

ABSTRACT

KARAMI HALASHI, BEHROUZ. A Reconstructed Discontinuous Galerkin Method for the Magnetohydrodynamics on Arbitrary Grids. (Under the direction of Dr. Hong Luo.)

A reconstructed discontinuous Galerkin (RDG) method based on a Hierarchical Weighted Essentially Non-oscillatory (WENO) reconstruction using a Taylor basis, designed not only to enhance the accuracy of discontinuous Galerkin methods but also to ensure the nonlinear stability of the RDG method, is developed for the solution of the magnetohydrodynamics (MHD) on arbitrary grids. In this method, a quadratic polynomial solution (P_2) is first reconstructed using a Hermite WENO (HWENO) reconstruction from the underlying linear polynomial (P_1) discontinuous Galerkin solution to ensure the linear stability of the RDG method and to improve the efficiency of the underlying DG method. By taking advantage of handily available and yet invaluable information, namely the derivatives in the DG formulation, the stencils used in the reconstruction involve only Von Neumann neighborhood (adjacent face-neighboring cells) and thus are compact and consistent with the underlying DG method. The gradients (first moments) of the quadratic polynomial solution are then reconstructed using a WENO reconstruction in order to eliminate spurious oscillations in the vicinity of strong discontinuities, thus ensuring the nonlinear stability of the RDG method. Temporal discretization is done using a 4th order explicit Runge-Kutta method. The HLLD Riemann solver, introduced in the literature for one dimensional MHD problems, is extended to three dimensional problems on unstructured grids and used to compute the flux functions at interfaces in the present work. Divergence free constraint is satisfied using the so-called Locally Divergence Free (LDF) approach. The LDF formulation

is especially attractive in the context of DG methods, where the gradients of independent variables are handily available and only one of the computed gradients needs simply to be modified by the divergence-free constraint at the end of each time step. The developed RDG method is used to compute a variety of fluid dynamics and magnetohydrodynamics problems in one, two, and three dimensions on arbitrary meshes to demonstrate its accuracy, efficiency, robustness, and versatility. The numerical experiments indicate that this RDG(P_1P_2) is able to achieve the designed third-order of accuracy: one order accuracy higher than the underlying DG method, and outperforms the third-order DG method (DG (P_2)) in terms of both computing costs and storage requirements.

A Reconstructed Discontinuous Galerkin Method for the Magnetohydrodynamics on Arbitrary
Grids

by
Behrouz Karami Halashi

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Mechanical Engineering

Raleigh, North Carolina

2012

APPROVED BY:

Dr. Hong Luo

Committee Chair

Dr. Jack Edwards

Dr. Terek Echeikki

Dr. Zhilin Li

DEDICATION

To my mother for her support.

BIOGRAPHY

Behrouz Karami Halashi was born on May 3, 1980 in Kermanshah, Iran and stayed there till the end of high school.

In 1999, after passing national wide entrance exam, he joined the Mechanical engineering program at Tabriz University located Tabriz, East Azarbaijan, Iran.

He moved to Tehran, capital of Iran in 2004 to pursue his MSc of mechanical engineering in thermo-fluid. His research at MSc was Spectral Element simulation for the compressible Euler equations under advisement of Professor Vahid Esfahanian.

After graduation of MSc at February 2007 he worked for professor Esfahanian at his research institute at University of Tehran. At Vehicle, Fuel, and Environment Research Institute (VFERI) he worked on energy and internal combustion engine related research projects till June 2009.

In August 2009, he moved to Raleigh, NC to pursue his PhD degree in Aerospace Engineering at the department of mechanical and aerospace engineering of North Carolina State University. At NCSU he worked on numerical simulation of magnetohydrodynamics using discontinuous Galerkin methods under the advisement of Dr. Hong Luo.

ACKNOWLEDGMENTS

I would like to thank Dr. Hong Luo for giving me the opportunity to work in his research group and pursue my education under his advisement. His research is advanced and I learned so much while working in his research group. He is also helpful and has been tolerant to me when I faced with challenging problems in my research.

I would like also to thank Dr. Daniel Spicer for his help with providing HLLERiemann solver to compare my developed Riemann solver for MHD problems and also help with correcting and finding some bugs in code and also other useful hints and comments.

I appreciate all of my committee members, Dr. Jack Edwards and Dr. Tarek Echehki from MAE department and Dr. Zhilin Li from Math Department for reviewing my thesis and providing valuable suggestions.

I would like also to thank all of our research group members for their encouragement, understanding, assistance, and cooperation during my research, especially Yidong Xia.

Finally, I would like to thank my family for their love, support, and prayers in all of the years during my studies.

TABLE OF CONTENTS

| | |
|--|------|
| LIST OF TABLES | viii |
| LIST OF FIGURES | ix |
| Chapter 1 Introduction..... | 1 |
| 1.1 Background of Magnetohydrodynamics | 1 |
| 1.2 Background of Discontinuous Galerkin methods | 10 |
| 1.3 Outline..... | 18 |
| Chapter 2 Governing Equations | 20 |
| 2.1 Governing equation of Ideal MHD | 21 |
| 2.2 Eigenvalues of Ideal MHD..... | 22 |
| Chapter 3 Numerical Methods..... | 24 |
| 3.1 The Finite Difference Method..... | 24 |
| 3.1.1 General Principle | 24 |
| 3.1.2 Taylor Series | 25 |
| 3.1.3 Approximation of the Second Derivative | 27 |
| 3.1.4 Finite Difference Formulation for a One-Dimensional Problem..... | 28 |
| 3.2 Finite Element Method..... | 29 |
| 3.2.1 Method of Weighted Residuals..... | 30 |
| 3.2.2 Weak Formulation | 31 |
| 3.3 Finite Volume Method | 34 |
| 3.3.1 Integral Form of Conservation Law..... | 35 |
| 3.4 Discontinuous Galerkin..... | 37 |
| 3.4.1 Discontinuous Galerkin method using a Taylor Basis..... | 41 |

| | | |
|------------|--|-----|
| 3.4.2 | Discontinuous Galerkin Method $DG(P_1)$ | 45 |
| 3.4.3 | Reconstructed Discontinuous Galerkin Method | 47 |
| 3.4.4 | WENO reconstruction at P_2 : $HWENO(P_1P_2)$ | 51 |
| 3.4.5 | WENO reconstruction at P_1 : $HWENO(P_1P_2)$ | 55 |
| 3.4.6 | WENO reconstruction at P_1 : $HWENO(P_1P_1)$ | 58 |
| 3.5 | HLLD Riemann Solver | 65 |
| 3.6 | Divergence Free | 74 |
| 3.6.1 | Vector Potential | 74 |
| 3.6.2 | Projection Method..... | 75 |
| 3.6.3 | Locally Divergence Free..... | 76 |
| Chapter 4 | Temporal Discretization | 78 |
| 4.1 | Euler's Method..... | 78 |
| 4.2 | Fourth order Runge-Kutta Method..... | 81 |
| Chapter 5 | Results..... | 90 |
| 5.1 | One Dimensional Problems..... | 90 |
| 5.2 | Two Dimensional Test Cases..... | 99 |
| 5.2.1 | MHD Blast Wave..... | 99 |
| 5.2.2 | Two Dimensional Vortex..... | 105 |
| 5.2.3 | Two Dimensional MHD Vortex | 111 |
| 5.3 | Three Dimensional Problems | 118 |
| 5.3.1 | Three Dimensional MHD Vortex | 118 |
| Chapter 6 | . Concluding Remarks..... | 123 |
| References | | 127 |

| | |
|--|-----|
| Appendices..... | 144 |
| Appendix A: Derivation of Governing Equations | 145 |
| Appendix B: Eigenvalues of Ideal MHD Flux Jacobian | 149 |

LIST OF TABLES

| | |
|---|----|
| Table 4.1 Value of temperature at $t=480$ second for different step sizes | 87 |
|---|----|

LIST OF FIGURES

| | |
|--|-----|
| Figure 3.1: Representation of polynomial solutions using finite element shape functions | 40 |
| Figure 3.2 Representation of polynomial solutions using a Taylor series expansion..... | 42 |
| Figure 3.3 Two neighboring cells i and j for reconstruction..... | 52 |
| Figure 3.4 Von Neumann neighborhood of cell e to use in Hermit WENO reconstruction... | 63 |
| Figure 3.5 Schematic structure of 4 states Riemann fan for HLLD Riemann Solver | 69 |
| Figure 3.6: Final 3 states Riemann fan for HLLD Riemann solver after above assumptions.. | 73 |
| Figure 4.1 Graphical interpretation of the first step of Euler’s method..... | 79 |
| Figure 4.2 General graphical interpretation of Euler’s method. | 80 |
| Figure 4.3 Comparison of Runge-Kutta 4 th order method with exact solution for different step sizes..... | 87 |
| Figure 4.4 Effect of step size in Runge-Kutta 4th order method. | 88 |
| Figure 4.5 Comparison of Runge-Kutta methods of 1 st (Euler), 2 nd , and 4 th order..... | 88 |
| Figure 5.1 One dimensional shock tube problem with $\gamma=5/3$ at time $t=0.1$ Results of different solvers DGP0, DGP1, and DGP1-WENO-limiter are shown for density, pressure, u, v , and B_y | 92 |
| Figure 5.2 One dimensional shock tube problem with $\gamma=5/3$ at time $t=0.2$ Results of different solvers DGP ₀ , DGP ₁ , and WENO limiter-DGP ₁ with HLLD Riemann solvers are shown for: density(top left), Pressure (top right), u (mid left), v (mid right) B_y (bottom left) | 97 |
| Figure 5.3 Hexahedral and Prismatic cells used for MHD Blast wave in a domain of $[0,1] \times [0,1]$ | 100 |
| Figure 5.4 Density contours for MHD blast wave on hexahedral grid..... | 101 |
| Figure 5.5 Pressure contours for MHD blast wave on hexahedral grid..... | 101 |
| Figure 5.6 X-velocity component contours for MHD blast wave on hexahedral grid | 101 |
| Figure 5.7 Y-velocity component contours for MHD blast wave on hexahedral grid | 101 |
| Figure 5.8 B_x contours for MHD blast wave on hexahedral grid..... | 102 |
| Figure 5.9 B_y contours for MHD blast wave on hexahedral grid | 102 |

| | |
|---|-----|
| Figure 5.10 Velocity magnitude contours for MHD blast wave on hexahedral grid..... | 102 |
| Figure 5.11 Magnetic field magnitude contours for MHD blast wave on hexahedral grid.. | 102 |
| Figure 5.12 Density contours for MHD blast wave on Prismatic cells | 103 |
| Figure 5.13 Pressure contours for MHD blast wave on Prismatic cells | 103 |
| Figure 5.14 X-velocity contours for MHD blast wave on Prismatic cells..... | 103 |
| Figure 5.15 Y-velocity contours for MHD blast wave on Prismatic cells..... | 103 |
| Figure 5.16 B_x contours for MHD blast wave on Prismatic cells..... | 104 |
| Figure 5.17 B_y contours for MHD blast wave on Prismatic cells..... | 104 |
| Figure 5.18 Velocity magnitude contours for MHD blast wave on Prismatic cells | 104 |
| Figure 5.19 Magnetic field magnitude contours for MHD blast wave on Prismatic cells.... | 104 |
| Figure 5.20 Effect of WENO limiter on DGP1 results for MHD blast wave. Density values at $y=0.5$ | 105 |
| Figure 5.21 computational grid used for isentropic vortex, left: Hexahedral cells on $x-y$ plane, Right Prismatic cells in $x-y$ plane..... | 107 |
| Figure 5.22 Density contours for fluid vortex problem, left: density on structured grid, right: density on unstructured grid..... | 107 |
| Figure 5.23 Pressure contours for fluid vortex problem, left: pressure on structured grid, right: pressure on unstructured grid..... | 108 |
| Figure 5.24 u contours for fluid vortex problem, left: velocity on structured grid, right: velocity on unstructured grid | 108 |
| Figure 5.25 v contours for fluid vortex problem, left: velocity on structured grid, right: velocity on unstructured grid | 109 |
| Figure 5.26 Velocity magnitude contours for fluid vortex problem, left: velocity on structured grid, right: velocity on unstructured grid | 109 |
| Figure 5.27 Error and order of accuracy for fluid vortex problem on Hexahedral cells. left: DGP ₁ , right: RDG | 110 |
| Figure 5.28 Error and order of accuracy for fluid vortex problem with Prismatic cells. left: DG(P ₁), right: RDG(P ₁ P ₂)..... | 110 |
| Figure 5.29 Density contours for MHD vortex on Prismatic cells | 113 |

| | |
|--|-----|
| Figure 5.30 Pressure contours for MHD vortex on Prismatic cells | 113 |
| Figure 5.31 u contours for MHD vortex on Prismatic cells..... | 114 |
| Figure 5.32 v contours for MHD vortex on Prismatic cells | 114 |
| Figure 5.33 Velocity magnitude contours for MHD vortex on Prismatic cells | 115 |
| Figure 5.34 B_x of magnetic field contours for MHD vortex on Prismatic cells | 115 |
| Figure 5.35 B_y of magnetic field contours for MHD vortex on Prismatic cells | 116 |
| Figure 5.36 B_z of magnetic field contours for MHD vortex on Prismatic cells | 116 |
| Figure 5.37 Magnetic pressure contours for MHD vortex on Prismatic cells | 117 |
| Figure 5.38 Error and order of accuracy for MHD vortex problem on hexahedral cells. left: DGP ₁ , right: RDG | 117 |
| Figure 5.39 Error and order of accuracy for MHD vortex problem on Prismatic cells. left: DGP ₁ , right: RDG | 118 |
| Figure 5.40 Errors and order of accuracy for DGP1 and RDG, three dimensional MHD vortex on Hexahedral cells..... | 120 |
| Figure 5.41 Density contours for three dimensional MHD vortex | 121 |
| Figure 5.42 Pressure contours for three dimensional MHD vortex | 121 |
| Figure 5.43 u contours for three dimensional MHD vortex..... | 121 |
| Figure 5.44 v contours for three dimensional MHD vortex..... | 121 |
| Figure 5.45 B_x of magnetic field contours for three dimensional MHD vortex | 122 |
| Figure 5.46 B_y of magnetic field contours for three dimensional MHD vortex | 122 |
| Figure 5.47 B_z of magnetic field contours for three dimensional MHD vortex | 122 |
| Figure 5.48 Magnetic pressure contours for three dimensional MHD vortex | 122 |

Chapter 1 Introduction

A background about this research is given in this chapter. The main research effort presented in this work is to develop a reconstructed Discontinuous Galerkin Method for solving on Magneto hydrodynamic flow problems. First we provide a background about magnetohtdrodynamics and its applications. Then we introduce Discontinuous Galerkin methods and give a brief review about their history, applications, advantages, and disadvantages. Finally, the outline of this research and a brief review of each chapter are presented.

1.1 Background of Magneto hydrodynamics

MagnetoHydroDynamics (MHD) explains the area that combines classical fluid dynamics with electro dynamics. It deals with flows of an electrically conducting fluid which are subject to magnetic field or electric field driven by external source.

The interaction of moving conducting fluids with electric and magnetic fields provides a rich variety of phenomena associated with electro-fluid-mechanical energy conversion. Effects from such interactions can be observed in liquids, gases, two-phase mixtures, or plasmas. Numerous scientific and technical applications exist, such as heating and flow control in metals processing, power generation from two-phase mixtures or seeded high temperature gases, magnetic confinement of high-temperature plasmas, even dynamos that create magnetic fields in planetary bodies. Several terms have been applied to the broad field of electromagnetic effects in conducting fluids, such as magneto-fluid mechanics, magneto-

gas-dynamics, and the more common one used here, magnetohydrodynamics, or “MHD” in short.

The presence of the Lorentz force on the flow of conducting *liquids* can alter the velocity and pressure characteristics of the flow. The interaction with a magnetic field also can significantly delay the onset of turbulent fluctuations. These two effects together or individually can dramatically alter the heat transfer characteristics and fluid drag in closed or open channel flows. Technological applications of such phenomena include cooling systems for magnetic fusion reactors and reduced-drag ship hulls and airplane fuselages.

Practical MHD devices have been in use since the early part of the 20th century. For example, an MHD pump prototype was built as early as 1907 [1]. More recently, MHD devices have been used for stirring, levitating, and otherwise controlling flows of liquid metals for metallurgical processing and other applications [2]. Gas-phase MHD is probably best known in MHD power generation. Since 1959 [3], [4] major efforts have been carried out around the world to develop this technology in order to improve electric conversion efficiency, increase reliability by eliminating moving parts, and reduce emissions from coal and gas plants. Closed-cycle liquid metal MHD systems using both single phase and two-phase flows also have been explored.

Still more novel applications are in development or on the horizon. For example, recent research has shown the possibility of seawater propulsion using MHD [5] and control of turbulent boundary layers to reduce drag [6]. Extensive worldwide research on magnetic

confinement of plasmas has led to attainment of conditions approaching those needed to sustain fusion reactions [7].

MHD is relatively new topic in science and engineering started with work of Hartmann on liquid metal channel flow with strong external magnetic field and Alfven on gas MHD. Today MHD has developed to field of applied and fundamental research in engineering and physical science. Moreover MHD devices are used in modern metallurgical technologies and may become important in power generation in future. As an example of MHD in industry we introduce some applications as follow.

One important application of MHD is electromagnetic pumping of liquid metals. The MHD force can be applied in such a way that useful work can be done. For example, EM pumps can be designed to precisely control liquid flows, liquid metal flows, in particular, where high temperature and corrosive tendencies prohibit the use of seals in standard mechanical pumps. Such pumps have no moving parts and are extremely reliable. The electrical cross-current \mathbf{I} through the liquid interact with the external magnetic field \mathbf{B} and generate electromagnetic force (Lorenz Force) which pushes liquid through tube. Usually high electrical current needed to achieve enough volumetric flow rates. So efficiency of this kind of pump is low. But they have advantage of being very simple. This device can be constructed with no moving parts and no direct contact with the working liquid. This is a distinct advantage if high temperature and/or corrosive liquids must be handled. The absence of seals or moving parts leads to a highly reliable system. In addition, EM pumps are

typically controllable, and even reversible, by varying the magnitude and direction of the applied current.

Another potential use of EM pumping technology is MHD thrusters for ship propulsion. Seawater has a moderate electrical conductivity, of the order of $5 \Omega^{-1}\text{m}^{-1}$, and under the appropriate set of conditions can be pumped by the Lorentz force. Care must be taken to avoid large losses in conducting walls in this application, but this is more easily done when the working fluid is seawater, rather than high temperature liquid metals. Conduction pump thrusters [8] are more commonly envisioned for MHD ship propulsion because of the difficulty inducing large currents in poorly conducting water. For a given size channel (usually limited by the size of the craft under consideration), a given applied voltage (usually limited by the power supply aboard the craft, *e.g.* a battery) and a given liquid (seawater), the mechanical power is maximized at $\eta_e=50\%$ [9] This means one half of the electrical power supplied is lost as Ohmic heating. Thruster designers must decide whether their goal is to maximize mechanical power, or to minimize energy consumption. For a moderately sized submarine (10-m diameter, 83-m length) using four conduction thrusters with length $L=55$ m, $b=5$ cm and $a=15$ cm, a 5 T field is sufficient to generate reasonable thrust and efficiency. At a speed of 36 knots, the thrusters will consume about 66 MW of electric power, requiring a 200 MW thermal nuclear plant with a typical thermal conversion efficiency (excluding power needed for other boat systems). This level of power is not unreasonable for a submarine of this size. Superconducting magnets are necessary for this field strength and core size, since the Ohmic losses in resistive magnets would be

unacceptable. At least one design using an EM pump thruster has been advanced. The “ripple motor” described by Mitchell and Gubser [10] utilizes a AC solenoidal winding around a core of sea water. An annulus of liquid sodium or other liquid metal serves as an intermediate layer separated from the sea water by a flexible membrane. The thickness of the sodium layer is matched to the skin depth of the AC field. The traveling magnetic field sets up a traveling pressure wave in the sodium, and thus a traveling wave on the flexible membrane. This wave pushes along the seawater and eventually ejects it out of the trailing end of the thruster, providing the thrust.

MHD forces can have a large effect on the turbulence structure of liquid flows. Not only does the induction of a current density result in Ohmic dissipation of energy, a new energy loss mechanism that augments the viscous dissipation, but the field also changes the average velocity profiles, resulting in new turbulence creation scenarios as compared to non-MHD flows. The magnetic field is typically thought to laminarize already turbulent flows, or to prevent the transition to turbulence in laminar flows. In theory, it is possible to laminarize any flow with a sufficiently strong magnetic field. Control of turbulence near the wall of a ship or submarine can in theory reduce the overall drag on the structure. Early work on MHD channels flows [11] showed that the pressure drop in an initially turbulent LM duct flow could be reduced by the judicious application of a magnetic field. For the control of turbulence near ships, one must contend with the fact that sea water is a poor electrical conductor, and that induced currents alone will not dissipate enough energy to stabilize a turbulent boundary layer. Instead, currents must be generated by an applied voltage. One

such scheme to reduce drag on, and radiated noise from, a flat plate is to construct the surface with alternating north and south poles magnets interspersed between positive and negative electrodes. The criss-crossing lines of magnetic field and current induce a Lorentz force in the stream wise direction, acting as a sort of one sided conduction pump. Preliminary experiments [12] have shown that turbulent fluctuations can be reduced over much of the boundary layer when the modified interaction parameter is order one or larger. The boundary layer is found to approach an asymptotic value, rather than growing indefinitely and breaking down due to instability. Work in this area by a number of researchers is continuing.

Open channel flows of liquids in magnetic fields are of interest for metallurgical and welding applications where melts and melt layers are influenced by electric currents and applied magnetic fields. There is also interest in open channel MHD flows in magnetic fusion energy reactors where it might be advantageous to have high heat flux surfaces facing the burning plasma be covered with a flowing liquid metal layer. When the problem of open channel MHD flows is examined closely, one finds that the complicated motion of closed channel duct flows described above become even more complicated when the liquid interface (free surface) is allowed to move in response to MHD forces. The interfacial boundary condition for open channel flows requires that the tangential component of the viscous stress must be continuous. The term “free surface” implies the less general case where the liquid surface is unhampered by friction with a gas phase outside the liquid region, and so the tangential component of the stress vanishes. However, this condition is changed in MHD flows where the total stress, the sum of the tangential viscous and magnetic stresses, must be

continuous. Some simple cases of flow down an inclined plate are analyzed by Alpher [13], Aitov [14], Morley [15] and others.

Since 1959, substantial effort has been devoted to exploring the conditions under which a conducting gas moving through a magnetic field might generate useful electrical power.

The primary motivation for the development and use of MHD generators in central-station power plants is the production of power at lower cost through reduced fuel costs per unit of energy produced, traded off against additional capital and operating costs. Operation at high thermal conversion efficiency provides the added benefit of reduced thermal discharge from the plant, thus reducing thermal pollution as compared with conventional steam plants with $\eta \sim 40\%$ or nuclear plants with efficiency as low as 30%.

As originally envisioned, the MHD generator was a “topping” unit on an otherwise conventional steam turbine-generator station. In this case, electric power is generated in the MHD unit, and its exhaust heat, with high temperature as high as 2200 K, is used to generate steam. The limiting Carnot efficiency for such a station might be raised from a maximum of about 65% ($T_1=850$ K, $T_2=300$ K) upward toward 85% ($T_1=2600$ K, $T_2=420$ K). The net efficiency of the combined cycle can be expressed as $\eta_1 + \eta_2(1 - \eta_1)$, where η_1 is the efficiency of the MHD generator and η_2 is the efficiency of the “bottom” steam plant. Typical values are $\eta_1=0.25$ and $\eta_2=0.4$, for an overall efficiency of 0.55.

Perhaps the greatest importance of the MHD steam plant, as now envisioned, is its potential for very low air pollution while burning high-sulfur coal [16]. The SO_2 , NO_2 and particulate

emissions are all reduced to very low levels by their interaction with the MHD “seed” material. In pilot plant tests, 2.2 wt.% sulfur coal was burned in a cyclone furnace at 2200°C with seed concentration of 1 g.mole K_2CO_3 /kg coal with 99.8% removal of SO_2 , leaving only 5 ppm SO_2 in the gaseous effluent. This occurs because of an affinity that the potassium seed material has for SO_2 . So seed recovery in the MHD system, which is necessary for economic reasons, also removes the SO_2 . The seed removal costs are calculated as approximately 1/5 of the SO_2 removal costs in a conventional coal-fired plant.

In the same tests, through the use of 2-stage combustion, NO_x emissions were reduced below 150 ppm, and complete combustion of CO was achieved [16]-[17].

Plasma physics is a rapidly expanding field of science. The conjunction of magnetohydrodynamics and plasma physics bears the promise of industrial applications, to convert directly thermal energy into electricity, without rotating machines. The application of MHD makes conversion of heat to electricity possible [19]. In this way a highly pressurized and hot gas (~3500K) and thus ionized gas flow through a nozzle which located in strong magnetic field. In this way charge separation occurs at hot gas under imposed external magnetic field and an electrical potential difference induced in electrodes inserted in channel wall aligned with the direction of the magnetic field. This MHD power generation process utilized as topping cycle in a combined thermal power plant gives a potential for high thermal efficiency. However there are some problems with respect to corrosion of electrodes facing with high temperature gas.

Gas MHD has a very important role in geophysics and astrophysics as many planets and stars and even galaxies carry magnetic fields whose existence and origin is explained by dynamo effect. The dynamo effect describes the conversion of mechanical energy of fluid motion into electromagnetic field energy. For this to occur the fluid must be electrically conducting and its motion has to have swirling property and moreover fluid velocity has to exceed a certain critical value which depends on the flow pattern and the conductivity. For planetary magnetic field such as the earth magnetic field, it is suggested that thermal convection inside the liquid core of the earth under the influence of the planets rotation provides the energy to sustain the observed magnetic field.

The plasma is sometimes considered as a particular state of matter. Indeed, matter can be found in four different states: *solid*, *liquid*, *gas* or *plasma*. It is well known that when enough energy is added to a solid, the solid will change its state and become a liquid, which in turn with enough energy added to it, will become a gas. The molecules of such a gas will assume a variety of degrees of freedom. If one keeps adding energy, the kinetic energy of the particles will increase and molecular impacts will become so intense that dissociation will result between some of electrons and the rest of each molecule. The gas will become a conglomeration of positively and negatively charged particles and neutrality particles too, which is on the average neutral. The gas will become plasma which physical properties are different from gas. The plasma state is actually by far the most common form of matter (up to 99 per cent in the universe) and it is also the most energetic state.

The fields of science and engineering in which plasma physics can be applied are increasing more and more in number. Plasma physics applications have been identified with those of electrical discharges in gases, but behavior of plasmas submitted to magnetic fields is very important. The notions of magnetic contraction of electrical discharges, of magnetic confinement in thermonuclear devices, and those applications to plasma physics belonging to the vast class of magnetohydrodynamic phenomena, are soon likely to be important fields of application in science and engineering. In the earth's atmosphere, the fast moving bodies produce plasmas, which are likely to create new fields of interest in aeronautics and communications science.

1.2 Background of Discontinuous Galerkin methods

Computational MHD is a very active research field due to the increasing demand for quantitative results for realistic magnetic configurations on the one hand and availability of powerful computers on the other hand [20]. Many MHD phenomena cannot be described by analytical methods in all of their complexity although simplified analytical models led to inaccurate insight into the fundamental physics of MHD Processes. The fast increase of computer speed and memory allows simulations with ever more physics in the equation and also taking into account full three dimensional geometrical effects.

The governing equations of ideal MHD equations are a set of nonlinear hyperbolic partial differential equations. Many suitable combinations of spatial discretizations (Finite Volume Methods, Finite Difference Methods, Finite Element Methods, etc) and temporal

discretizations (explicit, implicit) can be used to solve this set of equations. However the conservation law nature of ideal MHD equations introduces additional challenges when discontinuous shock dominated evolutions are computed.

The discontinuous Galerkin methods (DGM) [21]- [29] have recently become popular for the solution of systems of conservation laws. Originally introduced for the solution of neutron transport equations[21], nowadays they are widely used in computational fluid dynamics, computational acoustics, and computational magneto-hydrodynamics.

The discontinuous Galerkin (DG) methods are locally conservative, stable, and high-order accurate methods which can easily handle complex geometries, irregular meshes with hanging nodes, and allow approximations that have polynomials of different degrees in different elements. These properties, which render them ideal to be used with *hp*-adaptive strategies, not only have brought these methods into the main stream of computational fluid dynamics, for example, in gas dynamics [31], [31], [32], compressible [33], [34], [36], [35] and incompressible [37], [38], [39] flows, turbo machinery [40], magneto-hydrodynamics [41], granular flows [42], [43], semiconductor device simulation [44], [45], viscoplastic crack growth and chemical transport [46], viscoelasticity [47], [48], [49] and transport of contaminant in porous media [50], [51], [52], [53], but have also prompted their application to a wide variety of problems for which they were not originally intended like, for example, Hamilton-Jacobi equations [54], [55], [56], second-order elliptic problems [57], [58], [59], [60], [61], [62], [63], [64], elasticity [65], [66], and Korteweg-deVries equations [67], [68]. An introduction to DG methods can be found in the short monograph [69]. A history of their

development and the state of the art up to 1999 can be found in [70]. Finally, a fairly complete and updated review is given in [71].

The discontinuous Galerkin methods combine two advantageous features commonly associated with finite element and finite volume methods. As in classical finite element methods, accuracy is obtained by means of high-order polynomial approximation within an element rather than by wide stencils as in the case of finite volume methods. The physics of wave propagation is, however, accounted for by solving the Riemann problems that arise from the discontinuous representation of the solution at element interfaces. In this respect, the DG methods are similar to finite volume methods. The discontinuous Galerkin methods have many attractive features:

- 1) They have several useful mathematical properties with respect to conservation, stability, and convergence;
- 2) These methods can be easily extended to higher-orders ($>2^{\text{nd}}$) approximation;
- 3) These methods are well suited for complex geometries since they can be applied on unstructured grids. In addition, the methods can also handle non-conforming elements, where the grids are allowed to have hanging nodes;
- 4) The methods are highly parallelizable, as they are compact and each element is independent. Since the elements are discontinuous, and the inter-element communications are minimal, domain decomposition can be efficiently employed. The compactness also allows for structured and simplified coding for the methods;

5) They can easily handle adaptive strategies, since refining or coarsening a grid can be achieved without considering the continuity restriction commonly associated with the conforming elements. The methods allow easy implementation of *hp*-refinement, for example, the order of accuracy, or shape, can vary from element to element;

6) They have the ability to compute low Mach number flow problems without recourse to the time-preconditioning techniques normally required for the finite volume methods.

In contrast to the enormous advances in the theoretical and numerical analysis of the DGM, the development of a viable, attractive, competitive, and ultimately superior DG method over the more mature and well-established second order methods is relatively an untouched area. This is mainly due to the fact that the DGM have a number of weaknesses that have yet to be addressed, before they can be robustly used for problems of practical interest in magnetohydrodynamics in a complex configuration environment. In particular, how to effectively control spurious oscillations in the presence of strong discontinuities, and how to reduce the computing costs for the DGM remain the two most challenging and unresolved issues in the DGM. Indeed, compared to the finite element methods and finite volume methods, the DGM require solutions of systems of equations with more unknowns for the same grids. Consequently, these methods have been recognized as expensive in terms of both computational costs and storage requirements.

In order to reduce high costs associated with the DGM, Dumbser et al [72][73][74] have introduced a new family of reconstructed DGM, termed P_nP_m schemes and referred to as RDG (P_nP_m) in this dissertation, where P_n indicates that a piecewise polynomial of degree of

n is used to represent a DG solution, and P_m represents a reconstructed polynomial solution of degree of m ($m \geq n$) that is used to compute the fluxes. The RDG($P_n P_m$) schemes are designed to enhance the accuracy of the discontinuous Galerkin method by increasing the order of the underlying polynomial solution. The beauty of RDG ($P_n P_m$) schemes is that they provide a unified formulation for both finite volume and DGM, and contain both classical finite volume and standard DG methods as two special cases of RDG ($P_n P_m$) schemes, and thus allow for a direct efficiency comparison. When $n=0$, i.e. a piecewise constant polynomial is used to represent a numerical solution, RDG ($P_0 P_m$) is nothing but classical high order finite volume schemes, where a polynomial solution of degree m ($m \geq 1$) is reconstructed from a piecewise constant solution. When $m=n$, the reconstruction reduces to the identity operator, and RDG ($P_n P_n$) scheme yields a standard DG method.

Obviously, the construction of an accurate and efficient reconstruction operator is crucial to the success of the RDG ($P_n P_m$) schemes. In Dumbser's work, a higher order polynomial solution is reconstructed using a L_2 projection, requiring it indistinguishable from the underlying DG solutions in the contributing cells in the weak sense. The resultant over-determined system is then solved using a least-squares method that guarantees exact conservation, not only of the cell averages but also of all higher order moments in the reconstructed cell itself, such as slopes and curvatures. However, this conservative least-squares reconstruction approach is computationally expensive, as the L_2 projection, i.e., the operation of integration, is required to obtain the resulting over-determined system. Furthermore, the reconstruction might be problematic for a boundary cell, where the number

of the face-neighboring cells might be not enough to provide the necessary information to recover a polynomial solution of a desired order. Fortunately, the projection-based reconstruction is not the only way to obtain a polynomial solution of higher order from the underlying discontinuous Galerkin solutions. In a reconstructed DG method using a Taylor basis [75][76][77] developed by Luo et al. for the solution of the compressible Euler and Navier-Stokes equations on arbitrary grids, a higher order polynomial solution is reconstructed by use of a strong interpolation, requiring point values and derivatives to be interpolated on the face-neighboring cells. The resulting over-determined linear system of equations is then solved in the least-squares sense. This reconstruction scheme only involves Von Neumann neighborhood, and thus is compact, simple, robust, and flexible. Like the projection-based reconstruction, the strong reconstruction scheme guarantees exact conservation, not only of the cell averages but also of their slopes due to a judicious choice of the Taylor basis. More recently, Zhang et al. [78][79] presented a class of hybrid DG/FV methods for the conservation laws, where the second derivatives in a cell are obtained from the first derivatives in the cell itself and its neighboring cells using a Green-Gauss reconstruction widely used in the finite volume methods. This provides a fast, simple, and robust way to obtain higher-order polynomial solutions. Lately, Luo et al. [80][81] have conducted a comparative study for these three reconstructed discontinuous Galerkin methods RDG(P_1P_2) by solving 2D Euler equations on arbitrary grids. It is found that all three reconstructed discontinuous Galerkin methods can deliver the desired third order of accuracy and significantly improve the accuracy of the underlying second-order DG method, although

the least-squares reconstruction method provides the best performance in terms of both accuracy and robustness.

Unfortunately, the attempt to extend RDG method to solve 3D Euler equations on tetrahedral grids was not successful. Like the second order cell-centered finite volume methods RDG (P_0P_1), the resultant RDG (P_1P_2) method is unstable. Although RDG (P_0P_1) methods are in general stable in 2D and on Cartesian or structured grids in 3D, they suffer from the so-called linear instability on unstructured tetrahedral grids, when the reconstruction stencils only involve Von Neumann neighborhood, i.e., adjacent face-neighboring cells [82]. The RDG (P_1P_2) method exhibits the same linear instability, which can be overcome by using extended stencils. However, this is achieved at the expense of sacrificing the compactness of the underlying DG methods. Furthermore, these linear reconstruction-based DG methods will suffer from non-physical oscillations in the vicinity of strong discontinuities for the compressible Euler equations. Alternatively, ENO, WENO, and HWENO can be used to reconstruct a higher-order polynomial solution, thereby not only enhancing the order of accuracy of the underlying DG method but also achieving both linear and non-linear stability. This type of hybrid HWENO+DG schemes has been developed on 1D and 2D structured grids by Balsara et al.[83], where the HWENO reconstruction is relatively simple and straightforward.

Recently, Luo et al. have developed a reconstructed discontinuous Galerkin method based on a Hierarchical WENO reconstruction [84], HWENO(P_1P_2) [85][86], using a Taylor basis for the solution of the compressible Euler equations on unstructured tetrahedral grids. The

HWENO(P_1P_2) method is designed not only to reduce the high computing costs of the DGM, but also to avoid spurious oscillations in the vicinity of strong discontinuities, thus effectively addressing the two shortcomings of the DGM. In this HWENO(P_1P_2) method, a quadratic solution is first reconstructed to enhance the accuracy of the underlying DG method in two steps: (1) all second derivatives on each cell are first reconstructed using the solution variables and their first derivatives from adjacent face-neighboring cells via a strong interpolation; (2) the final second derivatives on each cell are then obtained using a WENO strategy based on the reconstructed second derivatives on the cell itself and its adjacent face-neighboring cells. This reconstruction scheme, by taking advantage of handily available and yet valuable information namely the gradients in the context of the DG methods, only involves Von Neumann neighborhood and thus is compact, simple, robust, and flexible. As the underlying DG method is second-order, and the basis functions are at most linear functions, fewer quadrature points are then required for both domain and face integrals, and the number of unknowns (the number of degrees of freedom) remains the same as for the DG(P_1). Consequently, this RDG method is more efficient than its third order DG(P_2) counterpart. The gradients of the quadratic polynomial solutions are then modified using a WENO reconstruction in order to eliminate non-physical oscillations in the vicinity of strong discontinuities, thus ensuring the non-linear stability of the RDG method. The developed HWENO(P_1P_2) method is used to compute a variety of flow problems on tetrahedral grids to demonstrate its accuracy, robustness, and non-oscillatory performance. The presented numerical results indicate that this HWENO (P_1P_2) method is able to capture shock waves

within once cell without any spurious oscillations, and achieve the designed third-order of accuracy: one order accuracy higher than the underlying DG(P_1) method, and thus significantly increase its accuracy without significant increase in computing costs and memory requirements.

1.3 Outline

The objective of the efforts presented in this work is to extend the reconstructed discontinuous Galerkin method for solving the magnetohydrodynamics (MHD) equations on arbitrary grids. The HLLD Riemann solver, introduced in the literature for one dimensional MHD problems, and extended for three dimensional problems in the present work, is used to compute the flux functions at interfaces. Divergence free constraint is satisfied using the so-called Locally Divergence Free (LDF) approach. The LDF formulation is especially attractive in the context of DG methods, where the gradients of independent variables are handily available and only one of the computed gradients needs simply to be modified by the divergence-free constraint at the end of each time integration step. Temporal discretization is done using a 4th order explicit Runge-Kutta method. Numerical experiments for a variety of test cases are conducted to demonstrate the accuracy and efficiency of the developed RDG method.

The remainder of this dissertation is organized as follows: Chapter 2 describes the governing equations of Magnetohydrodynamics in conservative form that consist of a system of eight nonlinear partial differential equations. The eigenvalue and eigenvectors of this

system are then presented. The numerical methods are discussed in Chapter 3. Some of the traditional spatial discretization methods, including Finite Difference, Finite Element and Finite Volume methods are first reviewed. A discontinuous Galerkin method based on a Taylor basis is then presented followed by a detailed description of the baseline linear discontinuous Galerkin method, $DG(P_1)$. The different reconstructed discontinuous Galerkin (RDG) methods based on Weighted Essentially Non-Oscillatory (WENO) reconstruction are finally presented. In this chapter, we also discuss the HLLD Riemann solver to compute numerical fluxes at interface for the MHD equations. Some divergence free techniques are presented in the last section of this chapter, including the **L**ocal **D**ivergence **F**ree (LDF) approach used in this research. Chapter 4 reviews some of the classic temporal discretization techniques for solving ordinary differential equations. Many of these temporal discretization techniques with different order of accuracy and stability limitation can be used for the integration of the MHD equations in time. Both an explicit first order Euler method and a 4th order explicit Runge-Kutta method are described, although the explicit 4th order Runge-Kutta method is mainly used for integrating the MHD equations in the current work because of its high-order accuracy and large stability bound. The computational results for a variety of 1D, 2D, and 3D test cases are presented in Chapter 5. Finally, conclusions and suggestions for future work are given in Chapter 6.

Chapter 2 Governing Equations

The equations of magnetohydrodynamics can be derived from hydrodynamic equations, the Lorentz force and Ohm's law, and the Maxwell equations. The most important assumption is that the macroscopic velocity V , of plasma is much less than speed of light c . the displacement current $\partial E/\partial t$ can be ignored, and the current can be expressed from the magnetic field B . the electric field E can be expressed from Lorentz force and the conductivity σ . In this way electromagnetic waves are eliminated, which simplify the numerical solution. Both the total pressure P_{tot} and total energy density e will have contributions from the magnetic field. Following equations hold:

$$J = \frac{1}{\mu} \nabla \times B \quad (2.1)$$

$$E = \frac{J}{\sigma} - V \times B \quad (2.2)$$

$$P = (\gamma - 1) \left(\rho e - \rho \frac{V \cdot V}{2} - \frac{B \cdot B}{2\mu} \right) \quad (2.3)$$

$$P_T = P + \frac{B \cdot B}{2\mu} \quad (2.4)$$

where ρ , p , and e denote the density, pressure, and specific total energy of the fluid, respectively, and $V=(u,v,w)$ and $B=(B_x,B_y,B_z)$ are velocity and magnetic field components in the x , y , and z -direction, respectively. Furthermore, γ is ratio of specific heat, σ is the electric

conductivity. Here the resistivity $\eta = \frac{1}{\sigma}$ will be used, and we choose units of magnetic field such that $\mu = 1$.

After eliminating the electric field and the current, the magnetized plasma can be fully described by the primitive variables which are functions of time and three spatial coordinates. For Ideal MHD= 0 , which in resistive MHD, $\eta > 0$ is not negligible at least in some parts of flow.

2.1 Governing equation of Ideal MHD

The MHD equations can be expressed in various mathematical forms however, for numerical methods the conservative form is preferred. This form is more convenient since we are looking for weak solution of these coupled partial differential equations. The 3D resistive MHD equations can be written in conservative form as

$$\rho_t + \nabla \cdot (\rho V) = 0 \quad (2.5)$$

$$(\rho V)_t + \nabla \cdot (\rho V V^T + P_T(I) - B B^T) = 0 \quad (2.6)$$

$$(\rho e)_t + \nabla \cdot ((\rho e + P_T)V - B(V \cdot B) - B \times \eta J) = 0 \quad (2.7)$$

$$B_t + \nabla \cdot (V B^T - B V^T) + \nabla \times (\eta J) = 0 \quad (2.8)$$

The above equation can be expressed in vector form as:

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = 0 \quad (2.9)$$

In which Q is vector of independent variables and F , G , and H are flux vectors in x, y, and z direction respectively:

$$\begin{aligned}
 Q &= \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \\ B_x \\ B_y \\ B_z \end{pmatrix} & F &= \begin{pmatrix} \rho u \\ \rho u^2 - B_x^2 + P_T \\ \rho uv - B_y B_x \\ \rho uw - B_z B_x \\ (\rho E + P_T)u - B_x(V \cdot B) \\ 0 \\ uB_y - vB_x \\ uB_z - wB_x \end{pmatrix} \\
 G &= \begin{pmatrix} \rho v \\ \rho vu - B_x B_y \\ \rho v^2 - B_y^2 + P_T \\ \rho vw - B_z B_y \\ (\rho E + P_T)v - B_y(V \cdot B) \\ vB_x - uB_y \\ 0 \\ vB_z - wB_y \end{pmatrix} & H &= \begin{pmatrix} \rho w \\ \rho wu - B_x B_z \\ \rho wv - B_y B_z \\ \rho w^2 - B_z^2 + P_T \\ (\rho E + P_T)w - B_z(V \cdot B) \\ wB_x - uB_z \\ wB_y - vB_z \\ 0 \end{pmatrix}
 \end{aligned} \tag{2.10}$$

2.2 Eigenvalues of Ideal MHD

From the flux vectors of (2.10), we can obtain the Jacobian matrices

$$A_x = \frac{\partial F}{\partial Q}, \quad A_y = \frac{\partial G}{\partial Q}, \quad A_z = \frac{\partial H}{\partial Q} \tag{2.11}$$

It is observed that G and H can be obtained from properly index permuting. Therefore, A_y and A_z have similar structure as A_x . The eigenvalues and eigenvectors for A_x have been extensively studied by many authors, Brio & Wu [87]; Ryu & Jones[88]; Roe & Balsara [89]. There are two set of eigenvectors for the eigen-system of A_x . One is a direct extension of the one-dimensional system and the other is based on the modification proposed by Powell et al

[92]. In the direct extension, the 6th component is ignored and the rest is identical to the one-dimensional flux where B_x is a constant. The eigenvalues for the 7×7 systems are

$$\lambda_{1,7} = v_x \pm c_f, \quad \lambda_{2,6} = v_x \pm c_a, \quad \lambda_{3,5} = v_x \pm c_s \quad \lambda_4 = v_x \quad (2.12)$$

In which Alfven wave speeds defined as follow:

$$C_{alfven} = \frac{|B_x|}{\sqrt{\rho}} \quad (2.13)$$

And sound speed as:

$$C_{sound} = \sqrt{\frac{\gamma p}{\rho}} \quad (2.14)$$

Two fast and slow speeds are defined as follow:

$$C_{fast,slow} = \frac{1}{\sqrt{2}} \left\{ C_{sound}^2 + C_{alfven}^2 \pm \sqrt{(C_{sound}^2 + C_{alfven}^2)^2 - 4C_{sound}^2 C_{alfven}^2} \right\}^{\frac{1}{2}} \quad (2.15)$$

In the 8×8 system of Powell, the corresponding eigenvalues are:

$$\lambda_{1,8} = v_n \pm c_f, \quad \lambda_{2,7} = v_n \pm c_a, \quad \lambda_{3,6} = v_n \pm c_s \quad \lambda_{4,5} = v_n \quad (2.16)$$

Where n is direction normal to the surface. Eigenvalues and eigenvectors of MHD described with more details at appendix B

Chapter 3 Numerical Methods

This chapter introduces some spatial discretization techniques used in this research. Some fundamental discretization techniques: Finite Difference Method, Finite Element Method (FEM), and Finite Volume Method (FVM) are introduced. Based on these, discontinuous Galerkin Method (DGM) is described. The spatial discretization to the governing equations leads to a semi-discrete system of nonlinear equations. This set of the ordinary differential equations (ODEs) can then be integrated in time.

3.1 The Finite Difference Method

The finite difference approximations for derivatives are one of the simplest and of the oldest methods to solve differential equations. It was already known by L. Euler, in one dimension of space and was probably extended to dimension two by C. Runge. The advent of finite difference techniques in numerical applications began in the early 1950s and their development was stimulated by the emergence of computers that offered a convenient framework for dealing with complex problems of science and technology. Theoretical results have been obtained during the last five decades regarding the accuracy, stability and convergence of the finite difference method for partial differential equations.

3.1.1 General Principle

The principle of finite difference methods is close to the numerical schemes used to solve ordinary differential equations. It consists in approximating the differential operator by

replacing the derivatives in the equation using differential quotients. The domain is partitioned in space and in time and approximations of the solution are computed at the space or time points. The error between the numerical solution and the exact solution is determined by the error that is committed by going from a differential operator to a difference operator. This error is called the discretization error or truncation error. The term truncation error reflects the fact that a finite part of a Taylor series is used in the approximation. For the sake of simplicity, we shall consider the one-dimensional case only. The main concept behind any finite difference scheme is related to the definition of the derivative of a smooth function u at a point $x \in R$:

$$u'(x) = \lim_{h \rightarrow 0} \frac{u(x+h) - u(x)}{h} \quad (3.1)$$

and to the fact that when h tends to 0 (without vanishing), the quotient on the right-hand side provides a "good" approximation of the derivative. In other words, h should be sufficiently small to get a good approximation. The approximation is good when the error committed in this approximation (i.e. when replacing the derivative by the differential quotient) tends towards zero when h tends to zero. If the function u is sufficiently smooth in the neighborhood of x , it is possible to quantify this error using a Taylor expansion.

3.1.2 Taylor Series

Suppose the function u is C^2 continuous in the neighborhood of x . For any $h > 0$ we have:

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2} u''(x+h_1) \quad (3.2)$$

Where h_1 is a value between 0 and h (i.e. $x+h_1$ is point of $]x, x+h[$). For the treatment of problems, it is convenient to retain only the first two terms of the previous expression:

$$u(x + h) = u(x) + hu'(x) + O(h^2)$$

Where the term $O(h^2)$ indicates that the error of the approximation is proportional to h^2 . From the equation (3.1), we deduce that there exists a constant $C > 0$, such that for $h > 0$ sufficiently small we have:

$$\left| \frac{u(x + h) - u(x)}{h} - u'(x) \right| \leq Ch \quad (3.3)$$

for $h \leq h_0$ ($h_0 > 0$ given). The error committed by replacing the derivative $u'(x)$ by the differential quotient is of order h . The approximation of u' at point x is said to be consistent at the first order.

This approximation is known as the forward difference approximant of u' . More generally, we define an approximation at order p of the derivative.

The approximation of the derivative u' at point x is of order p ($p > 0$) if there exists a constant $C > 0$, independent of h , such that the error between the derivative and its approximation is bounded by Ch^p (i.e. is exactly $O(h^p)$). Likewise, we can define the first order backward difference approximation of u' at point x as:

$$u(x - h) = u(x) - hu'(x) + O(h^2)$$

Obviously, other approximations can be considered. In order to improve the accuracy of the approximation, we define a consistent approximation, called the central difference approximation, by taking the points $x - h$ and $x + h$ into account. Suppose that the function u is three times differentiable in the vicinity of x :

$$u(x + h) = u(x) + hu'(x) + \frac{h^2}{2} u''(x) + \frac{h^3}{6} u'''(x)$$

$$u(x - h) = u(x) - hu'(x) + \frac{h^2}{2} u''(x) - \frac{h^3}{6} u'''(x)$$

By subtracting these two expressions we obtain:

$$\frac{u(x + h) - u(x - h)}{2h} = u'(x) + \frac{h^3}{6} u'''(x)$$

Hence we have the following bound on the approximation error:

$$\left| \frac{u(x + h) - u(x - h)}{2h} - u'(x) \right| \leq Ch^2$$

This defines a second order consistent approximation to u' .

3.1.3 Approximation of the Second Derivative

We suppose u is a C^4 continuous function on an interval $[x-h_0, x+h_0]$, $h_0 > 0$. Then, there exists a constant $C > 0$ such that for every $h \in]0, h_0[$ we have:

$$\left| \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} - u''(x) \right| \leq Ch^2 \quad (3.4)$$

The differential quotient $\frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$ is a consistent second-order approximation of the second derivative u'' of u at point x .

To prove that, we use Taylor expansions up to the 4th order to achieve the result:

$$u(x + h) = u(x) + hu'(x) + \frac{h^2}{2} u''(x) + \frac{h^3}{6} u'''(x) + \frac{h^4}{24} u''''(x)$$

$$u(x - h) = u(x) - hu'(x) + \frac{h^2}{2} u''(x) - \frac{h^3}{6} u'''(x) + \frac{h^4}{24} u''''(x)$$

Like previously, we write:

$$\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = u''(x) + \frac{h^2}{12} u''''(x)$$

3.1.4 Finite Difference Formulation for a One-Dimensional Problem

We consider a bounded domain $\Omega =]0, 1[\subset \mathbf{R}$ and the non-homogeneous Dirichlet problem:

$$\mathfrak{D} \begin{cases} -u''(x) + c(x)u(x) = f(x) \\ u(0) = \alpha, \quad u(1) = \beta \end{cases} \quad (3.5)$$

Where c and f are two given functions, defined on Ω , $c \geq 0$.

We introduce the equally distributed grid points $(x_j)_{0 \leq j \leq N+1}$ given by $x_j = jh$, where N is an integer and the spacing h is given by $h = 1/(N+1)$. Typically, the spacing becomes very small as the number of grid points will become very large. At the boundary of Ω , we have $x_0 = 0$ and $x_{N+1} = 1$.

At each of these points, we are looking for numerical value of the solution, $u_j = u(x_j)$. We impose $u(x_0) = \alpha$ and $u(x_{N+1}) = \beta$ and we use the differential quotient introduced before to approximate the second order derivative of the equation (3.5).

The unknowns of the discrete problem are all the values $u(x_1), \dots, u(x_N)$ and we introduce the vector $u_h \subset \mathbf{R}^N$ of components u_j , for $j \in \{1, \dots, N\}$. The problem is then to find $u_h \in \mathbf{R}^N$, such that $u_i = u(x_i)$, for all $i \in \{1, \dots, N\}$, where u is the solution of problem (3.5). Introducing the approximation of the second order derivative by a differential quotient, we consider the following discrete problem:

$$-\frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + C(x_j)u_j = f(x_j), \quad j \in \{1, \dots, N\} \quad (3.6)$$

$$u_0 = \alpha \quad , \quad u_{N+1} = \beta$$

This problem has been discretized by a finite difference method based on a three-point centered scheme for the second-order derivative. The problem (3.6) can be written in the matrix form as:

$$A_h u_h = b_h$$

3.2 Finite Element Method

The Finite Element Method is a numerical method for solving differential equations and integrals, and it is primarily used for solving problem in applied engineering and science. The Finite Element Method [99] is a generalization of the well-established variation approach, which is based on the idea that the solution u of a differential equation can be approximated by a linear combination of the parameter c_j and appropriate functions Φ_j :

$$u \approx u_N = \sum_{j=1}^N c_j \Phi_j(x) \tag{3.7}$$

In Eq. (3.7), u stands for the exact and u_N for the approximated (with FEM) solution. Thus, u stands for the vector of the unknown.

The parameters c_j are generally determined with the help of a weighted integral, so that they are a solution of the differential equation of the problem. When selection the functions Φ_j , also called approximation or interpolation functions, it is important that they meet the boundary conditions. There are different methods for the variation approaches like for example the Rayleigh-Ritz method or the method of weighted residuals, while the latter can further be distinguished into the Galerkin method, the least square method, and so on. The

mentioned methods mainly differ in the choice of the weighting function ψ and the approximation function Φ . The Galerkin method for example requires the weighting function to be equal to the approximation function ($\psi = \Phi$). This is the reason why the introduction emphasizes this method.

3.2.1 Method of Weighted Residuals

Here we refer to the method of weighted residuals with application of the Galerkin method.

$$D(u) = q \quad (3.8)$$

First we have a look at the general differential equation (3.8), for which we want to find the solution u . Here D is a linear operator, in this case a differential operator, and q is some kind of source term. If we substitute the approximation u_N from (3.7) into (3.8), the initial equation is not exactly satisfied anymore, and a remainder, also called residual, is generated.

$$R = D(u_N) - q = D\left(\sum_{j=1}^N c_j \Phi_j(x)\right) - q \neq 0 \quad (3.9)$$

Assuming u to be a function of only x and y (i.e. a two-dimensional, steady problem), the residual R is also a function of x and y , but also of c_j . With help of the method of weighted residuals, the parameters c_j are chosen so that the residual R approaches zero. The weighted integral below has to be solved.

$$\int_{\Omega} R(x, y, c_j) \cdot \psi_i(x, y) dx dy = 0 \quad i = 1, \dots, N \quad (3.10)$$

Integration is over the area Ω (two-dimensional area) and ψ_i are the weighting functions, that are principally different from the approximation functions ϕ_j . Only for the Galerkin method ψ_i and ϕ_j are set equal.

Equations (3.7) to (3.10) are strictly speaking not a finite element formulation. Equation (3.7) has to be modified first:

$$u(x, y) \approx u^e(x, y) = \sum_{j=1}^N u_j^e \psi_j^e(x, y) \quad (3.11)$$

Where $u^e(x, y)$ is an approximation of the solution, $u(x, y)$ over the element Ω_e with the vertex count, u_j^e is the value of the function of $u^e(x, y)$ at the vertex j of the element and $\psi_j^e(x, y)$ is the approximation function for the element. Note that the definition $\psi = \phi$ has already been made in equation (3.11) according to the Galerkin method.

If equation. (3.11) is substituted into (3.10), we obtain the following general expression for the finite element form according to the method of weighted residuals:

$$\int_{\Omega} \left\{ D \left(\sum_{j=1}^N u_j^e \psi_j^e(x, y) \right) - q \right\} \cdot \psi_i(x, y, y) dx dy = 0 \quad i = 1, \dots, N \quad (3.12)$$

3.2.2 Weak Formulation

The weak form of a differential equation is a weighted integral similar as in equation (3.12) and it satisfies as well the differential equation it is based on as the natural boundary conditions it is connected with. The natural boundary conditions are integrated into the weighted integral expression by means of the boundary integrals.

A simple example shall clarify the derivation of the weak form. The goal is the steady temperature distribution $T(x,y)$ in a two-dimensional isotropic medium Ω with the boundary Γ . The following equation (Poisson equation) can be stated to describe this problem:

$$-\left\{\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)\right\}=Q \quad (3.13)$$

Where k is the heat conduction coefficient in direction of the x and y axis and $Q(x,y)$ is the provided heat generation per volume unit (source term).

The following boundary conditions are defined prior to solving the differential equations

$$T = T_s \quad \text{on} \quad \Gamma_T \quad (3.14)$$

$$\left(k\frac{\partial T}{\partial x}n_x + k\frac{\partial T}{\partial y}n_y\right) + q_c = q(s) \quad \text{on} \quad \Gamma_q \quad (3.15)$$

The boundaries Γ_T and Γ_q do not overlap and $\Gamma = \Gamma_q + \Gamma_T$, S is the coordinate along the boundary line, (n_x, n_y) is the unit vector normal to the boundary and q_c is the convective heat conduction defined by:

$$q_c = h_c(T - T_c) \quad (3.16)$$

And h_c is the convective heat conduction coefficient and T_c is the reference temperature. The following finite element approximation is introduced for the variable T (temperature):

$$T(x, y) \approx T^e(x, y) = \sum_{j=1}^N T_j^e \psi_j^e(x, y) \quad (3.17)$$

In analogy to equation (3.12) an integral for the weighted residuals will be derived as one of the first steps for the derivation of the weak form.

$$0 = \int_{\Omega_e} \psi \left\{ -\frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) - \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) - Q \right\} dx dy \quad (3.18)$$

Using chain rule in differentiation:

$$\frac{\partial(gh)}{\partial x} = g \frac{\partial h}{\partial x} + h \frac{\partial g}{\partial x} \Rightarrow -h \frac{\partial g}{\partial x} = g \frac{\partial h}{\partial x} - \frac{\partial(gh)}{\partial x} \quad (3.19)$$

And the gradient theorem:

$$\int_{\Omega_e} \frac{\partial}{\partial x} (gh) dx dy = \oint_{\Gamma_e} (gh) n_x dx \quad (3.20)$$

as well as the substitution:

$$h = k \frac{\partial T}{\partial x} \text{ bzw. } k \frac{\partial T}{\partial y} \quad (3.21)$$

Equation (3.18) can be transformed to:

$$0 = \int_{\Omega_e} \left(k \frac{\partial \psi}{\partial x} \frac{\partial T}{\partial x} + k \frac{\partial \psi}{\partial y} \frac{\partial T}{\partial y} - \psi Q \right) dx dy - \oint_{\Gamma_e} \psi q_n ds \quad (3.22)$$

Where q_n is the heat flux orthogonal to the boundary, which can also be written as:

$$q_n = k \frac{\partial T}{\partial x} n_x + k \frac{\partial T}{\partial y} n_y \quad (3.23)$$

The variables n_x and n_y are the components of the normal vector along the boundary. The integral expression from equation (3.22) is called “weak form” of equation (3.13). If we have a look at equation (3.22), we notice that the order of the differential referring to the variable T is not $m=2$ but $m=1$ in the integral (reduction), but at the same time a differential for y has been added (transport from T to y). Additionally, the natural boundary condition from equation (3.15) has been integrated into the weak form by means of the boundary integral.

In order to transfer the weak form into a finite element form in the next step, the finite element approximation for T from equation (3.17) is substituted into equation (3.22) and we obtain:

$$0 = \int_{\Omega_e} \left\{ \frac{\partial \psi_i}{\partial x} \left(k \sum_{j=1}^n T_j^e \frac{\partial \psi_j^e}{\partial x} \right) + \frac{\partial \psi_i}{\partial y} \left(k \sum_{j=1}^n T_j^e \frac{\partial \psi_j^e}{\partial y} \right) \psi_i Q \right\} dx dy - \oint_{\Gamma_e} \psi_i q_n ds \quad (3.24)$$

After a few modifications, this equation can be written in matrix form

$$[K^e] \{T^e\} = \{Q^e\} + \{q^e\} \quad (3.25)$$

With

$$K_{ij}^e = \int_{\Omega_e} \left(k \frac{\partial \psi_i^e}{\partial x} \frac{\partial \psi_j^e}{\partial x} + k \frac{\partial \psi_i^e}{\partial y} \frac{\partial \psi_j^e}{\partial y} \right) dx dy \quad (3.26)$$

$$Q_i^e = \int_{\Omega_e} Q \psi_i^e dx dy, \quad q_i^e = \oint_{\Gamma_e} q_n \psi_i^e ds \quad (3.27)$$

The matrix $[K^e]$ is the so-called coefficient matrix, also called stiffness matrix in statics. equation (3.25) above is important in our efforts to understand the finite element method, that is to say the (“weak”) finite element form of Eq. (3.13).

3.3 Finite Volume Method

Here we focus on introducing finite volume method for the solution of partial differential equations. These methods have been well accepted in recent years for their robustness, their straightforward formulation, and their computational advantages.

The partial differential equation we will focus on is a scalar equation that represents the transport of a substance under the influence of advection. The transport equation is frequently written in the advective form:

$$\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T = \nabla \cdot (\alpha \nabla T) \quad (3.28)$$

Where T is the substance transported, e.g. temperature, u is the velocity field assumed to be known, and α is the diffusion.

The velocity field cannot be arbitrary and must satisfy mass conservation equation. If flow assumed to be incompressible, mass conservation equation reduces to:

$$\nabla \cdot \vec{u} = 0 \quad (3.29)$$

The advective form can be interpreted as the time evolution of the T field along characteristic lines given by u . It is thus closest to a Lagrangian description of the flow where one follows individual particles. In the Eulerian frame, however, another important issue is the conservation of the tracer T for long period of times. A slightly different form of the equation called the conservative form can be derived and forms the starting point for the derivation of finite volume methods. Multiplying the continuity equation by T , adding it to the resultant equations to the advective form, we can derive the conservative form of the transport diffusion equation:

$$\frac{\partial T}{\partial t} + \nabla \cdot (\vec{u}T) = \nabla \cdot (\alpha \nabla T) \quad (3.30)$$

3.3.1 Integral Form of Conservation Law

The partial differential equation is valid at all points in the domain which we could consider as infinitesimal volumes. Anticipating that infinitesimal discrete volumes are unaffordable and would have to be “inflated” to a finite size, we proceed to derive the conservative form for a finite volume δV bounded by a surface δS . The integral over (3.30) yields:

$$\int_{\delta V} \frac{\partial T}{\partial t} dV + \int_{\delta V} \nabla \cdot (\vec{u}T) dV = \int_{\delta V} \nabla \cdot (\alpha \nabla T) dV \quad (3.31)$$

$$\frac{d}{dt} \left(\int_{\delta V} T dV \right) + \int_{\delta S} \vec{n} \cdot \vec{u}T dV = \int_{\delta S} \vec{n} \cdot \alpha \nabla T dV \quad (3.32)$$

Remarks

- We assume the volume δV to be fixed in space so we can interchange the order of integration in space and differentiation in time. The interpretation of the first integral on the left hand side of equation (3.32) is now simple: it is the time rate of change of the T budget inside volume δV .
- We have used the Gauss-divergence theorem to change the volume integrals of the flux and diffusion divergence into surface integrals. Here n is the outward unit normal to the surface δS . The surface integral on the left hand side accounts for the advective flux carrying T in and out of the volume δV across the surface δS ; the one on the right hand side accounts for the diffusive transport of T across δS .
- Equation (3.32) lends itself to simple physical interpretation: the rate of change of the T budget in δV is equal to the rate of transport of T through δS by advective fluxes (transport by the flow, wind, current) and diffusive fluxes. If the volume δV is closed to advection or diffusion: $\vec{n} \cdot \vec{u}=0$ and $\vec{n} \cdot \nabla T = 0$, then the rate of change is zero and the budget of T is conserved within δV .
- If there are additional physical processes affecting the budget of T , such as sources or sinks within δV , these should be accounted for also. No additional process is considered here.

• The derivatives appearing in the differential equation (3.28) are: first order for the advective term, and second order for the diffusion term, whereas the equivalent terms in the integral form have only a zero order derivative and first order derivative respectively. This lowering of the derivative order is important in dealing with solution which change so rapidly in space that the spatial derivative does not exist discontinuous function do not have derivatives at the location of discontinuity and mathematically speaking the partial differential form of the conservation equation is invalid there even though the conservation law underlying it is still valid. Special treatment is generally required for treating discontinuities and reducing order of the spatial derivative helps simplify the special treatment.

For these reasons Finite Volumes are preferred over finite differences to solve problems whose solution exhibit local discontinuities.

A slightly different form of equation (3.32) can be derived by introducing the average of T in δV and which we refer to as \bar{T} and is defined as:

$$\bar{T} = \frac{1}{\delta V} \int_{\delta V} T dV \quad (3.33)$$

The integral conservation law can now be recast as a time evolution equation for \bar{T}

$$\frac{d}{dt} \left(\int_{\delta V} T dV \right) + \int_{\delta S} \vec{n} \cdot \vec{u} T dV = \int_{\delta S} \vec{n} \cdot \alpha \nabla T dV \quad (3.34)$$

3.4 Discontinuous Galerkin

The governing MHD equations is discretized using a discontinuous Galerkin finite element formulation. To formulate the discontinuous Galerkin method, we first introduce the

following weak formulation, which is obtained by multiplying the $\frac{\partial Q}{\partial t} + \frac{\partial F_j}{\partial x_j} = 0$ by a test function W , integrating over the domain Ω , and then performing an integration by parts,

$$\int_{\Omega} \frac{\partial \mathbf{U}}{\partial t} \mathbf{W} d\Omega + \int_{\Gamma} \mathbf{F}_k \mathbf{n}_k d\Gamma - \int_{\Omega} \mathbf{F}_k \frac{\partial \mathbf{W}}{\partial x_k} d\Omega = 0, \quad \forall \mathbf{W} \in V \quad (3.35)$$

Where $\Gamma(=\partial\Omega)$ denotes the boundary of Ω , and n_j the unit outward normal vector to the boundary. We assume that the domain Ω is subdivided into a collection of non-overlapping elements Ω_e , which can be triangles, quadrilaterals, polygons, or their combinations in 2D and tetrahedra, prisms, pyramids, and hexahedra or their combinations in 3D. We introduce the following broken Sobolev space V_h^p

$$V_h^p = \left\{ v_h \in [L_2(\Omega)]^m : v_h|_{\Omega_e} \in [V_p^m] \forall \Omega_e \in \Omega \right\} \quad (3.36)$$

Which consists of discontinuous vector-values polynomial functions of degree p , and where m is the dimension of the unknown vector and

$$V^m = \text{span} \left\{ \prod x_i^{\alpha_i} : 0 \leq \alpha_i \leq p, 0 \leq i \leq d \right\} \quad (3.37)$$

Where α denotes a multi-index and d is the dimension of space. Then, we can obtain the following semi-discrete form by applying weak formulation on each element Ω_e . Find $U_h \in V_h^p$ such as

$$\frac{d}{dt} \int_{\Omega_e} \mathbf{U}_h \mathbf{W}_h d\Omega + \int_{\Gamma_e} \mathbf{F}_k(\mathbf{U}_h) \mathbf{n}_k \mathbf{W}_h d\Gamma - \int_{\Omega_e} \mathbf{F}_k(\mathbf{U}_h) \frac{\partial \mathbf{W}_h}{\partial x_k} d\Omega =, \quad \forall \mathbf{W}_h \in V_h^p \quad (3.38)$$

where U_h and W_h represent the finite element approximations to the analytical solution U and the test function W respectively, and they are approximated by a piecewise polynomial function of degrees p , which are discontinuous between the cell interfaces. Assume that B is

the basis of polynomial function of degrees p , this is then equivalent to the following system of N equations

$$\frac{d}{dt} \int_{\Omega_e} \mathbf{U}_h B_i d\Omega + \int_{\Gamma_e} \mathbf{F}_k(\mathbf{U}_h) \mathbf{n}_k B_i d\Gamma - \int_{\Omega_e} \mathbf{F}_k(\mathbf{U}_h) \frac{\partial B_i}{\partial x_k} d\Omega =, \quad 1 \leq i \leq N \quad (3.39)$$

Where N is dimension of the polynomial space. Since the numerical solution U_h is discontinuous between element interfaces, the interface fluxes are not uniquely defined. The choice of these fluxes is crucial for the DG formulation. Like in the finite volume methods, the flux function $\mathbf{F}_k(\mathbf{U}_h)\mathbf{n}_k$ appearing in the boundary integral can be replaced by a numerical Riemann flux function $\mathbf{H}_k(\mathbf{U}_h^L, \mathbf{U}_h^R, \mathbf{n}_k)$ where \mathbf{U}_h^L and \mathbf{U}_h^R are the conservative state vector at the left and right side of the element boundary. The computation of the viscous fluxes in the boundary integral has to properly resolve the discontinuities at the interfaces. This scheme is called discontinuous Galerkin method of degree p , or in short notation DG(P) method. Note that discontinuous Galerkin formulations are very similar to finite volume schemes, especially in their use of numerical fluxes. Indeed, the classical first-order cell-centered finite volume scheme exactly corresponds to the DG(P₀) method, i.e., to the discontinuous Galerkin method using a piecewise constant polynomial. Consequently, the DG(P_k) methods with $k > 0$ can be regarded as a natural generalization of finite volume methods to higher order methods. By simply increasing the degree P of the polynomials, the DG methods of corresponding higher order can be obtained.

In the traditional DGM, numerical polynomial solutions U_h in each element are expressed using either standard Lagrange finite element or hierarchical node-based basis as following

$$\mathbf{U}_h = \sum_{i=1}^N \mathbf{U}_i(t) B_i(x), \quad (3.40)$$

where B_i are the finite element basis functions. As a result, the unknowns to be solved are the variables at the nodes U_i , as illustrated in Figure 3.1 for linear and quadratic polynomial approximations.

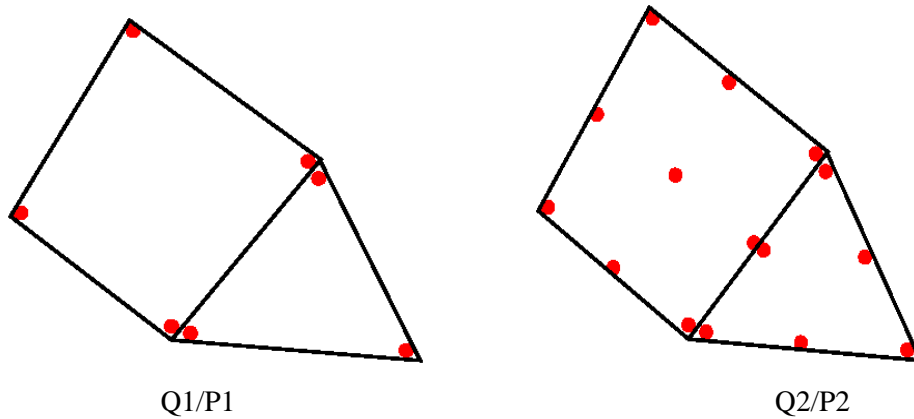


Figure 3.1: Representation of polynomial solutions using finite element shape functions

On each cell, a system of $N \times N$ has to be solved, where polynomial solutions are dependent on the shape of elements. For example, for a linear polynomial approximation in 2D as shown in Figure 3.1, a linear polynomial is used for triangular elements and the unknowns to be solved are the variables at the three vertices and a bi-linear polynomial is used for quadrilateral elements and the unknowns to be solved are the variables at the four vertices. However, numerical polynomial solutions U can be expressed in other forms as well.

3.4.1 Discontinuous Galerkin method using a Taylor Basis

In the present work, the numerical polynomial solutions are represented using a Taylor series expansion at the center of the cell. For example, if we do a Taylor series expansion at the cell centroid, the quadratic polynomial solutions can be expressed as follows

$$\begin{aligned} \mathbf{U}_h = & \mathbf{U}_c + \frac{\partial \mathbf{U}}{\partial x} /_c (x - x_c) + \frac{\partial \mathbf{U}}{\partial y} /_c (y - y_c) + \frac{\partial^2 \mathbf{U}}{\partial x^2} /_c \frac{(x - x_c)^2}{2} \\ & + \frac{\partial^2 \mathbf{U}}{\partial y^2} /_c \frac{(y - y_c)^2}{2} + \frac{\partial^2 \mathbf{U}}{\partial x \partial y} /_c (x - x_c)(y - y_c) \end{aligned} \quad (3.41)$$

Which can be further expressed as cell-averaged values and their derivatives at the center of the cell

$$\begin{aligned} \mathbf{U}_h = & \tilde{\mathbf{U}} + \frac{\partial \mathbf{U}}{\partial x} /_c (x - x_c) + \frac{\partial \mathbf{U}}{\partial y} /_c (y - y_c) + \frac{\partial^2 \mathbf{U}}{\partial x^2} /_c \left(\frac{(x - x_c)^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(x - x_c)^2}{2} d\Omega \right) \\ & + \frac{\partial^2 \mathbf{U}}{\partial y^2} /_c \left(\frac{(y - y_c)^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(y - y_c)^2}{2} d\Omega \right) + \frac{\partial^2 \mathbf{U}}{\partial x \partial y} /_c \left((x - x_c)(y - y_c) - \frac{1}{\Omega_e} \int_{\Omega_e} (x - x_c)(y - y_c) d\Omega \right) \end{aligned} \quad (3.42)$$

where \tilde{U} is the mean value of U in this cell. The unknowns to be solved in this formulation are the cell-averaged variables and their derivatives at the center of the cells, regardless of element shapes, as shown in Figure 3.2.

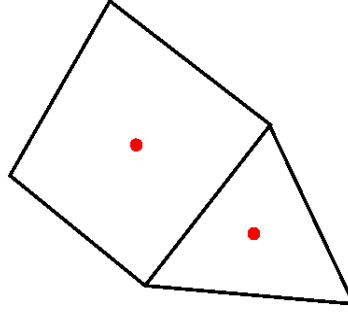


Figure 3.2 Representation of polynomial solutions using a Taylor series expansion

In this case, the dimension of the polynomial space is six and the six basis functions are

$$B_1 = 1$$

$$B_2 = x - x_c$$

$$B_3 = y - y_c$$

$$B_4 = \frac{(x - x_c)^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(x - x_c)^2}{2} d\Omega \quad (3.43)$$

$$B_5 = \frac{(y - y_c)^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(y - y_c)^2}{2} d\Omega$$

$$B_6 = (x - x_c)(y - y_c) - \frac{1}{\Omega_e} \int_{\Omega_e} (x - x_c)(y - y_c) d\Omega$$

The discontinuous Galerkin formulation then leads to the following six equations

$$\frac{d}{dt} \int_{\Omega_e} \tilde{\mathbf{U}} d\Omega + \int_{\Gamma_e} \mathbf{F}_k(\mathbf{U}_h) \mathbf{n}_k d\Gamma = 0, \quad i = 1 \quad (3.44)$$

$$M_{5 \times 5} \frac{d}{dt} \left(\frac{\partial \mathbf{U}}{\partial x} /_c \quad \frac{\partial \mathbf{U}}{\partial y} /_c \quad \frac{\partial^2 \mathbf{U}}{\partial x^2} /_c \quad \frac{\partial^2 \mathbf{U}}{\partial y^2} /_c \quad \frac{\partial^2 \mathbf{U}}{\partial x \partial y} /_c \right)^T + R_{5 \times 1} = 0$$

Note that in this formulation, equations for the cell-averaged variables are decoupled from equations for their derivatives due to the judicious choice of the basis functions and the fact that

$$\int_{\Omega_e} B_i B_j d\Omega = 0, \quad 2 \leq i \leq 6 \quad (3.45)$$

In the implementation of this DG method, the basis functions are actually normalized in order to improve the conditioning of the system matrix (3.44) as follow

$$B_1 = 1$$

$$B_2 = \frac{x - x_c}{\Delta x}$$

$$B_3 = \frac{y - y_c}{\Delta y}$$

$$B_4 = \frac{(x - x_c)^2}{2\Delta x^2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(x - x_c)^2}{2\Delta x^2} d\Omega \quad (3.46)$$

$$B_5 = \frac{(y - y_c)^2}{2\Delta y^2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(y - y_c)^2}{2\Delta y^2} d\Omega$$

$$B_6 = \frac{(x - x_c)(y - y_c)}{\Delta x \Delta y} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(x - x_c)(y - y_c)}{\Delta x \Delta y} d\Omega$$

Where $\Delta x=0.5(x_{\max}-x_{\min})$, and $\Delta y=0.5(y_{\max}-y_{\min})$, and x_{\max} , x_{\min} , y_{\max} , and y_{\min} are the maximum and minimum coordinates in the cell Ω_e in x , and y directions, respectively. A quadratic polynomial solution can then be rewritten as

$$\mathbf{U}_h = \tilde{\mathbf{U}} + \frac{\partial \mathbf{U}}{\partial x} /_c \Delta x B_2 + \frac{\partial \mathbf{U}}{\partial y} /_c \Delta y B_3 + \frac{\partial^2 \mathbf{U}}{\partial x^2} /_c \Delta x^2 B_4 + \frac{\partial^2 \mathbf{U}}{\partial y^2} /_c \Delta y^2 B_5 + \frac{\partial^2 \mathbf{U}}{\partial x \partial y} /_c \Delta x \Delta y B_6 \quad (3.47)$$

The above normalization is especially important to alleviate the stiffness of the system matrix for higher-order DG approximations.

This Taylor-basis DG method has a number of attractive features. Theoretically, this formulation allows us to clearly see the similarity and difference between DG and FV methods. In fact, the discretized governing equations for the cell-averaged variables and the assumption of polynomial solutions on each cell are exactly the same for both finite volume and DG methods. The only difference between them is the way how they obtain high-order ($p>1$) polynomial solutions. In the finite volume methods, the polynomial solution of degrees p are reconstructed using the mean values of the neighboring cells, which can be obtained using either TVD/MUSCL or ENO/WENO reconstruction schemes. Unlike the FV methods, the DG methods compute the derivatives in a manner similar to the mean variables. This is compact, rigorous, and elegant mathematically in contrast with arbitrariness characterizing the reconstruction schemes with respect how to compute the derivatives and how to choose the stencils in the FV methods. Furthermore, the higher order DG methods can be easily constructed by simply increasing the degree p of the polynomials locally, in contrast to the finite volume methods which use the extended stencils to achieve higher order of accuracy.

3.4.2 Discontinuous Galerkin Method DG(P₁)

The formulation of the Discontinuous Galerkin method DG(P₁) is described here, where the independent variable Q is approximated using a linear polynomial as:

$$Q_h = \sum_{i=1}^n Q_i(t) B_i(x, y, z) \quad (3.48)$$

where Q_i 's are the time dependent coefficients of the series expansion to be determined, and B_i 's are the Taylor basis functions as follow:

$$\begin{aligned} B_1 &= 1 \\ B_2 &= x - x_c \\ B_3 &= y - y_c \\ B_4 &= z - z_c \end{aligned} \quad (3.49)$$

Applying DG(P₁) over system of conservation law leads us to

$$\frac{\partial Q}{\partial t} + \frac{\partial F_j}{\partial X_j} = 0 \quad j=1,2,3 \quad (3.50)$$

$$\int_{\Omega} \left(\frac{\partial Q}{\partial t} + \frac{\partial F_j}{\partial X_j} \right) d\Omega = 0 \quad (3.51)$$

For first base function as mentioned earlier at equations (3.44) and (3.45) we will have

$$\frac{d}{dt} \int_{\Omega} Q_h B_1 d\Omega + \int_{\Gamma} F_j \cdot n_j B_1 d\Gamma = 0 \quad (3.52)$$

for cell averaged values, and

$$\frac{d}{dt} \int_{\Omega} Q_h B_i d\Omega + \int_{\Gamma} F_j \cdot n_j B_i d\Gamma - \int_{\Omega} F_j \frac{\partial B_i}{\partial X_j} d\Omega = 0 \quad i=2,3,4 \quad (3.53)$$

for first order derivatives. In the above equations F_i 's are the inviscid MHD fluxes. From governing equations of MHD, i.e., equations (10)(2.9) and (10)(2.10) we have:

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \\ B_x \\ B_y \\ B_z \end{pmatrix} \quad \mathbf{F}_n = \mathbf{F}_j \cdot \mathbf{n}_j = \begin{pmatrix} \rho v_n \\ \rho u v_n - B_x B_n + P_T n_x \\ \rho v v_n - B_y B_n + P_T n_y \\ \rho w v_n - B_z B_n + P_T n_z \\ (\rho E + P_T) v_n - B_n (\mathbf{U} \cdot \mathbf{B}) \\ v_n B_x - u B_n \\ v_n B_y - v B_n \\ v_n B_z - w B_n \end{pmatrix} \quad (3.54)$$

where $F_{i, i=1,2,3}$ is as follow:

$$F_1 = F, \quad F_2 = G, \quad F_3 = H, \quad (3.55)$$

The weak formulation (3.53) leads to the system of three equations and three unknowns for first order spatial derivatives

$$M_{3 \times 3} \frac{d}{dt} \begin{pmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial x} \end{pmatrix} + \begin{pmatrix} \int_{\Gamma} F_n B_2 d\Gamma - \int \mathbf{F} \frac{\partial B_2}{\partial x} d\Omega \\ \int_{\Gamma} F_n B_3 d\Gamma - \int \mathbf{G} \frac{\partial B_3}{\partial y} d\Omega \\ \int_{\Gamma} F_n B_4 d\Gamma - \int \mathbf{H} \frac{\partial B_4}{\partial z} d\Omega \end{pmatrix} = 0 \quad (3.56)$$

where local mass matrix M is 3×3 and elements of M are defined as:

$$M_{i,j} = \int_{\Omega} B_{i+1} B_{j+1} d\Omega \quad (3.57)$$

Ω is the cell volume, Γ is the cell surface, and F_n is the numerical flux normal to the cell surface. All the integration here is done by numerical technique and appropriate Gaussian quadratures.

3.4.3 Reconstructed Discontinuous Galerkin Method

The Discontinuous Galerkin method is more expensive in terms of computational costs and storage requirements in comparison with the Finite Volume method since it has more degrees of freedom, additional domain integration, and more Gauss quadrature points for the boundary integration. The FVM uses the reconstruction methods to achieve higher-order accuracy that is relatively inexpensive but less accurate and robust. The DGM that can be viewed as a different way to extend a FV method to higher orders are accurate and robust but expensive. The idea was originally introduced by Dumbser et al [72][73][74] in the frame of P_nP_m scheme, termed $RDG(P_nP_m)$ to combine the efficiency of the reconstruction methods and the accuracy of the DG methods. In which P_n indicates that a piecewise polynomial of degree of n is used to represent a DG solution, and P_m represents a reconstructed polynomial solution of degree of m ($m \geq n$) that is used to compute the fluxes and source terms. The beauty of $RDG(P_nP_m)$ schemes is that they provide a unified formulation for both finite volume and DG methods, and contain both classical finite volume and standard DG methods as two special cases of $RDG(P_nP_m)$ schemes, and thus allow for a direct efficiency comparison. When $n=0$, $RDG(P_0P_m)$ is nothing but classical high order finite volume schemes, where a polynomial solution of degree m ($m \geq 1$) is reconstructed from a piecewise constant solution. When $m=n$, the reconstruction reduces to the identity operator, and $RDG(P_nP_m)$ scheme yields a standard DG method. Clearly, an accurate and efficient reconstruction is the key ingredient in extending the underlying DG method to higher order accuracy. Discussion in this research is mainly focused on a third order $RDG(P_1P_2)$ method, as the benefits of higher-order (>3rd) methods diminish dramatically for engineering applications. Nevertheless, its extension to higher order DG methods is straightforward. The $RDG(P_1P_2)$ method is based on a hierarchical Hermite WENO reconstruction and designed not only to reduce the high computing costs of the DGM, but also to avoid spurious oscillations in the vicinity of strong discontinuities, thus ensuring the nonlinear stability of the RDG method. Similar to moment limiters, the hierarchical reconstruction methods[84] reconstruct the derivatives in a hierarchical manner. In the case of the $RDG(P_1P_2)$ method,

the second derivatives (curvatures) are first reconstructed and the first derivatives (gradients) are then reconstructed.

We use Taylor expansion in our RDG that in general multidimensional case expressed as follow:

$$f(x_1, x_2, \dots, x_n) = \sum_{j=0}^{\infty} \frac{1}{j!} \left[\sum_{k=1}^n (x_k - a_k) \frac{\partial}{\partial x_k} \right]^j f(a_1, a_2, \dots, a_n) \quad (3.58)$$

In which, $x_i, i=1,2,\dots,n$, are the independent variables, and $a_i, i=1,2,\dots,n$, are the coordinate of the point in which we expand function around it. For two dimensional space, $n=2$ and we will have the following relation for quadratic Taylor expansion

$$\begin{aligned} f(x, y) &= f(x_0, y_0) \\ &+ \frac{\partial f}{\partial x} \Big|_{x_0, y_0} (x - x_0) + \frac{\partial f}{\partial y} \Big|_{x_0, y_0} (y - y_0) \\ &+ \frac{\partial^2 f}{\partial x^2} \Big|_{x_0, y_0} \frac{(x - x_0)^2}{2} + \frac{\partial^2 f}{\partial y^2} \Big|_{x_0, y_0} \frac{(y - y_0)^2}{2} + \frac{\partial^2 f}{\partial xy} \Big|_{x_0, y_0} \frac{(x - x_0)(y - y_0)}{2} \\ &+ \text{Higher order terms (3th order)} \end{aligned} \quad (3.59)$$

And for three dimensional space, $n=3$, and Taylor expansion of function f is as follow

$$\begin{aligned} f(x, y, z) &= f(x_0, y_0, z_0) \\ &+ \frac{\partial f}{\partial x} \Big|_{x_0, y_0, z_0} (x - x_0) + \frac{\partial f}{\partial y} \Big|_{x_0, y_0, z_0} (y - y_0) + \frac{\partial f}{\partial z} \Big|_{x_0, y_0, z_0} (z - z_0) \\ &+ \frac{\partial^2 f}{\partial x^2} \Big|_{x_0, y_0, z_0} \frac{(x - x_0)^2}{2} + \frac{\partial^2 f}{\partial y^2} \Big|_{x_0, y_0, z_0} \frac{(y - y_0)^2}{2} + \frac{\partial^2 f}{\partial z^2} \Big|_{x_0, y_0, z_0} \frac{(z - z_0)^2}{2} \\ &+ \frac{\partial^2 f}{\partial xy} \Big|_{x_0, y_0, z_0} \frac{(x - x_0)(y - y_0)}{2} + \frac{\partial^2 f}{\partial xz} \Big|_{x_0, y_0, z_0} \frac{(x - x_0)(z - z_0)}{2} \end{aligned} \quad (3.60)$$

$$+ \frac{\partial^2 f}{\partial yz} \Big|_{x_0, y_0, z_0} \frac{(y - y_0)(z - z_0)}{2} + \text{higher order terms(3th order)}$$

But if we use equation (3.60), we see that it is not conservative, i.e., if we take integral from both sides, the resulting term is not equal.

$$\begin{aligned}
\int_{\Omega} f(x, y, z) d\Omega &= \int_{\Omega} f(x_0, y_0, z_0) d\Omega \\
&+ \frac{\partial f}{\partial x} \Big|_{x_0, y_0, z_0} \underbrace{\int_{\Omega} (x - x_0) d\Omega}_{=0} + \frac{\partial f}{\partial y} \Big|_{x_0, y_0, z_0} \underbrace{\int_{\Omega} (y - y_0) d\Omega}_{=0} \\
&+ \frac{\partial f}{\partial z} \Big|_{x_0, y_0, z_0} \underbrace{\int_{\Omega} (z - z_0) d\Omega}_{=0} + \frac{\partial^2 f}{\partial x^2} \Big|_{x_0, y_0, z_0} \int_{\Omega} \left(\frac{(x - x_0)^2}{2} \right) d\Omega \\
&+ \frac{\partial^2 f}{\partial y^2} \Big|_{x_0, y_0, z_0} \int_{\Omega} \left(\frac{(y - y_0)^2}{2} \right) d\Omega + \frac{\partial^2 f}{\partial z^2} \Big|_{x_0, y_0, z_0} \int_{\Omega} \left(\frac{(z - z_0)^2}{2} \right) d\Omega \\
&+ \frac{\partial^2 f}{\partial xy} \Big|_{x_0, y_0, z_0} \int_{\Omega} \left(\frac{(x - x_0)(y - y_0)}{2} \right) d\Omega \\
&+ \frac{\partial^2 f}{\partial xz} \Big|_{x_0, y_0, z_0} \int_{\Omega} \left(\frac{(x - x_0)(z - z_0)}{2} \right) d\Omega \\
&+ \frac{\partial^2 f}{\partial yz} \Big|_{x_0, y_0, z_0} \int_{\Omega} \left(\frac{(y - y_0)(z - z_0)}{2} \right) d\Omega \\
&+ \text{higher order terms(3th order)}
\end{aligned} \tag{3.61}$$

So the result is as follow

$$\begin{aligned}
\int_{\Omega} f(x, y, z) d\Omega &= f(x_0, y_0, z_0) \Omega \\
&+ \frac{\partial^2 f}{\partial x^2} \Big|_{x_0, y_0, z_0} \int_{\Omega} \left(\frac{(x - x_0)^2}{2} \right) d\Omega
\end{aligned} \tag{3.62}$$

$$\begin{aligned}
& + \frac{\partial^2 f}{\partial y^2} \Big|_{x_0, y_0, z_0} \int_{\Omega} \left(\frac{(y - y_0)^2}{2} \right) d\Omega \\
& + \frac{\partial^2 f}{\partial z^2} \Big|_{x_0, y_0, z_0} \int_{\Omega} \left(\frac{(z - z_0)^2}{2} \right) d\Omega \\
& + \frac{\partial^2 f}{\partial xy} \Big|_{x_0, y_0, z_0} \int_{\Omega} \left(\frac{(x - x_0)(y - y_0)}{2} \right) d\Omega \\
& + \frac{\partial^2 f}{\partial xz} \Big|_{x_0, y_0, z_0} \int_{\Omega} \left(\frac{(x - x_0)(z - z_0)}{2} \right) d\Omega \\
& + \frac{\partial^2 f}{\partial yz} \Big|_{x_0, y_0, z_0} \int_{\Omega} \left(\frac{(y - y_0)(z - z_0)}{2} \right) d\Omega \\
& + \text{higher order terms (3th order)}
\end{aligned}$$

Here we change basis function for second order terms and use cell average value for first term, in this way conservation is satisfied. So function u can be reconstructed using 3rd order Taylor expansion using averaged values in each cell, i.e. \tilde{u}_i^R is average of function u at cell i .

$$\begin{aligned}
u_i^R &= \tilde{u}_i^R + u_{xi}^R B_2 \Delta x + u_{yi}^R B_3 \Delta y + u_{zi}^R B_4 \Delta z \\
&+ u_{xxi}^R B_5 \Delta x^2 + u_{yyi}^R B_6 \Delta y^2 + u_{zzi}^R B_7 \Delta z^2 + u_{xyi}^R B_8 \Delta x \Delta y + u_{xzi}^R B_9 \Delta x \Delta z \\
&+ u_{yzi}^R B_{10} \Delta y \Delta z
\end{aligned} \tag{3.63}$$

where the shape functions are expressed as follow:

$$\begin{aligned}
B_1 &= 1 \\
B_2 &= \frac{x - x_{ci}}{\Delta x} \\
B_3 &= \frac{y - y_{ci}}{\Delta y}
\end{aligned} \tag{3.64}$$

$$B_4 = \frac{z - z_{ci}}{\Delta z}$$

$$B_5 = \frac{(x - x_{ci})^2}{2\Delta x^2} - \frac{1}{\Omega} \int_{\Omega} \frac{(x - x_{ci})^2}{2\Delta x^2} d\Omega$$

$$B_6 = \frac{(y - y_{ci})^2}{2\Delta y^2} - \frac{1}{\Omega} \int_{\Omega} \frac{(y - y_{ci})^2}{2\Delta y^2} d\Omega$$

$$B_7 = \frac{(z - z_{ci})^2}{2\Delta z^2} - \frac{1}{\Omega} \int_{\Omega} \frac{(z - z_{ci})^2}{2\Delta z^2} d\Omega$$

$$B_8 = \frac{(x - x_{ci})(y - y_{ci})}{\Delta x \Delta y} - \frac{1}{\Omega} \int_{\Omega} \frac{(x - x_{ci})(y - y_{ci})}{\Delta x \Delta y} d\Omega$$

$$B_9 = \frac{(x - x_{ci})(z - z_{ci})}{\Delta x \Delta z} - \frac{1}{\Omega} \int_{\Omega} \frac{(x - x_{ci})(z - z_{ci})}{\Delta x \Delta z} d\Omega$$

$$B_{10} = \frac{(y - y_{ci})(z - z_{ci})}{\Delta y \Delta z} - \frac{1}{\Omega} \int_{\Omega} \frac{(y - y_{ci})(z - z_{ci})}{\Delta y \Delta z} d\Omega$$

3.4.4 WENO reconstruction at P_2 : HWENO(P_1P_2)

The reconstruction of the curvatures consists of two steps: a quadratic polynomial solution (P_2) is first reconstructed using a least-squares method from the underlying linear polynomial (P_1) discontinuous Galerkin solution, and the final quadratic polynomial solution is then obtained using a WENO reconstruction, which is necessary to ensure the linear stability of the RDG method [80]. The resulting RDG method is referred to as WENO(P_1P_2) in this work, where the quadratic polynomial solution is obtained from the underlying linear DG solution via a WENO reconstruction.

For reconstruction of second order derivative, since we have average value and first order derivatives in hand, we need to reconstruct u_j^R in which $j=1,\dots,m$ are the adjacent cells to cell i

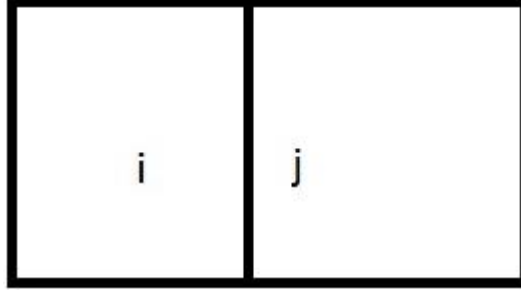


Figure 3.3 Two neighboring cells i and j for reconstruction

At point C_j (centroid of cell j) we can reconstruct point wise values of variable u and its first order derivatives using values of cell i

$$\begin{aligned}
 u_j^R &= \tilde{u}_i^R + u_{xi}^R B_2^j \Delta x_i + u_{yi}^R B_3^j \Delta y_i + u_{zi}^R B_4^j \Delta z_i + \\
 &u_{xxi}^R B_5^j \Delta x_i^2 + u_{yyi}^R B_6^j \Delta y_i^2 + u_{zzi}^R B_7^j \Delta z_i^2 + u_{xyi}^R B_8^j \Delta x_i \Delta y_i + u_{xzi}^R B_9^j \Delta x_i \Delta z_i + u_{yzi}^R B_{10}^j \Delta y_i \Delta z_i \\
 \frac{\delta u_j^R}{\delta x} &= u_{xi}^R + u_{xxi}^R B_2^j \Delta x + u_{xyi}^R B_3^j \Delta y + u_{xzi}^R B_4^j \Delta z \\
 \frac{\delta u_j^R}{\delta y} &= u_{yi}^R + u_{yyi}^R B_3^j \Delta y + u_{xyi}^R B_2^j \Delta x + u_{yzi}^R B_4^j \Delta z \\
 \frac{\delta u_j^R}{\delta z} &= u_{zi}^R + u_{zzi}^R B_4^j \Delta z + u_{xzi}^R B_2^j \Delta x + u_{yzi}^R B_3^j \Delta y
 \end{aligned} \tag{3.65}$$

For non-dimensionalization of coefficient matrix we multiply both sides of equations of derivatives by Δx_j :

$$\Delta x_j \left(\frac{\delta u_j^R}{\delta x} \right) = \Delta x_j (u_{xi}^R + u_{xxi}^R B_2^j \Delta x + u_{xyi}^R B_3^j \Delta y + u_{xzi}^R B_4^j \Delta z)$$

$$\Delta x_j \left(\frac{\delta u_j^R}{\delta y} \right) = \Delta x_j (u_{yi}^R + u_{yyi}^R B_3^j \Delta y + u_{xyi}^R B_2^j \Delta x + u_{yzi}^R B_4^j \Delta z)$$

$$\Delta x_j \left(\frac{\delta u_j^R}{\delta z} \right) = \Delta x_j (u_{zi}^R + u_{zzi}^R B_4^j \Delta z + u_{xzi}^R B_2^j \Delta x + u_{yzi}^R B_3^j \Delta y)$$

Putting together all the 4 equalities we have system of four equations and six unknowns for second order derivatives at cell i obtained from cell j :

$$\begin{bmatrix} B_5^j & B_6^j & B_7^j & B_8^j & B_9^j & B_{10}^j \\ \frac{\Delta x_j}{\Delta x_i} B_2^j & 0 & 0 & \frac{\Delta x_j}{\Delta x_i} B_3^j & \frac{\Delta x_j}{\Delta x_i} B_4^j & 0 \\ 0 & \frac{\Delta y_j}{\Delta y_i} B_3^j & \frac{\Delta y_j}{\Delta y_i} B_2^j & 0 & 0 & \frac{\Delta y_j}{\Delta y_i} B_4^j \\ 0 & 0 & \frac{\Delta z_j}{\Delta z_i} B_4^j & 0 & \frac{\Delta z_j}{\Delta z_i} B_2^j & \frac{\Delta z_j}{\Delta z_i} B_3^j \end{bmatrix} \begin{bmatrix} u_{xxi}^R x_i^2 \\ u_{yyi}^R y_i^2 \\ u_{zzi}^R z_i^2 \\ u_{xyi}^R \Delta x \Delta y \\ u_{xzi}^R \Delta x \Delta z \\ u_{yzi}^R \Delta y \Delta z \end{bmatrix} = \begin{bmatrix} - \left(u_j - \tilde{u}_i^R - u_{xi}^R B_2^j \Delta x_i - \right) \\ u_{yi}^R B_3^j \Delta y_i - u_{zi}^R B_4^j \Delta z_i \\ \Delta x_j (u_{xj}^R - u_{xi}^R) \\ \Delta y_j (u_{yj}^R - u_{yi}^R) \\ \Delta z_j (u_{zj}^R - u_{zi}^R) \end{bmatrix} \quad (3.66)$$

This matrix will be repeated for all adjacent cells. So for hexahedral cells we have 5 more neighboring cells and will have over determined system of

$$M_{24 \times 6} X_6 = B_{24}$$

This can be solved by normalization or any appropriate method of solving over-determined systems. Here we used the fact that linear polynomial of the reconstruction scheme must be conservative, and the values of the reconstructed first derivatives are equal to the ones of the first derivatives of the underlying DG solution at the centroid i of each cell. Due to the judicious choice of Taylor basis in our DG formulation, these four degrees of freedom simply coincide with the ones from the underlying DG solution. In this way we have just six unknowns at our RDG that should be obtained.

But the least square reconstruction for three-dimensional problems specially with tetrahedral cells, suffers from the so-called linear instability, that is also observed in the second-order

cell-centered finite volume methods, i.e., RDG(P₀P₁) [82]. This linear instability is attributed to the fact that the reconstruction stencils only involve Von Neumann neighborhood, i.e., adjacent face-neighboring cells. The linear stability can be achieved using extended stencils, which will unfortunately sacrifice the compactness of the underlying DG methods. Furthermore, such a linear reconstruction-based DG method cannot maintain the non-linear instability, leading to non-physical oscillations in the vicinity of strong discontinuities. Alternatively, ENO/WENO can be used to reconstruct a higher-order polynomial solution, which can not only enhance the order of accuracy of the underlying DG method but also achieve both linear and non-linear stability. Specifically, the WENO scheme introduced by Dumber et al. is adopted in this work, where an entire quadratic polynomial solution on cell i is obtained using a nonlinear WENO reconstruction as a convex combination of the least-squares reconstructed second derivatives at the cell itself and its four face-neighboring cells,

$$\frac{\partial^2 \mathbf{U}}{\partial x_m \partial x_n} \Big|_i = \sum_{k=1}^{N+1} w_k \frac{\partial^2 \mathbf{U}}{\partial x_m \partial x_n} \Big|_k \quad (3.67)$$

where N is number of neighboring cells of cell i the normalized nonlinear weights w_k are computed as

$$w_k = \frac{\tilde{w}_k}{\sum_{i=1}^{N+1} \tilde{w}_i} \quad (3.68)$$

The non-normalized nonlinear weights \tilde{w}_i are functions of the linear weights λ_i and the so-called oscillation indicator o_i

$$\tilde{w}_i = \frac{\lambda_i}{(\varepsilon + o_i)^\gamma} \quad (3.69)$$

where ε is a small positive number used to avoid division by zero, and γ an integer parameter to control how fast the non-linear weights decay for non-smooth stencils. The oscillation indicator for the reconstructed second order polynomials is simply defined as

$$o_k = [(\frac{\partial^2 U}{\partial x_m \partial x_n} |_{k})^2]^{1/2} \quad (3.70)$$

The linear weights λ_i can be chosen to balance the accuracy and the non-oscillatory property of the RDG method. Note that the least-squares reconstructed polynomial at the cell itself serves as the central stencil and the least-squares reconstructed polynomials on its four face-neighboring cells act as biased stencils in this WENO reconstruction. This reconstructed quadratic polynomial solution is then used to compute the domain and boundary integrals of the underlying DG(P_1) method. As demonstrated in [80], the resulting WENO(P_1P_2) method is able to achieve the designed third order of accuracy, maintain the linear stability, and significantly improve the accuracy of the underlying second-order DG method without significant increase in computing costs and storage requirements.

3.4.5 WENO reconstruction at P_1 : HWENO(P_1P_2)

Although the WENO (P_1P_2) method does not introduce any new oscillatory behavior for the reconstructed curvature terms (second derivatives) due to the WENO reconstruction, it cannot remove inherent oscillations in the underlying DG (P_1) solutions. Consequently, the WENO (P_1P_2) method still suffers from the non-linear instability for flows with strong discontinuities. In order to eliminate non-physical oscillations in the vicinity of strong discontinuities and thus maintain the non-linear instability, the first derivatives need to be

reconstructed using a WENO reconstruction. The resulting reconstructed discontinuous Galerkin method based on this Hierarchical WENO reconstruction is termed as HWENO(P_1P_2) at this work, where a hierarchical reconstruction (successively from high order to low order) strategy [84] is adopted.

The WENO reconstruction for the first derivatives is based on the reconstructed quadratic polynomial solutions of the flow variables for each cell in the mesh. The stencils are only chosen in the Von Neumann neighborhood in order to maintain the compactness of the underlying DG method. More precisely, for a cell i , (Tetrahedral for example) the following four stencils (i,j_1,j_2,j_3) , (i,j_1,j_2,j_4) , (i,j_1,j_3,j_4) , and (i,j_2,j_3,j_4) , where j_1 , j_2 , j_3 , and j_4 designate the four adjacent face-neighboring cells of the cell i , are chosen to construct a Lagrange polynomial such that

$$\mathbf{U}_j = \tilde{\mathbf{U}}_i + \mathbf{U}_{xi}^R B_2 + \mathbf{U}_{yi}^R B_3 + \mathbf{U}_{zi}^R B_4 + \mathbf{U}_{xxi}^R B_5 + \mathbf{U}_{yyi}^R B_6 + \mathbf{U}_{zzi}^R B_7 + \mathbf{U}_{xyi}^R B_8 + \mathbf{U}_{xzi}^R B_9 + \mathbf{U}_{yzi}^R B_{10} \quad (3.71)$$

and the following four stencils (i,j_1) , (i,j_2) , (i,j_3) , and (i,j_4) are chosen to construct a Hermite polynomial such that

$$\begin{aligned} \frac{\partial \mathbf{U}}{\partial x} /_j &= \mathbf{U}_{xi}^R \frac{1}{\Delta x_i} + \mathbf{U}_{xxi}^R \frac{B_2}{\Delta x_i} + \mathbf{U}_{xyi}^R \frac{B_3}{\Delta x_i} + \mathbf{U}_{xzi}^R \frac{B_4}{\Delta x_i} \\ \frac{\partial \mathbf{U}}{\partial y} /_j &= \mathbf{U}_{yi}^R \frac{1}{\Delta y_i} + \mathbf{U}_{yyi}^R \frac{B_3}{\Delta y_i} + \mathbf{U}_{xyi}^R \frac{B_2}{\Delta y_i} + \mathbf{U}_{yzi}^R \frac{B_4}{\Delta y_i} \\ \frac{\partial \mathbf{U}}{\partial z} /_j &= \mathbf{U}_{zi}^R \frac{1}{\Delta z_i} + \mathbf{U}_{zzi}^R \frac{B_4}{\Delta z_i} + \mathbf{U}_{xzi}^R \frac{B_2}{\Delta z_i} + \mathbf{U}_{yzi}^R \frac{B_3}{\Delta z_i} \end{aligned} \quad (3.72)$$

These reconstructed gradients $(\mathbf{U}_{xi}^R, \mathbf{U}_{yi}^R, \text{ and } \mathbf{U}_{zi}^R)$ serving as the biased stencils and the gradient from the DG solution itself at cell i $(\mathbf{U}_{xi}, \mathbf{U}_{yi}, \text{ and } \mathbf{U}_{zi})$ acting as the central stencil

are used to modify the first derivatives based on the WENO reconstruction as a convex combination of these nine derivatives

$$\frac{\partial \mathbf{U}}{\partial x_m} \Big|_i = \sum_{k=1}^9 w_k \frac{\partial \mathbf{U}}{\partial x_m} \Big|_k \quad (3.73)$$

where the normalized nonlinear weights w_k are computed as

$$w_k = \frac{\tilde{w}_k}{\sum_{i=1}^9 \tilde{w}_k} \quad (3.74)$$

The non-normalized nonlinear weights \tilde{w}_i are functions of the linear weights λ_i and the so-called oscillation indicator o_i

$$\tilde{w}_i = \frac{\lambda_i}{(\varepsilon + o_i)^\gamma} \quad (3.75)$$

where ε is a small positive number used to avoid division by zero, and γ an integer parameter to control how fast the non-linear weights decay for non-smooth stencils. The oscillation indicator is simply defined as

$$o_k = \left[\left(\frac{\partial \mathbf{U}}{\partial x_m} \Big|_k \right)^2 \right]^{1/2} \quad (3.76)$$

The present choice of stencils is symmetric, compact, and most importantly consistent with the underlying DG methods, as van Neumann neighbors are only involved in the reconstruction. This means that no additional data structure is required and the compactness of the DG methods is intact. Note that this WENO reconstruction at P_1 is the extension of a HWENO limiter developed for the DG(P_1) by the authors in [23]. From the perspective of both computational cost and solution accuracy, the above WENO reconstruction on P_1 should

only be used in the regions where strong discontinuities exist. All numerical experiments presented in the next section are performed by applying the P_1 reconstruction everywhere in an effort to ensure that the computational results are not affected by a shock detector, and to demonstrate the superior properties of the designed HWENO(P_1P_2) method.

3.4.6 WENO reconstruction at P_1 : HWENO(P_1P_1)

Clearly, a HWENO limiter developed for the DG(P_1) by the authors in [23] is now becoming the WENO(P_1P_1) method in the present frame work of RDG formulation based on WENO reconstruction, where the first derivatives are reconstructed based on a Hermite reconstruction. Like any higher-order schemes (>1), the discontinuous Galerkin methods will suffer from non-physical oscillations in the vicinity of discontinuities that exist in problems governed by hyperbolic conservation laws. Classical techniques of slope limiting are not directly applicable for high-order DGM because of the presence of volume terms in the formulation. Therefore, the slope limiter is not integrated in the computation of the residual, but effectively acts as a post-processing filter. Many slope limiters used in the finite volume method (FVM) can then be used or modified to meet the needs of the DGM. Unfortunately, the use of the limiter will reduce the order of accuracy to first order in the presence of discontinuities. Furthermore, the active limiters in the smooth extremas will pollute the solution in the flow field and ultimately destroy the higher-order accuracy of DGM [103]. Indeed, the limiters used in TVD/MUSCL finite volume methods are less robust than the strategies of essential non-oscillatory (ENO) and weighted ENO (WENO) finite volume methods. The ENO schemes were initially introduced by Harden et al. [104] in which

oscillations up to the order of the truncation error are allowed to overcome the drawbacks and limitations of limiter-based schemes. ENO schemes avoid interpolation across high-gradient regions through biasing of the reconstruction. This biasing is achieved by reconstructing the solution on several stencils at each location and selecting the reconstruction which is in some sense the smoothest. This allows ENO schemes to retain higher-order accuracy near high-gradient regions. However, the selection process can lead to convergence problems and loss of accuracy in regions with smooth solution variations. To counter these problems, the so-called weighted ENO scheme introduced by Liu et al. [105] is designed to present better convergence rate for steady state problems, better smoothing for the flux vectors, and better accuracy using the same stencils than the ENO scheme. WENO scheme uses a suitably weighted combination of all reconstructions rather than just the one which is judged to be the smoothest. The weighting is designed to favor the smooth reconstruction in the sense that its weight is small, if the oscillation of a reconstructed polynomial is high and its weight is order of one, if a reconstructed polynomial has low oscillation. Qiu and Shu initiated the use of WENO scheme as limiters for the DG method [106] for solving 1D and 2D Euler equations on structured grids.

Later on, they constructed a class of WENO schemes based on Hermite polynomials, termed as HWENO (Hermite WENO) schemes and applied this HWENO as limiters for the DG methods [107][108]. The main difference between HWENO and WENO schemes is that the former has a more compact stencil than the latter for the same order of accuracy.

Unfortunately, implementation of both ENO and WENO schemes is fairly complicated on arbitrary meshes, especially in 3D. In fact, there are very few results obtained using ENO/WENO on unstructured grids in 3D especially for higher-order reconstruction. Harden and Chakravarthy [109], Abgrall [110], and Sonar [111] presented the first implementation of ENO schemes on unstructured triangular grids. Implementations of WENO methods on unstructured triangular grids were also presented by Friedrich [112] and Hu and Shu [113].

In the present work, a WENO reconstruction scheme based on the Hermite polynomials is presented and used as a non-linear limiter for a discontinuous Galerkin method to solve the MHD equations on arbitrary grids. The new reconstruction scheme makes use of the invaluable information, namely derivatives that are handily available in the context of the discontinuous Galerkin method, thus making the implementation of WENO schemes straightforward on unstructured grids in both 2D and 3D. Only the van Neumann neighborhood is required for the construction of stencils, regardless of the order of solution polynomials to be reconstructed. The resulting HWENO reconstruction keeps full conservation of mass, momentum, and energy, is uniformly accurate with no overshoots and undershoots, is easy to implement on arbitrary meshes, has good convergence properties, and is computationally efficient.

A typical WENO cell-centered finite volume reconstruction scheme for a function u consists of the following steps:

1. Identify a number of admissible stencils $S_1; S_2; \dots; S_m$ for a cell Ω_e consisting of the neighboring cells, such that the cell itself Ω_e belongs to each stencil. Note that a stencil can be thought of as a set of cells that can be used to obtain a polynomial reconstruction.
2. Reconstruct the function u for each stencil S_i by a polynomial P_i based on the mean values of the function u on each cell in the stencil S_i
3. Compute an oscillation indicator o_i for each reconstructed polynomial P_i .
4. Calculate weights w_i for each P_i using oscillation indicator such that the sum of w_i is one.
5. Find the reconstruction polynomial p as the weighted sum of the P_i , $p = \sum_{i=1}^m w_i P_i$

There are some contradictory arguments here, On the one hand, the stencils should have a small diameter and are well centered with respect to Ω_e to obtain high accuracy and stability in smooth regions. The number of stencils should be small to reduce the computational cost; On the other hand, ENO methods are based on the idea that in case of non-smooth regions one-sided stencils are chosen to avoid interpolation across discontinuities. The number of stencils should be large enough to avoid any oscillation in the solutions and keep the scheme stable.

For the construction of a polynomial of degree p , the dimension of the polynomial space, $N = N(P,d)$ depends on the degree of the polynomials of the expansion p , and the number of spatial dimensions d . One must have 3, 6, and 10 cells in 2D and 4, 10, and 20 cells in 3D for the construction of a linear, quadratic, and cubic Lagrange polynomial, respectively. Undoubtedly, it is an overwhelmingly challenging, if not practically impossible, task to judiciously choose a set of admissible and proper stencils that have such a large number of

cells on unstructured grids especially for higher-order polynomials and higher dimensions. This explains why the application of higher-order ENO/WENO methods hardly exists on unstructured grids, in spite of their tremendous success on structured grids and their superior performance over the MUSCL/TVD methods.

However, the number of cells needed for a polynomial reconstruction can be significantly reduced, if a Hermite polynomial is used instead of a Lagrange one. This is only possible, if the first derivatives of the function to be reconstructed are known on the cells. Fortunately, this is exactly the case for the discontinuous Galerkin methods where the derivatives are handily available on each cell. Here, we confine ourselves to the case of Hermite WENO reconstruction for a linear polynomial P1. However, the idea can be used for the reconstruction of higher-order polynomials as well, though the second and higher derivatives need to be taken into consideration. The reconstruction process of the HWENO schemes for the DG methods is based on the approximation of mean and first derivative values of the flow variables for each cell in the mesh. The stencils are only chosen in a Von Neumann neighborhood in order to be compact and consistent with the DG method. More precisely, for cell Ω_e , the following three stencils ($\Omega_e\Omega_a\Omega_b$; $\Omega_e\Omega_b\Omega_c$; $\Omega_e\Omega_c\Omega_a$), shown in Figure 3.4, are chosen to construct a Lagrange polynomial such that

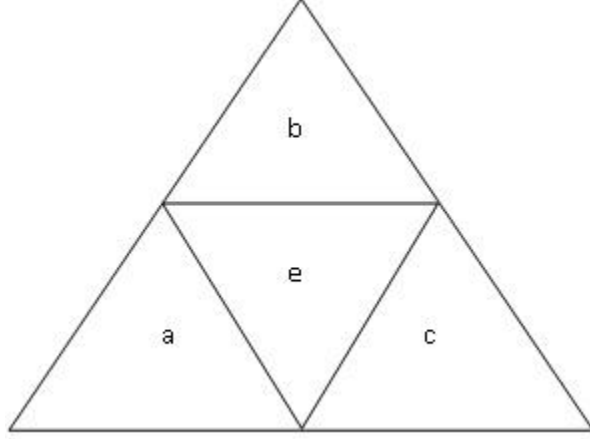


Figure 3.4 Von Neumann neighborhood of cell e to use in Hermit WENO reconstruction

$$\begin{aligned}
 \frac{1}{|\Omega_e|} \int_{\Omega_e} P_1 d\Omega &= u_e & \frac{1}{|\Omega_j|} \int_{\Omega_j} P_1 d\Omega &= u_j \\
 \frac{1}{|\Omega_k|} \int_{\Omega_k} P_1 d\Omega &= u_k & (j, k) &= (a, b; b, c; c, a)
 \end{aligned} \tag{3.77}$$

and the following four stencils ($\Omega_e\Omega_e$; $\Omega_e\Omega_a$; $\Omega_e\Omega_b$; $\Omega_e\Omega_c$) are chosen to construct a Hermite polynomial such that

$$\frac{1}{|\Omega_e|} \int_{\Omega_e} P_1 d\Omega = u_e \quad \frac{1}{|\Omega_k|} \int_{\Omega_k} \frac{\partial P_1}{\partial x_i} d\Omega = \left. \frac{\partial P_1}{\partial x_i} \right|_k \quad (k = e, a, b, c) \tag{3.78}$$

The present choice is unique, symmetric, compact, and most importantly consistent with the underlying DG methods, as only van Neumann neighbors are involved in the reconstruction. This means that no additional data structure is required and the compactness of the DG methods is maintained. Note that the number of the resulting stencils, except the solution polynomial itself, is six in 2D, exactly the same as the ones used in [104] [115] and eight in 3D. Since the first derivatives are handily available in the DG method, there is no need to

actually reconstruct the four Hermite polynomials in 2D (five in 3D) and only the three Lagrange polynomials in 2D (four in 3D) need to be reconstructed, representing a big saving in computing costs.

After the polynomial reconstruction is performed for each cell, an oscillation indicator is sought to assess the smoothness of P . Following the results presented in the literature [115], the oscillation indicator used in the present work is the one proposed by Jiang and Shu [116], which was later modified by Friedrich [115]. The oscillation indicator for the reconstructed polynomial P_i can be defined as

$$o_i = \left[\int_{\Omega_i} h^{-2} \left(\frac{\partial P_i}{\partial x_k} \right)^2 d\Omega \right]^{\frac{1}{2}} \quad (3.79)$$

where h is the mesh width.

Unlike the ENO schemes, the WENO schemes use all the computed polynomials. These polynomials are added together through the use of weights which are determined for each one of the polynomials as proportional to its respective oscillation indicator. The main idea in the WENO reconstruction is to attribute the computed weights for each polynomial with the aim of reconstructing a new polynomial as $p = \sum_{i=1}^m w_i P_i$. The weights are computed as

$$w_i = \frac{(\epsilon + o_i(P_i))^{-\gamma}}{\sum_{k=1}^p (\epsilon + o_i(P_k))^{-\gamma}} \quad (3.80)$$

where γ is a positive number. This type of limiting is fundamentally different from the one used in TVD schemes. Reconstruction scheme based on the WENO limiting weights gradients obtained from neighboring stencils in order to continuously eliminate these which cause oscillations. From the perspective of both computational cost and numerical accuracy,

the above HWENO limiting should only be used in the regions where strong discontinuities exist. This can be accomplished using the so-called discontinuity detectors, which are helpful to distinguish regions where solutions are smooth and discontinuous. The beauty of this HWENO limiter is that in case that the limiting is mistakenly applied in the smooth cells, the uniform high-order accuracy can still be maintained, unlike the slope limiters, which, when applied near smooth extrema, will have a profoundly adverse impact on solution in the smooth region, leading the loss of the original high-order accuracy. This remarkable feature of the HWENO limiter in turn alleviates the burden on the discontinuity detectors, as no discontinuity detectors can really either in theory or in practice makes a distinction between a stagnation point and a shock wave, as flow gradients near the stagnation point are even larger than the ones near the shock wave in some cases. All numerical experiments to be presented in the next section are performed by applying the limiters everywhere in an effort to ensure that the computational results are not affected by a shock detector, and to demonstrate the superior properties of the designed HWENO limiter.

3.5 HLLD Riemann Solver

In Finite Volume (FVM) method and also Discontinuous Galerkin (DG) Method we use integral form of equations. To do so, for example in traditional FV that is base for DG we take integral of equation (2.9) over domain Ω :

$$\int_{\Omega} \left(\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} \right) d\Omega = 0 \quad (3.81)$$

Applying divergence theorem over equation (3.81):

$$\int_{\Omega} \left(\frac{\partial \vec{Q}}{\partial t} \right) d\Omega + \int_{\Gamma} \left(\frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} \right) \cdot \vec{n} d\Gamma = 0 \quad (3.82)$$

In which \vec{n} is outward unit vector normal to boundaries Γ of domain Ω :

$$\vec{n} = (n_x \hat{i} + n_y \hat{j} + n_z \hat{k}) \quad (3.83)$$

It is more convenient for Finite Volume (FV) method to write equation (3.82) as below:

$$\int_{\Omega} \left(\frac{\partial \vec{Q}}{\partial t} \right) d\Omega + \int_{\Gamma} \mathbf{F}_n d\Gamma = 0 \quad (3.84)$$

In which \mathbf{F}_n is flux vector normal to the interface between two cells (normal to the boundary of domain):

$$\mathbf{F}_n = \begin{pmatrix} \rho v_n \\ \rho u v_n - B_x B_n + P_T n_x \\ \rho v v_n - B_y B_n + P_T n_y \\ \rho w v_n - B_z B_n + P_T n_z \\ (\rho E + P_T) v_n - B_n (\mathbf{V} \cdot \mathbf{B}) \\ v_n B_x - u B_n \\ v_n B_y - v B_n \\ v_n B_z - w B_n \end{pmatrix} \quad (3.85)$$

Where normal velocity and normal magnetic vectors are defined as below:

$$v_n = (u \hat{i} + v \hat{j} + w \hat{k}) \cdot (n_x \hat{i} + n_y \hat{j} + n_z \hat{k}) = u n_x + v n_y + w n_z \quad (3.86)$$

$$B_n = (B_x \hat{i} + B_y \hat{j} + B_z \hat{k}) \cdot (n_x \hat{i} + n_y \hat{j} + n_z \hat{k}) = B_x n_x + B_y n_y + B_z n_z \quad (3.87)$$

We define Alfven wave speeds as follow normal to boundary:

$$C_{alfven} = \frac{|B_n|}{\sqrt{\rho}} \quad (3.88)$$

In which \mathbf{B}_n is the magnetic vector normal to boundary and ρ is density. Also speeds of sound as follow:

$$C_{sound} = \sqrt{\frac{\gamma p}{\rho}} \quad (3.89)$$

Where γ is specific heat ratio and p is pressure. In MHD there are two fast and slow waves defining as below:

$$C_{fast,slow} = \frac{1}{\sqrt{2}} \left\{ C_{sound}^2 + C_{alfven}^2 \pm \sqrt{(C_{sound}^2 + C_{alfven}^2)^2 - 4C_{sound}^2 C_{alfven}^2} \right\}^{\frac{1}{2}} \quad (3.90)$$

With these definitions flux vector F_n at equation (3.85) has 8 eigenvalues as below:

$$\lambda_{1,8} = v_n \pm c_f, \quad \lambda_{2,7} = v_n \pm c_a, \quad \lambda_{3,6} = v_n \pm c_s, \quad \lambda_{4,5} = v_n \quad (3.91)$$

As we see in equation (3.84), we face with Riemann problem. To solve this problem there are different ways. The Godunov method and its high-order extensions require the solution of the Riemann problem. In practical computation this is solved billions of times, making the Riemann problem solution process the single most demanding task in the numerical method. The exact Riemann solvers requires an iterative procedure to solve a nonlinear problem and the associated computational effort may not always be justified. This effort may increase dramatically by the complexity of the equations of state or the complexity of the particular system of equations being solved. For MHD, the exact Riemann solver is far more costly than the case of Hydrodynamic.

The HLLC-type Riemann solvers for MHD as reviewed by Miyoshi and Kusano [93] may have some inconsistency with respect to the jump conditions without a particular treatment. They suppose that the HLLC-type solvers may include inconsistency between the assumption of constant normal velocity and the two-state approximation of the intermediate states in the Riemann fan. Therefore, they constructed the multi-state (more than two-state) HLL

Riemann solver for the MHD equations based on the same basic assumption as that in the HLLC Riemann solver for the Euler equations.

They assumed that the normal velocity is constant over the Riemann fan. Their assumption which is the same as in the HLLC solver [94] leads to the following noticeable conclusions: The normal velocity in the Riemann fan corresponds to the speed of the middle (entropy) wave. The total pressure is constant over the Riemann fan. Slow shocks cannot be formed inside the Riemann fan. Rotational discontinuities propagating with the Alfvén waves, on the other hand, may be generated. The latter two conclusions suggest that, in order to construct a more accurate HLL Riemann solver for MHD than the single-state HLL solver, the Riemann fan may be divided into four intermediate states, U_L^* U_L^{**} U_R^{**} U_R^* as illustrated in Figure 3.5. Therefore, we consider the approximate Riemann problem in the four-state Riemann fan separated by one entropy and two Alfvén waves, S_M and S_L ; S_R . Miyoshi and Kusano [93] developed their formulation for MHD problem in x direction and neglected flux terms in y and z direction. Here we extend their formulation for general one dimensional problem normal to the boundaries as we need in FV and DG method. Here we use equations of (3.84) and (3.85) and vector values are normal values.

The choice of S_M , in the present solver, is to evaluate the average normal velocity from the HLL average (3.91) as Batten et al. [95] and Gurski [96] did:

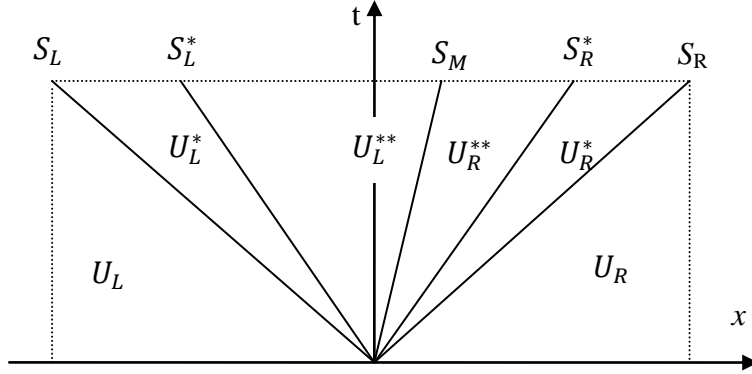


Figure 3.5 Schematic structure of 4 states Riemann fan for HLLD Riemann Solver

$$S_M = \frac{(S_R - v_{n,R})\rho_R v_{n,R} - (S_L - v_{n,L})\rho_L v_{n,L} - P_{T,R} + P_{T,L} + B_{n,R}^2 - B_{n,L}^2}{(S_R - v_{n,R})\rho_R - (S_L - v_{n,L})\rho_L} \quad (3.92)$$

In order to obtain the exact upper and lower bounds of the signal speeds for the Riemann problem, complicated exact solutions for the MHD Riemann problem are needed[95]. Therefore, S_L and S_R may be estimated approximately by [93] or [96]. Here we have the explicit expressions of the largest and smallest eigenvalues for MHD in (3.90) and (3.91), other estimations may be utilized, for instance such that

$$S_L = \min(v_{n,L}, v_{n,R}) - \max(C_{fast,L}, C_{fast,R}) \quad (3.93)$$

$$S_R = \max(v_{n,L}, v_{n,R}) + \max(C_{fast,L}, C_{fast,R}) \quad (3.94)$$

Having S_L and S_R in hand, we can write jump condition around S_L and S_R :

$$\begin{aligned}
S_\alpha \begin{pmatrix} \rho_\alpha^* \\ \rho_\alpha^* u_\alpha^* \\ \rho_\alpha^* v_\alpha^* \\ \rho_\alpha^* w_\alpha^* \\ \rho_\alpha^* E_\alpha^* \\ B_{x,\alpha}^* \\ B_{y,\alpha}^* \\ B_{z,\alpha}^* \end{pmatrix} - \begin{pmatrix} \rho_\alpha^* S_M \\ \rho_\alpha^* u_\alpha^* S_M - B_{x,\alpha}^* B_M + P_{T,\alpha}^* n_x \\ \rho_\alpha^* v_\alpha^* S_M - B_{y,\alpha}^* B_M + P_{T,\alpha}^* n_y \\ \rho_\alpha^* w_\alpha^* S_M - B_{z,\alpha}^* B_M + P_{T,\alpha}^* n_z \\ (\rho_\alpha^* E_\alpha^* + P_{T,\alpha}^*) S_M - B_M (\mathbf{V}_\alpha^* \cdot \mathbf{B}_\alpha^*) \\ S_M B_{x,\alpha}^* - u_\alpha^* B_M \\ S_M B_{y,\alpha}^* - v_\alpha^* B_M \\ S_M B_{z,\alpha}^* - w_\alpha^* B_M \end{pmatrix} &= \\
S_\alpha \begin{pmatrix} \rho_\alpha \\ \rho_\alpha u_\alpha \\ \rho_\alpha v_\alpha \\ \rho_\alpha w_\alpha \\ \rho_\alpha E_\alpha \\ B_{x,\alpha} \\ B_{y,\alpha} \\ B_{z,\alpha} \end{pmatrix} - \begin{pmatrix} \rho_\alpha v_{n,\alpha} \\ \rho_\alpha u_\alpha v_{n,\alpha} - B_{x,\alpha} B_{n,\alpha} + P_{T,\alpha} n_x \\ \rho_\alpha v_\alpha v_{n,\alpha} - B_{y,\alpha} B_{n,\alpha} + P_{T,\alpha} n_y \\ \rho_\alpha w_\alpha v_{n,\alpha} - B_{z,\alpha} B_{n,\alpha} + P_{T,\alpha} n_z \\ (\rho_\alpha E_\alpha + P_{T,\alpha}) v_{n,\alpha} - B_{n,\alpha} (\mathbf{V}_\alpha \cdot \mathbf{B}_\alpha) \\ v_{n,\alpha} B_{x,\alpha} - u_\alpha B_{n,\alpha} \\ v_{n,\alpha} B_{y,\alpha} - v_\alpha B_{n,\alpha} \\ v_{n,\alpha} B_{z,\alpha} - w_\alpha B_{n,\alpha} \end{pmatrix} &
\end{aligned} \tag{3.95}$$

In which α is L or R and \mathbf{B}_M is defined as below:

$$B_M = \frac{B_{n,R} + B_{n,L}}{2} \tag{3.96}$$

The normal velocity and total pressure assumed to be constant all over the Riemann fan; this assumption is consistent with jump condition

$$v_{n,L}^* = v_{n,R}^* = v_{n,L}^{**} = v_{n,R}^{**} = S_M \tag{3.97}$$

$$P_{T,L}^* = P_{T,R}^* = P_{T,L}^{**} = P_{T,R}^{**} = P_T^* \tag{3.98}$$

$$P_{T,\alpha}^* = \rho_\alpha (S_\alpha - v_{n,\alpha}) (S_M - v_{n,\alpha}) + B_M^2 - B_{n,\alpha}^2 + P_{T,\alpha} \tag{3.99}$$

First row of equation (3.95) gives us the following relation for density at star region

$$\rho_\alpha^* = \rho_\alpha \frac{S_\alpha - v_{n,\alpha}}{S_\alpha - S_M} \tag{3.100}$$

Manipulating rows 2-4 of equation (3.95) gives the following relations for velocity at star region:

$$u_{\alpha}^* = \left\{ \begin{array}{l} u_{\alpha}(\rho_{\alpha}(S_{\alpha} - v_{n,\alpha})(S_{\alpha} - S_M) - B_M B_{n,\alpha}) \\ + B_{x,\alpha}(B_{n,\alpha}(S_{\alpha} - S_M) - B_M(S_{\alpha} - v_{n,\alpha})) \\ + (P_T^* - P_{T,\alpha})(S_{\alpha} - S_M)n_x \end{array} \right\} / \{\rho_{\alpha}(S_{\alpha} - v_{n,\alpha})(S_{\alpha} - S_M) - B_M^2\} \quad (3.101)$$

$$v_{\alpha}^* = \left\{ \begin{array}{l} v_{\alpha}(\rho_{\alpha}(S_{\alpha} - v_{n,\alpha})(S_{\alpha} - S_M) - B_M B_{n,\alpha}) \\ + B_{y,\alpha}(B_{n,\alpha}(S_{\alpha} - S_M) - B_M(S_{\alpha} - v_{n,\alpha})) \\ + (P_T^* - P_{T,\alpha})(S_{\alpha} - S_M)n_y \end{array} \right\} / \{\rho_{\alpha}(S_{\alpha} - v_{n,\alpha})(S_{\alpha} - S_M) - B_M^2\} \quad (3.102)$$

$$w_{\alpha}^* = \left\{ \begin{array}{l} w_{\alpha}(\rho_{\alpha}(S_{\alpha} - v_{n,\alpha})(S_{\alpha} - S_M) - B_M B_{n,\alpha}) \\ + B_{z,\alpha}(B_{n,\alpha}(S_{\alpha} - S_M) - B_M(S_{\alpha} - v_{n,\alpha})) \\ + (P_T^* - P_{T,\alpha})(S_{\alpha} - S_M)n_z \end{array} \right\} / \{\rho_{\alpha}(S_{\alpha} - v_{n,\alpha})(S_{\alpha} - S_M) - B_M^2\} \quad (3.103)$$

From fifth equation we have the following relation for total energy

$$\rho_{\alpha}^* E_{\alpha}^* = \frac{P_T^* S_M - P_{T,\alpha} v_{n,\alpha} + \rho_{\alpha} E_{\alpha}(S_{\alpha} - v_{n,\alpha}) + B_{n,\alpha}(\mathbf{V}_{\alpha} \cdot \mathbf{B}_{\alpha}) - B_M(\mathbf{V}_{\alpha}^* \cdot \mathbf{B}_{\alpha}^*)}{(S_{\alpha} - S_M)} \quad (3.104)$$

We also do same thing for rows 6-8 for magnetic filed

$$B_{x,\alpha}^* = \frac{-u_{\alpha}^* B_M + B_{x,\alpha}(S_{\alpha} - v_{n,\alpha}) + u_{\alpha} B_{n,\alpha}}{(S_{\alpha} - S_M)} \quad (3.105)$$

$$B_{y,\alpha}^* = \frac{-v_{\alpha}^* B_M + B_{y,\alpha}(S_{\alpha} - v_{n,\alpha}) + v_{\alpha} B_{n,\alpha}}{(S_{\alpha} - S_M)} \quad (3.106)$$

$$B_{z,\alpha}^* = \frac{-w_{\alpha}^* B_M + B_{z,\alpha}(S_{\alpha} - v_{n,\alpha}) + w_{\alpha} B_{n,\alpha}}{(S_{\alpha} - S_M)} \quad (3.107)$$

From jump condition around any wave speed S^* :

$$\rho_{\alpha}^* = \rho_{\alpha}^{**} \quad (3.108)$$

We also define the intermediate wave speeds:

$$S_L^* = S_M - \frac{|B_M|}{\sqrt{\rho_L^*}} \quad S_R^* = S_M + \frac{|B_M|}{\sqrt{\rho_R^*}} \quad (3.109)$$

If $B_M \neq 0$ writing jump condition around S_M :

$$\begin{array}{l} u_L^{**} = u_R^{**} = u^{**}, \quad B_{xL}^{**} = B_{xR}^{**} = B_x^{**} \\ v_L^{**} = v_R^{**} = v^{**}, \quad B_{yL}^{**} = B_{yR}^{**} = B_y^{**} \\ w_L^{**} = w_R^{**} = w^{**}, \quad B_{zL}^{**} = B_{zR}^{**} = B_z^{**} \end{array} \quad (3.110)$$

Substituting equations (3.108), (3.109), and (3.110) into integral conversation law over the Riemann fan, i.e. :

$$\begin{aligned} (S_R - S_R^*)U_R^* + (S_R^* - S_M)U_R^{**} + (S_M - S_L^*)U_L^{**} + (S_L^* - S_L)U_L^* \\ - S_R U_R + S_L U_L + F_R - F_L = 0 \end{aligned} \quad (3.111)$$

Gives the following relations for vector components:

$$B_x^{**} = \frac{\begin{pmatrix} B_{x,R}^*(S_R^* - S_R) - B_{x,L}^*(S_L^* - S_L) + B_{x,R}(S_R - v_{n,R}) \\ -B_{x,L}(S_L - v_{n,L}) + u_R B_{n,R} - u_L B_{n,L} \end{pmatrix}}{\left(\frac{|B_M|}{\sqrt{\rho_R^*}} + \frac{|B_M|}{\sqrt{\rho_L^*}}\right)} \quad (3.112)$$

$$B_y^{**} = \frac{\begin{pmatrix} B_{y,R}^*(S_R^* - S_R) - B_{y,L}^*(S_L^* - S_L) + B_{y,R}(S_R - v_{n,R}) \\ -B_{y,L}(S_L - v_{n,L}) + v_R B_{n,R} - v_L B_{n,L} \end{pmatrix}}{\left(\frac{|B_M|}{\sqrt{\rho_R^*}} + \frac{|B_M|}{\sqrt{\rho_L^*}}\right)} \quad (3.113)$$

$$B_z^{**} = \frac{\begin{pmatrix} B_{z,R}^*(S_R^* - S_R) - B_{z,L}^*(S_L^* - S_L) + B_{z,R}(S_R - v_{n,R}) \\ -B_{z,L}(S_L - v_{n,L}) + w_R B_{n,R} - w_L B_{n,L} \end{pmatrix}}{\left(\frac{|B_M|}{\sqrt{\rho_R^*}} + \frac{|B_M|}{\sqrt{\rho_L^*}}\right)} \quad (3.114)$$

$$u^{**} = \frac{\begin{pmatrix} (S_R^* - S_R)\rho_R^* u_R^* - (S_L^* - S_L)\rho_L^* u_L^* + \\ (S_R - v_{n,R})\rho_R u_R - (S_L - v_{n,L})\rho_L u_L + \\ B_{x,R} B_{n,R} - B_{x,L} B_{n,L} - (P_{T,R} - P_{T,L})n_x \end{pmatrix}}{(|B_M|(\sqrt{\rho_R^*} + \sqrt{\rho_L^*}))} \quad (3.115)$$

$$v^{**} = \frac{\begin{pmatrix} (S_R^* - S_R)\rho_R^* v_R^* - (S_L^* - S_L)\rho_L^* v_L^* + \\ (S_R - v_{n,R})\rho_R v_R - (S_L - v_{n,L})\rho_L v_L + \\ B_{y,R} B_{n,R} - B_{y,L} B_{n,L} - (P_{T,R} - P_{T,L})n_y \end{pmatrix}}{(|B_M|(\sqrt{\rho_R^*} + \sqrt{\rho_L^*}))} \quad (3.116)$$

$$w^{**} = \frac{\begin{pmatrix} (S_R^* - S_R)\rho_R^* w_R^* - (S_L^* - S_L)\rho_L^* w_L^* + \\ (S_R - v_{n,R})\rho_R w_R - (S_L - v_{n,L})\rho_L w_L + \\ B_{z,R} B_{n,R} - B_{z,L} B_{n,L} - (P_{T,R} - P_{T,L})n_z \end{pmatrix}}{(|B_M|(\sqrt{\rho_R^*} + \sqrt{\rho_L^*}))} \quad (3.117)$$

And finally writing jump condition around S_L^* and S_R^* the following relations for total energy obtained:

$$\rho_L^{**} E_L^{**} = \rho_L^* E_L^* - \text{sign}(B_M) \sqrt{\rho_L^*} \{(\mathbf{U}_L^* \cdot \mathbf{B}_L^*) - (\mathbf{U}_L^{**} \cdot \mathbf{B}_L^{**})\} \quad (3.118)$$

$$\rho_R^{**} E_R^{**} = \rho_R^* E_R^* + \text{sign}(B_M) \sqrt{\rho_R^*} \{(\mathbf{U}_R^* \cdot \mathbf{B}_R^*) - (\mathbf{U}_R^{**} \cdot \mathbf{B}_R^{**})\} \quad (3.119)$$

In which “*sign*” is Sign function. Finally we calculate fluxes using the following relation:

$$F_{HLLD} = \begin{cases} F_L & \text{if } S_L > 0 \\ F_L^* & \text{if } S_L \leq 0 \leq S_L^* \\ F_L^{**} & \text{if } S_L^* \leq 0 \leq S_M \\ F_R^{**} & \text{if } S_M \leq 0 \leq S_R^* \\ F_R^* & \text{if } S_R^* \leq 0 \leq S_R \\ F_R & \text{if } S_R < 0 \end{cases} \quad (3.120)$$

After above assumption for ** regions we see that variables for U_L^{**} and U_R^{**} are the same. So as shown in Figure 3.6 the four states Riemann fan reduces to three states Riemann fan. Miyoshi Kusano [93] named this Riemann solver as HLLD Riemann solver in which HLL comes from **H**arten, **L**ax, and Van **L**eer. Here, “D” stands for **D**iscontinuities. In absence of magnetic field equation (3.109) reduces to S_M and HLLD Riemann Solver become HLLC Riemann solver.

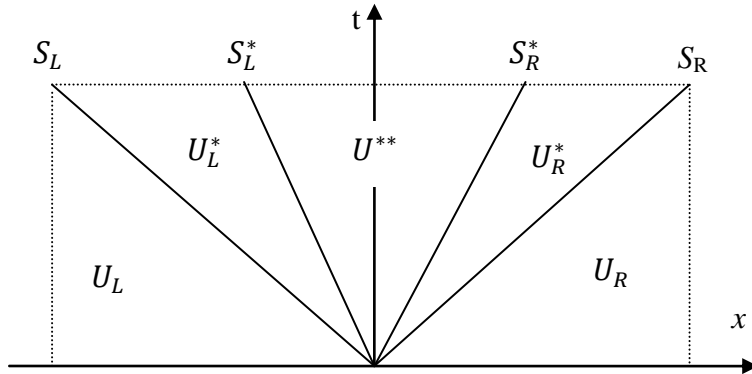


Figure 3.6: Final 3 states Rimann fan for HLLD Riemann solver after above assumptions

3.6 Divergence Free

For multidimensional MHD one also needs to handle the non-trivial $\nabla \cdot \mathbf{B} = 0$ constraint. Even if satisfied at $t=0$ one can still generate $\nabla \cdot \mathbf{B} \neq 0$ numerically due to the nonlinearity of the equations. Beside the fact that this is undesirable physically, if no correction taken at all, accumulated errors can lead to numerical divergence of the method.

A thorough comparison of seven different methods on nine different two dimensional test cases has been done on [97]. Here we briefly mention few different methods to handle divergence free criteria in modern state of the art MHD codes.

3.6.1 Vector Potential

One way of satisfying divergence free is replacing magnetic field by a vector potential defined by:

$$\mathbf{B} = \nabla \times \mathbf{A} \tag{3.121}$$

Analytically it guarantees the solenoidal magnetic field, but when evaluating \mathbf{B} in some discrete manner from \mathbf{A} the operator should still be chosen to satisfy

$$\nabla \cdot (\nabla \times \mathbf{A}) = 0 \tag{3.122}$$

Using \mathbf{A} unavoidably increases the order of the occurring spatial derivatives. The need to express boundary conditions on \mathbf{A} instead of \mathbf{B} when using ghost cells is also a non-trivial complication. This can be avoided by directly prescribing fluxes at the boundary instead. From the discussion of characteristic based solvers, it is clear that they conflict with using the vector potential as basic solvers, though the vector potential can be successfully exploited.

3.6.2 Projection Method

This method originally proposed by Brackbill and Barnes [98] in this method one can combine any multi-dimensional method with a projection scheme strategy which controls numerical values of divergence of magnetic field ($\nabla \cdot B = 0$) in a particular discretization to a given accuracy. The Idea is to correct the B^* computed numerically which does not satisfy divergence free ($\nabla \cdot B^* \neq 0$) by projecting it on the sub-space of zero divergence solutions. Hence, we modify B^* by subtracting the gradient of a scalar field ϕ to be computed from the poisson equation

$$\nabla^2 \phi = \nabla \cdot B^* \quad (3.123)$$

By construction this yields a solenoidal $B = B^* - \nabla \phi$ which is then used in the next time step. This process can be repeated after each or only after certain number of time steps. It is important to note that accuracy up to which the poisson problem is solved need not be machine precision, and that this approach keeps the order of accuracy of the base scheme, while not violating its conservation properties [97]. This elliptic divergence cleaning indirectly affects the local balance of thermal to kinetic contributions to the total energy. It is possible to rather keep thermal energy unmodified, at the cost of given up exact conservation of total energy.

The projection scheme in essence uses numerical concept from incompressible fluid flow and on uniform Cartesian grids was shown [97] to make the smallest possible correction to remove the divergence created by the base scheme. It corresponds to minimizing $|B - B^*|$

under the solenoidal constraint. For the cases that initial state is not known analytically, it may even be useful to determine finite numerical divergence errors of the discrete initial \mathbf{B} .

3.6.3 Locally Divergence Free

The idea of Locally Divergence Free (LDF) first used in Maxwell equations by Cockburn et al. [90] then extended to satisfying divergence free criteria in MHD by Li and Shu [91]. The advantage of this method is the use of approximate solutions that are exactly divergence-free inside each element for the magnetic field. As a result, using LDF has a smaller computational cost than the traditional discontinuous Galerkin method with standard piecewise polynomial spaces. They formulate the locally divergence-free discontinuous Galerkin method for the ideal MHD equations and performed extensive one and two-dimensional numerical experiments for both smooth solutions and solutions with discontinuities. their results showed that the locally divergence-free discontinuous Galerkin method, with a reduced cost comparing to the traditional discontinuous Galerkin method, can maintain the same accuracy for smooth solutions and can increase the numerical stability of the method and reduce certain nonphysical oscillations in some of the test cases.

They tried to find $U_h \in V_h$, such that for all $v \in V_h$ such that equation (3.38) holds, where V_h is the solution space, which is the same as the test space and given by

$$V_h = V_h^k = \left\{ v|_k \in P^k(k), \left(\frac{\partial v_6}{\partial x} + \frac{\partial v_7}{\partial y} \right) |_{k} = 0, \forall K \right\} \quad (3.124)$$

with $P^k(K) = (P^k(K))$, and $P^k(K)$ denotes the space of polynomials in K of degree at most k .

By this definition, they had solution spaces which contain magnetic fields with zero

divergence within each element. They call these spaces the locally divergence-free spaces, and call the discontinuous Galerkin methods with these solution spaces the locally divergence-free discontinuous Galerkin methods. Since the solution space is a subspace of the standard piecewise polynomial space, we expect a saving in computational cost by using the locally divergence-free spaces. For $\bar{X} = \frac{(x-x_i)}{\Delta x_i}$ and $\bar{Y} = \frac{(y-y_j)}{\Delta y_j}$ one set of orthogonal bases

for two dimensional problems would be

when $k=1$

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} \Delta x_i \bar{X} \\ -\Delta y_j \bar{Y} \end{pmatrix}, \quad \begin{pmatrix} \bar{Y} \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ \bar{X} \end{pmatrix} \quad (3.125)$$

And for $k=2$ the following term should be added

$$\begin{pmatrix} \Delta x_i (12\bar{X}^2 - 1) \\ -24\Delta y_j \bar{X}\bar{Y} \end{pmatrix}, \quad \begin{pmatrix} -24\Delta x_i \bar{X}\bar{Y} \\ \Delta y_j (12\bar{Y}^2 - 1) \end{pmatrix}, \quad \begin{pmatrix} 12\bar{Y}^2 - 1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 12\bar{0}^2 - 1 \end{pmatrix} \quad (3.126)$$

Here for our three dimensional MHD we try to find $U_h \in V_h$, such that for all $v \in V_h$ equation (3.38) holds, where V_h is the solution space, which is the same as the test space and given by

$$V_h = V_h^k = \left\{ v|_k \in P^k(k), \left(\frac{\partial v_6}{\partial x} + \frac{\partial v_7}{\partial y} + \frac{\partial v_8}{\partial z} \right) \Big|_k = 0, \forall K \right\} \quad (3.127)$$

Here after each time step we force divergence of magnetic field to be zero, i.e. after each time step we substitute derivative of B_z in z direction by sum of derivative of B_x in x direction and derivative of B_y in y direction.

Chapter 4 Temporal Discretization

In this Chapter, we discuss two different approaches for temporal discretization. First we introduce simple 1st order Euler method, and then 4th order Runge-Kutta method. Since our spatial discretization is at least 2nd order, here we do not use 1st order Euler method, because the temporal discretization is dominant and it will deteriorate the accuracy. We use 4th order Runge-Kutta, and since we use $CFL < 1.0$ and the method is 4th order, temporal discretization error is not dominant anymore, so we can study spatial order of our method. Here 1st order Euler method is presented just for comparison with 4th order Runge-Kutta.

4.1 Euler's Method

Euler's method is a numerical technique to solve ordinary differential equations of the form

$$\frac{dy}{dx} = f(x, y) \quad , \quad y(0) = y_0 \quad (4.1)$$

So only first order ordinary differential equations can be solved by using Euler's method. As an example we have the following equation

$$\frac{dy}{dx} + 2y = 1.3e^{-x}, y(0) = 5 \quad (4.2)$$

Rewriting in the form of equation (4.1) we will have

$$\frac{dy}{dx} = 1.3e^{-x} - 2y, y(0) = 5$$

In this case

$$f(x, y) = 1.3e^{-x} - 2y$$

At $x=0$ we are given the value of $y = y_0$. Lets call $x=0$ as x_0 . Now since we know the slope of y with respect to x , which is $f(x, y)$, then at $x = x_0$, the slope is $f(x_0, y_0)$. Both x_0 and y_0 are known from the initial condition $y(x_0) = y_0$.

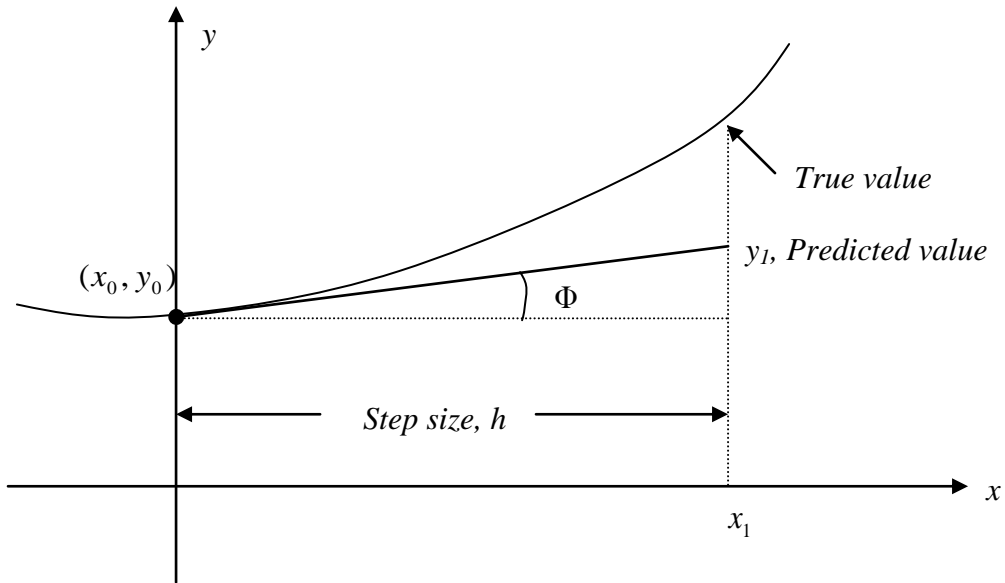


Figure 4.1 Graphical interpretation of the first step of Euler's method.

So the slope at $x = x_0$ as shown in Figure 4.1 is

$$\text{Slope} = \frac{\text{Rise}}{\text{Run}} = \frac{y_1 - y_0}{x_1 - x_0} = f(x_0, y_0) \quad (4.3)$$

From here

$$y_1 = y_0 + f(x_0, y_0)(x_1 - x_0) \quad (4.4)$$

Calling $x_1 - x_0$ the step size h , we get

$$y_1 = y_0 + f(x_0, y_0)h$$

One can now use the value of y_1 (an approximate value of y at $x = x_1$) to calculate y_2 , and that would be the predicted value at x_2 , given by

$$y_2 = y_1 + f(x_1, y_1)h$$

$$x_2 = x_1 + h$$

Based on the above equations, if we now know the value of $y = y_i$ at x_i , then

$$y_{i+1} = y_i + f(x_i, y_i)h \tag{4.5}$$

This formula is known as Euler's method and is illustrated graphically at Figure 4.2. In some books, it is also called the Euler-Cauchy method.

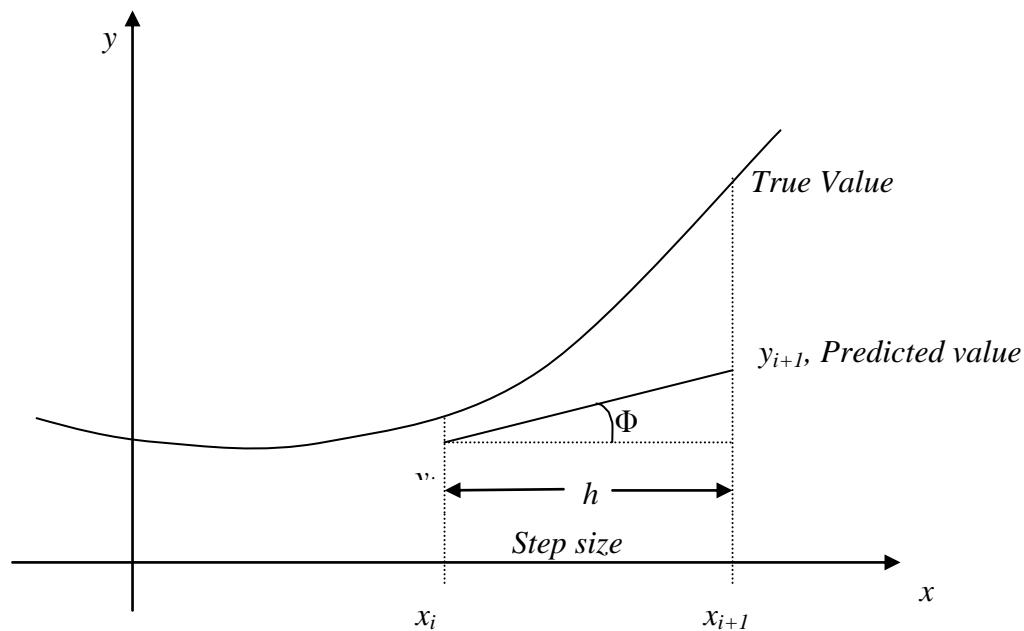


Figure 4.2 General graphical interpretation of Euler's method.

Euler's method has large errors. This can be illustrated using the Taylor series.

$$\begin{aligned}
y_{i+1} &= y_i + \left. \frac{dy}{dx} \right|_{x_i, y_i} (x_{i+1} - x_i) + \frac{1}{2!} \left. \frac{d^2 y}{dx^2} \right|_{x_i, y_i} (x_{i+1} - x_i)^2 + \frac{1}{3!} \left. \frac{d^3 y}{dx^3} \right|_{x_i, y_i} (x_{i+1} - x_i)^3 + \dots \\
&= y_i + f(x_i, y_i)(x_{i+1} - x_i) + \frac{1}{2!} f'(x_i, y_i)(x_{i+1} - x_i)^2 + \frac{1}{3!} f''(x_i, y_i)(x_{i+1} - x_i)^3 + \dots
\end{aligned} \tag{4.6}$$

As you can see the first two terms of the Taylor series

$$y_{i+1} = y_i + f(x_i, y_i)h$$

are Euler's method. If we divide both sides by h , the true error in the approximation is given by

$$E_i = \frac{f'(x_i, y_i)}{2!} h + \frac{f''(x_i, y_i)}{3!} h^2 + \dots \tag{4.7}$$

So this method is first order and is not appropriate for our simulation in which we want to solve MHD equation with DG method that is at least 2nd order accurate in space.

4.2 Forth order Runge-Kutta Method

Runge-Kutta methods are single-step methods, however, with multiple stages per step. These methods do not require derivatives of the right-hand side function f in the code, and are therefore general-purpose initial value problem solvers. Runge-Kutta methods are among the most popular ODE solvers. They were first studied by Carle Runge and Martin Kutta around 1900. Modern developments are mostly due to John Butcher in the 1960s. The Runge-Kutta 4th order method is based on the following

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2 + a_3 k_3 + a_4 k_4)h \tag{4.8}$$

where knowing the value of $y=y_i$ at x_i we can find the value of $y=y_{i+1}$ at x_{i+1} and

$$h = x_{i+1} - x_i \quad (4.9)$$

Equation (4.8) is equated to the first five terms of Taylor series

$$y_{i+1} = y_i + \frac{dy}{dx} \Big|_{x_i, y_i} (x_{i+1} - x_i) + \frac{1}{2!} \frac{d^2 y}{dx^2} \Big|_{x_i, y_i} (x_{i+1} - x_i)^2 + \frac{1}{3!} \frac{d^3 y}{dx^3} \Big|_{x_i, y_i} (x_{i+1} - x_i)^3 + \frac{1}{4!} \frac{d^4 y}{dx^4} \Big|_{x_i, y_i} (x_{i+1} - x_i)^4 \quad (4.10)$$

Knowing that $\frac{dy}{dx} = f(x, y)$ and $x_{i+1} - x_i = h$

$$y_{i+1} = y_i + f(x_i, y_i)h + \frac{1}{2!} f'(x_i, y_i)h^2 + \frac{1}{3!} f''(x_i, y_i)h^3 + \frac{1}{4!} f'''(x_i, y_i)h^4 \quad (4.11)$$

Based on equating equation (4.8) and equation (4.10), one of the popular solutions used is

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1h\right) \quad (4.12)$$

$$k_3 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2h\right)$$

$$k_4 = f(x_i + h, y_i + k_3h)$$

Here as an example we evaluate the following heat transfer problem to investigate 4th order Runge-Kutta method.

A ball at 1200 K is allowed to cool down in air at an ambient temperature of 300 K. Assuming heat is lost only due to radiation, the differential equation for the temperature of the ball is given by

$$\frac{d\theta}{dt} = -2.2067 \times 10^{-12} (\theta^4 - 81 \times 10^8), \theta(0) = 1200 \text{ K}$$

where θ is in Kelvin and t in seconds. We want to find the temperature at $t = 480$ seconds using Runge-Kutta 4th order method. Assume a step size of $h = 240$ seconds.

$$\frac{d\theta}{dt} = -2.2067 \times 10^{-12} (\theta^4 - 81 \times 10^8)$$

$$f(t, \theta) = -2.2067 \times 10^{-12} (\theta^4 - 81 \times 10^8)$$

$$\theta_{i+1} = \theta_i + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)h$$

For $i = 0$, $t_0 = 0$, $\theta_0 = 1200 \text{ K}$

$$\begin{aligned} k_1 &= f(t_0, \theta_0) \\ &= f(0, 1200) \\ &= -2.2067 \times 10^{-12} (1200^4 - 81 \times 10^8) \\ &= -4.5579 \end{aligned}$$

$$\begin{aligned} k_2 &= f\left(t_0 + \frac{1}{2}h, \theta_0 + \frac{1}{2}k_1h\right) \\ &= f\left(0 + \frac{1}{2}(240), 1200 + \frac{1}{2}(-4.5579) \times 240\right) \\ &= f(120, 653.05) \\ &= -2.2067 \times 10^{-12} (653.05^4 - 81 \times 10^8) \\ &= -0.38347 \end{aligned}$$

$$\begin{aligned}
k_3 &= f\left(t_0 + \frac{1}{2}h, \theta_0 + \frac{1}{2}k_2h\right) \\
&= f\left(0 + \frac{1}{2}(240), 1200 + \frac{1}{2}(-0.38347) \times 240\right) \\
&= f(120, 1154.0) \\
&= -2.2067 \times 10^{-12} (1154.0^4 - 81 \times 10^8) \\
&= -3.8954
\end{aligned}$$

$$\begin{aligned}
k_4 &= f(t_0 + h, \theta_0 + k_3h) \\
&= f(0 + 240, 1200 + (-3.894) \times 240) \\
&= f(240, 265.10) \\
&= -2.2067 \times 10^{-12} (265.10^4 - 81 \times 10^8) \\
&= 0.0069750
\end{aligned}$$

$$\begin{aligned}
\theta_1 &= \theta_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h \\
&= 1200 + \frac{1}{6}(-4.5579 + 2(-0.38347) + 2(-3.8954) + (0.069750))240 \\
&= 1200 + (-2.1848) \times 240 \\
&= 675.65 \text{ K}
\end{aligned}$$

θ_1 is the approximate temperature at

$$\begin{aligned}
t &= t_1 \\
&= t_0 + h
\end{aligned}$$

$$= 0 + 240$$

$$= 240$$

$$\theta_1 = \theta(240)$$

$$\approx 675.65 \text{ K}$$

For $i=1, t_1 = 240, \theta_1 = 675.65 \text{ K}$

$$k_1 = f(t_1, \theta_1)$$

$$= f(240, 675.65)$$

$$= -2.2067 \times 10^{-12} (675.65^4 - 81 \times 10^8)$$

$$= -0.44199$$

$$k_2 = f\left(t_1 + \frac{1}{2}h, \theta_1 + \frac{1}{2}k_1h\right)$$

$$= f\left(240 + \frac{1}{2}(240), 675.65 + \frac{1}{2}(-0.44199)240\right)$$

$$= f(360, 622.61)$$

$$= -2.2067 \times 10^{-12} (622.61^4 - 81 \times 10^8)$$

$$= -0.31372$$

$$k_3 = f\left(t_1 + \frac{1}{2}h, \theta_1 + \frac{1}{2}k_2h\right)$$

$$= f\left(240 + \frac{1}{2}(240), 675.65 + \frac{1}{2}(-0.31372) \times 240\right)$$

$$= f(360, 638.00)$$

$$= -2.2067 \times 10^{-12} (638.00^4 - 81 \times 10^8)$$

$$= -0.34775$$

$$k_4 = f(t_1 + h, \theta_1 + k_3 h)$$

$$= f(240 + 240, 675.65 + (-0.34775) \times 240)$$

$$= f(480, 592.19)$$

$$= 2.2067 \times 10^{-12} (592.19^4 - 81 \times 10^8)$$

$$= -0.25351$$

$$\theta_2 = \theta_1 + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) h$$

$$= 675.65 + \frac{1}{6} (-0.44199 + 2(-0.31372) + 2(-0.34775) + (-0.25351)) \times 240$$

$$= 675.65 + \frac{1}{6} (-2.0184) \times 240$$

$$= 594.91 \text{K}$$

θ_2 is the approximate temperature at

$$t = t_2$$

$$= t_1 + h$$

$$= 240 + 240$$

$$= 480$$

$$\theta_2 = \theta(480)$$

$$\approx 594.91 \text{K}$$

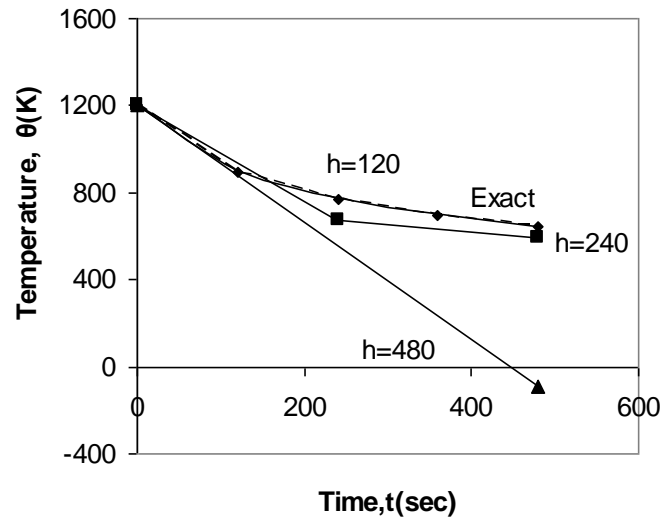


Figure 4.3 Comparison of Runge-Kutta 4th order method with exact solution for different step sizes

Figure 4.3 compares the exact solution with the numerical solution using the Runge-Kutta 4th order method with different step sizes. Table 4.1 and Figure 4.4 show the effect of step size on the value of the calculated temperature at $t=480$ seconds.

Table 4.1 Value of temperature at $t=480$ second for different step sizes

| Step size, h | $\theta(480)$ | E_t |
|----------------|---------------|------------|
| 480 | -90.278 | 737.85 |
| 240 | 594.91 | 52.660 |
| 120 | 646.16 | 1.4122 |
| 60 | 647.54 | 0.033626 |
| 30 | 647.57 | 0.00086900 |

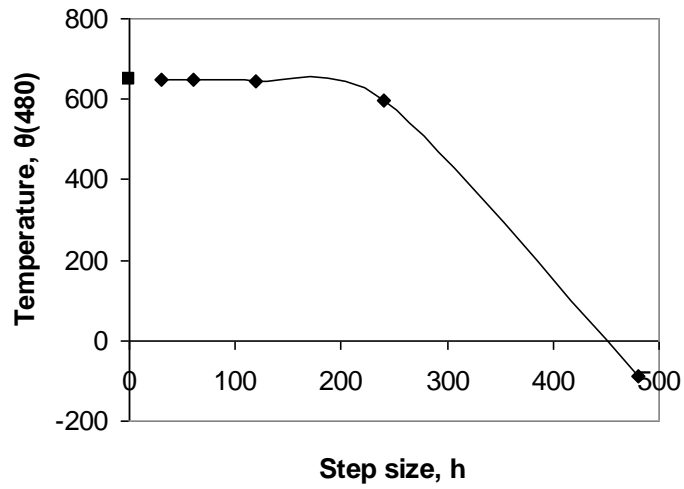


Figure 4.4 Effect of step size in Runge-Kutta 4th order method.

In Figure 4.5 the exact results with Euler's method (Runge-Kutta 1st order method), Heun's method (Runge-Kutta 2nd order method), and Runge-Kutta 4th order method are compared.

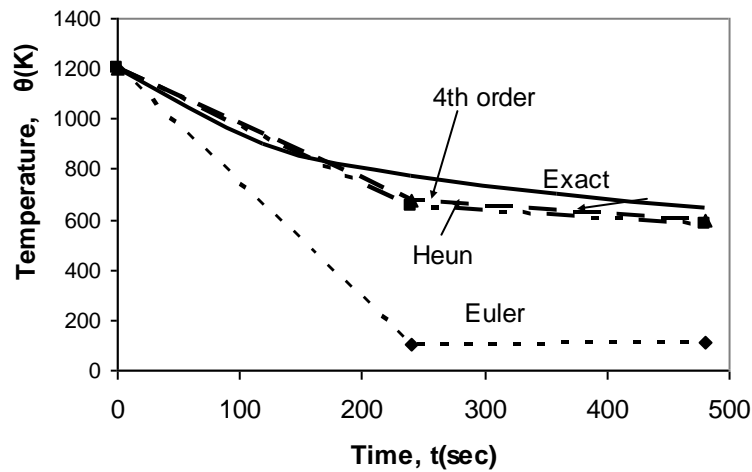


Figure 4.5 Comparison of Runge-Kutta methods of 1st (Euler), 2nd, and 4th order.

The formula described in this chapter was developed by Runge. There are other versions of the 4th order method just like there are several versions of the second order methods. The formula developed by Kutta is

$$\begin{aligned}
 y_{i+1} &= y_i + \frac{1}{8}(k_1 + 3k_2 + 3k_3 + k_4)h \\
 k_1 &= f(x_i, y_i) \\
 k_2 &= f\left(x_i + \frac{1}{3}h, y_i + \frac{1}{3}hk_1\right) \\
 k_3 &= f\left(x_i + \frac{2}{3}h, y_i - \frac{1}{3}hk_1 + hk_2\right) \\
 k_4 &= f(x_i + h, y_i + hk_1 - hk_2 + hk_3)
 \end{aligned}
 \tag{4.13}$$

This formula is the same as the Simpson's 3/8 rule, if $f(x, y)$ is only a function of x .

Chapter 5 Results

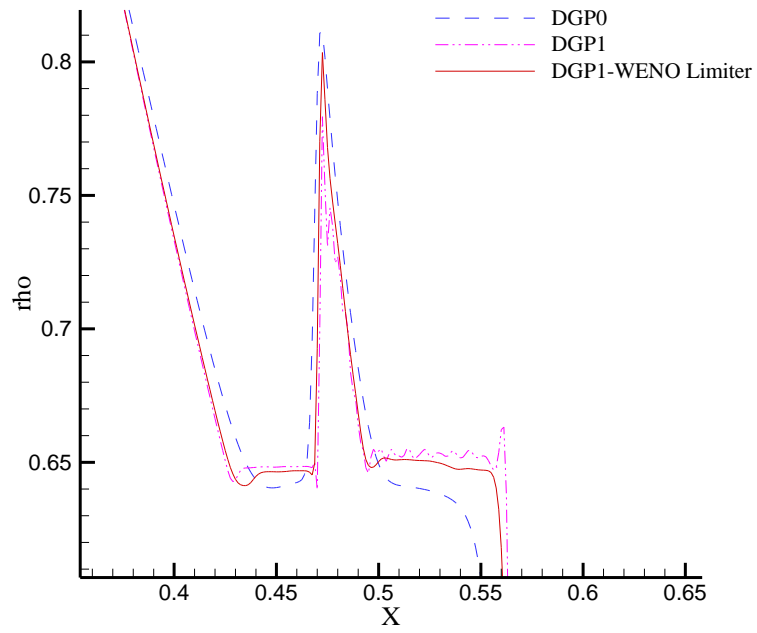
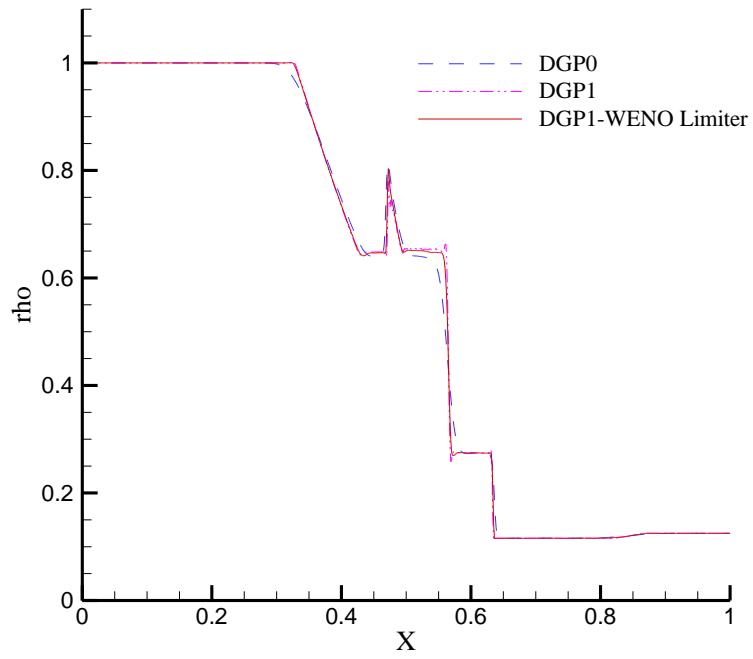
Computational results for a number of test cases are presented in this Chapter. All the algorithms presented above are coded using the programming language Fortran F95. Since the present RDG method is developed and implemented on arbitrary grids, which can contain one or combination of the four most common element types: tetrahedra, prisms, pyramids, and hexahedra, the resultant RDG3D code has the ability to compute 1D, 2D, and 3D problems, which greatly alleviates the need and pain for the code maintenance and upgrade. Results for one-dimensional flow problems can be readily obtained by setting the number of cells in both y- and z-directions to be 1 using a hexahedral grid. For two-dimensional problems, the number of cells in the z-direction is simply set to be 1, while both hexahedral and prismatic cells can be adopted, that are equivalent to quadrilateral and triangular grids in 2D, respectively. A number of test cases are chosen to test if the designed RDG(P₁P₂) method is able to achieve the designed third-order rate of convergence for smooth flows problems, while a number of test cases are used to assess the non-oscillatory property of the HWENO(P₁P₂) method for flows with strong discontinuities. Time integration is carried out by the 4th order Runge-Kutta method to ensure that the spatial discretization errors are dominant in these cases.

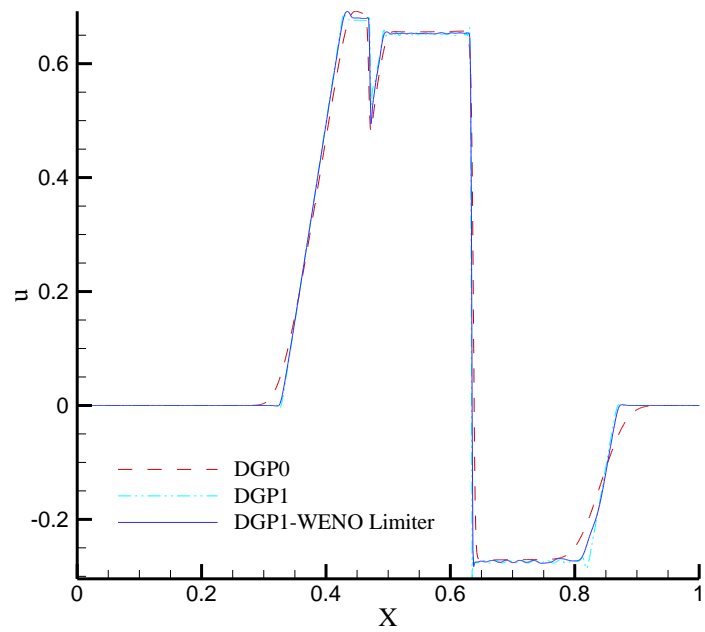
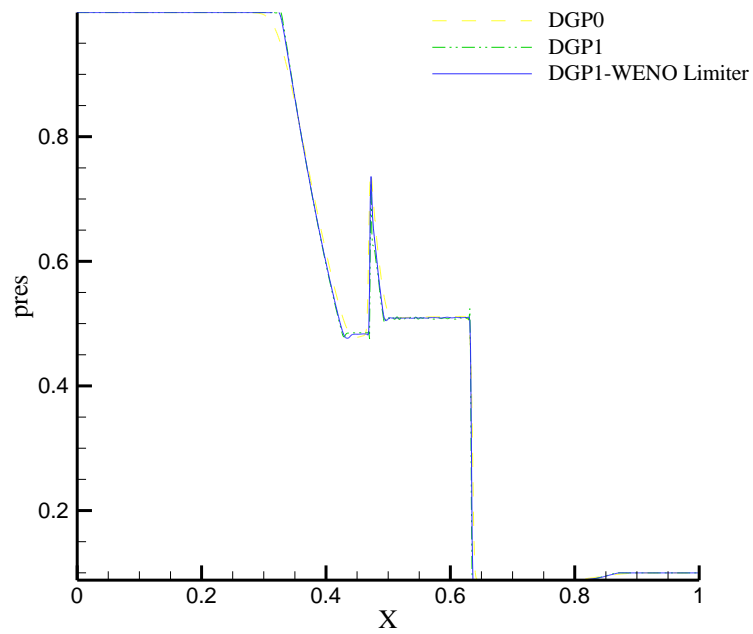
5.1 One Dimensional Problems

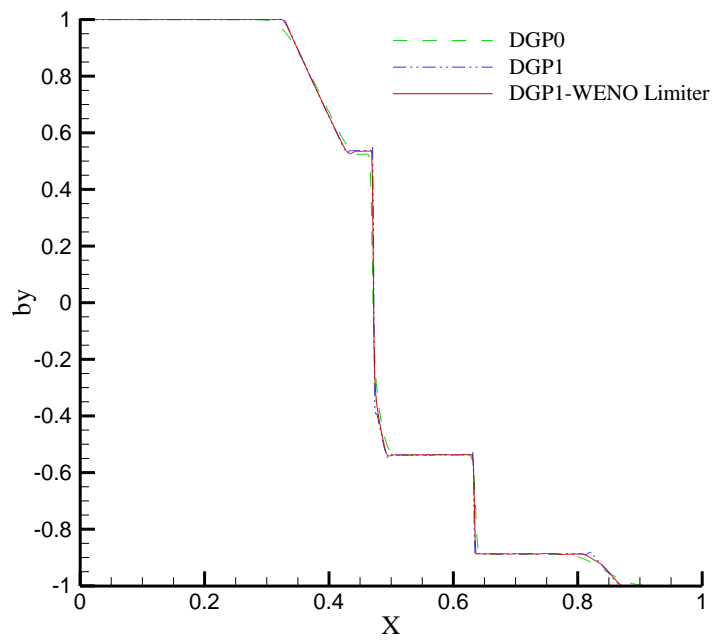
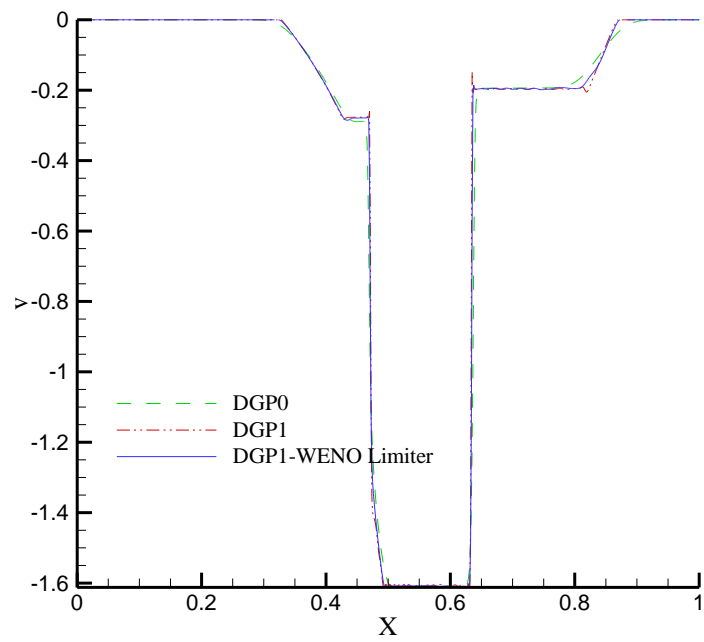
Two one-dimensional shock tube problems presented at [93] are considered in this section. The numerical solutions are obtained using 800 grid cells and a CFL number of 0.6. The

initial conditions for the first test are the following : $(\rho ,p ,u ,v ,w ,B_x ,B_y ,B_z)=(1.0 ,1.0 ,0.0 ,0.0 ,0.0 , 0.75, 1.0 , 0.0)$ for $0\leq x\leq 0.5$ and $(\rho ,p ,u ,v ,w ,B_x ,B_y ,B_z)=(0.125 ,0.1 ,0.0 ,0.0 ,0.0 , 0.75, -1.0 , 0.0)$ for $0.5\leq x\leq 1.0$. The numerical solutions obtained by the three DG(P0), DG(P1), and WENO(P1P1) (DGP1-WENO limiter) methods at time $t=0.1$ are presented in Fig. 5.1. Since this is 1D problem, the flux functions in both Y- and Z- directions, \mathbf{G} and \mathbf{H} , are set to be zero.

Figure 5.1 One dimensional shock tube problem with $\gamma=5/3$ at time $t=0.1$ Results of different solvers DGP0, DGP1, and DGP1-WENO-limiter are shown for density, pressure, u, v , and B_y

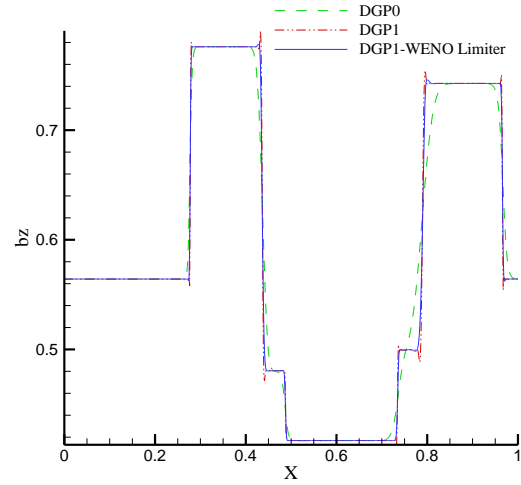
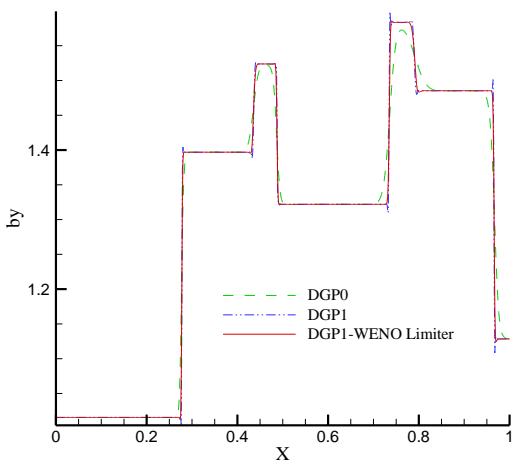
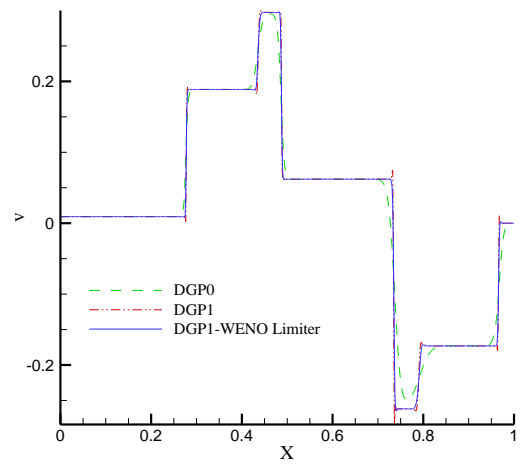
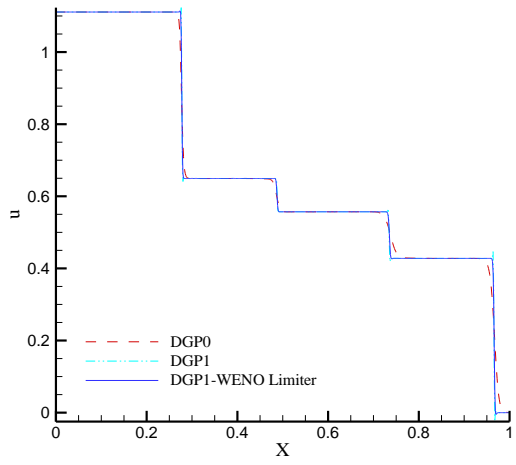
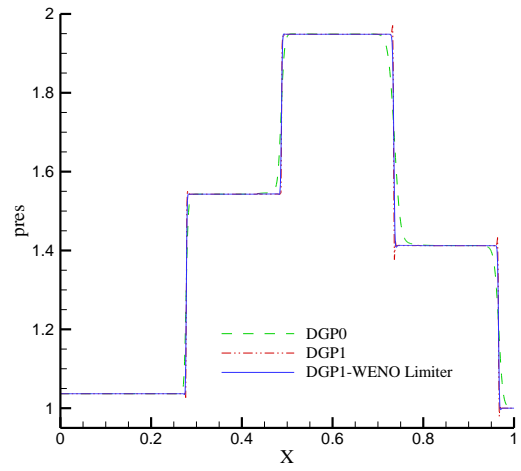
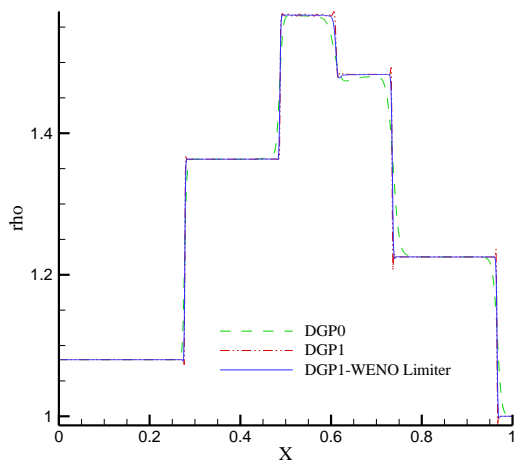






The initial conditions for the second test case are set to be $(\rho, p, u, v, w, B_x, B_y, B_z) = (1.08, 0.95, 1.2, 0.01, 0.5, 4/\sqrt{4\pi}, 3.6/\sqrt{4\pi}, 2.0/\sqrt{4\pi})$ for $0 \leq x \leq 0.5$ and $(\rho, p, u, v, w, B_x, B_y, B_z) = (1.0, 1.0, 0.0, 0.0, 0.0, 4/\sqrt{4\pi}, 4.0/\sqrt{4\pi}, 2.0/\sqrt{4\pi})$ for $0.5 \leq x \leq 1.0$. This problem has a strong discontinuity in the middle of domain that propagates in terms of shock and expansion waves across domain. The numerical solutions at time $t=0.2$ obtained by the DG(P₀) and DG(P₁) and WENO(P₁P₁), are presented in Figure 5.2, where the HLLD Riemann solver is used to compute the numerical fluxes at interfaces. One can observe that DG(P₁) with WENO limiter, ie., the WENO(P₁P₁) method yields the best results among the three methods, being less dissipative and dispersive around sharp corners, though all three methods produce the same results for the constant smooth regions as expected.

Figure 5.2 One dimensional shock tube problem with $\gamma=5/3$ at time $t=0.2$ Results of different solvers DGP₀, DGP₁, and WENO limiter-DGP₁ with HLLD Riemann solvers are shown for: density(top left), Pressure (top right), u (mid left), v (mid right) B_y (bottom left)



5.2 Two Dimensional Test Cases

This section presents the numerical results obtained using the RDG methods for the two-dimensional MHD problems on both structured and unstructured grids. Since these are 3D simulations of 2D problems, the number of cells in the Z-direction is set to be one.

5.2.1 MHD Blast Wave

The classical MHD blast wave problem [93] is considered in this test case, where $\gamma=1$. The computational domain for this problem is a square with a length of unity, i.e. $[0,1]*[0,1]$. In the middle of domain there is a high pressure part with a radius of 0.1 which has pressure of 1000 and pressure at rest of domain is 0.1. Initially, the flow is at rest and has density of unity and $\vec{B} = (100/\sqrt{4\pi}, 0, 0)$. Figure 5.3 shows two types of computational grids used for this MHD blast wave problem. The computational results at time $t=0.01$ obtained on structured hexahedral grids are shown in Figs 5.4-11, while those obtained on unstructured prismatic grids are shown in Figure 5.12-5.19. Figure 5.20 compares the density profiles at $y=0.5$ obtained by the DG(P_0) and WENO(P_1P_1) solutions with different values of γ in Eq. (3.80). Clearly, the numerical solutions are not very sensitive to the choice of weighting coefficient γ . Note that the DG(P_1) method suffers from the non-linear instability because of the strong shocks, unable to produce a stable solution for this problem.

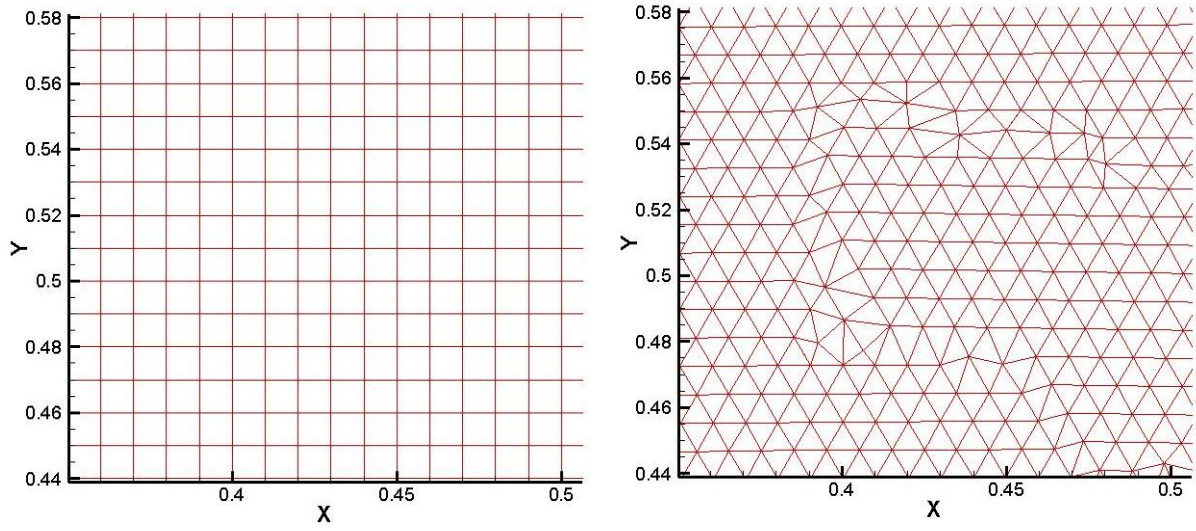


Figure 5.3 Hexahedral and Prismatic cells used for MHD Blast wave in a domain of $[0,1] \times [0,1]$

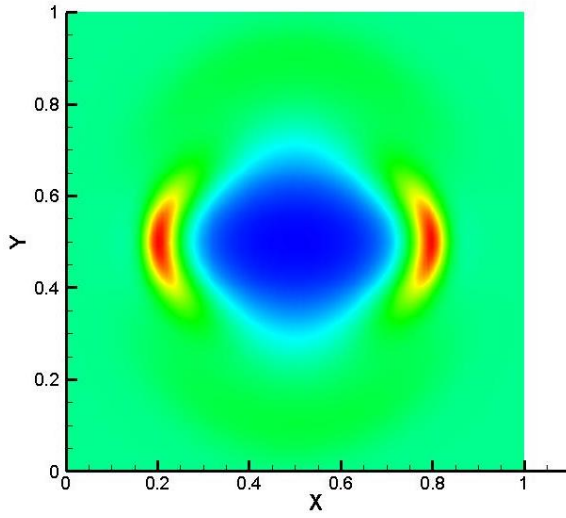


Figure 5.4 Density contours for MHD blast wave on hexahedral grid

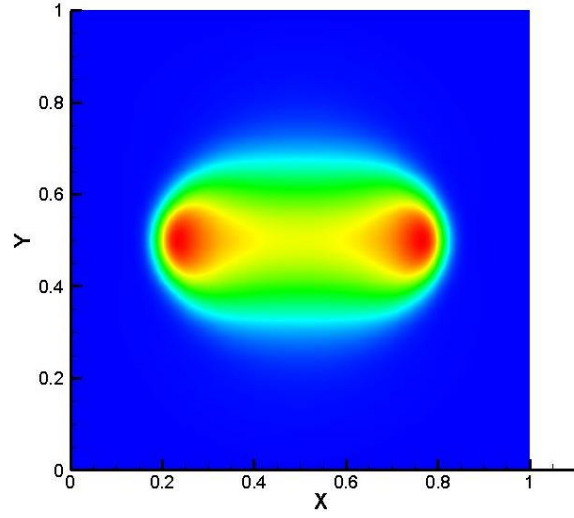


Figure 5.5 Pressure contours for MHD blast wave on hexahedral grid

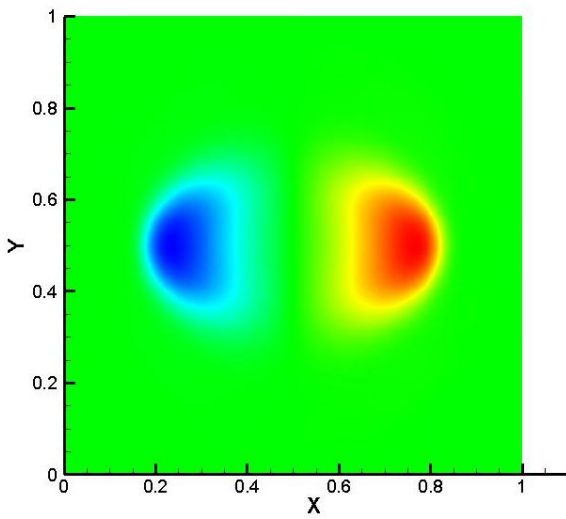


Figure 5.6 X-velocity component contours for MHD blast wave on hexahedral grid

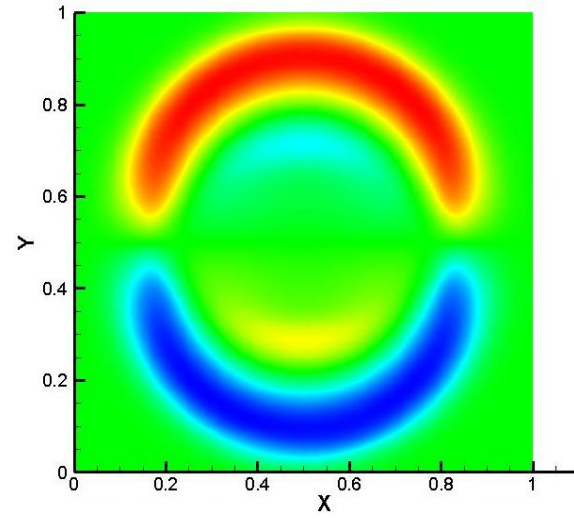


Figure 5.7 Y-velocity component contours for MHD blast wave on hexahedral grid

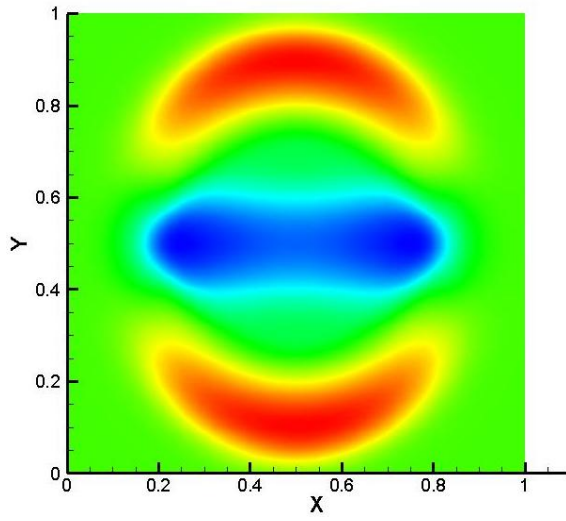


Figure 5.8 B_x contours for MHD blast wave on hexahedral grid

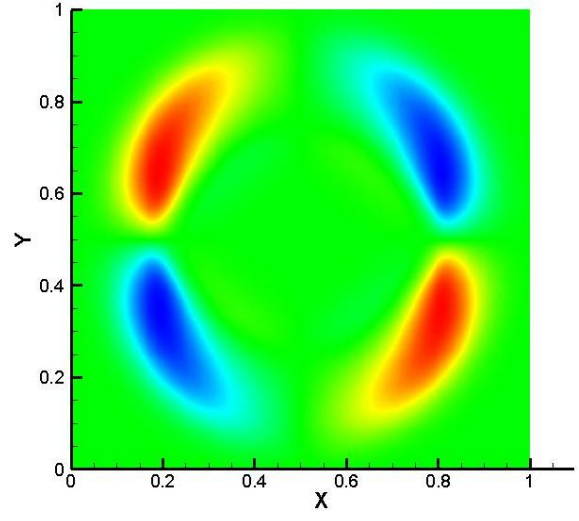


Figure 5.9 B_y contours for MHD blast wave on hexahedral grid

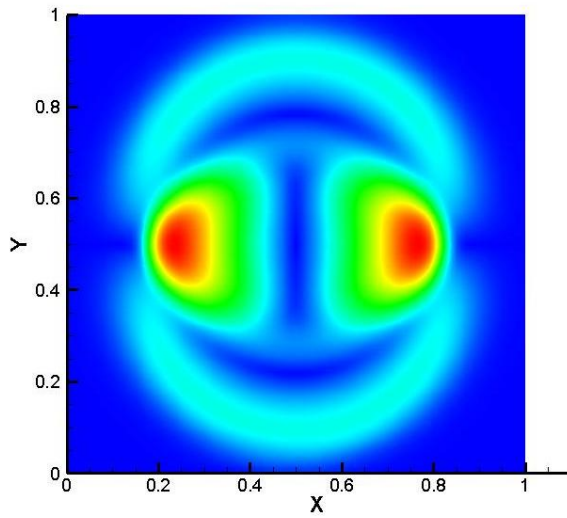


Figure 5.10 Velocity magnitude contours for MHD blast wave on hexahedral grid

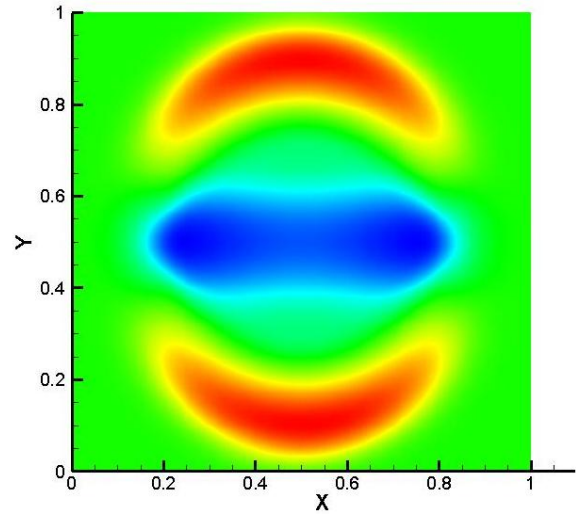


Figure 5.11 Magnetic field magnitude contours for MHD blast wave on hexahedral grid

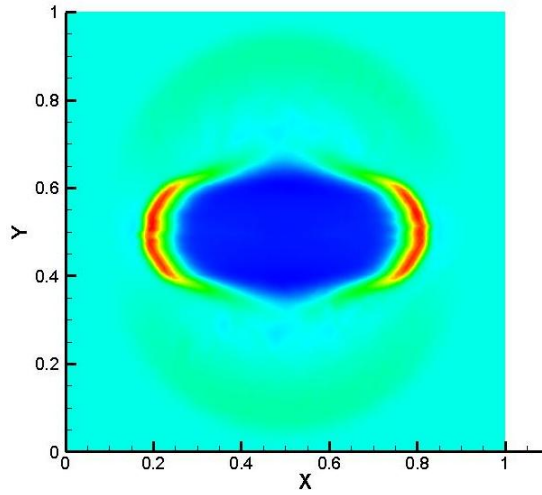


Figure 5.12 Density contours for MHD blast wave on Prismatic cells

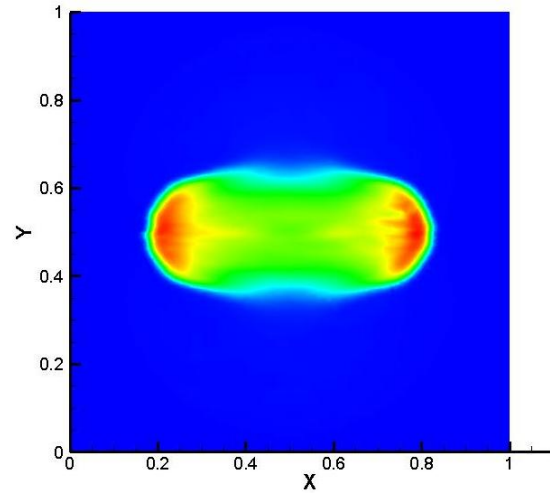


Figure 5.13 Pressure contours for MHD blast wave on Prismatic cells

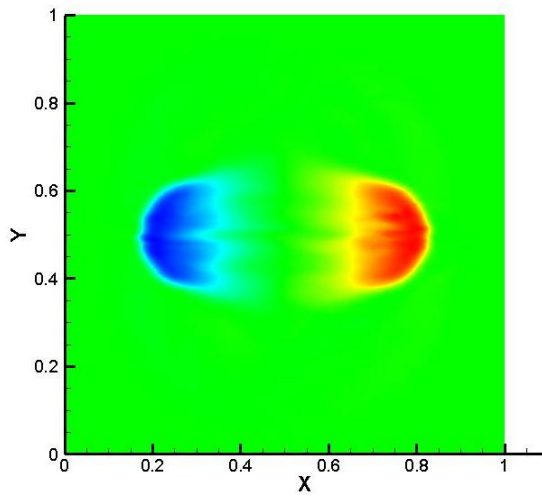


Figure 5.14 X-velocity contours for MHD blast wave on Prismatic cells

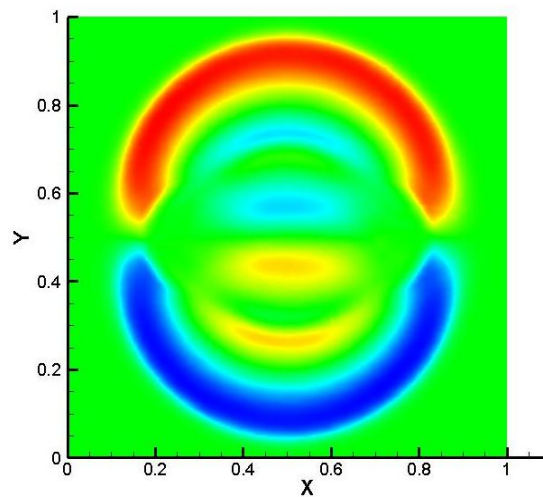


Figure 5.15 Y-velocity contours for MHD blast wave on Prismatic cells

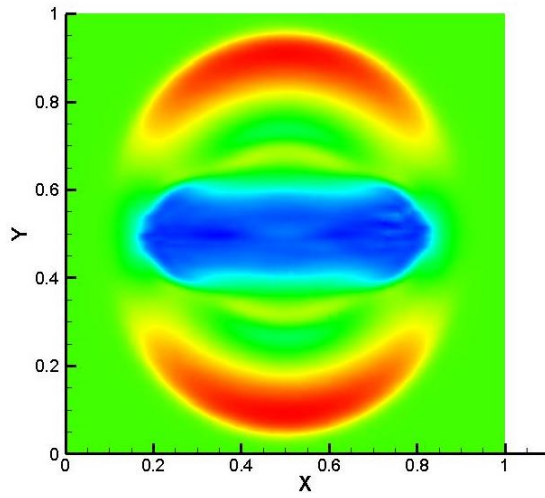


Figure 5.16 B_x contours for MHD blast wave on Prismatic cells

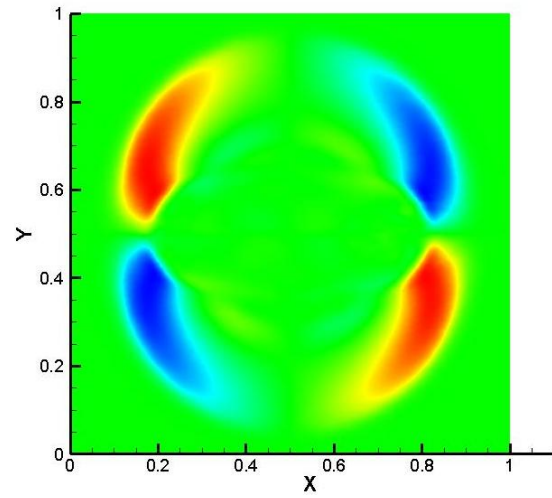


Figure 5.17 B_y contours for MHD blast wave on Prismatic cells

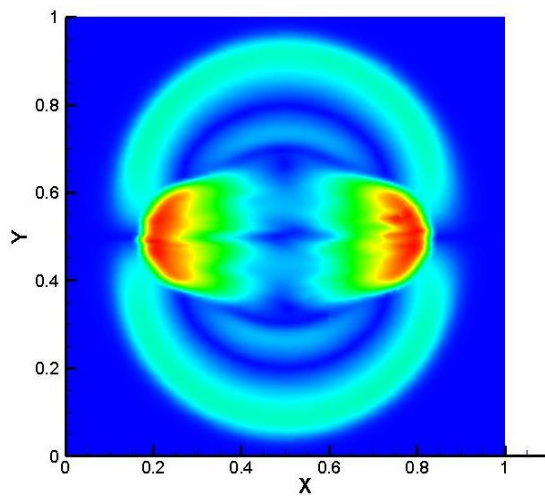


Figure 5.18 Velocity magnitude contours for MHD blast wave on Prismatic cells

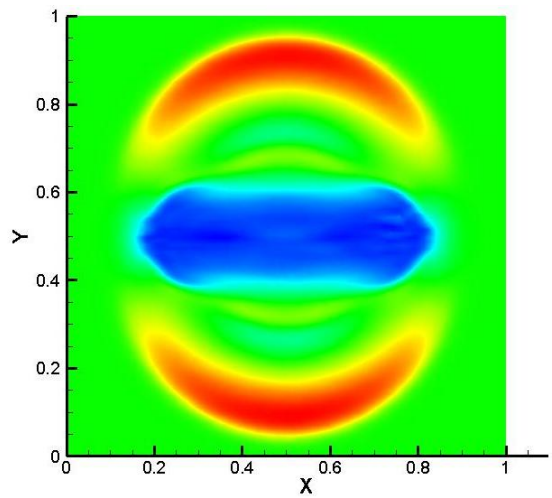


Figure 5.19 Magnetic field magnitude contours for MHD blast wave on Prismatic cells

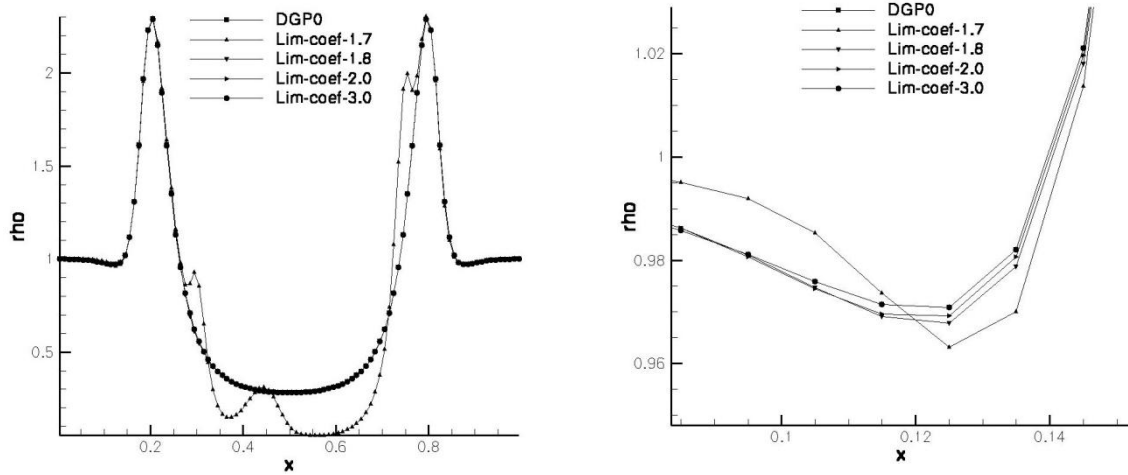


Figure 5.20 Effect of WENO limiter on DGP1 results for MHD blast wave. Density values at $y=0.5$

5.2.2 Two Dimensional Vortex

The convection of an inviscid isentropic vortex is a well-known test case, which is widely used in the literature to assess the accuracy of the numerical methods. The exact solution for this case at any time t is the initial solution translated over a distance $V_\infty t$ for a convecting vortex, providing a valuable reference for measuring the accuracy of the numerical solution. The initial condition is a linear superposition of a mean uniform flow with some perturbations.¹ The domain assumed to have a uniform flow with initial conditions $(\rho_\infty, u_\infty, v_\infty, p_\infty, B_x, B_y, B_z) = (1, 1, 1, 1, 0, 0, 0)$ all over the field. The perturbation as follow is added to the main flow

$$\begin{aligned} \begin{pmatrix} \delta u \\ \delta v \end{pmatrix} &= \frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} \begin{pmatrix} -(y - y_0) \\ (x - x_0) \end{pmatrix} \\ \delta S &= 0 \\ \delta T &= -\frac{(\gamma - 1)\varepsilon^2}{8\pi^2\gamma} e^{1-r^2} \end{aligned}$$

Where $r^2 = (x - x_0)^2 + (y - y_0)^2$, x_0 and y_0 are the coordinate of center of vortex, and ε is the vortex strength. From $\rho = \rho_\infty + \delta\rho$, $u = u_\infty + \delta u$, $v = v_\infty + \delta v$, $T = T_\infty + \delta T$, and the isentropic relation,

other physical variables can be determined as follows

$$\rho = T^{\frac{1}{\gamma-1}}$$

$$P = \rho^\gamma$$

$$e = \frac{p}{\rho(\gamma - 1)} + \frac{1}{2}(u^2 + v^2)$$

This leads

$$\begin{pmatrix} \rho \\ u \\ v \\ w \\ p \\ T \\ B_x \\ B_y \\ B_z \end{pmatrix} = \begin{pmatrix} \rho \\ u_\infty + \delta u \\ v_\infty + \delta v \\ 0 \\ p \\ T_\infty + \delta T \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

In this test case, the vortex strength is $\varepsilon=5$, and the center of vortex is located at (5,5). The computational domain is a square of $[0,10] \times [0,10]$ and the periodic boundary conditions are imposed. The numerical solutions are obtained after one period of time $t=1$, and compared with the exact solution simply given by the initial condition. Figure 5.21 shows the two different types of meshes used in this test case. The computed results obtained on these two grids are presented in Figure 5.22-5.26. Figs. 5.27-5.28 provide the details of the spatial convergence in terms of L_1 -, L_2 -, and L_∞ -norm for this numerical experiment obtained by the DG (P_1) and RDG (P_1P_2) methods on prismatic and hexahedral grids, respectively. As expected, on both prismatic and hexahedral grids, the DG method exhibits a full $O(h^{p+1})$ order of convergence and the RDG method does offer a full $O(h^{p+2})$ order of the convergence, adding one order of accuracy to the underlying DG(P) method.

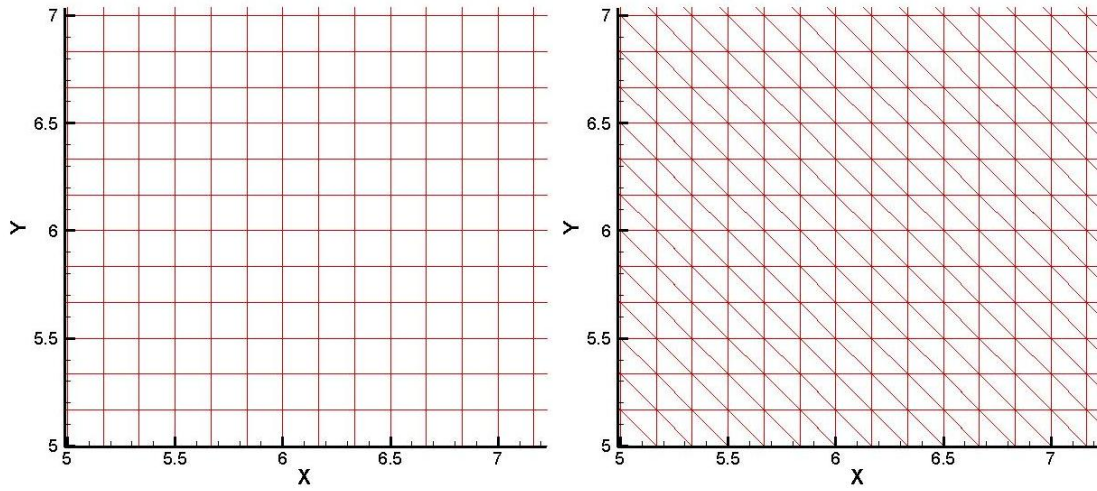


Figure 5.21 computational grid used for isentropic vortex, left: Hexahedral cells on x - y plane, Right Prismatic cells in x - y plane

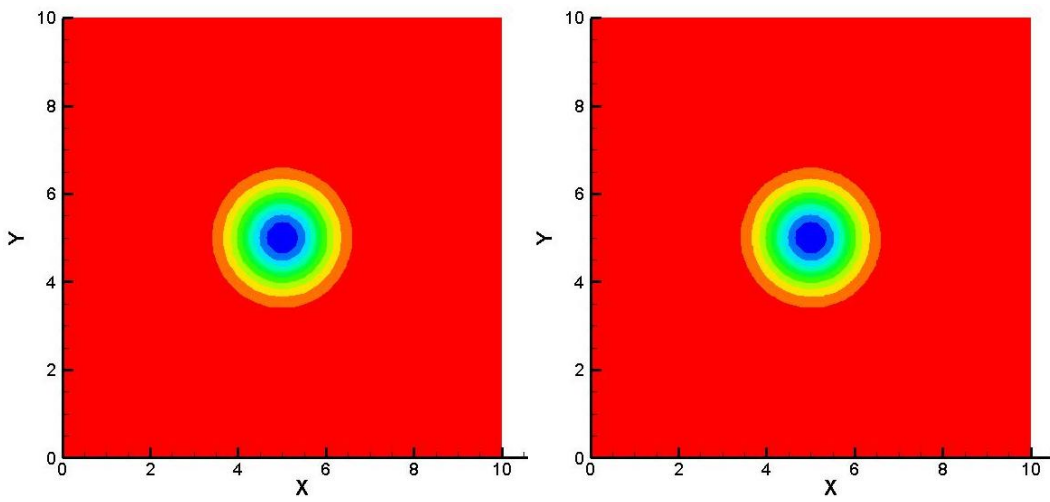


Figure 5.22 Density contours for fluid vortex problem, left: density on structured grid, right: density on unstructured grid

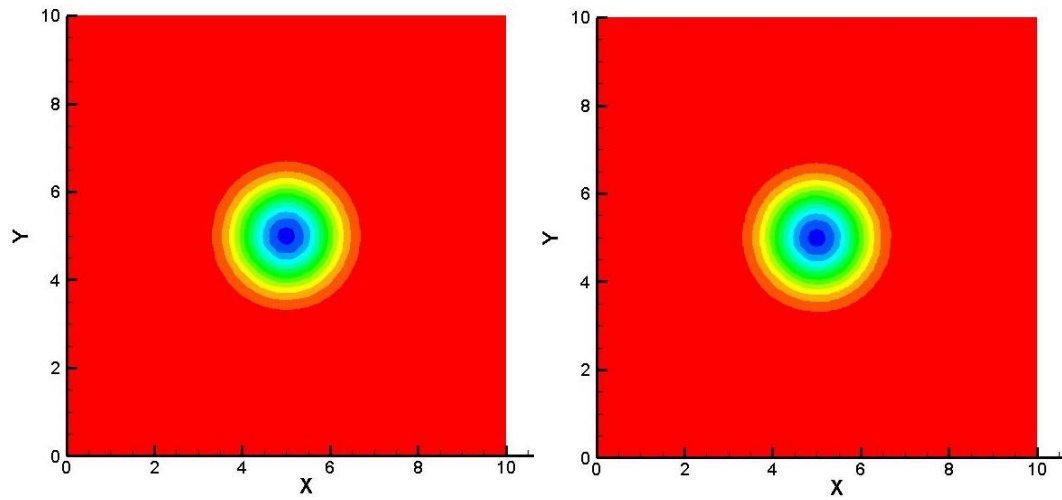


Figure 5.23 Pressure contours for fluid vortex problem, left: pressure on structured grid, right: pressure on unstructured grid

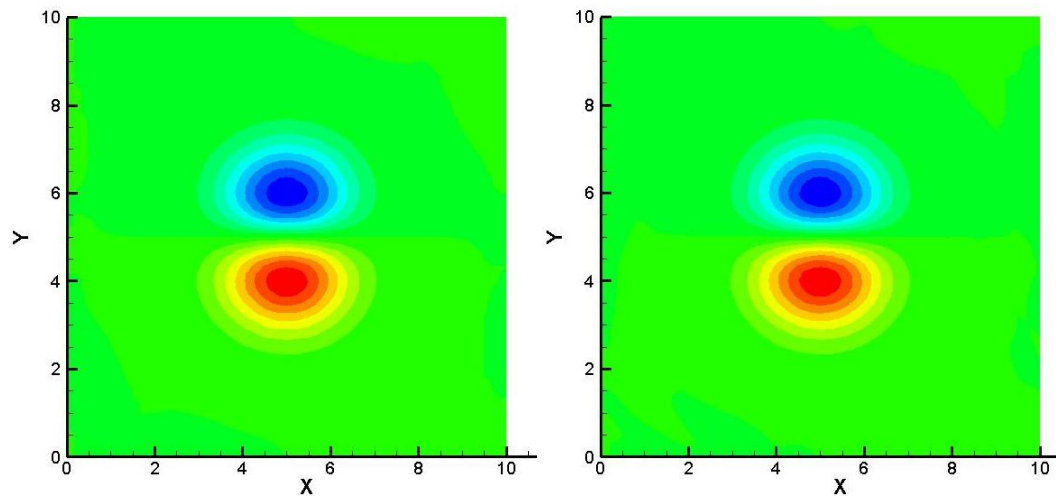


Figure 5.24 u contours for fluid vortex problem, left: velocity on structured grid, right: velocity on unstructured grid

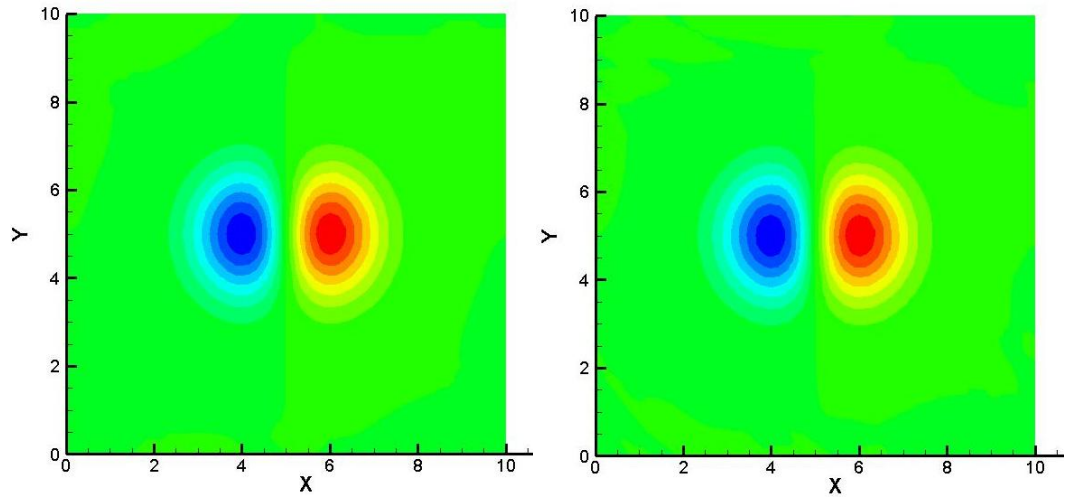


Figure 5.25 v contours for fluid vortex problem, left: velocity on structured grid, right: velocity on unstructured grid

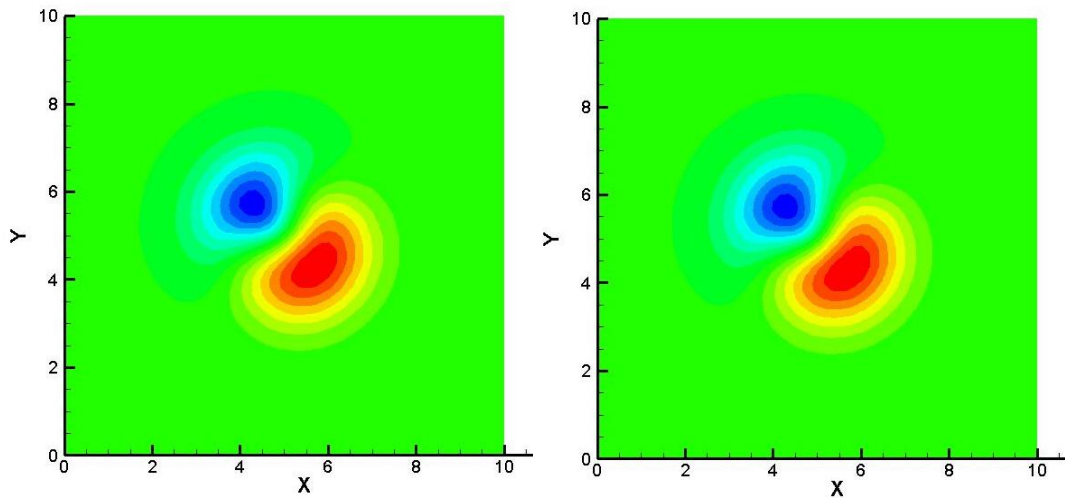


Figure 5.26 Velocity magnitude contours for fluid vortex problem, left: velocity on structured grid, right: velocity on unstructured grid

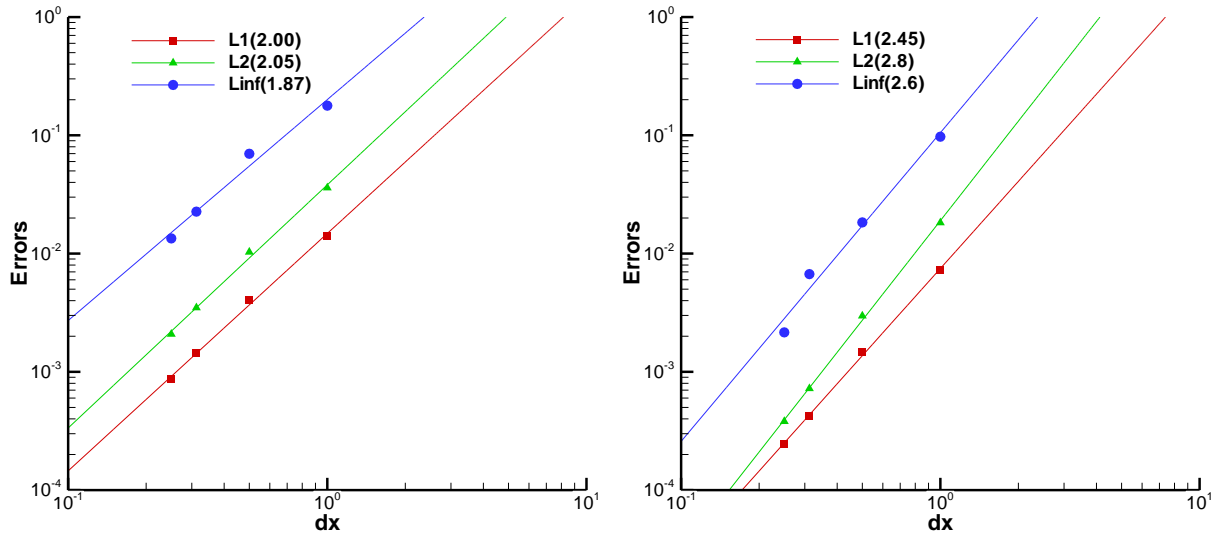


Figure 5.27 Error and order of accuracy for fluid vortex problem on Hexahedral cells. left: DGP₁, right: RDG

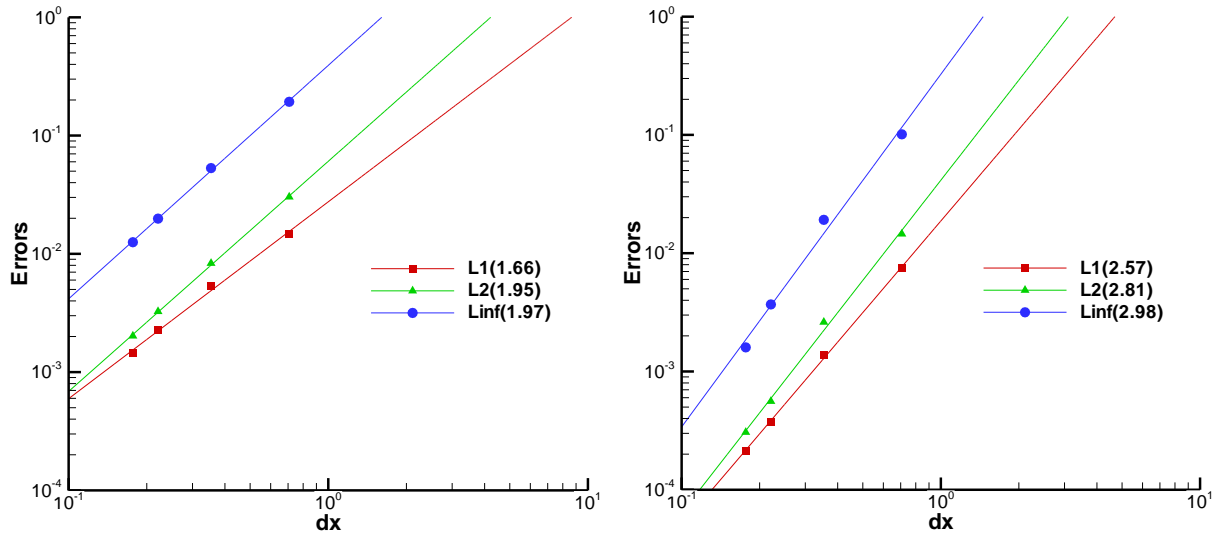


Figure 5.28 Error and order of accuracy for fluid vortex problem with Prismatic cells. left: DG(P₁), right: RDG(P₁P₂)

5.2.3 Two Dimensional MHD Vortex

This test case is the two-dimensional isentropic vortex problem [20] and very similar to the previous isentropic vortex, but here there exists perturbation at magnetic field as well. The vortex is initialized on a square $[0:10]*[0:10]$ with periodic BC's. The uniform flow condition is $(\rho, p, u, v, w, B_x, B_y, B_z) = (1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0)$ with perturbation on primitive variables as follow

$$\begin{aligned} \begin{pmatrix} \delta u_x \\ \delta u_y \end{pmatrix} &= \frac{\eta}{2\pi} \exp\left(\frac{1-r^2}{2}\right) \begin{pmatrix} (y_c - y) \\ -(x_c - x) \end{pmatrix} \\ \begin{pmatrix} \delta B_x \\ \delta B_y \end{pmatrix} &= \frac{\xi}{2\pi} \exp\left(\frac{1-r^2}{2}\right) \begin{pmatrix} (y_c - y) \\ -(x_c - x) \end{pmatrix} \\ \delta p &= (\xi^2(1-r^2) - \eta^2) \frac{1}{8\pi^2} \exp\left(\frac{1-r^2}{2}\right) \end{aligned}$$

where (x_c, y_c) is the coordinate of center of domain, and $r^2 = (x - x_c)^2 + (y - y_c)^2$, $\gamma=5/3$, $\eta=\xi=1$. All the computations are performed using a CFL number of 0.5. The computed density pressure, velocity, velocity magnitude $\left(\frac{u^2+v^2+w^2}{2}\right)$, and magnetic pressure $\left(\frac{b_x^2+b_y^2+b_z^2}{2}\right)$ contours at $t=10$ are presented in Fig. 5.29-5.37, respectively. For the sake of brevity, we just show the results obtained on prismatic grids, as the results obtained on hexahedral grids are virtually identical. As observed DGP₁ is 2nd order and RDG is 3rd order accurate in this problem. Figs. 5.38-5.39 provide the details of the spatial convergence in terms of L₁-, L₂-, and L_∞-norm for this two dimensional MHD vortex problem obtained by the DG(P₁) and RDG(P₁P₂) methods on prismatic and hexahedral grids, respectively. Again, on both prismatic and hexahedral grids, the DG(P₁) method exhibits a designed second order of

convergence and the RDG(P1P2) method offers a third order of the convergence, adding one order of accuracy to the underlying DG(P1) method.

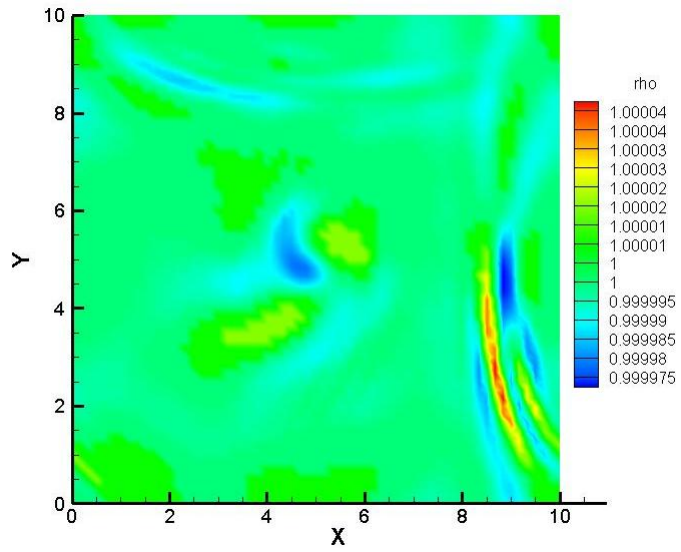


Figure 5.29 Density contours for MHD vortex on Prismatic cells

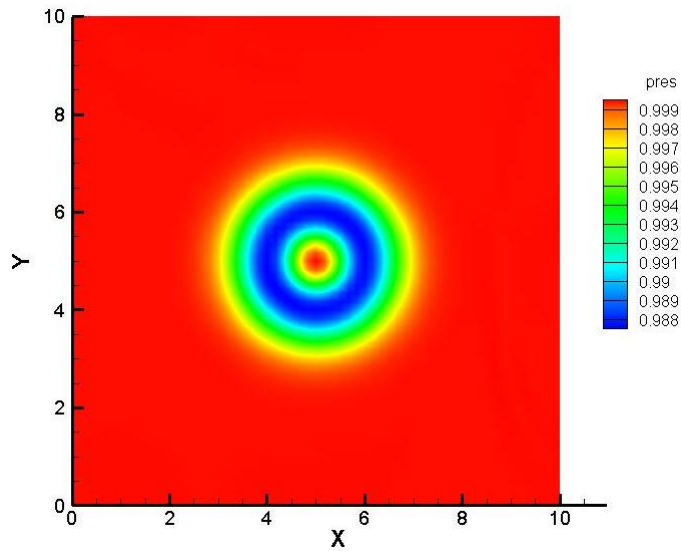


Figure 5.30 Pressure contours for MHD vortex on Prismatic cells

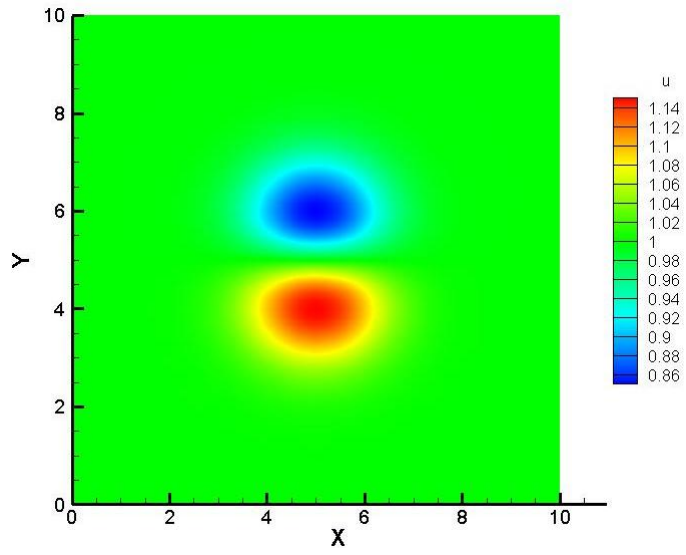


Figure 5.31 u contours for MHD vortex on Prismatic cells

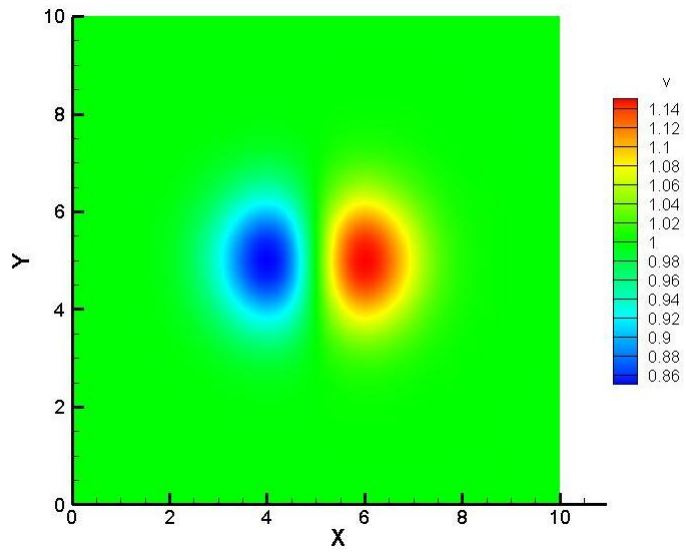


Figure 5.32 v contours for MHD vortex on Prismatic cells

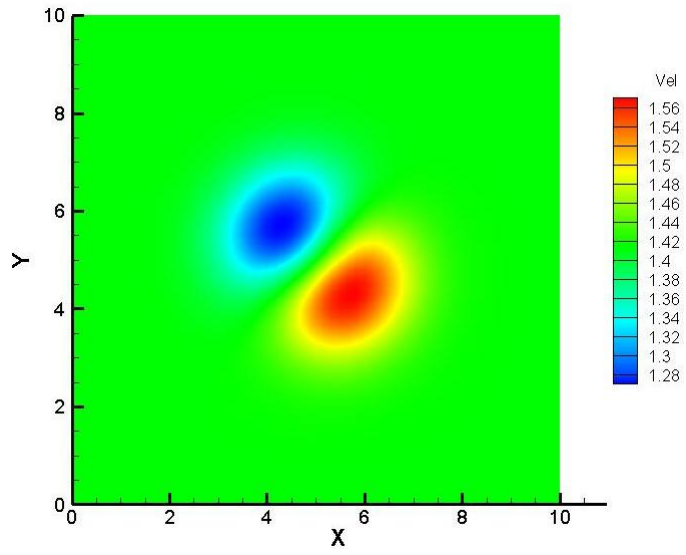


Figure 5.33 Velocity magnitude contours for MHD vortex on Prismatic cells

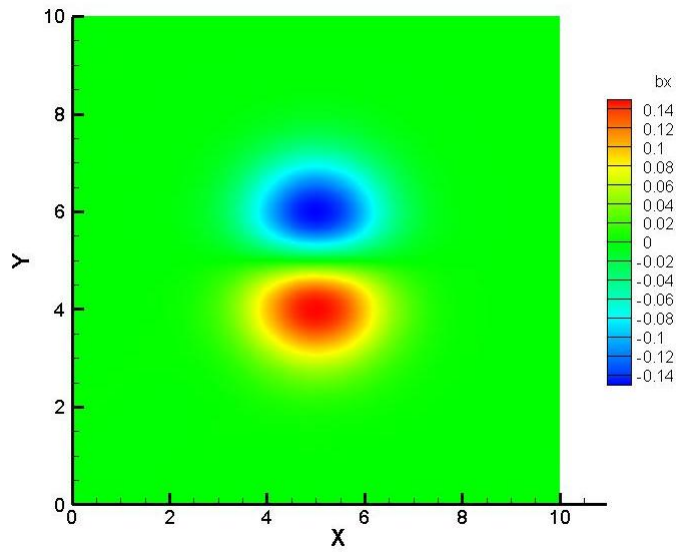


Figure 5.34 B_x of magnetic field contours for MHD vortex on Prismatic cells

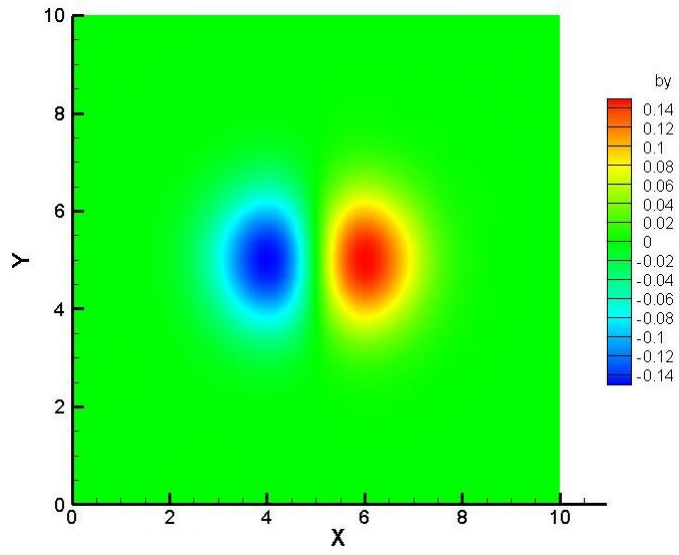


Figure 5.35 B_y of magnetic field contours for MHD vortex on Prismatic cells

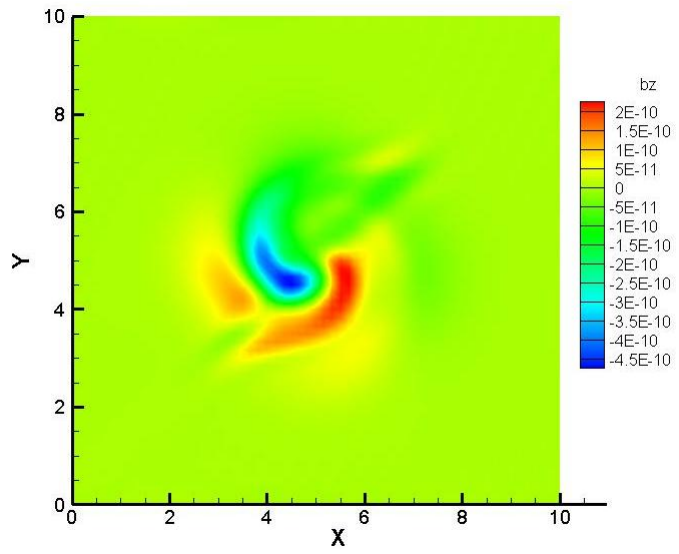


Figure 5.36 B_z of magnetic field contours for MHD vortex on Prismatic cells

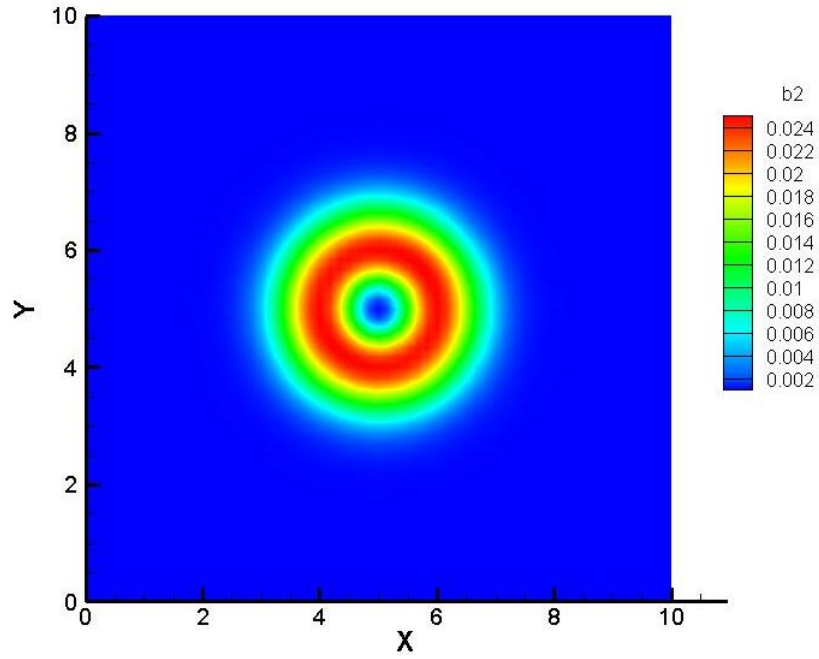


Figure 5.37 Magnetic pressure contours for MHD vortex on Prismatic cells

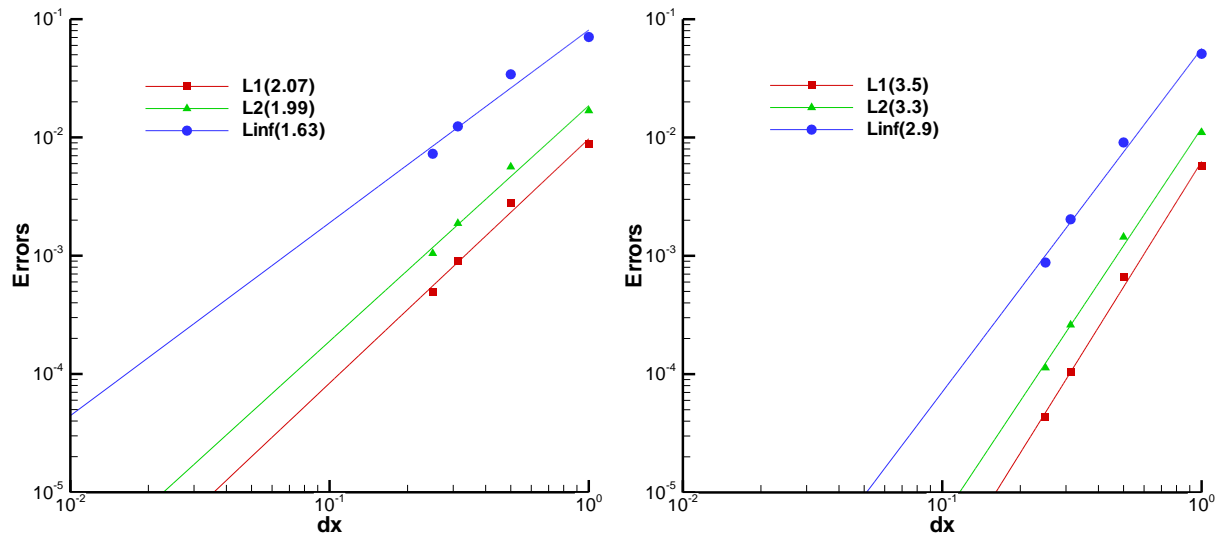


Figure 5.38 Error and order of accuracy for MHD vortex problem on hexahedral cells. left: DGP₁, right: RDG

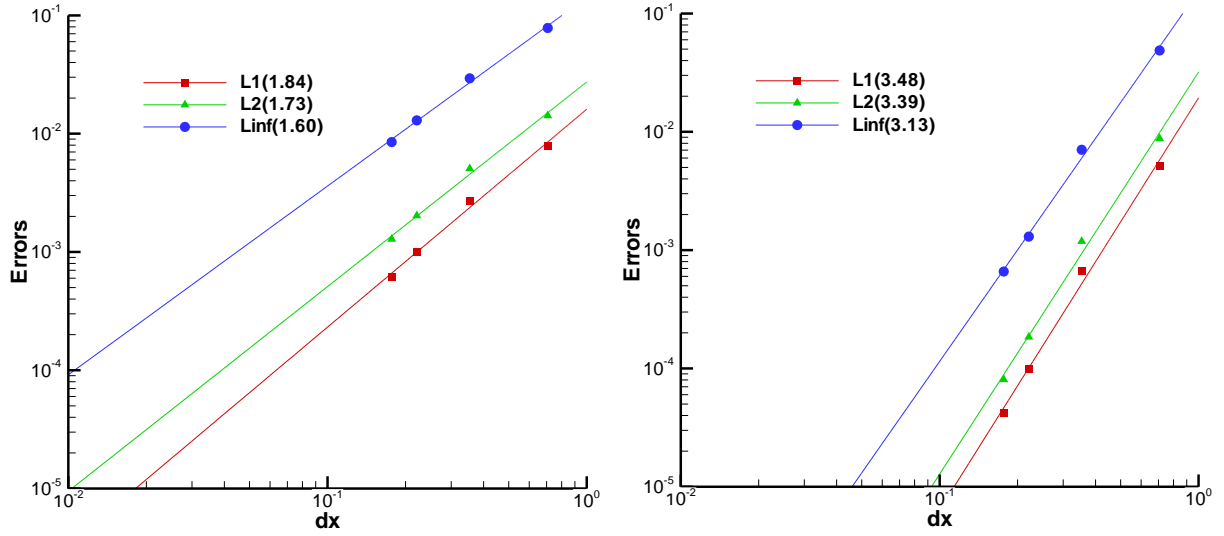


Figure 5.39 Error and order of accuracy for MHD vortex problem on Prismatic cells. left: DGP_1 , right: RDG

5.3 Three Dimensional Problems

Finally, a three dimensional MHD problem is considered in this test case. This test case is chosen to assess the accuracy of the RDG method for 3D problems.

5.3.1 Three Dimensional MHD Vortex

The initial condition for this problem comes from the two dimensional MHD vortex, where we rotate vector of initial condition for two dimensional MHD, 45 degree around y axis. The initial conditions are again $(\rho_\infty, u_\infty, v_\infty, p_\infty, B_x, B_y, B_z) = (1, 1, 1, 1, 0, 0, 0)$ all over the field. The perturbation added to the main flow is as follows

$$** \left\{ \begin{array}{l} \begin{array}{l} (\delta u) = \frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} \begin{pmatrix} -(y-y_0) \\ (x-x_0) \end{pmatrix} \\ \delta S = 0 \end{array} \\ \begin{array}{l} (\delta B_x) \\ (\delta B_y) \end{array} = \frac{\xi}{2\pi} \exp\left(\frac{1-r^2}{2}\right) \begin{pmatrix} (y_c - y) \\ -(x_c - x) \end{pmatrix} \\ \delta p = (\xi^2(1-r^2) - \eta^2) \frac{1}{8\pi^2} \exp\left(\frac{1-r^2}{2}\right) \end{array} \right.$$

where $r^2 = (x - x_0)^2 + (y - y_0)^2$, x_0 and y_0 are the coordinate of center of vortex, and ε is the vortex strength. The three dimensional rotation matrix around the y axis is

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix}$$

where $\theta = \frac{\pi}{4}$. The new coordinate system is then

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Using this new coordinate system at equations ** for initial condition and the rotational matrix over vector variables we will have the final initial condition

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{pmatrix} u_\infty + \delta u \\ v_\infty + \delta v \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} B_x \\ B_y \\ B_z \end{pmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{pmatrix} \delta B_x \\ \delta B_y \\ 0 \end{pmatrix}$$

In this test case, the vortex strength is $\varepsilon=5$, and the center of vortex is located at (0, 0). The computational domain, before rotation is a cube extended from [-5, 5] in three directions with

periodic boundary conditions. The numerical solutions obtained by both DG(P₁) and RDG(P₁P₂) are presented at one period of time t=10. Fig. 5.40 provides the details of the spatial convergence in terms of L₁-, L₂-, and L_∞-norm for this three dimensional MHD vortex problem obtained by the DG(P₁) and RDG(P₁P₂) methods, respectively. Similar to the 2D test cases, one can observe again that the DG(P₁) method exhibits a designed second order of convergence and the RDG(P₁P₂) method offers a third order of the convergence, adding one order of accuracy to the underlying DG(P₁) method for this 3D MHD problem. Finally, the computed density pressure, velocity, velocity magnitude $(u^2+v^2+w^2)/2$, and magnetic pressure $(b_x^2+b_y^2+b_z^2)/2$ contours at t=10 are presented in Fig. 5.41-5.48, respectively.

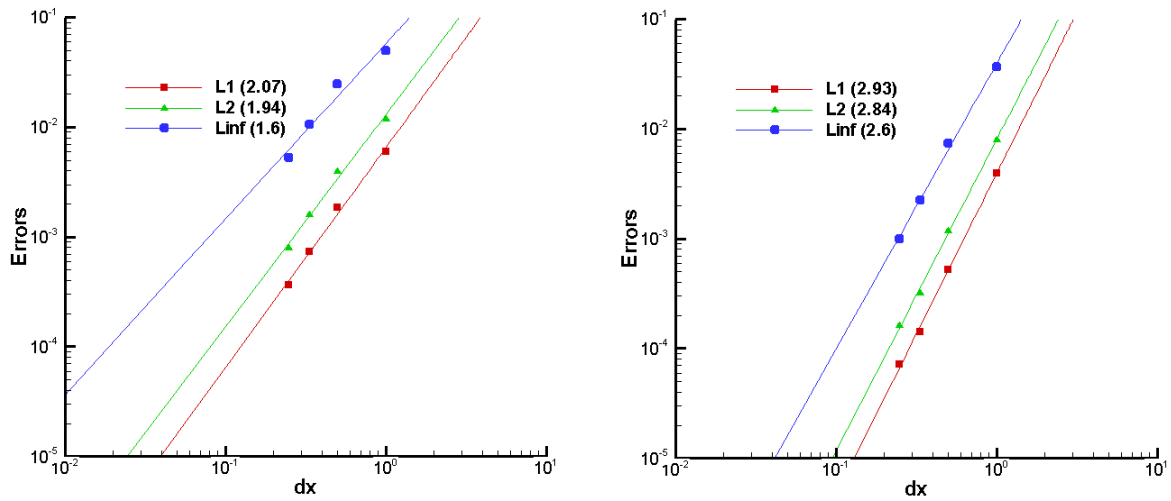


Figure 5.40 Errors and order of accuracy for DGP1 and RDG, three dimensional MHD vortex on Hexahedral cells

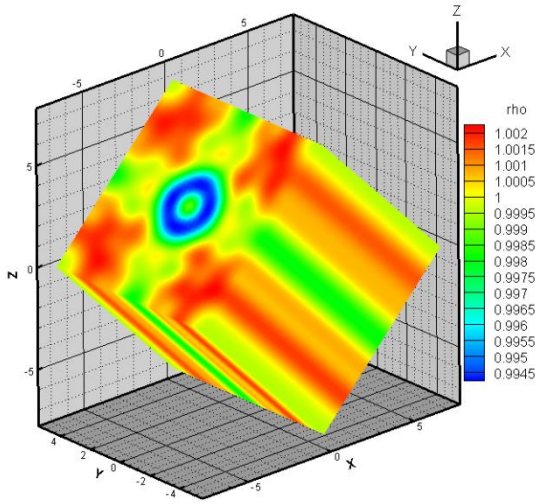


Figure 5.41 Density contours for three dimensional MHD vortex

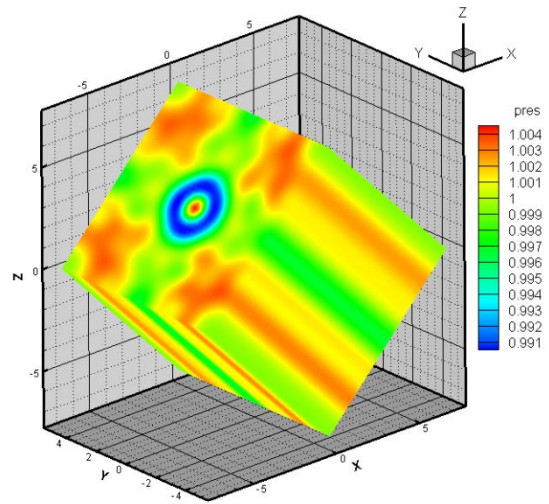


Figure 5.42 Pressure contours for three dimensional MHD vortex

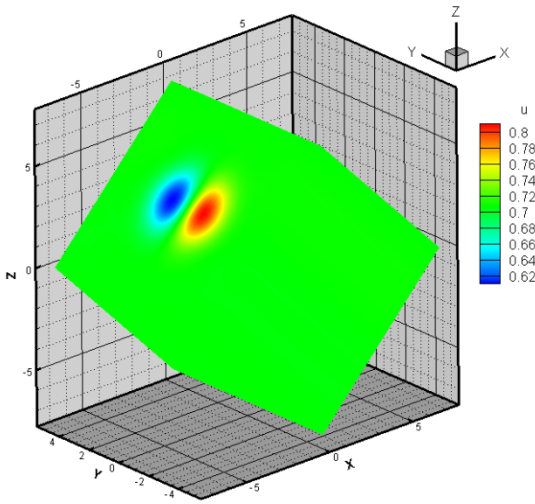


Figure 5.43 u contours for three dimensional MHD vortex

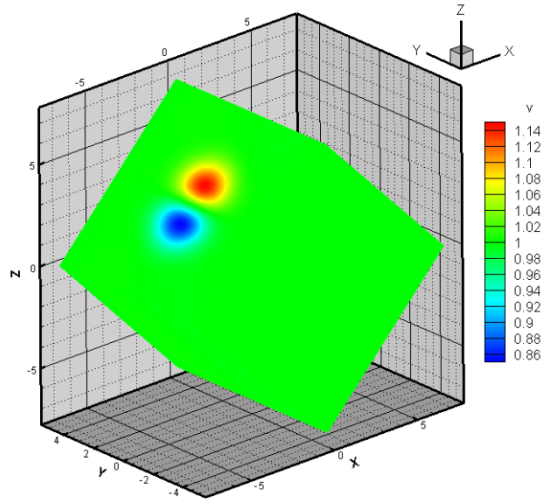


Figure 5.44 v contours for three dimensional MHD vortex

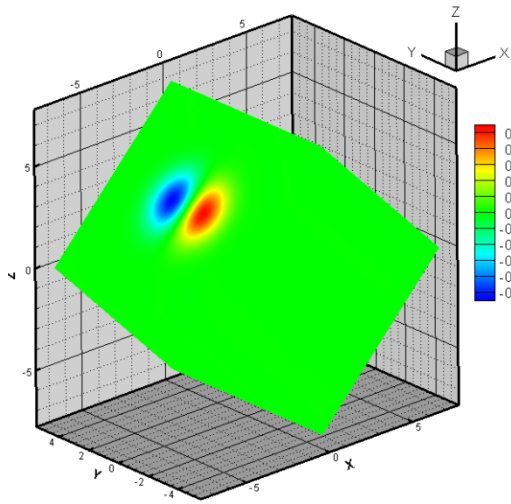


Figure 5.45 B_x of magnetic field contours for three dimensional MHD vortex

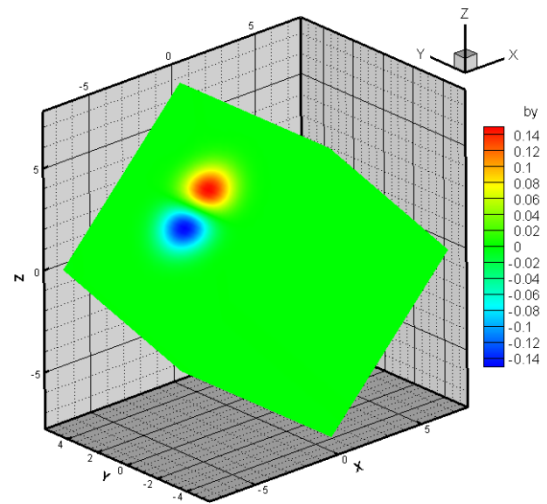


Figure 5.46 B_y of magnetic field contours for three dimensional MHD vortex

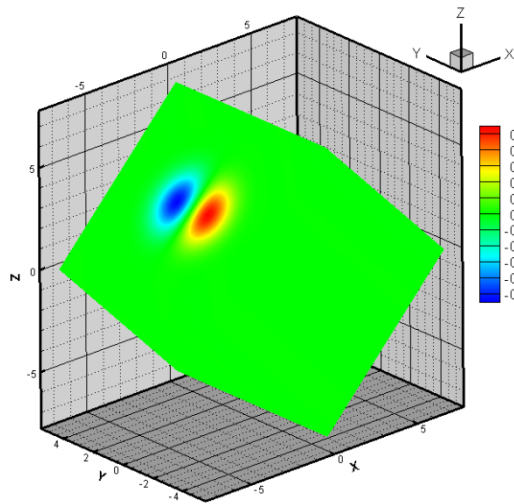


Figure 5.47 B_z of magnetic field contours for three dimensional MHD vortex

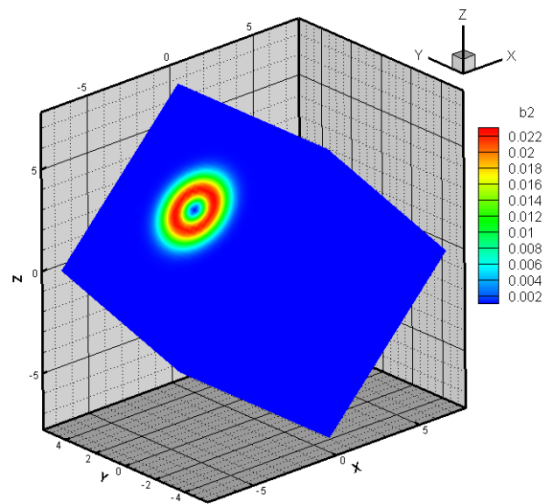


Figure 5.48 Magnetic pressure contours for three dimensional MHD vortex

Chapter 6 . Concluding Remarks

A reconstructed discontinuous Galerkin method based on a Hierarchical WENO reconstruction: HWENO (P_1P_2), originally introduced for solving the compressible Euler equations, has been extended to solve the ideal Magnetohydrodynamic (MHD) equations on arbitrary grids in this research. The HWENO (P_1P_2) method is designed not only to enhance the accuracy of the underlying discontinuous Galerkin method, but also to avoid non-physical oscillations in the vicinity of strong discontinuities. The Locally Divergence Free (LDF) technique is adopted to satisfy the divergence-free constraint in this work. The implementation of the LDF method is especially simple and straightforward in the context of our DG formulation, as the first derivatives of the solution are handily available due to the use of a Taylor basis in our DG formulation. The HLLD Riemann solver, introduced in the literature for one dimensional MHD problems, has been reformulated in the context of unstructured grid methods in the present work and extended to compute the flux functions at interfaces for three dimensional problems. A number of numerical experiments for a variety of flow conditions have been conducted to demonstrate the accuracy, robustness, and non-oscillatory performance of the HWENO (P_1P_2) method. The numerical results obtained indicate that the developed HWENO (P_1P_2) method is able to provide sharp resolution of shock waves without over- and under-shoots in the vicinity of strong discontinuities and achieve the designed third-order of accuracy for smooth flows: one order accuracy higher than the underlying DG method, thus significantly increasing the accuracy of the underlying DG method without significant increase in computing costs and memory requirements.

Three main contributions have been made in this research towards the development of a higher-order numerical method to simulate 3D MHD flows on hybrid grids: extension of the RDG methods for solving MHD equations, extension of the HLLD scheme for computing the numerical fluxes of the MHD equations at interfaces, and adoption and implementation of the LDF scheme to impose the divergence-free constraint. Although some attractive features of the RDG methods, including more efficient per degree of freedom than either the FV or DG methods and lower memory requirement for implicit schemes to achieve a given order of accuracy than the DG methods, have been demonstrated for solving the CFD problems, the research conducted in this work strongly indicates the potential of the RDG methods for solving the MHD problems. A number of Riemann solvers, including the Roe Riemann solver, HLL Riemann solver, HLLC Riemann solver, Lax-Friedrich scheme, and HLLD Riemann solver, have been developed in the literature to compute the numerical fluxes at interfaces for the MHD equations which are much more complex than the compressible Euler equations. The HLLD Riemann solver, originally developed for one-dimensional problems and mainly used on Cartesian and structured grids, has been successfully extended to compute the fluxes for the MHD equations on hybrid grids. The HLLD method is less dissipative and more accurate and robust than other Riemann solvers for MHD problems and enjoys the property of positivity preserving. Divergence free of magnetic field is a constraint that should be satisfied in numerical MHD solutions implicitly, which still represents a great challenge on unstructured or hybrid grids. A number of different numerical methods have been developed in the literature to satisfy the divergence free constraint. Locally Divergence

Free (LDF) technique have been adopted and successfully implemented in our RDG method. The use of the LDF strategy is especially attractive and advantageous in the context of our DG formulation, where the first order derivatives of independent variables are handily available due to the use of the Taylor basis. In our implementation, one of the spatial derivatives of the magnetic field is simply replaced by the sum of the other two derivatives at the end of each time step, leading to a simple and efficient approach to impose the divergence free constraint.

The current work only represents a first step towards developing an accurate, efficient, and robust computer code to simulate a variety of 3D MHD flows on arbitrary grids. The future work in the direction of the present research can be concentrated in two main categories: first the improvement and extension of the RDG method, and second the assessment and application of the developed RDG code to the MHD problems of practical and scientific interests.

An implicit temporal discretization method will need to be implemented in order to obtain steady solutions or effectively simulate the problems of low reduced frequency phenomena with disparate temporal and spatial scales. As demonstrated for CFD problems in the literature, the development of such an implicit method for solving MHD problems can tremendously benefit from the RDG formulation, where the memory requirement for the RDG methods is substantially lower than for the DG methods. The low storage requirement is especially important for solving the MHD equations, consisting of a nonlinear system of eight equations in 3D. Although the RDG formulation has the potential of delivering higher

accuracy with less computing cost for solving the MHD equations, the overall computational requirements are still very substantial for complex problems. As a consequence, practical computations for complex geometries require the use of parallel computing to produce results within an acceptable timescale. This is true both for production calculations and during the code development. The current RDG code needs to be transformed to run on massively parallel computing architectures so that it can become practically useful for realistic MHD flow applications.

The current RDG code must undergo extensive validation and verification process against benchmark test cases and test cases of interest to NASA before it can be trusted and used for simulating a variety of MHD flows. This is especially crucial for a MHD code, because of a lack of available experimental data in MHD. The performance of the developed RDG code needs to be assessed in terms of both accuracy and efficiency in comparison with the ADAPT3D code at NASA Goddard Space Flight Center for these test cases. A comparative study between the RDG and ADAPT3D codes will ultimately tell us if the higher-order RDG code can outperform the second order finite volume-based ADAPT3D code for these test cases relevant to the NASA mission.

References

- [1] E. F. Northrup, "Some Newly Observed Manifestations of Forces in the Interior of an Electrical Conductor," *Phys. Rev.* **24** (6), p.474, 1907.
- [2] A. F. Kolesnichenko, "Electromagnetic Processes in Liquid Material in the USSR and Eastern European Countries," *Iron and Steel Institute of Japan (ISIJ)* **30** (1) pp. 8–26, 1990.
- [3] P. Sporn and A. Kantrowitz, Magnetohydrodynamics: Future Power Process, *Power* 103 (11), pp. 62–65, Nov. 1959.
- [4] L. Steg and G. W. Sutton, "Prospects of MHD Power Generation," *Astronautics* 5, pp. 22–25, August 1960.
- [5] P. Graneau, "Electrodynamic Seawater Jet: An Alternative to the Propeller?," *IEEE Transactions of Magnetics* **25**(5) pp. 3275-3277, 1989.
- [6] A. Tsinober, "MHD Flow Drag Reduction," in *Viscous Drag Reduction in Boundary Layers*, American Institute of Astronautics and Aeronautics, 1990.
- [7] C. C. Baker, R. W. Conn, F. Najmabadi, and M. S. Tillack, "Status and Prospects for Fusion Energy from Magnetically Confined Plasmas," *Energy* **23** (7/8) pp. 649-694, 1998.
- [8] J. B. Gilbert and T. F. Lin, "Analyses of Underwater Magnetohydrodynamic Propulsion," *Proceedings of the 25th Intersociety Energy Conversion Engineering Conference, Reno Nevada, Vol. 5, IEEE 90CH2942-1*, pp.514-520, 1990

- [9] M. S. Tillack and N. B. Morley, Standard Handbook for Electrical Engineers, McGraw Hill 14th edition, 1998
- [10] D. L. Mitchell and D. U. Gubser, "Induction-Drive Magnetohydrodynamic Propulsion," Journal of Superconductivity, Vol. 6, No. 4, pp. 227-235, 1993.
- [11] W. Murgatroyd, "Experiments on magneto-hydrodynamic channel flow," Philosophical Magazine, Vol. 44, p. 1348, 1953.
- [12] C. Henoeh and J. Stace, "Experimental investigation of a salt water turbulent boundary layer modified by an applied streamwise magnetohydrodynamic body force," Physics of Fluids, Vol. 78, No. 6, pp. 1371-1383, 1995.
- [13] R. A. Alpher, et al., "Some Studies of Free Surface Mercury Magnetohydrodynamics," Review of Modern Physics, Vol. 32, No. 4, p. 758, 1960.
- [14] T. N. Aitov, et al., "Flow of Electrically Conducting Fluid in a Thin Layer with a Free Surface Under the Action of a Strong Magnetic Field," Magnetohydrodynamics, Vol. 21, No. 3, pp 71-76, 1985.
- [15] N. B. Morley and M. A. Abdou, "Study of Fully-Developed Liquid-Metal Open-Channel Flow in a Nearly Coplanar Magnetic Field," Fusion Technology, Vol. 31, pp. 135-153, 1997.
- [16] D. Beinstock, *et al.*, "Air Pollution Aspects of MHD Power Generation," 13th Symposium on Engineering Aspects of Magnetohydrodynamics, Stanford CA, 1973.

- [17] F. A. Hals and P. F. Lewis, "Control Techniques for Nitrogen Oxides in MHD Power Plants," 12th Symposium on Engineering Aspects of Magnetohydrodynamics, Argonne IL, 1972.
- [18] D. Y. Hsieh, "Stability of a Conducting Fluid Flowing Down an Inclined Plane in a Magnetic Field," *Physics of Fluids*, Vol. 8, No. 10, pp. 1785-1791, 1965.
- [19] U. Muller and L. Buhler, magnetofluidynamics in channel and containers, Springer-Verlag Berlin Heidelberg 2001.
- [20] J.P. Goedbloed, R. Keppens, S. Poedts, Advanced magnetohydrodynamics : with applications to laboratory and astrophysical plasmas, Cambridge University Press, 2010.
- [21] Reed, W.H. Reed and T.R. Hill, "Triangular Mesh Methods for the Neutron Transport Equation," Los Alamos Scientific Laboratory Report, LA-UR-73-479, 1973.
- [22] B. Cockburn, S. Hou, and C. W. Shu, "TVD Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for conservation laws IV: the Multidimensional Case," *Mathematics of Computation*, Vol. 55, pp. 545-581, 1990.
- [23] H. Luo, J. D. Baum, and R. Löhner, A Discontinuous Galerkin Method Using Taylor Basis for Compressible Flows on Arbitrary Grids, *Journal of Computational Physics*, DOI: 210.1016/j.jcp.2008.06.035, Vol. 227, No 20, pp. 8875-8893, October 2008.

- [24] H. Luo, J.D. Baum, and R. Löhner, On the Computation of Steady-State Compressible Flows Using a Discontinuous Galerkin Method, *International Journal for Numerical Methods in Engineering*, Vol. 73, No. 5, pp. 597-623, 2008.
- [25] H. Luo, J. D. Baum, and R. Löhner, A Hermite WENO-based Limiter for Discontinuous Galerkin Method on Unstructured Grids, *Journal of Computational Physics*, Vol. 225, No. 1, pp. 686-713, 2007.
- [26] H. Luo, J.D. Baum, and R. Löhner, A p-Multigrid Discontinuous Galerkin Method for the Euler Equations on Unstructured Grids, *Journal of Computational Physics*, Vol. 211, No. 2, pp. 767-783, 2006.
- [27] H. Luo, J.D. Baum, and R. Löhner, A Fast, p-Multigrid Discontinuous Galerkin Method for Compressible Flows at All Speeds, *AIAA Journal* , Vol. 46, No. 3, pp.635-652, 2008.
- [28] C. E. Baumann and J. T. Oden, A Discontinuous hp Finite Element Method for the Euler and Navier-Stokes Equations, *International Journal for Numerical Methods in Fluids*, Vol. 31, 1999.
- [29] H. Luo, L. Luo and K. Xu, A Discontinuous Galerkin Method Based on a BGK Scheme for the Navier-Stokes Equations on Arbitrary Grids, *Advances in Applied Mathematics and Mechanics*, Vol. 1, No. 3, pp. 301-318, 2009.
- [30] F. Bassi,; S. Rebay,; High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys.*, 138 (1997), 251–285.

- [31] B. Cockburn, C. W. Shu,: The Runge-Kutta discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems. *J. Comput. Phys.*, 141 (1998), 199–224.
- [32] C. Baumann, J. Oden: A discontinuous *hp* finite element method for the solution of the Euler equation of gas dynamics. In: 10th. International Conference on Finite Element in Fluids, 1998.
- [33] F. Bassi,; S. Rebay,; A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 131 (1997), 267–279.
- [34] I. Lomtev, G. Karniadakis,: Simulations of viscous supersonic flows on unstructured meshes. In: 35th. Aerospace Sciences Meeting, Reno, Nevada, 1997. AIAA-97-0754.
- [35] I. Lomtev, G. Karniadakis: A discontinuous Galerkin method for the Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 29 (1999), 587–603.
- [36] I. Lomtev, G. Karniadakis: Spectral *hp* methods for viscous compressible flows on unstructured 2D meshes. *J. Comput. Phys.*, 144 (1998), 325–357.
- [37] C. Baumann, J. Oden: A discontinuous *hp* finite element method for the Navier-Stokes equations. In: 10th. International Conference on Finite Element in Fluids, 1998.

- [38] B. Cockburn, G. Kanschat, D. Schotzau, C. Schwab: Local discontinuous Galerkin methods for the Stokes system. *SIAM J. Numer. Anal.*, 40 (2002) 1, 319–343.
- [39] B. Cockburn, G. Kanschat, D. Schotzau, C. Schwab: Local discontinuous Galerkin methods for the Oseen equations. *Math. Comp.* to appear.
- [40] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, M. Savini.: A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In: Decuipere, R.; Dibelus, G. (eds.): 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics. Technologisch Instituut, Antwerpen, Belgium, March 5–7 1997, pp. 99–108.
- [41] T. Warburton, G. Karniadakis: A discontinuous Galerkin method for the viscous MHD equations. *J. Comput. Phys.*, 152 (1999), 1–34.
- [42] P. A. Gremaud, J. Mattheus: Simulation of gravity flow of granular materials in silos. In: B. Cockburn, G. Karnadakis.;C. W. SHU (eds.): *Discontinuous Galerkin Methods. Theory, Computation and Applications. Lecture Notes in Computational Science and Engineering*, 11. Springer Verlag, February 2000, pp. 125–134.
- [43] P. Gremaud, J. Mattheus: On the computation of steady hopper flows i. stress determination for coulomb materials. *J. Computational Phys.*, 166 (2001), 63–83.
- [44] Z. Chen; B. Cockburn; J. Jerome; C. W. Shu.: Mixed-RKDG finite element methods for the 2-D hydrodynamic model for semiconductor device simulation. *VLSI Design*, 3 (1995), 145–158.

- [45] Z. Chen; B. Cockburn, C. Gardner; J. Jerome: Quantum hydrodynamic simulation of hysteresis in the resonant tunneling diode. *J. Computational. Phys.*, 117 (1995), 274–280.
- [46] F. Carranza; B Fang; R. Haber.: An adaptive discontinuous Galerkin model for coupled viscoplastic crack growth and chemical transport. In: B. Cockburn; G. Karnadakis; C. W. Shu. (eds.): *Discontinuous Galerkin Methods. Theory, Computation and Applications. Lecture Notes in Computational Science and Engineering*, 11. Springer Verlag, February 2000, pp. 277–283.
- [47] M; Fortin, A.: New approach for the finite element method simulation of viscoelastic flows. *J. Non-Newt. Fluid Mech.*, 32 (1989), 295–310.
- [48] F. Baaijens.: Application of low-order discontinuous Galerkin methods to the analysis of viscoelastic flows. *J. Non-Newt. Fluid Mech.*, 52 (1994), 37–57.
- [49] A. Bahaar, J. Barannger; D. Sandri: Galerkin discontinuous approximation of the transport equation and viscoelastic fluid flow on quadrilaterals. *Numerical Methods Partial Differential Equations*, 14 (1998), 97–114.
- [50] C. Dawson.; V. Aizinger.; B. Cockburn.: Local discontinuous Galerkin methods for problems in contaminant transport. In: B. Cockburn, G. Karnadakis, C. W. SHU, (eds.): *Discontinuous Galerkin Methods. Theory, Computation and Applications. Lecture Notes in Computational Science and Engineering*, 11. Springer Verlag, February 2000, pp. 309–314.

- [51] V. Aizinger, C. Dawson, B. Cockburn; P. Castillo.: Local discontinuous Galerkin method for contaminant transport. *Advances in Water Resources*, 24 (2000), 73–87.
- [52] B. Cockburn, C. Dawson: Some extensions of the local discontinuous Galerkin method for convection-diffusion equations in multidimensions. In: WHITEMAN, J. (ed.): *The Proceedings of the Conference on the Mathematics of Finite Elements and Applications: MAFELAP X*. Elsevier, 2000, pp. 225–238.
- [53] B. Cockburn, C. Dawson: Approximation of the velocity by coupling discontinuous Galerkin and mixed finite element methods for flow problems. *Computational Geosciences (Special issue on locally conservative numerical methods for flow in porous media)*, 6(2002), 502–522.
- [54] C. Hu; C. W. Shu: A discontinuous Galerkin finite element method for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21 (1999), 666–690.
- [55] C. Hu; O. Lepsky, C. W. Shu: The effect of the least square procedure for discontinuous Galerkin methods for Hamilton-Jacobi equations. In: B. Cockburn, G. Karnadakis, C. W. Shu. (eds.): *Discontinuous Galerkin Methods. Theory, Computation and Applications. Lecture Notes in Computational Science and Engineering*, 11. Springer Verlag, February 2000, pp. 343–348.
- [56] O. Lepsky, C. Hu, C. W. Shu.: Analysis of the discontinuous Galerkin method for Hamilton-Jacobi equations. *Appl. Numer. Math.*, 33 (2000), 423–434.

- [57] B. Cockburn; C. W. Shu: The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.*, 35 (1998), 2440–2463.
- [58] J. Oden, I. Babuska, C. Baumann: A discontinuous *hp* finite element method for diffusion problems. *J. Computational Physics.*, 146 (1998), 491–519.
- [59] I. Babuska, C. Baumann, J. Oden: A discontinuous *hp* finite element method for diffusion problems: 1-D analysis. *Comput. Math. Appl.*, 37 (1999), 103–122.
- [60] B. Riveterere, M. Wheeler, V Girault: Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems. Part I. *Comp. Geo.*, (1999) 3, 337–360.
- [61] F. Brezzi, G. Manzini, D. Marini, P. Pietra, A. Russo: Discontinuous Galerkin approximations for elliptic problems. *Numerical Methods for Partial Differential Equations*, 16 (2000), 365–378.
- [62] P. Castillo, B. Cockburn, I. Perugia, D. Schotzao: An a priori error analysis of the local discontinuous Galerkin method for elliptic problems. *SIAM J. Numer. Anal.*, 38 (2000), 1676–1706.
- [63] B. Cockburn, G. Kanschat, I. Perugia, D. Schotzao: Superconvergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids. *SIAM J. Numer. Anal.*, 39 (2001), 264–285.

- [64] D. Arnold, F. Brezzi, B. Cockburn, D. Marini: Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39 (2001), 1749–1779.
- [65] P. Hansbo, M. Larson: Discontinuous finite element methods for incompressible and nearly incompressible elasticity by use of Nitsche’s method. *Computational. Methods Applied. Mechanic. Engineering*, 191 (2002), 1895–1908.
- [66] G. Engel, K. Garikipati, T. Hughes, M. Larson, L. Mazzei, R. Taylor: Continuous/discontinuous finite element approximations of fourth-order elliptic problems in structural and continuum mechanics with applications to thin beams and plates, and strain gradient elasticity. *Computational. Methods Applied. Mechanic. Engineering*, 191 (2002), 3669–3750.
- [67] C. W. Shu, J. Yan: Local discontinuous Galerkin for partial differential equations with higher order derivatives. *J. Sci. Comput.*, 17 (2002), 27–47.
- [68] C. W. Shu, J. Yan: A local discontinuous Galerkin method for KdV type equations. *SIAM J. Numer. Anal.* to appear. COCKBURN, B.: Discontinuous Galerkin Methods 25
- [69] B. Cockburn: Discontinuous Galerkin methods for convection-dominated problems. In: T. Barth, H. Deconink (eds.): *High-Order Methods for Computational Physics. Lecture Notes in Computational Science and Engineering*, 9. Springer Verlag, 1999, pp. 69–224.

- [70] B. Cockburn, G. Karnadakis, C. W. Shu: The development of discontinuous Galerkin methods. In: B. Cockburn, G. Karnadakis, C. W. Shu (eds.): Discontinuous Galerkin Methods. Theory, Computation and Applications. Lecture Notes in Computational Science and Engineering, 11. Springer Verlag, February 2000, pp. 3–50.
- [71] B. Cockburn, C. W. Shu: Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16 (2001), 173–261.
- [72] M. Dumbser, D.S. Balsara, E.F. Toro, 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*, 227:8209-8253, 2008.
- [73] M. Dumbser, O. Zanotti. Very high order PNPM schemes on unstructured meshes for the resistive relativistic MHD equations. *Journal of Computational Physics*, 228:6991-7006, 2009.
- [74] M. Dumbser. Arbitrary High Order PNPM Schemes on Unstructured Meshes for the Compressible Navier-Stokes Equations. *Computers & Fluids*, 39: 60-76. 2010.
- [75] H. Luo, L. Luo, and R. Nourgaliev, A Reconstructed Discontinuous Galerkin Method for the Euler Equations on Arbitrary Grids, doi: 10.4028/cicp.250911.030212a, in press, *Communication in Computational Physics*, 2012.
- [76] H. Luo, L. Luo, R. Nourgaliev, V.A. Mousseau, and N. Dinh, A Reconstructed Discontinuous Galerkin Method for the Compressible Navier-Stokes

- Equations on Arbitrary Grids, *Journal of Computational Physics*, Vol. 229, pp. 6961-6978, 2010.
- [77] H. Luo, L. Luo, A. Ali, R. Nourgaliev, and C. Cai, A Parallel, Reconstructed Discontinuous Galerkin Method for the Compressible Flows on Arbitrary Grids, *Communication in Computational Physics*, Vol. 9, No. 2, pp. 363-389, 2011.
- [78] L.P. Zhang, W. Liu, L.X. He, X.G. Deng, and H.X. Zhang, A Class of Hybrid DG/FV Methods for Conservation Laws I: Basic Formulation and One-Dimensional System, *Journal of Computational Physics*, Vol. 23, No. 4, pp. 1081-1103, 2012.
- [79] L.P. Zhang, W. Liu, L.X. He, X.G. Deng, and H.X. Zhang, A Class of Hybrid DG/FV Methods for Conservation Laws II: Two dimensional Cases, *Journal of Computational Physics*, Vol. 23, No. 4, pp. 1104-1120, 2012.
- [80] H. Luo, Y. Xia, R. Nourgaliev, and C. Cai, A Class of Reconstructed discontinuous Galerkin Methods for the Compressible Flows on Arbitrary Grids, AIAA-2011-0199, 2011.
- [81] H. Luo, H. Xiao, R. Nourgaliev, and C. Cai, A Comparative Study of Different Reconstruction Schemes for Reconstructed Discontinuous Galerkin Methods for the Compressible Flows on Arbitrary Grids, AIAA-2011-3839, 2011.
- [82] D. F. Haider, J.P. Croisille, and B. Courbet, Stability Analysis of the Cell Centered Finite-Volume MUSCL Method on Unstructured Grids, *Numirische Mathematik*, Vol. 113, No. 4 pp. 555-600, 2009.

- [83] D. Balsara, C. Altmann, C.D. Munz and M. Dumbser, A sub-cell based indicator for troubled zones in RKDG schemes and a novel class of hybrid RKDG + HWENO schemes, *Journal of Computational Physics*, Vol. 226, pp. 586–620, 2007.
- [84] Z. Xu, Y Liu, H Du, and CW Shu, Point-wise hierarchical Reconstruction for Discontinuous Galerkin and Finite Volume Method for Solving Conservation Laws, *Journal of Computational Physics*, Vol. 230, pp. 6843-6865, 2011.
- [85] 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*, Vol. 231, pp. 5489-5503, 2012.
- [86] H. Luo, Y. Xia, R. Nourgaliev, A Hierarchical Hermit WENO Reconstruction-Based Discontinuous Galerkin Method for the Compressible Flows on Tetrahedral Grids, AIAA-2012-2838, 42nd AIAA Fluid Dynamics Conference and Exhibit, 25-28 June 2012, New Orleans, LA.
- [87] M. Brio and C. C. Wu, An upwinding differencing scheme for the equations of ideal magne-tohydrodynamic. *J. Comput. Phys.*, 75(1988), 400.
- [88] D. Ryu and T. W. Jones, Numerical magnetohydrodynamics in astrophysics: algorithm and tests for one-dimensional flow., *The astrophy. Journal*, 442 (1995), 228-258.
- [89] P. L. Roe and D. S. Balsara, Notes on the eigensystem of magnetohydrodynamics, *SIAM J.Appl. Math.*, 56 (1996) 57-67.

- [90] B. Cockburn F. Li C. W. Shu, Locally divergence-free discontinuous Galerkin methods for the Maxwell equations, *J. Comput. Physics* Vol. 194, Issue 2, 588–610
- [91] F. Li and C. W. Shu, Locally Divergence-Free Discontinuous Galerkin Methods for MHD Equations, *J. Scientific Computing*, Vol. 22 and 23, June 2005
- [92] K. G. Powell, P. L. Roe, T. J. Linde, T. I. Gombosi, and D. L. DeZeeuw, A solution-adaptive upwinding scheme for ideal magnetohydrodynamics, *J. Comt. Phys.*, 154 (1999), 284.
- [93] T. Miyoshi and K. Kusano, A multi-state HLL approximate Riemann solver for ideal magnetohydrodynamics, *J. Compu. Phys.* 208 (2005) 315–344
- [94] P. Batten, N. Clarke, C. Lambert, D.M. Causon, On the choice of wave speeds for the HLLC Riemann solver, *SIAM J. Sci.Statist. Comput.* 18 (1997) 1553.
- [95] P. Batten, N. Clarke, C. Lambert, D.M. Causon, On the choice of wave speeds for the HLLC Riemann solver, *SIAM J. Sci. Statist. Comput.* 18 (1997) 1553.
- [96] K.F. Gurski, An HLLC-type approximate Riemann solver for ideal magnetohydrodynamics, *SIAM J. Sci. Comput.* 25 (2004) 2165.
- [97] G. Toth, The $\nabla \cdot B = 0$ constraint in shock capturing magnetohydrodynamics codes, *J Comput. Physics* 161 (2000), 605-652
- [98] J. U. Brackbill and D. C. Barnes, the effect of nonzero $\nabla \cdot B$ on the numerical solution of manetohydrodynamic equations, *J. Comput. Physics* 35 (1980) 426-430
- [99] J. N. REDDY, D. K. GARTLING, *The Finite Element Method in Heat Transfer and Fluid Dynamics*, CRC Press, 1994.

- [100] Zhiliang Xu, Yingjie Liu, Chi-Wang Shu, Hierarchical reconstruction for discontinuous Galerkin methods on unstructured grids with a WENO-type linear reconstruction and partial neighboring cells, *Journal of Comp. Phys.* 228 (2009) 2194–2212
- [101] Zhiliang Xu, Yingjie Liu, Huijing Dua, Guang Lin, Chi-Wang Shu, Point-wise hierarchical reconstruction for discontinuous Galerkin and finite volume methods for solving conservation laws, *Journal of Comp. Phys.* 230 (2011) 6843–6865
- [102] Hong Luo, Joseph D. Baum, Rainald Lo`hner, A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids *Journal of Comp. Phys.* 225 (2007) 686–713
- [103] H. Luo, J.D. Baum, R. Lo`hner, On the computation of steady-state compressible flows using a discontinuous Galerkin method, in: *Proceedings of the Fourth International Conference on Computational Fluid Dynamics, Ghent, Belgium, 10–14, July, 2006.*
- [104] A. Harden, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high-order accurate essential non-oscillatory schemes III, *Journal of Computational Physics* 71 (1987) 231–303.
- [105] X. Liu, S. Osher, T.F. Chen, Weighted essential non-oscillatory schemes, *Journal of Computational Physics* 115 (1994) 200–212.

- [106] J. Qiu, C.W. Shu, Runge–Kutta discontinuous Galerkin method using WENO limiters, *SIAM Journal of Scientific Computing* 26 (2005) 907–929.
- [107] J. Qiu, C.W. Shu, Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one dimensional case, *Journal of Computational Physics* 193 (2004) 115–135.
- [108] J. Qiu, C.W. Shu, Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method II: two dimensional case, *Computers & Fluids* 34 (2005) 642–663.
- [109] A. Harden, S.R. Chakravarthy, Multidimensional ENO schemes on general geometries, ICASE Report No. 91-76, 1991.
- [110] R. Abgrall, On essential non-oscillatory schemes on unstructured meshes, *Journal of Computational Physics* 114 (1994) 45–58.
- [111] T. Sonar, On the construction of essential non-oscillatory finite volume approximation to hyperbolic conservation laws on general triangulations: polynomial recovery, accuracy, and stencil selection, *Computer Methods in Applied Mechanics and Engineering* 140 (1997) 157–182.
- [112] O. Friedrich, Weighted essential non-oscillatory schemes for the interpolation of mean values on unstructured grids, *Journal of Computational Physics* 144 (1998) 194–212.
- [113] C. Hu, C.W. Shu, Weighted essential non-oscillatory schemes on unstructured triangular meshes, *Journal of Computational Physics* 150 (1999) 97–127.

- [114] T. Sonar, On the construction of essential non-oscillatory finite volume approximation to hyperbolic conservation laws on general triangulations: polynomial recovery, accuracy, and stencil selection, *Computer Methods in Applied Mechanics and Engineering* 140 (1997) 157–182.
- [115] O. Friedrich, Weighted essential non-oscillatory schemes for the interpolation of mean values on unstructured grids, *Journal of Computational Physics* 144 (1998) 194–212.
- [116] G.S. Jiang, C.W. Shu, Efficient implementation of weighted ENO schemes, *Journal of Computational Physics* 126 (1) (1996) 202–228.
- [117] T. Miyoshi, K. Kusano, “A Multi-state HLL Approximate Riemann Solver for Ideal Magnetohydrodynamics,” *Journal of Computational Physics*, Vol 208, 315-344, 2005.
- [118] K. G. Powell, P. L. Roe, R. S. Moyle, T. Gombosi, and D. De Zeeuw, An upwind scheme for magnetohydrodynamics, in K. W. Morton, and M. J. Bines, editors, *Numerical Methods for fluid dynamics*, V pages 163-180, 1995
- [119] B. Udea, An advanced implicit solver for MHD, PhD thesis, University of Washington, 1999.

Appendices

Appendix A: Derivation of Governing Equations

At this appendix we present the conservation equations for MHD. Since Maxwell equations, which are fundamental equations of electrodynamics, used in derivation of equations, they are introduced first. The *Maxwell equations* describe how electric and magnetic field generated, interact with each other are defined as a set of PDE's as flow [119]:

$$\nabla \cdot E = 0 \quad (\text{A. 1})$$

$$\nabla \cdot B = 0 \quad (\text{A. 2})$$

$$\nabla \times E = -\frac{\partial B}{\partial t} \quad (\text{A. 3})$$

$$\nabla \times B = \mu_0 J \quad (\text{A. 4})$$

For *conservation of mass* we have continuity equation as follow:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho V) = 0 \quad (\text{A. 5})$$

Where the ρ is density and V is the velocity vector. The first term here is time variation of density in control volume and the second term is convection of mass flux.

Conservation of momentum we have

$$\frac{\partial \rho V}{\partial t} + \nabla \cdot (\rho V V) + \nabla \cdot \vec{p} = J \times B \quad (\text{A. 6})$$

In which J is electrical current and B is magnetic field. The pressure tensor can be written as its diagonal and stress tensor. So the pressure tensor is

$$\vec{p} = \vec{\tau} + \vec{I}p \quad (\text{A. 7})$$

Where p is the scalar pressure and τ is tension tensor. Replacing pressure tensor in conservation of momentum equation

$$\frac{\partial \rho V}{\partial t} + \nabla \cdot (\rho V V) + \nabla \cdot \vec{\tau} + \nabla \cdot \vec{I} p = J \times B \quad (\text{A. 8})$$

Here we observe that momentum equation for MHD is the same as fluid dynamic momentum plus $J \times B$. This term is body force added to the momentum equation because of electrical conductivity of fluid and presence of magnetic field. We can express this body force in terms of magnetic field using Maxwell equation (A. 4)

$$J = \frac{1}{\mu_0} \nabla \times B$$

After some Vector algebra and manipulation we will have

$$J \times B = \frac{1}{\mu_0} (\nabla \times B) \times B = \frac{1}{\mu_0} \left[\nabla \cdot (BB) - \frac{1}{2} B^2 - (\nabla \cdot B) B \right] \quad (\text{A. 9})$$

For further simplification we substitute Maxwell equation (A. 2) in equation (A. 9).

$$J \times B = \frac{1}{\mu_0} \left[\nabla \cdot (BB) - \frac{1}{2} B^2 \right] \quad (\text{A. 10})$$

So the momentum equation of MHD is as follow:

$$\frac{\partial \rho V}{\partial t} + \nabla \cdot (\rho V V) = -\nabla \cdot \vec{I} p - \nabla \cdot \vec{\tau} + \frac{1}{\mu_0} \left[\nabla \cdot (BB) - \frac{1}{2} B^2 \right] \quad (\text{A. 11})$$

Here we see that $\nabla \cdot (BB)$ acts like force and B^2 acts like pressure at field. Here we use the shear stress tensor for definition of stress tensor.

$$\tau_{ij} = \mu \left[-\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + \frac{2}{3} \delta_{ij} \nabla \cdot V \right] \quad (\text{A. 12})$$

Here μ is viscosity which is property of fluid and is function of temperature, i, j are indices correspond to spatial coordinates ($v_1=u, v_2=v, v_3=w$).

The relationship between the total conduction current $J=n_e e(Vol_i-Vol_e)$ is known as generalized ohm's law. To obtain Ohm's law, the electron conservation of momentum is obtained by multiplication of the ion's electric charge and subtraction from the ion momentum equation which is multiplied by ions charge. The first approximation made in **Ohm's law** is that electron's inertia is neglected. The resulting equation is

$$\frac{m_e}{n_e^2} \frac{\partial J}{\partial t} = E + V \times B - \frac{1}{n_e} J \times B + \frac{1}{n_e} \nabla \cdot \overrightarrow{p_e} - \vec{\eta} \cdot J \quad (\text{A. 13})$$

Which describe how electric current varies in time. We approximate that characteristic frequencies is much lower than the electron frequency allows us to neglect left hand side. The electron pressure tensor is expressed in terms of electron tension tensor and the electron scalar pressure as $\overrightarrow{p_e} = \overrightarrow{\tau_e} + \vec{I} p_e$. Using some simplification

$$E + V \times B = \frac{1}{n_e} J \times B - \frac{1}{n_e} \nabla \cdot (\vec{I} p_e) - \frac{1}{n_e} \nabla \cdot \overrightarrow{\tau_e} + \vec{\eta} \cdot J \quad (\text{A. 14})$$

This equation is known as **generalized Ohm's Law**. The first term in right hand side is contribution of Hall's current in the Ohm's law. The second term is electron diamagnetic drift effect. The third term is divergence of electron stress tensor and normally is neglected due to its low order of magnitude relative to Hall current and diamagnetic effect. The fourth term represents the contribution of momentum transfer in electron-ion collisions and $\vec{\eta}$ is plasma electric resistivity tensor. With this assumption

$$E + V \times B = \frac{1}{n_e} J \times B - \frac{1}{n_e} \nabla \cdot (\vec{I} p_e) + \vec{\eta} \cdot J \quad (\text{A. 15})$$

Since we use **Ideal MHD** equation

$$E + V \times B = 0 \quad (\text{A. 16})$$

From equation (A. 3) we drive the variation of magnetic field with time. Substituting equation (A. 16) in equation (A. 3) the induction equation becomes

$$\frac{\partial B}{\partial t} + \frac{1}{\mu_0} \nabla \cdot (vB^T - Bv^T) = 0 \quad (\text{A. 17})$$

The energy equation is obtained as

$$\frac{\partial e}{\partial t} + \nabla \cdot \left[\left(e + p + \frac{1}{2\mu_0} B^2 \right) V - B \cdot V \frac{B}{\mu_0} \right] = V \cdot \left\{ V \cdot \vec{\tau} + \vec{k} \cdot \nabla T - \frac{1}{\mu_0^2} [\vec{\eta} \cdot (\nabla \times B)] \times B \right\} \quad (\text{A. 18})$$

Where \vec{k} is thermal, conductivity tensor, T is temperature.

Appendix B: Eigenvalues of Ideal MHD Flux Jacobian

The conservative variable of ideal MHD written in vector form is

$$q = [\rho, \rho u, \rho v, \rho w, B_x, B_y, B_z, \rho e]^T \quad (\text{B.1})$$

And the primitive variables are

$$W = [\rho, u, v, w, B_x, B_y, B_z, p]^T \quad (\text{B.2})$$

Starting with one dimensional conservation law, the formulation for transformation from conservative to primitive flux Jacobian is found

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} = \frac{\partial q}{\partial t} + A_c \frac{\partial q}{\partial x} = 0 \quad (\text{B.3})$$

To write conservative equation in terms of primitive equations we apply chain rule over equation (B.3)

$$\frac{\partial q}{\partial W} \frac{\partial W}{\partial t} + A_c \frac{\partial q}{\partial W} \frac{\partial W}{\partial x} = 0 \quad (\text{B.4})$$

Where $A_c = \frac{\partial f}{\partial q}$ is flux Jacobian derived in terms of conservative variables. By Multiplying equation

(B.4) with $\left(\frac{\partial q}{\partial W}\right)^{-1}$ we will have,

$$\frac{\partial W}{\partial t} + \left(\frac{\partial q}{\partial W}\right)^{-1} A_c \frac{\partial q}{\partial W} \frac{\partial W}{\partial x} = \frac{\partial W}{\partial t} + A_p \frac{\partial W}{\partial x} = 0 \quad (\text{B.5})$$

Where $A_p = \frac{\partial f}{\partial W}$ is flux Jacobian derived in terms of primitive variables.

So the transformation from primitive to conservative flux Jacobian is

$$A_p = \frac{\partial W}{\partial q} A_c \frac{\partial q}{\partial W} \quad (\text{B.6})$$

The transformation matrices are

$$\frac{\partial W}{\partial q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 & 0 & 0 & 0 \\ v & 0 & \rho & 0 & 0 & 0 & 0 & 0 \\ w & 0 & 0 & \rho & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{V^2}{2} & \rho u & \rho v & \rho w & B_x & B_y & B_z & \frac{1}{\gamma'} \end{bmatrix} \quad (\text{B.7})$$

And

$$\frac{\partial q}{\partial W} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 \\ \frac{-w}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{\gamma V^2}{2} & \gamma' u & \gamma' v & \gamma' w & \gamma' B_x & \gamma' B_y & \gamma' B_z & \gamma' \end{bmatrix} \quad (\text{B.8})$$

Where $V^2 = u^2 + v^2 + w^2$ and $\gamma' = \gamma - 1$.

The left and right eigenvector of the flux Jacobian can also be transformed from primitive to conservative form and vice-versa. The transformation formula for eigenvector matrices are obtained starting from the eigenvector decomposition of the primitive flux Jacobian

$$A_p = R_p \Lambda L_p = \frac{\partial W}{\partial q} A_c \frac{\partial q}{\partial W} = \frac{\partial W}{\partial q} R_c \Lambda L_c \frac{\partial q}{\partial W} \quad (\text{B.9})$$

Where Λ is diagonal matrix having eigenvalues of the flux Jacobian as elements. Identifying terms in equation (A. 9)(B.9) will give us transformation formulas

$$\begin{aligned}
R_p &= \frac{\partial W}{\partial q} R_c & R_c &= \frac{\partial q}{\partial W} R_p \\
L_p &= L_c \frac{\partial q}{\partial W} & L_c &= L_p \frac{\partial W}{\partial q}
\end{aligned}
\tag{B.10}$$

For example here we explain the flux Jacobian in x direction and its corresponding eigenvalue and eigenvectors. The eigenvectors are not normalized and first published by Powell [118].

The flux Jacobian in direction normal to cell face calculated in terms of primitive variables in equation (B.2) and its

$$A_p = \begin{bmatrix}
u & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & u & 0 & 0 & 0 & \frac{B_y}{\rho} & \frac{B_z}{\rho} & \frac{1}{\rho} \\
0 & 0 & u & 0 & 0 & \frac{-B_x}{\rho} & 0 & 0 \\
0 & 0 & 0 & u & 0 & 0 & \frac{-B_x}{\rho} & 0 \\
0 & 0 & 0 & 0 & u & 0 & 0 & 0 \\
0 & B_y & -B_x & 0 & 0 & u & 0 & 0 \\
0 & B_z & 0 & -B_x & 0 & 0 & u & 0 \\
0 & \gamma p & 0 & 0 & 0 & 0 & 0 & u
\end{bmatrix}
\tag{B.11}$$

The expression of flux Jacobian derived using primitive variables is simpler than that of conservative variables. The set of eigenvalues and eigenvectors of flux Jacobian in equation (B.11) are:

Entropy wave $\lambda_e = v_x$

$$r_e = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T
\tag{B.12}$$

$$l_e = \left[1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -\frac{1}{c^2} \right]^T
\tag{B.13}$$

Alfven waves $\lambda_a = v_x \pm c_a$

$$r_A^\pm = [0 \quad 0 \quad -B_z \quad B_y \quad 0 \quad \pm B_z \sqrt{\rho} \quad \mp B_y \sqrt{\rho}]^T \quad (\text{B.14})$$

$$l_A^\pm = \left[0 \quad 0 \quad -B_z \quad B_y \quad 0 \quad \frac{\pm B_z}{\sqrt{\rho}} \quad \frac{\mp B_y}{\sqrt{\rho}} \right]^T \quad (\text{B.15})$$

Fast magneto-acoustic waves: $\lambda_f = v_x \pm c_f$

$$r_f^\pm = \left[\rho \pm c_f, \quad \mp \frac{B_x B_y c_f}{\rho c_f^2 - B_x^2}, \quad \mp \frac{B_x B_z c_f}{\rho c_f^2 - B_x^2}, \quad 0, \quad \frac{B_y \rho c_f^2}{\rho c_f^2 - B_x^2}, \quad \frac{B_z \rho c_f^2}{\rho c_f^2 - B_x^2}, \quad \gamma p \right]^T \quad (\text{B.16})$$

$$l_f^\pm = \left[\pm \rho c_f, \quad \mp \frac{B_x B_y \rho c_f}{\rho c_f^2 - B_x^2}, \quad \mp \frac{B_x B_z \rho c_f}{\rho c_f^2 - B_x^2}, \quad 0, \quad \frac{B_y \rho c_f^2}{\rho c_f^2 - B_x^2}, \quad \frac{B_z \rho c_f^2}{\rho c_f^2 - B_x^2}, \quad 1 \right]^T \quad (\text{B.17})$$

Slow magneto-acoustic waves: $\lambda_s = v_x \pm c_s$

$$r_s^\pm = \left[\rho \pm c_s, \quad \mp \frac{B_x B_y c_s}{\rho c_s^2 - B_x^2}, \quad \mp \frac{B_x B_z c_s}{\rho c_s^2 - B_x^2}, \quad 0, \quad \frac{B_y \rho c_s^2}{\rho c_s^2 - B_x^2}, \quad \frac{B_z \rho c_s^2}{\rho c_s^2 - B_x^2}, \quad \gamma p \right]^T \quad (\text{B.18})$$

$$l_s^\pm = \left[\pm \rho c_s, \quad \mp \frac{B_x B_y \rho c_s}{\rho c_s^2 - B_x^2}, \quad \mp \frac{B_x B_z \rho c_s}{\rho c_s^2 - B_x^2}, \quad 0, \quad \frac{B_y \rho c_s^2}{\rho c_s^2 - B_x^2}, \quad \frac{B_z \rho c_s^2}{\rho c_s^2 - B_x^2}, \quad 1 \right]^T \quad (\text{B.19})$$

Magnetic flux wave $\lambda_d = v_n$

$$r_f = [0, \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^T \quad (\text{B.20})$$

$$l_f = [0, \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^T \quad (\text{B.21})$$

In which c_a is Alfvén speed and $c_{f,s}$ are fast and slow magneto acoustic waves respectively.

These speeds are defined as follow:

$$c_a = \frac{B_x}{\sqrt{\rho}} \quad (\text{B.22})$$

$$c_{f,s}^2 = \frac{1}{2} \left[\frac{\gamma p + B^2}{\rho} \pm \sqrt{\left(\frac{\gamma p + B^2}{\rho} \right)^2 - 4 \frac{\gamma p B_x^2}{\rho^2}} \right] \quad (\text{B.23})$$

Where the fast magneto-acoustic wave corresponds to the plus sign in front of square root and slow magneto-acoustic wave correspond to minus in front of square root. Also

$$B^2 = B_x^2 + B_y^2 + B_z^2$$