High-Order Finite-Volume Schemes for Magnetohydrodynamics

by

Andree Susanto

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Applied Mathematics

Waterloo, Ontario, Canada, 2014

© Andree Susanto 2014

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

My PhD thesis research was conducted under the supervision of Prof. Hans De Sterck, and in collaboration with postdoctoral researcher Dr. Lucian Ivan. My research has been implemented in a large-scale parallel and adaptive simulation framework that was initially developed for compressible gas dynamics in the research group of Prof. Groth at the University of Toronto, focusing on combustion simulations. Since 2009, this simulation framework has been further developed in a collaboration between Prof. De Sterck's group at the University of Waterloo and the Toronto group. Work at Waterloo focuses on MHD, cubed-sphere grids, and extension of high-order discretizations to MHD and 3D adaptive grids. My PhD thesis contributes to this collaborative effort.

The original research contributing to my PhD thesis is described in chapters 3 to 6 and Appendix B of this thesis.

The work presented in chapters 3 and and 4 was published in several journal and conference papers [1, 2, 3]. I am the lead author of a paper on 2D ideal MHD simulation [1], which forms the basis for Chapter 3. I was responsible for extending the 2D CENO method to MHD, including implementing the parabolic-hyperbolic divergence correction method and boundary conditions, and integrating it into the simulation framework. I defined and implemented the test problems and executed the convergence tests for the work described in Chapter 3. I am a co-author on papers [2, 3], which introduce 3D high-order CENO methods for hyperbolic conservation laws on block-structured grids with general hexahedral cells. The lead author for these papers is Dr. Ivan, who developed high-order CENO discretizations on cubed-sphere grids (addressing the difficulty posed by non-planar cell surfaces) and oversaw the project. My contributions were in formulating, implementing and conducting 3D high-order MHD tests on Cartesian grids, generating the inner part of the cubed-sphere mesh (Appendix B), and co-developing rotation algorithms for generating stencils near degenerate edges of the cubed-sphere grids.

I am the lead author of the work presented in chapters 5 and 6, which, as of the completion of this thesis, has not been submitted for publication yet. Chapter 5 describes extension of the high-order CENO method to resistive MHD, which I developed, implemented and tested under the supervision of Prof. De Sterck and Dr. Ivan. Chapter 6 applies the parallel MHD simulation framework to simulations of the interaction between the solar wind and the Moon. I formulated the test problems, implemented them, and executed the tests, under the supervision of Prof. De Sterck.

Abstract

New high-order finite-volume numerical schemes for the magnetohydrodynamics equations are proposed in two and three dimensions. Two different sets of magnetohydrodynamics equations are considered. The first set is the ideal magnetohydrodynamics system, which assumes that the fluid can be treated as a perfect conductor. The second set is resistive MHD, which involves non-zero resistivity. A high-order central essentially non-oscillatory (CENO) approach is employed, which combines unlimited k-exact polynomial reconstruction with a monotonicity preserving scheme. The CENO schemes, which were originally developed for compressible fluid flow, are applied to the MHD equations, along with two possible control mechanisms for divergence error of the magnetic field. The hyperbolic fluxes are calculated by solving a Riemann problem at each cell interface, and elliptic fluxes are computed through k-exact gradient interpolation where point-wise values of the gradients are required. Smooth test problems and test cases with discontinuities (weak or strong) are considered, and convergence studies are presented for both the ideal and resistive MHD systems. Several potential space physics applications are explored. For these simulations, cubed-sphere grids are used to model the interaction of the solar wind with planetary bodies or their satellites. The basic cubed-sphere grid discretizes a simulation domain between two concentric spheres using six root blocks (corresponding to the six faces of a cube). Conditions describing the atmosphere of the inner body can be applied at the boundary of the inner sphere. For some problems we also need to solve equations within the inner sphere, for which we develop a seven-block cubed-sphere grid where the empty space inside the interior sphere is discretized as a seventh root block. We consider lunar flow problems for which we employ the seven-block cubed-sphere mesh. Ideal MHD is solved between the inner and outer spheres of the grid, and the magnetic diffusion equations are solved within the inner sphere, which represents the lunar interior. Two cases are considered: one is without intrinsic magnetic field, where only a wake is expected without any bow shock forming ahead of the Moon, and the second is with a small dipole moment to model a lunar crustal magnetic anomaly, in which case a small-scale magnetosphere is expected ahead of the region with the magnetic anomaly.

Acknowledgements

So many great individuals have contributed to the completion of this thesis; without them, I would never have been able to finish this work. I would first and foremost like to express my gratitude to my PhD supervisor, Professor Hans De Sterck, for his expert guidance in magnetohydrodynamics, physics and scientific computing. This thesis is mine as much as it is his. Special thanks go to Dr. Lucian Ivan for his extensive help in coding and programming, and his patience in providing me technical assistance with CFFC, Tecplot and C++.

I would like to also thank Professor Clinton Groth for his feedback on our papers, and his involvement in the development of the CFFC framework. I would like to thank my PhD committee members, Professor Michael Waite and Professor Lilia Krivodonova and my external examiners, Professor Cécile Devaud and Professor Paul Charbonneau. I would like to thank my undergraduate and my Master's supervisors: Professor Dana Dabiri and Professor Uri Shumlak; thanks go to them for introducing me to the world of research and academia. I would like to thank and acknowledge Dr. Ammar Hakim for showing me a good modular programming practice and his generous programming help while I was working at Washington. Professionally, I would like to thank Barton Satchwill, Professor Robert Rankin, Everett Toews, May Lynn Lee and the rest of Cybera team for letting us participate and collaborate on the cloud project.

I would like to thank Helen Warren, Rina Salazar, Stephanie Martin, Maureen Fraser, Helen Chen, Cyntia Brătan, Laura Frazee, Carol Morrison, and Marie O'Brien-Stockie for taking care of a lot of administrative work.

The pursuit of PhD is professional as much as it is personal, and I would like to start acknowledging those whom I know on a personal level as well. I would like to acknowledge Tante Linda and Om Kim Koo for being family outside of home. Thanks a lot for all the caring and support you have provided me during these years.

I would like to thank my close friend, Kavin Sin, and his significant other, Michelle Nurwandi. Thank you so much for providing much needed friendship and connections. One could not have hoped for much better friends. Likewise, I would like to express my gratitude to my brothers and sister in the Sin family: Koeun Sin, Dr. Sopheaktra Sin, Jamie Sin-Keo, Peter, David, Mach Nhean, SG uncle, and the rest.

To Jeroen De Maeseneire, thanks a lot for showing the value of hard work and for providing much needed entertainment in the form of complaining. And I would like to thank and acknowledge the rest of the $09 - 10$ computational math group: Mike Sun Xiaolu, Johann Setiawan and Daniel Cui. Thank you for a lot of interesting conversations.

To Dr. Venkata Satya Kumar Manem, with whom I have mostly disagreed, and to his significant other, Pampa Dey, with whom I have always had to agree, I would like to express my gratitude for our friendship (and discussions and debates) all these years. To

Dr. Dhanaraja Kasinathan and Mrs. Nithya Agatheeswaran Dhanaraja, I am grateful to have known you two and for putting up with all the trouble and bother I may have caused over the years.

I would like to mention Dr. Puneet Sharma and his wife, Mrs. Akanksha Sharma, and their children, Saumya Sharma and Shreeya Sharma, for their useful and resourceful advices, for providing hospitality on the so many times I have come to visit and bother them at their place and for much needed friendship over the years.

I would like to thank my friend and my mentor, Dr. Mohamad Alwan. Thanks to him for tolerating my questionable TA-ship skills, and for various and numerous discussions on morality, religion, politics, football and everything else under the sun and the kitchen sink.

Thanks also go to my friend, Abdulhamed Alsisi, for several all-nighters we pulled together, for all the lively discussions and debates on politics and religion, and also for all the food given me.

I would like to thank Dr. Zhang Hongtao, Dr. Vaibhav Madhok, Subasha Wickramarachchi, and Daniel Otero for being such a good company all these years. Shivang Vyas, you have been an embodiment of a troll. These years would have been orders of magnitude more boring without your constant trolling.

I would like to mention Vivek Kant, Professor Li Zicheng, Dr. Moosa Ayati, Tung Hoang, Dr. Ravindra Reddy, Nikky Pathak, Shilpa Reddy Pantula, Andy Jiang, Matt Shen, Victor Feng, Vibhor Arora, Steve George, Avinash Singh Tomar, Manu and Divya Hegde, Dr. Liu Wentao, Dr. Alex Shum, Dr. Hamid Molavian, Dr. Rahul Sharma, Dr. Mukesh Meshram, Dr. Ahmet Ozkan Ozer, Arman Tavakoli, Amir Issaei, Dr. Anju Mary Philip, Zhang Kexue, Yusuf Kamil, Mahyar Shafiei, Anoop Jagannath, Colin Phipps, Sina Khani, Taghreed Sugati, Art of Living Kitchener, and numerous other individuals in and outside of the department, in or outside of the university. This list will be far too long if I was to list everybody I have been indebted to, and to those whom I have missed I sincerely apologize.

Special thanks to my family, Emi and Freddy Susanto, Anthonius Susanto and Eric Susanto, for supporting me in everything that I have done, everything that I do, and everything that I will do; and to CPS for literally putting colors on an otherwise very dull and grey period of my life during the thesis writing phase.

"If I have seen further, it is by standing on the shoulders of giants"

– Sir Isaac Newton

Dedication

I dedicate this thesis to the body of knowledge of humanity.

"To know, is to know that you know nothing. That is the meaning of true knowledge."

– A Socratic Paradox

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

The magnetohydrodynamic (MHD) equations combine the Euler equations of gas dynamics and the Maxwell's equations of electrodynamics. These equations describe electrically conducting fluids, which include space physics plasmas. Due to the highly nonlinear nature of these equations, it is usually impossible to find analytical solutions to the MHD system of equations. Numerical simulation, however, has in recent time become a popular means by which the magnetohydrodynamic equations are solved. Numerical simulations of these phenomena are often expensive, even with the help of supercomputers. One possible way to reduce this cost is to employ advanced simulation methods, in particular, methods with order of accuracy greater than two. The contribution of this thesis is in the development of high-order methods for MHD which are relevant for space physics simulations, and in illustrating how the resulting methods can be used to simulate space physics flows such as the interaction of the solar wind with the Moon.

1.1.1 Magnetohydrodynamics Models for Space Physics

The magnetohydrodynamic (MHD) equations describe electrically conducting fluids, which include plasmas and liquid metal. In particular, the ideal MHD equations are applicable for fluids which are highly collisional, with particle distributions that are Maxwellian, have low resistivity and have much larger length scales than the ion skin depth and Larmor radius perpendicular to the magnetic field [5]. Despite the fact that any MHD model ignores most of the kinetic effects, hence, is a low-order approximation of plasma behaviour, MHD models have become popular within the computational scientist and astrophysicist communities, e.g. to simulate space physics phenomena, and have proven to be extremely valuable and insightful in furthering our understanding of such processes [6].

1.1.2 High-Order Finite-Volume Method to Improve Accuracy of MHD Simulations

Consider, for example, simulations of space weather, which are an important class of MHD space physics simulations. For space weather forecasting models to be of practical use, it is imperative that the model is capable of running simulations faster than real-time (about three to four days), which necessitates high-performance computing. But simulations are often computationally very expensive, owing to the large variations in temporal and spatial scales of the system. It is mentioned in [7] that the coronal mass ejection, CME, phenomena can easily span over nine orders of magnitude in scale, which leads to a need for extremely fine grid resolutions. Coupled with small time steps, which are required to keep consistency in temporal accuracy, and also due to stability limitations, space-physics simulations can be prohibitively expensive, even for supercomputers.

The MHD equations can be written as a system of conservation laws with source term, which, in divergence form, is given as

$$\frac{\partial \vec{\mathbf{U}}}{\partial t} + \nabla \cdot \vec{\mathbf{F}} = \vec{\mathbf{S}}. \tag{1.1}$$

Consequently, we can then apply numerical schemes developed for such systems (for example, finite-volume methods). In particular, a high-order central essentially non-oscillatory (CENO) finite-volume scheme proposed in [4], which is suitable for solving systems of conservation laws, is applied to MHD systems. A preliminary study of the CENO method as applied to a simple convection problem produced very encouraging results. In [4], the authors compared the errors of a periodic linear advection of density variation for the Euler equations of gas dynamics, solved using the second- and the fourth-order CENO schemes. The density variation of the problem is given as $\rho(x, y) = 1.0 + 0.5 \cos(\pi\mathcal{T}(x))$ $\sin(5\pi\mathcal{T}(X))$, where $\mathcal{T}$ is the linear mapping function of the domain $[-100, 100]$ to $[-1, 1]$ (see Fig. 1.1). Table 1.1 shows that the second-order method requires about $80,000$ grid cells to match the level of accuracy of the fourth-order scheme using only $4,000$ grid cells. The time saving is quite significant, with the second-order method requiring more than twelve hours to finish the simulation, whereas the fourth-order method requires less than twenty minutes. At least for this particular test case (smooth, periodic, Euler equations), the benefit is clear, and it is hoped that one can gain a similar benefit for MHD, hence space physics applications, as well.

1.1.3 Cubed-Sphere Meshes to Represent Conditions Around a Planetary Body

In this, we are mostly interested in performing numerical simulations of space physics phenomena, in particular, the interaction between the solar wind, and planetary or lunar

Table 1.1: L_2-norm error, time and memory requirement comparison of the periodic Euler advection problem (results from [4])

# Cells		$\mathcal{O}(\Delta x^2)$ Limited	$\mathcal{O}(\Delta x^4)$ CENO
4,000	L_2:	3.33×10^{-2}	2.19×10^{-4}
(200×20)	$Time(h)$:	0:01:48	0:18:05
	$Mem(kB)$:	20,336	31,232
80,000	L_2:	2.25×10^{-4}	-
(4000×20)	$Time(h)$:	12:27:14	-
	$Mem(kB)$:	203,680	-

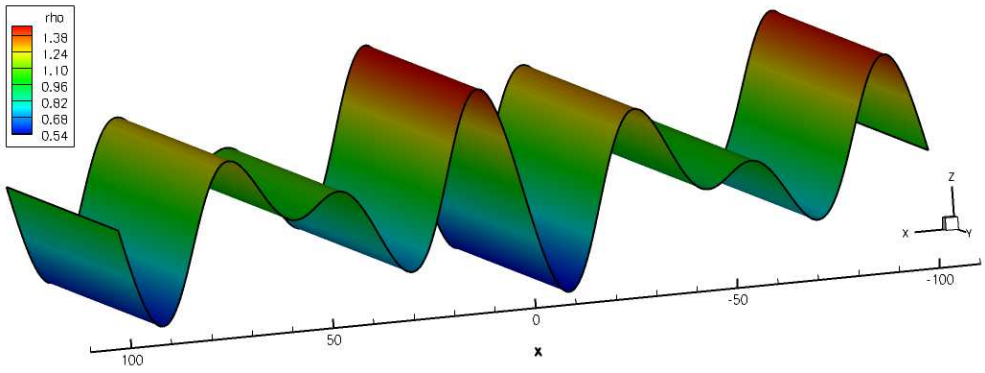


Figure 1.1: Density variation for periodic linear advection (Euler equations), with $\rho(x, y)$ = 1.0 + 0.5 $\cos(\pi\mathcal{T}(x))$ $\sin(5\pi\mathcal{T}(X))$. $\mathcal{T}(x)$, here, is the linear mapping function from the [-100,100] domain to the [-1,1] domain. The velocity for this problem is constant, and one-directional, so the solution will just advect. Because the boundaries are periodic, it is expected to recover the initial conditions exactly after 1 period (Figure from [4]).

bodies. Since most planetary bodies possess a spherical shape, it may be appropriate to consider spherical shell domains for many of these simulations. In particular, numerical domains which are defined as the space between two concentric spheres are often of interest and have recently become very popular in the field of computational physics [8]. While it may seem natural to consider meshes based on the spherical coordinate system, the fact that the grid would be generated using lines of constant longitude and latitude may lead to difficulties in dealing with polar singularities.

Phillips, in [9], provides desirable criteria for coordinate mapping of a spherical geometry, and one of the features put forward is that the mesh should be free of singularities. A grid based on generating lines of constant latitude and longitude would not satisfy this requirement. This pole problem was also briefly discussed in [10] and [11]. Ronchi [12] proposed a grid based on transforming a cube onto a spherical surface through six different coordinate transformations, where each face of the cube gets mapped to a face of the sphere. This transformation is illustrated more clearly in Fig. 1.2. The cubed-sphere mesh

is attractive, because not only does it satisfy the four criteria set forth in [9], but is also very competitive in terms of accuracy and scalability. While meshes based on the spherical coordinate system suffer from what is commonly referred to as the pole problem, where grid cell sizes around the poles are much smaller than those closer to the equator, the issue is much more manageable on a cubed sphere, where the smallest and the largest of cells differ in size by just around a factor of 2 [12]. The cubed-sphere mesh has gained immense popularity in recent years, with fields ranging from climate modeling and atmospheric science [13, 14, 15, 16, 17, 18, 19, 20, 21], to MHD [3, 8, 22] and astrophysics [23, 24].

The cubed-sphere grid is defined on a volumetric domain between two concentric spheres. The mesh is defined by two spherical radii: R_i, or the inner radius, which may represent the atmosphere of the planetary body in question, and R_o, or the outer radius, which denotes the outer extent of the simulation domain, usually where the incoming solar-wind conditions are defined. The space in between these two spheres is filled with concentric spheres with radii in between R_i and R_o. Keep in mind that this arrangement leads to a hollow sphere geometry, with an empty space in the middle the size of a sphere with a radius R_i. This arrangement of a cubed sphere is also called the regular cubed sphere, or the "6-block" cubed-sphere grid, and has been useful in simulating several space physics application [2, 3, 8]. Boundary conditions can then be readily applied on the inner sphere, which usually represent the atmospheric conditions of the inner body. While the 6-block cubed-sphere grid is suitable in cases with simple boundary conditions, some problems may also require solving equations inside the inner sphere. An example of such problem is the interaction between the solarwind and the lunar surface, where magnetic field lines are expected to diffuse away within the inner body, in which case a diffusion equation needs to be solved therein.

1.2 Thesis Structure

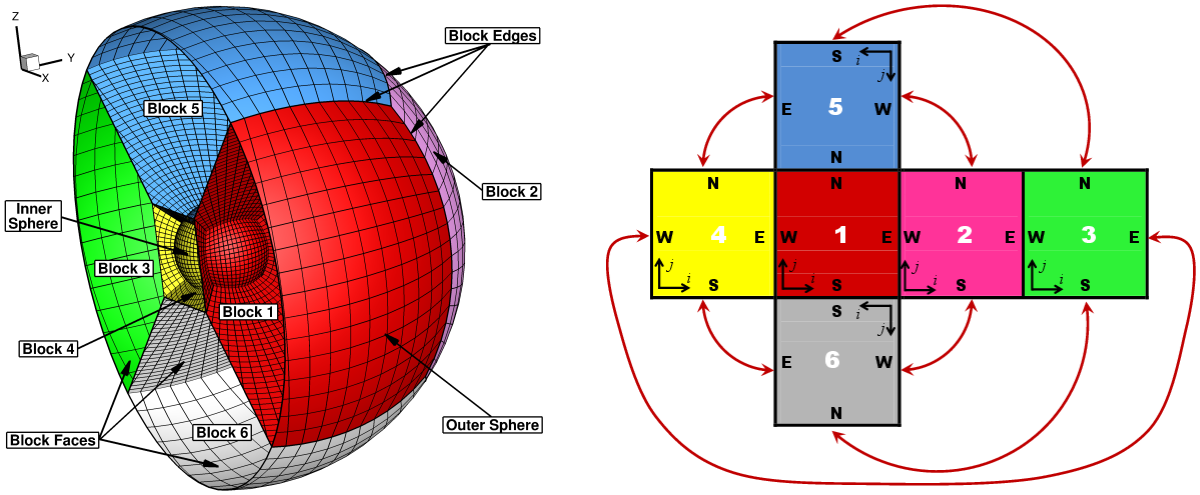
The main theme of the work presented in this thesis is numerical methods for space physics simulation. The work done for the purpose of this thesis can be categorized into several encompassing main ideas:

1. Formulation and implementation of a high-order CENO finite-volume scheme in two and three dimensions for MHD;

2. Utilization and implementation of cubed-sphere meshes to model conditions surrounding a planetary body or satellite;

3. Convergence studies of simulations conducted with high-order schemes and their comparisons with lower-order solutions;

4. Preliminary exploration of space physics applications



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

(b) Illustration of unstructured connectivity among root blocks

Figure 1.2: Three-dimensional cubed-sphere grid with six root blocks (corresponding to the six sectors of the grid) and depiction of inter-block connectivity. In panel (b), the block faces are denoted with the initials of the cardinal directions: North (N), East (E), South (S) and West (W). (Figures from [8].)

In accordance with the aforementioned main ideas, this thesis is broken down as follows. We describe the ideal and the resistive MHD equations in Chapter 2, with detailed discussions of divergence control techniques presented in Sec. 2.2. Chapter 3 and Chapter 4 deal with the formulation and implementation of the CENO finite-volume schemes for the ideal MHD equations in two and three dimensions respectively. These chapters present convergence studies of test cases in 2D and in 3D, which compare the accuracy of the high-order scheme with those of a second-order scheme. Problems with discontinuities are also presented. We also propose MHD extension to the Shu-Osher problem [25] in Sec. 3.2.2.2 and Sec. 4.2.2.2, which demonstrates the capability of the high-order scheme to handle highly oscillatory flow features in the presence of shocks. Chapter 5 extends the CENO scheme to the resistive MHD system. Since it is hard to analytically solve the resistive MHD system of equations, we propose a manufactured test case where some form of solutions has been assumed, and manufactured source terms are added to the system. Comparison of the errors between solutions produced by the second-order and high-order schemes are provided in the form of a convergence study. Finally, some preliminary space physics applications are presented in Chapter 6. Conclusions of the work are then presented in Chapter 7, with some directions for potential future work proposed in Sec. 7.1.

Chapter 2

Magnetohydrodynamics Equations

The equations that are of interest for this work are the magnetohydrodynamic equations, which will be covered in this chapter. We first describe two forms of the magnetohydrodynamics system in Sec. 2.1. Sec. 2.2 will briefly discuss the issues with numerically preserving the solenoidality constraint of the magnetic field [26], and will describe several techniques to control the divergence error of the magnetic field.

2.1 Magnetohydrodynamics Equations

In this section, magnetohydrodynamics equations are described. For space physics applications, diffusion times over any length scale of interest usually are on the order of hundreds of thousands of years. Since numerical simulations involve a physical time-scale that is much smaller than this diffusion time scale, it is usually reasonable to ignore the resistivity altogether. The class of MHD equations with no resistivity is called the ideal MHD system, and this system is described in Sec. 2.1.1. For slightly resistive fluids (such as when the interaction of the solarwind with the lunar atmosphere is considered), and for cases where kinetic effects are still not deemed significant, a resistivity term can be added directly to Ohm's law. This model is called the resistive MHD model, which is described in Sec. 2.1.2.

2.1.1 Ideal MHD Equations

The dimensionless ideal MHD system is described by the following equations in conservation form (refer to Appendix A for more details on the normalization procedure):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0, \tag{2.1}$$

$$\frac{\partial(\rho\vec{v})}{\partial t} + \nabla \cdot \left(\rho\vec{v}\vec{v} + \vec{I}(p + \frac{\vec{B}\cdot\vec{B}}{2}) - \vec{B}\vec{B} \right) = 0, \tag{2.2}$$

$$\frac{\partial \vec{B}}{\partial t} + \nabla \cdot (\vec{v}\vec{B} - \vec{B}\vec{v}) = 0, \tag{2.3}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot \left((e + p + \frac{\vec{B}\cdot\vec{B}}{2})\vec{v} - (\vec{v}\cdot\vec{B})\vec{B} \right) = 0. \tag{2.4}$$

Equations 2.1 to 2.4 are supplemented with a solenoidality condition for the magnetic field,

$$\nabla \cdot \vec{B} = 0. \tag{2.5}$$

The conserved quantities of the ideal MHD equation system are the density, ρ, the momentum, $\rho\vec{v}$ (with $\vec{v}$ being the velocity), the magnetic field, $\vec{B}$, and the energy, e. The plasma pressure, p, is given by the equation of state for a perfect gas

$$p = (\gamma - 1)\left(e - \frac{1}{2}\rho|\vec{v}|^2 - \frac{1}{2}|\vec{B}|^2 \right), \tag{2.6}$$

where γ is the adiabatic index. We use $\gamma = 5/3$ in our numerical tests except where noted.

2.1.1.1 MHD Wavespeeds and Eigenvalues

The ideal MHD system features three different waves, which are the Alfvén wave, the slow magnetosonic wave, and the fast magnetosonic wave. The speeds of these waves are given as

$$c_{f_x}^2 = \frac{1}{2}\left(\frac{\gamma p + B^2}{\rho} + \sqrt{(\frac{\gamma p + B^2}{\rho})^2 - 4\frac{\gamma p B_x^2}{\rho^2}} \right), \tag{2.7}$$

$$c_{s_x}^2 = \frac{1}{2}\left(\frac{\gamma p + B^2}{\rho} - \sqrt{(\frac{\gamma p + B^2}{\rho})^2 - 4\frac{\gamma p B_x^2}{\rho^2}} \right), \tag{2.8}$$

$$c_{A_x}^2 = \frac{B_x^2}{\rho}, \tag{2.9}$$

where c_{f_x} is the fast magnetosonic wavespeed, c_{s_x} is the slow magnetosonic wavespeed, and c_{A_x} is the Alfvén wavespeed. The MHD eigenvalues can then be computed as functions of the aforementioned wavespeeds, as given by

$$\lambda_{1,2} = v_x \pm c_{f_x}, \tag{2.10}$$

$$\lambda_{3,4} = v_x \pm c_{A_x}, \tag{2.11}$$

$$\lambda_{5,6} = v_x \pm c_{s_x}, \tag{2.12}$$

$$\lambda_7 = v_x, \tag{2.13}$$

$$\lambda_8 = 0. \tag{2.14}$$

The presence of the zero eigenvalue (Eq. 2.14) is significant, since this breaks the Galilean invariance of the ideal MHD system (because the zero eigenvalue would stay zero in all frames of reference).

2.1.2 Resistive MHD Equations

The resistive MHD system consists of a similar set of equations as the ideal MHD system, which are, in dimensionless form, given by the following (refer to Appendix A for more details on the normalization procedure)

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0, \tag{2.15}$$

$$\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot \left(\rho \vec{v}\vec{v} + \vec{I}(p + \frac{\vec{B} \cdot \vec{B}}{2}) - \vec{B}\vec{B} \right) = 0, \tag{2.16}$$

$$\frac{\partial \vec{B}}{\partial t} + \nabla \cdot (\vec{v}\vec{B} - \vec{B}\vec{v}) + \nabla \times (\eta \vec{J}) = 0, \tag{2.17}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot \left((e + p + \frac{\vec{B} \cdot \vec{B}}{2})\vec{v} - (\vec{v} \cdot \vec{B})\vec{B} + \eta \vec{J} \times \vec{B} \right) = 0. \tag{2.18}$$

The solenoidality constraint of the Maxwell's equations, $\nabla \cdot \vec{B} = 0$, also applies. Applying Ampere's law, and ignoring displacement current, we can express the current density, $\vec{J}$, as the curl of the magnetic field

$$\vec{J} = \nabla \times \vec{B}. \tag{2.19}$$

Unlike ideal MHD, Eq. 2.17, and the energy equation, Eq. 2.18, contain terms involving derivatives of $\vec{J}$, where $\vec{J}$ itself is already a derivative of the flow variable, $\vec{B}$.

2.1.2.1 Second-Order Flux Terms of the Induction Equation

We have noted that the induction equation of resistive MHD contains an elliptic flux term involving the derivative of the current density (an elliptic term is a term with second-order derivatives). In this work, resistivity, η, is assumed to be constant, so the resistivity can be safely taken out from the curl term

$$\nabla \times (\eta \vec{J}) = \eta \nabla \times J, \tag{2.20}$$

where, as previously mentioned, the current density, $\vec{J}$, can be expressed as the curl of the magnetic field, $\vec{B}$

$$\vec{J} = \nabla \times \vec{B} = \begin{bmatrix} \frac{\partial Bz}{\partial y} - \frac{\partial By}{\partial z} \\ \frac{\partial Bx}{\partial z} - \frac{\partial Bz}{\partial x} \\ \frac{\partial By}{\partial x} - \frac{\partial Bx}{\partial y} \end{bmatrix}. \tag{2.21}$$

Rewriting the $\nabla \times \vec{J}$ term in divergence form, we define a tensor, $\nabla \cdot \vec{\vec{T}}$, where

$$\nabla \times \vec{J} = \nabla \cdot \vec{\vec{T}} = \vec{\nabla} \cdot \begin{bmatrix} 0 & \frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y} & \frac{\partial B_z}{\partial x} - \frac{\partial B_x}{\partial z} \\ \frac{\partial B_x}{\partial y} - \frac{\partial B_y}{\partial x} & 0 & \frac{\partial B_z}{\partial y} - \frac{\partial B_y}{\partial z} \\ \frac{\partial B_x}{\partial z} - \frac{\partial B_z}{\partial x} & \frac{\partial B_y}{\partial z} - \frac{\partial B_z}{\partial y} & 0 \end{bmatrix}. \tag{2.22}$$

Here, $\nabla \cdot \vec{\vec{T}}$ can be interpreted as a divergence of a flux that involves the gradient of the magnetic field, similar in concept with the notion of elliptic flux, $\vec{\mathbf{F}}_E(\vec{\mathbf{U}}, \nabla\vec{\mathbf{U}})$, in Sec. 3.5 in [27], where a flux term depends not only on the variables, $\vec{\mathbf{U}}$, but also on their gradients, $\nabla\vec{\mathbf{U}}$.

Assuming constant resistivity, the induction equation, Eq. 2.17, can then be rewritten in the following divergence form

$$\frac{\partial \vec{B}}{\partial t} + \nabla \cdot (\vec{v}\vec{B} - \vec{B}\vec{v} + \eta\vec{\vec{T}}) = 0, \tag{2.23}$$

which fits nicely within the finite-volume framework, taking into account the fact that the tensor, $\vec{\vec{T}}$, depends on the derivatives of $\vec{B}$ as well (see Eq. 2.22).

2.1.2.2 Second-Order Flux Terms of the Energy Equation

Aside from the induction equation, the energy equation, Eq. 2.18, depends also on $\vec{J}$, which is a derivative of $\vec{B}$. Expanding the $\nabla \cdot (\vec{J} \times \vec{B})$ term of the energy equation, we get

$$\begin{aligned} \nabla \cdot (\vec{J} \times \vec{B}) &= \nabla \cdot \begin{bmatrix} (B_z J_y - B_y J_z) & (B_x J_z - B_z J_x) & (B_y J_x - B_x J_y) \end{bmatrix} \\ &= \frac{\partial}{\partial x}(B_z J_y - B_y J_z) + \frac{\partial}{\partial y}(B_x J_z - B_z J_x) + \frac{\partial}{\partial z}(B_y J_x - B_x J_y). \end{aligned} \tag{2.24}$$

Each component of $\vec{J}$ can be expressed in terms of the derivatives of the magnetic field components (see Eq. 2.21).

2.2 Control Schemes of the $\nabla \cdot \vec{\mathbf{B}}$ constraint

The ideal MHD system consists of time-dependent equations governing the evolution of density, momentum, magnetic field and energy. As mentioned in Sec. 2.1.1, these equations

are supplemented by the divergence-free condition, which acts on the magnetic field, Eq. 2.5. At the PDE level, this condition is automatically satisfied, provided that the magnetic field is also initially divergence-free [28, 29, 30]. Starting from the curl form of the induction equation

$$\frac{\partial \vec{B}}{\partial t} + \nabla \times \vec{E} = 0, \tag{2.25}$$

we can take the divergence of both sides

$$\frac{\partial \nabla \cdot \vec{B}}{\partial t} + \nabla \cdot (\nabla \times \vec{E}) = 0, \tag{2.26}$$

to obtain

$$\frac{\partial}{\partial t}(\nabla \cdot \vec{B}) = 0, \tag{2.27}$$

which implies that $\nabla \cdot \vec{B}$ is constant in time, so the solenoidality condition acts as an initial condition for $\nabla \cdot \vec{B}$.

It is, however, imperative to note that, while analytically the divergence of the magnetic field remains constant in time (by the virtue of the vector identity that the divergence of a curl is identically zero), numerically, how accurately the solenoidality condition is satisfied depends on how accurately the div-curl operator, $\nabla \cdot (\nabla \times)$, numerically approximates zero. While one can expect that the divergence error would reduce with grid size, large divergence errors would give rise to a Lorentz force, $\vec{J} \times \vec{B}$, with a significant component that is parallel to the magnetic field. Isolating the component of Lorentz force that is parallel to the magnetic field illustrates this

$$(\vec{J} \times \vec{B}) \cdot \vec{B} = (\vec{B} \cdot \vec{B}) \nabla \cdot \vec{B}. \tag{2.28}$$

It can be seen from Eq. 2.28 that the parallel force is analytically zero due to the solenoidality condition, $\nabla \cdot \vec{B} = 0$. However, numerically, the divergence error is not exactly zero, and the spurious parallel force may lead to numerical instabilities and incorrect solutions [26]. Thus, using the conservative form of the momentum equation (see Eq. 2.2) may not be appropriate when there is no guarantee that $\nabla \cdot \vec{B}$ is numerically zero [26].

There are several mechanisms that can be considered to control divergence errors, most of which fall under either of the two larger schools of thoughts for divergence correction mechanisms [29]:

1. To ensure that the solenoidality constraint is maintained down to machine accuracy in some discretization (projection schemes [26], constrained transport methods [31, 32, 33, 34, 35] and central differencing [29]),

2. To maintain the solenoidality condition to the accuracy of the truncation error (Powell's 8-wave MHD formulation [30] and GLM-MHD [28]).

Only schemes that fall under the second category will be discussed in this Section. Schemes under the first category will not be considered in this work for the following reasons

1. Schemes such as constrained transport methods rely on staggered placements of the different components of the magnetic field, which could be challenging from an implementation perspective when non-uniform non-Cartesian meshes are considered,
2. Constrained transport method preserve the divergence constraint only for a specific discretization,
3. The projection scheme relies on the efficiency and accuracy of the Poisson's solver, due to the elliptic system to be solved at the end of every time-step. For problems involving non-periodic boundary conditions, the extra step associated with solving the elliptic problems can be computationally costly.

Therefore, only Powell's 8-wave MHD formulation [30] and the GLM-MHD system [28] are described in subsequent sections (Sec. 2.2.1 and Sec. 2.2.2, respectively), and are utilized for simulations within this treatise.

2.2.1 Powell's 8-Wave MHD Formulation

In [30], the 8-wave MHD formulation was proposed to deal with the divergence constraint of the magnetic field. Following the symmetric form of ideal MHD proposed by Godunov [36], the ideal MHD equations in 'physical form' are rewritten in divergence form without making any assumption on the solenoidality of the magnetic field, as given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0, \tag{2.29}$$

$$\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot \left(\rho \vec{v}\vec{v} + \vec{I}(p + \frac{\vec{B} \cdot \vec{B}}{2}) - \vec{B}\vec{B} \right) = -\vec{B} \nabla \cdot \vec{B}, \tag{2.30}$$

$$\frac{\partial \vec{B}}{\partial t} + \nabla \cdot (\vec{v}\vec{B} - \vec{B}\vec{v}) = -\vec{v} \nabla \cdot \vec{B}, \tag{2.31}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot \left((e + p + \frac{\vec{B} \cdot \vec{B}}{2})\vec{v} - (\vec{v} \cdot \vec{B})\vec{B} \right) = -(\vec{v} \cdot \vec{B}) \nabla \cdot \vec{B}. \tag{2.32}$$

Note that Eq. 2.29 to Eq. 2.32 are exactly equivalent to the ideal MHD equations (Eq. 2.1 to Eq. 2.4) except for the source term that is proportional to $\nabla \cdot \vec{B}$. From Eq. 2.29 to Eq. 2.32, we can then derive the evolution equation for $\nabla \cdot \vec{B}$

$$\frac{\partial}{\partial t}(\nabla \cdot \vec{B}) + \nabla \cdot (\vec{u} \nabla \cdot \vec{B}) = 0, \tag{2.33}$$

which implies that $\nabla \cdot \vec{B}$ is constant in the direction of the flow. This also suggests that divergence error would get advected out of the domain by the velocity $\vec{u}$, providing a correction mechanism for non-zero divergence.

By rewriting the ideal MHD system without any assumption of solenoidality, we have arrived at a system of equations that is symmetrizable, and has an in-built correction mechanism to deal with non-zero divergence of the magnetic field [30]. The presence of the source term also leads to a Galilean invariant system, with the following replacing the zero eighth eigenvalue of the ideal MHD system (Eq. 2.14)

$$\lambda_8 = v_x. \tag{2.34}$$

Conservation, however, is no longer preserved, due to the presence of the source term. Tóth [29] briefly discussed the consequence of the loss of conservation in that, for some of his test cases, jump conditions are not calculated correctly, resulting in erroneous solution containing error that is not convergent to zero with grid resolution, even away from discontinuities. In spite of this deficiency, Powell method continues to be the divergence correction technique of choice, and has in general worked rather well to keep the divergence error in check.

2.2.2 GLM Control of the $\nabla \cdot \vec{B}$ constraint

The GLM-MHD formulation can be described as follows. Following a similar approach as for the Maxwell equations [37], the divergence constraint (Eq. 2.5) can be coupled with the induction equation through the introduction of a new potential variable, ψ [28]. The equations describing the evolution of the magnetic field, Eq. 2.3 and Eq. 2.5, are then replaced with the following equations

$$\frac{\partial \vec{B}}{\partial t} + \nabla \cdot (\vec{v}\vec{B} - \vec{B}\vec{v}) + \nabla\psi = 0, \tag{2.35}$$

$$\frac{\partial \psi}{\partial t} + c_h^2 \nabla \cdot \vec{B} = -\frac{c_h^2}{c_p^2}\psi. \tag{2.36}$$

As can be seen from these equations, the system is still conservative except for the evolution equation of ψ, which is not a physical variable. This preservation of conservation for physical variables is the main advantage of the GLM method over the Powell method that was proposed earlier to approximately satisfy the divergence constraint [30]. Replacing the zero eigenvalue (Eq. 2.14), two new eigenvalues arise in the GLM-MHD formulation, which are $\pm c_h$. The coefficients c_p and c_h control the amount of diffusion in ψ and the advection speed, respectively. The 'purely hyperbolic' correction can be obtained by taking

c_p to infinity (no diffusion). Following [28], we choose these two coefficients to be related in our numerical simulations through the following expression

$$c_r = \frac{c_p^2}{c_h}, \tag{2.37}$$

with c_r chosen to be 0.18. The coefficient c_h, then, determines how fast the divergence of the magnetic field is advected out of the domain, whereas c_p controls its dissipation. Therefore, to ensure that the error is advected as fast as possible, it is desirable to set c_h as high as possible. However, because the two new eigenvalues have magnitude c_h, it is also important to set c_h small enough so that it will not affect the time-step criterion of the simulation. Thus, c_h is often chosen to be the largest of all MHD eigenvalues in the whole domain over all cell interfaces (i, j) in 2D, which can be written as

$$c_h = \max_{i,j} \left(|v_n| + c_{f_n} \right), \tag{2.38}$$

where v_n and c_{f_n} are the plasma velocity and the fast magnetosonic wave speed (Eq. 2.7) in the direction normal to the interfaces (i, j). In [28] it is recommended to choose c_p by setting the parameter $c_r = c_p^2/c_h$ to a constant value of 0.18. In [38, 39], it is instead suggested to choose c_p by setting the parameter $\bar{\alpha} = \Delta x \, c_h/c_p^2$ to a constant value in $[0, 1]$, where Δx is a measure of the grid spacing. Note that $\bar{\alpha}$ is a dimensionless quantity reflecting the ratio of the diffusive and advective time scales.

2.2.2.1 Boundary Condition Treatment of ψ at Inflow and Outflow Boundaries

The choice of c_h as given by Eq. 2.38 ensures that no eigenvalue will exceed the largest physical eigenvalue in the domain, while at the same time, it guarantees that the divergence error will be advected out of the simulation domain with the fastest physical wave speed in the flow solution. Since the two additional eigenvalues are $\pm c_h$ regardless of the actual plasma velocities and wave speeds, eigenvalues of both signs will always exist at all cell interfaces. This means that treatment similar to subsonic inlet and outlet boundary conditions (see [40]) is always required for inflow and outflow boundary conditions. Since the waves with eigenvalues $\pm c_h$ only carry changes in the normal magnetic field and ψ [28], only these two variables need to be taken into account at boundaries to accommodate these waves. For example, consider superfast inflow boundary conditions and assume without loss of generality that $v_n > 0$. Since the inflow velocity is faster than the fast magnetosonic wave, all the MHD eigenvalues are positive (information travels into the computational domain). However, for GLM-MHD one cannot just prescribe all variables, because one of the eigenvalues, $-c_h$, is necessarily negative, even when the flow is superfast at the inflow boundary. One of either ψ or the normal magnetic field has to be extrapolated from the interior solution, and because the inflow magnetic field is prescribed at the boundary, it is ψ that has to be extrapolated from the interior. The same logic applies to superfast

outflow. Assume again that $v_n > 0$. Without GLM, all of the variables would just be extrapolated from the inside of the domain, since all eigenvalues are positive, hence no information is propagating into the domain. However, due to the negative eigenvalue $-c_h$, ψ needs to be prescribed at the outflow boundary. A suitable choice for ψ is to set it to zero at superfast outflow boundaries. (This is consistent with Yalim et al. [41], who set ψ to a constant at the superfast outlet boundaries.)

Chapter 3

High-Order Finite-Volume Scheme for 2D Ideal MHD

In this chapter, a high-order accurate finite-volume scheme is proposed for the 2D ideal magnetohydrodynamics equations. Many recent developments employ discontinuous Galerkin (DG) finite element methods [42, 43, 44, 45], and others are based on essentially non-oscillatory (ENO) FV schemes and on weighted ENO (WENO) FV schemes [38, 39, 46, 47, 48, 49, 50]. Most of these high-order approaches were only described and implemented for regular Cartesian grids. Our high-order MHD scheme uses a different approach. It is based on Barth's k-exact reconstruction procedure [51], which uses a least-squares approach on overdetermined stencils to compute polynomial reconstruction coefficients, in a multi-dimensional way that can handle general polygonal grids. In order to control spurious oscillations at shocks, we use the CENO monotonicity procedure that was introduced by Ivan and Groth [4] for the Euler equations, and has since been extended to the Navier-Stokes equations [3, 52]. Our implementation of this CENO monotonicity procedure switches between an unlimited piecewise cubic reconstruction (fourth-order accurate) and a limited piecewise-linear reconstruction (second-order accurate), with the switching based on the smoothness indicator introduced in [4]. Note that the scheme we describe can in principle be implemented with arbitrary order, but fourth-order accuracy is a suitable practical choice for the numerical results to be presented in this paper. The smoothness indicator is computed in each cell to determine whether the flow is locally smooth and well-resolved. For cells containing non-smooth or under-resolved solution content, the unlimited k-exact reconstruction is switched to limited piecewise linear reconstruction. The smoothness indicator can also be used directly to formulate a criterion for AMR. The CENO scheme is called central because both the high-order and the low-order stencils are central with respect to the cell. The method is an ENO method, thus, not necessarily monotonicity preserving in the strict sense (such as TVD or TVB [52]), but rather, it is

monotone in the ENO sense [53]:

$$TV(u^{n+1}) = TV(u^n) + \mathcal{O}(\Delta x^{k+1}), \tag{3.1}$$

where u^n denotes a solution variable u at time level n, Δx is the grid spacing, k is the order of polynomial reconstruction, and TV stands for total variation. The ENO property allows the presence of small spurious oscillations that have a magnitude on the order of the truncation error, but it does not allow $O(1)$ Gibbs-like oscillations at discontinuities [53]. It is important to note that the CENO method proposed by Ivan and Groth [4] does not choose between asymmetric stencils as most other methods do that try to enforce the ENO property, but instead uses a hybrid approach that chooses between high-order and limited low-order central reconstructions. Note that Harten and Chakravarthy [54] also proposed a technique on Cartesian grids to obtain an ENO reconstruction using central stencils by hybridizing a high-order reconstruction with a first-order formulation, and this served as an inspiration for the CENO approach of [4, 52]. The fixed stencil used during the CENO reconstruction procedure avoids the complexity of considering multiple non-central stencil configurations that characterizes traditional ENO schemes. Note also that the CENO method of Ivan and Groth is not a central ENO method in the sense of Nessyahu and Tadmor's staggered mesh philosophy [55], but it uses non-staggered central stencils of different order. We note that our limited low-order least-squares scheme for MHD with GLM divergence cleaning is similar to the second-order discretization proposed by Yalim et al. in [41] (implemented on unstructured grids), and our high-order method is a high-order extension of this approach that combines least-squares reconstruction with GLM. Our CENO-GLM high-order MHD scheme thus provides an alternative to high-order DG and ENO/WENO methods for MHD, and is attractive because it can naturally be applied on general grids.

The organization of this chapter is as follows. The high-order CENO scheme is described in Sec. 3.1 for two-dimensional space, combined with GLM-MHD to control divergence error of the magnetic field, which is described in Sec. 2.2.2. Numerical results will be presented in Sec. 3.2 to illustrate the accuracy of the scheme in handling smooth problems (Sec. 3.2.1) and also problems with discontinuities (Sec. 3.2.2). Lastly, the dynamic mesh adaptation capabilities of the approach (described in detail in [52]) are demonstrated using adaptive time-dependent simulations of the Orszag-Tang vortex problem [56] with high-order accuracy and unprecedented effective resolution (Sec. 3.2.3).

3.1 High-Order CENO Scheme for Ideal MHD in 2D

In this section we give a detailed description of the proposed high-order CENO scheme for MHD, which is obtained by combining Ivan and Groth's CENO approach for compressible gas dynamics [4] with GLM divergence cleaning. We first describe the high-order

FV framework, followed by discussions on Barth's k-exact reconstruction and the CENO reconstruction selection process using the CENO smoothness indicator. While the CENO method applies to general mesh topologies, the discussion here is restricted to the application to 2D, multi-block, body-fitted (logically Cartesian) AMR grids having quadrilateral computational elements of the type considered by Groth and co-researchers [57, 58, 59, 60]. Numerical flux calculation and source term integration for our high-order MHD CENO method are described next, followed by a discussion on how our MHD CENO implementation obtains high-order accuracy at curved boundaries.

3.1.1 High-Order Finite-Volume Formulation

Consider hyperbolic conservation law

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \vec{\mathbf{F}} = \mathbf{S}, \tag{3.2}$$

where $\mathbf{U}$ is the vector of conserved variables, $\vec{\mathbf{F}}$ consists of the flux terms of the system, and $\mathbf{S}$ is a source term vector. For a quadrilateral cell (i, j), the semi-discrete FV form of Eq. 3.2 is given as

$$\frac{d\overline{\mathbf{U}}_{i,j}}{dt} = -\frac{1}{A_{i,j}} \sum_{l=1}^{4} \sum_{m=1}^{N_g} (\omega \, \vec{\mathbf{F}}_{num} \cdot \vec{n} \Delta l)_{i,j,l,m} + \overline{\mathbf{S}}_{i,j}, \tag{3.3}$$

where $\overline{\mathbf{U}}_{i,j}$ is the numerical approximation of the average value of $\mathbf{U}$ in cell (i, j), $\vec{\mathbf{F}}_{num}$ is the numerical flux function, $A_{i,j}$ is the area of the computational cell (i, j), N_g is the number of Gauss quadrature points on each cell face and ω is the associated Gauss quadrature weight to each of the Gauss points. The actual number of flux quadrature points, N_g, depends on the order of solution reconstruction, with two Gauss quadrature points per face for third- and fourth-order accurate schemes, but only one Gauss quadrature point per face for second-order or lower [4]. The order of the polynomial reconstruction then determines the spatial accuracy of the solution by providing more accurate approximations of the solution values at the Gauss quadrature points for flux calculation. In general, an order-k polynomial reconstruction provides an order-$(k + 1)$ accurate spatial discretization for smooth problems. We use standard explicit second-order and fourth-order Runge-Kutta methods [40] to integrate Eq. 3.3 in time for the second-order and fourth-order accurate spatial discretizations to be compared in our time-dependent numerical test problems. For steady-state simulations, we use a five-stage optimally smoothing method regardless of the solution accuracy [61].

3.1.2 k-Exact Piecewise Polynomial Reconstruction

Following Barth [51], the variation of a solution variable, u, at any location within the quadrilateral computational cell (i,j), assumes the form

$$u_{i,j}^{k}(\vec{X}) = \sum_{\substack{p_1=0 \\ (p_1+p_2\leq k)}}^{k} \sum_{p_2=0}^{k} \left(x-\overline{x}_{i,j}\right)^{p_1} \left(y-\overline{y}_{i,j}\right)^{p_2} D_{p_1p_2}^{k}, \tag{3.4}$$

where k is the order of the polynomial function, $\vec{X} = (x,y)$ are the coordinates at which the solution is sought, $(\overline{x}_{i,j}, \overline{y}_{i,j})$ are the coordinates of the centroid of cell (i,j), and $D_{p_1p_2}^{k}$ are high-order polynomial coefficients that will need to be determined for each of the primitive variables for every cell, based on a set of cell averages, $\overline{u}_{\gamma,\delta}$, in the neighbourhood of cell (i,j). For the test cases presented in this chapter, linear ($k=1$) and cubic ($k=3$) reconstructions are chosen to obtain second- and fourth-order accurate schemes. The monotonicity-preserving procedure, which is discussed in Sect. 3.1.3, reduces k to 1 and applies limiters in regions of the flow that are deemed under-resolved or to contain discontinuities.

The coefficients $D_{p_1p_2}^{k}$ are determined by solving an overdetermined system of linear equations in a least-squares sense, fitting the reconstruction polynomial to the solution averages for cell (i,j) and for its neighbouring cells in the reconstruction stencil of cell (i,j). For a polynomial of degree k, the number of coefficients $D_{p_1p_2}^{k}$ is given by $\mathcal{N}_D = \frac{(k+1)(k+2)}{2}$ [4, 27, 52]. Thus, there are 3 coefficients to be determined for $k=1$ or linear reconstruction and 10 coefficients for $k=3$ or cubic reconstruction. Following the requirements imposed by Barth [51], it is important that these coefficients are determined in such a way that the following conditions are satisfied:

- **Conservation of the mean.** The average of the reconstructed polynomial function over cell (i,j) should recover exactly the cell-averaged value $\overline{u}_{i,j}$:

$$\overline{u}_{i,j} = \frac{1}{A_{i,j}} \iint_{\mathcal{A}_{i,j}} u_{i,j}^{k}(\vec{X})\, dA. \tag{3.5}$$

- **k-exactness.** The reconstructed polynomial function should be able to reconstruct polynomials of degree up to k exactly [51]:

$$u_{i,j}^{k}(\vec{X}) - u_{\text{exact}}(\vec{X}) = \mathcal{O}(\Delta x^{k+1}). \tag{3.6}$$

- **Compact support.** The reconstructed polynomial function should depend only on average values within a relatively small neighbourhood [51]. Only the cell-averaged data within the supporting stencil is used for reconstruction purposes.

In theory, $\mathcal{N}_D$ determines the minimum size of the supporting stencil, but in practice more neighbours are included to make the reconstruction more robust for complicated and stretched meshes [4, 27, 52]. On our 2D body-fitted structured grid blocks, first-degree neighbours are included for $k = 0$ and $k = 1$ reconstruction stencils (a total of 8 neighbours), and first- and second-degree neighbours are included for $k = 2$ and $k = 3$ reconstruction stencils (a total of 24 neighbours).

Consider reconstruction for cell (i, j). In the reconstruction step an overdetermined system $\mathbb{A}\mathbf{D} - \mathbf{B} = 0$ is solved in the least-squares sense, together with the constraint of Eq. 4.10, which is imposed exactly. Here, $\mathbf{D}$ is the array of polynomial coefficients, $D^k_{p_1p_2}$, and the equations $\mathbb{A}\mathbf{D} - \mathbf{B} = 0$ are given by

$$(\mathbb{A}\mathbf{D} - \mathbf{B})_{\gamma,\delta} = \left(\frac{1}{A_{\gamma,\delta}} \iint_{\mathcal{A}_{\gamma,\delta}} u^k_{i,j}(\vec{X})\, dA \right) - \overline{u}_{\gamma,\delta} = 0. \tag{3.7}$$

There is one equation for each cell (γ, δ) in the stencil of cell (i, j). Each equation matches the actual cell average $\overline{u}_{\gamma,\delta}$ in cell (γ, δ) with the average over cell (γ, δ) of the reconstructed polynomial $u^k_{i,j}(\vec{X})$ for cell (i, j). Equation 4.10 is enforced analytically by replacing $u^k_{i,j}$ with Eq. 3.4 and expressing the first coefficient, D^k_{00}, as a function of the other $M = \mathcal{N}_D - 1$ polynomial unknowns as

$$D^k_{00} = \overline{u}_{i,j} - \sum_{\substack{p_1=0 \\ (p_1+p_2\neq 0)}}^{k} \sum_{p_2=0}^{k} D^k_{p_1p_2} \left(\overline{x^{p_1} y^{p_2}}\right)_{i,j} , \tag{3.8}$$

where the geometric moment $\left(\overline{x^{p_1} y^{p_2}}\right)_{i,j}$ of powers (p_1, p_2) is given by

$$\left(\overline{x^{p_1} y^{p_2}}\right)_{i,j} = \frac{1}{A_{i,j}} \iint_{\mathcal{A}_{i,j}} (x - \bar{x}_{i,j})^{p_1} (y - \bar{y}_{i,j})^{p_2}\ dA\,. \tag{3.9}$$

Substituting $u^k_{i,j}$ from Eq. 3.4 in Eq. 4.11 and using Eq. 3.8 for D^k_{00} the following overdetermined linear system for the M unknowns is obtained

$$\begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_J \\ \vdots \\ L_{N_n} \end{bmatrix}_{N_n \times M} \begin{pmatrix} D^k_{01} \\ D^k_{02} \\ \vdots \\ D_{p_1p_2} \\ \vdots \\ D^k_{k0} \end{pmatrix}_{M\times 1} - \begin{pmatrix} w_1(\overline{u}_1 - \overline{u}_I) \\ w_2(\overline{u}_2 - \overline{u}_I) \\ \vdots \\ w_J(\overline{u}_J - \overline{u}_I) \\ \vdots \\ w_{N_n}(\overline{u}_{N_n} - \overline{u}_I) \end{pmatrix}_{N_n\times 1} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{N_n\times 1} \tag{3.10}$$

where a unique index $J = (\gamma, \delta)$ has been assigned to each of the N_n neighbours in the supporting reconstruction stencil and the index $I = (i, j)$ denotes the cell having the

solution reconstructed. The generic row L_J of the matrix $\mathbb{A}$ for a neighbouring cell J is given by

$$L_J = \left(w_J \left(\widehat{x^0 y^1} \right)_{IJ} \quad w_J \left(\widehat{x^0 y^2} \right)_{IJ} \quad \dots \quad w_J \left(\widehat{x^{p_1} y^{p_2}} \right)_{IJ} \quad \dots \quad w_J \left(\widehat{x^k y^0} \right)_{IJ} \right) \ , \tag{3.11}$$

in which w_J is a geometric weight specific to each neighbour J which serves the purpose of improving the locality of the reconstruction, becoming especially important for stretched meshes with boundary curvature [62]. (In essence, equations corresponding to close-by neighbour cells in the reconstruction stencil get larger weights in the least-squares solution than neighbour cells that are further away.) The matrix coefficients $\left(\widehat{x^{p_1} y^{p_2}} \right)_{IJ}$ for the pair of I and J cells have the expression

$$\left(\widehat{x^{p_1} y^{p_2}} \right)_{IJ} = \left(\frac{1}{A_{\gamma,\delta}} \iint_{\mathcal{A}_{\gamma,\delta}} (x - \bar{x}_{i,j})^{p_1} \, (y - \bar{y}_{i,j})^{p_2} \ dA \right) - (\overline{x^{p_1} y^{p_2}})_{i,j} \ , \tag{3.12}$$

where the quantities $\left(\widehat{x^{p_1} y^{p_2}} \right)_{IJ}$ depend only on the geometry of I and J cells, and involve a monomial integration that can be computed by applying quadrature rules. An efficient way to calculate the geometric moments $\left(\widehat{x^{p_1} y^{p_2}} \right)_{IJ}$ using only the $(\overline{x^{p_1} y^{p_2}})$ moments is described in [52].

QR factorization or multiplication with the pseudo-inverse of $\mathbb{A}$ can be used to determine the solution of Eq. 4.14, as described in more detail in [4, 27, 52]. The complete solution of the constrained least-squares problems is then obtained by calculating D_{00} using Eq. 3.8. In each time step, the constrained least-squares reconstruction problem is solved for each cell and for each primitive variable. Matrix $\mathbb{A}$ depends completely on the geometry and is the same for all least-squares problems in a given cell (i, j) (i.e., for each solution variable) and for all time steps, so it can be precomputed and stored for computational efficiency (see [4, 27, 52] for details). As explained in Sect. 3.1.5, one-sided stencils and additional constraints on the least-squares solution are used to handle boundary conditions with high-order accuracy at curved boundaries.

3.1.3 CENO Smoothness Indicator to Enforce Monotonicity

The CENO method controls monotonicity throughout the computational domain by selecting a limited linear reconstruction in cells where the flow is deemed to be non-smooth or under-resolved, and a high-order k-exact reconstruction elsewhere. The limited linear reconstruction is based on k-exact reconstruction with $k = 1$ combined with the standard Venkatakrishnan limiter, see [52, 63]. To estimate whether the flow in cell (i, j) is under-resolved or non-smooth, a variable $\mathcal{S}$, the smoothness indicator, is computed [4]:

$$\mathcal{S} = \frac{\alpha \, c_s}{\max(1 - \alpha, \epsilon)}, \tag{3.13}$$

where α is given by

$$\alpha = 1 - \frac{\sum_{\gamma}\sum_{\delta}\left(u^k_{\gamma,\delta}(\vec{X}_{\gamma,\delta}) - u^k_{i,j}(\vec{X}_{\gamma,\delta})\right)^2}{\sum_{\gamma}\sum_{\delta}\left(u^k_{\gamma,\delta}(\vec{X}_{\gamma,\delta}) - \overline{u}_{i,j}\right)^2} \tag{3.14}$$

and $c_s = (\mathcal{N}_{SOS} - \mathcal{N}_D)/(\mathcal{N}_D - 1)$ is a positive constant. Here, $\mathcal{N}_{SOS}$ stands for 'size of stencil' used for reconstruction, $\mathcal{N}_D$ stands for 'degrees of freedom' and denotes the number of unknown polynomial coefficients, and ϵ is introduced to avoid division by zero (we use $\epsilon = 10^{-8}$). Further, γ and δ denote the indices of the neighbouring cells to the cell (i, j) that are part of its reconstruction stencil, and $\vec{X}_{\gamma,\delta}$ is the centroid of cell (γ, δ). (Note that the stencil used for computing the smoothness indicator can also be chosen smaller than the reconstruction stencil. In our numerical results, we compute the smoothness indicator associated with each primitive solution variable in cell (i, j) using a stencil with nine cells, i.e., the cell (i, j) and its eight first-degree neighbours.) The parameter α basically measures how accurately centroidal solution values of neighbouring cells can be reproduced using the reconstruction for cell (i, j). The range of α is $-\infty < \alpha \leq 1$: for smooth variation, the second term of the right-hand side of Eq. 3.14 tends to be close to zero and α is very close to one; for cells close to a discontinuity or with an under-resolved feature, the magnitude of α tends away from one and it can also become negative. The range of the smoothness indicator $\mathcal{S}$ is $-c_s < \mathcal{S} < c_s/\epsilon$: for smooth variation ($\alpha$ very close to one), $\mathcal{S}$ is large; for non-smooth or under-resolved features (α away from one), $\mathcal{S}$ is small. The smoothness indicator $\mathcal{S}$ is then compared with a cutoff value $\mathcal{S}_\mathcal{C}$: when $\mathcal{S} > \mathcal{S}_\mathcal{C}$ the solution is deemed locally smooth and the high-order reconstruction is used, and for $\mathcal{S} \leq \mathcal{S}_\mathcal{C}$ the solution is locally non-smooth or under-resolved, and the limited low-order reconstruction is used. We also use $\mathcal{S}$ in our adaptive procedure to refine regions where the solution is non-smooth or under-resolved. A potential disadvantage of this approach is that it is not fully parameter-free. However, we have found it easy to pick suitable values of $\mathcal{S}_\mathcal{C}$ based on the range recommended in [52] for the numerical tests shown in Sect. 3.2. The selection of an appropriate cutoff value is also made easier by the use of the transition function $\alpha/(1 - \alpha)$, which rapidly magnifies small variations in α very close to one. Additionally, it is worth emphasizing that a single value $\mathcal{S}_\mathcal{C}$ is selected and applied to all solution variables and all mesh resolutions used for solving a particular problem. Note also that the use of the adjustment coefficient, c_s, in the expression of $\mathcal{S}$ helps making the selection of $\mathcal{S}_\mathcal{C}$ relatively independent of the order of the scheme and making the smoothness indicators comparable for different stencil sizes that may occur at domain boundaries. Robustness of the scheme then depends on carefully choosing a user-defined parameter, namely, the smoothness indicator.

The form of the smoothness indicator is inspired by the definition of multiple-correlation coefficients and least-squares goodness-of-fit testing; see [52] for a more detailed discussion

with further motivation for the approach. As is shown by extensive testing for the Euler and Navier-Stokes equations in [4, 27, 52, 64] and is further confirmed by the numerical MHD tests presented below, the CENO approach with smoothness indicator $\mathcal{S}$ is a useful scheme in terms of providing high-order accurate numerical approximations while avoiding spurious oscillations at discontinuities.

3.1.4 Numerical Flux Function and Source Term Integration

In this subsection we discuss numerical flux computation and treatment of the GLM source term (in Eq. 2.36) for the high-order MHD CENO scheme.

We use the Lax-Friedrichs numerical flux function for the implementation of the proposed high-order MHD CENO scheme. As a function of the left and the right reconstructed states ($\mathbf{U}_L$ and $\mathbf{U}_R$ respectively) at each of the Gauss quadrature points, the Lax-Friedrichs flux function is given as

$$\mathcal{F}(\mathbf{U}_L, \mathbf{U}_R) = \frac{\vec{\mathbf{F}}(\mathbf{U}_R) + \vec{\mathbf{F}}(\mathbf{U}_L)}{2} - \lambda_{max} \frac{\mathbf{U}_R - \mathbf{U}_L}{2}, \tag{3.15}$$

where λ_{max} is the maximum of the absolute values of the normal eigenvalues of the states in the two cells. Following Dedner et al. [28], the equations for B_x and ψ are decoupled from the rest of the system, so the Lax-Friedrichs numerical fluxes are applied only to the other seven variables. The fluxes at the interfaces for B_x and ψ are calculated by setting these variables to the following values at the cell interfaces [28]:

$$B_{x,m} = \frac{1}{2}(B_{x,r} + B_{x,l}) - \frac{1}{2\,c_h}(\psi_r - \psi_l), \tag{3.16}$$

$$\psi_m = \frac{1}{2}(\psi_r + \psi_l) - \frac{c_h}{2}(B_{x,r} - B_{x,l}), \tag{3.17}$$

where the subscripts l and r denote the left and right reconstructed states at cell interfaces and c_h is the global maximum of $|v_x| + c_{f_x}$ at cell interfaces. These values are substituted directly into the exact flux formulas for the B_x and ψ equations. In a multi-dimensional setting, B_x is effectively B_n, which is the magnetic field component normal to the interface. These ψ_m and $B_{n,m}$ values are also used for flux calculation of the other seven variables, which uses the Lax-Friedrichs numerical flux with local values of $|v_x| + c_{f_x}$ as the largest wave speed that determines the size of the numerical dissipation.

As an alternative, one can also apply the standard Lax-Friedrichs flux (Eq. 3.15) directly to the full system with nine variables, without decoupling the 2×2 system. One can expect this to be more diffusive since in this case c_h (the global maximum of $|v_x| + c_{f_x}$) determines the numerical diffusion, but we have not found much difference with the decoupled approach when trying this for our numerical tests. Nevertheless, in the numerical results presented

below we use the decoupled approach. Other flux functions such as Roe [65] and HLLE [66] can also be considered. Wheatley et al. [67] compared flux functions for high-order DG methods, and found that using more accurate Riemann solvers improves results in some cases (e.g., at shocks), but does often not make much difference in smooth regions of the flow. This is also expected for CENO since the intercellular solution jumps diminish in size as the reconstruction order increases, but a detailed investigation of this for the CENO MHD scheme is beyond the scope of this work.

The ideal (non-modified) MHD system (Eq. 2.1 - Eq. 2.4) is a hyperbolic system of equations, so it easily fits within the CENO framework. The GLM-MHD formulation adds a source term to the ψ-update equation (Eq. 2.36), which can be treated in two different ways in our implementation.

The first option, as proposed in Dedner et al. [28], is to incorporate the source term using an operator splitting approach, where Eq. 2.36 is split into two parts:

- a homogenous conservation law, corresponding to the hyperbolic part of Eq. 2.36:

$$\frac{\partial \psi}{\partial t} + c_h^2 \nabla \cdot \vec{B} = 0, \tag{3.18}$$

- an ODE, which represents only the source term of Eq. 2.36:

$$\frac{d\overline{\psi}_{i,j}}{dt} = -\frac{c_h^2}{c_p^2}\overline{\psi}_{i,j}, \tag{3.19}$$

  which is solved for each cell in each time step.

Eq. 3.18 is a homogeneous conservation law, which is a part of the GLM-MHD system of equations. Eq. 3.19 is an ODE, which can be analytically determined. The idea is to solve these two parts separately in an alternating manner for each time step, where the GLM-MHD equations are solved separately (without including the source term, Eq. 3.19) [68]. This step produces the intermediate value for ψ, which we will call ψ^*. To complete the process and update the value of ψ from $t = t_n$ to $t = t_{n+1}$, Eq. 3.19 is analytically solved with $\psi = \psi^*$ as the initial condition

$$\psi^{n+1} = e^{-\Delta t_n \frac{c_h^2}{c_p^2}} \psi^*, \tag{3.20}$$

where $\Delta t_n = t_{n+1} - t_n$. The operator splitting is only formally first-order accurate in time, and it is possible to increase the order of accuracy to second order by using Strang splitting [69], though LeVeque [68] found that in practice, the splitting error between the operator spliting technique described above and the Strang splitting technique is not significant, where in most cases both splitting methods produce solutions which are second-order

accurate [68]. In fact, for Eq. 2.36 in particular, it is found that operator splitting does not degrade the accuracy of the solution in any way (please refer to Sec. 3.1.4.1).

The second option is to integrate the source term as part of the hyperbolic system update. Note that this can be done automatically with high-order accuracy: integrating Eq. 2.36 over cell (i, j) gives

$$\frac{d}{dt}\left(\iint_{\mathcal{A}_{i,j}} \psi \; dA\right) = -c_h^2 \left(\iint_{\mathcal{A}_{i,j}} \nabla \cdot \vec{B} \; dA\right) - \frac{c_h^2}{c_p^2}\left(\iint_{\mathcal{A}_{i,j}} \psi \; dA\right), \tag{3.21}$$

which directly leads to the discrete equation

$$\frac{d\overline{\psi}_{i,j}}{dt} = -\frac{1}{A_{i,j}} \sum_{l=1}^{4} \sum_{m=1}^{N_g} (\omega \vec{\mathbf{f}}_{num} \cdot \vec{n} \Delta l)_{i,j,l,m} - \frac{c_h^2}{c_p^2} \overline{\psi}_{i,j}, \tag{3.22}$$

where $\vec{\mathbf{f}}_{num}$ is the numerical flux function for Eq. 2.36. This is a high-order discretization of Eq. 2.36 as long as the fluxes are computed with high-order accuracy, relying on high-order polynomial reconstruction.

An advantage of the operator splitting approach is that exact analytical integration of Eq. 3.19 does not impose an additional stability constraint on the time step. In contrast, integrating the source term numerically in a coupled fashion as part of the hyperbolic update may incur an additional source term time step constraint of the type $\Delta t \leq C_S \, \Delta t_S$, with $\Delta t_S = 2 \, c_p^2 / c_h^2$ and C_S a constant of $O(1)$ related to the time integration scheme. (For example, C_S would be one for Forward Euler time integration.) This should be compared with the hyperbolic time step constraint of the type $\Delta t \leq C_H \, \Delta t_H$, with C_H the CFL number of the scheme and $\Delta t_H = \Delta x / c_h$, where Δx is a measure of the grid spacing. Recalling the ratio of the diffusive and advective time scales, $\bar{\alpha} = \Delta x \, c_h / c_p^2$ [38, 39], and assuming that $C_S \approx C_H$, it can be seen that the ratio of the hyperbolic and source term time step limits is approximately given by $\Delta t_H / \Delta t_S = \bar{\alpha}/2$. It follows that, for example, for the choices of $\bar{\alpha}$ advocated in [38, 39], namely, $\bar{\alpha} \in [0, 1]$, the source term time step constraint would normally be less stringent than the hyperbolic time step constraint. Also, for constant c_h and c_p, the hyperbolic time step constraint becomes increasingly dominant as the grid is refined. It also follows that setting $\bar{\alpha}$ to a constant for a sequence of grid sizes implies that the ratio between the hyperbolic and source term time step limits remains the same on those grids, which may be an advantage if the source term is integrated numerically as part of the hyperbolic step and one wants to make sure that the hyperbolic time step restriction dominates the source term time step restriction on all grids in the sequence.

While an operator splitting approach may in principle reduce the order of accuracy, depending on the type of the equation, it is often observed that it does not reduce accuracy in practice even if the formal order of accuracy is reduced [68]. We have done extensive numerical comparisons of the two mechanisms for integrating the GLM source term in our

code and have not found any sign of reduced accuracy or reduced convergence order for the operator splitting approach. Further investigation revealed that, in the case of GLM-MHD with mixed hyperbolic-parabolic correction, it can be shown formally that the splitting error vanishes, see Sec. 3.1.4.1.

On a related note, we have also confirmed in numerical tests for smooth flows that employing low-order (e.g., linear) reconstruction for ψ while using reconstruction with degree-three polynomials for the physical variables does not lead to convergence degradation: fourth-order accuracy is maintained. This can be explained by relying on similar arguments as those used in Sec. 3.1.4.1 to show that the operator splitting error vanishes: due to the fact that the exact solution of $\psi(x,t)$ is the zero function for smooth flow, all constants in the Taylor series expansion of the exact solution for ψ vanish, which implies that discretization of ψ does not introduce truncation errors and high-order reconstruction is not required for accurately approximating ψ (which converges to the zero function). Note, however, that during the convergence process the error in ψ is nonzero: it is generated by the truncation error in the components of the magnetic field, and is of the same order of magnitude. Low-order reconstruction of ψ may in principle lead to some computational savings, but it may make implementation somewhat more complex since the reconstruction process for ψ is then different from the other variables.

In our implementation we have several options for integrating the GLM source term, and for choosing the order of reconstruction for ψ and the value of c_p. In the numerical tests presented in Sec. 3.2 section we choose the following options. We reconstruct ψ with the same polynomial order as the physical variables, and we integrate the source term numerically as part of the hyperbolic update. For setting c_p, we followed [28] and set c_p by fixing the constant $c_r = c_p^2/c_h$ to a value of 0.18. We have verified that, for this choice, the hyperbolic time step restriction was dominant for all problems and grids we considered.

3.1.4.1 GLM-MHD Splitting Error Analysis

Consider the conservation-form equations of ideal MHD with mixed hyperbolic-parabolic GLM correction, given by Eqs. 2.1, 2.2, 2.4, 2.35 and 2.36. For simplicity, we consider planar 2D MHD.

Let $\mathbf{U} = [\rho, \rho v_x, \rho v_y, B_x, B_y, e, \psi]^T$ be the vector of conserved variables, and let $\mathbf{V} = [\rho, v_x, v_y, B_x, B_y, p, \psi]^T$ be the vector of primitive variables. Consider smooth solutions of the conservation law. The quasi-linear form of the equations in conservative variables is given by

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbb{A}_c \frac{\partial \mathbf{U}}{\partial x} + \mathbb{B}_c \frac{\partial \mathbf{U}}{\partial y} + \mathbb{C}_c \mathbf{U} = 0, \tag{3.23}$$

and in primitive variables by

$$\frac{\partial \mathbf{V}}{\partial t} + \mathbb{A}_p \frac{\partial \mathbf{V}}{\partial x} + \mathbb{B}_p \frac{\partial \mathbf{V}}{\partial y} + \mathbb{C}_p \mathbf{V} = 0. \tag{3.24}$$

Here, the flux Jacobian matrices $\mathbb{A}_c$ and $\mathbb{B}_c$ are related to the coefficient matrices $\mathbb{A}_p$ and $\mathbb{B}_p$ as follows:

$$\mathbb{A}_c = \frac{\partial \mathbf{U}}{\partial \mathbf{V}} \mathbb{A}_p \frac{\partial \mathbf{V}}{\partial \mathbf{U}}, \qquad \mathbb{B}_c = \frac{\partial \mathbf{U}}{\partial \mathbf{V}} \mathbb{B}_p \frac{\partial \mathbf{V}}{\partial \mathbf{U}}. \tag{3.25}$$

These expressions can be used to compute $\mathbb{A}_c$ and $\mathbb{B}_c$ from

$$\mathbb{A}_p = \begin{bmatrix} v_x & \rho & 0 & 0 & 0 & 0 & 0 \\ 0 & v_x & 0 & -B_x/\rho & B_y/\rho & 1/\rho & 0 \\ 0 & 0 & v_x & B_y/\rho & -B_x/\rho & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & B_y & -B_x & -v_y & v_x & 0 & 0 \\ 0 & \gamma p & 0 & (\gamma-1)\vec{v}\cdot\vec{B} & 0 & v_x & -(\gamma-1)B_x \\ 0 & 0 & 0 & c_h^2 & 0 & 0 & 0 \end{bmatrix}, \tag{3.26}$$

$$\mathbb{B}_p = \begin{bmatrix} v_y & 0 & \rho & 0 & 0 & 0 & 0 \\ 0 & v_y & 0 & -B_y/\rho & -B_x/\rho & 0 & 0 \\ 0 & 0 & v_y & B_x/\rho & -B_y/\rho & 1/\rho & 0 \\ 0 & -B_y & B_x & v_y & -v_x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \gamma p & 0 & (\gamma-1)\vec{v}\cdot\vec{B} & v_y & -(\gamma-1)B_y \\ 0 & 0 & 0 & 0 & c_h^2 & 0 & 0 \end{bmatrix}, \tag{3.27}$$

(see [28]), and from

$$\frac{\partial \mathbf{U}}{\partial \mathbf{V}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_x & \rho & 0 & 0 & 0 & 0 & 0 \\ v_y & 0 & \rho & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \vec{v}\cdot\vec{v}/2 & \rho v_x & \rho v_y & B_x & B_y & 1/(\gamma-1) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.28}$$

$$\frac{\partial \mathbf{V}}{\partial \mathbf{U}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -v_x/\rho & 1/\rho & 0 & 0 & 0 & 0 & 0 \\ -v_y/\rho & 0 & 1/\rho & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \gamma\vec{v}\cdot\vec{v}/2 & -\gamma v_x & -\gamma v_y & -\gamma B_x & -\gamma B_y & \gamma & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.29}$$

It is easily seen from Eq. 2.36 that

$$\mathbb{C}_c = \mathbb{C}_p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_h^2/c_p^2 \end{bmatrix}. \tag{3.30}$$

Further, for Eq. 3.23, define the differential operator

$$\mathbb{D}_c = \mathbb{A}_c \frac{\partial}{\partial x} + \mathbb{B}_c \frac{\partial}{\partial y}. \tag{3.31}$$

Following Section 17.3 of [68], the splitting error $\mathbf{E}$ arising from operator splitting on the time integration of Eq. 3.23 at some time t is given by

$$\mathbf{E} = \frac{1}{2}\Delta t^2 (\mathbb{D}_c\mathbb{C}_c - \mathbb{C}_c\mathbb{D}_c)\,\mathbf{U} + O(\Delta t^3), \tag{3.32}$$

where $\mathbf{U}$ is the exact solution of Eq. 3.23 at that time t. Simple multiplication of the operators $\mathbb{C}_c$ and $\mathbb{D}_c$ shows that

$$(\mathbb{D}_c\mathbb{C}_c - \mathbb{C}_c\mathbb{D}_c)\,\mathbf{U} = \frac{c_h^2}{c_p^2} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{\partial \psi}{\partial x} \\ \frac{\partial \psi}{\partial y} \\ 0 \\ 0 \end{bmatrix} - \frac{c_h^4}{c_p^2} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{\partial B_x}{\partial x} + \frac{\partial B_y}{\partial y} \end{bmatrix}. \tag{3.33}$$

This vanishes because the exact solution of Eq. 3.23 satisfies $\psi(x, y, t) = 0$ and $\nabla \cdot \vec{B}(x, y, t) = 0$ for all x, y and t, showing that operator splitting is at least third-order accurate locally in time (and second-order accurate globally). Again following [68], all higher-order error terms also vanish when $(\mathbb{D}_c\mathbb{C}_c - \mathbb{C}_c\mathbb{D}_c)\,\mathbf{U} = 0$. This shows that no extra error arises from performing the time integration using operator splitting: the result is accurate up to the order of accuracy of the methods used to compute the solution in the separate steps of the operator splitting. Note that this result is obtained essentially because all quantities in the equation for ψ converge to zero for smooth flows, and because there is a source term in the equation for ψ only. The same result can be obtained for the case of 3D MHD in an analogous fashion.

3.1.5 High-Order Accuracy at Curved Boundaries

In our CENO MHD implementation, two general mechanisms are available to prescribe boundary conditions. The first mechanism uses ghost cells. Every grid block in our hierarchical block-adaptive body-fitted quadrilateral grid framework is equipped with three or four layers of ghost cells. In the numerical results to be presented in Sect. 3.2, we compare second-order results with fourth-order results. The second-order simulations employ three layers of ghost cells for each block, and the fourth-order results employ four layers of ghost cells for each block. All blocks have the same size, and the parallelization strategy distributes blocks over parallel message passing interface (MPI) processes [70, 71] as uniformly as possible (with typically multiple blocks per MPI process and one MPI process per CPU core), resulting in adequate load balancing. The ghost cells enable the message passing that parallelizes the code. They are also used in the adaptivity mechanism to transfer information between coarse and fine blocks, as is explained in Sec. 3.2.3. Note that the number of ghost cell layers is one greater than the number required to enable reconstruction in the first ghost cell layer; this additional ghost cell layer is necessary for computing the smoothness indicator in the first layer of ghost cells (which determines whether the high-order or low-order reconstruction is used there) [52]. The ghost cells can also be used to impose boundary conditions at the domain boundaries in standard ways. All second-order simulations use ghost cells to impose boundary conditions. Ghost cells are also used to impose boundary conditions for our fourth-order tests in certain cases, for example in the case of periodic boundary conditions. However, for high-order accuracy near curved boundaries, a more accurate second mechanism for boundary conditions is needed.

The second boundary condition mechanism relies on accurate representation of the curved boundaries with high-order piecewise polynomial splines. It uses one-sided stencils near boundaries that only contain cells within the computational domain, and it imposes additional constraints on the least-squares reconstruction problem at the Gauss points [72]. It is also important to compute the geometric data such as cell areas, centroid locations, etc. to the same order of accuracy as that of the interior scheme [27, 52]. We represent curved boundaries with piecewise polynomial splines of an order consistent with that of the FV numerical scheme, which allows us to locate Gauss quadrature points and compute flux integrals with high accuracy. One-sided reconstruction stencils are used for the first and second layer of cells in the computational domain at the boundaries, and constraints are added to the least-squares reconstruction of the cells in the first layer to accurately impose certain types of boundary conditions on the curved boundaries at the Gauss points. When ghost cells are not used, the one-sided reconstructed values at the Gauss points are directly plugged into the exact MHD flux functions to obtain the numerical flux. For variables to be left free at the boundaries (extrapolation from the computational domain), no additional constraints are necessary. For variables to be imposed at the boundaries, the appropriate constraints are added at the Gauss points used in the flux integration.

More generally, our framework accepts Robin boundary conditions (which consist of

linear combinations of Dirichlet and Neumann conditions), and it also accepts linear relations among variables which form a coupling constraint for a set of reconstructed variables [52]. These coupling constraints can be used to impose wall conditions at curved boundaries with high accuracy. This has been explained for Euler flows in [4, 52], and we extend it here to perfectly conducting walls in MHD problems. For perfectly conducting walls, we impose that $\vec{B} \cdot \vec{n} = 0$ and $\vec{v} \cdot \vec{n} = 0$ in each Gauss quadrature point. Let (n_x^g, n_y^g) be the normal vector in Gauss quadrature point g of cell (i, j), and let (x_g, y_g) be its coordinates. Then, using the polynomial expansion of Eq. 3.4, the conditions $\vec{B} \cdot \vec{n} = 0$ and $\vec{v} \cdot \vec{n} = 0$ at the Gauss point can be expressed as

$$\sum_{\substack{p_1=0 \\ (p_1+p_2\leq k)}}^{k} \sum_{p_2=0}^{k} \left(x_g - \overline{x}_{i,j}\right)^{p_1} \left(y_g - \overline{y}_{i,j}\right)^{p_2} n_x^g \, (D^k_{p_1p_2})_{B_x} + \sum_{\substack{p_1=0 \\ (p_1+p_2\leq k)}}^{k} \sum_{p_2=0}^{k} \left(x_g - \overline{x}_{i,j}\right)^{p_1} \left(y_g - \overline{y}_{i,j}\right)^{p_2} n_y^g \, (D^k_{p_1p_2})_{B_y} = 0, \tag{3.34}$$

and

$$\sum_{\substack{p_1=0 \\ (p_1+p_2\leq k)}}^{k} \sum_{p_2=0}^{k} \left(x_g - \overline{x}_{i,j}\right)^{p_1} \left(y_g - \overline{y}_{i,j}\right)^{p_2} n_x^g \, (D^k_{p_1p_2})_{u} + \sum_{\substack{p_1=0 \\ (p_1+p_2\leq k)}}^{k} \sum_{p_2=0}^{k} \left(x_g - \overline{x}_{i,j}\right)^{p_1} \left(y_g - \overline{y}_{i,j}\right)^{p_2} n_y^g \, (D^k_{p_1p_2})_{v} = 0 \tag{3.35}$$

with $(D^k_{p_1p_2})_{B_x}$ the polynomial coefficients for the B_x magnetic field component, and similar for the B_y, u and v vector components. To impose $\vec{B} \cdot \vec{n} = 0$ in the reconstruction, we solve the least-squares reconstruction problems for the B_x and B_y polynomials together, with the additional constraints of Eq. 3.34 for each Gauss point. Similarly, the least-squares reconstruction problems for u and v are solved together to impose $\vec{v} \cdot \vec{n} = 0$, with the additional constraints of Eq. 3.35. For full implementation details, see [52].

3.2 Numerical Results

In this section we present numerical results that demonstrate high-order convergence for smooth flows and robustness against oscillations for flows with shocks. We present four continuous test problems followed by two problems with discontinuities, including a new MHD extension of the well-known Shu-Osher test problem [25]. Finally, we demonstrate the dynamic AMR capabilities of our implementation using adaptive time-dependent simulations of the Orszag-Tang vortex problem [56] with high-order accuracy and unprecedented effective resolution.

Note that the proposed finite-volume method has been implemented in parallel using the the C++ programming language and MPI [70, 71] and closely follows the techniques and implementations described by Groth and co-workers [57, 58, 59, 60], with the necessary extensions to high-order accuracy as described by Ivan and Groth [27]. All of the numerical

results presented below were obtained using a parallel computing cluster. In particular, the computations were performed on a cluster consisting of 3,780 Intel Xeon E5540 (2.53GHz) nodes with 16GB RAM per node. The cluster nodes were interconnected with a high-speed, low-latency, InfiniBand switched network. We used from 16 up to 512 cores, depending on the problem. Please refer [52] for an assessment of the parallel performance of the high-order finite-volume scheme with AMR, demonstrating the near-ideal parallel efficiency and scalability of the parallel implementation.

3.2.1 Continuous Problems

We first present two smooth test problems on Cartesian grids, which are the rotated Alfvén problem from [29], and the magnetostatic problem from [42]. We then present two continuous test problems on body-fitted multi-block structured grids with non-rectangular cells and curved boundaries: the rotating radial outflow problem and the expanding tube problem from [73].

To quantify the accuracy of the numerical solution, the errors are measured in the L_1, L_2, and L_∞ norms:

$$L_1 = |E|_1 = \frac{1}{A_T} \sum_{i,j} \iint_{\mathcal{A}_{i,j}} \left| u^k_{i,j}(x,y) - u_{exact}(x,y) \right| \, dA, \tag{3.36}$$

$$L_2 = |E|_2 = \sqrt{\frac{1}{A_T} \sum_{i,j} \iint_{\mathcal{A}_{i,j}} \left[u^k_{i,j}(x,y) - u_{exact}(x,y) \right]^2 \, dA}, \tag{3.37}$$

$$L_\infty = |E|_\infty = \max_{i,j} \left(\frac{1}{A_{i,j}} \iint_{\mathcal{A}_{i,j}} \left| u^k_{i,j}(x,y) - u_{exact}(x,y) \right| \, dA \right), \tag{3.38}$$

where A_T is the total area of the computational domain. The integrals are evaluated with high-order accurate Gaussian quadrature, see [52] for details. In most of our numerical tests, we compare convergence for four numerical methods: fourth-order CENO, fourth-order unlimited k-exact reconstruction, second-order CENO, and second-order unlimited k-exact reconstruction. The CENO methods switch between the k-exact reconstruction and the limited piecewise linear reconstruction based on the smoothness indicator. We use a smoothness indicator cut-off value $\mathcal{S}_\mathcal{C} = 800$ except where noted.

3.2.1.1 Rotated Alfvén Travelling Wave Propagation

The circularized Alfvén wave problem from [29] represents analytical solutions of the MHD equations for arbitrary amplitudes. The wave propagates with an angle of $\alpha = 30°$ with respect to a Cartesian grid, and assumes the initial conditions (as in [29]): $\rho = 1$, $v_\parallel = 0$, $p = 0.1$, $B_\parallel = 1$, $v_\perp = B_\perp = 0.1 \sin(2\pi(x\cos(\alpha) + y\sin(\alpha))$, and $v_z = B_z =$

$0.1\cos(2\pi(x\cos(\alpha)+y\sin(\alpha)]$. The parallel velocity, $v_{\parallel}$, is set to zero, which corresponds to the travelling wave test case. The perpendicular and parallel directions are defined with respect to the direction of wave propagation. These initial conditions give an Alfvén speed of 1, which corresponds to a transit period of 1. The computational domain is set to be periodic (using ghost cells), with ranges [0,1/cos(α)] for x, and [0,1/sin(α)] for y. As in [29], the number of cells in the x-direction is equal to the number of cells in the y-direction, which corresponds to a ratio of $1/\sqrt{3}$ between Δx and Δy. The simulations are run for 5 transit periods (or up to $t = 5$). Density and other scalar variables are expected to be constant throughout the simulation since they are not perturbed by the Alfvén wave, and the errors for these variables are much smaller than for $\vec{v}$ and $\vec{B}$, which is consistent with the finding of Toth [29]. Thus, only the accuracy of the $\vec{v}$ and $\vec{B}$ fields were assessed for convergence studies.

As can be seen from Fig. 3.1, the expected order of convergence is achieved for the x-direction magnetic field, at least in the asymptotic limit. For the sake of brevity, only the results of the x-direction magnetic field are shown, but the other variables behave in a similar manner. The effect of the CENO monotonicity-preserving reconstruction switching procedure ($\mathcal{S}_{\mathcal{C}} = 800$) can be seen: a "transition" regime occurs where the mesh is not fine enough and the smooth flow features are not sufficiently resolved. This transition regime does not occur for the fourth-order method because it sufficiently resolves the flow already with low resolution, see also [27]. The high-order scheme represents significant savings in the number of computational cells required for some specific level of accuracy: a 64-by-64 grid resolution was sufficient for the fourth-order scheme to obtain a smaller error than the limited second-order scheme on a 384-by-384 grid.

3.2.1.2 Two-Dimensional Magnetostatic Problem

We next consider the magnetostatic problem from [42]. The exact solution of this stationary problem is known: $\rho = 1, v_x = 0, v_y = 0, v_z = 0, B_x = -\cos(\pi x)e^{-\pi y}, B_y = \sin(\pi x)e^{-\pi y}, B_z = 0, p = 19.84\,(\gamma - 1), \psi = 0$. Following Warburton et al. [42], this exact solution is used as the initial condition for the simulation, and the error at steady-state is a measure of the deviation of the numerical solution from the exact solution. The second-order methods use ghost cells to impose boundary conditions: $\vec{v}$ and $\vec{B}$ are imposed in the ghost cells (accurate average values of the exact solution, obtained by numerical quadrature), and ρ, p and ψ are extrapolated to the ghost cells (the average values are linearly extrapolated). The fourth-order methods use one-sided reconstruction, with the exact values of $\vec{v}$ and $\vec{B}$ imposed at the Gauss points using constraints, and ρ, p and ψ are left free at the Gauss points. Figure 3.2 shows how the error norm of B_x converges to zero as a function of grid size with the expected order. The fourth-order scheme requires much fewer computational cells to achieve a specified level of error (in this case, the error can differ by as much as 4 orders of magnitude for the same number of cells).

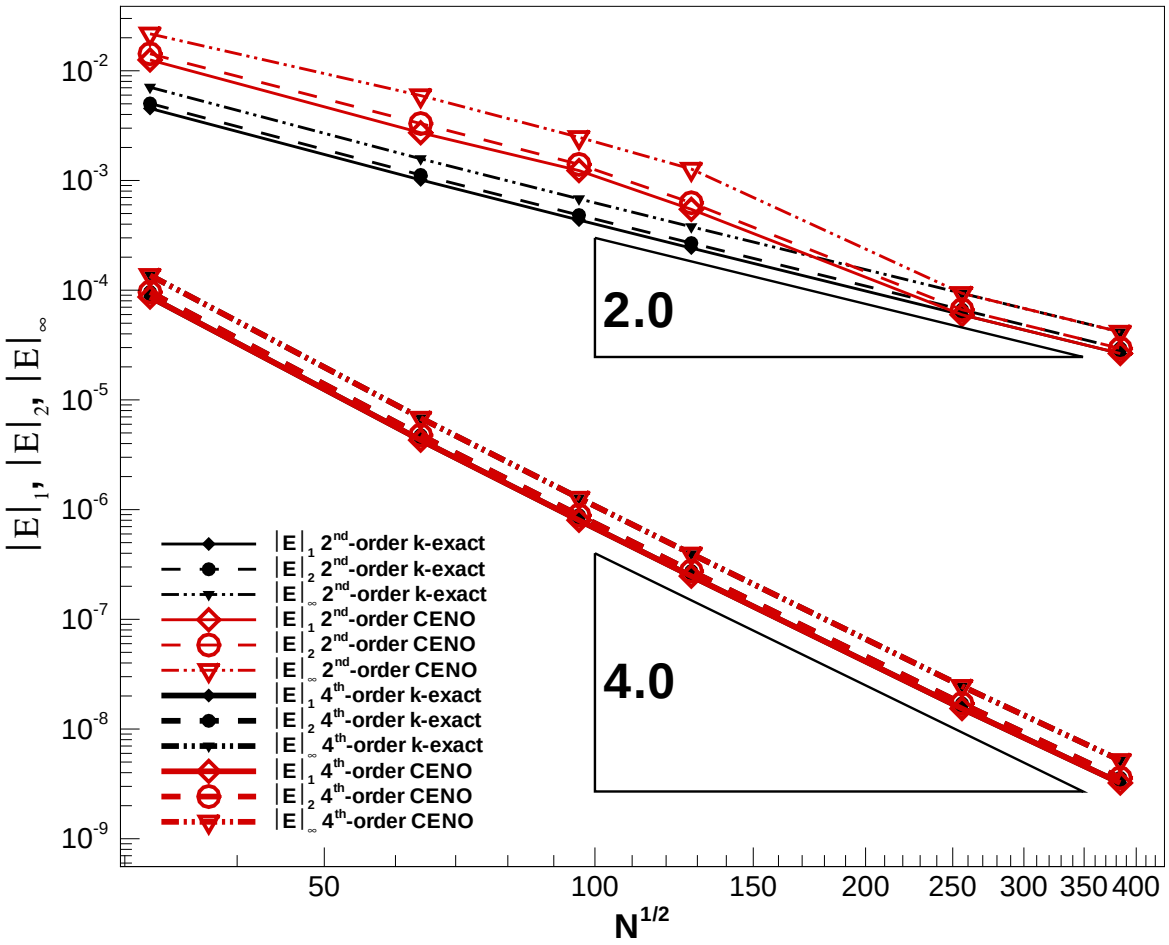


Figure 3.1: The L_1-, L_2-, and L_∞-norm errors for the magnetic field in the x-direction for the rotated Alfvén wave problem, calculated at $t = 5$ (five transit periods). N is the total number of grid cells. The solution is compared with the initial conditions to compute the error. The error converges to zero with the expected order of accuracy in the asymptotic limit. A transition region is observed for the second-order CENO scheme, consistent with the findings of Ivan and Groth [27, 52].

3.2.1.3 Superfast Rotating Outflow From a Cylinder

We next consider the rotated outflow problem from [73] on a body-fitted structured grid with non-rectangular cells and curved boundaries. While the exact analytical solution is not available, several theoretical flow invariants are available [73], with which the corresponding computed quantities can be compared. We measure error in the entropy, s, and the radial magnetic field, B_r.

The problem is defined on a domain between two concentric circles, and superfast inflow

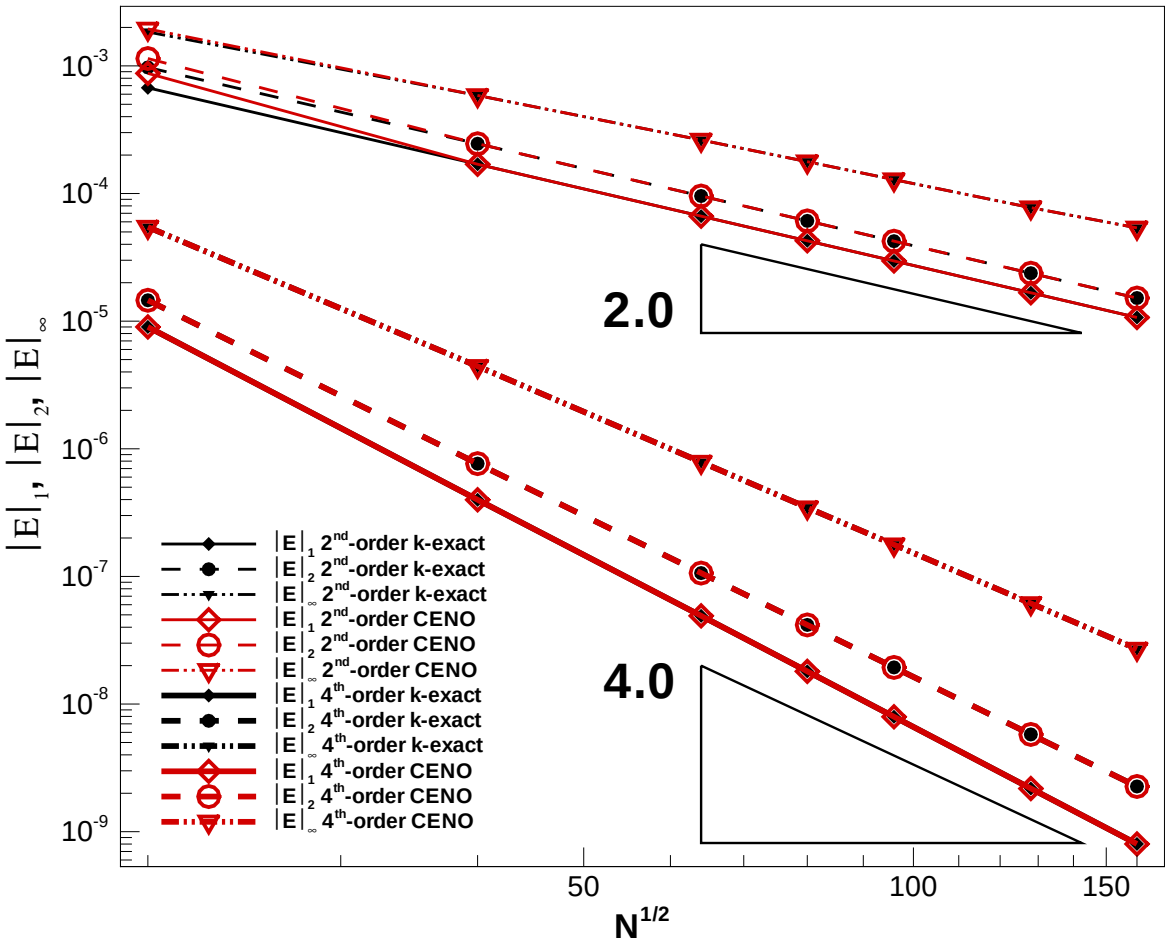


Figure 3.2: The L_1-, L_2-, and L_∞-norm errors for the magnetic field in the x-direction for the magnetostatic problem. The steady-state solution is compared with the initial conditions to compute the error. The error converges to zero with the expected order of accuracy in the asymptotic limit.

conditions (normal velocity faster than the fast magnetosonic wave speed as given in Eq. 2.7) are imposed at the inner circle. The domain goes from $r = 1$ to $r = 6$, and the inflow conditions imposed at the $r = 1$ boundary are $\rho = 1$, $p = 1$, $v_r = 3$, $v_\theta = 1$, and $B_r = 1$. The second-order methods use ghost cells to impose boundary conditions. At the inner boundary (inflow), ρ, p, $\vec{v}$ and $\vec{B}$ are imposed in the ghost cells using linear interpolation (to impose the desired values exactly at the domain boundary), and ψ is extrapolated linearly from the interior of the domain. At the outer boundary (outflow), ρ, p, $\vec{v}$ and $\vec{B}$ are extrapolated linearly, and ψ is set to zero using linear interpolation (to impose the desired value exactly at the domain boundary). The fourth-order methods use high-order piecewise polynomial spline representation of the curved boundaries, combined with one-

sided reconstruction and constraints. At the inner boundary (inflow), ρ, p, $\vec{v}$ and $\vec{B}$ are imposed by constraints at the Gauss points, and ψ is left free. At the outer boundary (outflow), ψ is set to zero by constraints at the Gauss points, and ρ, p, $\vec{v}$ and $\vec{B}$ are left free.

The steady-state solution of the rotated outflow problem obtained with the fourth-order CENO scheme on a mesh with 80-by-80 cells can be seen in Fig. 3.3. The magnetic

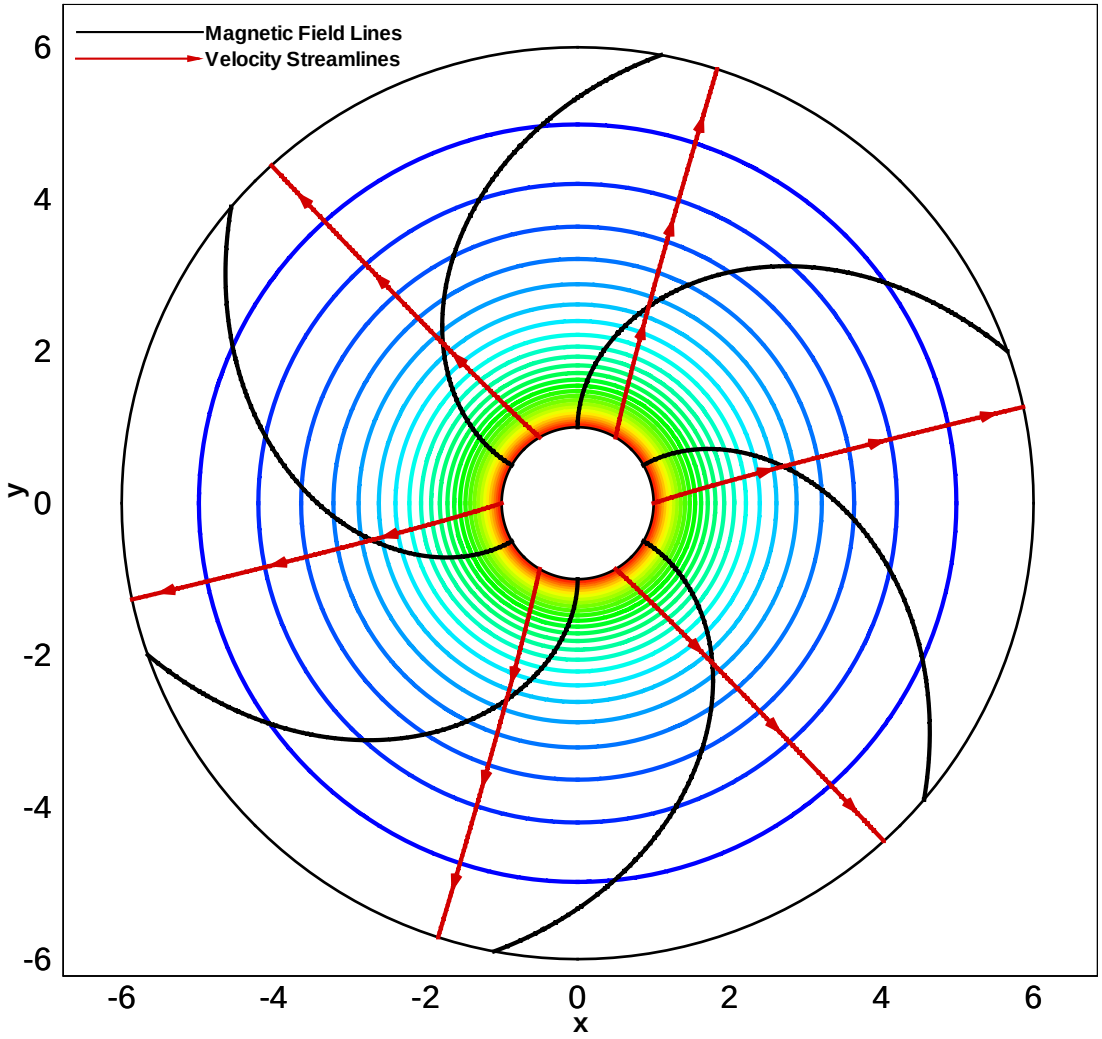
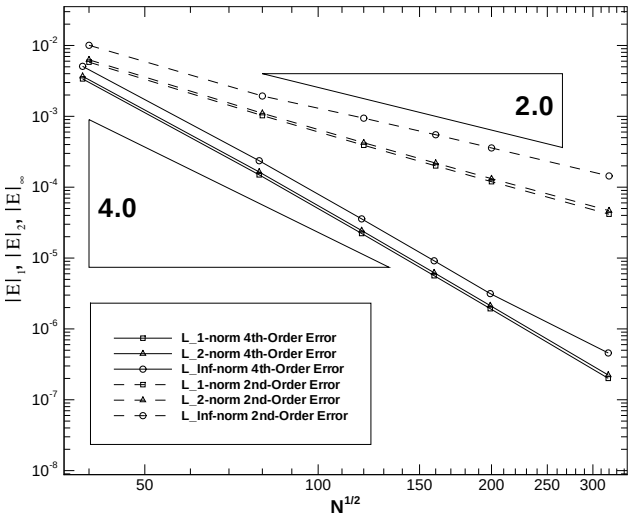
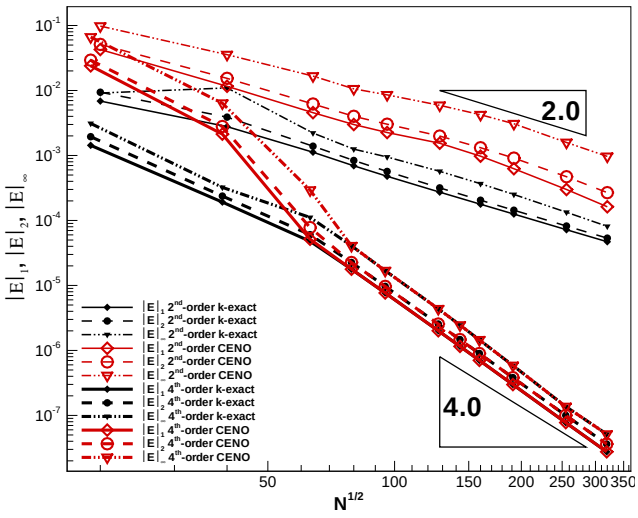


Figure 3.3: Density contour lines, magnetic field lines, and streamlines for the rotated outflow problem, obtained on a mesh with 80-by-80 cells. The magnetic field lines and the streamlines are not aligned. The flow is smooth throughout the entire domain, which enables high-order convergence rates. The density contour lines are equally spaced in the range (0.17,0.97).

field lines are not aligned with the streamlines. The solutions obtained with second- and fourth-order CENO schemes are compared in Fig. 3.4.



(a) The L_1-, L_2-, and L_∞-norm errors for entropy, which is constant in the domain.

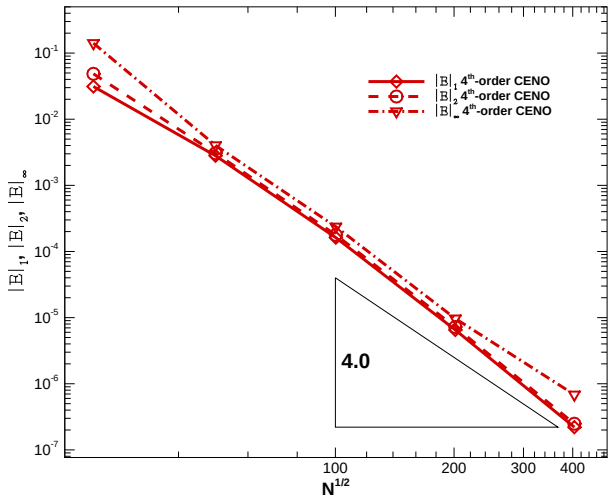


(b) The L_1-, L_2-, and L_∞-norm errors for the radial magnetic flux. While there is no analytical solution for the full magnetic field, the radial component of the magnetic field can be determined, due to the conservation of the radial magnetic flux.

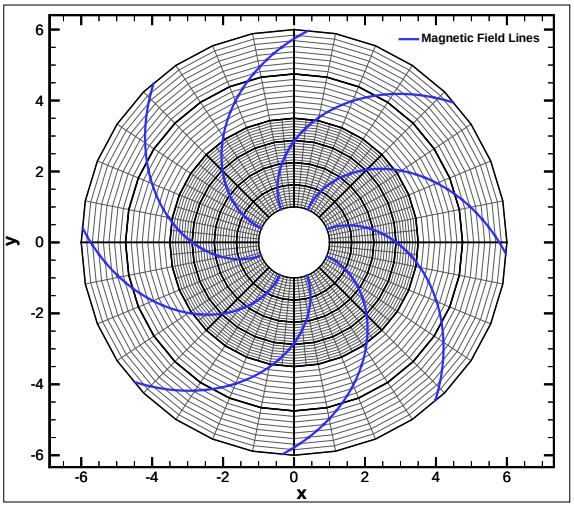
Figure 3.4: Convergence study for the rotated outflow problem using both unlimited k-exact reconstruction (black lines) and CENO with $\mathcal{S}_\mathcal{C} = 800$ (red lines).

It can be seen that the errors converge to zero with the expected order of accuracy. For this problem, the second-order scheme has not reached the asymptotic regime beyond the transition region yet for the resolutions we tested, and the second-order CENO error remains above the unlimited second-order error due to ongoing switching from unlimited k-exact reconstruction to limited second-order reconstruction, especially for the error in the radial magnetic field as seen in Fig. 3.4b. This is possibly due to the inability of the piecewise linear function to capture the curvature of the boundaries properly, so the switching procedure continues to see some cells close to the boundaries as under-resolved, thus limiting the reconstruction functions at these places and affecting the magnitude of the error. In contrast, for fourth-order CENO and the unlimited k-exact schemes produce the same error for resolutions above 80-by-80. It is clear that our approach can handle curved boundaries with high-order accuracy, and the fourth-order method requires significantly fewer cells than the second-order method to obtain a given error level.

In Fig. 3.5 we illustrate that the high-order CENO-GLM approach can naturally handle resolution changes on block-adaptive grids and the errors introduced by AMR restric-



(a) The L_1-, L_2-, and L_∞-norm errors for the radial magnetic flux obtained with the fourth-order CENO scheme on a grid with resolution change between blocks (right panel). Fourth-order accuracy is maintained.



(b) The grid with resolution change used for the convergence study in the left panel. This grid has 8 coarse blocks and 32 fine blocks, each with 8×8 cells. In each successive point of the convergence study, all blocks in the grid are refined by dividing them into four blocks. The magnetic field lines of the solution on this grid are plotted in blue. The smoothness of the magnetic field lines is not adversely affected by the jump in grid resolution.

Figure 3.5: Convergence study for the rotated outflow problem on a grid with change in grid resolution between blocks, illustrating that the high-order CENO-GLM approach naturally handles resolution changes on block-adaptive grids.

tion/interpolation. We have performed a grid convergence study on a series of grids where there is a change in resolution between some blocks (corresponding to one level of refinement). In each successive point of the convergence study, all blocks in the grid of Fig. 3.5b are refined by dividing them into four blocks. The convergence plot of Fig. 3.5a shows that fourth-order convergence is automatically maintained by the CENO-GLM approach, and magnetic field lines remain smooth at the jump in grid resolution.

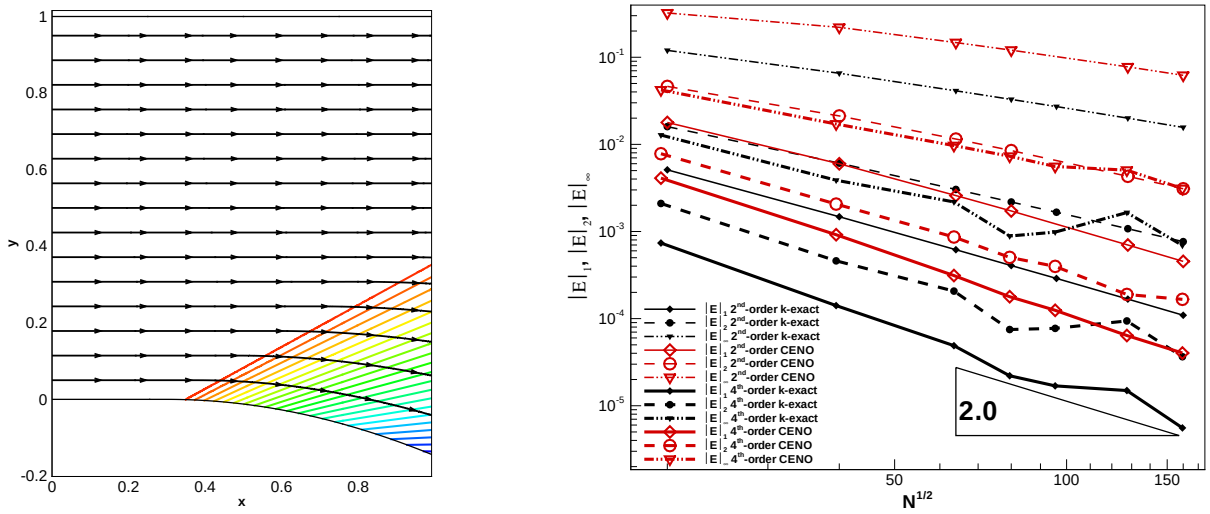
3.2.1.4 Expanding Tube Problem

The expanding tube problem from [73] is another continuous problem that uses a body-fitted structured grid with curved boundaries. It models plasma flow in an expanding tube, which gives rise to an MHD solution that contains a rarefaction wave with a weak

discontinuity at the edge of the rarefaction (see Fig. 3.6). Across the weak discontinuity, the first spatial derivative of the flow variables is discontinuous. Therefore, even if fourth-order reconstruction accuracy is targeted (by employing degree-3 polynomial functions to reconstruct the solution), the solution accuracy is still limited to second-order near the weak discontinuity.

The flow is simulated on a domain with $x \in [0, 1]$, and $y \in [y_0(x), 1]$, where $y_0(x) = \cos\left(\frac{\pi}{4}(x - 0.3)\right) - 1$ for $x \in [0.3, 1]$, and zero elsewhere in the domain. The lower wall starts to curve at $x = 0.3$, giving rise to a rarefaction wave downstream of the weak discontinuity. The boundary curve follows a cosine function rather than a straight line to avoid a geometrical singularity in the boundary, which results in the rarefaction wave not converging to a single point, as can be seen from Fig. 3.6a [73]. At the $x = 0$ boundary, a uniform inflow with the following conditions is imposed: $\rho = 1$, $p = 1$, $v_x = 8$, and $B_x = 4$. These initial conditions correspond to superfast horizontal inflow conditions, with an acoustic Mach number $v_x/c = 8\sqrt{3/5}$, and Alfvénic Mach number $v_x/c_{A_x} = 2$. The second-order methods use ghost cells to impose boundary conditions. The superfast inflow and outflow boundary conditions at the left and right boundaries, respectively, are implemented as for the rotating outflow test problem of Sect. 3.2.1.3. For the top and bottom boundaries, standard wall boundary conditions are implemented that symmetrically copy ρ, p and ψ to the ghost cells, and mirror $\vec{v}$ and $\vec{B}$ with respect to the wall. The fourth-order methods use high-order piecewise polynomial spline representation of the wall boundaries, combined with one-sided reconstruction and constraints. For the top and bottom wall boundaries, ρ, p, ψ and the tangential components of $\vec{v}$ and $\vec{B}$ are left free, while the normal components of $\vec{v}$ and $\vec{B}$ are set to zero at the Gauss points using constraints. The high-order outflow boundary condition is handled as in the rotating outflow test problem of Sect. 3.2.1.3. We simply use ghost cells for the high-order inflow boundary condition, since the flow remains uniform close to the inflow boundary and ρ, p, $\vec{v}$ and $\vec{B}$ can just be imposed in all ghost cell layers, while ψ can be extrapolated linearly.

To assess the accuracy of the solution, entropy, which is one of the invariants for this flow, is measured. Figure 3.6b shows convergence analysis of the entropy error. As can be seen in this figure, second-order accuracy is achieved for the L_1-norm error of the entropy for both the second-order and the fourth-order accurate methods. While Fig. 3.6b illustrates how the weak discontinuity in the solution limits the order of accuracy, reduction in the total error is still observed when higher-order polynomial functions (the fourth-order method) are used to represent the solution. It is interesting to note that the log error of the CENO solution decreases linearly, whereas some zigzagging is present in the unlimited k-exact error plot. This can be explained by the fact that the flow is not fully smooth, and the weak discontinuity that exists can potentially generate spurious oscillations when monotonicity is not enforced, though the level at which these oscillations occur is apparently much smaller than the solution variation. Note also that, even at the highest attempted resolution, the convergence plots of the CENO error do not converge to those of unlimited k-exact reconstruction (as was the case for the other test cases), implying that, due to the



(a) The expanding tube flow, solved on a 160-by-160 grid with the fourth-order CENO scheme. The contour lines are equally spaced in the range (0.45,0.95) (21 contours).

(b) Convergence study for the expanding tube problem. The entropy is compared with the entropy at the inflow to compute the error,and nearly second-order accuracy for the L_1-norm error is observed for both the second- and fourth-order methods.

Figure 3.6: Expanding tube flow: density contour lines and entropy convergence study. The error converges with at most second-order accuracy, due to the non-existence of higher-order derivatives across the weak discontinuity. Convergence study is performed for entropy using both unlimited k-exact reconstruction (black lines) and CENO with $\mathcal{S}_\mathcal{C} = 800$ (red lines).

weak discontinuity, reconstruction switching is always performed for at least a few cells.

3.2.2 Problems with Discontinuities

It is clear from Sec. 3.2.1 that the high-order scheme is indeed advantageous. It is also important to assess how well the scheme handles problems with discontinuity, as many realistic flows involve some forms of discontinuities. A continuous problem with a weak discontinuity has already been considered in Sec. 3.2.1.4, in the form of the expanding tube problem (as proposed in [73]). From this test case, we can see how the high-order scheme improves the accuracy of the solution, but is limited to second-order accuracy in the asymptotic limit. In this section, we consider two discontinuous problems: the classic MHD Brio-Wu shock tube test case (Sec. 3.2.2.1) and a new MHD extension of the Shu-Osher problem (Sec. 3.2.2.2), to illustrate the capability of the proposed high-order scheme in handling discontinuities in the solution.

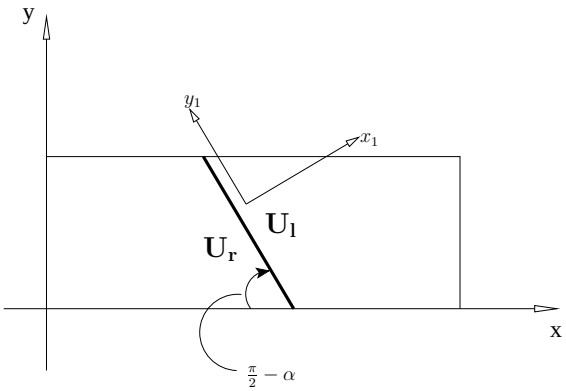


Figure 3.7: Setup for the rotated one-dimensional problems with discontinuities. The discontinuity is rotated 45° counterclockwise with respect to the x-axis. The solution thus exhibits a translational symmetry in the y_1-direction.

3.2.2.1 Rotated Brio-Wu Shock Tube Problem

The Brio-Wu shock tube problem [74] is a standard test case to demonstrate the capability of a numerical MHD scheme to handle discontinuities.

The initial conditions are given by

$$(\rho, v_\perp, v_\parallel, v_z, B_\perp, B_\parallel, B_z, p, \psi) = \begin{cases} (1, 0, 0, 0, 0.75, 1, 0, 1, 0) \text{ for } x_1 < 0, \\ (0.125, 0, 0, 0, 0.75, -1, 0, 0.1, 0) \text{ for } x_1 > 0, \end{cases} \tag{3.39}$$

with $\gamma = 2$. Here, x_1 is the coordinate variable perpendicular to the shock, given by $x_1 = x\cos\alpha + y\sin\alpha$ with α the angle at which the shock frame of reference is rotated with respect to the x-axis (we choose α =45°). This setup is illustrated in Fig. 3.7.

Ghost cells are used and constant extrapolation boundary conditions are applied to all boundaries, though the top and the bottom boundaries require that the cells not only be copied to the ghost cells, but also shifted to the left or the right by one cell (similar to Fig. 10 from [29]). It is important to note that, for this boundary condition to work, the ratio between the spacing in the x-direction and the spacing in the y-direction needs to be 1, because otherwise the 45° symmetry would not translate to a (1,1) translational symmetry, see also [29, 39, 75].

The Brio-Wu problem gives rise to several types of waves and shocks: fast rarefaction waves, a contact discontinuity, a slow compound wave, and a slow shock [74]. Simulations were performed for the rotated cases using 600 cells in the x-direction, and 4 cells in the y-direction. The density plot is shown in Fig. 3.8, and illustrates that our method is robust with respect to spurious oscillations. The fourth-order solution has slightly sharper features than the second-order solution. All of the important wave features are captured well without spurious oscillations, except for a slight undershoot between the fast rarefaction

(FR) and the slow compound wave (SM) (which is also observed in other work on high-order MHD schemes [39, 45]).

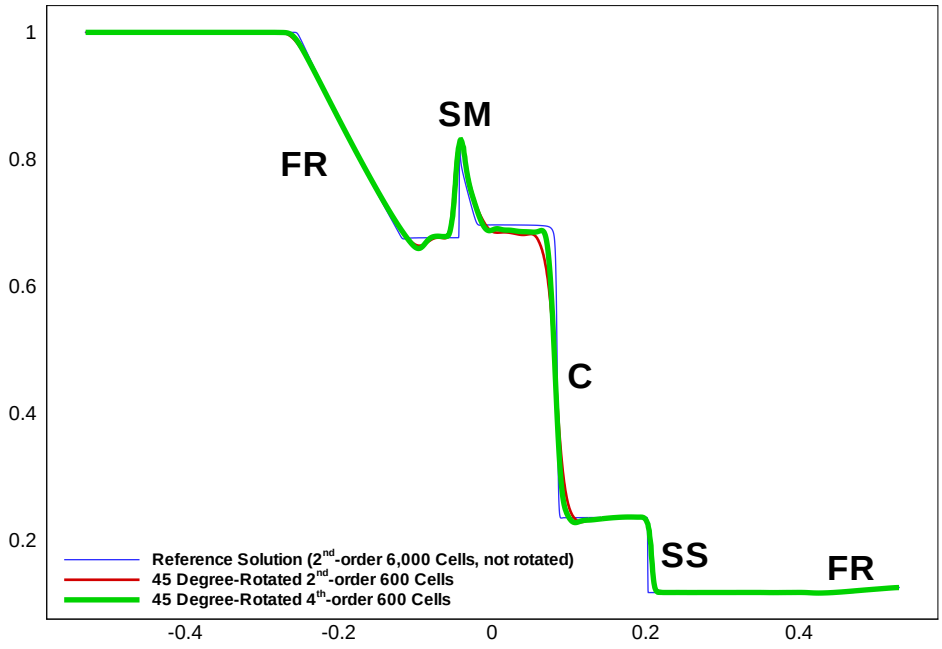


Figure 3.8: Comparison of the density solution of the Brio-Wu Shock Tube Problem at $t = 0.1414$, rotated at 45°. Here, FR denotes fast rarefaction, SM slow compound wave, C contact discontinuity, and SS slow shock. A cutoff value of $\mathcal{S}_\mathcal{C} = 8,000$, is chosen for these simulations.

3.2.2.2 MHD Extension to Shu-Osher Shock Tube Problem

The shock tube problem proposed by Shu and Osher [25] is commonly used to test the ability of high-order numerical schemes to resolve small-scale flow features in the presence of shocks. A sinusoidal density perturbation is added downstream of a purely advecting supersonic shock wave. The interaction of the shock wave with the sinusoidal part of the density field gives rise to fast oscillations and complex flow features downstream to the shock. The Shu-Osher shock tube problem provides an excellent testbed to highlight the benefits of the improved accuracy of high-order numerical schemes, while at the same time the presence of the shock puts the robustness and stability of the schemes to test. In what follows, we develop a new MHD version of the Shu-Osher shock tube problem.

First consider a fast travelling shock wave without the sinusoidal density perturbation. The fast shock wave advects with shock speed s, satisfying the Rankine-Hugoniot condition:

$$s(\mathbf{U_r} - \mathbf{U_l}) = \mathbf{F}(\mathbf{U_r}) - \mathbf{F}(\mathbf{U_l}) \tag{3.40}$$

where $\mathbf{U_r}$ and $\mathbf{U_l}$ denote the state vectors of the right and left state, respectively, and $\mathbf{F}(\mathbf{U_i})$ denotes the flux evaluated at state i. The stable purely advecting fast shock that is desired for the MHD equivalent of the Shu-Osher shock tube problem needs to satisfy the following conditions in addition to the Rankine-Hugoniot condition:

1. In the shock frame, the flow should move in the direction from low pressure to high pressure to ensure that entropy increases across the shock;
2. In the shock frame, velocities normal to the shock need to be faster than the fast magnetosonic wave speed (Eq. 2.7) upstream, and faster than the Alfvén wave speed (Eq. 2.9) but slower than the fast magnetosonic wave speed downstream;
3. In the simulation frame of reference, the normal velocity downstream of the shock should be zero so the density perturbation stays intact until the shock goes through it;
4. The magnetic field normal to the shock should be continuous to ensure zero magnetic field divergence.

In the shock frame, the shock is stationary ($s = 0$), so the Rankine-Hugoniot condition (Eq. 3.40) simplifies to

$$\mathbf{F}(\mathbf{U_l}) = \mathbf{F}(\mathbf{U_r}). \tag{3.41}$$

We choose the following initial conditions that satisfy conditions 1-4 and the Rankine-Hugoniot condition:

$$(\rho, u_\perp, u_\parallel, u_z, B_\perp, B_\parallel, B_z, p, \psi) = \begin{cases} (1, 0, 0, 0, 1, 1, 0, 1, 0) \text{ for } x < 4, \\ (3.5, 5.8846, 1.1198, 0, 1, 3.6359, 0, 42.0267, 0) \text{ for } x > 4. \end{cases} \tag{3.42}$$

The numbers in Eq. 4.43 were obtained by numerically solving the MHD Rankine-Hugoniot condition, and were rounded to four decimal digits (which is sufficiently accurate for the numerical tests). These initial conditions lead to a shock that travels unperturbed to the left with a speed of 8.2385 (rounded).

Equation 4.43 represents the unperturbed portion of our newly proposed MHD version of the Shu-Osher shock tube problem. Similar to the Shu-Osher problem, sinusoidal perturbation is added to the downstream part of the density field (because the shock and the flow travel to the left, the sinusoidal perturbation is added to ρ_l). The initial density function is then chosen as

$$\rho_l = 1 + 0.2\sin(5x), \;\; \rho_r = 3.5 \tag{3.43}$$

and all the other variables are kept as given in Eq. 4.43.

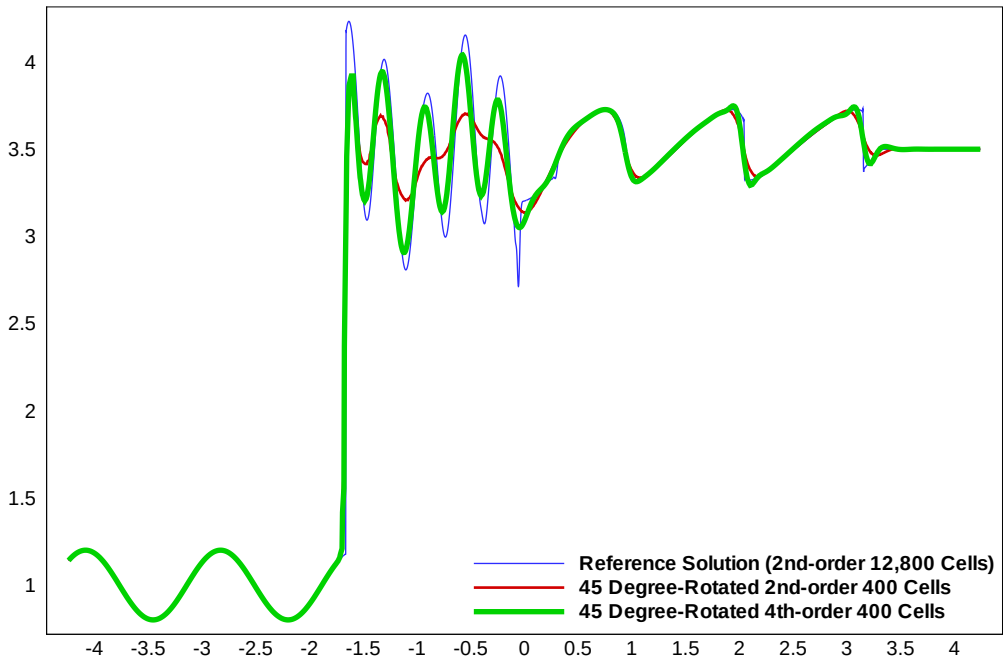


Figure 3.9: Comparison of the density solution of the MHD Shu-Osher problem at $t = 0.6906$, rotated at 45°. As can be observed from the figure, the fourth-order method produces results that are much closer to the non-rotated reference result in the highly oscillatory region, illustrating the benefits of high-order accuracy. A cutoff value of $\mathcal{S}_\mathcal{C} = 80$ is used for these simulations.

As in the case of the rotated Brio-Wu problem, the initial condition given by Eq. 4.43 and 4.44 has been applied in the rotated frame of reference $x_1 - y_1$ (see Fig. 3.7). The boundary conditions for our simulation of this problem are as explained in Sect. 3.2.2.1. The left and right boundaries are taken sufficiently far from the initial discontinuity, such that they do not influence the solution. The comparison of the density profiles between the different methods is shown in Fig. 3.9. The benefit of the high-order method is clear: using the same number of cells, the fourth-order method captures the small-scale flow features much better than the second-order method. For all simulations performed for this section, no stability or overshoot problem were observed, which indicates that the monotonicity-preserving mechanism is doing its job properly to ensure that the method is stable in the presence of discontinuities.

3.2.3 Application of CENO with Dynamic Adaptive Mesh Refinement: Orszag-Tang Vortex Problem

In this section we demonstrate the pre-existing dynamic AMR capabilities of our framework using adaptive time-dependent simulations of the Orszag-Tang vortex problem [45, 49, 56] with high-order accuracy and unprecedented effective resolution. The problem is challenging because it is time-dependent and contains multiple complex and interacting discontinuities. The Orszag-Tang vortex problem is a good test for our dynamic adaptive mesh refinement and coarsening procedure. We include this problem in this thesis because it illustrates convincingly how the high-order CENO-GLM approach we developed fits directly into the pre-existing AMR framework and how our approach can be used with high-order accuracy on grids with dynamically adaptive resolution changes between blocks.

We have implemented our new high-order MHD scheme that combines CENO and GLM into a hierarchical quadtree block-based AMR procedure for multi-block body-fitted quadrilateral mesh that is based on the previous work of Groth and co-workers [57, 58, 59, 60] and is extended to high-order accuracy as in [27]. We give a brief summary of the approach, and details are described in [27, 52, 58]. In our hierarchical quadtree block-based AMR algorithm, mesh adaptation is accomplished by dividing and coarsening appropriate solution blocks. In regions requiring increased cell resolution, a 'parent' block is refined by dividing it into four 'children'. Each of the four quadrants or sectors 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. This process can be reversed in regions that are deemed over-resolved and four children can be coarsened into a single parent block. The mesh refinement is constrained such that the grid resolution changes by at most a factor of two between adjacent blocks, and the minimum resolution is not less than that of the initial mesh. A hierarchical quadtree data structure and additional interconnects between the 'leaves' of the trees are used to keep track of mesh refinement and the connectivity between solution blocks. The hybrid CENO solution reconstruction procedure is used in conjunction with standard multigrid-type restriction and interpolation operators to evaluate the solution on all blocks created by the coarsening and division processes. Interpolation is performed with high-order accuracy by computing reconstructed polynomials for solution variables in each coarse-grid cell and integrating them over the fine-grid children cells to determine the fine-grid cell averages with high-order accuracy (see [27] for details). Restriction and interpolation are performed in such a way that conservation is maintained, but in our CENO-GLM MHD approach no special treatment is required for restricting or interpolating the cell-centred magnetic fields: restriction or interpolation may introduce errors of the order of the discretization error, and they are handled properly by the GLM mechanism for controlling $\nabla \cdot \vec{B}$.

Grid refinement and coarsening are based on the maximum value of the CENO smooth-

ness indicator over each block for the density variable. For each cell, the variable

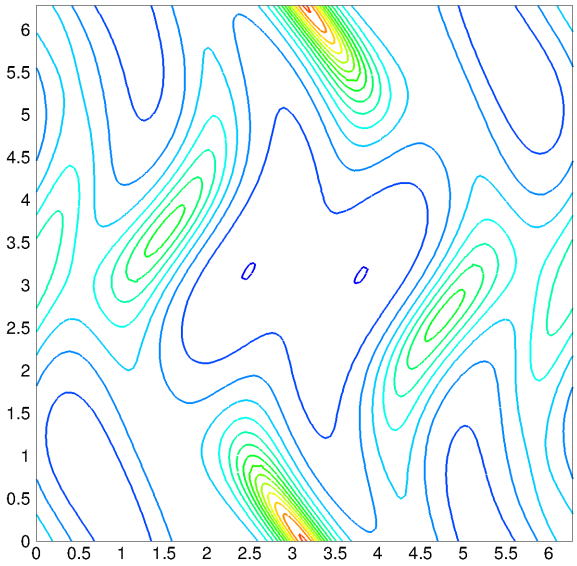
$$\mathcal{R}_c = e^{-\frac{\max(0,\mathcal{S})}{\mathcal{S}_c}} \tag{3.44}$$

is calculated, where $\mathcal{S}$ is the value of the smoothness indicator and $\mathcal{S}_c$ is the cutoff value for the smoothness indicator. The range of $\mathcal{R}_c$ is (0,1]. The maximum $\mathcal{R}_c^B$ of all $\mathcal{R}_c$ values within a block is computed. In blocks with $\mathcal{R}_c^B$ close to 0, all cells are smooth and resolved, and blocks with $\mathcal{R}_c^B$ close to 1 have cells that are nonsmooth or under-resolved. The block-based $\mathcal{R}_c^B$ values are compared with refinement and coarsening thresholds to determine if a block should undergo refinement, or if a group of blocks should be combined for coarsening. Full details on the algorithm followed for coarsening and refinement are given in [58] and [52]. The refinement/coarsening algorithm is invoked at regular intervals during the simulation to obtain dynamic AMR.

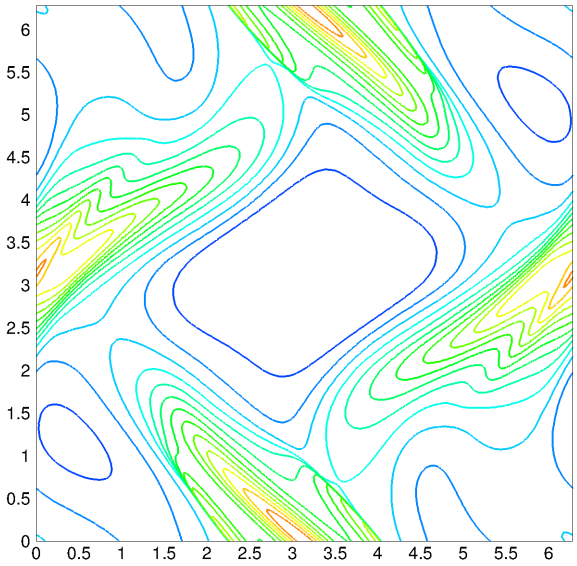
For the Orszag-Tang vortex problem, the same initial conditions and domain as in [45] are used, with $\rho = \gamma^2$, $v_x = -\sin(y)$, $v_y = \sin(x)$, $B_x = -\sin(y)$, $B_y = \sin(2x)$, and $p = \gamma$. The remaining variables (v_z, B_z, and ψ) are initialized to zero. The computational domain is a square with x and y values between 0 and 2π, and periodic boundary conditions (ghost cells are used). The simulation is performed with CENO cutoff tolerance $\mathcal{S}_\mathcal{C} = 500$. The mesh is refined every 0.025 seconds up to $t = 1$. For later times, AMR is performed every 50 time steps because Δt decreases rapidly. The contour lines of the density for the Orszag-Tang vortex problem are shown at $t = 0.5$, $t = 1.0$, $t = 2.0$, and $t = 3.0$ in Fig. 3.10. The results show agreement with results shown in other papers [29, 45, 49, 76]. Figure 3.11 shows the sequence of adaptive meshes. Comparing the density contour lines shown in Fig. 3.10 with the way the grid is refined as shown in Fig. 3.11, it can be seen that the refinement closely follows the parts of the solution where interesting flow features and discontinuities occur, illustrating the effectiveness of the smoothness indicator-based refinement criterion.

Following [45] and [49], pressure distribution cuts at $t = 2.0$ and $t = 3.0$ along the line $y = 1.9635$ are shown in Fig. 3.12. The AMR results are compared to results on a uniform 1,024-by-1,024 mesh. The uniform mesh corresponds to the smallest cell resolution at 7 levels of refinement, while 8 levels of refinement are used in the AMR results, so that the smallest cell in the adaptive mesh (Fig. 3.11) corresponds to a resolution of 2,048-by-2,048 if done uniformly. From Fig. 3.12, it can be seen that the AMR results in general agree well with the uniform reference results. The AMR solution (as shown in Fig. 3.10) agrees well with the uniform reference solution (which is not shown) and with solutions shown in the literature, but our results have a much higher effective resolution than previously shown results (and they are fourth-order accurate). It is also interesting to note that, while the uniform mesh has 1,048,576 computational cells, the AMR mesh has 865,408 cells at $t = 3.0$, which is smaller than the uniform mesh, despite having twice the effective resolution at the highest level of refinement. Note also that, before $t = 3.0$, much fewer cells are used by the adaptive simulation (see Fig. 3.11). Numerical experiments show that,

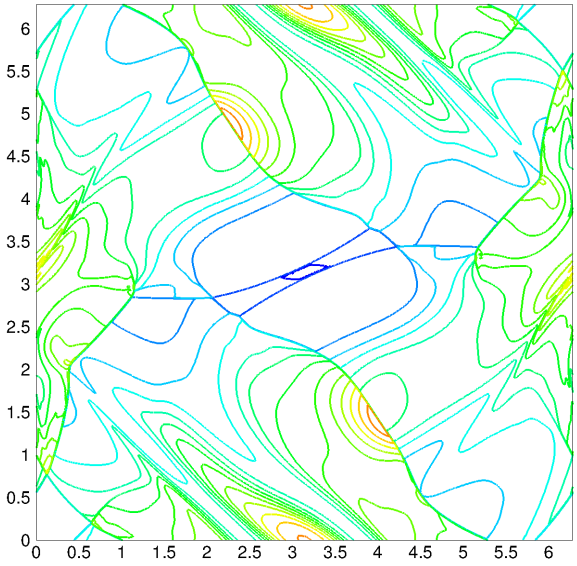
for the adaptive simulation of Fig. 3.11, the speedup compared to a uniform simulation with the same maximal resolution (on a 2048-by-2048 mesh) is 2.066. This illustrates the effectiveness of the CENO scheme in combination with the block-based AMR algorithm to reduce the number of required computational cells.



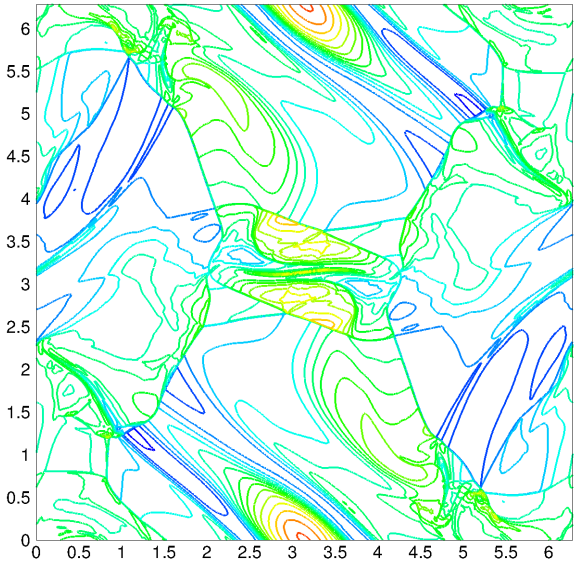
(a) Density solution at $t = 0.5$. The contour lines are equally spaced in the range (2.11,5.82) (15 contours).



(b) Density solution at $t = 1.0$. The contour lines are equally spaced in the range (1.25,6.9) (15 contours).

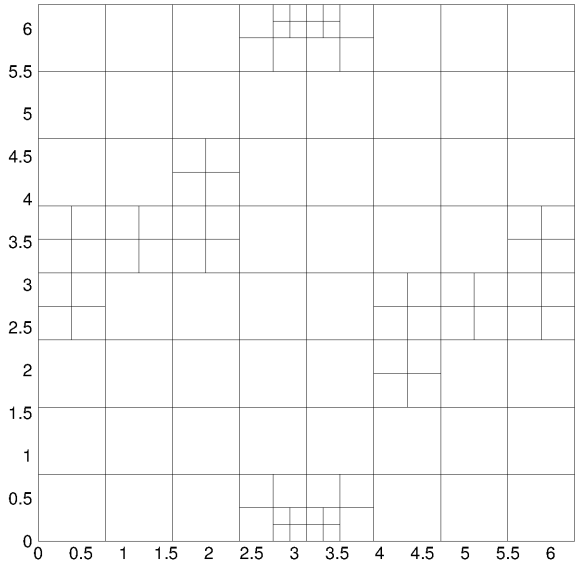


(c) Density solution at $t = 2.0$. The contour lines are equally spaced in the range (0.62,6.41) (15 contours).

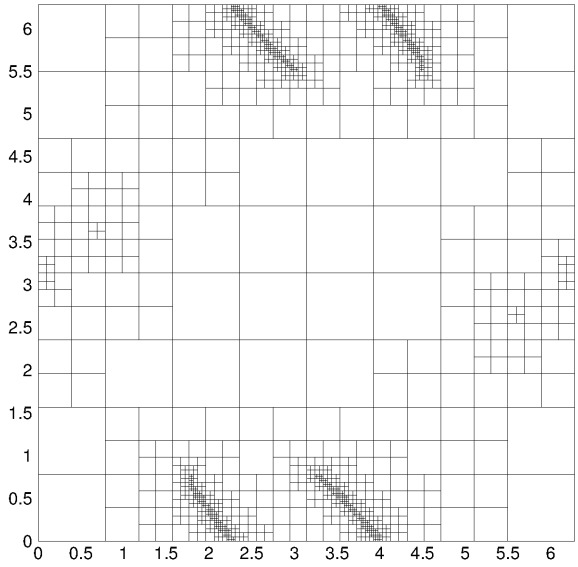


(d) Density solution at $t = 3.0$. The contour lines are equally spaced in the range (1.16,6.42) (15 contours).

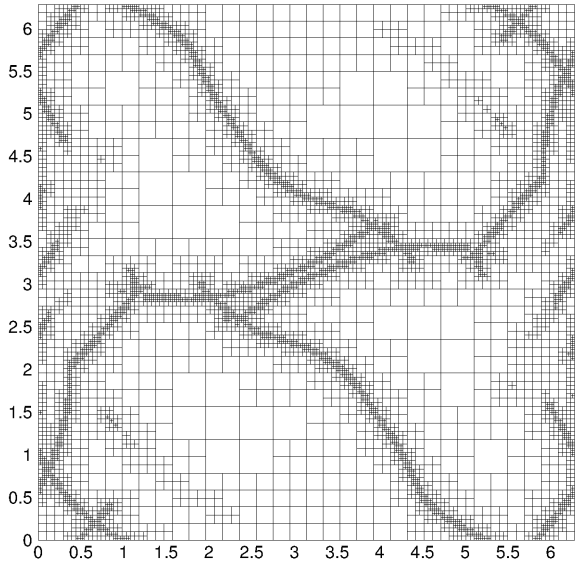
Figure 3.10: The evolution of density for the Orszag-Tang vortex problem at different times: $t = 0.5$, $t = 1.0$, $t = 2.0$, and $t = 3.0$. The ranges for the contour lines shown here are as in [45]. These fourth-order accurate results were obtained using dynamic grid adaption with the meshes shown in Fig. 3.11.



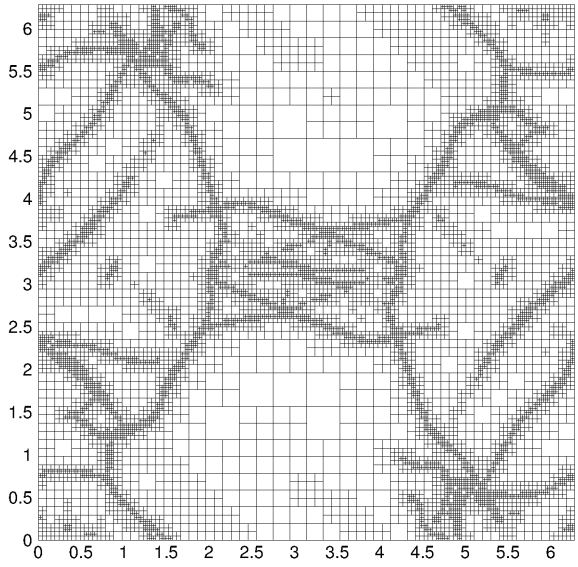
(a) AMR as applied to the Orszag-Tang vortex problem at $t = 0.5$. At this point, the mesh consists of 118 8-by-8 blocks, or 7,552 cells in total.



(b) AMR as applied to the Orszag-Tang vortex problem at $t = 1.0$. At this point, the mesh consists of 1,474 8-by-8 blocks, ' or 95,810 cells in total.



(c) AMR as applied to the Orszag-Tang vortex problem at $t = 2.0$. At this point, the mesh consists of 8,428 8-by-8 blocks, or 539,136 cells in total.



(d) AMR as applied to the Orszag-Tang vortex problem at $t = 3.0$. At this point, the mesh consists of 13,522 8-by-8 blocks, or 865,408 cells in total.

Figure 3.11: The evolution of the mesh for the simulation of Fig. 3.10 with adaptive refinement. Up to $t = 1.0$, the mesh is refined every 0.025 seconds, after which it is refined every 50 time steps. The lines in the figure represent the boundaries of the 8-by-8 Cartesian blocks.

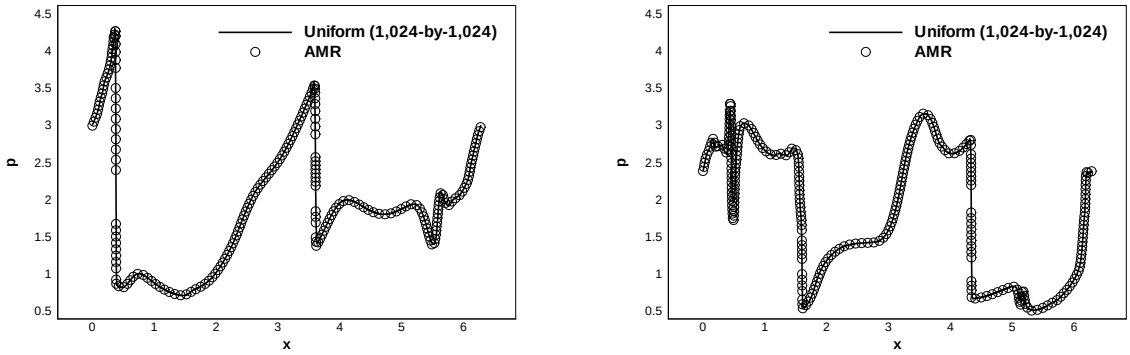


Figure 3.12: Pressure cuts at $y = 1.9635$ at two different times ($t = 2.0$ [left], and $t = 3.0$ [right]). High-order results obtained in combination with adaptive mesh refinement are compared with uniform high-order high-resolution results on a 1024-by-1024 mesh, and found to be similar. Our results agree with the results from [45] and [49].

Chapter 4

High-Order Finite-Volume Scheme for 3D Ideal MHD on Cartesian and Cubed-Sphere Grids

In this chapter, a high-order accurate finite-volume scheme is proposed for ideal magnetohydrodynamics equations in three dimensions. The development of high-order accurate and robust discretizations for hyperbolic conservation laws in 3D remains a challenging and active area of research. In recent years, significant advances have been made in proposing new discretization schemes of various types for 3D hyperbolic conservation laws, including finite difference methods (e.g., [39]), discontinuous Galerkin methods (e.g., [42, 77, 78]), finite-volume methods (e.g. [48, 79, 80, 81]), and combinations of these approaches (e.g., [82]). The 3D CENO method proposed in this chapter offers a different approach on general hexahedral grids that combines high-order accuracy with robustness at discontinuities, and it is attractive because it is inherently multi-dimensional by employing a K-exact overdetermined reconstruction scheme, and it avoids the complexity of considering multiple non-central stencil configurations that characterizes traditional ENO schemes. This approach is general and can be extended to unstructured grids [83]. For 3D MHD flows, the 2D fourth-order accurate CENO MHD solver from Chapter 3 is extended to 3D, using the generalized Lagrange multiplier (GLM) approach from [28] to control errors in the divergence of the magnetic field for the high-order accurate simulation (as also in [39]). Most MHD frameworks for parallel space physics simulations are only second-order accurate [84, 85, 86], and the parallel 3D fourth-order MHD method on general hexahedral grids proposed in this chapter significantly advances the state-of-the-art of large-scale simulation frameworks.

This chapter is structured as follows. We describe the three-dimensional implementation (which is a conceptually trivial extension of the two-dimensional implementation) in Sec. 4.1. The cubed-sphere mesh acts as a domain for a significant number of simulations

performed in this chapter. Due to the presence of degenerated edges on domains of this type, we present a rotation mechanism to generate stencils close to these edges in Sec. 4.1.5. Numerical results are presented in Sec. 4.2, which illustrate the effectiveness of the proposed high-order scheme to resolve smooth and non-smooth flows in three dimensions.

4.1 Fourth-Order CENO Method for 3D Hyperbolic Conservation Laws

The CENO method in 3D is an extension of the two-dimensional version, which is described in Sec. 3.1. It is important to note, however, that the choice of stencils will be much more varied in 3D than in 2D, and a brief discussion on stencil choice is presented in Sec. 4.1.2. Also specific to three-dimensional implementation, cell faces may not necessarily be planar, so a trilinear representation of a non-planar face is used, to map a reference unit cube onto a hexahedral cell (see Sec. 4.1.4). The nonplanar representation was developed and implemented by Dr. Ivan, but a summary of these developments is included in this thesis for completeness in describing the high-order simulation framework I co-developed and used in further work.

4.1.1 Finite-Volume Formulation

We consider nonlinear conservation laws of the form

$$\partial_t \mathbf{U} + \vec{\nabla} \cdot \vec{\mathbf{F}} = \mathbf{S} + \mathbf{Q}\,, \tag{4.1}$$

where $\mathbf{U}$ is the vector of conserved variables, $\vec{\mathbf{F}}$ is the flux dyad, and $\mathbf{S}$ and $\mathbf{Q}$ are numerical and physical source terms that may arise for certain equation sets and application problems. While our fourth-order method is developed for general conservation laws, we consider in this the particular cases of the MHD and Euler equations. For MHD (with GLM approach to control divergence errors as in [1, 28]) $\mathbf{U}$ is given by

$$\mathbf{U} = \left[\ \rho,\quad \rho\vec{V},\quad \vec{B},\quad \rho e,\quad \psi\ \right]^{\mathrm{T}}\,, \tag{4.2}$$

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, ρe is the total energy and ψ is the generalized Lagrange multiplier employed to control errors in the divergence of the magnetic field. Here, the total energy is given by $\rho e = p/(\gamma - 1) + \rho V^2/2 + B^2/2$, where V and B are the magnitudes of the velocity and magnetic field vectors, respectively, and γ is the ratio of specific heats. The flux dyad, $\vec{\mathbf{F}}$,

is given by

$$\vec{\mathbf{F}} = \begin{bmatrix} \rho\vec{V} \\ \rho\vec{V}\vec{V} + \left(p + \dfrac{\vec{B}\cdot\vec{B}}{2}\right)\mathbb{I} - \vec{B}\vec{B} \\ \vec{V}\vec{B} - \vec{B}\vec{V} + \psi\,\mathbb{I} \\ \left(\rho e + p + \dfrac{\vec{B}\cdot\vec{B}}{2}\right)\vec{V} - (\vec{V}\cdot\vec{B})\vec{B} \\ c_h^2\vec{B} \end{bmatrix}, \tag{4.3}$$

where $\mathbb{I}$ is the 3×3 identity matrix. The numerical source term employed to control magnetic field divergence (refer to Sec. 2.2.2), $\mathbf{S}$, is given by

$$\mathbf{S} = \begin{bmatrix} 0, & \vec{0}, & \vec{0}, & 0, & -\frac{c_h^2}{c_p^2}\psi \end{bmatrix}^{\mathrm{T}}. \tag{4.4}$$

The parameters c_h and c_p are chosen as in Chapter 3, following [28]. When $\vec{B}$ and ψ are set to zero, the equations reduce to the Euler equations of gas dynamics.

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

$$\frac{d\overline{\mathbf{U}}_{ijk}}{dt} = \frac{1}{V_{ijk}}\left[-\oiint_{\partial\mathcal{V}_{ijk}} \vec{\mathbf{F}}\cdot\vec{n}\,\mathrm{da} + \iiint_{\mathcal{V}_{ijk}} (\mathbf{S}+\mathbf{Q})\,\mathrm{dv}\right], \tag{4.5}$$

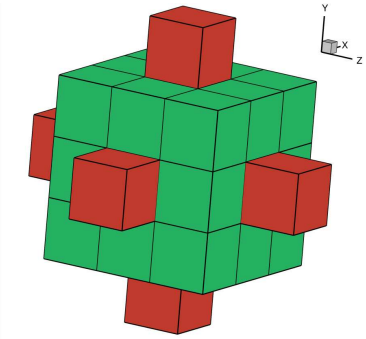
where V_{ijk} is the volume of cell $\mathcal{V}_{ijk}$, $\vec{n}$ is the unit outward normal of the cell surface $\partial\mathcal{V}_{ijk}$, and da and dv are surface and volume elements, respectively. Approximating the surface and volume integrals numerically we obtain

$$\frac{d\overline{\mathbf{U}}_{ijk}}{dt} = -\frac{1}{V_{ijk}}\sum_{f=1}^{6}\sum_{m=1}^{N_g}\left(\tilde{\omega}\vec{\mathbf{F}}_{\mathrm{num}}\cdot\vec{n}\right)_{i,j,k,f,m} + \overline{\mathbf{S}}_{ijk} + \overline{\mathbf{Q}}_{ijk} = \mathbf{R}_{ijk}(\overline{\mathbf{U}}), \tag{4.6}$$

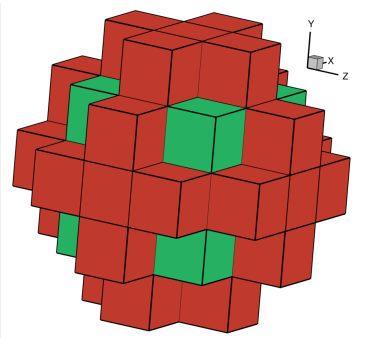
where f is used to indicate the six interfaces of the hexahedral cell, and m indicates the Gauss quadrature points over each interface. Here, $\vec{\mathbf{F}}_{\mathrm{num}}$ is the numerical flux function, $\overline{\mathbf{S}}_{ijk}$ and $\overline{\mathbf{Q}}_{ijk}$ are cell-averaged source terms (obtained using Gaussian quadrature), and $\mathbf{R}_{ijk}$ denotes the residual vector in cell ijk, which depends on the set of cell averages $\overline{\mathbf{U}}$.

4.1.2 K-Exact Reconstruction and Stencil Choice

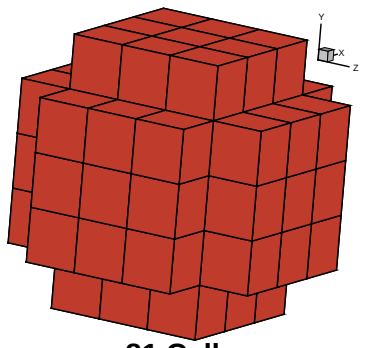
The 2D high-order CENO method of [1, 4, 27, 52, 87, 88] achieves high-order accuracy by employing K-exact polynomial reconstruction [51] of the solution field in each cell using



(a) 27 first-ring cells plus 6 extra second-ring cells for a total of 33 cells



(b) 27 first-ring cells plus 6 × 5 extra second-ring cells for a total of 57 cells



(c) 27 first-ring cells plus 6 × 9 extra second-ring cells for a total of 81 cells

Figure 4.1: Examples of several central reconstruction stencils for a cell ijk ranging in size from 33 to 81 cells that can be used to determine the 20 polynomial coefficients of a cubic reconstruction ($K = 3$). The first-degree neighbours of the cell ijk are shown in green whereas dark red is used for the second-degree neighbours.

the cell average and the cell averages of neighboring cells in a reconstruction stencil. In our 3D extension, we compute a K-exact polynomial reconstruction for each solution variable in cell ijk of the form

$$u_{ijk}^{K}(\vec{X}) = \sum_{\substack{p_1=0 \\ (p_1+p_2+p_3\leq K)}}^{K} \sum_{p_2=0}^{K} \sum_{p_3=0}^{K} (x-\bar{x}_{ijk})^{p_1} (y-\bar{y}_{ijk})^{p_2} (z-\bar{z}_{ijk})^{p_3} D_{p_1p_2p_3}^{K}, \tag{4.7}$$

where K is the degree of the polynomial, $\vec{X} = (x, y, z)$ is the coordinate vector, $(\bar{x}_{ijk}, \bar{y}_{ijk}, \bar{z}_{ijk})$ are the coordinates of the centroid of cell ijk, and the $D_{p_1p_2p_3}^{K}$ are the polynomial coefficients that are determined such that $u_{ijk}^{K}(\vec{X})$ matches cell averages in the reconstruction stencil with high accuracy. There are

$$\mathcal{N}_D = \frac{(K+1)(K+2)(K+3)}{6} \tag{4.8}$$

polynomial coefficients $D_{p_1p_2p_3}^{K}$, and they can be computed such that a function $u_{\text{exact}}(\vec{X})$ is represented with accuracy

$$u_{ijk}^{K}(\vec{X}) - u_{\text{exact}}(\vec{X}) = \mathcal{O}(\Delta x^{K+1}), \tag{4.9}$$

while a polynomial with degree K can be reconstructed exactly [51]. We use cubic reconstruction ($K = 3$) to obtain a fourth-order accurate numerical scheme. In this case,

$\mathcal{N}_D = 20$. We choose the coefficients such that the cell average of the reconstruction exactly equals the cell average of variable u in cell ijk,

$$\bar{u}_{ijk} = \frac{1}{V_{ijk}} \iiint_{\mathcal{V}_{ijk}} u^K_{ijk}(\vec{X})\, \mathrm{dv}, \tag{4.10}$$

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

$$\left(\frac{1}{V_{\gamma\delta\zeta}} \iiint_{\mathcal{V}_{\gamma\delta\zeta}} u^K_{ijk}(\vec{X})\, \mathrm{dv} \right) - \bar{u}_{\gamma\delta\zeta} = 0. \tag{4.11}$$

As in [1, 4, 27, 51, 52, 64] we use overdetermined reconstruction stencils, and impose Eqs. (4.11) in the least-squares sense, while Eq. (4.10) is imposed exactly. Note that the use of overdetermined stencils is especially attractive in the case of cubed-sphere grids (refer to Fig. 1.2), where the grid topology is unstructured near root-block edges and smaller numbers of neighbour cells are available than in regular Cartesian topology. Our approach naturally allows for reconstruction stencils of reduced size, without reducing the order of accuracy. The reconstruction stencils we consider in this paper are depicted in Fig. 4.1.

Equation (4.10) can be enforced analytically by replacing u^K_{ijk} with Eq. (4.7) and expressing the first coefficient, D^K_{000}, as a function of the other $M = \mathcal{N}_D - 1$ polynomial unknowns as

$$D^K_{000} = \bar{u}_{ijk} - \sum_{\substack{p_1=0 \\ (1 \le p_1+p_2+p_3 \le K)}}^{K} \sum_{p_2=0}^{K} \sum_{p_3=0}^{K} D^K_{p_1p_2p_3} \left(\overline{x^{p_1} y^{p_2} z^{p_3}}\right)_{ijk}, \tag{4.12}$$

where the geometric moment $\left(\overline{x^{p_1} y^{p_2} z^{p_3}}\right)_{ijk}$ of powers (p_1, p_2, p_3) is given by

$$\left(\overline{x^{p_1} y^{p_2} z^{p_3}}\right)_{ijk} = \frac{1}{V_{ijk}} \iiint_{\mathcal{V}_{ijk}} (x - \bar{x}_{ijk})^{p_1} (y - \bar{y}_{ijk})^{p_2} (z - \bar{z}_{ijk})^{p_3}\, \mathrm{dv}\,. \tag{4.13}$$

Substituting u^K_{ijk} from Eq. (4.7) in Eq. (4.11) and using Eq. (4.12) for D^K_{000} the following overdetermined linear system for the M unknowns is obtained:

$$\mathbb{L}\mathbf{D} - \mathbf{B} = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_J \\ \vdots \\ L_{N_n} \end{bmatrix}_{N_n \times M} \begin{pmatrix} D^K_{001} \\ D^K_{002} \\ \vdots \\ D^K_{p_1p_2p_3} \\ \vdots \\ D^K_{K00} \end{pmatrix}_{M \times 1} - \begin{pmatrix} w_1(\bar{u}_1 - \bar{u}_I) \\ w_2(\bar{u}_2 - \bar{u}_I) \\ \vdots \\ w_J(\bar{u}_J - \bar{u}_I) \\ \vdots \\ w_{N_n}(\bar{u}_{N_n} - \bar{u}_I) \end{pmatrix}_{N_n \times 1} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{N_n \times 1}, \tag{4.14}$$

where N_n is the number of neighbours in the reconstruction stencil. Here we have used the shorthand notation I to refer to cell ijk about which we do the reconstruction, and we have used indices $1, 2, \ldots, J, \ldots, N_n$ to refer to the other cells in the stencil of cell ijk. The w_J are geometric weights for each control volume J which serve the purpose of improving the locality of the reconstruction, becoming especially important for stretched meshes with surface curvature [62]. We use

$$w_J = \frac{1}{\left|\left|\Delta\vec{X}_{IJ}\right|\right|^{\theta}}\,, \tag{4.15}$$

with $\Delta\vec{X}_{IJ} = (\Delta x_{IJ}, \Delta y_{IJ}, \Delta z_{IJ}) = \vec{X}_J - \vec{X}_I$ and $\theta = 1, 2$. We use $\theta = 2$ except where noted otherwise. Row L_J of matrix $\mathbb{L}$ for a neighbouring cell J is given by

$$L_J = \left(w_J \left(\widehat{x^0 y^0 z^1}\right)_{IJ} \quad w_J \left(\widehat{x^0 y^0 z^2}\right)_{IJ} \quad \ldots \quad w_J \left(\widehat{x^{p_1} y^{p_2} z^{p_3}}\right)_{IJ} \quad \ldots \quad w_J \left(\widehat{x^K y^0 z^0}\right)_{IJ} \right)\,. \tag{4.16}$$

The matrix coefficients $\left(\widehat{x^{p_1} y^{p_2} z^{p_3}}\right)_{IJ}$ can be computed efficiently using only the $(\overline{x^{p_1} y^{p_2} z^{p_3}})$ moments:

$$\begin{aligned}
\left(\widehat{x^{p_1} y^{p_2} z^{p_3}}\right)_{IJ} &= \left(\frac{1}{V_J} \iiint\limits_{\mathcal{V}_J} (x - x_I)^{p_1} (y - y_I)^{p_2} (z - z_I)^{p_3} \, \mathrm{dv} \right) - (\overline{x^{p_1} y^{p_2} z^{p_3}})_I \\
&= \sum_{\ell=0}^{p_1} \sum_{q=0}^{p_2} \sum_{m=0}^{p_3} \left[C_{p_1}^{\ell} C_{p_2}^{q} C_{p_3}^{m} \Delta x_{IJ}^{\ell} \Delta y_{IJ}^{q} \Delta z_{IJ}^{m} \left(\overline{x^{\,(p_1-\ell)}\, y^{\,(p_2-q)}\, z^{\,(p_3-m)}} \right)_J \right] - (\overline{x^{p_1} y^{p_2} z^{p_3}})_I \,,
\end{aligned} \tag{4.17}$$

where the binomial coefficients C_α^β can be computed efficiently by recursion as

$$C_\alpha^\beta = \frac{\alpha - \beta + 1}{\beta} C_\alpha^{\beta-1}\,, \quad C_\alpha^0 = 1\,. \tag{4.18}$$

See [52] for the equivalent expressions in 2D.

The solution of the overdetermined linear system Eq. 4.14 can be obtained using QR factorization or by multiplication with the pseudo-inverse of $\mathbb{L}$ [52, 89]. In each time step, the constrained least-squares reconstruction problem is solved for each cell and for each primitive variable. Matrix $\mathbb{L}$ depends completely on geometry and is the same for all least-squares problems in a given cell ijk and for all time steps, so its inverse can be precomputed and reused to provide computational speedup (see [4, 27, 52] for details in the 2D case).

A specific technical difficulty in extending the 2D high-order CENO finite-volume method to 3D grids composed of general hexahedral cells with nonplanar surfaces, namely, the high-order accurate computation of the surface and volume integrals that were introduced in this and the previous subsection, is discussed in Section 4.1.4.

4.1.3 CENO Monotonicity Enforcement

In order to control spurious oscillations at discontinuities, we use the CENO monotonicity procedure that was introduced by Ivan and Groth [4, 27] for the 2D Euler equations, and has since been extended to the Navier-Stokes equations [52, 64, 87] and MHD in 2D (Chapter 3). The CENO procedure switches between an unlimited high-order accurate reconstruction (piecewise cubic in this paper, leading to a fourth-order accurate scheme) and a limited piecewise-linear reconstruction (second-order accurate), with the switching based on the smoothness indicator introduced in [4]. The smoothness indicator is computed in each cell for every reconstructed variable to determine whether the flow is locally smooth and well-resolved. For cells containing non-smooth or under-resolved solution content, the unlimited K-exact reconstruction is switched to limited piecewise linear reconstruction. It should be emphasized that local switching to lower order is only performed for those reconstructed variables that are deemed non-smooth or under-resolved. Note that the CENO scheme is called central because both the high-order and the low-order stencils are central with respect to the cell. The method is called an ENO method because it is monotone in the ENO sense, as defined in [53], which allows the presence of small spurious oscillations that have a magnitude on the order of the truncation error, but it does not allow $O(1)$ Gibbs-like oscillations at discontinuities. The fixed stencil used during the CENO reconstruction procedure avoids the complexity of considering multiple non-central stencil configurations that characterizes traditional ENO schemes.

Full details on the CENO smoothness indicator and switching mechanism are described in the 2D context in [4, 27, 52, 87, 88] and Chapter 3, and here we briefly present the generalization of the relevant formulas to 3D. For every cell ijk and primitive solution variable we compute a variable $\mathcal{S}$, the smoothness indicator, by

$$\mathcal{S} = \frac{\alpha\, c_s}{\max(1-\alpha, \epsilon)}, \tag{4.19}$$

where α is given by

$$\alpha = 1 - \frac{\sum_{\gamma}\sum_{\delta}\sum_{\zeta}\left(u^K_{\gamma\delta\zeta}(\vec{X}_{\gamma\delta\zeta}) - u^K_{ijk}(\vec{X}_{\gamma\delta\zeta})\right)^2}{\sum_{\gamma}\sum_{\delta}\sum_{\zeta}\left(u^K_{\gamma\delta\zeta}(\vec{X}_{\gamma\delta\zeta}) - \bar{u}_{ijk}\right)^2}. \tag{4.20}$$

and c_s is given by

$$c_s = \frac{\mathcal{N}_{\mathrm{SOS}} - \mathcal{N}_{\mathrm{D}}}{\mathcal{N}_{\mathrm{D}} - 1}. \tag{4.21}$$

Here, $\mathcal{N}_{SOS}$ indicates the size of the stencil used for reconstruction, $\mathcal{N}_D$ (number of degrees of freedom) denotes the number of unknown polynomial coefficients (e.g., 20 for $K = 3$), $\epsilon = 10^{-8}$ is introduced to avoid division by zero, and $\vec{X}_{\gamma\delta\zeta}$ is the centroid of cell $\gamma\delta\zeta$ in

the stencil of cell ijk. (The stencil used for computing the smoothness indicator can be chosen smaller than the reconstruction stencil. In particular, we compute the smoothness indicator using a stencil with 27 cells.)

The parameter α, with range $-\infty < \alpha \leq 1$, measures how accurately centroidal solution values of neighbouring cells can be reproduced using the reconstruction for cell ijk: in smooth flow α is close to one, while at cells with a discontinuity or an under-resolved feature, α tends away from one. The smoothness indicator $\mathcal{S}$ has range $-c_s < \mathcal{S} < c_s/\epsilon$: for smooth variation $\mathcal{S}$ is large, and for nonsmooth or under-resolved features $\mathcal{S}$ is small. We then choose a cutoff value, $\mathcal{S}_\mathcal{C}$, and when $\mathcal{S} > \mathcal{S}_\mathcal{C}$ the high-order reconstruction is used, while the limited low-order reconstruction is used otherwise. The form of the smoothness indicator is inspired by the definition of multiple-correlation coefficients and least-squares goodness-of-fit testing; see [52] for a more detailed discussion with further motivation for the approach. Thus, the robustness of the CENO scheme depends on carefully choosing a user-defined parameter, namely, the smoothness indicator, $\mathcal{S}_\mathcal{C}$. As such, it is not a parameter-free scheme as is often pursued in CFD, but it is a useful alternative to other high-order numerical techniques.

As discussed in [87], using standard Taylor approximation theory and assuming u is a continuous and differentiable function and the mesh is Cartesian with uniform spacing Δx, it is rather straightforward to show that

$$u^K_{\gamma\delta\zeta}(\vec{X}_{\gamma\delta\zeta}) - u^K_{ijk}(\vec{X}_{\gamma\delta\zeta}) \approx \mathcal{O}\left(\Delta x^{K+1}\right) . \tag{4.22}$$

Similarly, it can be shown that

$$u^K_{\gamma\delta\zeta}(\vec{X}_{\gamma\delta\zeta}) - \bar{u}_{ijk} \approx \mathcal{O}\left(\Delta x\right) . \tag{4.23}$$

This implies that

$$\alpha \approx 1 - \mathcal{O}\left(\Delta x^{2K}\right) , \tag{4.24}$$

for smooth resolved solution content. In this case, it is evident that $\alpha \to 1$ and $\mathcal{S} \to \infty$ as $\Delta x \to 0$ and this occurs at a rate that is significantly more rapid than the formal order of accuracy of the scheme. In this way, the asymptotic accuracy of the K-exact reconstruction is recovered as $\Delta x \to 0$. Conversely, for non-smooth solutions it is expected that

$$u^K_{\gamma\delta\zeta}(\vec{X}_{\gamma\delta\zeta}) - u^K_{ijk}(\vec{X}_{\gamma\delta\zeta}) \approx \mathcal{O}\left(1\right) , \qquad u^K_{\gamma\delta\zeta}(\vec{X}_{\gamma\delta\zeta}) - \bar{u}_{ijk} \approx \mathcal{O}\left(1\right) , \tag{4.25}$$

and therefore α will generally not be close to one.

As explained in [1], it is beneficial for stability to detect nearly-uniform regions by considering

$$\xi_{ijk} = \sqrt{\frac{1}{\mathcal{N}_\mathrm{D} - 1} \sum_{\substack{p_1=0 \\ (1\leq p_1+p_2+p_3\leq K)}}^{K} \sum_{p_2=0}^{K} \sum_{p_3=0}^{K} \left(D^K_{p_1p_2p_3}\right)^2 \xi_V(p_1,p_2,p_3)} , \tag{4.26a}$$

$$\xi_V(p_1,p_2,p_3) = \left(V_{ijk}^{\frac{2}{3}}\right)^{p_1+p_2+p_3}, \tag{4.26b}$$

which measures the variability of solution variable u in cell ijk. (It takes into account derivatives at the centroid of cell ijk.) When ξ_{ijk} is smaller than a threshold value (low variability), high-order reconstruction is always used, and only when ξ_{ijk} is greater than the threshold the smoothness indicator is computed and the CENO switching mechanism is activated. In particular, the smoothness indicator for the solution variable u is evaluated in cell ijk when

$$\xi_{ijk} > \epsilon_A + \epsilon_R \overline{u}_{ijk}, \tag{4.27}$$

where ϵ_A and ϵ_R represent absolute and relative variability thresholds, chosen to be 10^{-5} for the simulations performed in this paper, except where noted otherwise. Notice that the ξ_V term provides an appropriate length scaling for each derivative based on the cell volume, and it has only K distinct values due to the common variation range for p_1, p_2 and p_3. Moreover, using only the first order derivatives to measure the variability of solution variable u is less expensive and has been found to perform equally well in our numerical studies. This simplification is obtained in Eq. (4.26a) by taking $\mathcal{N}_{\mathrm{D}}=4$ and computing the summation terms with $K=1$. This simplification is used in our numerical tests.

4.1.4 High-Order Integration for Hexahedral Cells with Nonplanar Cell Faces

A specific technical difficulty in obtaining high-order accuracy on 3D cubed-sphere grids is to properly handle the nonplanar cell faces of the cubed-sphere grid cells. We use a trilinear representation for the nonplanar cell faces which can be exploited to obtain sufficiently accurate volume and surface integrals [90, 91, 92], as we now briefly explain.

Given a hexahedral cell in physical space with vertices defined by position vectors $\vec{X}_{000}$, $\vec{X}_{100}$, $\vec{X}_{010}$, $\vec{X}_{001}$, $\vec{X}_{110}$, $\vec{X}_{101}$, $\vec{X}_{011}$, $\vec{X}_{111}$, a trilinear mapping from a reference unit cube can be defined as

$$\vec{X}(p,q,s) = \vec{A} + \vec{B}p + \vec{C}q + \vec{D}s + \vec{E}pq + \vec{F}ps + \vec{G}qs + \vec{H}pqs\,, \tag{4.28}$$

where, p, q, s are the coordinates in the reference domain, and $\vec{A} = \vec{X}_{000}$, $\vec{B} = \vec{X}_{100} - \vec{A}$, $\vec{C} = \vec{X}_{010} - \vec{A}$, $\vec{D} = \vec{X}_{001} - \vec{A}$, $\vec{E} = \vec{X}_{110} - \vec{A} - \vec{B} - \vec{C}$, $\vec{F} = \vec{X}_{101} - \vec{A} - \vec{B} - \vec{D}$, $\vec{G} = \vec{X}_{011} - \vec{A} - \vec{C} - \vec{D}$, $\vec{H} = \vec{X}_{111} - \vec{A} - \vec{B} - \vec{C} - \vec{D} - \vec{E} - \vec{F} - \vec{G}$ (see, e.g., [90] for details). Note that the vertices of the hexahedral cell in physical space are obtained for $(p,q,s) \in \{0,1\} \times \{0,1\} \times \{0,1\}$. Thus, $\vec{X}_{000} = \vec{X}(0,0,0)$, $\vec{X}_{100} = \vec{X}(1,0,0)$, $\vec{X}_{010} = \vec{X}(0,1,0)$, $\vec{X}_{001} = \vec{X}(0,0,1)$, $\vec{X}_{110} = \vec{X}(1,1,0)$, $\vec{X}_{101} = \vec{X}(1,0,1)$, $\vec{X}_{011} = \vec{X}(0,1,1)$, $\vec{X}_{111} = \vec{X}(1,1,1)$, and the six trilinear surfaces are described by fixing one of p, q and s to

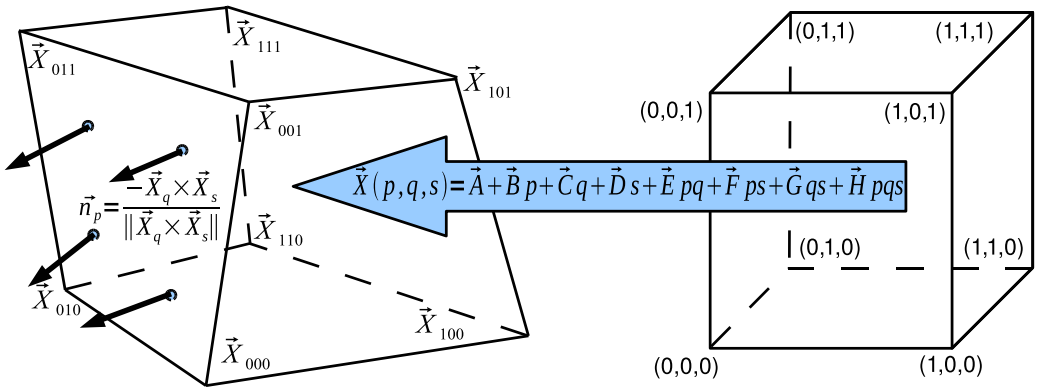


Figure 4.2: A general hexahedral cell in physical space having faces with nonplanar vertices (left) to which a reference unit cube (right) is mapped by defining a trilinear transformation $\vec{X}(p,q,s)$. The position vectors for the hexahedral cell vertices, $\vec{X}_{000}$, $\vec{X}_{100}$, $\vec{X}_{010}$, $\vec{X}_{001}$, $\vec{X}_{110}$, $\vec{X}_{101}$, $\vec{X}_{011}$ and $\vec{X}_{111}$, are mapped one-to-one to the vertices of the reference cube, where the Cartesian coordinates in the unit cube, (p,q,s), take binary values. Additionally, the unit normals at four points located on the hexahedral face defined by constant coordinate $p = 0$ are shown. Figure from [3].

0 or 1. The tangent vectors to the coordinate lines are defined by

$$\frac{\partial \vec{X}}{\partial p} \equiv \vec{X}_p(q,s) = \vec{B} + \vec{E}q + \vec{F}s + \vec{H}qs\,, \tag{4.29a}$$

$$\frac{\partial \vec{X}}{\partial q} \equiv \vec{X}_q(p,s) = \vec{C} + \vec{E}p + \vec{G}s + \vec{H}ps\,, \tag{4.29b}$$

$$\frac{\partial \vec{X}}{\partial s} \equiv \vec{X}_s(p,q) = \vec{D} + \vec{F}p + \vec{G}q + \vec{H}pq\,. \tag{4.29c}$$

The determinants of the Jacobians for volume and surface integration are given by

$$\det \mathbf{J}(p,q,s) \equiv \left| \frac{\partial(x,y,z)}{\partial(p,q,s)} \right| = \vec{X}_p \cdot (\vec{X}_q \times \vec{X}_s)\,, \tag{4.30a}$$

$$\det \mathbf{J}_p(q,s) \equiv \left|\left| \vec{X}_q \times \vec{X}_s \right|\right|\Big|_{p=\mathrm{ct}}\,, \tag{4.30b}$$

$$\det \mathbf{J}_q(p,s) \equiv \left|\left| \vec{X}_s \times \vec{X}_p \right|\right|\Big|_{q=\mathrm{ct}}\,, \tag{4.30c}$$

$$\det \mathbf{J}_s(p,q) \equiv \left|\left| \vec{X}_p \times \vec{X}_q \right|\right|\Big|_{s=\mathrm{ct}}\,, \tag{4.30d}$$

and the normal vectors to the coordinate planes by

$$\vec{n}_p(q,s) \equiv -\left.\frac{\vec{X}_q \times \vec{X}_s}{\det \mathbf{J}_p}\right|_{p=\mathrm{ct}}\,, \tag{4.31a}$$

$$\vec{n}_q(p,s) \equiv -\left.\frac{\vec{X}_s \times \vec{X}_p}{\det \mathbf{J}_q}\right|_{q=\mathrm{ct}}\,, \tag{4.31b}$$

$$\vec{n}_s(p,q) \equiv -\left.\frac{\vec{X}_p \times \vec{X}_q}{\det \mathbf{J}_s}\right|_{s=\mathrm{ct}}\,. \tag{4.31c}$$

4.1.4.1 Volumetric Integrals

The volume of hexahedral cell $I = ijk$ is defined by

$$V_I = \iiint_{\mathcal{V}_{ijk}} \mathrm{dv}, \tag{4.32}$$

where $\mathrm{dv} = \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$ is the volume element. The cell centroid is

$$\vec{X}_I = \frac{1}{V_I} \iiint_{\mathcal{V}_{ijk}} \vec{g}(\vec{X})\,\mathrm{dv}, \tag{4.33}$$

where the vector-valued function $\vec{g}(\vec{X}) = [x,y,z]^{\mathrm{T}}$, which is the position vector in the physical space, Thus, the more general procedure of evaluating a volumetric integral of a continuous smooth function, $g(\vec{X})$, over a control volume $\mathcal{V}_{ijk}$ is discussed here, which recovers the volume calculation for $g(\vec{X}) = 1$ and, for example, the x-coordinate of the cell centroid for $g(\vec{X}) = x$.

To evaluate the volumetric integral,

$$\mathcal{I} = \iiint_{\mathcal{V}_{ijk}} g(\vec{X})\,\mathrm{dv}, \tag{4.34}$$

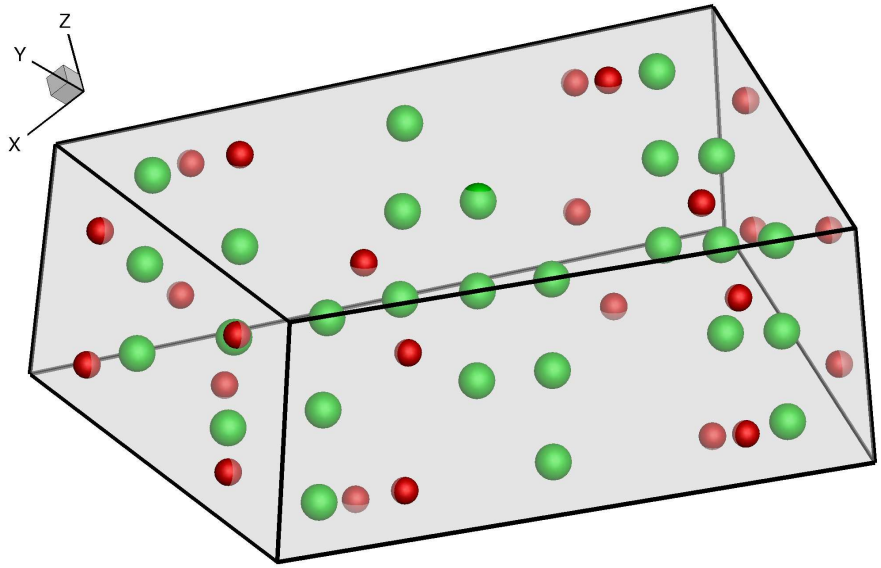


Figure 4.3: Volumetric (green) and surface (red) Gauss quadrature points used in the evaluation of volumetric and surface integrals over a hexahedral control volume to achieve fourth-order accuracy. Note that there are 27 volumetric points, and 4 surface points for each hexahedral face.

the variables and integration domain are changed to those of the reference unit cube by making use of the trilinear transformation, $\vec{X} = \vec{X}(p,q,s)$, and its transformation Jacobian determinant, $\det \mathbf{J}$, [90]. Thus, the volumetric integral, $\mathcal{I}$, is calculated in the canonical space (p,q,r) as

$$\mathcal{I} = \int_0^1 \int_0^1 \int_0^1 g(\vec{X}(p,q,s)) \det \mathbf{J} \, \mathrm{dp} \, \mathrm{dq} \, \mathrm{dr} \tag{4.35a}$$

$$\simeq \sum_{m=1}^{N_v} g\left(\vec{X}(p_m,q_m,r_m)\right) (\det \mathbf{J})_m \, \omega_m = \sum_{m=1}^{N_v} g(\vec{X}_m) \, \tilde{\omega}_m \,, \tag{4.35b}$$

where N_v is the number of volumetric Gauss points (see Fig. 4.3). Note that the Gaussian abscissas $\vec{X}_m$ and weights $\tilde{\omega}_m = (\det \mathbf{J})_m \, \omega_m$ are needed for initial computation of volumetric integrals of various quantities (see Table 4.1) but need not be stored during the actual simulation, unless required for integration of time-dependent volumetric source terms. Table 4.1 summarizes the different volumetric integrals that need to be computed in the 3D CENO scheme.

Symbol	Description	Definition	Section
V_I	volume of control volume I	Eq.(4.32)	4.1.4.1
$\vec{X}_I$	centroid of control volume I	Eq.(4.33)	4.1.4.1
$(\overline{x^{p_1} y^{p_2} z^{p_3}})_I$	geometric moment of powers (p_1, p_2, p_3) of control volume I about its own centroid	Eq.(4.13)	4.1.2
$\bar{u}_I$	mean value of u in control volume I	Eq.(4.10)	4.1.2
$(\overline{\mathbf{Q}})_I$	average source term in control volume I	Eq.(4.5)	4.1.1

Table 4.1: Summary of volumetric integrals used in the formulation of the CENO finite-volume scheme.

4.1.4.2 Surface Integrals

Surface integrals are evaluated in a similar fashion. For example, to evaluate a surface integral along a surface with constant p-coordinate, $\mathcal{A}_p$, the following expression is used:

$$\mathcal{I}_p = \iint_{\mathcal{A}_p} g(\vec{X})\,\mathrm{da} = \int_0^1\int_0^1 g(\vec{X}(p,q,s))\,\det\mathbf{J}_p\,\mathrm{d}q\,\mathrm{d}s \tag{4.36a}$$

$$\simeq \sum_{m=1}^{N_g} g\left(\vec{X}(p_m,q_m,r_m)\right)\,(\det\mathbf{J}_p)_m\,\omega_m = \sum_{m=1}^{N_g} g(\vec{X}_m)\,\tilde{\omega}_m\,, \tag{4.36b}$$

and the expressions for integrals along surfaces $\mathcal{A}_q$ and $\mathcal{A}_s$ follow by cyclic permutation. Here, N_g is the number of Gauss points on the surfaces (see Fig. 4.3). The $\vec{X}_m$ and $\tilde{\omega}_m = (\det\mathbf{J}_p)_m\,\omega_m$ are stored in our framework to increase the computational performance. Table 4.2 indicates that surface integrals are used to compute the fluxes through cell faces in the 3D CENO scheme.

Symbol	Description	Definition	Section
$\oiint_{\partial\mathcal{V}_I} \vec{\mathbf{F}}\cdot\vec{n}\,\mathrm{da}$	flux through the boundaries of control volume I	Eq.(4.5)	4.1.1

Table 4.2: Surface integrals used in the formulation of the CENO finite-volume scheme.

4.1.5 Rotation Mechanism for Generating Consistent Stencils Near Degenerate Block Edges

In a multiblock grid with regular Cartesian topology throughout the grid, four grid blocks are incident to any edge of a grid block. In the cubed-sphere grid of Fig. 1.2, however, the connectivity of the six root blocks is unstructured (i.e., the topology is non-Cartesian), and any grid block edge along one of the eight radial lines that go through the eight corners of the cubed-sphere grid have only three incident grid blocks (as depicted in a 2D projection in Fig. 4.4c). We call grid block edges along which the number of incident grid blocks is less than four *degenerate* edges. In our implementation, blocks along these edges have missing ghost cells in the opposite corner along the degenerate edge (see Figs. 4.4a and 4.4b) (we say that the ghost cells in the corner are *collapsed* or *inactive*, see also [8]). This means that the CENO reconstruction stencils we use (Fig. 4.1) need to be adjusted since the topology is different and not all cells used in the original stencils are available. In this section we propose a general rotation-based mechanism to automatically derive modified stencils near degenerate edges. The mechanism is sufficiently general that it can also be used to derive modified stencils for grid blocks that have three degenerate edges intersecting in a corner. This occurs, for example, when considering a cubed-sphere grid in which the interior of the inner sphere is filled by a seventh root block with Cartesian topology, see Fig. 4.5. This seventh block may, for example, represent the interior of a lunar or planetary object, in which a magnetic diffusion equation may be solved when modelling interaction with the solar wind [93]. The grid inside the interior sphere is generated as in [94], which will be briefly discussed in Appendix B. In this type of grid, grid blocks that contain one of the eight corners of the interior root block, have three degenerate edges emanating from that corner, see Figs. 4.5 and 4.6. (In fact, four degenerate block edges emanate from the corner (including one in the radial direction).) We will apply the proposed rotation mechanism about each of the three degenerate edges to obtain modified stencils in three-way degenerate blocks in a systematic way.

The rotation mechanism we propose functions as follows. We explain the mechanism for the case depicted in Fig. 4.4, where we want to use the 33-cell base reconstruction stencil of Fig. 4.1 for computing the reconstruction in a ghost cell of a grid block that lies next to the corner of the block. The grid block is one-way degenerate with the degenerate edge in the z direction (indicated by the dash-dotted line 'Edge of Collapsed Corner' in Fig. 4.4a). Since the corner ghost cells are missing (light brown/transparent cells), the stencil needs to be adjusted. Consider, for simplicity, the central xy plane of the stencil. By folding the two remaining layers of active ghost cells toward each other (as in Fig. 4.4c), it can be seen that the two green cells now become distance-two neighbors of the pink cell (where the reconstruction is computed), and can be included in the modified stencil (since distance-two cells are included in the original stencil). More generally, the non-Cartesian topology makes the collapsed (inactive) ghost cells unavailable to the stencil, but equivalent cells (in terms of distance) in the remaining ghost cell layers are available to complete the

stencil. A general mechanism that works for all stencils of Fig. 4.1 can be formulated in terms of a rotation mechanism: in Figs. 4.4b and 4.4d, we rotate the collapsed ghost cells that were part of the original stencil (light brown) by 90 degrees about the degenerate edge away from the reconstruction cell (pink). A rotated cell may land on a cell that does not form part of the existing stencil cells, in which case those new cells (the green cells in Figs. 4.4b and 4.4c) are added to the modified stencil. In case a rotated cell lands on an existing stencil cell (e.g., the brown cell next to the two green cells), no new cell is added to the stencil. The same mechanism is applied to the stencil planes above and below the central plane in Fig. 4.4, resulting in two more green cells added to the modified stencil. As a result, the 33-cell stencil, which was missing seven collapsed ghost cells, is modified by including four new rotated collapsed ghost cells, resulting in a final 30-cell stencil. This rotation mechanism provides modified stencils that are consistent with the original stencils from Fig. 4.1 in a double sense: first, the number of cells dropped in the degenerate stencil is minimized by incorporating new cells taking into account the new neighbor configuration and distances in the non-Cartesian topology; and second, degenerate stencils generated for ghost cells are identical to the degenerate stencils generated for the corresponding physical cells in neighbouring blocks (which is required for consistency of flux calculation, which we rely on for conservation in the implementation of our finite-volume methods).

In terms of implementation, we obtain the coordinates of rotated collapsed ghost cells by the following matrix rotation mechanism. For simplicity, we formulate the equations in the 2D setting corresponding to Fig. 4.4d. We choose the origin of the ij coordinate system for the block in the center of the lower left cell of Fig. 4.4d (which is located in the collapsed ghost cell region). Since there are four layers of ghost cells for the block depicted in the figure, the rotation axis (the degenerate block edge) is located at coordinate $\vec{I}_0 = (3 + 1/2, 3 + 1/2)$. Let $\vec{I}$ be the coordinate of a collapsed ghost cell that is rotated towards the active ghost cell layer about the degenerate block edge (e.g., $\vec{I} = (2, 3)$ for the brown cell in Fig. 4.4d), and let $\vec{I}_R$ be the coordinate of its image under the rotation ($\vec{I}_R = (4, 2)$ for the green cell). The cell coordinate after rotation, $\vec{I}_R$, is obtained by

$$\vec{I}_R = \vec{I}_0 + \mathbb{R}\left(\vec{I} - \vec{I}_0\right), \tag{4.37}$$

where the rotation matrix for the case of Fig. 4.4d is given by

$$\mathbb{R} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \tag{4.38}$$

Note that the 27-cell stencil we use for our second-order scheme is also reduced at degenerate edges, and we use the same rotation mechanism to determine the degenerate stencil. On the other hand, the 27-cell stencil cannot be used for cubic reconstruction (fourth-order scheme), since distance-two neighbours need to be added at least in the directions of the grid lines to have a well-posed reconstruction problem.

We emphasize the value of a systematic automatic procedure for determining these degenerate stencils. While in principle these special stencils could be hand-derived and hard-coded, this would be a formidable task and would be error-prone due to the multitude of cases that occur for the different stencils of Fig. 4.1 and for the many different geometrical configurations of degenerate blocks in 3D space and their ghost cells and physical cells located at different positions in the block. Hard-coding these special cases would also lead to unmaneagable additional complexity of the block-parallel 3D computer code. For this reason, our automatic procedure to select appropriate smaller stencils using a general rotation-based mechanism is an important ingredient of our approach. Furthermore, the mechanism is attractive because it is sufficiently flexible to handle three-way degenerate blocks that occur, for example, when a seventh root block is incorporated into the cubed-sphere grid to discretize the interior of the inner sphere of the grid.

4.1.6 Summary of Further Implementation Aspects

We use standard explicit second-order and fourth-order Runge-Kutta time integration methods for the second-order and fourth-order accurate spatial discretizations to be compared in our time-dependent numerical test problems (see [1, 52] and references therein). For steady-state simulations, we use a five-stage optimally smoothing method regardless of the solution accuracy. We use an MPI-parallel multiblock method with octree data structures as described in [8], and boundary conditions are imposed using ghost cells.

4.2 Numerical Results

In grid convergence studies based on an exact solution, the L_1, L_2, and L_∞ norms of the numerical solution error are computed as follows:

$$L_1 = |E|_1 = \frac{1}{V_T} \sum_{i,j,k} \iiint_{\mathcal{V}_{ijk}} \left| u^K_{ijk}(\vec{X}) - f(\vec{X}) \right| \, \mathrm{dv} \,, \tag{4.39}$$

$$L_2 = |E|_2 = \sqrt{\frac{1}{V_T} \sum_{i,j,k} \iiint_{\mathcal{V}_{ijk}} \left[u^K_{ijk}(\vec{X}) - f(\vec{X}) \right]^2 \, \mathrm{dv}} \,, \tag{4.40}$$

$$L_\infty = |E|_\infty = \max_{i,j,k} \left(\frac{1}{V_{ijk}} \iiint_{\mathcal{V}_{ijk}} \left| u^K_{ijk}(\vec{X}) - f(\vec{X}) \right| \, \mathrm{dv} \right) \,, \tag{4.41}$$

where V_T is the total volume of the computational domain, $f(\vec{X})$ is the exact solution evaluated at point $\vec{X}$, and the summation is taken over all the grid cells.

4.2.1 Three-Dimensional CENO Reconstruction

We describe two examples of smooth function reconstructions to demonstrate the accuracy of the CENO reconstruction that lies at the core of the proposed high-order finite-volume scheme. These reconstruction tests proceed by first computing highly accurate cell averages for a given function, then using these cell averages to compute high-order polynomial reconstructions in the cells, and finally computing the error between the original function and the polynomial reconstruction by high-accuracy numerical integration over each cell. The order of convergence of this error as grids are refined measures the order of accuracy of the CENO reconstruction, which determines the order of accuracy of the numerical simulation method. The discrete initial data in these cases is generated by accurately integrating the exact solution over cells to obtain the required cell averages using the integration procedure outlined in [3] in combination with an adaptive approach [52, 95], and this adaptive approach is also used to compute the error with high accuracy.

4.2.1.1 Reconstruction of a Smooth Function in a box with Distorted Grid

To assess the accuracy of the 3D high-order CENO procedure on meshes containing hexahedral cells with nonplanar faces, reconstructions of the smooth function $f(x, y, z) = (\cos(\pi(y + 1)) - \cos(\pi z))e^{-\pi(x+1)}$ are compared to the exact solution on distorted meshes for a rectangular box domain. The meshes are distorted in such a way that cell faces are nonplanar. The computational domain is the rectangular box defined by $0 < x < 1$ and $-1 < y, z < 1$. Fig. 4.7a illustrates how the interior nodal points have been perturbed randomly, and depicts the high-order solution reconstruction obtained on a structured 3D mesh with eight blocks of $4 \times 8 \times 8$ cells and 2,048 total computational cells.

Fig. 4.7b shows grid convergence reconstruction studies with the 4th-order CENO method on a series of meshes. The initial mesh has one block with $8 \times 16 \times 16$ cells and 2,048 total cells and the final mesh has 4,096 blocks and $\mathbf{N} = 8,388,608$ total computational cells. The expected theoretical asymptotic convergence rate of the 4th-order accurate method is achieved in all error norms. As the mesh is refined, the slopes of the L_1-, L_2-, and L_∞-error norms approach -4.087, -4.075 and -3.836, respectively, thereby providing validation for the proposed trilinear-based CENO reconstruction procedure to general hexahedral cells with nonplanar faces.

4.2.1.2 Reconstruction of a Spherically Symmetric Function on the Cubed Sphere

The distribution of errors on cubed-sphere grids is not expected to be uniform at any given radius, for several reasons: grid lines have kinks at the boundaries of the six cubed-sphere grid sectors, cells are smaller and more deformed near sector boundaries and corners,

and the grid topology is unstructured (with varying stencil size) at the sector corners. It is therefore of interest to investigate the error distribution in reconstructions obtained by the proposed fourth-order CENO procedure. In the previous two test problems we have established that fourth-order convergence is obtained overall (including near sector corners where CENO stencil sizes are reduced), but it is also desirable that the magnitude of the error for any grid resolution is not excessively larger near sector boundaries and corners than in other grid regions, which we investigate in this section. We do this for the function $f(R)=R^{-2.5}$, which is spherically symmetric, such that any non-uniformity in the reconstruction error can only be attributed to non-symmetric variations in the grid and to the differences in the selection of the reconstruction stencil at sector corners of the cubed-sphere grid. We reconstruct the function $f(R)=R^{-2.5}$ on a spherical shell with inner and outer radii $R_i=2$ and $R_o=3.5$. The cubed-sphere grid used consists of six 16×16×16 blocks and the error distribution is plotted in Fig. 4.8 on a sphere of radius $R=2.6$ in the interior of the domain to avoid effects from the boundary condition implementation. Note that local L_1 and L_2 errors are plotted according to Eqs. (4.39) and (4.40), where the integrals are taken over one cell and are normalized by the cell volume. We find a fairly symmetric error distribution, indicating that the least-squares based K-exact reconstruction succeeds in providing almost uniform accuracy, including in irregular cells near sector boundaries and corners.

It is interesting to note in Fig. 4.8a that, on the gnomonic cubed-sphere grid we use, the error is actually the smallest near the sector boundaries (away from the corners), and this is likely due to the smaller size of the cells there (and the slope discontinuities in the grid lines are small at those locations). The cells near the centre of the sectors are somewhat larger and the error is somewhat larger there. Not unexpectedly the error is largest near the sector corners, where grid lines have significant slope discontinuities and cells are deformed, and reconstruction stencils have reduced size. Nevertheless, the overall difference in error is within a factor of about two.

Comparing stencils with 33, 57 and 81 cells, it is interesting to note that the smaller stencil (33 cells) features a maximum L_1 error that is about 50% smaller than for the larger stencils. This may seem counter-intuitive at first, but can be explained by observing that we use each stencil with a fourth-order accurate reconstruction, so the accuracy of the numerical results depends on the accuracy of the polynomial coefficients that are obtained by the least-squares fitting process. Figure 4.8 indicates that polynomial reconstruction is more accurate for the smaller-size stencils, which can be expected since they use information that is more local to the cell for which the reconstruction is computed, whereas the larger stencils also use more distant information. We conclude that the size-33 stencil is the best choice when using fourth-order reconstruction, while larger stencils may be used when polynomials of degree higher than four are used for reconstruction (in which case 33 cells do not provide enough conditions to constrain the polynomial coefficients).

4.2.2 Three-Dimensional Flow Problems in a Rectangular Box

4.2.2.1 Magnetostatic MHD Problem

We consider the 3D magnetohydrostatic test case proposed by Warburton and Karniadakis [42]. In this problem the fluid is static (i.e., has zero velocity) and the magnetic field is irrotational. The domain of the problem is the rectangular box defined by $0 < x < 1$ and $-1 < y, z < 1$. The 3D analytical solution is given by

$$\begin{aligned} \rho &= 1\,, \\ \vec{V} &= \vec{0}\,, \\ B_x &= (\cos(\pi(y+1)) - \cos(\pi z))f(x)\,, \\ B_y &= \cos(\pi z)f(y) + \sin(\pi(y+1))f(x)\,, \\ B_z &= \sin(\pi z)(f(y) - f(x))\,, \\ p &= 5(\gamma - 1)\,, \\ \psi &= 0\,, \end{aligned} \tag{4.42}$$

where $f(u) = e^{-\pi(u+1)}$ and $\gamma = 5/3$. Following [42], this test has been performed as an initial value problem with the exact solution used to provide the initial condition and the values for the Dirichlet boundary conditions for all domain boundaries. Numerical simulations for this problem have been performed with the fourth-order CENO scheme on a series of Cartesian meshes ranging in size from $4\times8\times8 = 256$ to $128\times256\times256 = 8,388,608$ cells, until the solution reached the steady state. We use $\mathcal{S}_\mathcal{C} = 500$ for the smoothness indicator cut-off value, and use the local Lax-Friedrichs (LF) flux function.

Fig. 4.9a shows the magnitude of the magnetic field vector obtained using the fourth-order ($K=3$) CENO-GLM scheme on a Cartesian mesh with one $8\times16\times16$ block. The L_1, L_2, and L_∞ norms of the error in the x-component of the magnetic field, B_x, are given in Fig. 4.9b. Error measurements in the other components of the magnetic field, B_y and B_z, behave similarly. The results of the convergence study in Fig. 4.9b clearly show that the fourth-order theoretical accuracy is achieved by the numerical scheme in all error norms in the asymptotic limit, thereby demonstrating the high-order accuracy of the proposed CENO-GLM formulation for ideal MHD simulations.

4.2.2.2 3D Rotated MHD Shu-Osher Problem

To demonstrate the advantages of our fourth-order scheme in regions of rapid smooth variation near a discontinuity, we consider a 3D rotated version of the shock tube problem which was proposed by Shu and Osher [25], and extended to MHD for 2D in Chapter 3. As depicted in Fig. 4.10a, a sinusoidal density perturbation is added downstream of a purely advecting superfast shock wave. The interaction of the shock wave with the sinusoidal part

of the density field gives rise to fast oscillations and complex flow features downstream of the shock. The Shu-Osher shock tube problem provides an excellent testbed to highlight the benefits of the improved accuracy of high-order numerical schemes, while at the same time the presence of the shock puts the robustness and stability of the schemes to test.

The unperturbed initial conditions (in terms of vector field components along and perpendicular to the x_1 direction of the unrotated shock tube problem) are given in Chapter 3:

$$(\rho, v_\perp, v_\parallel, v_z, B_\perp, B_\parallel, B_z, p, \psi) = \begin{cases} (1,0,0,0,1,1,0,1,0) \text{ for } x_1 < 4, \\ (3.5, 5.8846, 1.1198, 0, 1, 3.6359, 0, 42.0267, 0) \text{ for } x_1 > 4, \end{cases} \tag{4.43}$$

and a sinusoidal density perturbation is added to the downstream part of the density field:

$$\rho_l = 1 + 0.2\sin(5x_1), \;\; \rho_r = 3.5. \tag{4.44}$$

The one-dimensional shock tube problem (along the x_1-axis) is rotated to the $(1,1,1)$ diagonal direction in the x, y, z coordinate system, resulting in angles of φ =45° in the xy plane and β =35.2644° in the xz plane. The comparison of the density profiles between the fourth-order and second-order methods is shown in Fig. 4.10b. We use $\mathcal{S}_\mathcal{C} = 180$ for the smoothness indicator cut-off value, use the local Lax-Friedrichs (LF) flux function, and choose $\gamma = 5/3$ and $\epsilon_A = \epsilon_R = 2\ 10^{-5}$. It is clear that, for the same number of points, the fourth-order CENO method captures the small-scale flow features much better than the second-order method. At the same time, the high-order CENO approach is stable at the shock and does not produce spurious oscillations.

4.2.3 MHD Iso-Density Vortex on a 7-Block Cubed Sphere

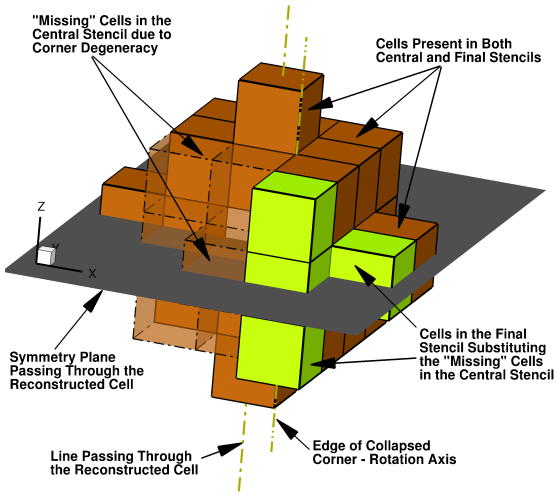
As our final test problem, we discuss the 3D iso-density MHD vortex advection problem from [39]. This is a smooth time-dependent 3D test problem with an exact solution, consisting of a magnetized vortex structure in force equilibrium that is advected by a uniform flow field. This test was performed by Dr. Ivan, but it is included in this thesis because it demonstrates high-order accuracy with the rotation mechanism at degenerate block boundaries that I co-developed.

The stationary 3D iso-density MHD vortex flow from [39] is given by

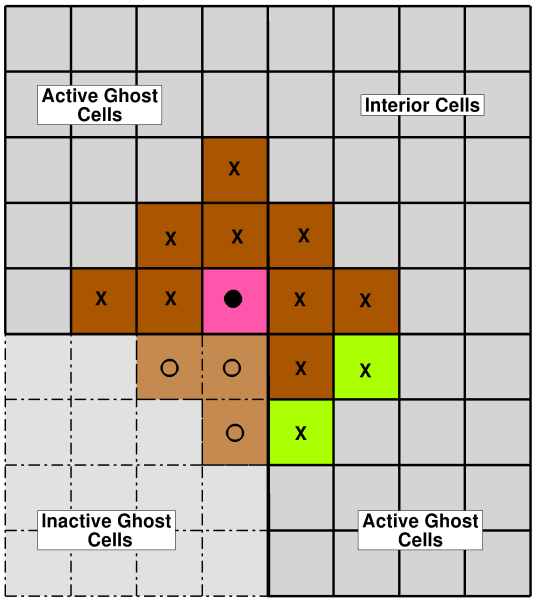
$$\begin{aligned} \rho &= 1, \\ \vec{V} &= (-y, x, 0)\ \kappa \exp\left(q(1-R^2)\right), \\ \vec{B} &= (-y, x, 0)\ \mu \exp\left(q(1-R^2)\right), \\ p &= 1 + \frac{1}{4q}\ \left(\mu^2\left(1 - 2q(R^2 - z^2)\right) - \kappa^2\rho\right) \exp\left(q(1-R^2)\right), \end{aligned}$$

where $\mu = \kappa = 1/(2\pi)$, and $q = 1$. This solution is translated with background velocity $(1, 1, 2)$.

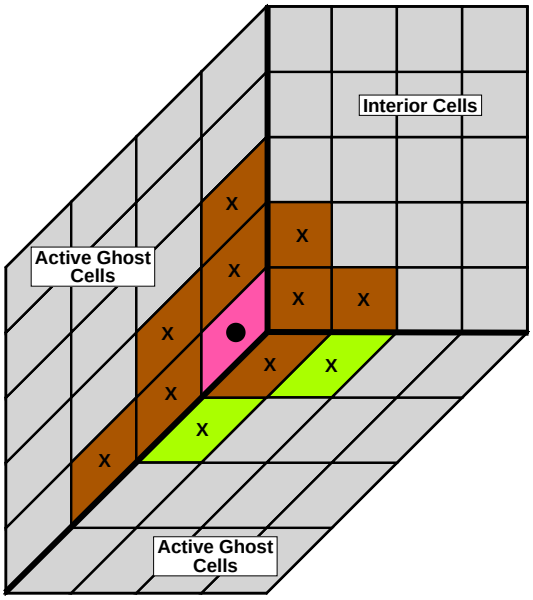
Simulations are performed on the 7-block cubed-sphere grid, with an inner radius, $R_i = 1$, and an outer radius, $R_o = 9$. In this problem the 3D vortex is initially located outside the inner sphere (centred at (-2, -2.5, -3)), and is then translated with velocity (1,1,2) over the cubed-sphere grid, passing through the inner sphere. The simulation is performed until $t = 3$, when the centre of the vortex is located at (1, 0.5, 3). Figure 4.11a shows the flow solution on the cubed-sphere grid at $t = 1.57812$, when the vortex passes through a corner of the seventh root block. The initial grid consists of 7 blocks with $16 \times 16 \times 16$ cells, corresponding to a total of 28,672 cells The final grid uses 3,584 blocks of $32 \times 32 \times 32$ cells corresponding to a total of 117,440,512 cells on 1,792 CPU cores. Figure 4.11b demonstrates that this simulation achieves fourth-order accuracy for a time-dependent flow on the cubed-sphere grid, and as such confirms the validity of the adopted approaches for achieving high-order accuracy on grids with nonplanar cell surfaces and using a rotation mechanism to determine degenerate stencils at block boundaries and corners with unstructured grid topology [3].



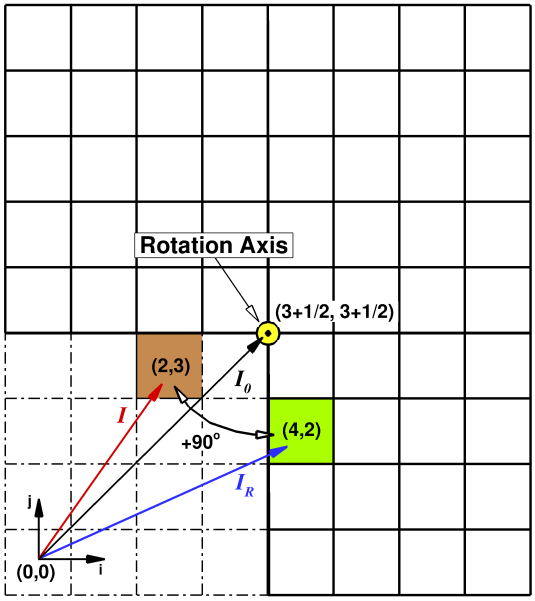
(a) 3D diagram of the stencil.



(b) 2D diagram showing the stencil in the symmetry plane and active and inactive ghost cells.

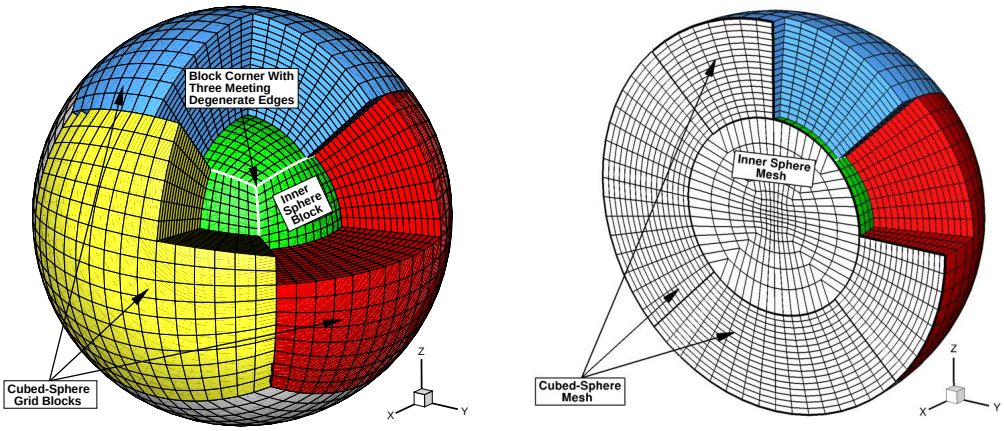


(c) Compact representation of the stencil in the symmetry plane.



(d) Illustration of the index vectors used in Eq. 4.37 to map cell (2,3) into (4,2).

Figure 4.4: Example of a degenerate stencil with 33 cells for a reconstructed ghost cell touching a degenerate edge. The cells of the original 33-cell central stencil remaining active in the final stencil are coloured dark brown. The cells coloured in green are replacing the cells identified as "missing" in the original stencil, which are coloured light brown/transparent. The final stencil has 30 cells, with three cells being "lost" due to the degeneracy of the grid block edge and the collapsing of the ghost cells in the corner opposite to the interior cells of the grid block.



(a) 3D Representation. (b) Cross Section.

Figure 4.5: Nested spherical grid comprising one spherical root block filling the interior of a cubed-sphere grid with six root blocks.

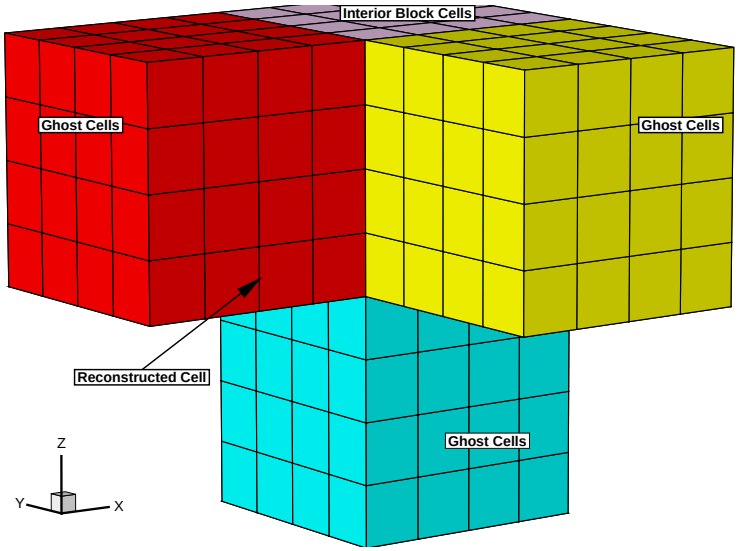
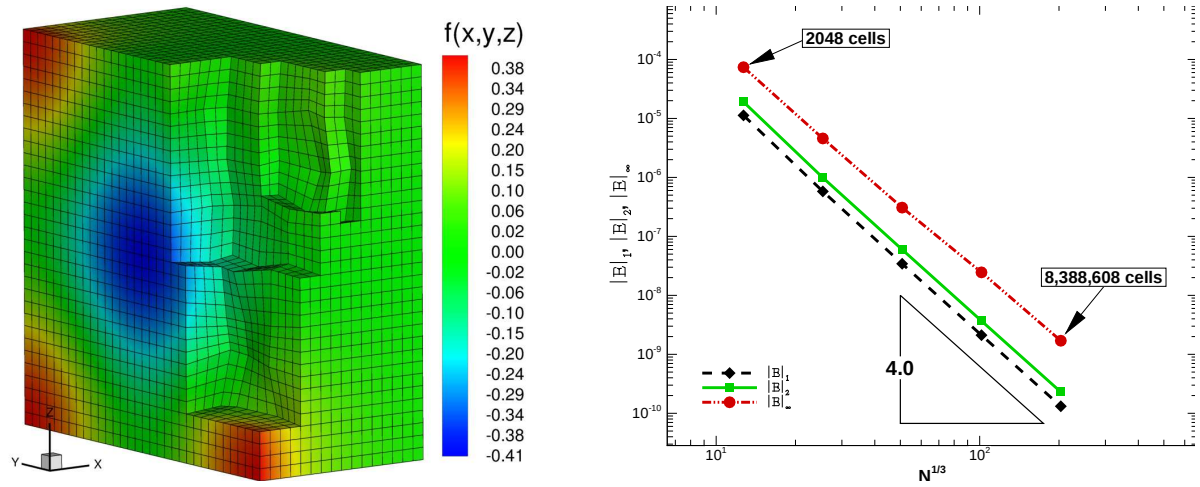
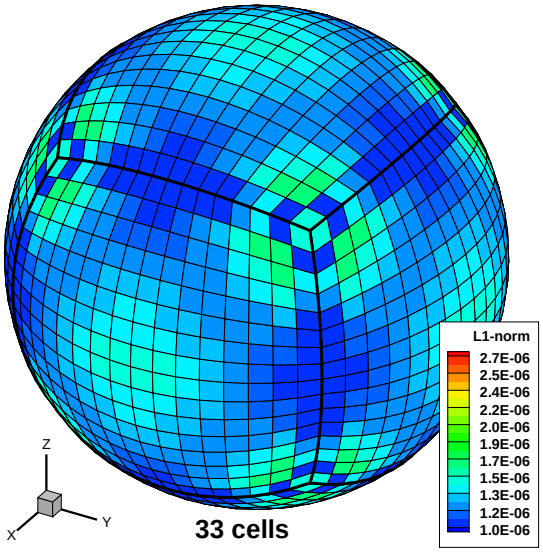


Figure 4.6: Grid block with three degenerate edges. The green dots indicate the degenerate stencil obtained using the proposed rotation mechanism for a base stencil with 81 cells centred about the reconstruction cell (purple). All ghost cells lying in the non-coloured blocks are collapsed. The resulting degenerate stencil has 66 cells.

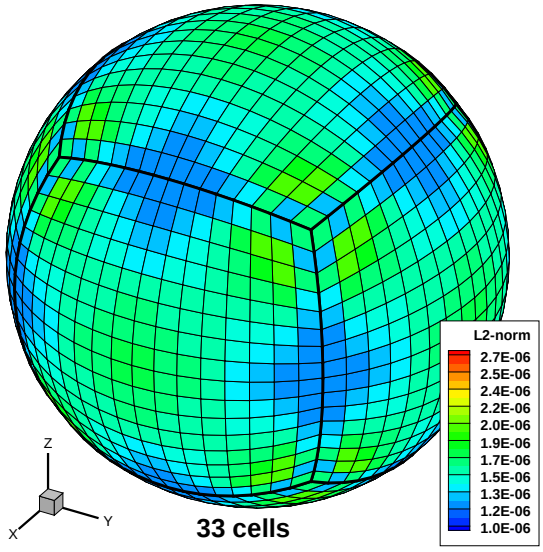


(a) (b)

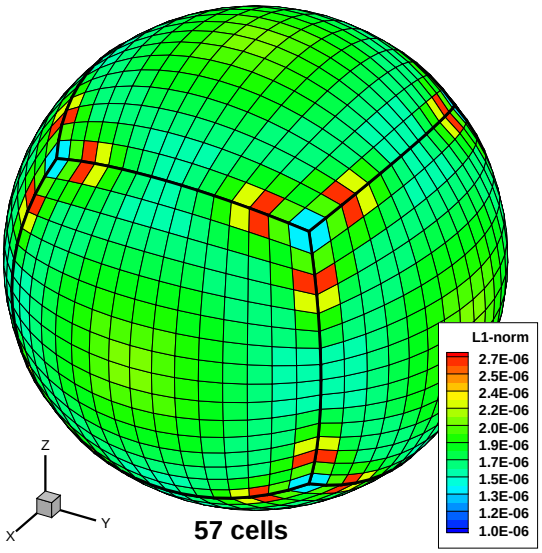
Figure 4.7: Solution reconstruction obtained using the 4th-order CENO scheme on a mesh with 8 blocks of $4\times8\times8$ and 2,048 cells (left) and error norms (right).



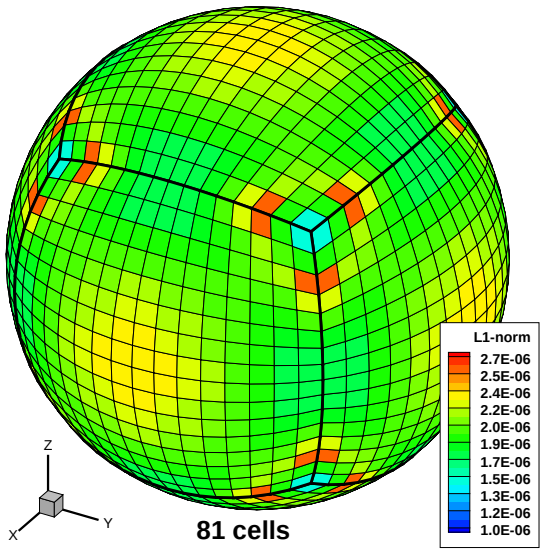
(a) L_1 error norm for the stencil with 33 cells.



(b) L_2 error norm for the stencil with 33 cells.

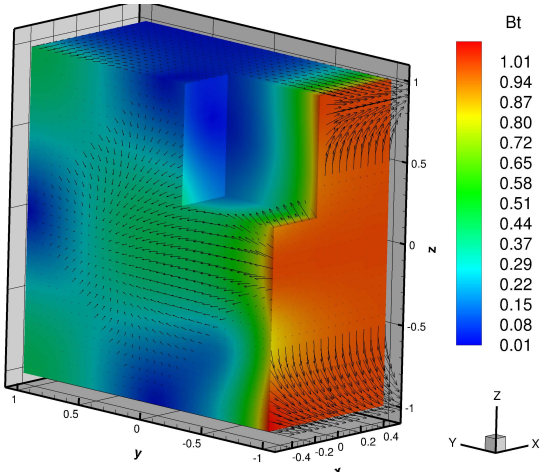


(c) L_1 error norm for the stencil with 57 cells.

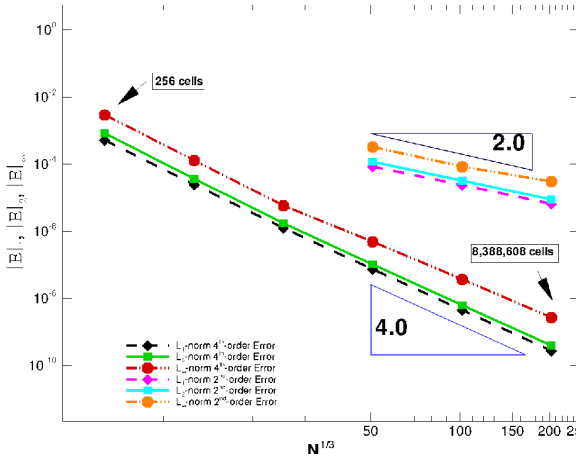


(d) L_1 error norm for the stencil with 81 cells.

Figure 4.8: Error norms of cubic reconstruction at $R = 2.6$ for $f(R) = R^{-2.5}$.

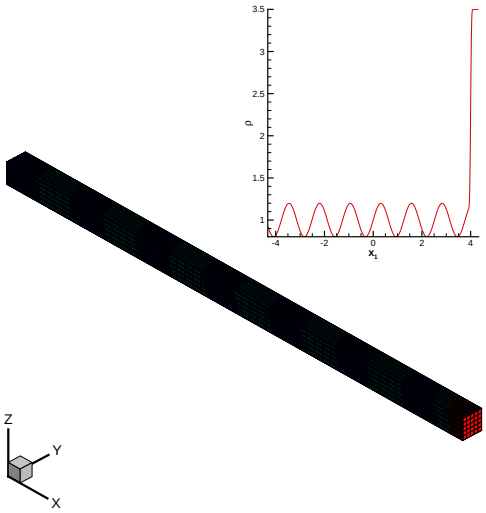


(a) Depiction of the magnetic vector field and its magnitude. The top-front corner has been removed for allowing the visualization of the interior solution.

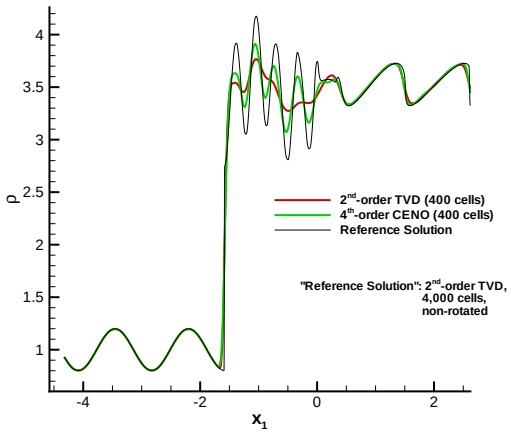


(b) Convergence of L_1-, L_2-, and L_∞-error norms.

Figure 4.9: (a) Fourth-order ($K=3$) CENO-GLM results for the total magnetic field, Bt, for the magnetohydrostatic test case on a Cartesian grid with 2,048 cells; and (b) L_1, L_2, and L_∞ norms of the error in the x-component of the magnetic field, B_x, as a function of mesh density for the 4th-order CENO-GLM scheme.

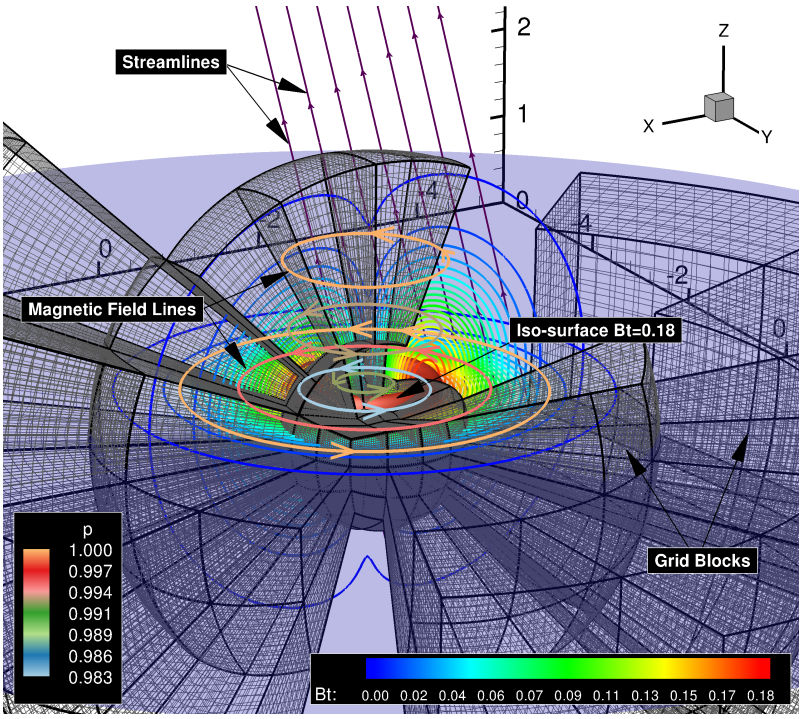


(a) Initial density profile as a function of the rotated coordinate x_1 and as a three-dimensional representation in the rotated frame

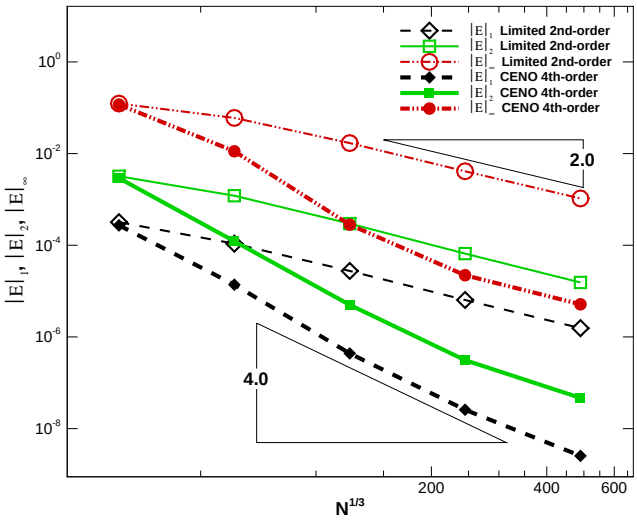


(b) Final density along the x_1 axis.

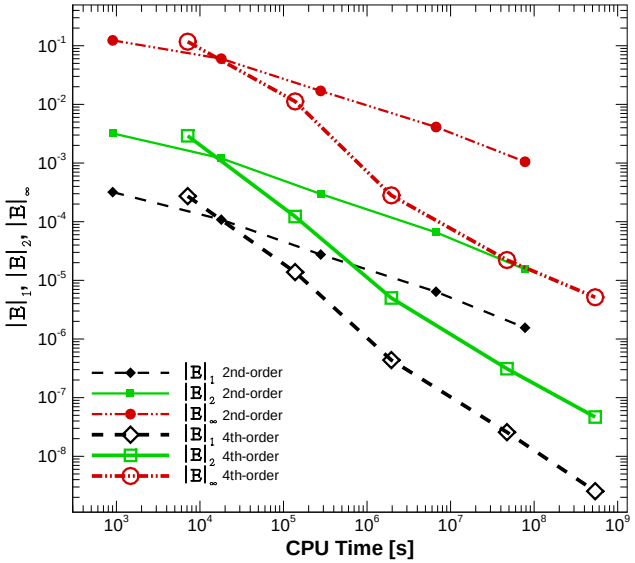
Figure 4.10: (a) Initial density profile in the MHD Shu-Osher problem, rotated with $\psi = 45°$ in the xy plane and $\beta = 35.2644°$ in the xz plane; and (b) comparison of the density solution at t = 0.6906. As can be observed from the figure, the fourth-order method produces results that are much closer to the non-rotated reference result in the highly oscillatory region, illustrating the benefits of high-order accuracy.



(a) Flow solution on cubed-sphere grid (selected grid blocks shown) at simulation time t=1.57812. (Total magnetic field contours, streamlines, and magnetic field lines coloured by pressure values.)



(b) Error norms versus grid size.



(c) Error norms versus total compute time.

Figure 4.11: Iso-density vortex problem solved on a 7-block cubed-sphere grid.

Chapter 5

High-Order CENO Finite-Volume Scheme for Resistive MHD

In this chapter, a high-order accurate finite-volume CENO scheme is proposed for the resistive MHD equations. The presence of second-order derivative terms creates an extra numerical challenge to solve the resistive MHD system. To handle these second-order terms, similar calculation mechanisms as those that were previously applied to the elliptic fluxes of the Navier-Stokes equations are adopted (see Sec. 3.5 in [27]), as presented in Sec. 5.1.

5.1 Numerical Treatment of Second-Derivative Flux Terms

As discussed in Sec. 2.1.2.1 and Sec. 2.1.2.2, the resistive MHD system requires also numerical values of fluxes that are elliptical in nature. While hyperbolic fluxes depend only on flow variables, $\vec{\mathbf{U}}$, the elliptic fluxes depend on their gradients, $\nabla\vec{\mathbf{U}}$, as well. In particular, the induction and the energy equations contain terms that include the second derivatives of the magnetic field.

The resistive MHD equations, as given by Eq. 2.15 - Eq. 2.18, can be written in a general conservation form with hyperbolic and elliptic fluxes as

$$\frac{\partial\vec{\mathbf{U}}}{\partial t}+\nabla\cdot\vec{\mathbf{F}}=\frac{\partial\vec{\mathbf{U}}}{\partial t}+\nabla\cdot\vec{\mathbf{F}}_H(\vec{\mathbf{U}})+\nabla\cdot\vec{\mathbf{F}}_E(\vec{\mathbf{U}},\nabla\vec{\mathbf{U}})=\vec{\mathbf{S}}. \tag{5.1}$$

Through applications of the divergence theorem, we can reformulate the above equation into a flux integral form (see Eq. 4.5). This necessitates finding the values of flow variables and their gradients at each Gauss quadrature point (refer to Fig. 4.3 for approximate

locations of these Gauss quadrature points). A k-exact piecewise polynomial reconstruction [51] is used to interpolate the values of the flow variables and their gradients at every Gauss quadrature point, the formulae of which are given as

$$u^k_{i,j,k}(\vec{r}) = \sum_{\substack{p_1=0 \\ (p_1+p_2+p_3\leq k)}}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} \left(x-\bar{x}_{i,j}\right)^{p_1} \left(y-\bar{y}_{i,j}\right)^{p_2} \left(z-\bar{z}_{i,j}\right)^{p_3} D^k_{p_1p_2p_3} \tag{5.2}$$

for polynomial reconstruction of any of the flow variables. Eq. 5.2 provides a $(k+1)^{th}$-order approximation to the solution, u. Likewise, the derivatives of a flow variable, u, can be approximated to the k^{th} order by differentiating Eq. 5.2, which is just a polynomial function. In the x-, y-, and z-directions, we can express the derivatives as follows

$$\left.\frac{\partial u^k}{\partial x}\right|_{\vec{r}} = \sum_{\substack{p_1=0 \\ (p_1+p_2+p_3\leq k)}}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} p_1 \left(x-\bar{x}_{i,j}\right)^{p_1-1} \left(y-\bar{y}_{i,j}\right)^{p_2} \left(z-\bar{z}_{i,j}\right)^{p_3} D^k_{p_1p_2p_3}, \tag{5.3}$$

$$\left.\frac{\partial u^k}{\partial y}\right|_{\vec{r}} = \sum_{\substack{p_1=0 \\ (p_1+p_2+p_3\leq k)}}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} p_2 \left(x-\bar{x}_{i,j}\right)^{p_1} \left(y-\bar{y}_{i,j}\right)^{p_2-1} \left(z-\bar{z}_{i,j}\right)^{p_3} D^k_{p_1p_2p_3}, \tag{5.4}$$

$$\left.\frac{\partial u^k}{\partial z}\right|_{\vec{r}} = \sum_{\substack{p_1=0 \\ (p_1+p_2+p_3\leq k)}}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} p_3 \left(x-\bar{x}_{i,j}\right)^{p_1} \left(y-\bar{y}_{i,j}\right)^{p_2} \left(z-\bar{z}_{i,j}\right)^{p_3-1} D^k_{p_1p_2p_3}. \tag{5.5}$$

The reconstructed gradient can then be calculated as a vector addition of the above derivatives, with each being a vector component of the gradient. So to compute a k^{th}-order accurate reconstruction of the gradient of u at the Gauss quadrature point, $\vec{r}_{GQP}$, we apply the following

$$\nabla u^k(\vec{r}_{GQP}) = \left.\frac{\partial u^k}{\partial x}\right|_{\vec{r}_{GQP}} \hat{i} + \left.\frac{\partial u^k}{\partial y}\right|_{\vec{r}_{GQP}} \hat{j} + \left.\frac{\partial u^k}{\partial z}\right|_{\vec{r}_{GQP}} \hat{k}. \tag{5.6}$$

It is reemphasized that in order to achieve a k^{th}-degree accuracy, the gradient will need to be reconstructed using k^{th}-order piecewise polynomial, which is $(k+1)^{th}$-order accurate for reconstruction of the flow variable.

Similar to Navier-Stokes, resistive MHD fluxes are composed of both hyperbolic and elliptic fluxes

$$\vec{\mathbf{F}} \cdot \vec{n} = \vec{\mathbf{F}}_H \cdot \vec{n} + \vec{\mathbf{F}}_E \cdot \vec{n}. \tag{5.7}$$

Treating hyperbolic and elliptic fluxes within the context of the CENO scheme has been discussed in detail in [27]. We will only briefly discuss elliptic flux evaluation in this section, and interested readers are referred to Sec. 3.4 and 3.5 of [27] for more detailed explanations on CENO (hyperbolic and elliptic) flux evaluations.

For elliptic flux evaluation, arithmetic means of the left and the right states are taken for both the flow variables, $\vec{\mathbf{U}}$, and their gradients, $\nabla\vec{\mathbf{U}}$,

$$\vec{\mathbf{F}}_E \cdot \vec{n} = \vec{\mathbf{F}}_E(\mathbf{U}, \nabla\mathbf{U}) \cdot \vec{n}, \tag{5.8}$$

$$\mathbf{U} = \frac{1}{2}[\mathbf{U}_l^k + \mathbf{U}_r^k], \tag{5.9}$$

$$\nabla\mathbf{U} = \frac{1}{2}[\nabla\mathbf{U}_l^k + \nabla\mathbf{U}_r^k]. \tag{5.10}$$

The left and the right states are calculated using the formulae given by Eq. 5.2 to Eq. 5.5 in the Gauss quadrature points at cell faces (two reconstructed values [left and right] are possible at every Gauss quadrature point on a cell face since each cell face is shared by two adjacent cells).

The evaluation of elliptic fluxes in terms of simple arithmetic averages has been implemented successfully within the context of CENO [27], and it is certainly more simple and straightforward, and holds computational advantage over more complicated elliptic flux evaluation techniques such as the generalized Riemann problem formulation for diffusion problems [96, 97], or the elliptic flux evaluation as found in the direct discontinuous Galerkin method for diffusion problems [98].

5.2 Convergence Studies

To demonstrate the capability of the proposed high-order method to resolve resistive MHD flows, we consider two test cases in 3D, for which analytical solutions are known. Convergence studies are performed by comparing the computed solutions and the analytical solutions using the formulae given by Eq. 4.39 - Eq. 4.41 to calculate the error norms. Similar as in Chapter 4, unless specified otherwise, the values of inviscid fluxes are calculated using the local Lax-Friedrichs function. We use the standard second-order and fourth-order Runge-Kutta time integration techniques.

5.2.1 One-Dimensional Kinematics Diffusion Equation

A one-dimensional resistive MHD test case is considered. Rather than the whole resistive MHD system, we consider only the induction equation. To further restrict the scope of the simulation, velocity is set to zero, and the solution is set to vary only in the x-direction, with the magnetic field having a component only in the y-direction. Ignoring the back reaction of the Lorentz force on the fluid (which would change the velocity), the induction equation reduces to

$$\frac{\partial B_y}{\partial t} - \eta\frac{\partial^2 B_y}{\partial x^2} = 0. \tag{5.11}$$

This practice of ignoring the fluid and field interaction to focus only on the induction equation is called 'kinematics of MHD' [5].

Supplying a sinusoidal initial condition for B_y (as proposed by problem 4.6 of [99]),

$$B_y(x, t = 0) = B_0 + \sin(\frac{\pi x}{L}) + \sin(\frac{2\pi x}{L}), \tag{5.12}$$

and setting $B_y(0, t)$ and $B_y(L, t)$ to B_0 as boundary conditions, the time-dependent exact solution for B_y can be obtained, which is given by

$$B_y(x, t) = B_0 + \sin(\frac{\pi x}{L})e^{-\eta(\frac{\pi}{L})^2 t} + \sin(\frac{2\pi x}{L})e^{-\eta(\frac{2\pi}{L})^2 t}. \tag{5.13}$$

The simulations were run until $t = 0.01$, with B_0, η, and L set to 1. Initial, and final, solutions are illustrated in Fig. 5.1. Even though this is a one-dimensional problem, the simulation is run using the 3D code, with only the number of cells in the x-direction varied for the purpose of convergence studies. The error norms of the exact solution are computed at the end of the simulation (at $t = 0.01$), using the same formulae as given by Eq. 4.39 to Eq. 4.41. We consider resolutions of 40 cells to 640 cells. Fig. 5.2 illustrates how the error norms converge to zero with the expected order of accuracy, with a difference of as much as five orders of magnitude between second- and fourth-order schemes for the same grid resolution.

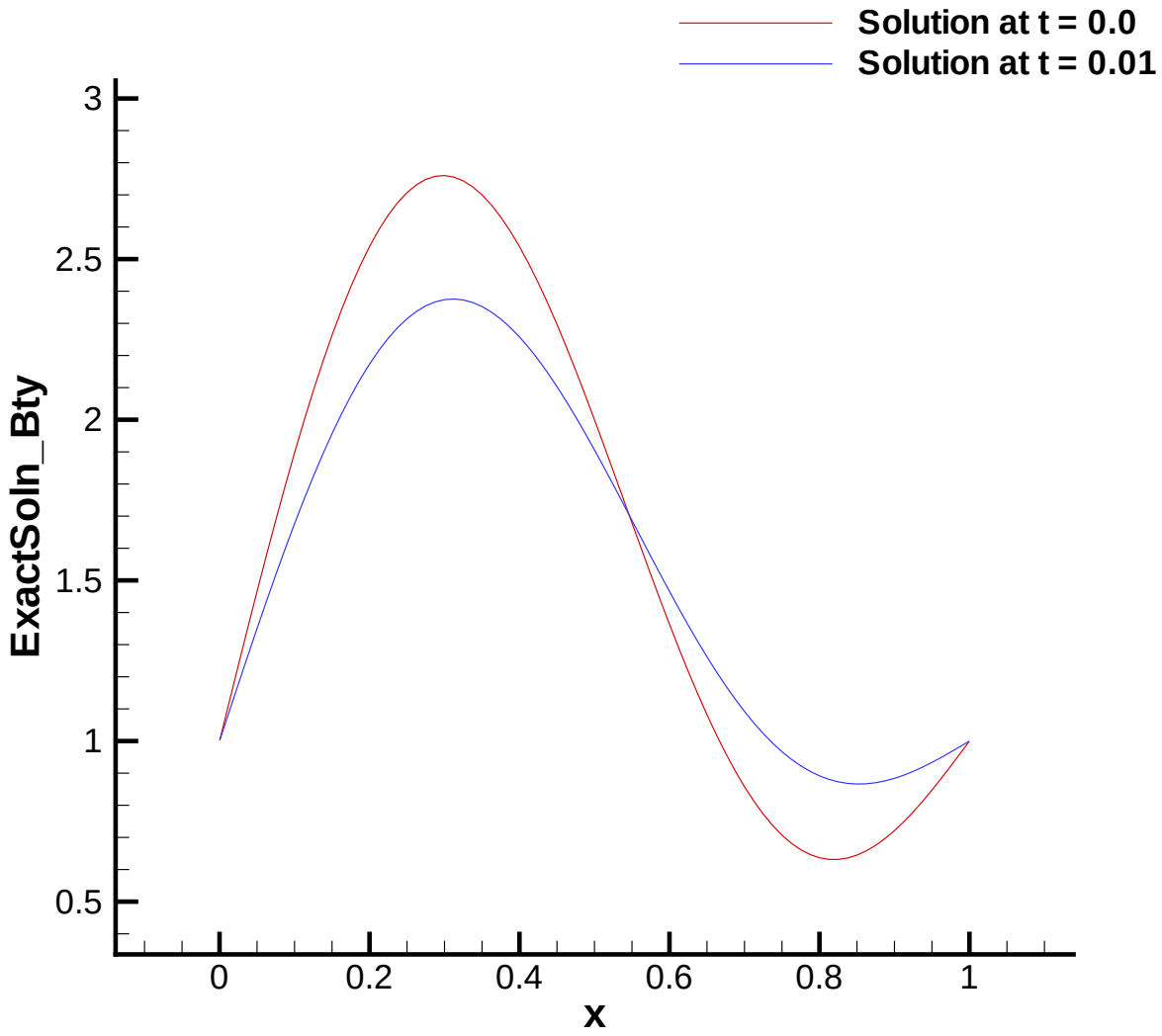


Figure 5.1: Shown here is the comparison of the initial condition and the exact solution at $t = 0.01s$

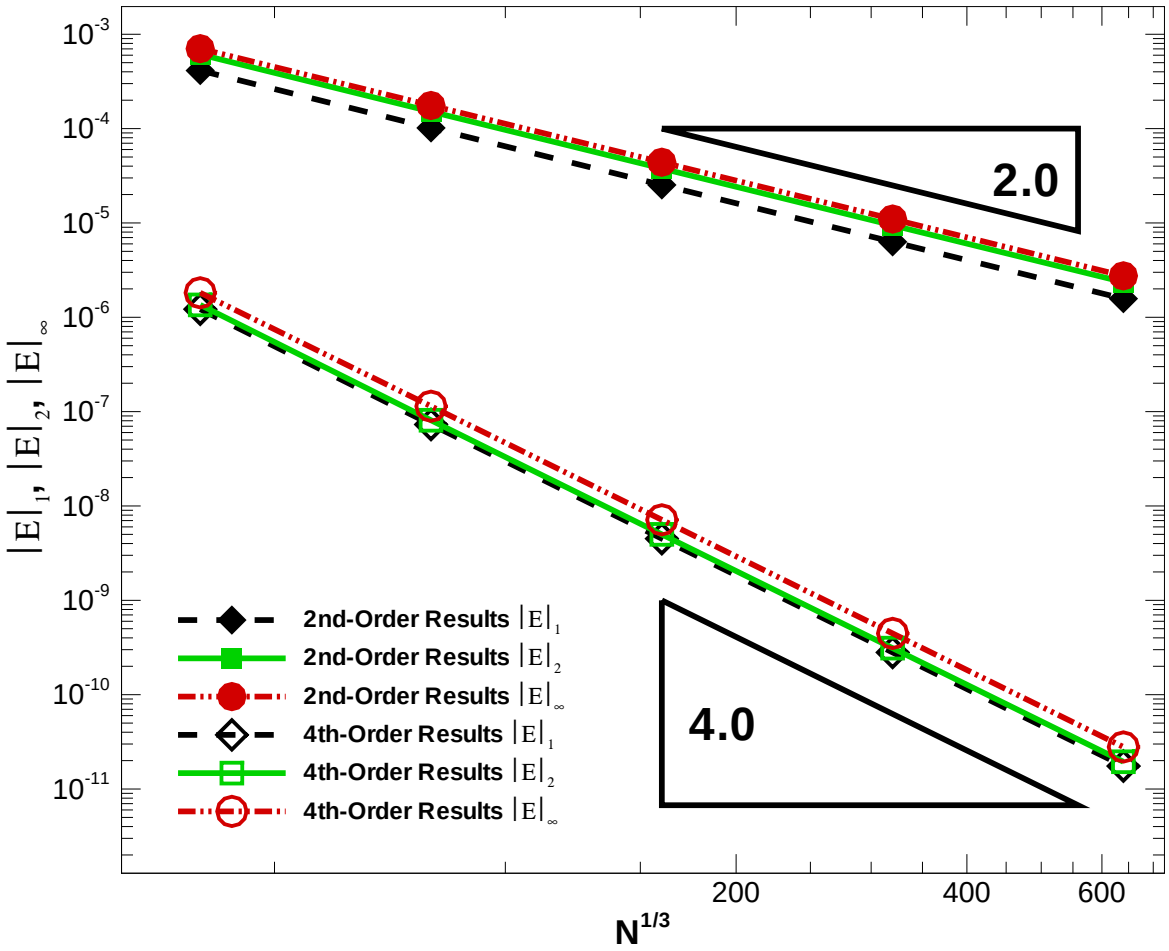


Figure 5.2: Convergence study for purely resistive MHD. Second-order accuracy is achieved through quadratic spatial reconstruction, and fourth-order accuracy through quartic reconstruction, which is one order more than is required for the advection terms.

5.2.2 Manufactured 3D Heat-Kernel Resistive MHD

To assess the accuracy of the proposed scheme in handling fully resistive MHD flow, convergence studies for a 3D manufactured solution on a 7-block cubed-sphere grid are considered. We construct the solution by prescribing a solution to the magnetic diffusion equation, Eq. 2.17

$$B_x(y,z,t) = \frac{1}{4\eta(t+\delta t)} e^{\frac{-(y^2+z^2)}{4\eta(t+\delta t)}}, \tag{5.14}$$

$$B_y(x,z,t) = \frac{1}{4\eta(t+\delta t)} e^{\frac{-(x^2+z^2)}{4\eta(t+\delta t)}}, \tag{5.15}$$

$$B_z(x,y,t) = \frac{1}{4\eta(t+\delta t)} e^{\frac{-(x^2+y^2)}{4\eta(t+\delta t)}}. \tag{5.16}$$

Note that each component of the magnetic field depends only on the two orthogonal directions other than the direction of the vector component itself. The decision to choose such form of the magnetic field is not only to ensure that the magnetic diffusion equation, Eq. 2.17 is satisfied, but also to make sure that, analytically, the magnetic field remains divergence-free at all times ($\nabla \cdot \vec{B} = 0$).

Unlike the test case of Sec. 5.2.1, where we kinematically isolate the induction equation from the rest of the system by ignoring the fluid-field interaction through the Lorentz force (see Chapter 4 of [5]), we now consider the full resistive MHD system. As exact solutions, velocity is chosen to be zero at all times, whereas pressure and density are set to be constant in both time and space. While these exact solutions satisfy continuity (Eq. 2.15), and induction (Eq. 2.17) equations, they do not satisfy the momentum (Eq. 2.16) and the energy (Eq. 2.18) equations. Similar to Problem 6 in Sec. 3.3.1 of [3], manufactured source terms, S_m and S_e, are added to the momentum and energy equations

$$S_m = \nabla \cdot \left(\vec{I}(p + \frac{\vec{B}\cdot\vec{B}}{2}) - \vec{B}\vec{B} \right), \tag{5.17}$$

$$S_e = \frac{\partial e}{\partial t} + \nabla \cdot \left(\eta \vec{J} \times \vec{B} \right). \tag{5.18}$$

For the $\frac{\partial e}{\partial t}$ term in S_e (Eq. 5.18), the exact solution for energy (hence, its time-derivative as well) can be obtained by making use of the relationship between energy and all other variables, i. e. the equation of state as given by Eq. 2.6. Likewise, all the other terms contained within the source terms, S_m and S_e, can be analytically obtained. Thus, the

solution given by

$$\mathbf{W}(x,y,z,t) = \begin{bmatrix} \rho_0, \\ 0, \\ 0, \\ 0, \\ \frac{1}{4\eta(t+\delta t)} e^{\frac{-(y^2+z^2)}{4\eta(t+\delta t)}}, \\ \frac{1}{4\eta(t+\delta t)} e^{\frac{-(x^2+z^2)}{4\eta(t+\delta t)}}, \\ \frac{1}{4\eta(t+\delta t)} e^{\frac{-(x^2+y^2)}{4\eta(t+\delta t)}}, \\ p_0 \end{bmatrix}, \tag{5.19}$$

is an exact solution to the resistive MHD equations with additional manufactured source terms, given as follows

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0, \tag{5.20}$$

$$\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot \left(\rho \vec{v}\vec{v} + \vec{I}(p + \frac{\vec{B} \cdot \vec{B}}{2}) - \vec{B}\vec{B} \right) = S_m, \tag{5.21}$$

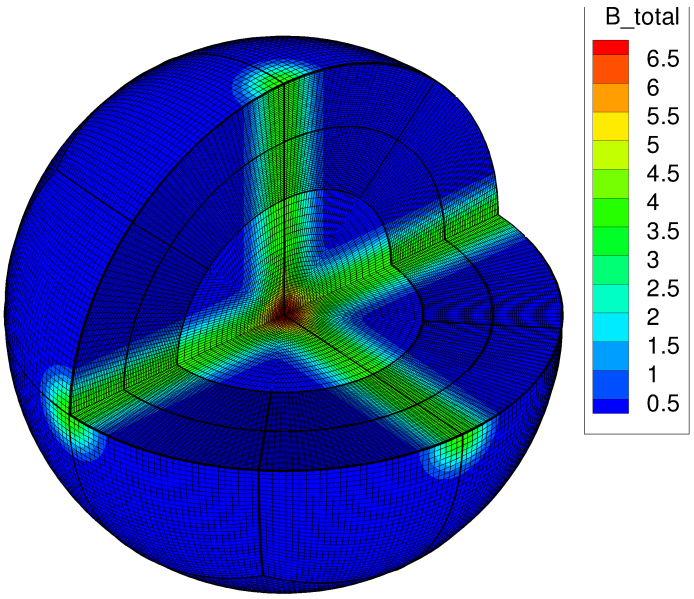
$$\frac{\partial \vec{B}}{\partial t} + \nabla \cdot (\vec{v}\vec{B} - \vec{B}\vec{v}) + \nabla \times (\eta \vec{J}) = 0, \tag{5.22}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot \left((e + p + \frac{\vec{B} \cdot \vec{B}}{2})\vec{v} - (\vec{v} \cdot \vec{B})\vec{B} + \eta \vec{J} \times \vec{B} \right) = S_e, \tag{5.23}$$

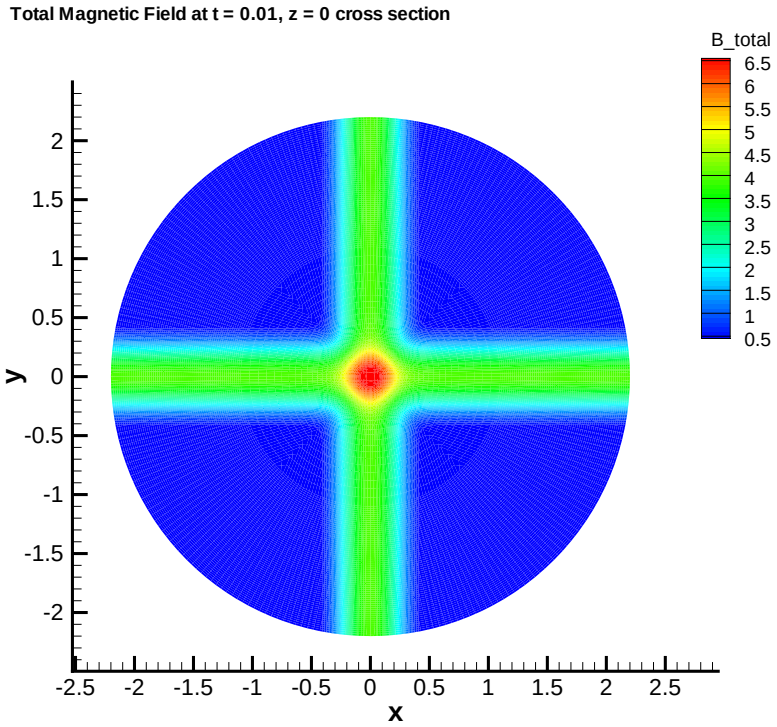
with S_m keeping the velocity zero at all times by zeroing the residual of the momentum equations, and S_e making sure that the energy is consistent with the temporal and spatial evolution of the magnetic field, $\vec{B}$, and the fact that ρ, p and $\vec{v}$ are constant in both space and time. The mechanism of adding source terms into an equation system is called the method of manufactured solution, or MMS [100], and has also been used to verify the CENO method as applied to the ideal MHD system and cubed-sphere grid [3, 8].

The simulation is performed up to $t = 0.01$, the solution of which can be seen in Fig. 5.3a and Fig. 5.3b. For the purpose of this simulation, the 7-block cubed-sphere grid is used as the computational domain. The inner and outer radii are given as $R_i = 1.1$ and $R_o = 2.2$, respectively. Fig. 5.4 gives error norms at $t = 0.01$ in the x-direction magnetic field as a function of grid size. It can be seen that the fourth-order method converges with approximately fourth-order accuracy for all error norms, and produces much higher accuracy when compared with the second-order method for a given grid resolution. These results were obtained on a series of grids ranging in size from seven 16×16×16 initial blocks to 3,584 16×16×16 cubed-sphere blocks, which corresponds to $28,672$ and $14,680,064$ total

cells, respectively. As the mesh is refined, the slopes of the L_1, L_2, and L_∞ norms approach, in the asymptotic limit, -1.871, -1.856, and -1.733 respectively, for the 2nd-order scheme, and -3.911, -3.885 and -3.792 for the 4th-order ($K=4$) scheme. These results clearly show that the expected orders of accuracy have been achieved, demonstrating the validity of the numerical treatment for elliptic flux evaluation (as discussed in Sec. 5.1). The resistive MHD terms will be used in the next chapter to simulate space physics flows.



(a) Contour plot of the total magnetic field, 3D view.



(b) Contour plot of the total magnetic field, 2D cross-section at z = 0.

Figure 5.3: Total magnetic field solution at $t = 0.01$ of the 3D manufactured heat kernel problem.

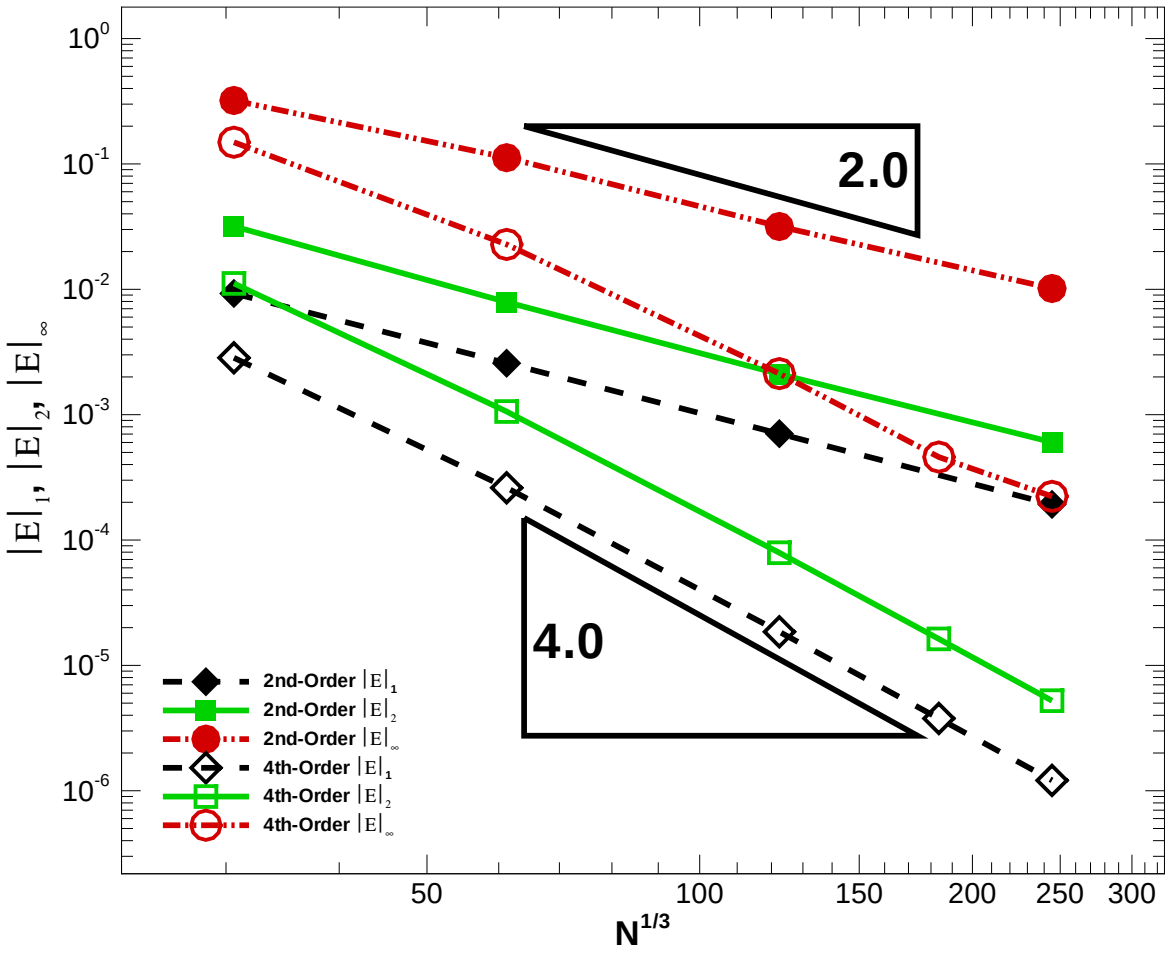


Figure 5.4: L_1, L_2, and L_∞ norms of the error in the x-component of the magnetic field, B_x, as a function of mesh resolution for the 3D manufactured heat kernel problem

Chapter 6

Preliminary Exploration of Space Physics Applications

This chapter serves as a culmination of implementations and applications of numerical tools developed in the previous chapters: finite-volume method and formulation as applied to resistive MHD, numerical treatment of second-order derivative flux terms, and development and utilization of 7-block cubed-sphere mesh to simulate a planetary object and its surrounding. At present, we have only conducted space physics simulations presented in this chapter with the second-order scheme. High-order simulations of space physics flows considered in this chapter are currently work in progress and subject for future work, and several issues have arised related to the implementation of such:

- High-Order accuracy would require implementation of high-order curved boundaries in three dimensions, especially at the interface between the interior and the exterior of the body, which is a significant effort beyond the scope of this thesis;
- We found that Powell source terms are preferable to GLM for the problems considered, and Powell's method has not been pursued for high-order accuracy.

We therefore delegate performing these space physics applications with a high-order scheme as a potential project for future work.

6.1 Parameters of Space Physics Simulations

Several space physics applications are considered. In particular, simulations pertaining to the conditions around the moon are performed, which assume the following solar wind parameters

$$n = 10^6 \text{ particles } /m^3, \tag{6.1}$$

$$V_{sw} = 400 \ \frac{km}{s}, \tag{6.2}$$

$$B_z = -5 \ nT, \tag{6.3}$$

$$T = 140,000 \ K, \tag{6.4}$$

where the velocity is in the x-direction, and the other components of the magnetic field are zero. Note that these solar wind conditions are the same conditions as used in [93, 101, 102] for Lunar simulations. These parameters are then normalized, following the procedures outlined in A.5, which result in the following unit-less initial conditions

$$\hat{\rho} = 7.0, \tag{6.5}$$

$$\hat{v} = 9.94464, \tag{6.6}$$

$$\hat{B_z} = -7.17372, \tag{6.7}$$

$$\hat{p} = 5.0, \tag{6.8}$$

where $\hat{v}$ is the velocity in the x-direction. Unless specified otherwise, the simulations considered in this chapter are performed on the 7-block cubed-sphere grid, with Powell's source terms to control divergence error (see Sec. 2.2.1). Ideal MHD equations are solved for the outer shell (by setting $\eta = 0$), or blocks 1 - 6 from Fig. 6.1a, which represent the flow domain outside the moon, whereas only magnetic diffusion equation is solved in the inner sphere (to let the magnetic field lines diffuse through the Moon). Thus, these simulations are effectively a hybrid combination of kinematics of MHD in the inner sphere, and dynamics of MHD on the outer shells.

6.2 Boundary Conditions

Boundary conditions describing the atmosphere of the inner body are set at the surface of the inner sphere. For this purpose, the surface of the inner sphere is divided into two parts: dayside, and nightside (see Fig. 6.1b). Following [93], a number density of $10^5 \ m^{-3}$ is imposed on the dayside. For the nightside, a number density of $10^4 \ m^{-3}$ is imposed. The temperature of the lunar atmosphere is chosen to be 400 K ([93, 103]).

With temperature and density imposed on the surface of the inner sphere, we can apply the ideal gas law to compute the dayside and nightside values of the pressure. Following the normalization procedures set out in Sec. A.5, the following boundary conditions for density and pressure can be obtained

$$\hat{\rho}_{dayside} = 0.7, \tag{6.9}$$

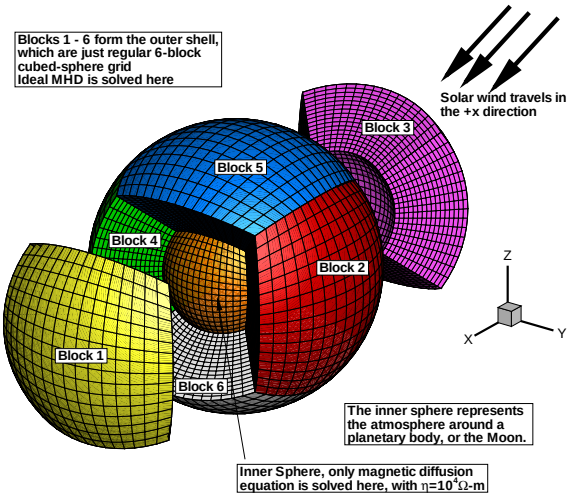
$$\hat{\rho}_{nightside} = 0.07, \tag{6.10}$$

$$\hat{p}_{dayside} = 0.00142857142857, \tag{6.11}$$

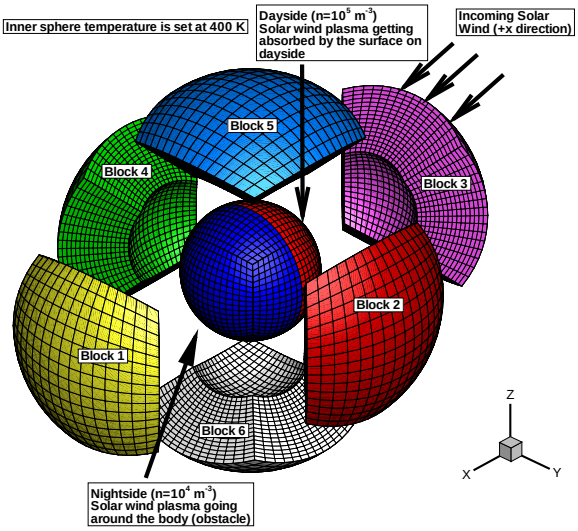
$$\hat{p}_{nightside} = 0.000142857142857. \tag{6.12}$$

To imitate particle absorption on the dayside, an extrapolation boundary condition is applied to velocity for the simulation domain exterior to the planet (the outer shell of the nested cubed-sphere mesh). More specifically, the plasma velocity near the boundary of the outer 6 blocks is extrapolated to the ghost cells of those blocks, to mimic the fact that the plasma is absorbed on the Moon's surface and does not see an obstacle due to its low density. On the nightside, reflection boundary conditions are implemented for the velocity vector by setting the normal component to zero. This is done to replicate the behavior of a solid impenetrable wall. Note that a similar boundary condition implementation can also be found in [104], which simulates solar-wind interaction with the Moon without modelling the Lunar interior, and without considering the presence of an intrinsic magnetic field, as we will do.

The magnetic field within the inner sphere is advanced in time by solving only the magnetic diffusion equation (by setting $\vec{v} = 0$ in Eq. 2.17). On the outer shells, the full ideal MHD system of equations is considered. In both the dayside and nightside cases, the values of the magnetic field at the surface of the inner sphere (or the bottom part of the outer shell) are imposed through communication between adjacent blocks by means of ghost cells. Effectively, the interior (inner sphere) magnetic field acts as a boundary condition for the exterior field (outer shell), and vice versa.



(a) A three-dimensional view of the seven-block cubed-sphere grid. The inner sphere here represents the interior of a planetary object (with a thin atmosphere on its surface), a planet, or, in this case, the Moon, where a resistivity of $10^4\ \Omega - m$ is assumed (as given in [105]). Only the magnetic diffusion equation is solved in the inner sphere. The outer shell discretizes the domain around the object in question. In this shell, the full ideal MHD system is solved (setting the resistivity, η, to zero).



(b) The blown-up view of the 7-block cubed sphere mesh for the purpose of the lunar atmosphere simulation. The part of the inner sphere that faces the solar-wind directly ($x < 0$) is the dayside, whereas that opposite of the solar wind is the nightside. Boundary conditions are applied accordingly.

Figure 6.1: Three-dimensional views of the 7-block cubed-sphere grid.

6.3 Simulations Without Intrinsic Magnetic Field

6.3.1 Seven-Block Cubed-Sphere Simulation of Lunar Wake

The intrinsic magnetic field of the Moon is very weak in comparison to that of Earth [106]. In particular, it is almost entirely crustal rather than dipolar in nature (hence, the term "magnetic anomalies"). In this section, simulations of the interaction between the solar wind and the lunar atmosphere are considered and performed, and we assume that the Moon lacks any intrinsic magnetic field. As mentioned in the previous section, the simulations are performed on a 7-block cubed-sphere grid. Since the Interplanetary Magnetic Field (IMF) has to diffuse through the Moon, a resistivity of $10^4\ \Omega - m$ is chosen

inside the inner sphere. Only the magnetic diffusion equation is solved inside the inner sphere (since solar-wind particles cannot penetrate the body). Ideal MHD is solved in the outer shell.

For these simulations, we once again use the local Lax-Friedrichs function to calculate the values of inviscid fluxes, whereas the values of elliptic fluxes are obtained through arithmetic averaging of the left and the right states at each of the Gauss quadrature points (see Sec. 5.1). The second-order scheme employed for these simulations is based on k-exact reconstruction with $k = 1$ combined with the Venkatakrishnan limiter. Unless noted otherwise, the simulations are done on a nested-sphere grid with seven root blocks. Each of the six outer shell root blocks (see Fig. B.1) is further divided into 16 blocks in the radial direction. There are, in total, 96 blocks forming the outer shell, and 1 block to discretize the inner sphere. Each of these blocks consists of $64 \times 64 \times 64$ cells, which corresponds to $25,427,968$ total cells. To integrate in time, a five-stage optimally smoothing technique is used. Fig. 6.2 shows the solution of the lunar wake simulation when the GLM-MHD system (2.2.2) is employed to control divergence error. Neither bow-shock formation, nor a pile-up of the magnetic field lines, is observed ahead of the body, which is as expected from the interaction between the solar wind, and a body without any intrinsic magnetic field [99]. However, contrary to the physical fact that the incoming solar-wind possesses a super-fast speed ($v > c_f$), hence, no information should propagate in the negative direction, we can already see perturbation of the magnetic field lines upstream to the solar wind. We have briefly discussed the existence of both negative and positive eigenvalues in Sec. 2.2.2.1, and the solution as shown by Fig. 6.2 confirms the existence of the propagation of information in both direction (the magnetic field lines upstream of the Moon getting perturbed, even though the flow is faster than the fast magnetosonic wave).

The same simulation using Powell source terms was conducted, and the results are presented on Fig. 6.3. From the figure, it can be seen that the solution now looks more physically valid (qualitatively very similar to Fig. 7.24 from [99]), with magnetic field lines upstream of the Moon no longer perturbed. We can see the lunar wake, and a near-vacuum region just downstream of the Moon, which is also called the "evacuated region" [99], resulting from the solar wind plasma that travels along the flanks of the Moon. It can also be seen that the magnetic field lines seem to bend inward into the wake, which is the result of some magnetic flux getting carried into the wake by the solar wind plasma flow around the flanks [99]). Once again, a bow shock does not form ahead of the Moon, due to the lack of ionosphere, or lunar intrinsic magnetic field in the simulation. Unlike with GLM, the magnetic field lines are not perturbed upstream of the solar wind, which makes sense since the solar wind is faster than the fast magnetosonic wave, and no physical information should travel upstream of the direction of the solar wind. It is at this point that we realized that Powell's scheme to control divergence error would be the better choice for solar-wind related simulations, since our observation has pointed to the Powell scheme producing a more physically relevant solution when the flow is super-fast, and all the physical wave speeds are all positive. Thus, for the rest of this chapter, only the Powell

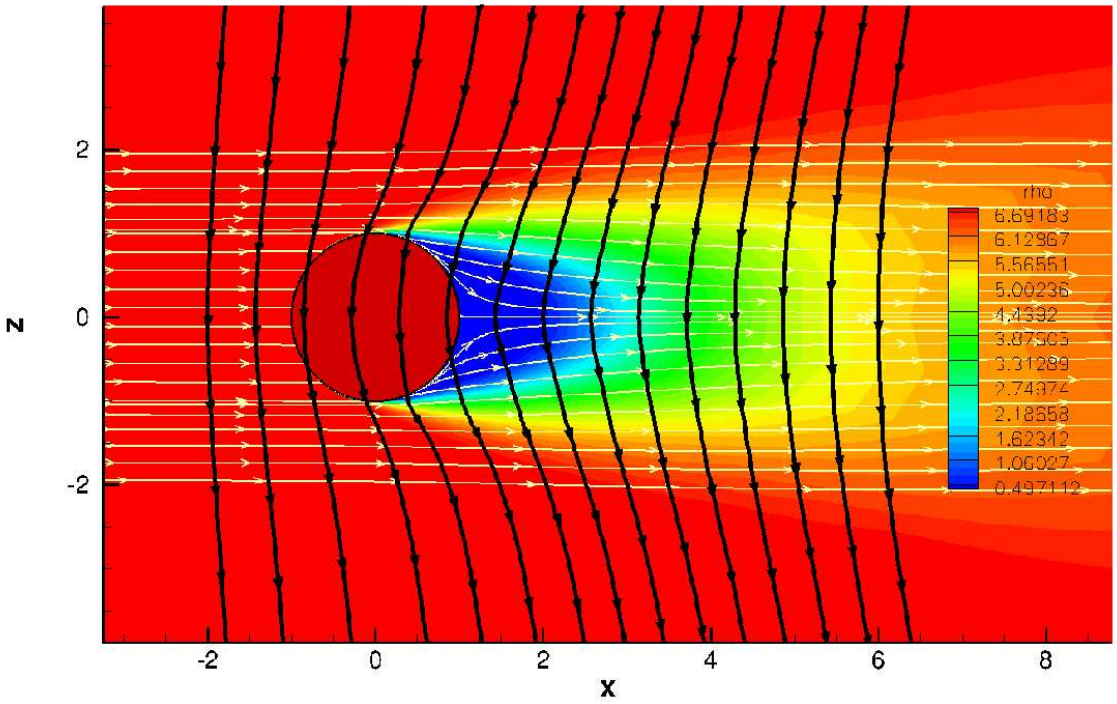
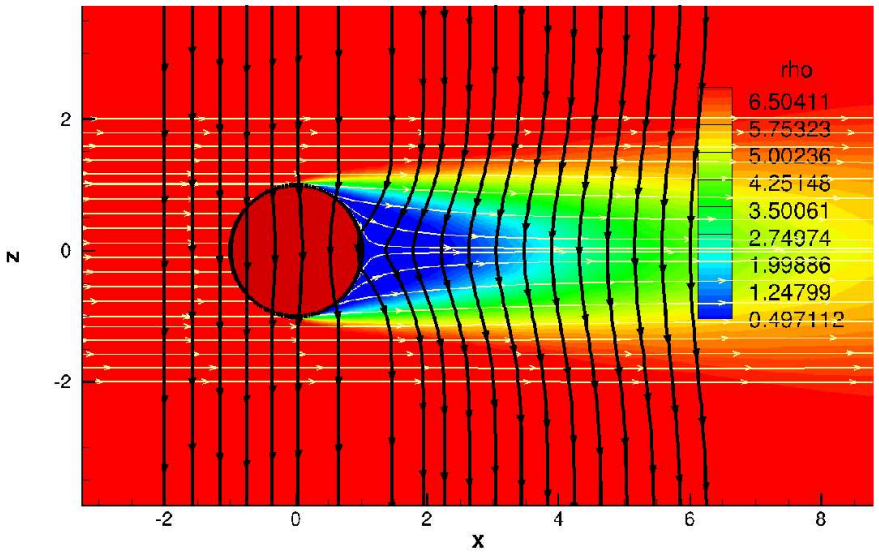
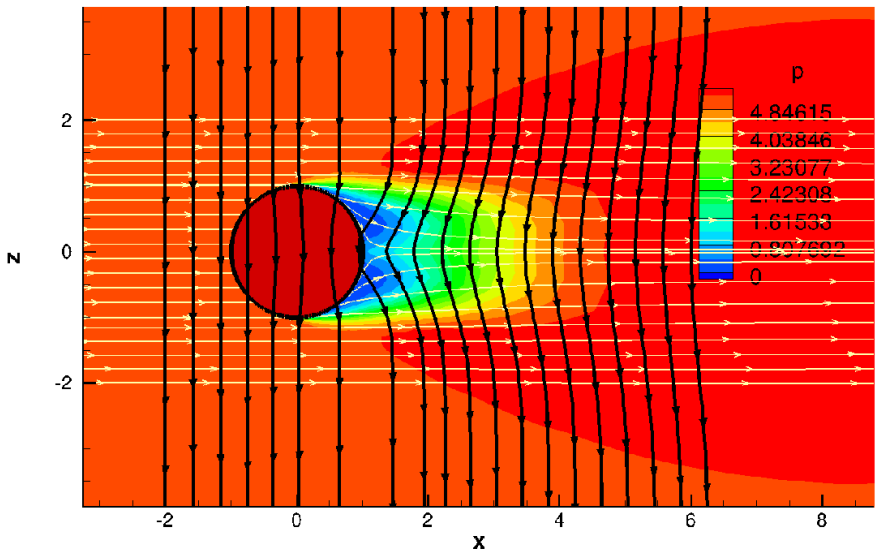


Figure 6.2: Solution of the lunar wake simulation. GLM-MHD is used to control divergence error. Perturbation of the magnetic field lines is apparent upstream of the Moon, despite the fact that the solar wind possesses a super-fast velocity, which should imply no propagation of information upstream into the solar wind. Here, the yellow lines represent the streamlines, and the black lines represent the magnetic field lines.

source term scheme (rather than GLM-MHD) is used to correct divergence error.



(a) Density contours of the lunar wake simulation. Black lines represent magnetic field lines, and yellow lines represent stream lines.



(b) Pressure contours of the lunar wake simulation.

Figure 6.3: Lunar wake simulation results using Powell source terms, cross section taken at $y = 0$.

6.4 Simulations With Intrinsic Magnetic Field

While the previous simulations show the lack of bow shock formation ahead of the Moon (see Sec. 6.3), Lunar Prospector data shows disappearance and reappearance of an increase in the magnetic field at the satellite altitude over regions with significant anomalous magnetic field [93, 107, 108, 109]. While it is true that the Moon indeed lacks a significant intrinsic magnetic field, the data from Lunar Prospector, which also reported an increase in dynamic pressure to coincide with the disappearance of the increase of the magnetic field, seems to suggest the presence of small-scale magnetospheres over regions with large magnetic anomalies.

To model the intrinsic magnetic field of the Moon, a small dipole is buried just underneath the surface, given by the following expressions

$$B_{0_x} = 3(x - x_0)(z - z_0)\frac{M_z}{r^5}, \tag{6.13}$$

$$B_{0_y} = 3(y - y_0)(z - z_0)\frac{M_z}{r^5}, \tag{6.14}$$

$$B_{0_z} = (3(z - z_0))^2 - r^2)\frac{M_z}{r^5}, \tag{6.15}$$

where x_0, y_0, and z_0 denote the x, y, and z coordinates of the dipole center. The radius, r, is calculated from the center of the dipole, and can be written as

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}. \tag{6.16}$$

The singularity, which occurs when $r = 0$, is regularized by making sure that the magnitude of the x, y, and z coordinates to be used to compute the dipole are not smaller than 10^{-6}

$$x_i = \begin{cases} x_i, & \text{if } |x_i| > 10^{-6} \\ \text{sign}(x_i)10^{-6}, & \text{if } |x_i| <= 10^{-6}, \end{cases} \tag{6.17}$$

where x_i can be either x, y or z.

The intrinsic magnetic field is included in the inviscid flux calculations for the induction equation, but excluded from the elliptic flux calculation. The induction equation can then be written as

$$\frac{\partial \vec{B}}{\partial t} + \nabla \cdot (\vec{v}\vec{B}_t - \vec{B}_t\vec{v}) + \nabla \times (\eta \vec{J}) = 0, \tag{6.18}$$

where

$$\vec{B}_t = \vec{B}_0 + \vec{B}_1, \tag{6.19}$$

and

$$\vec{J} = \nabla \times \vec{B}_1. \tag{6.20}$$

Here, $\vec{B}_0$ is the intrinsic field, $\vec{B}_1$ is the perturbation field, and $\vec{B}_t$ is the total field. Note that the total magnetic field, $\vec{B}_t = \vec{B}_0 + \vec{B}_1$, is included in the calculation of the inviscid flux, whereas only the perturbative magnetic field, $\vec{B}_1$, is included in the calculation of the elliptic flux. This practice preserves the intrinsic magnetic field, $\vec{B}_0$, throughout the course of the simulation, rather than letting it diffuse (as it would if the intrinsic magnetic field too were included as part of the elliptic flux), since the crustal magnetic of the Moon is embedded in the lunar rocks, and does not diffuse away within the considered short timescales of our simulations.

To enhance the robustness of the scheme, and to avoid negative pressure error, the absolute values of the pressure and density are taken when evaluating some of the wave speeds. This measure usually has worked when solving steady state solutions, and negative pressure is encountered during transient periods.

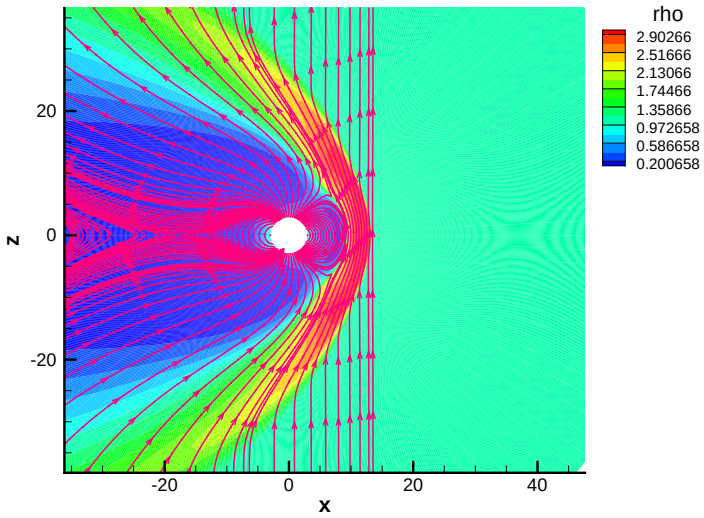
6.4.1 Simulation of Earth's Magnetosphere Solution on Regular Cubed-Sphere Grid

As a baseline case, we consider in this section a simulation of the interaction between the solar wind and the Earth's magnetosphere. For the purpose of this reference simulation, the same parameters as in [30] are used, and a 6-block cubed-sphere mesh (a hollow sphere, rather than the 7-block cubed solid nested sphere) is used to conduct the simulation. The mesh is defined by the inner and outer sphere surfaces with radii $R_i = 3.0$ and $R_o = 60.0$ (in normalized units), with $r = 1.0$ representing the radius of Earth. Similar grid resolutions are used as in Sec. 6.3, where each block consists of $64 \times 64 \times 64$ cells, and each outer shell block is divided into sixteen blocks in the radial direction, corresponding to a total of 96 blocks, or $25,165,824$ cells. The Lax-Friedrichs flux function is used to calculate the values of the fluxes. The second-order scheme employed for these simulations is based on k-exact reconstruction with $k = 1$ combined with the Venkatakrishnan limiter. Because this simulation is steady state, a five-stage optimally smoothing time-integration method is used.

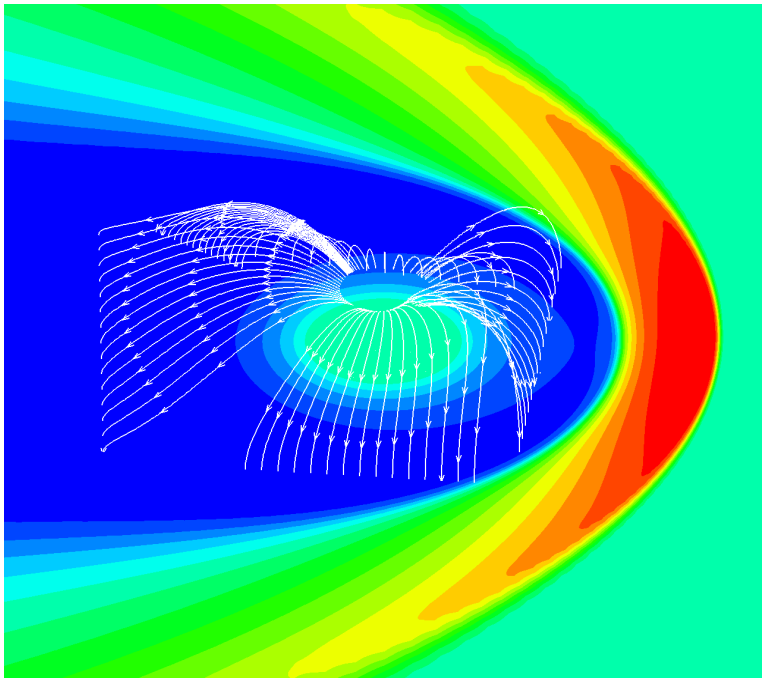
The incoming solar wind is modelled as having a velocity of 400 km/s in the negative x-direction. Number density is set to 5 $/cm^3$, or $5,000,000$ $/m^3$. The IMF is chosen to be 8 nT. After normalization, we have the following unit-less values $\rho = 1.0$, $p = 0.6$, $v = -8.0$ and $B_z = -1.950311$. A dipole, buried at the origin ($x_0 = 0.0$, $y_0 = 0.0$, and $z_0 = 0.0$), and intrinsic magnetic field, given by Eq. 6.13 to Eq. 6.15 is considered, with $M_z = 0.3$. These parameters correspond to the same solar wind test case as considered in Sec. 6 of [30], where two possible directions for the IMF are considered. Here, only the Southward IMF case is considered, where the IMF is set to be parallel with the terrestrial dipole moment, since this test case leads to more significant magnetic reconnection (also known as the "open magnetosphere" case) compared with the Northward IMF case, which represents the "closed magnetosphere" case. As boundary conditions, we follow the implementation

of [30] as well: $\rho = 1.0$, $v = 0.0$, $B_{1_r} = 0.0$, and $p = 8.0$. The other two components of the perturbative magnetic field are extrapolated from the interior cells. The intrinsic magnetic field is calculated by applying Eq. 6.13 to Eq. 6.15 at the boundary. These conditions are applied on the inner part of the 6-block cubed sphere to represent the atmospheric conditions about 3 times the Earth's radius away from the origin.

Fig. 6.4 illustrates the steady-state density solution for the case where the IMF is parallel to the direction of the earth's dipole moment (in this case, southward). This compares reasonably well, at least qualitatively, with Fig. 9 and Fig. 10 from [30]. Magnetic reconnection occurs between the intrinsic magnetic field lines of the planet and those of the IMF. The usual features of the magnetosphere are observed: a bow shock can be seen ahead of the body, with the magnetosphere shielding the dayside surface of the planet from the solar wind particles. A pile-up of magnetic field lines is observed between the magnetosphere and the bow shock, a region more commonly known as "magnetosheath". At the Nightside, an "x-like" structure can be observed, where the IMF and the intrinsic magnetic field lines are observed to "disconnect". Once again, this result is consistent, at least qualitatively, with that of [30].



(a) Cross section of the steady state solution at $y = 0$. We can here observe magnetic reconnection between the intrinsic dipole magnetic field lines of the planet and those of the IMF. An "x-like" structure is observed nightside, indicating the location where the magnetic field lines are reconnecting. Here, the yellow lines represent the streamlines, and the purple lines represent the magnetic field lines.



(b) View of the Atmosphere around the Northern hemisphere of the earth ($z > 0$). Pressure contours are presented here. The white lines represent the magnetic field lines. This figure compares favourably with Fig. 10 of [30].

Figure 6.4: Steady State Solution of Terrestrial magnetic field and solar wind interaction.

6.4.2 Mini-Magnetosphere Simulation on Seven-Block Cubed Sphere

Several simulations of the interaction between solar wind and the lunar atmosphere have already been conducted in Sec. 6.3.1. It is, however, important to emphasize that the presence of mini magnetosphere has been observed over the region with large magnetic anomalies [93, 108, 110, 111]. Both numerical [93, 102, 112] and laboratory simulations [112, 113] have confirmed this observation. Test cases with two different values for the dipole moment, M_z, are considered. The intrinsic magnetic field of the moon is modelled as a dipole, given by Eq. 6.13 to Eq. 6.15. Rather than located at the origin, the dipole is buried about 25° below the xz plane, at $y = 0$. The dipole center is located 0.95 radius away from the origin, which in turns gives: $x_0 = -0.86099$, $y_0 = 0.0$, and $z_0 = -0.401487$ (in normalized units where $r = 1$ represents the radius of the Moon).

For the first case, the dipole moment is chosen to be $M_z = 10^{-6}$, parallel to the incoming solar-wind magnetic field (the IMF). This choice of dipole moment is much smaller than the one used in Sec. 6.4.1. Also, unlike in Sec. 6.4.1, we use the nested solid sphere mesh (7-block cubed-sphere grid), generated as explained in Appendix B. We use the exact same grid resolutions as in Sec. 6.3.1, which gives a total of 97 blocks and $25,427,968$ total cells. Also similar as in the previous sections, a five-stage optimally smoothing time-integration technique is used, due to the steady-state nature of the mini-magnetosphere simulations.

A small-scale magnetosphere is clearly apparent, as shown by Fig. 6.5, producing a small bow shock, denoted by an increase in density just before the flow gets deflected. Density then decreases rapidly after the shock, effectively shielding the surface with high concentration of magnetic anomalies from the solar wind particles. To further analyze the so-called mini magnetosphere, a cut is taken across the bow-shock in the radial direction (see Fig. 6.6a). Because of the way the outer shells of the nested sphere are constructed, this cut (denoted by the straight line shown on Fig. 6.6a), is aligned with the grid, allowing for plotting several cell relevant quantities without interpolation. See Fig. 6.6b, Fig. 6.6c, and Fig. 6.6d for plots of density, pressure and total magnetic field (respectively) against the radial distance from the origin. The mini-magnetosphere shares a lot of similarities with the usual magnetosphere as observed in Sec. 6.4.1, albeit at a much smaller scale. The shielding of the lunar surface from the solar-wind particles denotes the area where the mini-magnetosphere is dominant. The boundary of this shielding denotes the magnetopause. Between the magnetopause and the bow shock, a very narrow magnetosheath region can be observed, where the density increases, and then reduces to the level observed inside the magnetosphere (as seen from Fig. 6.6b, the magnetosheath region spreads from roughly $r = 1.1$ to $r = 1.35$).

Keep in mind that the radial distance of 1.0 corresponds to the location of the surface of the Moon. From Fig. 6.6d, it can be gathered that the total magnetic field around the surface is around $1,300$ in normalized units, or about 900 nT, whereas at about 100 km of

altitude, the magnitude is about 325 in normalized units, or 225 nT. It is acknowledged that the fall-off of the dipole is much smaller than what was observed from Lunar Prospector, where a magnetic anomaly of 290 nT was observed on the lunar surface, which falls off to around 2 nT at 100 km altitude.

Several other cases have been attempted to increase the fall-off of the intrinsic magnetic field to better reflect the values provided by the Lunar Prospector. But unfortunately, the smallest value for which we have been able to reproduce the mini-magnetosphere was when we set the dipole moment to be $M_z = 5(10^{-7})$. When a value a little smaller than the aforementioned is chosen ($M_z = 10^{-7}$), the mini-magnetosphere fails to appear, and the solution produces only the wake without any small-scale bow-shock forming ahead of the flow. For the case of $M_z = 5(10^{-7})$, a cut in the radial direction was taken, and we can see from Fig. 6.7 that the size of the magnetosheath region slightly reduces (the first case has this region from $r = 1.1$ to $r = 1.35$, whereas this region reduces so that it goes from $r = 1.1$ to around $r = 1.25$), implying the expected direct correlation between the size of the magnetosheath and that of the dipole moment. From Fig. 6.6d, it can be gathered that the total magnetic field around the surface is around 640 in normalized units, or about 440 nT, whereas at about 100 km of altitude, the magnitude is about 160 in normalized units, or 110 nT.

Clearly, the fall-off of a factor of four within 100 km is not nearly as fast as what was observed from experiments [108], and this issue too was reported in [93] and [102], where choosing too small of a dipole moment would also lead to the disappearance of the mini-magnetosphere. The authors of [102] then attempted to mitigate the over-estimation for the magnetic field 100 km above the surface by using multiple dipoles to model the magnetic anomalies, even though they ended up concluding that the magnetic field at 100 km required to form a mini-magnetosphere is still larger than what was observed by the Lunar Prospector. It is then important to keep in mind that these dipolar magnetic field models are but low-order approximations of the actual magnetic anomalies, and more realistic models (using spherical harmonics [114], the crustal magnetic field data from Lunar Prospector [115] or a multiple dipole model [102]) for the magnetic anomalies would then be necessary. Such study is beyond the scope of this work, and may be considered for future work (see Sec. 7.1.2).

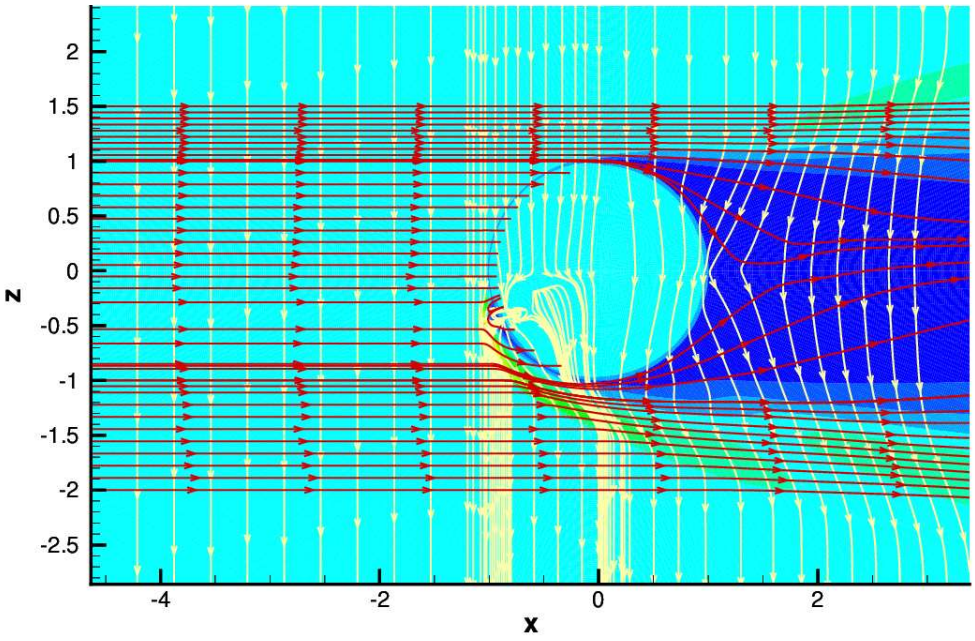
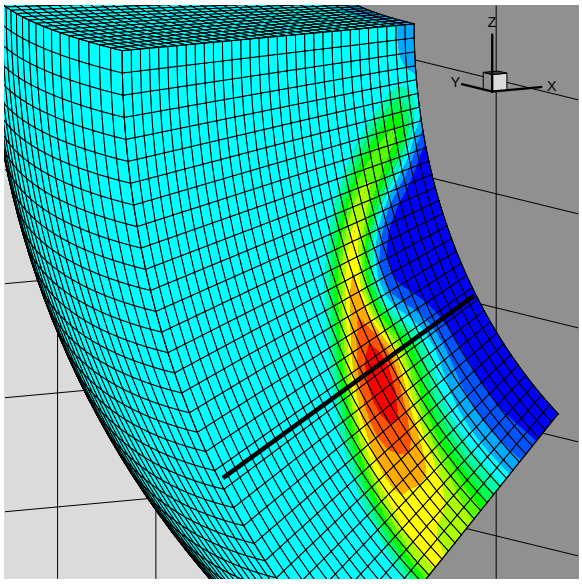
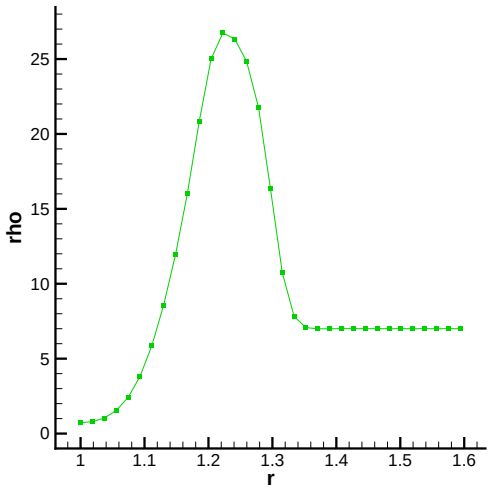


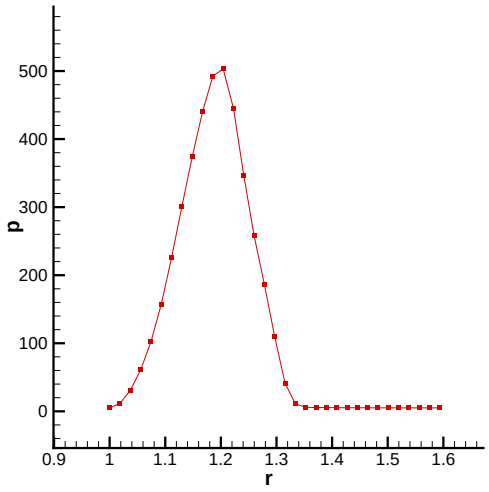
Figure 6.5: Simulation of mini-magnetosphere of the lunar atmosphere. A small dipole moment is buried, which is angled 25° from the x-axis in the xz plane. The dipole center is located at (in normalized units) $x = -0.86099$, $y = 0$, and $z = -0.401487$, with the radius of the Moon being the reference length for normalization. Here, yellow lines represent the magnetic field lines and black lines represent the streamlines.



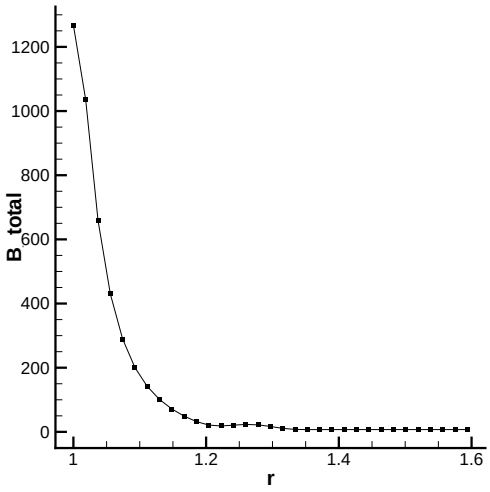
(a) Direction of the cut across the mini-magnetosphere. Note that the cut is in the radial direction, which is aligned with the grid.



(b) Density, ρ, as a function of radial distance from the centre of the sphere along the cut as described in (a). Every dot corresponds to a cell centre value.

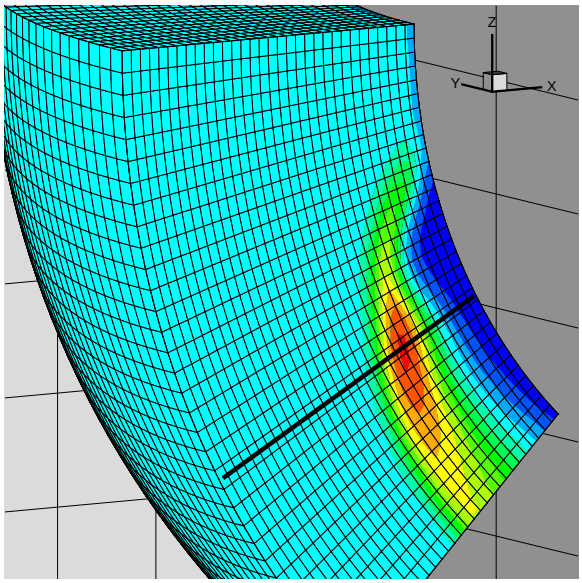


(c) Pressure, p, as a function of radial distance from the centre of the sphere along the cut described in (a).

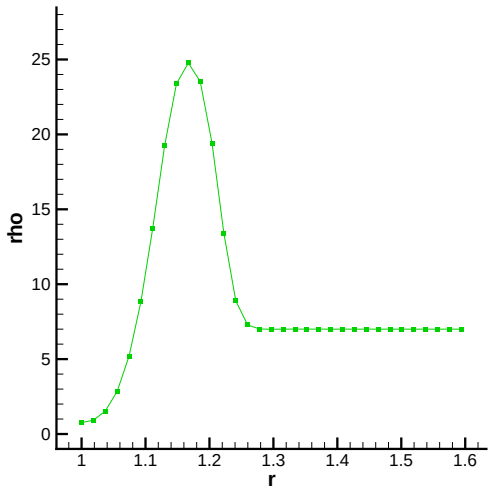


(d) Total magnetic field, B_t, as a function of radial distance from the centre of the sphere along the cut described in (a).

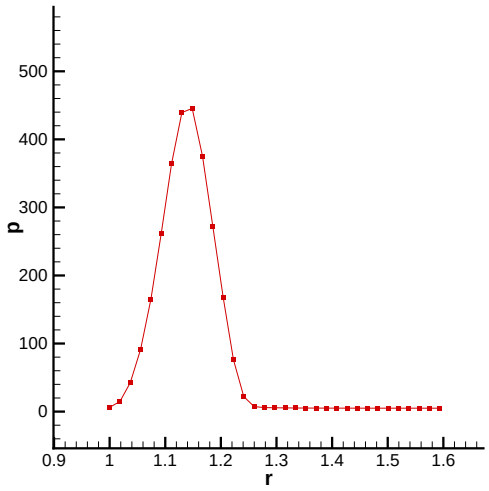
Figure 6.6: Radial Cuts across the mini-magnetosphere. A dipole moment of 10^{-6} is chosen to define the intrinsic magnetic field. In normalized units, the dipole center is located at $x = -0.86099$, $y = 0$, and $z = -0.401487$.



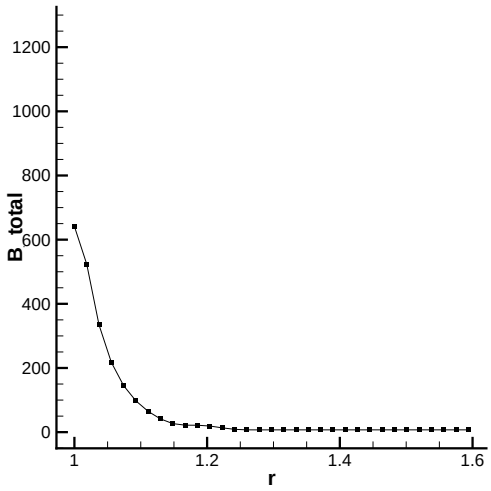
(a) Direction of the cut across the mini-magnetosphere. Note that the cut is in the radial direction, which is aligned with the grid.



(b) Density, ρ, as a function of radial distance from the centre of the sphere along the cut described in (a).



(c) Pressure, p, as a function of radial distance from the centre of the sphere along the cut described in (a).



(d) Total magnetic field, B_t, as a function of radial distance from the centre of the sphere along the cut described in (a).

Figure 6.7: Radial cuts across the mini-magnetosphere. A dipole moment of $5(10^{-7})$ is chosen to define the intrinsic magnetic field. On normalized unit, the dipole centre is located at $x = -0.86099$, $y = 0$, and $z = -0.401487$.

Chapter 7

Future Work and Concluding Remarks

High-order finite-volume schemes for MHD systems have been proposed, and results presented and analyzed. We have considered the ideal MHD system in both two and three dimensions (Chapter 3 and Chapter 4 respectively), resistive MHD (Chapter 5), and also a multiphysics context where ideal MHD is solved on some part of the domain, and magnetic diffusion on the rest (Chapter 6).

This study has found that, in many situations, simulations performed with high-order schemes produce much better solution when compared to the results of the second-order schemes for the same cost. This is apparent in both two dimensions and three dimensions. In particular, where analytical solutions are available, convergence studies have been performed to demonstrate the effectiveness of high-order methods in resolving the flows. As evident from the results of Sec. 3.2.1.1, Sec. 3.2.1.2, Sec. 3.2.1.3, Sec. 4.2.2, Sec. 5.2.1, and Sec. 5.2.2, fourth-order accuracy is observed as expected from the numerical method, and a very significant improvement over second-order methods is demonstrated. We also observed significant improvements on problems with shocks, such as the Shu-Osher test case, in both 2D and 3D, on rotated coordinate systems (Sec. 3.2.2.2 and Sec. 4.2.2.2 respectively), despite the method only being formally first-order accurate. Even with the presence of the shock, the Shu-Osher flow was resolved better by the high-order scheme, capturing the small-scale features more accurately than second-order scheme. Likewise, we see how convergence study of the expanding tube problem, Sec. 3.2.1.4, shows second-order convergence when using a high-order method, which is to be expected due to the presence of a weak discontinuity in the solution. Nevertheless, the errors are still lower than when using a second-order scheme. We also see that fourth-order of accuracy is achieved for problems involving resistive MHD, proving the capability of the schemes to handle second-order derivative terms with high spatial accuracy.

Lastly, several space physics applications have been considered. We first found that

using the Powell source term technique to control divergence error leads to better results for the Moon-solar wind interaction we considered. This outcome could have been foreseen judging from the analysis of the eigenvalues of the GLM-MHD system in Sec. 2.2.2.1, where it is noted that eigenvalues of both signs will always be present in the flow, regardless of the actual plasma velocities and wave speeds. The fact that solar-wind flows are usually super-fast (faster than the fastest wave-speed in the system) means that information should only propagate in the same direction as the flow, and all the physical eigenvalues are positive. However, it is clear from Sec. 6.3.1 that this creates a problem in the lunar-wake simulations with GLM-MHD, with perturbed magnetic field lines upstream of the body. This issue is not observed when the Powell technique is used, which produces physically correct solutions, leading to the decision to employ the Powell divergence control technique for all the other space-physics related applications. We also perform simulations which involve intrinsic magnetic field, which is used to approximately represent the Lunar magnetic anomalies. These simulations show the appearance of a mini-magnetosphere around the area with significant magnetic anomalies, which is consistent with the observational finding of the Lunar Prospector [108]. We also find that the total magnetic field strength does not fall off in magnitude as fast as what was observed from the Lunar Prospector data. Further study of this issue is beyond the scope of this thesis, and will be discussed further as a potential future work in Sec. 7.1.2.

The space physics applications represent the culmination of the development of numerical tools for magnetohydrodynamic systems. These space physics simulations made use of the hybrid system of ideal MHD and the magnetic diffusion equations, a nested solid cubed-sphere mesh, and also the rotation mechanisms developed to deal with stencil generation near degenerate edges. We have yet to perform these simulations in high-order, however, and implementation of this will be subject of future work.

7.1 Potential Future Work

"Dissertations are not finished; they are abandoned."

– Dr. Fred Brooks

7.1.1 High-Order Discretization of the Powell's Source Terms

Chapter 6 has preliminarily presented potential space physics applications. While we have been able to make use of the majority of the tools developed for the purpose of this work, all simulations conducted in Chapter 6 are performed using a second-order spatially accurate method. A task for the future is then to re-run these simulations with high-order, to see if any significant improvements can be achieved. While it is true that these space physics

simulations contain shocks or other forms of discontinuities, it may still be worthwhile to use high-order methods since significant improvements have been observed on, for example, some test cases with discontinuities such as the Shu-Osher problems, (Sec. 3.2.2.2 and Sec. 4.2.2.2), and the expanding tube problem (Sec. 3.2.1.4).

This brings us to the discussion of the Powell source terms (Sec. 2.2.1). While it is true that adding source terms that are proportional to the $\nabla \cdot \vec{B}$ breaks conservation, the problem with having both signs of eigenvalues at all times in GLM-MHD (which is discussed in Sec. 2.2.2.1) is a concern for simulations of super-fast flows, which physically should only have positive eigenvalues. These issues manifest themselves when we tried using GLM-MHD to resolve the lunar wake test case (Sec. 6.3.1), where perturbations of the magnetic field lines were observed even upstream to the solar-wind. Using Powell source terms, we can see how the magnetic field lines are straight on the dayside, and get bent only due to the diffusion of the inner sphere on the nightside. Fig. 6.2 and Fig. 6.3 clearly demonstrate this difference. It is therefore very important to ensure that the Powell source terms be computed with high-order accuracy, when high-order methods are requested. To discretize the source terms in high-order, the product rule is used before the divergence theorem can be used. The integral involves a multiplication of a scalar, ϕ, and the divergence of the magnetic field, $\nabla \cdot \vec{B}$

$$\int \phi \nabla \cdot \vec{B} \; dv = \oint \phi \vec{B} \cdot \hat{n} \; dS - \int \nabla \phi \cdot \vec{B} \; dv. \tag{7.1}$$

Within the context of the Powell source terms, ϕ takes on the values of the components of $\vec{B}$ for the momentum equation (Eq. 2.30), the components of $\vec{v}$ for the induction equation (Eq. 2.31), and $\vec{v} \cdot \vec{B}$ for the energy equation (Eq. 2.32). For first- and second-order accuracy, there is no issue since the integral of the left hand side of Eq. 7.1 can be approximated accurately enough (second-order accurate) using the midpoint rule for both ϕ and $\nabla \cdot \vec{B}$

$$\int \phi \nabla \cdot \vec{B} \; dv \approx (\phi)_{midpoint} (\nabla \cdot \vec{B})_{midpoint} \Delta V. \tag{7.2}$$

However, for high-order schemes, to achieve higher than second-order accuracy, the full form of Eq. 7.1 may need to be approximated accurately enough to the desired order of accuracy.

7.1.2 Better Numerical Models for Lunar Magnetic Anomalies

Another possibility for future research projects would be to consider different numerical models to approximate lunar magnetic anomalies. As reported in Sec. 6.4.2, the total magnetic field fall-off obtained at 100 km altitude is not nearly as fast as observed from Lunar Prospector data [108]. Several potential ways to deal with this issue may be considered. Harnett and Winglee [102] considered using multiple dipoles to model the lunar magnetic

anomalies. They found that using multiple dipoles to model the anomalies resulted in the reduction of the magnitude of the magnetic field at 100 km altitude, though the fall-off is still not as rapid as observed by the Lunar Prospector. Another way would be to consider a global spherical harmonics model to more realistically approximate the crustal magnetic field of the Moon [114], or to take advantage of experimental data provided by Lunar Prospector as already done by [115]. These projects would make good candidates for future work.

7.1.3 Non-Ideal MHD Simulations of Solar-Wind Interaction with the Atmosphere of Mars or the Moon on Nested Cubed-Sphere Grid

Similar as in the case of the Moon, Mars too lacks a significant magnetosphere, and most of Mars' intrinsic magnetic is crustal in nature [116], resembling the magnetic anomalies of the Moon. Several observational measurements resulting from the various instruments and stages of the MGS (Mars Global Surveyor) program [117, 118, 119, 120] are available, and have given rise to several spherical harmonic numerical models [121, 122] to approximate the Martian crustal magnetic anomalies. Numerical non-ideal MHD simulations have been conducted in three-dimensions, where a 90^{th}-degree spherical harmonic model [122] to model the Martian crustal magnetic anomalies was used. To more accurately approximate the behaviour of plasmas, kinetic effects such as Hall and anisotropic pressure gradient terms may be included. It would be interesting to try to reproduce this work as a proof of concept of our code, our schemes and the other numerical tools we have developed, and also as a stepping stone towards simulating more physical and realistic space physics applications.

APPENDICES

Appendix A

Resistive MHD Normalization

A.1 Dimensional Resistive MHD Equations

The dimensional resistive MHD equations are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0, \tag{A.1}$$

$$\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot \left(\rho \vec{v}\vec{v} + \vec{I}(p + \frac{\vec{B} \cdot \vec{B}}{2\mu_0}) - \frac{\vec{B}\vec{B}}{\mu_0} \right) = 0, \tag{A.2}$$

$$\frac{\partial \vec{B}}{\partial t} + \nabla \cdot (\vec{v}\vec{B} - \vec{B}\vec{v}) + \nabla \times (\eta \vec{J}) = 0, \tag{A.3}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot \left((e + p + \frac{\vec{B} \cdot \vec{B}}{2\mu_0})\vec{v} - \frac{(\vec{v} \cdot \vec{B})\vec{B}}{\mu_0} + \frac{\eta \vec{J} \times \vec{B}}{\mu_0} \right) = 0, \tag{A.4}$$

where the current, $\vec{J}$, is obtained by applying Ampere's circuital law (assuming a non-relativistic situation)

$$\vec{J} = \frac{\nabla \times \vec{B}}{\mu_0}. \tag{A.5}$$

The vacuum permeability constant, μ_0, assumes the value $4\pi(10^{-7})\frac{T\ m}{A}$ as written in SI units.

Pressure, p, and total energy, e, are related through the equation of state for a perfect gas [28]

$$e = \frac{p}{\gamma - 1} - \frac{\rho \vec{u}^2}{2} - \frac{\vec{B}^2}{2\mu_0}, \tag{A.6}$$

where γ is the ratio of specific heats, and is dimensionless, which means that both e and p are of the same units.

A.2 Normalization Parameters

To non-dimensionalize Eq. A.1 to Eq. A.4, we first need to non-dimensionalize all quantities that have units. Let us start with the four primitive quantities of the MHD equations: ρ, $\vec{v}$, p, and $\vec{B}$.

To accomplish that, we pick (or derive from physical values) the following reference quantities

- ρ_0: reference density,
- C_0: reference speed of sound,
- l_0: characteristic length,

from which we can derive other normalizing quantities as follows

$$p_0 = \rho_0 C_0^2, \tag{A.7}$$

$$B_0 = \sqrt{\rho_0 C_0^2 \mu_0}, \tag{A.8}$$

which are then used to normalize the primitive variables

$$\hat{\rho} = \frac{\rho}{\rho_0}, \tag{A.9}$$

$$\hat{\vec{v}} = \frac{\vec{v}}{C_0}, \tag{A.10}$$

$$\hat{p} = \frac{p}{p_0}, \tag{A.11}$$

$$\hat{\vec{B}} = \frac{\vec{B}}{B_0}. \tag{A.12}$$

These normalization techniques are consistent with the procedure outlined in [6].

Alternatively, we can choose C_0 to be the reference speed of sound by determining the reference temperature T_0, and applying the ideal gas law

$$C_0 = \sqrt{\frac{\gamma p_0}{\rho_0}} = \sqrt{\frac{\gamma \rho_0 R_0 T_0}{\rho_0}} = \sqrt{\gamma R_0 T_0}, \tag{A.13}$$

with R_0 being the specific gas constant, and T_0 the reference temperature. From first principles, we have the following expression to determine R_0

$$R_0 = \frac{k_B}{f m_p}, \tag{A.14}$$

where k_B is the Boltzmann constant ($1.3806488(10^{-23})\frac{J}{K}$), m_p is the molecular mass of a proton, and f is the mass fraction, used to account for the presence of electrons and other species in the gas mixture. The value of f for gas mixture is chosen to either be 0.6, or 0.5 (the latter was the same as in [123]). A proton's mass, m_p is $1.67262(10^{-27})kg$, so R_0 works out to be either $16,508.816\frac{J}{K-kg}$ for $f = 0.5$, or $13,757.347\frac{J}{K-kg}$ for $f = 0.6$. If we assume that there are only ions present, then $f = 1$, and $R_0 = 8,254.399\frac{J}{K-kg}$.

Thus, to complete normalization of the primitive variables, as mentioned, we pick ρ_0, C_0, and l_0. And also as mentioned, in some cases, picking the reference temperature, T_0, can be more convenient. After these three quantities are chosen, the normalization factors for pressure, which is p_0, and magnetic field, which is B_0, can be conveniently derived.

We use l_0 to non-dimensionalize the ∇ operator as follows

$$\hat{\nabla} = l_0 \nabla. \tag{A.15}$$

And to normalize time, we define t_0, which is related to the characteristic length, l_0, and the reference speed, C_0, through the following relationship

$$t_0 = \frac{l_0}{C_0}. \tag{A.16}$$

The normalized time quantity, $\hat{t}$, can then be defined as

$$\hat{t} = \frac{t}{t_0}. \tag{A.17}$$

Also, to non-dimensionalize current, we need to define $\hat{\vec{J}}$, such that

$$\hat{\vec{J}} = \hat{\nabla} \times \hat{\vec{B}}, \tag{A.18}$$

and this requires that we normalize $\vec{J}$ by some J_0, which we now derive. We rewrite Eq. A.5 in terms of non-dimensional variables and operators as follows

$$\vec{J} = \frac{\sqrt{\mu_0 C_0^2 \rho_0}}{l_0 \mu_0} \hat{\nabla} \times \hat{\vec{B}}, \tag{A.19}$$

which suggests we need to take

$$J_0 = \frac{\sqrt{\mu_0 C_0^2 \rho_0}}{l_0 \mu_0}, \tag{A.20}$$

to arrive at Eq. A.18, where

$$\hat{\vec{J}} = \frac{\vec{J}}{J_0}. \tag{A.21}$$

Lastly, the resistivity, η too, needs to be non-dimensionalized. Performing dimensional analysis, the appropriate normalizing quantity for η can be defined as

$$\eta_0 = C_0 l_0 \mu_0, \tag{A.22}$$

where we can define non-dimensional resistivity, $\hat{\eta}$, as

$$\hat{\eta} = \frac{\eta}{\eta_0}. \tag{A.23}$$

A.3 Normalization of the Resistive MHD Equations

A.3.1 Normalization of the Continuity Equation

To normalize the continuity equation, we first rewrite all of the dimensional quantities in terms of the normalized quantities,

$$\frac{\rho_0}{t_0}\frac{\partial \hat{\rho}}{\partial \hat{t}} + \frac{\rho C_0}{l_0}\hat{\nabla}\cdot(\hat{\rho}\hat{\vec{v}}) = 0. \tag{A.24}$$

Here, we can write $\frac{l_0}{C_0} = t_0$, to obtain

$$\frac{\rho_0}{t_0}\left(\frac{\partial \hat{\rho}}{\partial \hat{t}} + \hat{\nabla}\cdot(\hat{\rho}\hat{\vec{v}})\right) = 0, \tag{A.25}$$

which, after further simplification, leads to the normalized continuity equation as follows

$$\frac{\partial \hat{\rho}}{\partial \hat{t}} + \hat{\nabla}\cdot(\hat{\rho}\hat{\vec{v}}) = 0, \tag{A.26}$$

which has exactly the same form as the dimensional continuity equation (Eq. A.1).

A.3.2 Normalization of the Momentum Equation

Beginning with Eq. A.2, we can rewrite dimensional quantities in terms of their normalized quantities,

$$\frac{\rho_0 C_0}{t_0}\frac{\partial(\hat{\rho}\hat{\vec{v}})}{\partial \hat{t}} + \frac{\rho_0 C_0^2}{l_0}\hat{\nabla}\cdot\left(\hat{\rho}\vec{v}\vec{v} + \vec{I}(\hat{p} + \frac{\hat{\vec{B}}\cdot\hat{\vec{B}}}{2}) - \frac{\hat{\vec{B}}\hat{\vec{B}}}{2}\right) = 0, \tag{A.27}$$

which, after setting $\frac{l_0}{C_0} = t_0$, can be simplified further into

$$\frac{\rho_0 C_0}{t_0}\left[\frac{\partial(\hat{\rho}\hat{\vec{v}})}{\partial \hat{t}} + \hat{\nabla}\cdot\left(\hat{\rho}\vec{v}\vec{v} + \vec{I}(\hat{p} + \frac{\hat{\vec{B}}\cdot\hat{\vec{B}}}{2}) - \frac{\hat{\vec{B}}\hat{\vec{B}}}{2}\right)\right] = 0. \tag{A.28}$$

Finally, we arrive at the following normalized momentum equation for resistive MHD

$$\frac{\partial(\hat{\rho}\hat{\vec{v}})}{\partial\hat{t}} + \hat{\nabla}\cdot\left(\hat{\rho}\vec{v}\vec{v} + \vec{I}(\hat{p} + \frac{\hat{\vec{B}}\cdot\hat{\vec{B}}}{2}) - \frac{\hat{\vec{B}}\hat{\vec{B}}}{2}\right) = 0. \tag{A.29}$$

Note that the normalized momentum equation for resistive MHD is exactly the same as that of the ideal MHD (Eq. 2.2), as there is not any dependence on the resistive term.

A.3.3 Normalization of the Energy Equation

To non-dimensionalize the energy equation, we begin with the dimensional form of the equation, Eq. A.4. To normalize the total energy, e, we use the same quantity as we would for p, which is $p_0 = \rho_0 C_0^2$.

By substituting the non-dimensional quantities, and simplifying algebraically, we derive the following expressions

$$\frac{\rho_0 C_0^2}{t_0}\left(\frac{\partial\hat{e}}{\partial\hat{t}} + \hat{\nabla}\cdot\left((\hat{e} + \hat{p} + \frac{\hat{\vec{B}}\cdot\hat{\vec{B}}}{2})\hat{\vec{v}} - (\hat{\vec{v}}\cdot\hat{\vec{B}})\hat{\vec{B}} + \hat{\eta}\hat{\vec{J}}\times\hat{\vec{B}}\right)\right) = 0, \tag{A.30}$$

which, after further simplification, leads to the normalized energy equation as follows

$$\frac{\partial\hat{e}}{\partial\hat{t}} + \hat{\nabla}\cdot\left((\hat{e} + \hat{p} + \frac{\hat{\vec{B}}\cdot\hat{\vec{B}}}{2})\hat{\vec{v}} - (\hat{\vec{v}}\cdot\hat{\vec{B}})\hat{\vec{B}} + \hat{\eta}\hat{\vec{J}}\times\hat{\vec{B}}\right) = 0. \tag{A.31}$$

Note the similarity between the two forms (dimensional and non-dimensional), and the fact that Eq. A.31 no longer contains any dependence on μ_0.

A.3.4 Normalization of the Magnetic Field Equation

We start from Eq. A.3, which is just the induction equation for resistive MHD. Substituting non-dimensional quantities for the corresponding dimensional quantities gives us the following expression

$$\frac{\sqrt{\rho_0 C_0^2\mu_0}}{t_0}\frac{\partial\hat{\vec{B}}}{\partial\hat{t}} + \frac{1}{l_0}C_0\sqrt{\rho_0 C_0^2\mu_0}\hat{\nabla}\cdot(\hat{\vec{v}}\hat{\vec{B}} - \hat{\vec{B}}\hat{\vec{v}}) + \frac{1}{l_0}C_0 l_0\mu_0\frac{\sqrt{\rho_0 C_0^2\mu_0}}{l_0\mu_0}\hat{\nabla}\times(\hat{\eta}\hat{\vec{J}}) = 0. \tag{A.32}$$

After some algebraic manipulations, we can show that

$$\frac{\sqrt{\rho_0 C_0^2\mu_0}}{t_0} = \frac{1}{l_0}C_0\sqrt{\rho_0 C_0^2\mu_0} = \frac{1}{l_0}C_0 l_0\mu_0\frac{\sqrt{\rho_0 C_0^2\mu_0}}{l_0\mu_0}, \tag{A.33}$$

which allows for significant simplification of the non-dimensionalized induction equation equation

$$\frac{\partial \hat{\vec{B}}}{\partial \hat{t}} + \hat{\nabla} \cdot (\hat{\vec{v}}\hat{\vec{B}} - \hat{\vec{B}}\hat{\vec{v}}) + \hat{\nabla} \times (\hat{\eta}\hat{\vec{J}}) = 0. \tag{A.34}$$

Note that Eq. A.34 follows the exact same form as the non-dimensionalized version, Eq. A.3, except that we now no longer need to deal with μ_0 to convert the magnetic field to current. The normalized current, $\hat{\vec{J}}$, is exactly just the curl of the normalized magnetic field, $\hat{\vec{B}}$.

A.4 Non-Dimensional Resistive MHD Equations

We can summarize the non-dimensional resistive MHD equations as follows

$$\frac{\partial \hat{\rho}}{\partial \hat{t}} + \hat{\nabla} \cdot (\hat{\rho}\hat{\vec{v}}) = 0, \tag{A.35}$$

$$\frac{\partial (\hat{\rho}\hat{\vec{v}})}{\partial \hat{t}} + \hat{\nabla} \cdot \left(\hat{\rho}\hat{\vec{v}}\hat{\vec{v}} + \vec{I}(\hat{p} + \frac{\hat{\vec{B}} \cdot \hat{\vec{B}}}{2}) - \hat{\vec{B}}\hat{\vec{B}} \right) = 0, \tag{A.36}$$

$$\frac{\partial \hat{\vec{B}}}{\partial \hat{t}} + \hat{\nabla} \cdot (\hat{\vec{v}}\hat{\vec{B}} - \hat{\vec{B}}\hat{\vec{v}}) + \hat{\nabla} \times (\hat{\eta}\hat{\vec{J}}) = 0, \tag{A.37}$$

$$\frac{\partial \hat{e}}{\partial \hat{t}} + \hat{\nabla} \cdot \left((\hat{e} + \hat{p} + \frac{\hat{\vec{B}} \cdot \hat{\vec{B}}}{2})\hat{\vec{v}} - (\hat{\vec{v}} \cdot \hat{\vec{B}})\hat{\vec{B}} + \hat{\eta}\hat{\vec{J}} \times \hat{\vec{B}} \right) = 0, \tag{A.38}$$

and the non-dimensional Ampere's circuital law to make the connection between the normalized current, $\hat{\vec{J}}$, and the non-dimensional magnetic field, $\hat{\vec{B}}$, is given by

$$\hat{\vec{J}} = \hat{\nabla} \times \hat{\vec{B}}. \tag{A.39}$$

These equations look very similar to their dimensional counterparts shown in Sec. A.1, except for the obvious absences of μ_0.

A.5 Normalization of Solar Wind Parameters

Solar wind parameters for simulations done in this paper are given by Eq. 6.1 to Eq. 6.4. Keep in mind that we will need to convert these values to ones that are normalized ($\hat{\rho}$, $\hat{\vec{v}}$, $\hat{p}$, $\hat{\vec{B}}$), following the normalization mechanism given by Eq. A.9 to Eq. A.12.

To complete normalization, we choose reference number density, n_0, the reference temperature, T_0, and the reference length, l_0. Because the solar wind simulations are conducted within the context of the Moon, we choose the following reference parameters

$$n_0 = 142,857.14286\ /m^3, \tag{A.40}$$

$$T_0 = 140,000\ K, \tag{A.41}$$

$$l_0 = 1,737.10\ km, \tag{A.42}$$

where the reference density, ρ_0, can be obtained by the following relation

$$\rho_0 = n_0\ m_p. \tag{A.43}$$

Here, m_p is the mass of a proton, which is $1.67262(10^{-27})kg$.

Using the above reference values, and following Eq. A.7 - Eq. A.13, we obtain with the following normalized (unitless) solar wind parameters

$$\hat{\rho} = 7.0, \tag{A.44}$$

$$\hat{\vec{v}} = 9.94464, \tag{A.45}$$

$$\hat{B_z} = 7.17372, \tag{A.46}$$

$$\hat{p} = 5.0. \tag{A.47}$$

And as mentioned in Sec. 6, the resistivity of the moon is chosen to be $\eta = 10,000\ \Omega\ m$. Using Eq. A.23 to normalize η, we have $\hat{\eta} = 0.11389$.

As for boundary conditions, as mentioned in Sec. 6.2, the dayside part of the Lunar surface is set to a density of $10^5\ /m^3$ and the nightside to $10^4\ /m^3$. In normalized terms, they work out to be

$$\hat{\rho}_{dayside} = 0.7, \tag{A.48}$$

$$\hat{\rho}_{nightside} = 0.07. \tag{A.49}$$

Likewise, given that the surface temperature is set at 400 K, the normalized dayside and nightside pressures are given as

$$\hat{p}_{dayside} = 0.00142857142857, \tag{A.50}$$

$$\hat{p}_{nightside} = 0.000142857142857. \tag{A.51}$$

Appendix B

Generation of the Inner Cubed Sphere

The regular cubed-sphere mesh consists of six outer shell blocks, each of which can be obtained through mapping from a Cartesian computational mesh. The outer shell is a collection of concentric spheres with radii ranging from $r = R_i$ to $r = R_o$, each value of grid index k corresponding to a discretized concentrically spherical layer. We can see how this arrangement would, by construction, leave the mesh with a core that is hollow (unless we set $R_i = 0$, which would lead to a very skewed grid). Fig. B.1a illustrates how a cubed-sphere sector forms a collection of concentric spheres, and six of these sectors would then form the complete cubed-sphere mesh (see Fig. 1.2), which represents a hollow sphere.

As mentioned in Sec. 4.1.5, a nested sphere configuration may be required in some cases where it is required to model the diffusion of the magnetic field as it passes through a lunar or a planetary object. In fact, several of such problems have been considered in Chapter 6, where simulations of conditions around the lunar atmosphere have been conducted. As the regular cubed-sphere mesh produces only a hollow sphere, then to also volumetrically fill the middle, a mechanism is considered, which gradually transforms radial spherical projection for the outer sphere to a more Cartesian-like projection as it gets closer to the center (see Fig. B.2). This technique has been adapted from the ball generation radial projection method from Sec. 5.2 of [94], which is itself an extension of the radial projection mapping scheme (Sec. 3.1- 3.3 from [94]). The idea in 2D is to transform a Cartesian square grid into a circle through radial projection. With this mapping, a concentric square in the computational domain maps onto a concentric circle in the physical domain (see Fig. 3.1 of [94]).

A solid sphere (or a ball) can now be created by extending the aforementioned radial projection mapping scheme to three dimensions. An equi-angular gnomonic projection is employed to create a spherical surface. The equi-angular projection is based on the uniform discretization of the two angular parameters (denoted by ξ and η in [12]). Based on ξ and

η, values of the x_c, y_c, and z_c coordinates of the computational domain can be determined, which are then transformed to the physical x_p, y_p, and z_p coordinates. Eq. $1-14$ of [12] provides the mappings from the computational Cartesian domain to the cubed-sphere physical domain, and Sec. 5.2 of [94] provides interpolation and weighting mechanisms to transition from purely equi-angular gnomonic cubed-sphere projection for the outer part of the sphere onto more Cartersian-like close to the center. So we can write the mapping from computational coordinates, x_c, y_c, and z_c to physical coordinates, x_p, y_p, and z_p as [94]

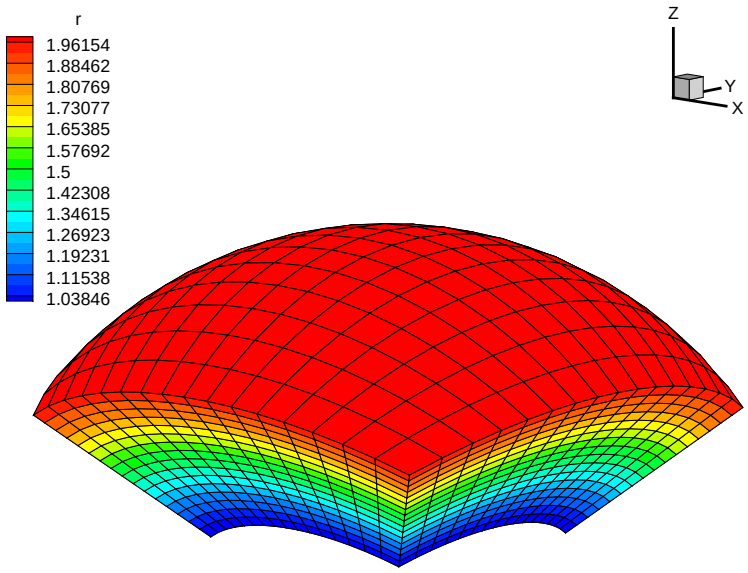
$$x_p = R\frac{d^3}{r}x_c + \frac{1-d^2}{\sqrt{3}}x_c, \tag{B.1}$$

$$y_p = R\frac{d^3}{r}y_c + \frac{1-d^2}{\sqrt{3}}y_c, \tag{B.2}$$

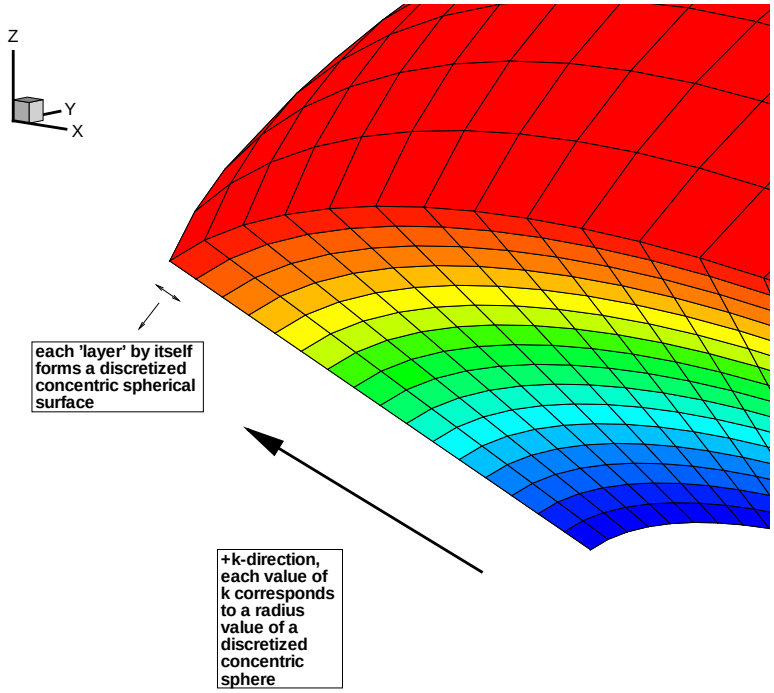
$$z_p = R\frac{d^3}{r}z_c + \frac{1-d^2}{\sqrt{3}}z_c, \tag{B.3}$$

where $r = \sqrt{x_c^2 + y_c^2 + z_c^2}$, $d = \mathbf{max}(x_c, y_c, z_c)$, and R is the radius of the sphere. The mapping as defined by Eq. B.1 to Eq. B.3 is then effectively a mapping from a cubic computational domain, with a domain of $[-1,1]\times[-1,1]\times[-1,1]$, to a solid sphere with a radius of R. This transition is illustrated by Fig. B.2.

The outer part of this solid sphere has coordinates that exactly match those of the concentric spheres that constitute our regular cubed sphere. We can now "encapsulate" the inner sphere with a regular cubed-sphere grid to produce a nested solid sphere, which consists of the six regular cubed-sphere sectors and a solid sphere to fill in the void in the middle (see Fig. B.3). The advantage of the 7-root-block nested sphere arrangement over just a single completely solid sphere (Fig. B.2), comes from the point of view of the implementation of the boundary conditions, where interior boundary conditions may be applied directly at the interface connecting the outer shell and the inner sphere, which is relevant especially to the space physics applications considered in Chapter 6. It allows for clear differentiation of the moon or a planetary body (which is represented by the inner sphere), and the domain surrounding the body (represented by the outer shells). This set-up is illustrated in Fig. 6.1, which is the set-up used for all the simulations related to the interaction between the lunar atmosphere and the solar wind.



(a) A sector of the regular cubed-sphere grid. Note that the sector is comprised of several discretized concentric spheres with radii ranging from R_i to R_o.



(b) A blow-up view of a sector of a regular cubed-sphere grid. Each radial layer represents a discretized concentric sphere.

Figure B.1: Three-dimensional view of a sector of the cubed-sphere grid.

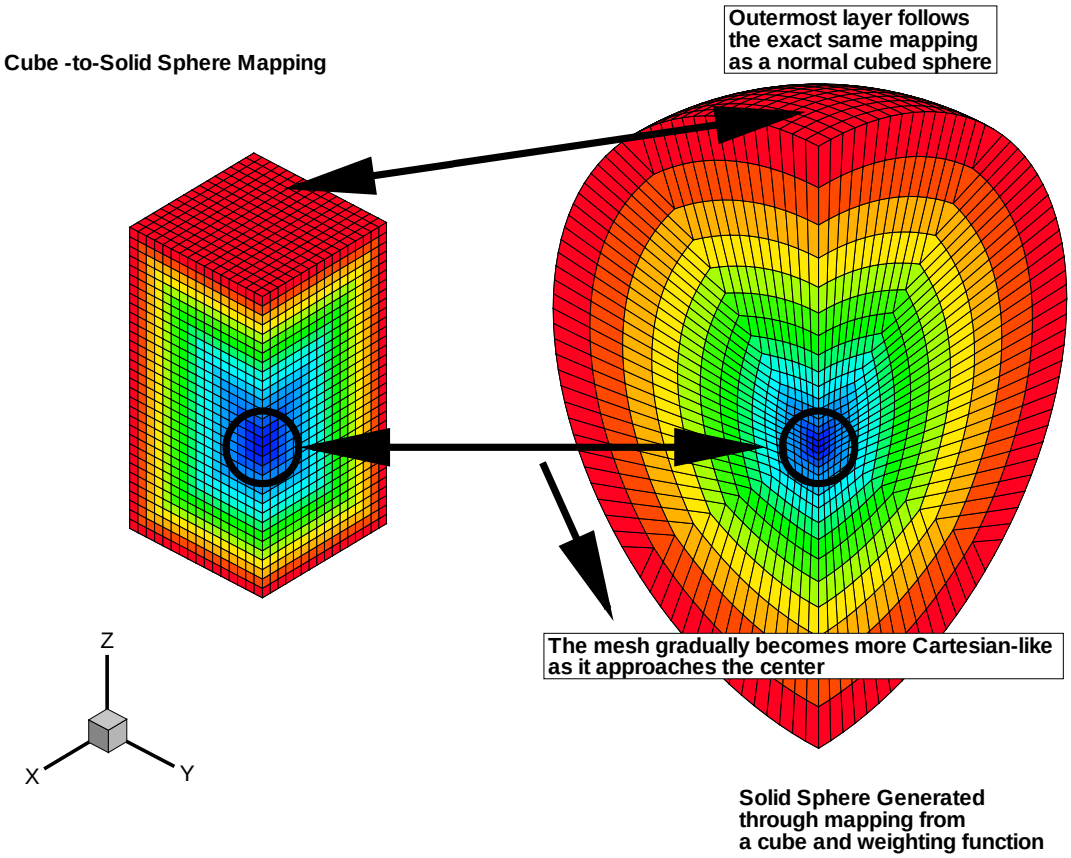


Figure B.2: A cut-away view of the inner sphere, to illustrate how the grid transitions from purely spherical (based on equi-angular gnomonic projection) on the outer edge to purely Cartesian as it gets closer to the center. The mapping here too is illustrated where each colored layer on the computational domain (left) maps onto the same colored layer on the physical domain (right).

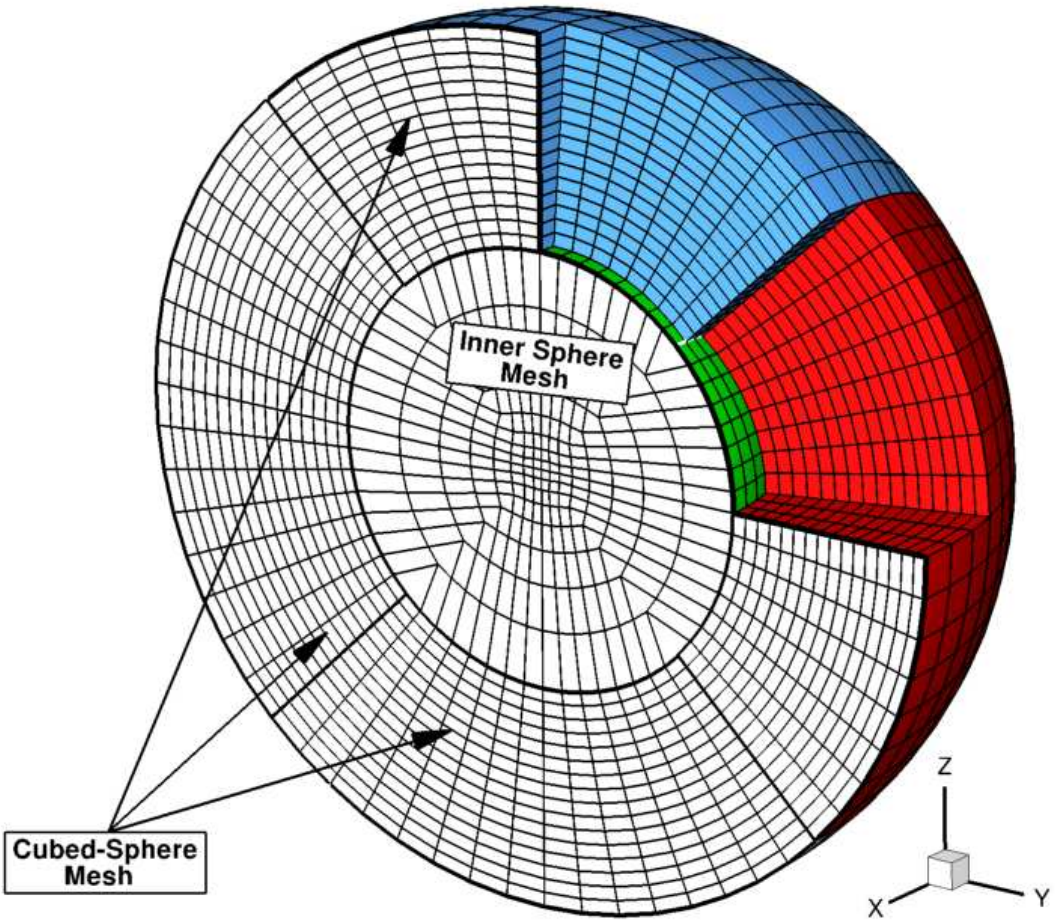


Figure B.3: Three-dimensional view of the cross section of the nested sphere for $x <= 0$. By construction, the outer shells would necessarily leave empty space in the middle (since they are just layers of concentric spheres, the only way one can have a complete sphere if a sphere with a zero radius is defined), so an inner sphere needs to be constructed in a way that will fill this void. The outer shells would then encapsulate the inner sphere, and it would then produce a solid nested sphere (Figure adapted from [3])

References

[1] A. Susanto, L. Ivan, H. De Sterck, and C. P. T. Groth. High-order central ENO finite-volume scheme for ideal MHD. *J. Comput. Phys.*, 250:141–164, 2013.

[2] L. Ivan, A. Susanto, H. De Sterck, and C. P. T Groth. High-order central ENO finite-volume scheme for MHD on three-dimensional cubed-sphere grids. In *Proceedings of the Seventh International Conference on Computational Fluid Dynamics*, June 2012.

[3] L. Ivan, H. De Sterck, A. Susanto, and C. P. T. Groth. High-order central ENO finite-volume scheme for conservation laws on three-dimensional cubed-sphere grids. 2014. Submitted to Journal of Computational Physics.

[4] L. Ivan and C. P. T. Groth. High-order Central ENO finite-volume scheme with adaptive mesh refinement. Paper 2007-4323, AIAA, June 2007.

[5] P. A. Davidson. *An Introduction to Magnetohydrodynamics*. Cambridge University Press, New York, NY, USA, 2001.

[6] C. P. T. Groth, D. L. De Zeeuw, T. I. Gombosi, and 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):25,053–25,078, 2000.

[7] C. R. Clauer et al. High performance computer methods applied to predictive space weather simulations. *IEEE Transactions on Plasma Science*, 28:1931–1937, 2000.

[8] L. Ivan, H. De Sterck, S. A. Northrup, and C. P. T. Groth. Hyperbolic conservation laws on three-dimensional cubed-sphere grids: A parallel solution-adaptive simulation framework. *J. Comput. Phys.*, 255:205–227, 2013.

[9] N. A. Phillips. Numerical integration of the primitive equations on the hemisphere. *Monthly Weather Review*, pages 333–345, 1959.

[10] R. Sadourny. Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. *Mon. Weather Rev.*, 100(2):136–144, 1972.

[11] G. L. Browning, J. J. Hack, and P. N. Swarztrauber. A comparison of three numerical methods for solving differential equations on the sphere. *Monthly Weather Review*, 117:10581075, 1989. http://dx.doi.org/10.1175/1520-0493(1989)117¡1058:ACOTNM¿2.0.CO;2.

[12] C. Ronchi, R. Iacono, and P. S. Paolucci. The "cubed sphere": A new method for the solution of partial differential equations in spherical geometry. *J. Comput. Phys.*, 124:93–114, 1996.

[13] A. Adcroft, J.-M. Campin, C. Hill, and J. Marhall. Implementation of an atmosphere-ocean general circulation model on the expanded spherical cube. *Mon. Weather Rev.*, 132:2845–2863, 2004.

[14] R. D. Nair, S. J. Thomas, and R. D. Loft. A discontinuous Galerkin transport scheme on the cubed sphere. *Mon. Weather Rev.*, 133:814–828, 2005.

[15] W. M. Putman and S.-J. Lin. Finite-volume transport on various cubed-sphere grids. *J. Comput. Phys.*, 227:55–78, 2007.

[16] A. St-Cyr, C. Jablonowski, J. M. Dennis, H. M. Tufo, and S. J. Thomas. A comparison of two shallow-water models with nonconforming adaptive grids. *Mon. Weather Rev.*, 136:1898–1922, 2008.

[17] C. Chen and F. Xiao. Shallow water model on cubed-sphere by multi-moment finite volume method. *Journal of Computational Physics*, 227(10):5019–5044, 2008.

[18] C. Yang, J. Cao, and X.-C. Cai. A fully implicit domain decomposition algorithm for shallow water equations on the cubed-sphere. *SIAM J. Sci. Comput.*, 32:418–438, 2010.

[19] Paul A. Ullrich, Christiane Jablonowski, and Bram van Leer. High-order finite-volume methods for the shallow-water equations on the sphere. *J. Comput. Phys.*, 229(17):6104 – 6134, 2010. DOI: 10.1016/j.jcp.2010.04.044.

[20] C. Chen, F. Xiao, and X. Li. An adaptive multimoment global model on a cubed sphere. *Monthly Weather Review*, 139(2):523–548, 2011.

[21] P. A. Ullrich and C. Jablonowski. MCore: A non-hydrostatic atmospheric dynamical core utilizing high-order finite-volume methods. *Journal of Computational Phsysics*, 231:50785108, 2012.

[22] J. A. Rossmanith. An unstaggered, high-resolution constained transport method for magnetohydrodynamic flows. *Journal of Computational Physics*, 213(2):629–658, 2006.

[23] A. V. Koldoba, M. M. Romanova, G. V. Ustyugova, and R. V. E. Lovelace. Three-dimensional magnetohydrodynamic simulations of accretion to an inclined rotator: the "cubed sphere" method. *Astrophys. J.*, 576:L53–L56, 2002.

[24] P. C. Fragile, C. C. Lindner, P. Anninos, and J. D. Salmonson. Application of the cubed-sphere grid to tilted black hole accretion disks. *Astrophys. J.*, 691:482–494, 2009.

[25] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes: II. *J. Comput. Phys.*, 83:32–78, 1989.

[26] J. U. Brackbill and D. C. Barnes. The effect of nonzero $\nabla\cdot\mathbf{B}$ on the numerical solution of the magnetohydrodynamic equations. *J. Comput. Phys.*, 35:426–430, 1980.

[27] L. Ivan and C. P. T. Groth. High-order solution adaptive central essentially non-oscillatory (CENO) method for viscous flows. *J. Comput. Phys.*, 257:830–862, 2014.

[28] A. Dedner, F. Kemm, D. Kroner, C-D. Munz, T. Schitzer, and M. Wesenberg. Hyberbolic divergence cleaning for the MHD equations. *J. Comput. Phys.*, 175:645–673, 2002.

[29] G. Tóth. The $\nabla\cdot\mathbf{B}=\mathbf{0}$ constraint in shock-capturing magnetohydrodynamics codes. *J. Comput. Phys.*, 161(2):605–652, 2000.

[30] K. G. Powell, P. L. Roe, T. J. Linde, T. I. Gombosi, and D. L. De Zeeuw. A solution-adaptive upwind scheme for ideal magnetohydrodynamics. *J. Comput. Phys.*, 154:284–309, 1999.

[31] C. R. Evans and J. F. Hawley. Simulation of magnetohydrodynamic flows: A constrained transport method. *Astrophys. J.*, 332:659, 1988.

[32] W. Dai and P. R. Woodward. Flux-corrected transport techniques for multidimensional compressible magnetohydrodynamics. *J. Comput. Phys.*, 142:331, 1998.

[33] D. Ryu, F. Miniati, T. W. Jones, and A. Frank. A divergence-free upwind code for multi-dimensional magnetohydrodynamic flows. *Astrophys. J.*, 509:244, 1998.

[34] H. De Sterck. Multi-dimensional upwind constrained transport on unstructured grids for 'shallow water' magnetohydrodynamics. *AIAA*, 2001-2623, 2001.

[35] D. S. Balsara and D. S. Spicer. staggered mesh algorithm using high order godunov fluxes to ensure solenoidal magnetic fields in magnetohydrodynamic simulations. *J. Comput. Phys.*, 149:270, 1999.

[36] S. K. Godunov. Symmetric form of the equations of magnetohydrodynamics. *Numerical Methods for Mechanics of Continuum Medium,* Siberian Branch of USSR Academy of Sciences, 1:26–34, 1972.

[37] C. D. Munz, P. Omnes, R. Schneider, E. Sonnendrücker, and U. Voss. Divergence correction techniques for Maxwell solvers based on a hyperbolic model. *Journal of Computational Physics*, 161:484, 2000.

[38] A. Mignone and P. Tzeferacos. A second-order unsplit Godunov scheme for cell-centered MHD: The CTU-GLM scheme. *Journal of Computational Physics*, 229:2117–2138, 2010.

[39] A. Mignone, P. Tzeferacos, and G. Bodo. High-order conservative finite difference GLM-MHD schemes for cell-centered MHD. *Journal of Computational Physics*, 229:5896–5920, 2010.

[40] C. Hirsch. *Numerical Computation of Internal and External Flows, Volume 2, Computational Methods for Inviscid and Viscous Flows.* John Wiley & Sons, Toronto, 1990.

[41] M. S. Yalim, D. Vanden Abeele, A. Lani, T. Quintino, and H. Deconinck. A finite volume implicit time integration method for solving the equations of ideal magnetohydrodynamics for the hyperbolic divergence cleaning method. *J. Comput. Phys.*, 230:6136–6154, 2011.

[42] T. C. Warburton and G. E. Karniadakis. A discontinuous Galerkin method for the viscous MHD equations. *J. Comput. Phys.*, 152:608–641, 1999.

[43] F. Li and C.-W. Shu. Locally divergence-free discontinuous Galerkin methods for MHD equations. *Journal of Scientific Computing*, 22-23:413–442, 2005.

[44] C. Altmann, Thomas Belat, Michael Gutnic, Philippe Helluy, Helene Mathis, Eric Sonnendruecker, Wilfredo Angulo, and Jean-Marc Herard. A local time-stepping discontinuous Galerkin algorithm for the MHD system. In *ESAIM: PROCEEDINGS*, volume 28, pages 33–54. EDP Sciences, August 2009.

[45] F. Li, L. Xu, and S. Yakovlev. Central discontinuous Galerkin methods for ideal MHD equations with the exactly divergence-free magnetic field. *Journal of Computational Physics*, 230:4828–4847, 2011.

[46] D. S. Balsara and J. Kim. A comparison between divergence-cleaning and staggered-mesh formulations for numerical magnetohydrodynamics. *Astrophys. J.*, 602:1079–1090, 2004.

[47] G-S. Jiang and C. Wu. A high-order WENO finite difference scheme for the equations of ideal magnetohydrodynamics. *Journal of Computational Physics*, 150:561–594, 1999.

[48] J. Kleimann, A. Kopp, H. Fichtner, R. Grauer, and K. Germaschewski. Three-dimensional MHD high-resolution computations with CWENO employing adaptive mesh refinement. *Computer Physics Communications*, 158:47–56, 2004.

[49] J. Balbás and E. Tadmor. Non-oscillatory central schemes for one- and two-dimensional MHD equations. II: High-order semi-discrete schemes. *SIAM Journal of Scientific Computing*, 28:533–560, 2006.

[50] D. S. Balsara. Divergence-free reconstruction of magnetic fields and WENO schemes for magnetohydrodynamics. *J. Comput. Phys.*, 228:5040–5056, 2009.

[51] T. J. Barth. Recent developments in high order k-exact reconstruction on unstructured meshes. Paper 93-0668, AIAA, January 1993.

[52] L. Ivan. *Development of High-Order CENO Finite-Volume Schemes with Block-Based Adaptive Mesh Refinement*. PhD thesis, University of Toronto, October 2010.

[53] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.*, 49:357–393, 1983.

[54] A. Harten and S. R. Chakravarthy. Multi-dimensional ENO schemes for general geometries. Report 91-76, ICASE, September 1991.

[55] H. Nessyahu and E. Tadmor. Non-oscillatory central differencing for hyperbolic conservation laws. *Journal of Computational Physics*, 87:408, 1990.

[56] S. Orszag and C. M. Tang. Small-scale structure of two-dimensional magnetohydrodynamic turbulence. *Journal of Fluid Mechanics*, 90:129–143, 1979.

[57] C. P. T. Groth and S. A. Northrup. Parallel implicit adaptive mesh refinement scheme for body-fitted multi-block mesh. Paper 2005-5333, AIAA, June 2005.

[58] J. S. Sachdev, C. P. T. Groth, and 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), 2005.

[59] X. Gao and C. P. T. Groth. A parallel adaptive mesh refinement algorithm for predicting turbulent non-premixed combusting flows. *Int. J. Comput. Fluid Dyn.*, 20(5):349–357, 2006.

[60] X. Gao, S. A. Northrup, and C. P. T. Groth. Parallel solution-adaptive method for two-dimensional non-premixed combusting flows. *Progress in Computational Fluid Dynamics*, 11(2):76–95, 2011.

[61] B. van Leer, W-T. Lee, P. L. Powell, and K. G. Powell and. Towards the ultimate conservative difference scheme. v. a second-order sequel to Godunov's method. *J. Comput. Phys.*, 32:101–136, 1979.

[62] D. J. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. Paper 2003-3986, AIAA, June 2003.

[63] V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions. Paper 93-0880, AIAA, January 1993.

[64] L. Ivan and C. P. T. Groth. High-order solution-adaptive central essentially non-oscillatory (CENO) method for viscous flows. Paper 2011-0367, AIAA, January 2011.

[65] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.

[66] B. Einfeldt. On Godunov-type methods for gas dynamics. *SIAM J. Numer. Anal.*, 25:294–318, 1988.

[67] V. Wheatley, H. Kumar, and P. Huguenot. On the role of Riemann solvers in discontinuous Galerkin methods for magnetohydrodynamics. *Journal of Computational Physics*, 229:660–680, 2010.

[68] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, Cambridge, 2002.

[69] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506–517, 1968.

[70] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI*. MIT Press, Cambridge, Massachussets, 1999.

[71] W. Gropp, E. Lusk, and R. Thakur. *Using MPI-2*. MIT Press, Cambridge, Massachussets, 1999.

[72] C. F. Ollivier-Gooch and M. Van Altena. A high-order accurate unstructured mesh finite-volume scheme for the advection-diffusion equation. *J. Comput. Phys.*, 181(2):729–752, 2002.

[73] H. De Sterck, A. Csík, D. Vanden Abeele, S. Poedts, and H. Deconinck. Stationary two-dimensional magnetohydrodynamic flows with shocks: Characteristic analysis and grid convergence study. *J. Comput. Phys.*, 166:28–62, 2001.

[74] M. Brio and C. C. Wu. An upwind differencing scheme for the equations of ideal magnetohydrodynamics. *J. Comput. Phys.*, 75:400–422, 1988.

[75] J. A. Rossmanith. An unstaggered, high-resolution constained transport method for magnetohydrodynamic flows. *SIAM Journal on Scientific Computing*, 28:1766–1797, 2006.

[76] J. Balbás, E. Tadmor, and Cheng-Chin Wu. Non-oscillatory central schemes for one- and two-dimensional MHD equations: I. *Journal of Computational Physics*, 201:261–285, 2004.

[77] B. Cockburn and C-W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems.

[78] H. Luo, Y. Xia, S. Li, R. Nourgaliev, and C. Cai. A Hermite WENO reconstruction-based discontinuous Galerkin method for the euler equations on tetrahedral grids. *Journal of Computational Physics*, 231(16):5489–5503, 2012.

[79] A. Nejat and C. F. Ollivier-Gooch. A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows. *J. Comput. Phys.*, 227:2582–2609, 2008.

[80] D. S. Balsara, T. Rumpf, M. Dumbser, and C-D. Munz. Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics. *J. Comput. Phys.*, 228:2480–2516, 2009.

[81] Dinshaw S. Balsara. Self-adjusting, positivity preserving high order schemes for hydrodynamics and magnetohydrodynamics. *J. Comput. Phys.*, 231(22):7504 – 7517, 2012.

[82] M. Dumbser, D. Balsara, E. F. Toro, and C-D. Munz. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227(18):8209–8253, 2008.

[83] M. R. J. Charest, C. P. T. Groth, and P. Q. Gauthier. High-order CENO finite-volume scheme for low-speed viscous flows on three-dimensional unstructured mesh. In *Proceedings of the 7th International Conference on Computational Fluid Dynamics, Hawaii, USA, July 9–13*, 2012. Paper ICCFD7-1002.

[84] G. Tóth, B. van der Holst, I. V. Sokolov, D. L. De Zeeuw, T. I. Gombosi, F. Fang, W. B. Manchester, X. Meng, D. Najib, K. G. Powell, Q. F. Stout, A. Glocer, Y.-J. Ma, and M. Opher. Adaptive numerical algorithms in space weather modeling. *J. Comput. Phys.*, In Press, Corrected Proof, 2011. doi:10.1016/j.jcp.2011.02.006.

[85] B. van der Holst and R. Keppens. Hybrid block-amr in cartesian and curvilinear coordinates: MHD applications. *J. Comput. Phys.*, 226(1):925 – 946, 2007.

[86] R. Keppens, Z. Meliani, A. J. van Marle, P. Delmont, A. Vlasis, and B. van der Holst. Parallel, grid-adaptive approaches for relativistic hydro and magnetohydrodynamics. *J. Comput. Phys.*, 231(3):718–744, 2012.

[87] L. Ivan and C. P. T. Groth. High-order solution adaptive central essentially non-oscillatory (CENO) method for viscous flows. 2012. Submitted to Journal of Computational Physics.

[88] L. Ivan and C. P. T. Groth. High-order central eno scheme with adaptive mesh refinement for hyperbolic conservation laws. *Communications in Computational Physics*, Submitted, 2013.

[89] C.L. Lawson and R.J. Hanson. *Solving least squares problems*. Prentice-Hall, INC, 1974.

[90] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*, volume 1. Butterworth-Heinemann, fifth edition, 2000.

[91] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. *Numerical Grid Generation—Foundations and Applications*. North-Holland, New York, 1985.

[92] C. A. Felippa. A compendium of FEM integration formulas for symbolic work. *Eng. Comput.*, 21(8):867–890, 2004.

[93] Erika M Harnett and Robert Winglee. Two-dimensional MHD simulation of the solar wind interaction with magnetic field anomalies on the surface of the moon. *Journal of Geophysical Research: Space Physics*, 105(A11):24997–25007, 2000.

[94] D. A. Calhoun, C. Helzel, and R. J. LeVeque. Logically rectangular finite volume grids and methods for circular and spherical domains. *SIAM Review*, 50:723–752, 2008.

[95] W. Gander and W. Gautschi. Adaptive quadrature − revisited. *BIT*, 40:84–101, 2000.

[96] G. Gassner, F. Lörcher, and C. D. Munz. A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes. *J. Comput. Phys.*, 224:1049–1063, 2007.

[97] F. Lörcher, G. Gassner, and C. D. Munz. An explicit discontinuous Galerkin methods for diffusion problems. *J. Comput. Phys.*, 227:5649–5670, 2008.

[98] H. Liu and J. Yan. The directi discontinuous Galerkin (DDG) methods for diffusion problems. *SIAM J. Numer. Anal.*, 41(1):675–698, 2009.

[99] T. E. Cravens. *Physics of Solar System Plasmas*. Cambridge University Press, New York, 1997.

[100] P. J. Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa Publisher, New Mexico, 1998.

[101] Erika M Harnett and Robert Winglee. 2.5D particle and MHD simulation of mini-magnetospheres at the Moon. *Journal of Geophysical Research: Space Physics*, 107(A12):1421, 2002. http://www.dx.doi.org/10.1029/2002JA009241.

[102] Erika M Harnett and Robert Winglee. 2.5D MHD simulation of the solar wind interaction with multiple dipoles on the surface of the Moon. *Journal of Geophysical Research: Space Physics*, 108(A2):1088, 2003. http://www.dx.doi.org/10.1029/2002JA009617.

[103] F. S. Johnson. Lunar atmosphere.

[104] L. H. Xie, L. Li, Y. T. Zhang, and D. L. D. Zeeuw. Three-dimensional MHD simulation of the lunar wake. *Science China: Earth Sciences*, 56:330–338, 2012. http://link.springer.com/article/10.1007/s11430-012-4383-6.

[105] C. P. Sonnet. Solar wind induction and lunar conductivity.

[106] D. S. Colburn, J. D. Mihalov, and C. P. Sonett. Magnetic observations of the lunar cavity. *J. Geophys. Res.*, 76(13):2940–2957, 1971.

[107] M. Fuller and S. M. Cisowski. Lunar paleomagnetism. *Geomagnetism, Vol. 2, p. 307 - 455*, 2:307–455, 1987.

[108] R. P. Lin, D. L. Mitchell, D. W. Curtis, K. A. Anderson, C. W. Carlson, J. McFadden, M. H. Acu na, L. L. Hood, and A. Binder. Lunar surface magnetic fields and their interactions with the solar wind: Results from lunar prospector. 281:1480–1484, 1998.

[109] L. L. Hood, A. Zakharian, J. Halekas, D. L. Mitchell, R. P. Lin, M. H. Acuña, and A. B. Binder. Initial mapping and interpretation of lunar crustal magnetic anomalies using Lunar Prospector magnetometer data. *JGR*, 106(E11):27825–27839, 2001.

[110] X. Q. Wang et al. The solar wind interactions with lunar magnetic anomalies: A case study of the Chang'e-2 plasma data near the serenitatis antipode. *Advances in Space Research*, 50:1600–1606, 2012.

[111] J. S. Halekas, G. T. Delory, D. A. Brain, R. P. Lin, and D. L. Mitchell. Density cavity observed over a strong lunar crustal magnetic anomaly in the solar wind: A mini-magnetosphere? *Planetary and Space Science*, 56:941–946, 2008.

[112] I. F. Shaikhislamov, V. M. Antonov, Y. P. Zakharov, E. L. Boyarintsev A. V. Melekhov, V. G. Posukh, , and A. G. Ponomarenko. Mini-magnetosphere: Laboratory experiment, physical model and hall mhd simulation. *Advances in Space Research*, 52:422–436, 2013.

[113] I. F. Shaikhislamov, Y. P. Zakharov, V. G. Posukh, A. V. Melekhov, E. L. Boyarintsev, V. M. Antonov, and A. G. Ponomarenko. Experimental study of a mini-magnetosphere. *Plasma Physics and Controlled Fusion*, 56(2), 2014. http://iopscience.iop.org/0741-3335/56/2/025004/.

[114] M. E. Purucker and J. B. Nicholas. Global spherical harmonic models of the internal magnetic field of the moon based on sequential and coestimation approaches. *Journal of Geophysical Research: Planets*, 115(E12), 2010. http://www.dx.doi.org/10.1029/2010JE003650.

[115] N. C. Richmond and L. L. Hood. A preliminary global map of the vector lunar crustal magnetic field based on Lunar Prospector magnetometer data. *J. Geophys. Res.*, 113(E02010), 2008. http://www.dx.doi.org/10.1029/2007JE002933.

[116] J. E. P. Connerney et al. Magnetic lineations in the ancient crust of mars. *Science*, 284:794–798, 1999.

[117] A.L. Albee, F. palluconi, and T. Thorpe. Overview of the Mars Global Surveyor mission. *J. Geophys. Res.*, 106(E10):23291–23316, 2001.

[118] M. H. Acuña et al. Mars observer magnetic fields investigation. *JGR*, 97(E5):7799–7814, 1992.

[119] M. H. Acuña et al. The magnetic field of mars: Summary results from the aerobraking and mapping orbits. *JGR*, 106(E10):23403–23417, 2001.

[120] D. L. Mitchell, R. P. Lin, C. Mazelle H. Réme, P. A Cloutier, J. E. P Connerney, M. H. Acuña, and N. F. Ness. Probing mars' crustal magnetic field and ionosphere with the MGS electron reflectometer. *J. Geophys. Res.*, 106(E10):23419–23427, 2001.

[121] J. Arkani-Ahmed. A 50-degree spherical harmonic model of the magnetic field of mars. *J. Geophys. Res.*, 106(E10):23197–23208, 2001.

[122] J. C. Cain, B. B. Ferguson, and D. Mozzoni. An n = 90 internal potential function of the martian crustal magnetic field. *J. Geophys. Res.*, 108(E2), 2003. http://www.dx.doi.org/10.1029/2000JE001487.

[123] C. Jacobs, B. van der Holst, and S. Poedts. Comparison between 2.5D and 3D simulations of coronal mass ejections. *Astron. Astrophys.*