents and children of solution blocks can be used to determine block connectivity. However, in order to reduce overhead associated with accessing

solution information from adjacent blocks, the neighbours of each block are pre-computed and stored, providing direct interconnects between blocks in the hierarchical data structure that are neighbours in the grid. In general, a given solution block in the interior of one of the grid sectors has a minimum of 26 neighbouring blocks (all neighbouring blocks in a $3 \times 3 \times 3$ block stencil), the actual number depending on the level of mesh resolution change encountered on each of the 26 boundary elements (6 faces, 12 edges and 8 vertices), and having a maximum value of 56 neighbouring blocks. For blocks adjacent to domain boundaries or to cubed sphere corners the number of neighbouring blocks is smaller. In order to facilitate information exchange between adjacent blocks, additional layers of "ghost" cells that overlap with interior cells of adjacent blocks are added to each solution block. The numerical scheme used in this work (see Sect. 4.3), handles transparently the computational cells near block interfaces by making use of the overlapping ghost cells and accommodating variable stencil sizes. For second-order accuracy two ghost cell layers are sufficient, which represents a small memory overhead relative to the benefits to be gained when the parallel aspects of the algorithm are considered, as described below. One of the advantages of the hierarchical octree data structure is that it readily permits local mesh refinement. Local modifications to the multi-block mesh can be performed without re-gridding the entire mesh and re-calculating all solution block connectivities.

4.2. Multi-Block Cubed-Sphere Grid with Unstructured Root-Block Connectivity

The main technical difficulty in applying the block-structured adaptive multi-block concept to cubed-sphere grids is to deal transparently with the unstructured connectivity between adjacent blocks that occurs at sector boundaries and sector corners. The first issue arises at sector corners (see Fig. 4): grid cells adjacent to one of the eight sector corners have only seven neighbouring cells (in 2D), while all other cells have 8 neighbours (these neighbours are used in stencils for solution reconstruction and flux calculation, see next section). This issue is dealt with in our approach by automatically detecting blocks with such corner cells, and by assigning "collapsed" corner ghost cells to those blocks sharing the relevant corner (as in [13]). In practice, this is implemented by using dummy values in those corner ghost cells. These collapsed ghost cells are not used in the stencils for reconstruction computation, so grid cells adjacent to sector corners employ reduced stencil sizes. The flexible finite-volume reconstruction mechanism of the proposed solution method (see the next subsection) handles this transparently without reducing the local order of accuracy. In our implementation, blocks with collapsed ghost cells are detected from the block connectivity data structure: if no neighbouring block is found in the direction of a block corner the corresponding corner ghost cells are taken to be collapsed.

The second issue is the connectivity between adjacent blocks (recall Fig. 1(b)): on cubed-sphere grids, it is no longer true that, for example, eastern block boundaries are always adjacent to western block boundaries, and it is necessary to carefully keep track of the block interface adjacency, and of the orientation of index axes in adjacent blocks (i.e., the i, j, and k indices in the logically Cartesian data structure of a block may correspond to a different orientation of the i, j, and k indices in a neighbouring block). We use the unstructured root-block approach of [13] to handle this issue in a systematic way. The logic employed follows the methodology proposed in the Computational Fluid Dynamics General Notation System (CGNS) [44]. A typical refined block stores the relative orientation of the neighbouring blocks in the directions of all 26 boundary elements (6 faces, 12 edges and 8 vertices), and for each neighbour the orientation of the i, j, and k axes relative to the orientation of the i, j and k axes in the current block is stored in compact form as a three-component transformation array. For example, consider the index system of block 1 in Fig. 1(b) in relationship to its south neighbour, block 6. The i and j indices of block 6 run in directions opposite to the i and j indices in block 1. This is indicated by negative signs in the block 1-to-block 6 transformation array, which is given by $[-1, -2, +3]$. The components 1, 2 and 3 in the array signify the i, j and k indices in block 6. The value 1 in the first component of the array means that the i index from block 6 is associated with the first index of block 1 (its i index), and the negative sign indicates that they run in opposite directions. The value -2 in the second position means that the j indices of the two blocks are also associated and running in opposite directions, and the value $+3$ in the third direction indicates that the k indices have the same orientation. Similarly, considering block 2, the block 2-to-block 5 transformation array is given by $[-2, +1, +3]$, because the second index in block 5 (j) corresponds to the first index in block 2 (i) (running in the opposite direction), the first index in block 5 (i) corresponds to the second index in block 2 (j) (running in the same direction), and the third index in block 5 (k) corresponds to the third index in block 2 (k) (running in the same direction). These transformation arrays provide a convenient short-hand notation for the transformation matrices [12, 13, 44] describing the relation between indices of two adjacent blocks, which can



(a) Compact view of a cubed-sphere grid and depiction of cells participating in reconstruction stencils in different regions. The data from the cells residing on different blocks than the reconstructed cell is locally reproduced by using overlapping layers of ghost cells, as shown in (b).

(b) Exploded view of the cubed-sphere grid shown in (a) illustrating the ghost cell layers for one block and the actual cells forming the reconstruction stencils. The interior cells marked with shaded gray are the cells from adjacent blocks which have their data duplicated in the ghost cells.

Figure 4: Examples of interior, boundary and corner stencils for different cells of a cubed-sphere mesh with six 12×12×8 blocks and a total of 6,912 cells. The cell of which the solution is reconstructed is marked with • symbol and the neighbouring cells that are part of the stencil are marked with ×, ◇ and ⊗ symbols for the interior, boundary and corner stencil, respectively.

be used to exchange solution information between blocks having common interfaces in a general and transparent way. For more details, see [13].

The connectivity information is propagated from the root blocks to refined blocks via the octree data structure, in such a way that each refined block stores a transformation array describing index axis alignment with all of its (typically 26) neighbour blocks. These block-to-block transformation arrays are used in the solution procedure to properly compute numerical fluxes through the block boundaries (via the ghost cell and reconstruction mechanisms, see below). Note that the transformation array mechanism is implemented uniformly for all blocks, but it only results in nontrivial action at block-block interfaces along sector boundaries, in a transparent manner.

4.3. Multi-Dimensional Second-Order Finite-Volume Scheme with Uniform Treatment of Sector Boundaries and Corners

Numerical solutions of Eq. 1 on the cubed-sphere grids are sought by applying a Godunov-type finite-volume spatial discretization procedure [45] in conjunction with second-order polynomial reconstruction and Riemann-solver based flux functions. Thus, the set of coupled ordinary differential equations (ODEs) resulting from the application of this finite-volume discretization to Eq. 1 for cell (i, j, k) of a multi-block mesh composed of hexahedral computational cells is given by

$$\frac{d\mathbf{U}_{i,j,k}}{dt} = -\frac{1}{V_{i,j,k}} \sum_{m=1}^{N_f} \left(\vec{\mathbf{F}} \cdot \vec{n}\, \Delta A\right)_{i,j,k,m} + \bar{\mathbf{S}}_{i,j,k} + \bar{\mathbf{Q}}_{i,j,k} = \mathbf{R}_{i,j,k}(\mathbf{U}) \,, \tag{5}$$

where $V_{i,j,k}$ is the cell volume, N_f is the number of cell faces, and $\vec{n}$ and ΔA are the unit outward normal vector and the area of cell face m, respectively. The numerical fluxes, $\vec{\mathbf{F}} \cdot \vec{\mathbf{n}}$, at the midpoint of each face of cell (i, j, k) are determined from the solution of a Riemann problem. Given the left and right interface solution states, $\mathbf{U}_l$ and $\mathbf{U}_r$, an upwind numerical flux is evaluated by solving a Riemann problem in the direction defined by the normal to the face. In our computational studies, the Lax-Friedrichs [46], HLLE [47], and modified HLLE [48] approximate Riemann

solvers are used in evaluating the numerical flux. The left and right initial states for the Riemann problem, $\mathbf{U}_l$ and $\mathbf{U}_r$, are determined based on the associated primitive variables, $\mathbf{W}_{l,r} = \left[\ \rho, \ \vec{V}, \ \vec{B}, \ p \ \right]_{l,r}$, at the interface. These primitive variable states $\mathbf{W}_l$ and $\mathbf{W}_r$ at the cell interfaces are obtained by performing the multi-dimensional limited piecewise linear solution reconstruction proposed by Barth [29] in conjunction with either Barth-Jespersen [29] or Venkatakrishnan [49] slope limiters, so as to avoid the occurrence of spurious oscillations and non-physical solutions (i.e., negative density and/or pressure). Thus, the limited linear representation of the primitive solution for cell (i, j, k) is written as a polynomial expansion around the cell centroid, $\vec{x}_{i,j,k}$, as

$$\mathbf{W}_{i,j,k}(\vec{x}) = \overline{\mathbf{W}}_{i,j,k} + \Phi_{i,j,k} \vec{\nabla}\mathbf{W} \cdot (\vec{x} - \vec{x}_{i,j,k}), \tag{6}$$

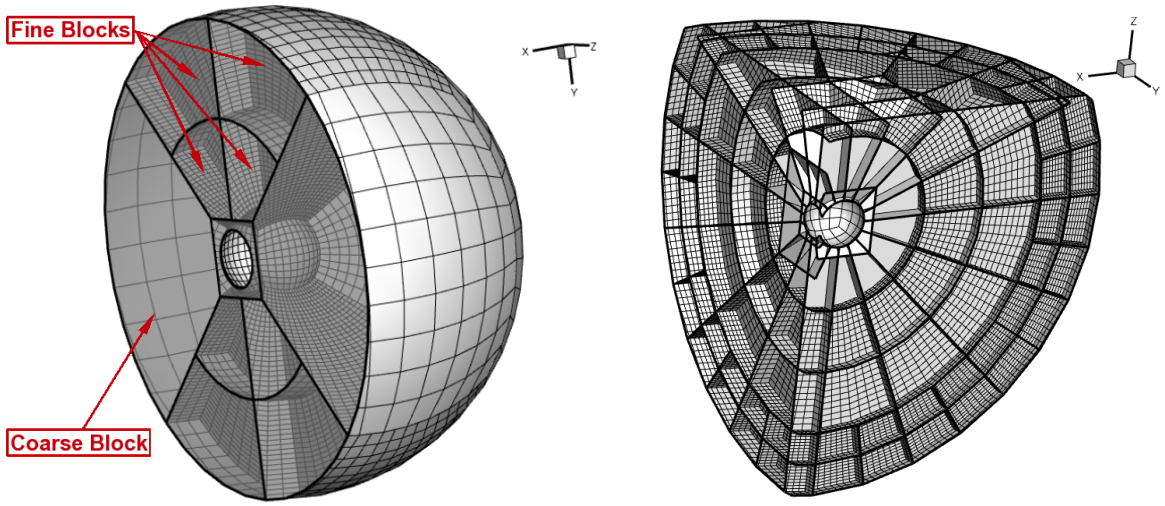
where $\Phi_{i,j,k}$ is the vector of limiter values that are determined individually for each solution variable, and $\overline{\mathbf{W}}_{i,j,k}$ is the cell average solution vector. The gradients of the primitive variables, $\vec{\nabla}\mathbf{W}$, follow from the least-squares solution of the linear reconstruction based on a three-dimensional supporting stencil which generally includes 27 cells in total (i.e., the $3 \times 3 \times 3$ block composed of the reconstructed cell plus the 26 nearest neighbouring cells). For each solution variable reconstruction, three unknowns are calculated by imposing 26 equations: the average of the reconstructed polynomial over each cell in the stencil is required to recover the existing volume average solution in that cell. This overdetermined system of linear equations is solved in the least-squares sense [29]. The reconstruction procedure could in principle be carried out with smaller stencils (for example, a seven-point stencil would lead to six equations), but it would then loose much of its multi-dimensional character, and we have indeed found that the 27-point stencil provides better robustness against non-uniformity in the grid and solution gradients that are not aligned with the grid. The least-squares procedure allows for flexible stencil sizes, without reducing the local order of accuracy. In particular, for grid cells adjacent to the radial lines that pass through cubed-sphere sector corners, the stencil size is reduced from 27 to 24, which is handled transparently by the least-squares reconstruction procedure.

Figure 4 shows the computational cells belonging to a spherical face that are included in the reconstruction stencil of representative cells that lie either in the interior, at a boundary, or at a corner of the cubed-sphere sectors. The complete 3D stencils are obtained by including the radial neighbours (i.e., above and below) of the cells depicted in Fig. 4(a) and (b). The overlapping layers of ghost cells depicted in these figures are used to compute updates in cells adjacent to the block boundary. Note that in our approach two layers of ghost cells are used: reconstruction is performed in interior cells but also in the first layer of ghost cells, using the solution values in the second layer of ghost cells, to avoid an additional communication step. The multi-dimensional reconstruction procedure is observed to perform accurately and robustly on cubed-sphere grids (see the convergence results below), avoiding any special interpolation treatment of cells near grid sector boundaries. Note that this is in contrast to most existing algorithms, which require either one-dimensional [5, 11] or one-sided interpolation [9, 15] near sector boundaries.

The numerical procedure outlined above results in a second-order accurate upwind finite-volume scheme. The average value of the Powell source term, $\bar{\mathbf{S}}_{i,j,k}$, of cell (i, j, k) is computed with second-order accuracy by estimating the quantity $\mathcal{S}$ based on the appropriate cell-averaged solution variables (i.e., the value of the reconstructed polynomial at the cell centroid) and multiplying it with the following discretization for $\nabla \cdot \vec{B}$:

$$\left(\nabla \cdot \vec{B}\right)_{i,j,k} = \frac{1}{V_{i,j,k}} \sum_{m=1}^{N_f} \left(\vec{B}_f \cdot \vec{n} \, \Delta A\right)_{i,j,k,m} , \tag{7}$$

where $\vec{B}_f$ is an interface magnetic field computed as the arithmetic mean of the left and right reconstructed values, that is, $\vec{B}_f = \left(\vec{B}_l + \vec{B}_r\right)/2$. The contribution of all other volumetric sources to the solution residual, $\mathbf{R}_{i,j,k}$, is also evaluated with second-order accuracy by computing the average source term $\bar{\mathbf{Q}}_{i,j,k}$ at the centroid of cell (i, j, k). To obtain steady-state solutions for the problems considered in this work, the coupled system of nonlinear ODEs given by Eq. 5 is solved using multi-stage explicit time-marching schemes [50] in conjunction with global or local time-steps obeying the Courant-Friedrichs-Lewy (CFL) stability condition. Future follow-on research will consider improving the computational efficiency of the cubed-sphere framework by combining the proposed spatial discretization procedure with more effective time-marching approaches such as the parallel implicit Newton-Krylov-Schwarz algorithm of Northrup and Groth [51, 52].



(a) Cross-section of a cubed-sphere grid with one refinement level

(b) Complex cut into an adapted cubed-sphere grid with a few refinement levels

Figure 5: Illustration of three-dimensional block-based adapted cubed-sphere grids showing the block boundaries and the associated meshes.

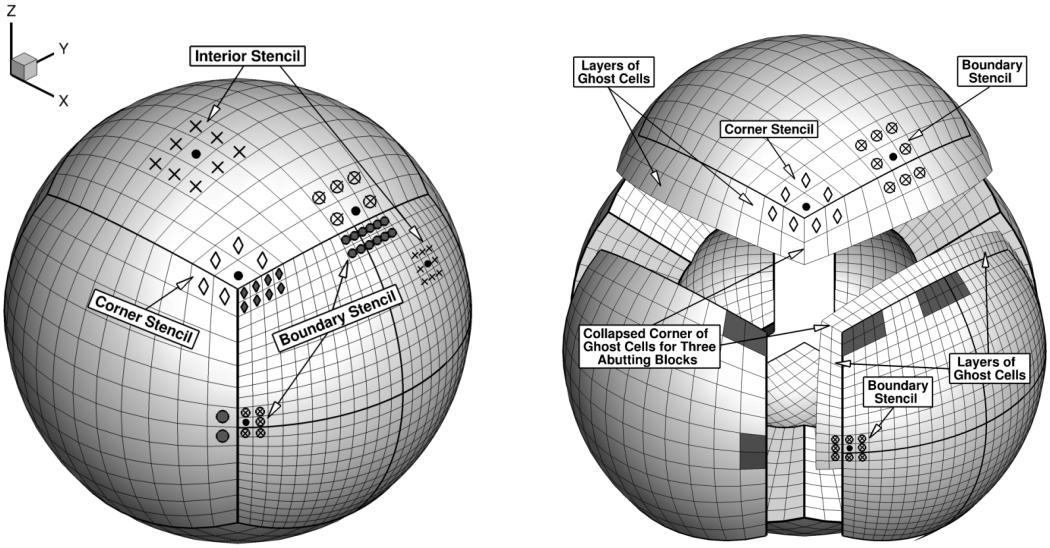
4.4. Block-Based Adaptive Refinement and Coarsening Strategy

The flexible block-based hierarchical data structure discussed in Sect. 4.1 is used in conjunction with the spatial discretization procedure outlined above to facilitate automatic solution-directed mesh adaptation on 3D cubed-sphere grids. Mesh adaptation is accomplished herein by refining and coarsening appropriate solution blocks. In regions requiring increased cell resolution, a "parent" block is refined by dividing it into eight "children". 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. 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. Mesh refinement criteria are used to decide on adaptive resolution changes in a procedure that can be summarized as follows (see [13, 51] for a more extensive discussion).

Local refinement and coarsening of the mesh is directed according to so-called physics-based refinement criteria. In particular, for the test cases considered in this paper, density gradient and divergence and curl of velocity are used in the decision to refine or coarsen a solution block. For definiteness, it is discussed here in the case where the density gradient is used as the basis for refinement.

When refinement and coarsening are desired in the course of a simulation, the following procedure is carried out. First, the maximum per-block value of the density gradient in each cell is computed, and then a normalized scale for the variation of the refinement measure per block is built based on the global minimal and maximal values for all solution blocks. Based on user-specified threshold values for refinement and coarsening on a scale between zero to one, blocks with maximum density gradient values above the refinement and below the coarsening thresholds are flagged for refinement and coarsening, respectively. All the other blocks are flagged to keep their current mesh resolution. The final decision about which blocks get refined/coarsened or are left unchanged employs a check for possible violations of the condition that adjacent blocks may not differ by more than a factor of two in resolution after refinement/coarsening. The conflicts are resolved according to the algorithm in [12, 13], in which precedence is given to block refinement. After these conflicts are resolved, the appropriately flagged blocks are refined and coarsened as scheduled. The newly created blocks during refinement are distributed evenly among computing cores (for load balancing) in accordance to a resource list mirrored on all computing cores which stores information about the distribution of solution blocks and the available computational resources.

For the steady-state computations to be presented in the numerical results section of this paper, this refinement/coarsening mechanism is invoked at regular intervals during the convergence history. Our implementation also



(a) Compact view of a refined cubed-sphere grid and depiction of cells participating in reconstruction stencils in different regions. The data from the cells residing on different blocks than the reconstructed cell is reproduced by using overlapping layers of ghost cells, as shown in (b).

(b) Exploded view of the refined cubed-sphere grid shown in (a) illustrating the ghost cell layers for one coarse and four refined blocks. Additionally, the actual cells forming the reconstruction stencils are shown for both low- and high-resolution blocks. The interior cells marked with shaded gray are the cells from adjacent blocks which provide data to the ghost cells at different transitions of mesh resolution (by duplication for no resolution change, by prolongation for coarse-to-fine change, and by restriction for fine-to-coarse change).

Figure 6: Examples of interior, boundary and corner reconstruction stencils on a refined cubed-sphere mesh with five coarse sectors and one refined sector (all 13 blocks in the grid are $12\times12\times8$, and there are 14,976 total cells). The cell of which the solution is reconstructed is marked with $\bullet$ symbol and the neighbouring cells that are part of the stencil are marked with $\times$, $\diamond$ and $\otimes$ symbols for the interior, boundary and corner stencil, respectively.

allows for time-dependent, dynamical refinement and coarsening on the cubed-sphere grid, simply by applying the procedure at regular intervals during time-dependent computations. Note that blocks at sector interfaces and sector corners of the cubed-sphere grid are handled by the refinement/coarsening mechanism in a fully transparent way. For example, when refined blocks are created, their neighbours are determined and blocks adjacent to sector corners receive less than the full amount of neighbouring blocks (due to "missing" corner blocks), and will automatically be assigned collapsed corner ghost cells. The transformation arrays for these blocks take into account the unstructured connectivity automatically.

Two illustrative examples of the application of the AMR algorithm described above to cubed-sphere grids are shown in Fig. 5. Note that the mesh refinement procedure applied to the cubed sphere is used both to enhance local solution features and to parallelize the algorithm via domain partitioning with more than six blocks. Note also that a body-fitted mesh is readily obtained for the cubed-sphere approach by simply ensuring that newly inserted mesh nodes at the inner and outer spherical boundaries belong exactly to the appropriate spherical shell.

Standard multi-grid-type restriction and prolongation operators [53] are used to evaluate the solution on all blocks created by the coarsening and division processes, respectively. A coarse cell has mean solution data calculated from the corresponding eight fine cells it overlaps with in a way that satisfies conservation [36, 53]. 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 volume-averaged solution of a fine cell is derived from the solution of a parent coarse cell with second-order accuracy by simply evaluating the limited linear reconstruction of the parent at the centroid of the fine cell. The same

restriction and prolongation operators are also used for ghost cells at block interfaces where the resolution changes, in all time steps (see Fig. 6). This procedure directly influences the domain of dependence of solution reconstructions for cells that include ghost cells in their supporting stencils. Thus, even though reconstruction stencils are always locally constructed in the fashion described in Sect. 4.3, the actual domain of dependence depends on the mesh resolution transition at block interfaces. As can be seen from Fig. 6(b), the stencil construction algorithm outlined in Sect. 4.3 can be applied in the same way to each of the cubed-sphere blocks regardless of their spatial resolution.

Within the AMR approach, additional inter-block communication is also required at interfaces with resolution changes to strictly enforce the flux conservation properties of the finite-volume scheme [35, 36]. In particular, the interface fluxes computed on more refined blocks are used to correct the interface fluxes computed on coarser neighbouring blocks and ensure that the solution fluxes are conserved across block interfaces.

4.5. Parallelization with Uniform Treatment of Sector Boundaries and Corners

With the mechanisms of unstructured root-block connectivity and multi-dimensional reconstruction with flexible stencil sizes in place, parallelization can be performed in a way that is fully transparent to sector boundaries and corners. As in the other previous work of Groth *et al.*, e.g., [13, 39–42], an efficient domain partitioning is achieved in our implementation by distributing the active solution blocks equally among available processor cores, with more than one block permitted per processor core. This approach efficiently exploits the self-similar nature of the solution blocks and readily produces an effective load balancing. Inter-processor communication is mainly associated with block interfaces and involves the exchange of ghost-cell solution values and conservative flux corrections at every stage of the multi-stage time integration procedure. To improve the efficiency of the parallel communication, message passing of the ghost-cell values and flux corrections is performed by consolidating inter-processor messages. Blocks adjacent to grid sector corners feature "collapsed" corner ghost cells, and since there is no neighbour block associated with the collapsed ghost cells in the data structure, messages are not sent for collapsed ghost cells. Blocks adjacent to grid sector boundaries may have local i, j, k index systems that are oriented differently from their neighbouring blocks across the grid sector boundary, but this is taken into account transparently when filling in ghost cells from messages sent by the neighbouring blocks, using the transformation array mechanism described above. In this way, all solution blocks are treated equally in a way that facilitates transparent parallelization across grid sector boundaries and corners. The combination of this scalable domain partitioning and the effective AMR-based block-multiplication procedure have allowed us to perform efficient parallel calculations on 3D cubed-sphere grids with in excess of 3,000 computing processor cores. A quantitative demonstration of the parallel performance is provided for a representative space-physics problem in Sect. 5.2.3.

5. Numerical Results

A set of numerical results demonstrating the accuracy and capabilities of the proposed cubed-sphere framework is now described for a range of flow problems with magnetized and non-magnetized plasma. Results for both fixed and AMR meshes are included.

5.1. Results for Non-Conducting Fluid

5.1.1. Fully Supersonic Flow

To assess the accuracy of the finite-volume scheme on cubed-sphere grids, uniform convergence studies with a spherically symmetric expansion of a supersonic non-magnetized gas flow (without gravitational field) are performed. Since the cubed-sphere grid is not spherically symmetric, this problem provides a relevant test case for our implementation (in contrast, this problem is not interesting for a spherical-polar grid). The computational domain of this test case is defined by inner and outer spheres of radius $R_i = 1$ and $R_o = 4$, respectively. With V_r the radial velocity and $\vec{V}_{\parallel}$ the velocity parallel to the inflow boundary, the boundary conditions at R_i are $\rho = 10$, $V_r = 4.5$, $\vec{V}_{\parallel} = 0$, and $p = 26$. The analytical solution of this flow problem can be obtained in spherical coordinates by integrating the Euler equations or by manipulating algebraically the flow invariants (i.e. mass flow rate, entropy and total enthalpy). Thus, the radial velocity, V_r, at any spherical radius, r, is obtained numerically by solving the non-linear expression

$$C_3 - \frac{1}{r^2 V_r \left[(C_2 - V_r^2)^{\frac{1}{\gamma-1}} \right]} = 0 \,, \tag{8}$$

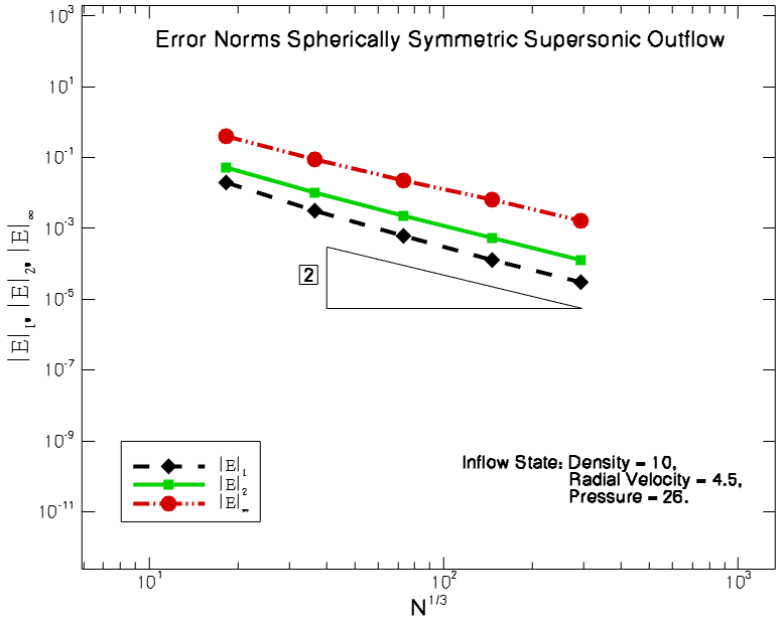


Figure 7: L_1, L_2, and L_∞ error norms in the predicted solution density for a supersonic expansion.

for V_r, where C_2 and C_3 are flow constants that can be determined based on the inflow parameters ρ_i, p_i and $v_{r,i}$:

$$\begin{aligned} C_3 &= \frac{1}{\left(\frac{2\gamma}{\gamma-1}\frac{p_i}{\rho_i}\frac{1}{\gamma-1}\right)R_i^2 v_{r,i}} \\ C_2 &= \frac{2\gamma}{\gamma-1}\frac{p_i}{\rho_i} + v_{r,i}^2 \,. \end{aligned} \tag{9}$$

Density and pressure as a function of radius can then be obtained from Eq. 9.

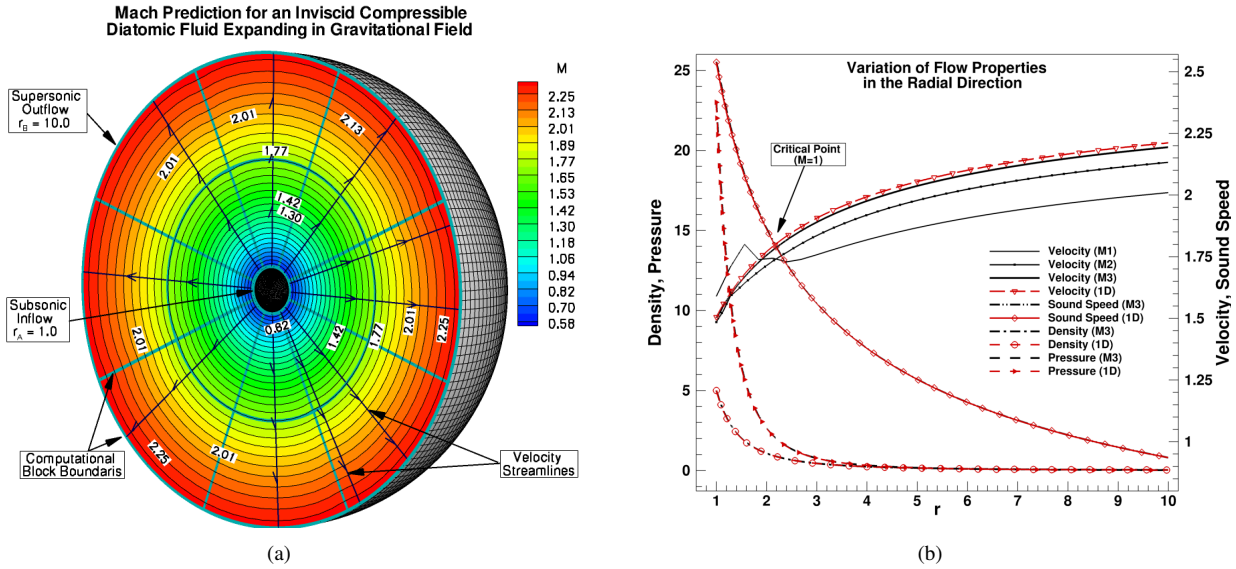
The L_1, L_2, and L_∞ norms of the error in the predicted solution density obtained on a series of grids ranging in size from 8×8×16 to 128×128×256 cells for each of the six cubed-sphere blocks, which corresponds to 6,144 and 25,165,824 total cells, respectively, are given in Fig. 7 for this supersonic flow. Note that the error norms are evaluated in this work based on the point-wise differences between the exact solution computed at the centroid of each computational cell and the corresponding average solution (i.e., the value of the reconstruction at the cell centroid). Each L_p-error norm is then computed by applying the appropriate p-norm to the vector of cell average errors, determined as the product of the point-wise errors and the corresponding cell volumes, and by normalizing with the volume of the whole domain for L_1 and L_2 norms, and that of the appropriate computational cell for L_∞ norm.

The results in Fig. 7 show that the second-order theoretical accuracy for this smooth flow is achieved in all error norms as the mesh is refined, indicating that the finite-volume algorithm handles accurately the non-orthogonal grid near the projection of the cube corners and at sector boundaries.

5.1.2. Spherically Symmetric Transonic Wind

To illustrate the accuracy of the algorithm for solar wind-like solutions, the expansion of an inviscid compressible non-conducting gas under the influence of a gravitational field is now considered. Similar validation cases have been studied in [54] and the Newton Critical Point (NCP) method outlined there for one-dimensional problems has been used here to provide a highly accurate adaptively refined reference solution with 918 points.

A spherically symmetric external gravitational field $\vec{g} = -g_*/r^2 \hat{\mathbf{e}}_r$ is considered. Here, r is the normalized distance to the solar or planet center, $\hat{\mathbf{e}}_r$ is the radial unit vector, and the non-dimensional constant $g_* = G\,M_*/(l_o\,a_o^2)$ is computed based on the gravitational constant, G, the solar or planetary mass, M_*, the reference length scale, l_o, and the ion-acoustic wave speed, a_o, of a suitable reference solution state. Thus, the volumetric source term $\mathbf{Q}_G$, which accounts



(a) (b)

Figure 8: (a) Predicted Mach number distribution for spherically symmetric transonic wind obtained on the uniform mesh M3 with 1,228,800 cells; and (b) variation of flow properties in the x-axis direction predicted on M3 are compared against those obtained with the NCP method in one-dimension. The velocity prediction on meshes M1 and M2 (see text) is also shown.

for the effect of gravitational acceleration, is expressed as a function of the radial position vector, $\vec{r} = r\hat{\mathbf{e}}_r$, in the following vector form

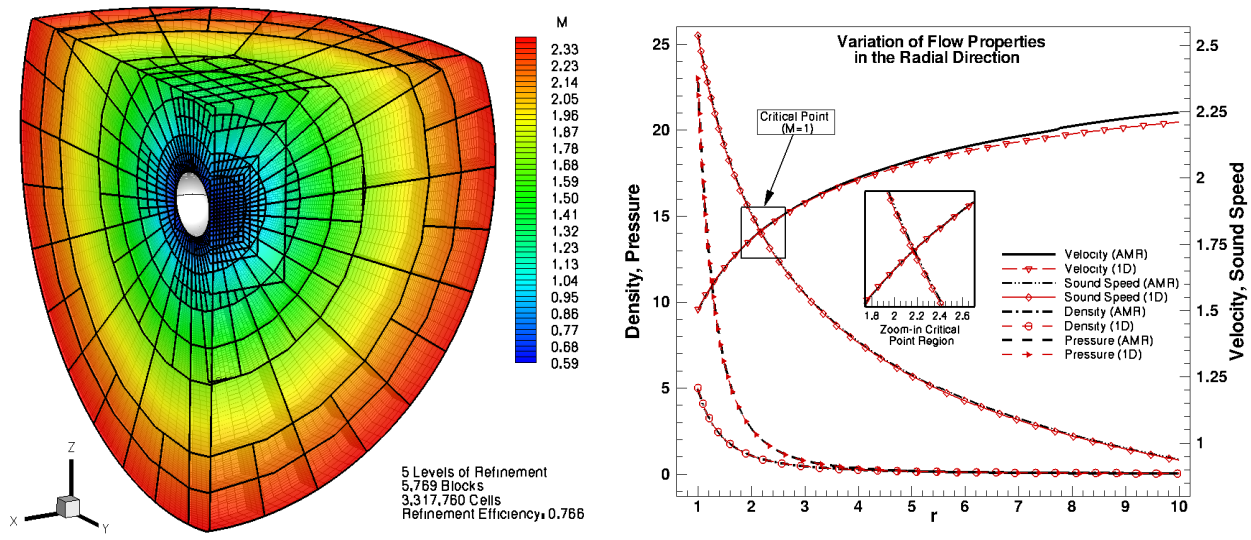
$$\mathbf{Q}_G = -\frac{\rho g_*}{r^3}\left[0, \vec{r}, \vec{0}, \vec{r}\cdot\vec{V}\right]^{\mathrm{T}} . \tag{10}$$

As in [54], normalization factors have been used to provide non-dimensional variables. The spherically symmetric subsonic inflow at the inner sphere $R_i = 1$ is defined by the dimensionless density $\rho = 5$ and pressure $p = 23$. We choose $GM_* = 14$ which allows for a stationary transonic outflow solution in which the radial flow velocity is initially subsonic, then passes through a critical point where the local Mach number is one, and subsequently takes on supersonic values beyond the critical radius [54]. Although a crude approximation of a real solar wind, the accurate capturing of the transonic behaviour outlined above is a good test case for our code.

The predicted Mach number distribution for this spherically symmetric transonic flow obtained on a uniform cubed-sphere mesh M3 with 1,228,800 cells is shown in Fig. 8(a). In this case, 128 cells are used to resolve the radial direction (i.e., 128 uniform radial points in the 3D simulation have to be compared to 918 highly adaptive points in the 1D reference solution). At the inner boundary, V_r is extrapolated from the computational domain, and $\vec{V}_{||} = 0$. At the outer boundary, all state variables are extrapolated to the ghost cells. Solving Riemann problems at the boundaries ensures the correct propagation of the flow characteristics, thereby allowing the solution residual to converge to machine accuracy. The result shows that the steady-state transonic solution has been correctly captured and the Mach number contour lines are close to circular indicating the good homogeneity of the flow solution at this grid resolution.

In Fig. 8(b), the flow properties of the predicted solution sampled along the x-axis are compared against those obtained with the NCP method on a non-uniform grid so as to capture the sharp density and pressure gradients near the inflow boundary. Additionally, the prediction of the radial velocity in the x-direction obtained with meshes M1 and M2 with 19,200 and 153,600 cells, respectively, has been also plotted. Note that these two meshes have 32 and 64 cells, respectively, in the radial direction. It is quite noticeable that the solution significantly improves as the mesh is refined, and it approaches in the convergence limit the 1D transonic solution predicted by the NCP method.

To assess the capability of the computational framework to improve the prediction of flow properties in regions of large solution gradients and the accuracy of the critical point location, the same transonic wind problem has been reconsidered with automatic solution-directed mesh adaptation. The initial mesh in this case consisted of 48 blocks,



(a) (b)

Figure 9: (a) Predicted Mach number distribution for spherically symmetric transonic wind obtained on the adapted cubed-sphere mesh after 5 refinement levels and with 3,317,760 cells; and (b) the flow properties in the x-axis direction are compared against those predicted with the NCP method.

each with $6\times6\times16$ cells. The contour plot of the Mach number obtained on the refined mesh after five levels of refinement applied to the 3D cubed-sphere grid is shown in Fig. 9(a). The block boundaries shown in the same figure indicate that a fairly symmetric refinement is automatically generated so as to improve the prediction of solution gradients near the inflow boundary, and demonstrate that the proposed AMR approach can be successfully applied to three-dimensional cubed-sphere grids. The comparison of flow properties predicted by the 1D NCP method and the 3D AMR algorithm described in this work along the x-axis are shown in Fig. 9(b). It is quite obvious that the two solutions show excellent agreement in the transonic region and only slight deviations are encountered close to the supersonic outflow due to reduced resolution in the adaptive grid. Note that the 3D adaptive solution has approximately 136 points in the radial direction, while the 1D reference solution has 918 adaptive points. Notice, however, that for a similar number of radial points the AMR grid captures the velocity variation significantly better than the finest uniform mesh, M3, for almost the whole domain. The somewhat under-resolved supersonic outflow in the AMR case is only a consequence of the parameters used for the gradient-based refinement criteria and of available computational resources, and does not represent a limitation of the AMR framework.

5.2. Results for Magnetized Plasma

5.2.1. Systematic Grid Convergence Studies Based on MHD Manufactured Solution

To assess the accuracy of the finite-volume scheme on cubed-sphere grids, we have performed convergence studies for a steady-state axi-symmetric exact solution of magnetized gas flowing outward at supersonic speeds. (Note that the cubed-sphere grid is not axi-symmetric, so this problem is adequate for testing the three-dimensionality of our framework.)

As far as we know, there are no non-trivial MHD test problems with an exact solution in a domain between two concentric spheres (except for problems with rather trivial radial 1D solutions similar to the problem of Section 5.1.1), which can be used for quantitative grid convergence studies. To fill this void, we propose a new fully supersonic axi-symmetric MHD test problem that has an exact solution in a domain between two concentric spheres, and can thus be used to test the order of convergence of MHD solvers in this type of domains.

In this test problem, the exact solution is specified as

$$\mathbf{U}(x,y,z) = \left[\; r^{-\frac{5}{2}}, \;\; \frac{x}{\sqrt{r}}, \;\; \frac{y}{\sqrt{r}}, \;\; \frac{z}{\sqrt{r}} + \kappa r^{\frac{5}{2}}, \;\; \frac{x}{r^3}, \;\; \frac{y}{r^3}, \;\; \frac{z}{r^3} + \kappa, \;\; r^{-\frac{5}{2}} \; \right]^T, \tag{11}$$

and the volumetric source term $\mathbf{Q}$ in Eq.(1) includes the term $\mathbf{Q}_O$, which represents the residual obtained for the exact solution (Eq. 11). (Note that an exact solution of a modified equation system obtained in this way is sometimes called a 'manufactured solution' [32]). Here $\kappa = 0.017$ is a perturbation parameter chosen such that the solution has significant latitudinal variation yet the flow remains supersonic in the whole domain. As can be observed easily from Eq. (11), the velocity and the magnetic field have been chosen aligned everywhere such that $\vec{V}\times\vec{B} = 0$ and therefore, no source terms arise in the induction equation. Moreover, the magnetic field has been constructed by considering a background inverse squared radial function, $\vec{B}_0 = r^{-2}\hat{\mathbf{e}}_r$, plus a perturbation field, $\vec{B}_1 = \nabla f$, derived from the first-order spherical harmonic function $f(x,y,z) = \kappa z$, and it thus automatically satisfies the divergence-free condition. This magnetic field is also irrotational.

Thus, the final form of $\mathbf{Q}_O$ for a di-atomic gas is a function only of the space coordinates and can be written as

$$\mathbf{Q}_O = \begin{bmatrix} 0, \\ \frac{1}{2}xr^{-\frac{5}{2}}\left(r^{-1} - 5r^{-2} - \kappa z\right), \\ \frac{1}{2}yr^{-\frac{5}{2}}\left(r^{-1} - 5r^{-2} - \kappa z\right), \\ \frac{1}{2}zr^{-\frac{5}{2}}\left(r^{-1} - 5r^{-2} - \kappa z\right) + \frac{5}{2}r^{-\frac{1}{2}}\kappa(1+\kappa r z) + \kappa r^{-\frac{1}{2}}, \\ \vec{0}, \\ \frac{1}{2}r^{-2} + \kappa z(3.5r^{-1} + 2\kappa z) + \frac{(\kappa r)^2}{2}(7 + 5\kappa r z) \end{bmatrix}. \tag{12}$$

The computational domain used for this convergence study is defined by inner and outer spheres of radius $R_i = 2$ and $R_o = 3.5$, respectively. The inflow boundary condition is specified based on the exact solution and the outflow uses linear extrapolation.

The L_1, L_2, and L_∞ norms of the error in the predicted solution density at cell centroids obtained on a series of grids ranging in size from 8×8×10 to 256×256×320 cells for each of the six cubed-sphere blocks, which corresponds to 3,840 and 125,829,120 total cells, respectively, are given in Fig. 10 for this supersonic flow. As the mesh is refined, the slopes of the L_1-, L_2- and L_∞-norms approach in the asymptotic limit -2.06, -2.10 and -2.05, respectively. The results clearly show that the second-order theoretical accuracy of the scheme can also be achieved in all error norms for smooth but non-radial flows with a magnetic field, thereby providing support for the validity of the applied spatial discretization procedure on cubed-sphere grids.

5.2.2. Magnetically Dominated MHD Bow Shock

We now consider the application of the proposed AMR algorithm to 3D MHD bow-shock flows around a perfectly conducting sphere. In particular, we consider the magnetically dominated upstream conditions from [55] in conjunction with solution-adaptive cubed-sphere grids. The inflow parameters used for this problem have been chosen as in [55] and are $\rho = 1$, $p = 0.2$, $B_x = 1$, $B_y = 0$, $v_x = 1.4943$ and $v_y = 0.1307$, which correspond to an upstream plasma characterized by $\beta = 2p/B^2 = 0.4$, an Alfvénic Mach number $M_{Ax} = 1.49$ along the upstream magnetic field lines and an angle $\theta_{vB} = 5^\circ$ between the upstream velocity and magnetic vector fields. As shown in [55], this particular upstream configuration gives rise to 3D intermediate shocks and multiple interacting shock fronts, that are sought here to be well resolved by the adaptive mesh refinement procedure.

The computational domain for solving this problem is like the one depicted in Fig. 11 and is defined by inner and outer spheres of radius $R_i = 1$ and $R_o = 8$. Note that this mesh contains only five blocks instead of the typical six associated with a complete cubed-sphere mesh. This extra flexibility in our implementation allows for better resolution of the interesting flow features occurring on the upstream side of the sphere with a smaller domain and fewer computational cells as compared to the case in which the domain at the back of the sphere would have been included. In this grid, the back panel domain boundaries are tilted at 15° relative to the Cartesian (y,z) plane. Reflection boundary conditions (BCs) are imposed along the inner sphere while a free-stream boundary condition is applied to the outer one. Linear extrapolation BCs are implemented for the back panels of the outer boundary based on the fact that the flow is superfast at those locations and all eight characteristic waves leave the domain. The initial multi-block

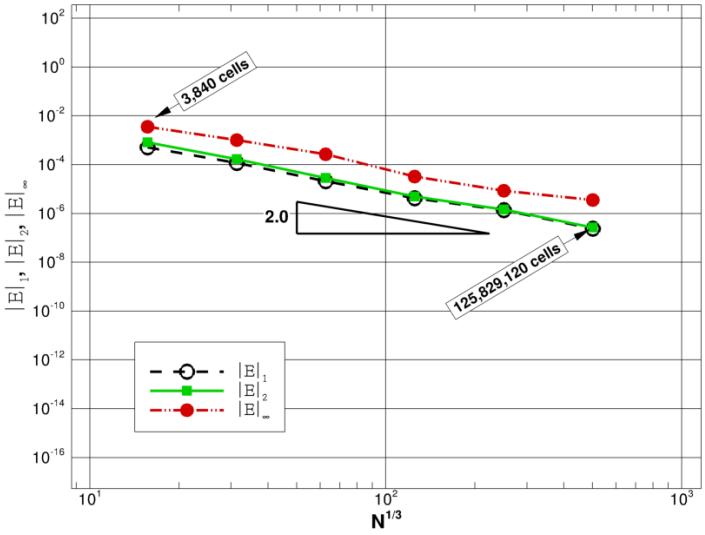


Figure 10: L_1, L_2, and L_∞ error norms in the predicted solution density for the manufactured MHD solution described in the text.

grid for this problem consists of a total of 20 8×8×10 blocks, with four blocks in the radial direction for each of the five sectors.

The predicted Mach number and density distribution in the Cartesian (x, y) plane obtained after 7 adaptive refinement levels on the final refined mesh consisting of 22,693 blocks and 14,523,520 computational cells is shown in Fig. 12. More detailed investigation of the numerical results (not shown) confirms that they are consistent with the results of [55]; all shocks arising in front of the sphere are captured by the refinement criteria and are well resolved by the 3D adaptation procedure, especially also in the region corresponding to the bifurcation of the shocks.

5.2.3. Time-Invariant Solar Wind Solution

Finally, the application of the cubed-sphere simulation framework to a more realistic space-physics problem is considered. This test problem represents an initial step towards developing more sophisticated models for the solar wind and carrying out more advanced and complete space physics studies with the algorithm described here.

A numerical solution of the "steady" solar wind representative of solar minimum conditions (i.e., for the quiet Sun) is obtained based on the global 3D MHD model of Groth *et al.* [2]. The steady-state solar wind from 1 R_s (i.e., solar radius) to 1 AU is modelled by assuming that, at 1 R_s, the inner solar corona is a large reservoir of hot plasma with an embedded magnetic multipole field described by a multipole expansion that includes terms up to the octupole moment (see [2, 56] for details). The plasma temperature (the sum of the ion and electron temperatures) of the reservoir is assumed to be $T_s = 2.85\times10^6$ K, and the plasma density is taken to be 1.5×10^{14} m^{-3}. The solar magnetic field is azimuthally symmetric about the magnetic axis which is aligned with the Cartesian z-coordinate direction. The multipole expansion has been chosen such as to obtain a maximum field strength of 8.4 G at the magnetic poles and a strength of 2.2 G at the solar magnetic equator. For these solar wind calculations, the MHD equations are solved for the perturbation of the magnetic field from the embedded ("intrinsic") magnetic multipole field, similarly to the approach used in Groth *et al.* [2] which follows the pioneering work of Tanaka [57]. The single-fluid plasma is assumed to have a particle mass equal to the proton mass, m_p, and a polytropic index $\gamma = 5/3$ is used everywhere in the domain. Note that more realistic solar-wind models have been previously considered which account for the rotation of the Sun and the tilting of the solar magnetic axis with respect to the ecliptic plane (see e.g., [2, 56]), and our approach can be extended to include these effects as well.

To reproduce a realistic solar wind using the ideal MHD description, which neglects the thermal conduction that is important in the vicinity of the Sun [58], and to allow the use of an ideal plasma gas with a polytropic index $\gamma = 5/3$ throughout the computational domain such that adiabatic cooling at larger heliospheric distances is correctly modelled, Groth *et al.* [2] have included an extra source term in the energy equation that mimics the effects of heat conduction

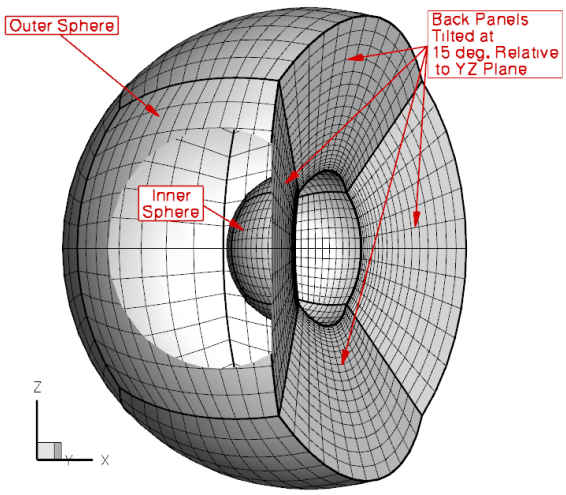


Figure 11: Cubed-sphere grid formed by only five root blocks, for simulation on one side of the sphere. A cut in the outer spherical geometry allows a better view of the inner spherical cap.

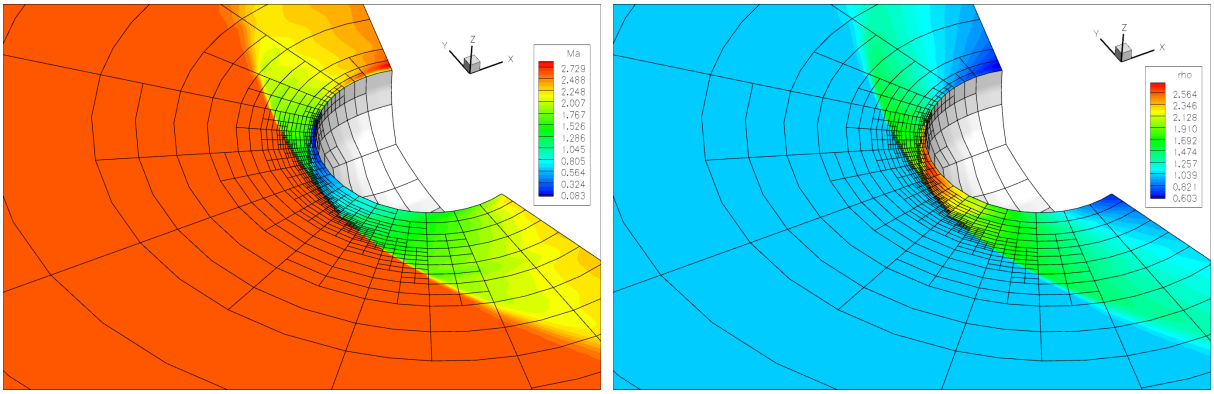
and energy dissipation above the transition region such as to reproduce solar wind characteristics determined by *in situ* observations. The global solar wind model with additional heating source term proposed by Groth *et al.* [2] has proved its use in several numerical simulations of coronal mass ejection (CME) evolution [2, 56, 59, 60] and the same heating term is used here as well. Thus, the heating source term $\mathbf{Q}_H$ is given by

$$\mathbf{Q}_H = -\rho \left[0 \, , \vec{0} \, , \vec{0} \, , \rho q \left(T_o - \gamma \frac{p}{\rho} \right) \right]^{\mathrm{T}} \, , \tag{13}$$

where $q(x, y, z)$ is an exponentially-decaying specific heat capacity function of the radial distance from the Sun and $T_o = T_o(x, y, z)$ is a prespecified "target" temperature which has a particular spatial distribution [56].

For this numerical simulation the computational domain is the spherical shell defined by inner and outer spheres of radius $R_i = 1$ and $R_o = 215$, which represent distances normalized by the solar radius, R_s. The cubed-sphere computational grid is stretched with a higher density of grid cells towards the inner sphere and consists of 768 self-similar 16×16×20 blocks and 3,932,160 cells, providing an average angular resolution of 1.4° and a minimum cell size at the solar surface of 1/4 R_s. Boundary conditions are implemented at the inner sphere, where the flow is subslow, according to the propagation of the characteristics, to conservation principles for radial mass and magnetic fluxes, and to physical conditions for the directions of velocity and magnetic field vectors in perfectly conducting fluids (see, e.g., [4] for details). Moreover, at the inner boundary linear extrapolation of density gradient is applied and a fixed solar corona temperature is enforced whereas the plasma is permitted to freely leave the reservoir, but no "backflow" is allowed. At the outer boundary, the flow is superfast and consequently all variables are simply extrapolated. The initial condition for the simulation is provided by Parker's isothermal solar wind [61] and the initial magnetic field is obtained from the aforementioned multipole expansion [2].

Figure 13 shows a meridional cut through the numerical solution of the steady-state solar wind obtained using the algorithm described above. The shading represents the magnitude of the velocity field, and the solid lines correspond to the predicted field lines which have been coloured based on the logarithm of the magnitude of the magnetic field, $\log_{10}(B)$. A three-dimensional representation of several magnetic field lines in the solar corona is shown in Fig. 14, in which the same colouring scheme as the one just mentioned was used. Additionally, this last figure depicts elements of the computational grid. Inspection of Fig. 13 reveals a bimodal outflow pattern with slow wind leaving the Sun near the equator, and high-speed solar wind of around 700 km/s in the regions of open magnetic field lines emanating from the coronal holes. This is an important feature of the solar wind which has been revealed by *in situ* observations [2, 62]. Other noticeable features present in the solution are the formation of a "helmet streamer" configuration and a heliospheric current sheet, the latter representing the relatively-thin equatorial region across which the magnetic field



(a) (b)

Figure 12: Predicted (a) acoustic Mach number and (b) density distribution in the Cartesian (x, y) plane for a magnetically dominated supersonic flow past a sphere obtained on the adapted cubed-sphere mesh after 7 refinement levels and with 22,693 blocks and 14,523,520 computational cells. Additionally, the block boundaries are depicted with solid black lines.

changes the polarity rapidly. It can also be observed that in the regions of closed field lines near the solar surface the predicted plasma velocity is very small and subsonic, in the range of 10-100 km/s.

The current problem also represents a practical test case for assessing the parallel performance of the proposed formulation on cubed-sphere grids by measuring/evaluating the parallel speedup and parallel efficiency relative to 48 processor cores defined as

$$S_p = \frac{t_{48}}{t_p}\,; \qquad E_p = \frac{S_p}{\frac{p}{48}}\,, \tag{14}$$

respectively, where t_{48} and t_p are the execution times required to solve the problem by 48 and by p processor cores, respectively. Here, the parallel speedup, also known as strong scaling, is measured by considering a fixed problem size of 6,144 8×8×8 blocks and a fixed number of 2,000 multi-stage explicit time steps, and performing the simulation on an increasing number of processor cores, p, up to $p = 3,072$. Note that a perfect speedup corresponds to the ideal situation in which the execution time of the simulation on p computing cores is $p/48$ times smaller than the reference time, t_{48}. In Fig. 15, the parallel speedup and efficiency achieved by the method as a function of the number of computing cores is plotted and compared with the corresponding ideal variations of the parallel-performance parameters. The results show that the multi-dimensional second-order finite-volume scheme applied to cubed-sphere grids and MHD flows is capable to provide very good parallel scalability up to 3,072 computing cores even though the computational blocks contain a relatively small number of cells, namely 512, and the amount of computational work per computing core decreases from 128 to a minimum of 2 blocks/core, or 1.56% of the initial value. Hence, the parallel efficiencies achieved by the numerical scheme are 0.923, 1.024, 1.011, 1.001, 0.956 and 0.960 on 96, 192, 384, 768, 1,536 and 3,072 computing cores, respectively. Note that the slightly higher values than one of several parallel efficiencies can be interpreted as being due to a cache effect, in which the increases in available cache for the program can overcome the computational cost of the additional inter-block parallel communication incurred. It is also important to note that the results depicted in Fig. 15 are for strong scaling tests, which are more challenging than weak scaling tests, in which the number of grid cells per processor is kept constant, and for which even better parallel efficiency can thus be expected.

6. Discussion and Concluding Remarks

A parallel block-based solution-adaptive simulation framework has been described for three-dimensional cubed-sphere grids and flows governed by hyperbolic conservation laws. In particular, the method has been developed for the Euler and MHD equations, targeting application to space-physics problems. The framework constitutes a significant

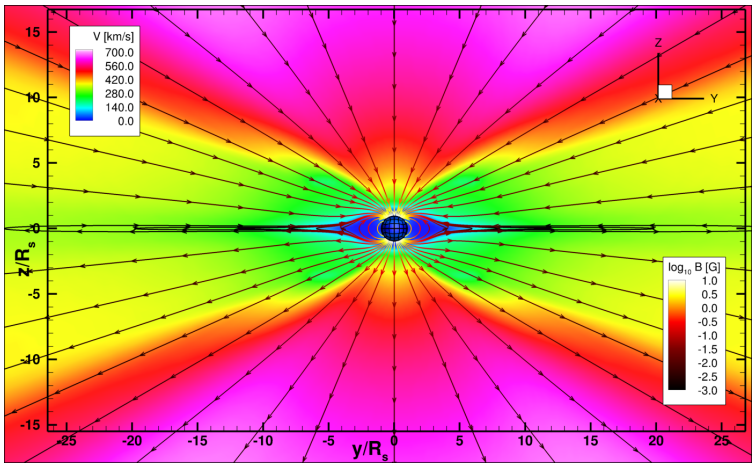


Figure 13: Representation of the steady-state solar wind solution in the (y,z) meridional plane. The colour shading represents the magnitude of the velocity vector, $V = \|\vec{V}\|_2$, in km/s and the solid lines are magnetic field lines coloured based on the magnitude of $\log_{10}(B)$. The arrows of the magnetic lines indicate the magnetic polarity chosen for this problem (lines leave from the North pole of the Sun and enter into the South pole). The boundaries of the mesh blocks at the solar surface are also shown in the figure.

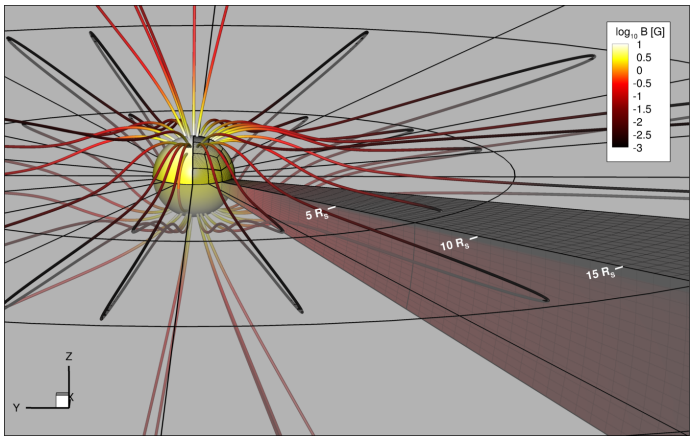


Figure 14: Three-dimensional representation of open and closed magnetic field lines in the corona region. Two blocks of the cubed-sphere grid and the associated meshes are completely shown together with the edges of the blocks in the equatorial plane and the edges and face nodes of several blocks at the inner sphere.

advance in 3D cubed-sphere grid simulation capabilities: it is the first 3D cubed-sphere grid framework with fully dynamical solution-adaptive grid refinement and coarsening, it is scalable up to several thousands of processor cores, and we show uniform second-order accuracy on a 3D cubed-sphere grid with the PDE system discretized uniformly in all three dimensions (i.e., without using a layered approach as is common in atmospheric applications). We have achieved this by employing a genuine and consistent multi-block approach with unstructured root connectivity (following [13]), and by utilizing a solution reconstruction method that is multi-dimensional with flexible stencil size [29], which allowed us to obtain a framework with uniform second-order accuracy, solution-adaptive refinement, and efficient parallelization that are fully transparent to the sector boundaries and corners of the cubed-sphere grid.

The accuracy of the algorithm has been assessed based on exact solutions, and the theoretically expected second-order accuracy has been achieved. A new fully supersonic axi-symmetric MHD test problem has been presented that has an exact solution in a domain between two concentric spheres, and can thus be used to test the order of convergence of MHD solvers in this type of domains. Furthermore, the predictive capabilities of the framework have been illustrated for other test problems from the literature, including a radial transonic wind flow, a standard solar-wind model and bow shock flows. We have also demonstrated the ability of the scheme to adequately handle and resolve

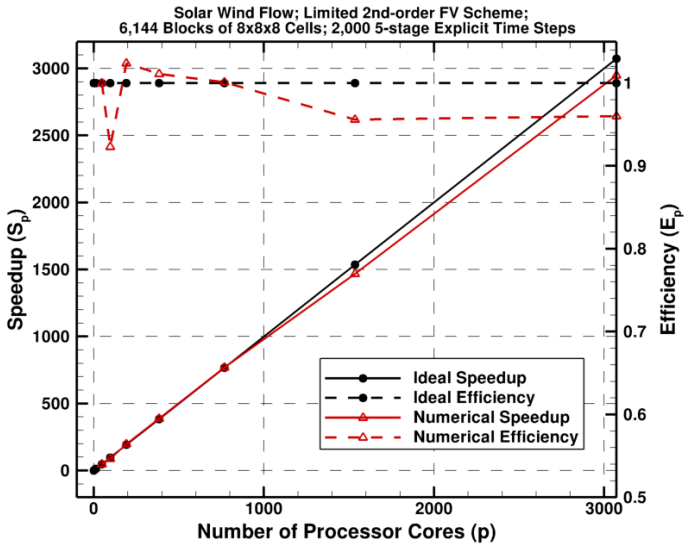


Figure 15: The parallel speedup and efficiency achieved for the time-invariant solar wind problem by the 2nd-order finite-volume algorithm on cubed-sphere grids as a function of the number of computing cores.

strong discontinuities/shocks with a number of computational elements that is reduced by adaptivity compared to uniformly refined grids. While this features have been demonstrated for the Euler and ideal MHD systems, it is believed that they will also carry over to other systems of conservation laws.

The parallel performance of the proposed 3D adaptive numerical procedure has been shown to be very high for problems with large numbers of solution blocks, computed on several thousands of processor cores. Future research will involve application of the algorithm to complex space-physics problems, the coupling with an effective parallel implicit algorithm (see, for example, the methods of Northrup and Groth [51, 52]) and the extension of the numerical scheme to high-order accuracy (i.e., accuracy higher than second-order) as considered by Ivan and Groth [41, 42, 63].

Acknowledgments

This work was supported by CSA (Canadian Space Agency) CGSM Contract No. 9F007-080157/001/ST. Computations were performed on the GPC supercomputer at the SciNet HPC Consortium. SciNet is funded by: the Canada Foundation for Innovation under the auspices of Compute Canada; the Government of Ontario; Ontario Research Fund - Research Excellence; and the University of Toronto.

References

[1] W. B. Manchester IV, A. Vourlidas, G. Tóth, N. Lugaz, I. I. Roussev, I. V. Sokolov, T. I. Gombosi, D. L. D. Zeeuw, M. Opher, Three-dimensional MHD Simulation of the 2003 October 28 coronal mass ejection: comparison with LASCO coronagraph observations, Astrophys. J. 684 (2008) 1448–1460.

[2] C. P. T. Groth, D. L. De Zeeuw, T. I. Gombosi, K. G. Powell, Global Three-Dimensional MHD Simulation of a Space Weather Event: CME Formation, Interplanetary Propagation, and Interaction with the Magnetosphere, J. Geophys. Res. 105 (A11) (2000) 25,053–25,078.

[3] 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, G. Tóth, Solution-Adaptive Magnetohydrodynamics for Space Plasmas: Sun-to-Earth Simulations, Comp. Sci. & Eng. 6 (2) (2004) 14–35.

[4] C. Jacobs, Magnetohydrodynamic modelling of the solar wind and coronal mass ejections, Ph.D. thesis, University of Leuven, 2007.

[5] C. Ronchi, R. Iacono, P. S. Paolucci, The "Cubed Sphere": A New Method for the Solution of Partial Differential Equations in Spherical Geometry, J. Comput. Phys. 124 (1996) 93–114.

[6] A. Adcroft, J.-M. Campin, C. Hill, J. Marhall, Implementation of an Atmosphere-Ocean General Circulation Model on the Expanded Spherical Cube, Mon. Weather Rev. 132 (2004) 2845–2863.

[7] W. M. Putman, S.-J. Lin, Finite-volume transport on various cubed-sphere grids, J. Comput. Phys. 227 (2007) 55–78.

[8] A. St-Cyr, C. Jablonowski, J. M. Dennis, H. M. Tufo, S. J. Thomas, A Comparison of Two Shallow-Water Models with Nonconforming Adaptive Grids, Mon. Weather Rev. 136 (2008) 1898–1922.
[9] P. A. Ullrich, C. Jablonowski, B. van Leer, High-order finite-volume methods for the shallow-water equations on the sphere, J. Comput. Phys. 229 (17) (2010) 6104 – 6134, doi: 10.1016/j.jcp.2010.04.044.
[10] A. V. Koldoba, M. M. Romanova, G. V. Ustyugova, R. V. E. Lovelace, Three-Dimensional Magnetohydrodynamic Simulations of Accretion to an Inclined Rotator: the "Cubed Sphere" Method, Astrophys. J. 576 (2002) L53–L56.
[11] P. C. Fragile, C. C. Lindner, P. Anninos, J. D. Salmonson, Application of the cubed-sphere grid to tilted black hole accretion disks, Astrophys. J. 691 (2009) 482–494.
[12] X. Gao, A Parallel Solution-Adaptive Method for Turbulent Non-Premixed Combusting Flows, Ph.D. thesis, University of Toronto, 2008.
[13] X. Gao, C. P. T. Groth, A parallel solution-adaptive method for three-dimensional turbulent non-premixed combusting flows, J. Comput. Phys. 229 (2010) 3250–3275.
[14] N. A. Phillips, Numerical Integration of the Primitive Equations on the Hemisphere, Mon. Weather Rev. (1959) 333–345.
[15] R. Sadourny, Conservative Finite-Difference Approximations of the Primitive Equations on Quasi-Uniform Spherical Grids, Mon. Weather Rev. 100 (2) (1972) 136–144.
[16] P. A. Ullrich, Atmospheric Modeling with High-Order Finite-Volume Methods, Ph.D. thesis, University of Michigan, 2011.
[17] 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, M. Opher, Adaptive numerical algorithms in space weather modeling, J. Comput. Phys. 231 (3) (2012) 870 – 903, URL http://www.sciencedirect.com/science/article/pii/S002199911100088X, doi:10.1016/j.jcp.2011.02.006.
[18] M. J. Aftomis, M. J. Berger, J. E. Melton, Robust and Efficient Cartesian Mesh Generation for Component-Base Geometry, AIAA J. 36 (6) (1998) 952–960.
[19] F. X. Giraldo, High-order triangle-based discontinuous Galerkin methods for hyperbolic equations on a rotating sphere, J. Comput. Phys. 214 (2006) 447–465.
[20] M. N. Levy, R. D. Nair, H. M. Tufo, High-order Galerkin methods for scalable global atmospheric models, Comput. & Geos. 33 (2007) 1022–1035.
[21] C. Yang, J. Cao, X.-C. Cai, A fully implicit domain decomposition algorithm for shallow water equations on the cubed-sphere, SIAM J. Sci. Comput. 32 (2010) 418–438.
[22] C. Jablonowski, M. Herzog, J. E. Penner, R. C. Oehmke, Q. F. Stout, B. van Leer, K. G. Powell, Block-Structured Adaptive Grids on the Sphere: Advective Experiments, Mon. Wea. Rev. 134 (2006) 3691–3713.
[23] K. G. Powell, P. L. Roe, T. J. Linde, T. I. Gombosi, D. L. De Zeeuw, A Solution-Adaptive Upwind Scheme for Ideal Magnetohydrodynamics, J. Comput. Phys. 154 (1999) 284–309.
[24] C. P. T. Groth, D. L. D. Zeeuw, K. G. Powell, T. I. Gombosi, Q. F. Stout, A Parallel Solution-Adaptive Scheme for Ideal Magnetohydrodynamics, Paper 99-3273, AIAA, 1999.
[25] B. van der Holst, R. Keppens, Hybrid block-AMR in cartesian and curvilinear coordinates: MHD applications, J. Comput. Phys. 226 (1) (2007) 925 – 946, doi:10.1016/j.jcp.2007.05.007.
[26] B. van der Holst, W. M. IV, I. Sokolov, G. Tth, T. Gombosi, D. DeZeeuw, O. Cohen, Breakout Coronal Mass Ejection or Streamer Blowout: The Bugle Effect, APJ 693 (2) (2009) 1178.
[27] M. L. Norman, The Impact of AMR in Numerical Astrophysics and Cosmology, in: Lecture Notes in Computational Science and Engineering, vol. 41, Springer Berlin Heidelberg, 413–430, 2005.
[28] W. M. Putman, Development of the Finite-Volume Dynamical Core on the Cubed-Sphere, Ph.D. thesis, The Florida State University, 2007.
[29] T. J. Barth, D. C. Jespersen, The Design and Application of Upwind Schemes on Unstructured Meshes, Paper 89-0366, AIAA, 1989.
[30] S. K. Godunov, Symmetric Form of the Equations of Magnetohydrodynamics, Numerical Methods for Mechanics of Continuum Medium, Siberian Branch of USSR Academy of Sciences 1 (1972) 26–34.
[31] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, M. Wesenberg, Hyperbolic Divergence Cleaning for the MHD Equations, J. Comput. Phys. 175 (2) (2002) 645 – 673, doi:10.1006/jcph.2001.6961.
[32] P. J. Roache, Verification and Validation in Computational Science and Engineering, Hermosa Publisher, New Mexico, 1998.
[33] J. J. Quirk, U. R. Hanebutte, A Parallel Adaptive Mesh Refinement Algorithm, Report 93-63, ICASE, 1993.
[34] M. J. Berger, J. S. Saltzman, AMR on the CM-2, Appl. Numer. Math. 14 (1994) 239–253.
[35] M. J. Berger, J. Oliger, Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations, J. Comput. Phys. 53 (1984) 484–512.
[36] M. J. Berger, P. Colella, Local Adaptive Mesh Refinement for Shock Hydrodynamics, J. Comput. Phys. 82 (1989) 67–84.
[37] Q. F. Stout, D. L. De Zeeuw, T. I. Gombosi, C. P. T. Groth, H. G. Marshall, K. G. Powell, Adaptive Blocks: A High-Performance Data Structure, in: Proceedings of SC97, San Jose, California, U.S.A., November 12–15, 1997.
[38] C. P. T. Groth, D. L. De Zeeuw, T. I. Gombosi, K. G. Powell, A Parallel Adaptive 3D MHD Scheme for Modeling Coronal and Solar Wind Plasma Flows, Space Sci. Rev, 87 (1999) 193–198.
[39] J. S. Sachdev, C. P. T. Groth, J. J. Gottlieb, A Parallel Solution-Adaptive Scheme for Predicting Multi-Phase Core Flows in Solid Propellant Rocket Motors, Int. J. Comput. Fluid Dyn. 19 (2).
[40] X. Gao, C. P. T. Groth, A parallel adaptive mesh refinement algorithm for predicting turbulent non-premixed combusting flows, Int. J. Comput. Fluid Dyn. 20 (5) (2006) 349–357.
[41] L. Ivan, C. P. T. Groth, High-Order Central ENO Finite-Volume Scheme with Adaptive Mesh Refinement, Paper 2007-4323, AIAA, 2007.
[42] L. Ivan, C. P. T. Groth, High-Order Solution-Adaptive Central Essentially Non-Oscillatory (CENO) Method for Viscous Flows, Paper 2011-0367, AIAA, 2011.
[43] R. Keppens, Z. Meliani, A. J. van Marle, P. Delmont, A. Vlasis, B. van der Holst, Parallel, grid-adaptive approaches for relativistic hydro and magnetohydrodynamics, J. Comput. Phys. 231 (3) (2012) 718–744.
[44] NASA, CGNS, CFD General Notation System, retrieved from <http://www.grc.nasa.gov/www/cgns/> (Accessed Mar. 26, 2012).
[45] S. K. Godunov, Finite-Difference Method for Numerical Computations of Discontinuous Solutions of the Equations of Fluid Dynamics, Mat. Sb. 47 (1959) 271–306.

[46] G. Tóth, D. Odstrčil, Comparison of Some Flux Corrected Transport and Total Variation Diminishing Numerical Schemes for Hydrodynamic and Magnetohydrodynamic Problems, J. Comput. Phys. 128 (1996) 82–100.
[47] A. Harten, P. D. Lax, B. van Leer, On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws, SIAM Rev. 25 (1) (1983) 35–61.
[48] T. Linde, A practical, general-purpose, two-state HLL Riemann solver for hyperbolic conservation laws, Int. J. Numer. Meth. Fluids 40 (2002) 391–402.
[49] V. Venkatakrishnan, On the Accuracy of Limiters and Convergence to Steady State Solutions, Paper 93-0880, AIAA, 1993.
[50] B. van Leer, C. H. Tai, K. G. Powell, Design of Optimally-Smoothing Multi-Stage Schemes for the Euler Equations, Paper 89-1933-CP, AIAA, 1989.
[51] S. A. Northrup, C. P. T. Groth, Prediction of Unsteady Laminar Flames Using a Parallel Implicit Adaptive Mesh Refinement Algorithm, in: Proceedings of the 6th U.S. National Combustion Meeting, 2009.
[52] S. A. Northrup, C. P. T. Groth, Parallel Implicit AMR Scheme for Unsteady Reactive Flows, in: Proceedings of the 18th Annual Conference of the CFD Society of Canada, London, Canada, 2010.
[53] D. J. Mavriplis, Multigrid Techniques for Unstructured Meshes, Report 95-27, ICASE, 1995.
[54] H. De Sterck, S. Rostrup, F. Tian, A fast and accurate algorithm for computing radial transonic flows, J. Comput. Appl. Math. 223 (2) (2009) 916–928.
[55] H. De Sterck, S. Poedts, Intermediate Shocks in Three-Dimensional Magnetohydrodynamic Bow-Shock Flows with Multiple Interacting Shock Fronts, Phys. Rev. Lett. 84 (24) (2000) 5524–5527.
[56] W. B. Manchester IV, T. I. Gombosi, I. I. Roussev, A. Ridley, D. L. D. Zeeuw, I. V. Sokolov, K. G. Powell, Modeling a space weather event from the Sun to the Earth: CME generation and interplanetary propagation, J. Geophys. Res. 109 (A0) (2004) 2107–2121.
[57] T. Tanaka, Finite Volume TVD Scheme on an Unstructured Grid System for Three-Dimensional MHD Simulation of Inhomogeneous Systems Including Strong Background Potential Fields, J. Comput. Phys. 111 (2) (1994) 381 – 389, doi:10.1006/jcph.1994.1071.
[58] N. Meyer-Vernet, Basics of the Solar Wind, Cambridge University Press, URL http://dx.doi.org/10.1017/CBO9780511535765, 2007.
[59] N. Lugaz, W. B. Manchester IV, T. I. Gombosi, The evolution of coronal mass ejection density structures, Astrophys. J. 627 (2005) 1019–1030.
[60] C. Jacobs, B. van der Holst, S. Poedts, Comparison between 2.5D and 3D simulations of coronal mass ejections, Astronomy and Astrophysics 470 (2007) 359–365.
[61] E. N. Parker, Dynamics of the Interplanetary Gas and Magnetic Fields, Astrophys. J. 128 (3) (1958) 664–676.
[62] T. E. Cravens, Physics of Solar System Plasmas, Cambridge University Press, New York, 1997.
[63] L. Ivan, Development of High-order CENO Finite-volume Schemes with Block-based Adaptive Mesh Refinement, Ph.D. thesis, University of Toronto, URL http://hdl.handle.net/1807/29759, 2011.