Three dimensional high current arc simulations for circuit breakers using real gas resistive magnetohydrodynamics

Author(s):
Kumar, Harish

Publication Date:
2009

Permanent Link:
https://doi.org/10.3929/ethz-a-005927654

Rights / License:
In Copyright - Non-Commercial Use Permitted
Three Dimensional High Current Arc Simulations for Circuit Breakers Using Real Gas Resistive Magnetohydrodynamics

A dissertation submitted to
ETH Zürich

for the degree of
Doctor of Sciences

presented by
HARISH KUMAR

MS Math. Sci. IISc. Bangalore
born October 30, 1980
citizen of India

accepted on the recommendation of
Prof. Dr. Rolf Jeltsch, ETH Zürich, examiner
Prof. Dr. Manuel Torrilhon, ETH Zürich, co-examiner

2009
Acknowledgment

This work has been carried out at the Seminar for Applied Mathematics (SAM), D-MATH at the ETH Zürich under the supervision of Prof. Rolf Jeltsch. This work was carried out under a joint project with ABB research center in Dätwil, and was also supported by KTI Switzerland.

I would like first to thank my thesis supervisor Prof. Rolf Jeltsch for giving me the opportunity to work with him and for his constant support and motivation. I thanks Prof. Christoph Schwab for his several important suggestions and advices that contributed to this project immensely. I would also like to thank Prof. Ralf Hiptmair and Prof. Manuel Torrilhon for many fruitful discussion. I would also like to thank Prof. W. Peterson for his helpful suggestions. I would like specially to thank Prof. Siddhartha Mishra for being a true friend and also for proof reading this thesis. Let me also thank people from ABB research center for valuable suggestions during the project.

I thanks all my colleagues(former and current) at SAM for making my studies at SAM comfortable and enjoyable. In particular I would like to thank Kersten, Paolo, Bastian, Cada, Imran, Roman, Holger, Sorab, Peter, Rafaela, Gisela, Daniel, Winter. I would like specially, to thanks Dr. Vincent Wheatley and Dr Patrick Huguenot who worked with me on this project. I thanks all my friends for their constant supports. Specially Nirav, Neel, Ravi and Shagun. Last but not least I would like to thanks my family for their constant support and encouragement.
Abstract

$SF_6$-based generator circuit breakers (GCB) are widely used to protect electric networks from the fault current that may occur in generators. The main function of the circuit breaker is to detect the fault current, remove the fault current safely from electric network, and once the fault condition is cleared, it should resume normal operations. When fault current occurs, a mechanical force separates the contacts, and an arc starts to burn between them. This high-temperature arc dissipates large amounts of energy, that has to be extinguished safely in order to protect the components of the circuit breakers. This makes the study of the arc essential as it provide important information to the designers and to help reduce the operating energy of the circuit breakers.

The aim of this thesis is to simulate the plasma arc for the high current generator circuit breaker using the equations of real gas resistive magnetohydrodynamics (MHD). We use the Nektar code developed at Brown University by the group of Prof. Karniadakis [46]. This code is based on Discontinuous Galerkin (DG) discretization of the underlying equations. In this work we have extended this code to include Runge-Kutta (RK) time-stepping, different Riemann solvers for MHD, various slope limiters, and $SF_6$ gas data. This code is then ported on a supercomputing cluster. We first use this code to test suitability and performance of the DG methods.

In order to investigate the importance of accurate Riemann solvers in RK-DG simulations, we first examine the computational efficiency of several recently developed MHD Riemann solvers, in terms of the computing time required to achieve a fixed error bound for a given problem. This was done for first- and second-order finite volume schemes. The HLLD and HLLC-L solvers were identified as good candidates for use in high-order simulations. From third- and fourth-order results for smooth solutions, we conclude that the minmod limiter is unsuitable for usage in high-order RK-DG methods. For third-order simulations of smooth problems, we find that using the HLLC-L solver results in lower errors and faster convergence than Lax-Friedrichs
flux. However, for fourth-order simulations of the same problem, we observe negligible differences in the solutions when different Riemann solvers are used. This supports the view that the results of high-order RK-DG simulations are insensitive to the choice of the Riemann solver, at least for smooth solutions. From our simulations of problems dominated by discontinuities, we find that in the vicinity of discontinuities high-order RK-DG methods behave in a similar manner as the second-order method due to use of the piecewise linear limiters in the method. Thus, the choice of Riemann solver used in a high-order method has a similarly high significance as for a second-order method. The spectral properties of RK-DG schemes are investigated by computing their approximate modified wavenumber behavior. The modified wavenumber behavior of high-order unlimited RK-DG schemes is found to be excellent. When limiting is required, however, the spectral performance of RK-DG schemes tends to that of the first-order method at high wavenumbers. This situation can only be improved by the development of high-order limiter functions for RK-DG methods.

To simulate the arc, we first perform axisymmetric arc simulations corresponding to the total current of 100kA. We show that it is possible to generate the arc using MHD equations. This was done by simulating a current profile to its steady state. We then use an external magnetic field to generate the rotation in the axis symmetric arc. Similarly we simulate a three dimensional arc using the MHD equations and use an external magnetic field to rotate the arc. This rotation can be used to build a pressure in a chamber and to extinguish the arc, which can leads to a significant reduction in operating energy of the circuit breakers.
Zusammenfassung


Um die Bedeutung der akkuraten Riemann-Löser bei RK-DG Simulationen zu verstehen, untersuchen wir zunächst die Effizienz verschiedener neuer MHD-Riemann-Löser auf Rechenzeit, die notwendig ist, um eine fixe Fehlerschranke bei einem gegebenen Problem zu erreichen. Das wurde für finite Volumen-Verfahren erster und zweiter Ordnung durchgeführt. Die HLLD und HLLC-Löser wurden als gute Kandidaten für Simulationen bei hoher Ordnung identifiziert. Aus Resultaten der dritten und vierten Ordnung bei glatten Lösungen schliessen wir, dass der minmod limiter für die RK-DG Methode höherer Ordnung unbrauchbar ist. Für Simulationen dritter Ordnung von glatten Problem

Um den Lichtbogen zu simulieren führen wir zunächst axialsymmetrische Simulationen aus, die dem Totalstrom von 100kA entsprechen. Wir zeigen, dass es möglich ist, den Lichtbogen mithilfe der MHD-Gleichungen zu generieren, was durch Simulieren von einem Stromprofil in seinen Gleichgewichtszustand durchgeführt wurde. Wir wenden dann ein externes Magnetfeld an, um Rotation im axialsymmetrischen Lichtbogen zu generieren. Auf eine ähnliche Weise simulieren wir den dreidimensionalen Lichtbogen mithilfe der MHD-Gleichungen und wenden ein externes Magnetfeld an, um den Lichtbogen zu drehen. Diese Rotation kann dazu genutzt werden, Druck in einer Kammer aufzubauen und den Lichtbogen zu löschen, was zu einer signifikanten Reduktion an Energieverbrauch im Generatorschalter führt.
1 Introduction to Circuit Breakers

1.1 Introduction to Circuit Breaker

A circuit breaker is an electrical switch designed to protect an electrical circuit from damage caused by short circuit or overload. Unlike a fuse, which operates once and then has to be replaced, a circuit breaker can be reset to resume normal operation. Circuit breakers are made and used in varying sizes, from small devices that protect an individual household appliance up to large switchgear designed to protect high voltage circuits feeding an entire city.

![Circuit Breaker Image](Figure 1.1: ABB High Current Generator Circuit Breaker system HEC-7/8, Rated for 200kA short-circuit breaking current (Figure © Copyright ABB))

A circuit breaker (see Fig. 1.1) must be able to detect a fault condition. Once a fault is detected, contacts within the circuit breaker open to interrupt
1 Introduction to Circuit Breakers

the circuit. Some mechanically stored energy within the breaker is used to separate the contacts, although some of the energy required may be obtained from the fault current itself. When current is interrupted, an arc is generated. This arc must be cooled, and extinguished in a controlled way, so that the gap between the contacts can again withstand the voltage in the circuit.

Hence, study of arc behavior in circuit breakers is of great importance to switching industry as it provides important information to the designer for improving the performance of the device. In most cases, the arc starts to burn after short time lag due to magnetic blow field of the current flowing towards contact pieces. Then the arc reaches a quenching area, consisting of the steel splitters plates. During the life-time of the arc, several phases needs to be considered. First the separation of contacts, then arc behavior in the chamber and finally removal of the arc. During these three phases, many physical phenomena occur in the circuit breaker, e.g. pressure build up, radiative transfer, convection, heat conduction, melting of contact material, magnetic and(or) electric effects. Finally, once the fault condition has been cleared, contacts must again be closed to restore power to the interrupted circuit.

1.2 $SF_6$ Gas Circuit Breakers

High-voltage circuit-breakers have considerably changed since they were introduced about 40 years ago. Several new techniques have been developed that have contributed to a large reduction of the operating energy. They are now available for vast number of indoor or outdoor applications and with various ratings, depending on the applications.

Current interruption in high-voltage circuit-breakers is obtained by separating two contacts by the $SF_6$(Sulfur Hexafluoride) gas, that has excellent dielectric and arc quenching properties. After contact separation, current is carried through an arc, it is interrupted when the arc is cooled by a gas blast of sufficient intensity. The gas blast applied on the arc must be able to cool it rapidly, so that it is able to withstand the recovery voltage that is applied across contacts after current interruption. Most of present day high-voltage circuit-breaker use $SF_6$ gas.
Several characteristics of $SF_6$ circuit breaker can explain their success:

- Reduction in the size of device due to the much higher dielectric strength of $SF_6$ gas than, e.g. air or dry nitrogen.
- Simplicity of the interrupting chamber which does not need an auxiliary breaking chamber.
- The possibility to obtain better performance.
- Short break time of 2 to 2.5 cycles.
- High electric endurance, allowing at least 25 years of operation without reconditioning.
- Reliability and availability.

The reduction in the number of interrupting chambers per pole has led to a considerable simplification of circuit breakers as well as the number of parts and seals required. As a direct consequence, the reliability of circuit breakers has improved.

Several new techniques have been developed over time with the objective of reducing the operating energy of circuit-breakers. Developments since 1996 have seen the use of the self-blast technique (see Fig. 1.2) of interruption for
1 Introduction to Circuit Breakers

$SF_6$ interrupting chambers. This technique has proved to be very efficient and has been widely applied for high voltage circuit breakers up to 550 kV. It has allowed the development of new ranges of circuit breakers operated by low energy spring-operated mechanisms. The reduction of operating energy was mainly achieved by lowering the energy used for gas compression and by making increasingly use of the arc energy to produce the pressure necessary to quench the arc and obtain current interruption. The thermal and self blast principles have enabled the use of low energy spring mechanisms for the operation of high voltage circuit breakers. They have progressively replaced the older puffer technique used in 1980s.

An significantly decrease in the operating energy can also be obtained by reducing the kinetic energy consumed during the tripping operation. One way of doing so is to displace two arcing contacts in opposite directions (see Fig. 1.3) so that the arc speed is half that of a conventional layout with a single mobile contact.

![Figure 1.3: Double action circuit breaker chamber](image)

The double motion technique halves the tripping speed of the moving part. In principle, the kinetic energy could be reduced to one forth, if the total moving mass was not increased. However, as the total moving mass is increased, the
practical reduction in kinetic energy is closer to 60 percent. The total tripping energy also includes the compression energy, which is almost the same for both techniques. Thus, reduction of the total tripping energy is lower, about 30 percent, although exact values depend on the application and the operating mechanism. Depending on the specific case, either double or single motion technique can be cheaper.

The operating energy has been reduced by 5 to 7 times during this period of 27 years. This illustrates the great progress made in the field of interrupting techniques for high-voltage circuit-breakers. In the near future, present interrupting technologies can be applied to circuit-breakers with higher rated breaking currents (63 kA to 80 kA) required in some networks with increasing power generation.

1.3 Generator Circuit Breaker

Generator circuit-breakers are $SF_6$ based circuit breakers that are installed between a generator and the step-up voltage transformer. They are generally used at the outlet of high power generators (100 MVA to 1800 MVA) in order to protect them in a reliable, fast and economic manner. Such circuit breakers must be able to allow passage of high permanent currents under continuous service (6.3 kA to 40 kA), and have a high breaking capacity (63 kA to 275 kA). They belong to the medium voltage range, but the TRV withstand capability required by ANSI/IEEE Standard C37.013 is such that the interrupting principles developed for the high-voltage range must be used. A particular set of the thermal blast technique has been developed and applied to generator circuit-breakers. The self-blast technique described above is also widely used in $SF_6$ gas generator circuit breakers, in which the contact system is driven by a low-energy, spring-operated mechanism. An example of such a device is shown in the Fig. 1.1. This circuit breaker is rated for 30 kV and 200 kA.

1.4 Literature Survey

In order to study arc behavior and related phenomena in the arc chamber of circuit breakers, several approaches can be considered: the experimental meth-
1 Introduction to Circuit Breakers

ods ([6],[7]) and the simulation approach ([4],[8]). Although large amount of information can be obtained by the first approach, optimization and dimensionless studies pose numerous problems. So, the MHD numerics approach seems to be the most complete way to consider important physical mechanism occurring in circuit breaker. Nevertheless the global model of arc and switching process remains incomplete due to the complexity of its geometry, and physical phenomena involved. Generally, these approaches do not model all the phenomena in arcing chamber and switching process. Instead they use extensive approximations, to simplify the model problem and to decrease the computational time. These approximations mainly concern interaction with walls, description of arc movements and influence of the magnetic effects on arc. For computational efficiency one major issue is computation of the magnetic field.

Various experimental and theoretical articles report, the study of circuit breakers: theoretical ones are numerous. Daube et al in [5] present an overview of the physical effects to be taken into account. One of the difficulty underline by the authors is computational time. However arc movement and external magnetic field is not taken into account. The magnetic field created by arc is calculated by the Biot-Savart law. Still the computation take 14 days on a single processor.

In [1] and [2], a more sophisticated three dimensional model has been derived. The effects of self induced magnetic field and external magnetic field are studied. Radiation terms are also included. In [1] an axis-symmetric, free burning arc column at low current level is modeled without the inclusion of self induced magnetic field. Then a three-dimensional air arc column at high-current levels with the self induced current is simulated. The numerical predictions are compared to analytical and experimental results. In [2], effects of the external magnetic fields and the gas materials on a three-dimensional high current arc is simulated. However position of the arc root stay the same during time and external magnetic field is imposed not calculated. In [9] and [10], the external magnetic field is calculated using Biot-Savart law, and the arc root is not fixed. More recently in [3], the authors study behavior of the arc and the influence of the external magnetic field. Influence of the nature of the gas on the arc velocity and on possible re-strike is also studied using a three dimensional model.
1.5 Conclusion

The mathematical models considered in [1], [2], [4], [5], [8], [9], [10] etc. are based on the Navier-Stokes equations and the Maxwell’s equations, which are solved simultaneously. They are coupled by adding the source term in momentum balance due to Lorentz force and Joule heating in energy balance equation. This is referred as the weakly coupled model here. These models are suitable for low magnetic Reynolds number but become unstable for high magnetic Reynolds number simulations. At very high currents (100kA-200kA), (specially near the arc) very high temperatures are expected that in turn gives rise to high magnetic Reynolds number. This makes weak models unsuitable for high current arc simulations. Consequently in this work we consider a model based on equation of magnetohydrodynamics(MHD), which is referred as strongly coupled model in this thesis.
1 Introduction to Circuit Breakers
2 Derivation of Equations of Magnetohydrodynamics

2.1 Introduction

A complete description of the plasma like arc in the circuit breaker, must rely on kinetic equations for each plasma species (e.g. electrons, ions). This results in the Boltzmann equation. This equation describes the change in distributions function of the species, due to advection, collisions and electric effects. The distribution function for each species is the function of seven variables (three space coordinates, three velocities and time), that make analysis and simulations of the Boltzmann equation complicated. This difficulty is overcome by using the moments of the distribution function with respect to velocity variables, and closing, resulting equations by using the closure assumptions which give rise to the Two-fluid equations and Magnetohydrodynamics (MHD) equations.

In this chapter equations of MHD are derived from the Boltzmann equation using its moments and closure assumptions by following [11]. In the Section 2.2 we present the Boltzmann equation. In Section 2.3 moments of Boltzmann equation with respect to velocity variables, are derived and the resulting equations are then closed by assuming of high collision frequency in Section 2.4. This results in Two-fluid equations. Subsequently assuming temperature equilibrium and quasi-neutrality, Two-fluid equations are simplified to the MHD equations in Section 2.5.
2 Derivation of Equations of Magnetohydrodynamics

2.2 Boltzmann Equation

Consider a plasma consisting of electrons and one type of ion. In a statistical description, the relevant physics can be expressed in terms of time-dependent distribution functions \( f_\alpha(x, v, t) \) for electrons and ions\( (\alpha \in \{e, i\}) \). These are defined as densities of representative points of particles of type \( \alpha \) in a six-dimensional phase space formed by three position coordinates \( x = (x, y, z) \) and three velocity coordinates \( v = (v_x, v_y, v_z) \). The probable number of particles of type \( \alpha \) in six-dimensional volume element \( dx dv \) centered at \( (x, v) \) at time \( t \) is then given by \( f_\alpha(x, v, t) dx dv \). The total number of particles, of type \( \alpha \), is given by \( N_\alpha = \int \int f_\alpha(x, v, t) dx dv \).

The motion of distribution function in phase space is described by the total time derivative of distribution function \( f_\alpha(x, v, t) \):

\[
\frac{df_\alpha}{dt} = \frac{\partial f_\alpha}{\partial t} + \frac{\partial f_\alpha}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial f_\alpha}{\partial v} \cdot \frac{dv}{dt} = \frac{\partial f_\alpha}{\partial t} + v \cdot \frac{\partial f_\alpha}{\partial x} + \frac{q_\alpha}{m_\alpha} (E + v \times B) \cdot \frac{\partial f_\alpha}{\partial v},
\]

where the expression for the acceleration \( dv/dt \) of the particles is inserted using equation of motion,

\[
m_\alpha \frac{dv}{dt} = q_\alpha (E + v \times B).
\]

This relation describe acceleration of the particle \( \alpha \) due to fields \( E \) and \( B \). The expression for variation in time of the distribution function is then given by a kinetic equation called, the Boltzmann equation:

\[
\frac{\partial f_\alpha}{\partial t} + v \cdot \frac{\partial f_\alpha}{\partial x} + \frac{q_\alpha}{m_\alpha} (E + v \times B) \frac{\partial f_\alpha}{\partial v} = C_\alpha := \left( \frac{\partial f_\alpha}{\partial t} \right)_{\text{coll}}.
\]

Here \( E \) and \( B \) consists of contributions of the external fields and of the averaged internal field respectively (given by the Maxwell equations), originating from the long range inter-particle interactions. The right hand side is the rate of change of distribution functions due to the short-range particle interaction i.e. collision. Neglecting collision terms leads to the collisionless Boltzmann equation also known as the Vlasov equation:

\[
\frac{\partial f_\alpha}{\partial t} + v \cdot \frac{\partial f_\alpha}{\partial x} + \frac{q_\alpha}{m_\alpha} (E + v \times B) \cdot \frac{\partial f_\alpha}{\partial v} = 0.
\]
2.2 Boltzmann Equation

A closed set of equations is then obtained by combining either Boltzmann equation or Vlasov equation with Maxwell equations to determine fields $E$ and $B$:

\[
\frac{\partial B}{\partial t} - \nabla \times E = 0, \quad \text{Faraday’s Law} \tag{2.4a}
\]

\[
\frac{1}{c^2} \frac{\partial E}{\partial t} + \mu_0 J - \nabla \times B = 0, \quad \text{Ampere’s Law, } c^2 = (\varepsilon_0 \mu_0)^{-1} \tag{2.4b}
\]

\[
\nabla \cdot E = \frac{\tau}{\varepsilon_0}, \quad \text{Poisson} \tag{2.4c}
\]

\[
\nabla \cdot B = 0. \quad \text{No magnetic monopoles} \tag{2.4d}
\]

The source terms in the Maxwell equations are related to the particle density and average velocities as follows:

\[
\tau = \sum_{\alpha} q_\alpha n_\alpha \tag{2.5a}
\]

\[
J = \sum_{\alpha} q_\alpha n_\alpha v_\alpha \tag{2.5b}
\]

where

\[
n_\alpha(x,t) = \int f_\alpha(x,v,t)dv, \tag{2.6}
\]

is the number density and

\[
v_\alpha(x,t) = \frac{1}{n_\alpha(x,t)} \int v f_\alpha(x,v,t)dv, \tag{2.7}
\]

is the velocity, of particles of type $\alpha$ at $(x,t)$ respectively. The above integration is over whole velocity phase space. Together with Eqn. (2.2) (or (2.3)) and Eq. (2.4a)-(2.4d) we have a closed set of equations.

Remark: If collision terms are considered, then we need an expression for the RHS of (2.2). The collision term $C_\alpha$ can be decomposed into its contributions $C_{\alpha\beta}$’s, that are due to the collision of the particle of type $\alpha$ with particle of type $\beta$, i.e

\[
C_\alpha = \sum_{\beta} C_{\alpha\beta} \quad \text{where} \quad C_{\alpha\beta} = C_{\alpha\beta}(f_\alpha, f_\beta). \tag{2.8}
\]

Here we have assumed that no new particle form due to the interaction between particles of different type, i.e. no fusion reaction. A typical example of the collision function is the Landau collision integral (see [11] Chapter 3).
2.3 Moments of Boltzmann Equation

The major difficulty in the analysis of the equation (2.2) (or (2.3)), is the fact that the distribution function depends on seven independent variables. To simplify analysis, one has to eliminate some of the independent variables. The most general approach is to remove velocity as an independent variable by taking the moments of the Boltzmann equation. This process will produce an infinite chain of equations that then have to be truncated in order to make sense. At this point assumptions, that impose restrictions on the domain of validity of these equations need to be made.

Different moments of Boltzmann equation are obtained by multiplying the Boltzmann equation with powers of $\mathbf{v}$ and integrating it over velocity phase space. It is assumed that distribution function vanishes sufficiently fast as $|\mathbf{v}| \to \infty$ so that all the integrals used below make sense and there is no boundary term at $|\mathbf{v}| = \infty$, in particular $\lim_{|\mathbf{v}| \to \infty} [g(\mathbf{v}) f_\alpha(\mathbf{v})] = 0$ for all functions $g(\mathbf{v})$ appearing hereafter.

Currently there is no need to consider any specific form of collision function for the analysis. Only some of its properties that follow from the conservation principles are needed. Since total number of particles of type $\alpha$ are not changed by the collision with particle of type $\beta$ (only velocities change i.e. no fusion reaction), so

$$\int C_{\alpha\beta} d\mathbf{v} = 0 \quad \forall \alpha, \beta \quad (2.9)$$

Also momentum and energy are conserved for a collision between like particles, i.e.

$$\int m_\alpha \mathbf{v} C_{\alpha\alpha} d\mathbf{v} = 0, \quad (2.10a)$$

$$\int \frac{1}{2} m_\alpha |\mathbf{v}|^2 C_{\alpha\alpha} d\mathbf{v} = 0. \quad (2.10b)$$

On other hand collision between different type of particles yield,

$$\int m_\alpha \mathbf{v} C_{\alpha\beta} d\mathbf{v} + \int m_\beta \mathbf{v} C_{\beta\alpha} d\mathbf{v} = 0, \quad (2.11a)$$

$$\int \frac{1}{2} m_\alpha |\mathbf{v}|^2 C_{\alpha\beta} d\mathbf{v} + \int \frac{1}{2} m_\beta |\mathbf{v}|^2 C_{\beta\alpha} d\mathbf{v} = 0, \quad (2.11b)$$
2.3 Moments of Boltzmann Equation

**Remark:** The Landau collision integral operator satisfy these conditions (see [11] Chapter 3).

The zeroth moment of the Boltzmann equation (2.2) is then calculated by integrating it over velocity phase space and using Eqn. (2.6) and Eqn. (2.7),

\[
\int \frac{\partial f_\alpha}{\partial t} d\mathbf{v} = \frac{\partial n_\alpha}{\partial t}, \quad (2.12a)
\]

\[
\int \mathbf{v} \cdot \frac{\partial f_\alpha}{\partial \mathbf{x}} d\mathbf{v} = \nabla \cdot (n_\alpha \mathbf{v}_\alpha), \quad (2.12b)
\]

\[
\int \frac{q_\alpha}{m_\alpha} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_\alpha}{\partial \mathbf{v}} d\mathbf{v} = 0, \quad (2.12c)
\]

\[
\int C_\alpha d\mathbf{v} = 0. \quad (2.12d)
\]

Adding these equations together result in the mass conservation equation for particles of type \( \alpha \),

\[
\frac{\partial n_\alpha}{\partial t} + \nabla \cdot (n_\alpha \mathbf{v}_\alpha) = 0, \quad (mass conservation) \quad (2.13)
\]

To obtain the first moment of Boltzmann equation we multiply Eqn. (2.2) with \( \mathbf{v} \) and integrate over the velocity space,

\[
\int \frac{\partial f_\alpha}{\partial t} \mathbf{v} d\mathbf{v} = \frac{\partial}{\partial t} (n_\alpha \mathbf{v}_\alpha), \quad (2.14a)
\]

\[
\int \mathbf{v} \cdot \frac{\partial f_\alpha}{\partial \mathbf{x}} \mathbf{v} d\mathbf{v} = \nabla \cdot \int \mathbf{v} \mathbf{v} f_\alpha d\mathbf{v} \quad (2.14b)
\]

\[
=: \nabla \cdot (n_\alpha \langle \mathbf{v} \mathbf{v} \rangle_\alpha),
\]

\[
\int \frac{q_\alpha}{m_\alpha} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_\alpha}{\partial \mathbf{v}} \mathbf{v} d\mathbf{v} = -\frac{q_\alpha n_\alpha}{m_\alpha} (\mathbf{E} + \mathbf{v}_\alpha \times \mathbf{B}), \quad (2.14c)
\]

\[
\int C_\alpha \mathbf{v} d\mathbf{v} = \int C_{\alpha\beta} \mathbf{v} d\mathbf{v}, \quad (\beta \neq \alpha)(2.14d)
\]

Multiplying these terms with particle mass \( m_\alpha \) and adding, results in the momentum conservation for particles of type \( \alpha \),

\[
\frac{\partial}{\partial t} (n_\alpha m_\alpha \mathbf{v}_\alpha) + \nabla \cdot (n_\alpha m_\alpha \langle \mathbf{v} \mathbf{v} \rangle_\alpha) - n_\alpha q_\alpha (\mathbf{E} + \mathbf{v}_\alpha \times \mathbf{B})
\]

\[
= \int C_{\alpha\beta} m_\alpha \mathbf{v} d\mathbf{v}, \quad (momentum conservation) \quad (2.15)
\]
Note that \( n_\alpha \langle vv \rangle_\alpha \) and the collision term require further explanation. Similarly to obtain the second moment equation, we multiply (2.2) with \( |v|^2 \), and integrate over velocity phase space to get,

\[
\int \frac{\partial f_\alpha}{\partial t} |v|^2 dv = \frac{\partial (n_\alpha \langle |v|^2 \rangle_\alpha)}{\partial t}, \quad (2.16a)
\]

\[
\int v \cdot \frac{\partial f_\alpha}{\partial x} |v|^2 dv =: \nabla \cdot (n_\alpha \langle |v|^2 v \rangle_\alpha), \quad (2.16b)
\]

\[
\int \frac{q_\alpha}{m_\alpha} (E + v \times B) \cdot \frac{\partial f_\alpha}{\partial v} |v|^2 dv = -2 \frac{q_\alpha n_\alpha}{m_\alpha} E \cdot v_\alpha, \quad (2.16c)
\]

\[
\int C_\alpha |v|^2 dv = \int C_\alpha \beta |v_\alpha|^2 dv, \quad (\beta \neq \alpha). (2.16d)
\]

Multiplying these terms with \( \frac{1}{2} m_\alpha \) and adding them, give equation for the energy conservation.

\[
\frac{\partial}{\partial t} \left( \frac{1}{2} m_\alpha n_\alpha \langle |v|^2 \rangle_\alpha \right) + \nabla \cdot \left( \frac{1}{2} m_\alpha n_\alpha \langle |v|^2 v \rangle_\alpha \right) - n_\alpha q_\alpha E \cdot v_\alpha
\]

\[
= \int \frac{1}{2} m_\alpha C_\alpha \beta |v|^2 dv, \quad (\text{energy conservation}) \quad (2.17)
\]

Again there are averages and collision terms that need further simplifications in terms of the known variables and this should result in a set of self-consistent macroscopic equations.

The chain of these moment equations can continue indefinitely. As one can see that the equation for each moment introduce new unknown whose temporal evolution is described by next order moment of the Boltzmann equation. This infinite process has to be truncated somewhere, to obtain a closed set of equations. In fluid dynamics, this number is just five, containing the first three moments i.e. Eqns (2.13), (2.15) and (2.17).

In order to close this set of equations, number of assumptions has to be made, which decide the domain of validity of these equations. Before that, it is convenient if the momentum and energy equations are in a form that has more macroscopic appearance. To do so, we divide velocity \( \mathbf{v} \) into two parts, containing a random velocity part \( \tilde{\mathbf{v}}_\alpha \) and the average velocity \( \mathbf{v}_\alpha \), i.e.

\[
\tilde{\mathbf{v}}_\alpha := \mathbf{v} - \mathbf{v}_\alpha, \quad \text{where} \quad \langle \tilde{\mathbf{v}}_\alpha \rangle = 0 \quad (2.18)
\]
The random velocity part $\langle |v|^2 \rangle_\alpha$ in energy Eqn. (2.17) gives rise to the quantity measuring mean kinetic energy of the particles in a frame moving with the velocity $v_\alpha$, which is the temperature $T_\alpha$ for particle type $\alpha$, given by

$$T_\alpha(x, t) := \frac{m_\alpha}{3k} \langle |\tilde{v}_\alpha|^2 \rangle,$$  \hspace{1cm} (2.19)

with constant $k$. Similarly the random velocity part of the term $\langle vv \rangle_\alpha$ in the moment equation (2.15) gives rise to the stress tensor $P_\alpha$ defined as

$$P_\alpha := n_\alpha m_\alpha \langle \tilde{v}_\alpha \tilde{v}_\alpha \rangle = p_\alpha I + \Pi_\alpha$$ \hspace{1cm} (2.20)

whose isotropic part is directly related to the temperature by

$$p_\alpha(x, t) := \frac{1}{3} Tr(P_\alpha) = \frac{1}{3} n_\alpha m_\alpha \langle |\tilde{v}_\alpha|^2 \rangle = n_\alpha k T_\alpha$$ \hspace{1cm} (2.21)

and traceless tensor $\Pi_\alpha$ is the contribution due to anisotropy of distribution function, given by

$$\Pi_\alpha := n_\alpha m_\alpha \langle \tilde{v}_\alpha \tilde{v}_\alpha - \frac{1}{3} |\tilde{v}_\alpha|^2 I \rangle.$$ \hspace{1cm} (2.22)

Finally the random velocity part of vector $\langle |v|^2 v \rangle_\alpha$ in energy equation (2.17), gives rise to a quantity,

$$h_\alpha := \frac{1}{2} n_\alpha m_\alpha \langle |\tilde{v}_\alpha|^2 \tilde{v}_\alpha \rangle$$ \hspace{1cm} (2.23)

which is the heat flow by the random motion of particles of type $\alpha$. Similarly we simplify the collision terms by transforming them to frame the which is moving with the velocity $v_\alpha$. Using Eqn. (2.8) and RHS of Eqn. (2.15),

$$\int C_{\alpha\beta} m_\alpha v dv = R_\alpha := \int C_{\alpha\beta} m_\alpha \tilde{v}_\alpha dv$$ \hspace{1cm} (2.24)

which is friction force i.e. mean momentum transfer from particles $\beta$ to $\alpha$. For energy Eqn. (2.17) using similar calculation, we get,

$$\int C_{\alpha\beta} \frac{1}{2} m_\alpha |v|^2 dv = \int C_{\alpha\beta} \left( m_\alpha v_\alpha \cdot \tilde{v}_\alpha + \frac{1}{2} m_\alpha |\tilde{v}_\alpha|^2 \right) dv = v_\alpha \cdot R_\alpha + Q_\alpha$$ \hspace{1cm} (2.25)

where

$$Q_\alpha := \int C_{\alpha\beta} \frac{1}{2} m_\alpha |\tilde{v}_\alpha|^2 dv \hspace{1cm} (\beta \neq \alpha)$$ \hspace{1cm} (2.26)
2 Derivation of Equations of Magnetohydrodynamics

is heat transfer to the system of particles of type $\alpha$ due to collision with unlike particles of type $\beta$. Now substituting Eqn. (2.19)–(2.26) into Eqn. (2.15) and Eqn. (2.17),

$$\frac{\partial}{\partial t} (n_\alpha m_\alpha v_\alpha) + \nabla \cdot (n_\alpha m_\alpha v_\alpha v_\alpha) + \nabla \cdot P_\alpha - n_\alpha q_\alpha (E + v_\alpha \times B) = R_\alpha$$

(2.27)

$$\frac{\partial}{\partial t} \left( \frac{1}{2} n_\alpha m_\alpha |v_\alpha|^2 \right) + \frac{\partial}{\partial t} \left( \frac{3}{2} n_\alpha kT_\alpha \right) + \nabla \cdot \left( \frac{1}{2} n_\alpha m_\alpha |v_\alpha|^2 v_\alpha + \frac{3}{2} n_\alpha kT_\alpha v_\alpha + v_\alpha \cdot P_\alpha + h_\alpha \right) - n_\alpha q_\alpha E \cdot v_\alpha = v_\alpha \cdot R_\alpha + Q_\alpha$$

(2.28)

The momentum Eqn. (2.27) can be simplified further, by using mass conservation equation. Also energy equation is simplified by removing the kinetic energy part, using Eqn. (2.13) and Eqn. (2.27). The three lowest moments then take the compact form,

$$\left( \frac{\partial}{\partial t} + v_\alpha \cdot \nabla \right) n_\alpha + n_\alpha \nabla \cdot v_\alpha = 0,$$

(2.29a)

$$n_\alpha m_\alpha \left( \frac{\partial}{\partial t} + v_\alpha \cdot \nabla \right) v_\alpha + \nabla (n_\alpha kT_\alpha) - n_\alpha q_\alpha (E + v_\alpha \times B) = -\nabla \cdot \Pi_\alpha + R_\alpha,$$

(2.29b)

$$\frac{3}{2} n_\alpha \left( \frac{\partial}{\partial t} + v_\alpha \cdot \nabla \right) kT_\alpha + n_\alpha kT_\alpha \nabla \cdot v_\alpha = -\Pi_\alpha : \nabla v_\alpha - \nabla \cdot h_\alpha + Q_\alpha,$$

(2.29c)

Remark: Note that the relation between $\Pi_\alpha$, $h_\alpha$, $R_\alpha$, $Q_\alpha$ and macroscopic quantities $n_\alpha,v_\alpha,T_\alpha$, electromagnetic quantities $E$ and $B$ is still needed, to get a close set of macroscopic equations (2.29a)-(2.29c).

2.4 Two Fluid Equations

Let us consider a plasma consisting of electrons with charge $q_e = -e$ and one kind of ions with charge number $Z$, and so charge $q_i = Ze$. Then the two-fluid
2.4 Two Fluid Equations

Eqns.[2.29a]-[2.29c] for this plasma take the form,

\[
\left( \frac{\partial}{\partial t} + \mathbf{v}_e \cdot \nabla \right) n_e + n_e \nabla \cdot \mathbf{v}_e = 0, \quad (2.30a)
\]

\[
\left( \frac{\partial}{\partial t} + \mathbf{v}_i \cdot \nabla \right) n_i + n_i \nabla \cdot \mathbf{v}_i = 0, \quad (2.30b)
\]

\[
n_e m_e \left( \frac{\partial}{\partial t} + \mathbf{v}_e \cdot \nabla \right) \mathbf{v}_e + \nabla(n_e kT_e) - n_e q_e (\mathbf{E} + \mathbf{v}_e \times \mathbf{B}) = -\nabla \cdot \Pi_e + \mathbf{R}_e, \quad (2.30c)
\]

\[
n_i m_i \left( \frac{\partial}{\partial t} + \mathbf{v}_i \cdot \nabla \right) \mathbf{v}_i + \nabla(n_i kT_i) - n_i q_i (\mathbf{E} + \mathbf{v}_i \times \mathbf{B}) = -\nabla \cdot \Pi_i + \mathbf{R}_i, \quad (2.30d)
\]

\[
\frac{3}{2} n_e \left( \frac{\partial}{\partial t} + \mathbf{v}_e \cdot \nabla \right) kT_e + n_e kT_e \nabla \cdot \mathbf{v}_e = -\Pi_e : \nabla \mathbf{v}_e - \nabla \cdot \mathbf{h}_e + \mathbf{Q}_e, \quad (2.30e)
\]

\[
\frac{3}{2} n_i \left( \frac{\partial}{\partial t} + \mathbf{v}_i \cdot \nabla \right) kT_i + n_i kT_i \nabla \cdot \mathbf{v}_i = -\Pi_i : \nabla \mathbf{v}_i - \nabla \cdot \mathbf{h}_i + \mathbf{Q}_i, \quad (2.30f)
\]

where

\[
p_e = n_e kT_e, \quad p_i = n_i kT_i, \quad (2.31)
\]

together with Maxwell Eqns.[2.4], they form a closed set of equations with charge density,

\[
\tau = -e(n_e - Zn_i), \quad (2.32)
\]

and current density

\[
\mathbf{J} = -e(n_e \mathbf{v}_e - Zn_i \mathbf{v}_i), \quad (2.33)
\]

as source terms.

**Remark:** The quantities \( \mathbf{R}_{e,i} \) depend on each other because of momentum and energy conservation of the unlike particle collisions (See Eqns. [2.11a]-[2.11b]), which results in,

\[
\mathbf{R}_e = -\mathbf{R}_i, \quad (2.34)
\]

and

\[
\mathbf{Q}_e = -(\mathbf{v}_e - \mathbf{v}_i) \cdot \mathbf{R}_e - \mathbf{Q}_i, \quad (2.35)
\]

To completely close the above equations the expressions for the transport coefficients in terms of macroscopic quantities are needed. The derivation of these expressions is one of the main objective of *transport theory* and quiet
2 Derivation of Equations of Magnetohydrodynamics

involved. Details for derivation of these relations can be found in [11] Chapter 3. Here we omit the details and only describe the final expressions,

\[ \Pi_\alpha \approx \nu_\alpha \nabla v_\alpha =: \nu_\alpha \pi_\alpha, \]  
\[ h_\alpha \approx -k_\alpha \nabla(kT_\alpha), \]
\[ R_\alpha \approx -q_\alpha n_\alpha \eta J. \]  

Here \( \nu_\alpha \) is viscosity, \( k_\alpha \) is heat conductivity, of particles \( \alpha \) and \( \eta \) is resistivity of the fluid. With these relations, Eqns. \[(2.30)\] with Maxwell’s Eqns. \[(2.4)\] form a closed set of equations.

2.5 Equations of Magnetohydrodynamics

To derive one-fluid MHD equations from two fluid MHD equations, we consider the following one fluid variables which are defined using two fluid variables,

\[ \rho := n_e m_e + n_i m_i, \]  
\[ \tau := -e(n_e - Zn_i), \]
\[ \rho v := n_e m_e v_e + n_i m_i v_i, \]  
\[ J := -e(n_e v_e - Zn_i v_i). \]  

**Notation:** We use notation \( v \) for the new definition of velocity given in \[(2.37c)\], not for the velocity of particles, used in previous sections. 
Now the assumption that temperature of the ions and electrons is same, i.e. temperature equilibrium has already taken place, is made. So, our time scale of interest is \( t_H \gg t_{eq} \). This allow the definition of one pressure for the whole fluid(including both ion and electron),

\[ p := p_e + p_i = (n_e + n_i)kT \]  

Note that except the temperature difference, the full information contained in the two-fluid equations is retained, while using these relations. The inverse
2.5 Equations of Magnetohydrodynamics

relations can be derived as follows:

\[ n_e = \frac{Z[\rho - (m_i/Ze)\tau]}{m_i(1 + \xi)} \approx \frac{Z}{m_i(1 + \xi)} \rho, \quad (2.39a) \]

\[ n_i = \frac{\rho + \xi(m_i/Ze)\tau}{m_i(1 + \xi)} \approx \frac{1}{m_i(1 + \xi)} \rho, \quad (2.39b) \]

\[ v_e = \frac{\rho v - (m_i/Ze)J}{\rho - (m_i/Ze)\tau} \approx v - \frac{m_i J}{Ze \rho}, \quad (2.39c) \]

\[ v_i = \frac{\rho v + \xi(m_i/Ze)J}{\rho + \xi(m_i/Ze)\tau} \approx v + \xi \frac{m_i J}{Ze \rho}, \quad (2.39d) \]

\[ p_e = n_e kT = \frac{n_e}{n_i + n_e} p \approx \frac{Z}{1 + Z} p, \quad (2.39e) \]

\[ p_i = n_i kT = \frac{n_i}{n_i + n_e} p \approx \frac{1}{1 + Z} p, \quad (2.39f) \]

Here \( \xi \) is ratio of the masses over charges and given by,

\[ \xi := \frac{Zm_e}{m_i}, \quad (2.40) \]

Now a ssumptions of quasi charge-neutrality is made, i.e.

\[ |n_e - Ze_i| \ll n_e, \text{ or } \frac{m_i}{Ze} |\tau| \ll \rho, \quad (2.41) \]

This assumption hold very well for the plasma phenomena in macroscopic hydrodynamic length scales, where

\[ \lambda_H \gg \lambda_D, \]

is true. Here \( \lambda_D \approx 70\sqrt{T/n} \) is Debye length which measures a typical size of the region over which charge imbalance can create thermal fluctuations i.e. under above approximations these local thermal fluctuation effects are ignored. This approximation hold very well for the arc in circuit breakers, as \( n \approx n_e \) is very large at high temperatures. Note that this assumptions does not implies that there are no charge particles, but only that there are very few free charge particles. This assumption allow us to make approximations in Eqns. (2.39).

To derive one fluid equations, we first multiplying mass conservation equations for electron (2.30a) and ion (2.30b) with there respective masses and add them
2 Derivation of Equations of Magnetohydrodynamics

which results in the equation of mass conservations:

\[ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \]  

(2.42)

whereas multiplying these equations by their respective charges and adding results in the equation of charge conservation.

\[ \frac{\partial \tau}{\partial t} + \nabla \cdot \mathbf{J} = 0. \]  

(2.43)

Similarly adding the pair of equations for electron (Eqn.(2.30c)) and ion velocities (Eqn.(2.30d)), and using approximations in Eqns.(2.39) leads to, equation of velocity,

\[ \rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \nabla \mathbf{v} + \nabla \left( \frac{m_i}{Ze} \right)^2 \nabla \cdot \left( \frac{1}{\rho} - \frac{1}{\rho} \mathbf{J} \mathbf{J} \right) + \nabla p - \tau \mathbf{E} - \mathbf{J} \times \mathbf{B} = -\nabla \cdot (\Pi_e + \Pi_i) \]  

(2.44)

Finally adding the Eqns. (2.30e) and (2.30f), results in the heat balance equation,

\[ \frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla p + \frac{\gamma p}{Z + 1} \mathbf{v} \cdot \nabla \mathbf{v} + \frac{\gamma p}{Z + 1} \mathbf{v} \cdot \nabla \left( \frac{m_i}{Ze} \right)^2 \nabla \cdot \left( \frac{1}{\rho} \mathbf{J} \mathbf{J} \right) + \nabla p - \gamma \mathbf{J} \cdot \nabla \left( \frac{p}{\rho} \right) \]  

\[ = -\left( \frac{\gamma}{Z + 1} \right) \left( \Pi_e : \nabla \mathbf{v} + \Pi_i : \nabla \mathbf{v} + \nabla \cdot (\mathbf{h}_e + \mathbf{h}_i) - (1 + \xi) \frac{m_i}{Ze} \mathbf{J} \cdot \mathbf{R}_e \right), \]  

(2.45)

where \( \gamma \) is ratio of the specific heats. Multiplying equations (2.30c) and (2.30d), with charges and dividing it by masses results in equation for the current, which is also known as generalized Ohm’s law,

\[ \frac{\partial \mathbf{J}}{\partial t} + \nabla \cdot \left( \mathbf{J} \mathbf{v} + \mathbf{v} \mathbf{J} - \frac{m_i}{Ze} (1 - \xi) \frac{1}{\rho} \mathbf{J} \mathbf{J} \right) + \frac{1}{\xi} \frac{Ze}{m_i} \left( (1 - \xi) \mathbf{J} \times \mathbf{B} - \frac{Z - \xi}{Z + 1} \nabla p \right) \]  

\[ - \frac{1}{\xi} \left( \frac{Ze}{m_i} \right)^2 \rho (\mathbf{E} + \mathbf{v} \times \mathbf{B}) = \frac{Ze}{\xi m_i} \left( \nabla \cdot (\Pi_e - \xi \Pi_i) - (1 + \xi) \mathbf{R}_e \right) \]  

(2.46)

Viscosity terms for the two fluid equations can be approximated by using the viscosity of one fluid as follows:

\[ -\nabla \cdot (\Pi_e + \Pi_i) \approx -\nabla \cdot \Pi = \nabla \cdot \nu \pi \]

and

\[ \Pi_e : \nabla \mathbf{v}_e + \Pi_i : \nabla \mathbf{v}_i \approx \Pi : \nabla \mathbf{v} = \nu \pi : \nabla \mathbf{v}. \]
Ion-electron momentum transfer $R_e$ can be written in terms of current density. Using

$$R_e = \eta e n_e J \approx \eta \frac{Z e}{(1 + \xi)m_i} \rho J.$$  \hfill (2.47)

and heat flux for both kind of particles can be approximated by using

$$h \approx h_e + h_i.$$  \hfill (2.48)

To derive the equations that model large scale dynamics, the maximal ordering for one fluid MHD needs to be introduced which is the Alfven speed ordering:

$$v_A^2 := \frac{|B|^2}{\mu_0 \rho} \sim \frac{\gamma p}{\rho}.$$  \hfill (2.49)

Hydrodynamic length scales and time scales are then choosen depending on the size of plasmas

$$\lambda_{MHD} := |\nabla|^{-1} \sim a \gg \lambda_D.$$  \hfill (2.50)

Note that,

$$\frac{m_i |\tau|}{Ze \rho} \sim \frac{v_{th,i} \lambda_D}{c a} \ll 1,$$  \hfill (2.51)

with additional small parameter $\frac{v_{th,i}}{c}$, which ratio of ion thermal speed with speed of light. At this point another assumption is needed which results in restriction on magnitude of current in the sense of maximal ordering.

$$|v_e - v_i| \ll |v| \quad \text{or} \quad \frac{m_i |J|}{Ze \rho |v|} \ll 1.$$  \hfill (2.52)

Remark: The restriction needs to be understood in the sense of maximal ordering, in a way that even for static plasmas ($|v| = 0$), there is an upper limit to the current density, where $|v|$ is replaced by $v_A$, in (2.52). So,

$$\frac{m_i |J|}{Ze \rho |v|} \ll 1,$$  \hfill (2.53)

Under these approximations the Eqn. (2.44) reduces to

$$\rho \frac{\partial v}{\partial t} + \rho v \cdot \nabla v + \nabla p - \tau E + J \times B = 0,$$  \hfill (2.54)
Derivation of Equations of Magnetohydrodynamics

and Eqn.(2.45) becomes,

\[
\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla p + \gamma p \nabla \cdot \mathbf{v} = (\gamma - 1) \eta |\mathbf{J}|^2. \tag{2.55}
\]

The equation for current (2.46) can be rewritten as,

\[
-\xi \left( \frac{m_i}{Ze} \right)^2 \frac{1}{\rho} \left[ \frac{\partial \mathbf{J}}{\partial t} + \nabla \cdot (\mathbf{Jv} + \mathbf{vJ}) \right] - \left( \frac{m_i}{Ze} \right) \left[ \frac{1}{\rho} \frac{(1 - \xi) \mathbf{J} \times \mathbf{B} - \frac{Z - \xi}{Z + 1} \nabla p}{\rho} \right] + \mathbf{E} + \mathbf{v} \times \mathbf{B} \right] = \eta \mathbf{J}. \tag{2.56}
\]

Note that the term containing time derivative is second order smaller, and the Hall current term \((1 - \xi) \mathbf{J} \times \mathbf{B} - \frac{Z - \xi}{Z + 1} \nabla p\), is first order smaller with respect to the parameter given by (2.53). Keeping these terms will conserve the complete current evolution equation, but the computation with these terms is going to be very inefficient because it would force computation to proceed on the time scale of two-fluid model instead of on the much longer time scale of one-fluid MHD model. So for the sake of consistency these terms have to be dropped, which results in

\[
\mathbf{J} = \sigma \mathbf{E}' \quad \text{where} \quad \mathbf{E}' = \mathbf{E} + \mathbf{v} \times \mathbf{B}
\]

with parameter \(\sigma := \eta^{-1}\). Finally collecting all the equation to form the set of one fluid MHD equations,

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad \text{mass conservation} \tag{2.57a}
\]

\[
\left[ \frac{\partial \tau}{\partial t} + \nabla \cdot \mathbf{J} \right] = 0, \quad \text{charge continuity} \tag{2.57b}
\]

\[
\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \mathbf{v} + \nabla p \left[ -\tau \mathbf{E} - \mathbf{J} \times \mathbf{B} \right] = -\nabla \cdot (\Pi), \quad \text{momentum} \tag{2.57c}
\]

\[
\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla p + \gamma p \nabla \cdot \mathbf{v} = (1 - \gamma) (\Pi : \nabla \mathbf{v} + \eta |\mathbf{J}|^2 \nabla \cdot \mathbf{h}), \quad \text{internal energy} \tag{2.57d}
\]

\[
\frac{\partial B}{\partial t} + \nabla \times \mathbf{E} = 0, \quad \text{Faraday law}, \tag{2.57e}
\]

\[
\left[ \frac{\partial E}{c^2 \partial t} + \mu_0 \mathbf{J} - \nabla \times \mathbf{B} \right] = 0, \quad \text{Ampere’s law} \tag{2.57f}
\]

where \(\eta \mathbf{J} = \mathbf{E} + \mathbf{v} \times \mathbf{B} \) \quad \text{Ohm’s law,} \tag{2.58}
and variable $\mathbf{E}$ and $\mathbf{B}$ have to satisfy initial conditions,

$$
\begin{align*}
\nabla \cdot \mathbf{E} &= \tau / \epsilon_0, \quad \text{Poisson}, \\
\nabla \cdot \mathbf{B} &= 0, \quad \text{No magnetic monopoles}.
\end{align*}
$$

First of all notice that Eqn. (2.57b) is just the consequence of Eqns. (2.57f) and (2.59a). So this equation can be ignored. Now the electromagnetic quantities under the maximal ordering assumption (2.51) have magnitude,

$$
\begin{align*}
 j &\sim \frac{B}{\mu_0 a}, \quad E \sim vB, \quad \tau \sim \frac{\epsilon_0 E}{a} \sim \frac{\epsilon_0 vB}{a}.
\end{align*}
$$

Using this,

$$
\frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} \approx \frac{v^2 B}{c^2 a} \ll |\nabla \times \mathbf{B}| \approx \frac{B}{a}.
$$

So in nonrelativistic MHD we can ignore this term in Eqn. (2.57f),

$$
\mathbf{J} = \mu_0^{-1}(\nabla \times \mathbf{B}).
$$

Now (2.62) implies that $\nabla \cdot \mathbf{J} = 0$ which is contradicting Eqn. (2.57b) and (2.59a). Using the ordering argument once again,

$$
|\tau'\mathbf{E}| \sim \frac{\epsilon_0 E^2}{a} \sim \frac{v^2 B^2}{c^2 \mu_0 a} \ll |\mathbf{J} \times \mathbf{B}| \sim \frac{B^2}{\mu_0 a}
$$

So the Eqn. (2.59a) has to be dropped. Finally collecting all the equations we have following set of equations for one fluid MHD,

$$
\begin{align*}
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) &= 0, \quad \text{mass conservation}, \\
\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p - \mathbf{J} \times \mathbf{B} &= -\nabla \cdot (\Pi), \quad \text{momentum} , \\
\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla p + \gamma p \nabla \cdot \mathbf{v} &= (1 - \gamma)(\Pi : \nabla \mathbf{v} + \eta |\mathbf{J}|^2) \\
&+ \nabla \cdot \mathbf{h}, \quad \text{internal energy}, \\
\frac{\partial B}{\partial t} + \nabla \times \mathbf{E} &= 0, \quad \text{Faraday law}, \\
\mu_0 \mathbf{J} - \nabla \times \mathbf{B} &= 0, \quad \text{Ampere’s law}, \\
\mathbf{E} + \mathbf{v} \times \mathbf{B} &= \eta \mathbf{J}, \quad \text{Ohm’s law} \\
\nabla \cdot \mathbf{B} &= 0, \quad \text{No magnetic monopoles}.
\end{align*}
$$
2 Derivation of Equations of Magnetohydrodynamics

The set of Eqns. (2.64a) - (2.64g) models one fluid plasma flows and effects of magnetic fields on it. These equations are easier to analyze compared to the Boltzmann equation (2.3). Also these equations provide us with an efficient model for computations as time stepping is not constraint due to presence of higher order terms. Although we have lost the complete description of plasma given by statistical approach, these equations are still a good approximation under the closure assumptions we have made.
3 Equations of Magnetohydrodynamics

3.1 Introduction

In Chapter 2 we have derived the equations of MHD, which describe the fluid description of plasma, using the moments of the Boltzmann equation. These equations model flow of plasma under the influence of magnetic field and have the form of balance laws. In this chapter, we will study various properties of these equations.

In Section 3.2, these equations are converted to equations for the conserved variables, that results in a set of conservation laws. To study the importance of various physical effects, at the scale of interest, we introduce non-dimensional variables in Section 3.3. This leads to the equations for the dimensionless variables with parameters like the Reynolds number, Lundquist’s Number etc. In the Section 3.4 we analyze the convection part of MHD equations and prove that equations of ideal MHD are system of hyperbolic conservation laws. We also present expressions for the complete set of eigenvalues and eigenvectors. These will be useful in deriving numerical methods in Chapter 4. In Section 3.5 we discuss the viscous part of the MHD equations.

3.2 MHD equations in conservative form

The equations derived in Section 2.5 are equations for the primitive variables i.e. velocity and pressure, instead of conservative variables like momentum and energy. In this section we will rewrite these equations, as equations for
3 Equations of Magnetohydrodynamics

the conservative variables. Recalling the Eqns. (2.64a)-(2.64g),

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad \text{mass conservation, (3.1a)}
\]

\[
\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p - \mathbf{J} \times \mathbf{B} = -\nabla \cdot (\Pi), \quad \text{momentum, (3.1b)}
\]

\[
\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla p + \gamma p \nabla \cdot \mathbf{v} = (1 - \gamma)(\Pi : \nabla \mathbf{v} + \eta |\mathbf{J}|^2 + \nabla \cdot \mathbf{h}), \quad \text{internal energy, (3.1c)}
\]

\[
\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0, \quad \text{Faraday law. (3.1d)}
\]

The equation for conservation of mass (3.1a) is already in the desired form so we consider the velocity Eqn.(3.1b). Using Eqn.(3.1a), first two terms can be combined as,

\[
\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \nabla \mathbf{v} = \frac{\partial \rho \mathbf{v}}{\partial t} + \mathbf{v} \nabla \cdot (\rho \mathbf{v}) + \rho \mathbf{v} \cdot \nabla \mathbf{v} = \frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v} + \mathbf{p})
\]

and magnetic field terms due to Lorentz’s Force, can be written as,

\[
-\mathbf{J} \times \mathbf{B} = \mathbf{B} \times \frac{1}{\mu_0} (\nabla \times \mathbf{B}) = \frac{1}{\mu_0} \nabla \left( \frac{1}{2} |\mathbf{B}|^2 - \mathbf{B} \cdot \mathbf{B} \right),
\]

using Ampere’s law (2.64c). Combining these expressions we get the equation for momentum balance in conservation form,

\[
\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \left( \rho \mathbf{v} \mathbf{v} + p + \frac{1}{\mu_0} \left( \frac{1}{2} |\mathbf{B}|^2 - \mathbf{B} \cdot \mathbf{B} \right) - \Pi \right) = 0. \quad (3.2)
\]

Eqn.(3.1d), for the magnetic field can be simplified using Ohm’s law (2.64f), i.e.

\[
\mathbf{E} = -\mathbf{v} \times \mathbf{B} + \eta \mathbf{J},
\]

and,

\[
\nabla \times (-\mathbf{v} \times \mathbf{B}) = \nabla \cdot (\mathbf{v} \mathbf{B} - \mathbf{B} \mathbf{v}).
\]

So,

\[
\nabla \times \mathbf{E} = \nabla \cdot (\mathbf{v} \mathbf{B} - \mathbf{B} \mathbf{v}) + \frac{1}{\mu_0} \nabla \times (\eta \nabla \times \mathbf{B}).
\]
3.2 MHD equations in conservative form

Using these expressions, finally (3.1d) can be written as,

\[ \frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{vB} - \mathbf{Bv}) + \frac{1}{\mu_0} \nabla \times (\eta \nabla \times \mathbf{B}) = 0 \] (3.3)

To derive the equation for the conservation of energy, one has to add expressions for the internal energy, kinetic energy and magnetic energy. The equation for internal energy is derived by dividing the equation for pressure (3.1c) with \((\gamma - 1)\) and using the ideal gas relation,

\[ \rho e = \frac{p}{\gamma - 1}. \] (3.4)

Where \(e\) is internal energy per unit of mass. This result in,

\[ \frac{\partial (\rho e)}{\partial t} + \mathbf{v} \cdot (\rho e) + \gamma \rho e \nabla \cdot \mathbf{v} - \Pi : \nabla \mathbf{v} - \eta |\mathbf{J}|^2 = 0. \] (3.5)

For the equation of the kinetic energy we take dot product of velocity Eqn. (3.1b) with \(\mathbf{v}\),

\[ \rho \mathbf{v} \cdot \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) + \mathbf{v} \cdot \nabla p - \mathbf{v} \cdot (\mathbf{J} \times \mathbf{B}) - \mathbf{v} \cdot (\nabla \cdot \Pi) = 0, \]

\[ \implies \frac{\partial}{\partial t} \left( \frac{1}{2} \rho \mathbf{v}^2 \right) - \left( \frac{1}{2} |\mathbf{v}|^2 \frac{\partial \rho}{\partial t} + \frac{1}{2} \rho \mathbf{v} \cdot \nabla |\mathbf{v}|^2 \right) + \mathbf{v} \cdot \nabla p - \mathbf{v} \cdot (\mathbf{J} \times \mathbf{B}) - \mathbf{v} \cdot (\nabla \cdot \Pi) = 0. \] (3.6)

Expression for the magnetic energy is derived by multiplying magnetic field Eqn. (3.3) with \(\mathbf{B}\),

\[ \mathbf{B} \cdot \frac{\partial \mathbf{B}}{\partial t} - \mathbf{B} \cdot \nabla \times (\mathbf{v} \times \mathbf{B}) + \mathbf{B} \cdot \nabla \times (\eta \mathbf{J}) = 0, \]

\[ \implies \frac{\partial}{\partial t} \left( \frac{1}{2} |\mathbf{B}|^2 \right) + \nabla \cdot (\mathbf{B} \times (\mathbf{v} \times \mathbf{B})) - (\mathbf{v} \times \mathbf{B}) \cdot \nabla \times \mathbf{B} + \nabla \cdot (\eta \mathbf{J} \times \mathbf{B}) + \eta |\mathbf{J}|^2 = 0, \]

which give us the final expression for magnetic energy,

\[ \frac{\partial}{\partial t} \left( \frac{1}{2} |\mathbf{B}|^2 \right) + \nabla \cdot (\mathbf{B} \mathbf{Bv} - \mathbf{v} \cdot \mathbf{BB}) + \mathbf{v} \cdot (\mathbf{J} \times \mathbf{B}) + \nabla \cdot (\eta \mathbf{J} \times \mathbf{B}) + \eta |\mathbf{J}|^2 = 0. \] (3.7)
Adding Eqns. (3.5), (3.6) and (3.7) we get the equation for the conservation of total energy,

$$\frac{\partial}{\partial t} \left( \rho e + \frac{1}{2} \rho |v|^2 + \frac{1}{2\mu_0} |B|^2 \right) + \nabla \cdot \left( \left( \frac{1}{2} \rho |v|^2 + \rho e + p + \frac{1}{2\mu_0} |B|^2 \right) v \right) \quad (3.8)$$

$$- v \cdot \frac{1}{\mu_0} BB \right) + \nabla \cdot \left( \Pi : v + \frac{\eta}{\mu_0} J \times B - \kappa \nabla T \right) = 0,$$

Remark: The last term can be simplified as,

$$\frac{\eta}{\mu_0} (J \times B) = \frac{1}{\mu_0^2} \left( \eta B \cdot \nabla B - \frac{1}{2} \nabla \left( \eta |B|^2 \right) \right)$$

using the identity $J = \frac{1}{\mu_0} (\nabla \times B)$. The Eqns. (3.1a), (3.2) and (3.8) are the equations of the conservative variables and are in the conservation form.

### 3.3 Non-dimensional MHD equations

In this section the non-dimensional MHD equations are derived using reference variables. Derivation of the non-dimensional equations gives rise to parameters like Reynolds Number that indicate the importance of various physical effects like viscosity at the scale of interest.

We first introduce non dimensional variable $x'$,

$$x = L_0 x', \quad (3.9)$$

where $L_0$ is the chosen reference length scale. Similarly for velocity we define non-dimensional velocity $v' = v/V_0$ with $V_0 = \sqrt{P_0/\rho_0}$, the reference velocity derived using reference pressure $P_0$ and reference density $\rho_0$. Reference density can be found using gas data, once reference pressure $P_0$ and reference temperature $T_0$ is chosen. Reference magnetic field is derived using the relation,

$$B_0 = \sqrt{P_0\mu_0}, \quad (3.10)$$

and non-dimensional magnetic field $B' = B/B_0$. Substituting $\rho = \rho' \rho_0$, $v = V_0 v'$, in Eqn.(3.1a), and length scales $x = (x, y, z) = L_0 x' = L_0(x', y', z')$,
results in, same equation, except that the dimensional variables are replaced by non dimensional ones.

Now consider Eqn.(3.2), converting derivative from of dimensional variable to that of non-dimensional variables we obtain,

\[
\frac{\partial \rho \mathbf{v}}{\partial t} = \frac{\rho_0 V_0}{t_0} \frac{\partial \rho' \mathbf{v}'}{\partial t'}, \quad \nabla \cdot (\nu \pi) = \frac{V_0}{L_0^2} \nabla_{x'} \cdot (\nu \pi'),
\]

\[
\nabla \cdot \left( \rho \mathbf{v} \mathbf{v} + p - \frac{1}{\mu_0} \left( \mathbf{B} \mathbf{B} + \frac{1}{2} |\mathbf{B}|^2 \right) \right) = \frac{\rho_0 V_0^2}{L_0} \nabla_{x'} \cdot \left( \rho' \mathbf{v}' \mathbf{v}' + p' - \mathbf{B}' \mathbf{B}' + \frac{1}{2} |\mathbf{B}'|^2 \right),
\]

where we have substituted \( \Pi = \nu \pi \). Combining these terms, and dividing the whole equation by \( \frac{\rho_0 V_0}{t_0} \), which is the scaling of advection terms, result in non-dimensional momentum conservation equation,

\[
\frac{\partial (\rho' \mathbf{v}')}{\partial t} + \nabla \cdot \left( \rho' \mathbf{v}' \mathbf{v}' - \mathbf{B}' \mathbf{B}' + p' + \frac{1}{2} |\mathbf{B}'|^2 - \frac{1}{Re \pi} \right) = 0,
\]

(3.11)

where \( Re = \frac{\rho_0 V_0 L_0}{\mu_0} \), is Reynolds number. Considering magnetic field Eqn.(3.3), and using scaling (3.10), for magnetic field,

\[
\frac{\partial \mathbf{B}}{\partial t} = \frac{B_0}{t_0 \partial t'} \frac{\partial \mathbf{B}'}{\partial t'}, \quad \nabla \times (\mathbf{B} \times \mathbf{v}) = \frac{V_0 B_0}{L_0} \nabla_{x'} \times (\mathbf{B} \times \mathbf{v}'),
\]

\[
\nabla \times \left( \frac{1}{\mu_0} \eta (\nabla \times \mathbf{B}) \right) = \frac{\eta B_0}{\mu_0 L_0} \nabla_{x'} \times (\nabla_{x'} \times \mathbf{B}').
\]

Combining these terms we have non-dimensional magnetic field equation,

\[
\frac{\partial \mathbf{B}'}{\partial t} + \nabla \times \left( \mathbf{v}' \times \mathbf{B}' + \frac{1}{S_r} (\nabla_{x'} \times \mathbf{B}') \right) = 0,
\]

(3.12)

where \( S_r = \frac{\mu_0 V_0 L_0}{\eta} \) is Lundquist Number and \( \eta = \frac{1}{\sigma} \). Repeating the same for energy Eqn.(3.8)

\[
\frac{\partial E}{\partial t} = \frac{\rho_0 V_0^2}{t_0} \frac{\partial E'}{\partial t'}, \quad \nabla \nu \pi \cdot \mathbf{v} = \frac{\nu V_0^2}{L_0^2} \nabla_{x'} \nu \pi' \cdot \mathbf{v}',
\]

\[
\nabla \cdot \left( (E+p) \mathbf{v} + \frac{1}{\mu_0} \left( \frac{1}{2} |\mathbf{B}|^2 - \mathbf{B} \mathbf{B} \right) \cdot \mathbf{v} \right) = \frac{\rho_0 V_0^3}{L_0} \nabla_{x'} \cdot \left( (E'+p') \mathbf{v}' + \left( \frac{1}{2} |\mathbf{B}'|^2 - \mathbf{B}' \mathbf{B}' \right) \cdot \mathbf{v}' \right),
\]

(3.11)
3 Equations of Magnetohydrodynamics

\[
\nabla \left( \frac{\eta}{\mu_0} \left( \mathbf{B} \cdot \nabla \mathbf{B} - \nabla \left( \frac{1}{2} |\mathbf{B}|^2 \right) \right) \right) = \frac{B_0^2}{L_0^2 \mu_0^2} \nabla_{x'} \left( \eta \mathbf{B}' \cdot \nabla \mathbf{B}' - \nabla \left( \frac{\eta}{2} |\mathbf{B}'|^2 \right) \right),
\]

\[
\nabla (\kappa \nabla T) = \frac{p_0}{\rho_0 R_0 L_0^2} \nabla_{x'} (\nabla_{x'} \kappa T').
\]

Using Eqn. (3.10),

\[
\frac{B_0^2 V_0}{\mu_0 L_0} = \frac{\rho_0 V_0^3}{L_0},
\]

we get,

\[
\frac{\partial E'}{\partial t'} + \nabla_{x'} \cdot \left( (E' + p') \mathbf{v}' + \left( \frac{1}{2} |\mathbf{B}'|^2 - \mathbf{B}' \mathbf{B}' \right) \right) \mathbf{v}'
\]

\[
- \frac{1}{Re} \pi' \cdot \mathbf{v}' + \frac{1}{S_r} \left( \mathbf{B} \cdot \nabla_{x'} \mathbf{B}' - \nabla_{x'} \left( \frac{1}{2} |\mathbf{B}'|^2 \right) \right) - \frac{1}{G_r} \nabla_{x'} T' = 0,
\]

Here \( G_r = \frac{\rho_0 L_0 V_0 R_0}{\kappa} \), with \( R_0 = R(T_0) \) gas constant. Finally collecting all the equations we have following set of conservation laws,

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0,
\]

\[
\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot \left( \rho \mathbf{v} \mathbf{v} - \mathbf{B} \mathbf{B} + \left( p + \frac{1}{2} |\mathbf{B}|^2 \right) - \frac{1}{Re} \Pi \right) = 0,
\]

\[
\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \left( \mathbf{v} \times \mathbf{B} + \frac{1}{S_r} (\nabla \times \mathbf{B}) \right) = 0,
\]

\[
\frac{\partial E}{\partial t} + \nabla \cdot \left( (E + p) \mathbf{v} + \left( \frac{1}{2} |\mathbf{B}|^2 \mathbf{I} - \mathbf{B} \mathbf{B} \right) \right) \mathbf{v}
\]

\[
- \frac{1}{Re} \pi \cdot \mathbf{v} + \frac{1}{S_r} \left( \mathbf{B} \cdot \nabla \mathbf{B} - \nabla \left( \frac{1}{2} |\mathbf{B}|^2 \right) \right) - \frac{1}{G_r} \nabla T = 0.
\]

with divergence free condition \((2.64g)\) and equation of state,

\[
E = \frac{p}{\gamma - 1} + \frac{1}{2} \rho |\mathbf{v}|^2 + \frac{1}{2} |\mathbf{B}|^2,
\]

for non-dimensional energy. Here we have changed the notations for the dimensionless variable to that of dimensional variable.
3.4 Ideal MHD

Eqns. (3.14) are a set of parabolic equations. These equations reduce to the set of Ideal MHD equation in the case where Lundquist number, Reynolds number and $G_r$ are very high i.e. contributions due to viscous, resistive and thermal conduction terms are ignored. This lead to a system of first order PDEs,

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

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot \left( \rho \mathbf{vv} - \mathbf{BB} + \left( p + \frac{1}{2} |\mathbf{B}|^2 \right) \right) = 0,$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{vB} - \mathbf{Bv}) = 0,$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left( (E + p) \mathbf{v} + \left( \frac{1}{2} |\mathbf{B}|^2 \mathbf{I} - \mathbf{BB} \right) \cdot \mathbf{v} \right) = 0,$$

$$\nabla \cdot \mathbf{B} = 0.$$

In this section we will prove that this is a system of hyperbolic conservation laws.

The magnetic field Eqn. (3.16c) preserves the divergence free condition (3.16e), so it can be considered as the condition on the initial conditions. The system of Eqns. (3.16) without the divergence constraint can be written in the form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial}{\partial x} (\mathbf{F}_x(\mathbf{U})) + \frac{\partial}{\partial y} (\mathbf{F}_y(\mathbf{U})) + \frac{\partial}{\partial z} (\mathbf{F}_z(\mathbf{U})) = 0,$$

where $\mathbf{U}$ is the state vector for conservative variables, i.e. $\mathbf{U} = (\rho, \rho v_x, \rho v_y, \rho v_z, B_x, B_y, B_z, E)^T$ and $\mathbf{F}_x, \mathbf{F}_y$ and $\mathbf{F}_z$ are the fluxes in $x, y$ and $z$ directions respectively. Flux in $x$ direction can be written as,

$$\mathbf{F}_x(\mathbf{U}) = \begin{pmatrix}
\rho v_x \\
\rho v_x^2 + p + \frac{1}{2}(B_y^2 + B_z^2 - B_x^2) \\
\rho v_x v_y - B_x B_y \\
\rho v_x v_z - B_x B_z \\
0 \\
v_x B_y - v_y B_x \\
v_x B_z - v_z B_x \\
(E + p + \frac{1}{2}(B_x^2 + B_y^2 + B_z^2))v_x - B_x(v_x B_x + v_y B_y + v_z B_z)
\end{pmatrix}$$
In one dimension case Eqn. (3.16e) reduces to,

\[ \frac{\partial}{\partial x} B_x = 0, \]  

(3.19)

which implies that \( B_x \) is constant w.r.t \( x \). On other hand, \( x \)-component of Eqn. (3.16c) also implies,

\[ \frac{\partial}{\partial t} B_x = 0, \]  

(3.20)

i.e. \( B_x \) is constant w.r.t \( t \). So in one dimension system reduced to,

\[ \frac{\partial}{\partial t} U + \frac{\partial}{\partial x} (F_x(U)) = 0, \]

(3.21a)

\[ B_x = \text{const.} \]

(3.21b)

For the calculation of the eigenvalues, it is convenient if we remove \( B_x \) components from the system. This leads to

\[ \frac{\partial}{\partial t} u + \frac{\partial}{\partial x} (f_x(u; B_x)) = 0, \]

(3.22a)

\[ B_x = \text{const.} \]

(3.22b)

where \( u = (\rho, \rho v_x, \rho v_y, \rho v_z, B_y, B_z, E)^\top \), and flux \( f_x(u; B_x) \) is given by,

\[
f_x(B_x; u) = \begin{pmatrix}
\rho v_x \\
\rho v_x^2 + p + \frac{1}{2}(B_y^2 + B_z^2) \\
\rho v_x v_y - B_x B_y \\
\rho v_x v_z - B_x B_z \\
v_x B_y - v_y B_x \\
v_x B_z - v_z B_x \\
(E + p + \frac{1}{2}(B_y^2 + B_z^2))v_x - B_x(v_y B_y + v_z B_z)
\end{pmatrix}
\]

(3.23)

Note that here expression for the total energy is now modified to,

\[ E := \frac{p}{\gamma - 1} + \frac{1}{2}(B_y^2 + B_z^2) + \frac{1}{2} \rho |v|^2 \]  

(3.24)

To connect three dimensional with one dimensional ideal MHD equations, we introduce the mapping \( A_{8 \to 7} : \mathbb{R}^8 \to \mathbb{R}^7, U \to u \), similarly for the flux we introduce the mapping, \( A_{7 \to 8} : \mathbb{R} \times \mathbb{R}^7 \to \mathbb{R}^8 \), so that \( F(U) = A_{7 \to 8}(B_x(U); f_x(B_x(U)); A_{8 \to 7}(U)) \).
For constant $B_z$ we have one to one mapping from 8 variable to 7 variable formulations. Let us now introduce the matrix as follows,  
For $n = (n_x, n_y, n_z) \in \mathbb{R}^3$ with $|n| = 1$, define,  
$$B(n) := \begin{pmatrix}
    n_x & n_y & n_z \\
    -\frac{n_y}{\sqrt{1-n^2}} & -\frac{n_z}{\sqrt{1-n^2}} & 0 \\
    -\frac{n_z n_x}{\sqrt{1-n^2}} & -\frac{n_y n_z}{\sqrt{1-n^2}} & \sqrt{1-n^2} \\
    0 & 0 & \text{sign}(n_z) \\
    0 & 0 & 1 \\
    -\text{sign}(n_z) & 0 & 0
\end{pmatrix}$$  
which is an orthogonal matrix with $\det B(n) = 1$. So it defines a rotation linear mapping $x \rightarrow B(n)y$. Also $B(n)n = (1, 0, 0)$. Now using matrix  
$$R(n) = \begin{pmatrix}
    1 & 0 & 0 & 0 \\
    0 & B(n) & 0 & 0 \\
    0 & 0 & B(n) & 0 \\
    0 & 0 & 0 & 1
\end{pmatrix}.$$  
A simple calculation shows that ideal MHD equations are rotation invariant, i.e.  
$$F_x n_x + F_y n_y + F_z n_z = R^{-1}(n)F_x(R(n)U).$$  
Also using the two mapping we have introduce before, we have the realtion  
$$F_x n_x + F_y n_y + F_z n_z = R^{-1}(n)A_{7\rightarrow 8}(B^n_x, f_x(B^n_x; R(n)U)).$$  
To calculate the eigenvector and eigenvalue of the system, it is efficient if we introduce a change of variable, $W = W(U)$. For this change of variable we have,  
$$\frac{\partial}{\partial t} U(W) + \frac{\partial}{\partial x} F_x(U(W)) = 0, \quad (3.29a)$$  
$$\iff \frac{\partial U}{\partial W} \frac{\partial W}{\partial t} + D F_x(U(W)) \frac{\partial U}{\partial W} \frac{\partial W}{\partial x} = 0, \quad (3.29b)$$  
$$\iff \frac{\partial}{\partial t} W + \frac{\partial W}{\partial U} D F_x(U(W)) \frac{\partial U}{\partial W} \frac{\partial W}{\partial x} = 0. \quad (3.29c)$$
3 Equations of Magnetohydrodynamics

We introduce following change of variables, \( \mathbf{W} = (\tau, \mathbf{v}, \mathbf{B}, p)^\top \) where \( \tau = \frac{1}{\rho} \). Calculations show that

\[
\mathcal{J}_8(\mathbf{w}) := \frac{\partial \mathbf{W}}{\partial \mathbf{U}} D\mathbf{f}_x(\mathbf{U}(\mathbf{W})) \frac{\partial \mathbf{U}}{\partial \mathbf{W}}
\]

\[
= \begin{pmatrix}
v_x & -\tau & 0 & 0 & 0 & 0 & 0 \\
0 & v_x & 0 & 0 & -B_x \tau & -B_y \tau & B_z \tau \\
0 & 0 & v_x & 0 & -B_y \tau & -B_x \tau & 0 \\
0 & 0 & 0 & v_x & -B_z \tau & 0 & -B_x \tau \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & B_y & -B_x & 0 & -u_y & v_x & 0 \\
0 & B_z & 0 & -B_x & 0 & v_x & 0 \\
0 & \gamma p & 0 & 0 & (\mathbf{B} \cdot \mathbf{v}) \frac{\partial p}{\partial E} & 0 & 0 & v_x \\
\end{pmatrix}
\]

(3.30)

Similarly for one dimensional case we introduce variables \( \mathbf{w} = (\tau, \mathbf{v}, B_y, B_z, p)^\top \) which result in the system,

\[
\frac{\partial \mathbf{u}(\mathbf{w})}{\partial t} + \frac{\partial \mathbf{f}_x(\mathbf{u}(\mathbf{w}))}{\partial x} = 0, \quad (3.31a)
\]

\[
\iff \quad \frac{\partial \mathbf{u}}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial t} + D\mathbf{f}_x(\mathbf{u}(\mathbf{w})) \frac{\partial \mathbf{u}}{\partial \mathbf{w}} \frac{\partial \mathbf{u}}{\partial x} = 0, \quad (3.31b)
\]

\[
\iff \quad \frac{\partial \mathbf{w}}{\partial t} + D\mathbf{f}_x(\mathbf{u}(\mathbf{w})) \frac{\partial \mathbf{w}}{\partial \mathbf{w}} \frac{\partial \mathbf{u}}{\partial x} = 0, \quad (3.31c)
\]

A simple calculation shows that

\[
\mathcal{J}_7(\mathbf{w}) := \frac{\partial \mathbf{w}}{\partial \mathbf{u}} D\mathbf{f}_x(\mathbf{u}(\mathbf{w})) \frac{\partial \mathbf{u}}{\partial \mathbf{w}} = \begin{pmatrix}
v_x & -\tau & 0 & 0 & 0 & 0 & 0 \\
0 & v_x & 0 & 0 & -B_y \tau & B_x \tau & \tau \\
0 & 0 & v_x & 0 & -B_x \tau & 0 & 0 \\
0 & 0 & 0 & v_x & 0 & -B_x \tau & 0 \\
0 & B_y & -B_x & 0 & v_x & 0 & 0 \\
0 & B_z & 0 & -B_x & 0 & v_x & 0 \\
0 & \gamma p & 0 & 0 & 0 & 0 & v_x \\
\end{pmatrix}
\]

(3.32)

Now the eigenvalue of matrix \( \mathcal{J}_7(\mathbf{w}) \), are given by,

\[
\lambda_{f\pm} = v_x \pm v_f \quad \text{(fast waves)} \quad (3.33a)
\]

\[
\lambda_{a\pm} = v_x \pm v_{ax} \quad \text{(Alfven waves)} \quad (3.33b)
\]

\[
\lambda_{s\pm} = v_x \pm v_s \quad \text{(slow waves)} \quad (3.33c)
\]

\[
\lambda_e = v_x \quad \text{(entropy wave)} \quad (3.33d)
\]
3.4 Ideal MHD

where the velocities \( v_f \geq v_{ax} \geq v_s \geq 0 \) are,

\[
v_{ax}^2 = \frac{B_x^2}{\rho} \quad \text{and} \quad v_{fs}^2 = \frac{1}{2} \left[ (v_a^2 + c_s^2) \pm \sqrt{(v_a^2 + c_s^2)^2 - 4v_{ax}^2c_s^2} \right]
\]

(3.34)

with

\[
c_s^2 = \frac{\gamma p}{\rho} \quad \text{and} \quad v_a^2 = \frac{B_x^2 + B_y^2 + B_z^2}{\rho}
\]

(3.35)

and the corresponding right eigenvectors are,

\[
\mathbf{r}_e = \begin{pmatrix} \tau \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad (3.36)
\]

\[
\mathbf{r}_{\pm v_{ax}} = \frac{v_f}{\sqrt{2}} \begin{pmatrix} 0 \\ 0 \\ \pm \beta_z \\ \mp \beta_y \\ -\text{sign}(B_x)\sqrt{\frac{1}{\tau}}\beta_z \\ \text{sign}(B_x)\sqrt{\frac{1}{\tau}}\beta_y \\ 0 \end{pmatrix}, \quad (3.37)
\]

\[
\mathbf{r}_{\pm v_f} = \mathbf{R}_{\pm v_f} \begin{pmatrix} \mp \alpha_f \tau \\ \alpha_f v_f \\ -\alpha_s \beta_y v_{ax} \text{sign}(B_x) \\ -\alpha_s \beta_z v_{ax} \text{sign}(B_x) \\ \pm \alpha_s \beta_y v_f \sqrt{\frac{1}{\tau}} \\ \pm \alpha_s \beta_z v_f \sqrt{\frac{1}{\tau}} \\ \pm \alpha_f \gamma p \end{pmatrix}, \quad (3.38)
\]
3 Equations of Magnetohydrodynamics

\[
\begin{align*}
\mathbf{r}_{\pm v_s} = \mathbf{R}_{\pm v_s} &= \begin{pmatrix}
\mp \alpha_s \tau \\
\alpha_s v_s \\
\alpha_f \beta_y c_s \text{sign}(B_x) \\
\pm \alpha_f \beta_z c_s \text{sign}(B_x) \\
\mp \alpha_f \beta_y c_s^2 \sqrt{\frac{1}{\tau}} \\
\pm \alpha_f \beta_z c_s^2 \sqrt{\frac{1}{\tau}} \\
\pm \alpha_s \gamma p
\end{pmatrix}, \\
(3.39)
\end{align*}
\]

Here

\[
\alpha_f = \begin{cases}
\frac{\sqrt{v_f^2 - v_{ax}^2}}{v_f^2 - v_s^2} (v_f^2 > v_s^2), \\
\frac{1}{v_f^2 - v_s^2} (v_f^2 = v_s^2)
\end{cases} \quad \alpha_s = \begin{cases}
\frac{\sqrt{v_f^2 - c_s^2}}{v_f^2 - v_s^2} (v_f^2 > v_s^2), \\
0 (v_f^2 = v_s^2)
\end{cases},
(3.40)
\]

\[
\beta_y = \begin{cases}
\frac{B_y}{\sqrt{B_y^2 + B_z^2}} (B_y^2 + B_z^2 > 0), \\
\frac{1}{\sqrt{2}} (B_y^2 + B_z^2 = 0)
\end{cases},
(3.41)
\]

\[
\beta_z = \begin{cases}
\frac{B_z}{\sqrt{B_y^2 + B_z^2}} (B_y^2 + B_z^2 > 0), \\
\frac{1}{\sqrt{2}} (B_y^2 + B_z^2 = 0)
\end{cases},
(3.42)
\]

and

\[
\mathbf{R}_{\pm v_f} := \frac{v_f}{\sqrt{\alpha_f^2 (v_f^2 + c_s^2) + \alpha_s^2 (v_f^2 + v_{ax}^2)}},
(3.43)
\]

\[
R_{\pm v_s} := \frac{v_s}{\sqrt{\alpha_f^2 c_s^2 (v_f^2 + c_s^2) + \alpha_s^2 v_f^2 (v_s^2 + c_s^2)}},
(3.44)
\]

with

\[
\text{sign}(B_x) = \begin{cases}
1 & (B_x \geq 0), \\
-1 & (B_x < 0)
\end{cases}.
(3.45)
\]

Similarly the left eigenvectors are given by,

\[
l_e = \left(\frac{1}{\tau}, 0, 0, 0, 0, 0, \frac{1}{\gamma p}\right),
(3.46)
\]

\[
l_{\pm v_{ax}} = \frac{1}{\sqrt{2} v_f} \left(0, 0, \pm \beta_z, \mp \beta_y, -\text{sign}(B_x) \sqrt{\tau} \beta_z, \text{sign}(B_x) \sqrt{\tau} \beta_y, 0\right),
(3.47)
\]
3.4 Ideal MHD

\[ l_{\pm v_f} = \frac{R_{\pm v_f}}{v_f^2} \left( 0, \alpha_f v_f, -\alpha_s \beta_y v_{ax} \text{sign}(B_x), -\alpha_s \beta_z v_{ax} \text{sign}(B_x), \right. \]
\[ \left. \pm \alpha_s \beta_y v_f \sqrt{\tau}, \pm \alpha_s \beta_z v_f \sqrt{\tau}, \pm \alpha_f \tau \right), \]  

\[ l_{\pm v_s} = \frac{R_{\pm v_s}}{v_f^2} \left( 0, \alpha_s v_s, \alpha_f \beta_y c_s \text{sign}(B_x), \alpha_f \beta_z c_s \text{sign}(B_x), \right. \]
\[ \left. \mp \alpha_f \beta_y c_s^2 \sqrt{\tau}, \mp \alpha_f \beta_z c_s^2 \sqrt{\tau}, \mp \alpha_s \tau \right). \]

We now have complete set of eigenvalues and eigenvectors for one dimension ideal MHD equations.

3.4.1 Hyperbolicity of Ideal MHD equations

**Definition 3.1.** A system of \( n \) partial differential equations of the form

\[ \frac{\partial U}{\partial t} + \sum_{i=1}^{d} \frac{\partial F_i(U)}{\partial x_i} = 0, \]  

with

\[ U(x, t) : D \times [0, T] \rightarrow \mathbb{R}^n, \]
\[ F_i : \mathbb{R}^n \rightarrow \mathbb{R}^n, (1 \leq i \leq n), \]
\[ x = (x_1, \ldots, x_n) \in D \subset \mathbb{R}^d \]
is called hyperbolic, if for all vectors \( e \in \mathbb{S}^{d-1} := \{(e_1, \ldots, e_n) \in \mathbb{R}^d \mid ||e||_2 \leq 1\} \) and for all \( U \in \mathbb{R}^m \) the matrix

\[ J(U) := \sum_{i=1}^{n} D F_i(U)e_i \]  

is diagonizable over \( \mathbb{R}^n \) with real eigenvalues. In addition, if all the eigenvalues are distinct then we call the system a strictly hyperbolic system.

Using this definition we have following result,
3 Equations of Magnetohydrodynamics

Theorem 3.2 (Hyperbolicity of one dimensional ideal MHD). The one dimensional MHD system \[ [3.22] \] is hyperbolic, since \( Df(u) \) has the same eigenvalues as \( J_7(w) \), and linearly-independent left and right eigenvectors are given by,

\[
\begin{align*}
    r_k^{Df} &= \frac{\partial u}{\partial w} r_k, \\
    l_k^{Df} &= l_k \frac{\partial w}{\partial u}
\end{align*}
\]

for \( 1 \leq k \leq 7 \). Moreover, eigensystem is still orthogonal, i.e. \( l_i^{Df} l_j^{Df} = \delta_{i,j} \) for \( 1 \leq i, j \leq 7 \).

Proof. Using \( Df(u) = \frac{\partial u}{\partial w} J_7(w) \frac{\partial w}{\partial u} \), we obtain

\[
Df(u) \frac{\partial u}{\partial w} r_k = \frac{\partial u}{\partial w} J_7(w) r_k = \lambda_k \frac{\partial u}{\partial w} r_k
\]

and

\[
l_k \frac{\partial w}{\partial u} Df(u) = l_k J_7(w) \frac{\partial w}{\partial u} = \lambda_k l_k \frac{\partial w}{\partial u}.
\]

This also implies that the eigenvectors are linearly independent because matrices \( \frac{\partial u}{\partial w} \) and \( \frac{\partial w}{\partial u} \) are regular. In the end for \( 1 \leq i, j \leq 7 \)

\[
l_i^{Df} l_j^{Df} = l_i \frac{\partial w}{\partial u} \frac{\partial u}{\partial w} r_j = l_i r_j = \delta_{i,j}.
\]

Definition 3.3 (\( \hat{J}_8 \)). We define \( \hat{J}_8 \) to be matrix \[ [3.30] \] with fifth column replaced with zeros.

From this definition it follows that,

Proposition 3.4 (Eigenvalues and Eigenvectors of \( J_8 \) and \( \hat{J}_8 \)). Let \( \lambda_k(B_x; w) \), \( r_k(B_x; w) \), \( r_k(B_x; w) \) for \( 1 \leq k \leq 7 \) be the eigenvalues and eigenvectors of \( J_7(B_x; w) \) as given by Eqns. \[ [3.33]-[3.50] \]. If \( B_x = W_5 \) and \( w = (W_1, W_2, W_3, W_4, W_5, W_6, W_7) \) then we find

1. The eigenvalues of \( J_8(W) \) and \( \hat{J}_8 \) are \( \lambda_k \) for \( 1 \leq k \leq 7 \) and \( \lambda_8 := \lambda_{B_x} := 0 \).

2. The right eigenvectors for \( J_8(W) \) corresponding to \( \lambda_k \) for \( 1 \leq k \leq 7 \) are given by, \( (r_k, 1, r_k, 2, r_k, 3, r_k, 4, 0, r_k, 5, r_k, 6, r_k, 7)^\top \); the left eigenvector corresponding to \( \lambda_8 = \lambda_{B_x} \) is \( (0, 0, 0, 0, 1, 0, 0, 0) \).
3.4 Ideal MHD

3. The right eigenvectors for \( \hat{J}_8(W) \) corresponding to \( \lambda_k \) for \( 1 \leq k \leq 7 \) are given by, \( ([r_k]_1, [r_k]_2[,r_k]_3[,r_k]_4, 0, [r_k]_5[,r_k]_6[,r_k]_7)^\top \); corresponding to \( \lambda_8 = \lambda_{B_x} \) is \( (0, 0, 0, 1, 0, 0, 0) \).

4. The left eigenvectors for \( \hat{J}_8(W) \) corresponding to \( \lambda_k \) for \( 1 \leq k \leq 7 \) are given by, \( ([l_k]_1, [l_k]_2[,l_k]_3[,l_k]_4, 0, [l_k]_5[,l_k]_6[,l_k]_7) \); ‘the left eigenvector corresponding to \( \lambda_8 = \lambda_{B_x} \) is \( (0, 0, 0, 1, 0, 0, 0) \).

5. The right are left eigenvectors of \( \hat{J}_8(W) \) are orthogonal.

**Proposition 3.5 (Hyperbolicity of [3.21]).** The quasi linear system,
\[
\frac{\partial U}{\partial t} + \frac{\partial U}{\partial W} \hat{J}_8(W) \frac{\partial W}{\partial U} \frac{\partial U}{\partial x} = 0
\]
is hyperbolic and for smooth solutions satisfying [3.21b] the system is equivalent to the system [3.21].

**Proof.** We can get the complete set of orthonormal left and right eigenvectors for the matrix \( \frac{\partial U}{\partial W} \hat{J}_8(W) \frac{\partial W}{\partial U} \), using same calculations as in lemma(add). So system is hyperbolic. The equivalence of the systems follows from [3.29], [3.21b] and that \( DF_x(U) = \frac{\partial U}{\partial W} \hat{J}_8(W) \frac{\partial W}{\partial U} \).

**Theorem 3.6 (Hyperbolicity of 3D MHD).** The three dimensional MHD system [3.17] is hyperbolic.

**Proof.** Consider the matrix
\[
A(U)u = DF_x(U)n_x + DF_y(U)n_y + DF_z(U)n_z.
\]
We need to show that this matrix is diagnolizable over \( \mathbb{R}^n \), with real eigenvalues. Let \( r_k^{DF_x}(U) \) the right eigenvectors of \( DF_x(U) \) for \( k \in \{1, \ldots, 8\} \) with \( \lambda_8 = \lambda_{B_x} \). Then using the the matrix \( R(n) \) introduced earlier we get,
\[
(A(U)u)R^{-1}(n)r_k(R(n)U) = (DF_x(U)n_x + DF_y(U)n_y + DF_z(U)n_z)R^{-1}(n)r_k(R(n)U)
\]
\[
= D(R^{-1}(n)F_x(R(n)U))R^{-1}(n)r_k^{DF_x}(R(n)U)
\]
\[
= (R^{-1}(n)DF_x(R(n)U)R(n))R^{-1}(n)r_k^{DF_x}(R(n)U)
\]
\[
= \lambda_k(R(n)U)R^{-1}(n)r_k^{DF_x}(R(n)U).
\]

\[\square\]
3 Equations of Magnetohydrodynamics

We have proved that three dimensional ideal MHD system is a system of hyperbolic conservation laws. We have also described the eigenvalues and eigenvectors of the system.

3.5 Viscous Fluxes of MHD equations

The equations (3.14) are advection diffusion equations and can be written in the form,

\[
\frac{\partial U}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} - \frac{\partial F^\text{visc}_x}{\partial x} - \frac{\partial F^\text{visc}_y}{\partial y} - \frac{\partial F^\text{visc}_z}{\partial z} = 0. \tag{3.53}
\]

In the last Section 3.4 we discussed the advection fluxes \(F_x, F_y\) and \(F_z\). In this Section we shall describe the viscous fluxes \(F^\text{visc}_x, F^\text{visc}_y\) and \(F^\text{visc}_z\). They are parabolic terms due to viscosity, heat dissipation and resistivity terms. Simplifying the Eqns (3.14), and comparing with (3.53), we obtain,

\[
F^\text{visc}_x(U) = \begin{cases} 
0, \\
\frac{1}{Re}(2\frac{\partial v_x}{\partial x} - \frac{2}{3}\nabla \cdot v), \\
\frac{1}{Re}(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y}), \\
\frac{1}{Re}(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z}), \\
\frac{1}{Sr}(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y}), \\
\frac{1}{Sr}(\frac{\partial B_z}{\partial x} - \frac{\partial B_x}{\partial z}), \\
F^\text{visc}_{xE},
\end{cases} \tag{3.54}
\]

where \(F^\text{visc}_{xE}\) is energy term in the flux \(F^\text{visc}_x\), which are given by

\[
F^\text{visc}_{xE} = \frac{1}{Re} \left( \left(2\frac{\partial v_x}{\partial x} - \frac{2}{3}\nabla \cdot v\right)v_x + \left(\frac{\partial v_y}{\partial y} + \frac{\partial v_x}{\partial y}\right)v_y + \left(\frac{\partial v_z}{\partial z} + \frac{\partial v_x}{\partial z}\right)v_z \right) + \frac{1}{Gr}\frac{\partial T}{\partial x}, \tag{3.55}
\]

The expressions for \(F^\text{visc}_y\) and \(F^\text{visc}_z\) are similar.
Note that Reynolds Number $Re$ derived in Section 3.3 is a dimensionless number which is nothing but the ratio of advection forces to that of viscous forces. So it tells us the importance of these forces in the flow. High Reynolds number means that viscous terms are less important and so they can be ignored. The dimensionless number $Gr$ which occurs in the energy diffusion flux is the ratio of advection forces to that of heat conduction. Similarly Lundquists Number $Sr$ is the ratio of advection forces to that of magnetic diffusion forces. High Lundquist number indicates that plasma is highly conductive. On the other hand low Lundquist number means that we have a resistive plasma.

In the simulation of arc plasma for circuit breakers, away from arc we have low conductivity which implies low Lundquist Number. This leads a stiff problem if one uses explicit time stepping, due to very high magnetic diffusivity coefficient.
3 Equations of Magnetohydrodynamics
4 Discontinuous Galerkin Methods for Magnetohydrodynamics

4.1 Introduction

To simulate plasma flows with small-scale features such as turbulence, high-order accurate, low dissipation numerical methods are required. Runge-Kutta Discontinuous Galerkin (RKDG) methods are high-order finite element methods for nonlinear hyperbolic conservation laws. They also incorporate several important features of finite volume schemes. It is important, because a high order of accuracy does not ensure good dissipation properties. For example, WENO schemes (see [12]) are formally high order accurate, but because they exhibit high numerical dissipation at high wave numbers, Hill et al. [13], find that they are not suitable for large eddy simulations (LES).

The discontinuous galerkin methods were first introduced by Hill et al. in [14] for neutron transport equations(linear hyperbolic equations). In [15] Saint et al. proved a rate of convergence of $(\Delta x)^k$ for general triangulations and of $(\Delta x)^{k+1}$ for Cartesian meshes. In case of general triangulation result was then improved by Johnson et al. in [16] to have order of convergence of $(\Delta x)^{k+1/2}$, which was confirmed to be optimal by Peterson in [17].

These methods were then generalized for system of hyperbolic conservation laws by Cockburn et al. in series of papers ( [18], [19], [20], [21], [22]). For the spacial discretization, the solution is approximated using piecewise polynomials over each cell and then exact or approximate Riemann solvers from finite volume methods are used for computation of the numerical flux, whereas limiters (see [23]) are used to achieve non-oscillatory behavior of solutions with shocks. Due to this reason, DG methods can be seen as generalizations of finite volume methods to higher order. For time discretization total variation diminishing (TVD) explicit Runge-Kutta (RK) methods proposed by Shu et
al. in [24] are used. These methods are known as Runge-Kutta Discontinuous Galerkin (RKDG) methods.

RKDG methods have several important advantages. Like finite element methods RKDG methods are better suited for the complicated geometries. These methods can easily handle adaptivity strategies, because of the assumed discontinuity of the solution at the cell faces. So, refining or unrefining of the grid can be done without taking into account continuity restrictions typical for conforming finite element methods. Similarly, the degree of polynomial inside an element can be changed without affecting the solution in other cells. Another important advantage is that these methods are highly parallelizable because to update the solution in a given cell, only information needed is from cells sharing a face with that cell.

In this chapter we present a overview of DG methods for MHD equations. In the Section 4.2, we describe the DG method for ideal MHD equations. We discuss the variational formulation, and describe the basis functions for two and three dimensional domains. Runge-Kutta methods for time stepping are also discussed. In the Section 4.3, we present various numerical fluxes based on approximate Riemann solvers. Limiters for the problems containing discontinuities are presented for one- and two- dimensional domains in Section 4.4. An implementation of the divergence free condition using Powell’s source term is briefly discussed in Section 4.5. For the discretization of viscous terms we use Local Discontinuous Galerkin (LDG) methods, presented in Section 4.6. Finally we present the complete algorithm for the computation, implemented in Nektar code.

### 4.2 Runge-Kutta Discontinuous Galerkin method for ideal MHD

Consider following equation for general conservation law:

\[
\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = 0,
\]

(4.1)

where \( \mathbf{U} \) is a vector of conserved variable and \( \mathbf{F} \) is the advection flux. In Section 3.4, the ideal MHD Eqns. (3.16) are written in this form. We also assume
4.2 Runge-Kutta Discontinuous Galerkin method for ideal MHD

that divergence free condition is only imposed only on the initial data. It is sufficient to consider that \( U = u \), a scalar, for the description of DG method because in case it is a vector, we can apply the same method component to component. In this case \( \mathbf{F} = \mathbf{f} \) is a vector. So, we describe DG methods for scalar conservation laws,

\[
\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f} = 0,
\]

(4.2)

4.2.1 Variational Form

Consider domain \( \Omega \subset \mathbb{R}^n \). Let us define a regular mesh \( T_h \) of \( \Omega \), a finite set of \( \{K_i\}_{i=1}^m \), \( m \in \mathbb{N} \), of non-degenerate polygons such that,

\[
\bar{\Omega} = \bigcup \{\bar{K}_i, i = 1, ..., m\},
\]

\[
K_i \cap K_j = \emptyset \iff i \neq j,
\]

(4.3a)

(4.3b)

and for all \( i, j \in \{1, ..., m\}, i \neq j \), the intersection \( \bar{K}_i \cap \bar{K}_j \) is either \( \emptyset \) or a vertex, edge or face of both \( K_i \) and \( K_j \). For \( n = 2, 3 \), we introduce the reference element \( \hat{K} \) and mapping \( \Phi_K \),

\[
\Phi_K : \hat{K} \rightarrow K
\]

(4.4)

which maps element any arbitrary element \( K \) to the reference element \( \hat{K} \) of same type.

The variational form for the Eqn.(4.2) is derived by multiplying it with the test function \( v \) and integrating over each element \( K \) separately. After using integration by parts on the divergence term, we obtain,

\[
\int_K \frac{\partial u}{\partial t} v \, dx + \int_{\partial K} v \mathbf{n} \cdot \mathbf{f}(u) \, ds - \int_K \nabla v \cdot \mathbf{f}(u) \, dx = 0.
\]

(4.5)

where \( \partial K \) is the boundary of element \( K \) and \( \mathbf{n} \) is outward normal at \( \partial K \). The flux vector \( \mathbf{f}(u) \) in the second term must be evaluated on the boundary of the element where \( u \) may be discontinuous and thus has two possible values; \( u_i \) on the interior of the element under consideration and \( u_e \) on the exterior. To account for this, we replace \( \mathbf{f}(u) \) with the numerical flux function \( \hat{\mathbf{f}}(u_i, u_e) \), which can be computed using approximated Riemann solvers and is topic of
Section 4.3. The final variational form is derived by using integration by parts once more on the third term to eliminate the gradient of the test function,

\[
\frac{\partial}{\partial t} \int_K uv \, dx + \int_{\partial K} v \left( \hat{f}(u_i, u_e) - f(u_i) \right) \cdot n \, ds + \int_K \nabla \cdot f(u) v \, dx = 0.
\] (4.6)

4.2.2 Basis Functions

The RKDG method we use is implemented in the Nektar code, developed by Karniadakis et al. (see [26]). The original code has been extended to include Runge-Kutta time stepping, slope limiters and accurate Riemann solvers, among other features. In the DG discretization, functions are approximated by using basis functions:

\[
f = \sum_i a_i \phi_i
\] (4.7)

where \( \phi_i \)'s are simple functions e.g polynomials. These functions are chosen in a way so that the whole algorithm is computationally efficient. The set of polynomial basis functions used in Nektar was proposed by Dubiner in [27] for two dimensions and extended to three dimensions in [26]. They are based on the tensor product of one dimensional basis functions which are derived using Jacobi polynomials. Here we describe two and three dimensional basis functions.

Two dimensional Basis functions

The one dimensional basis function are defined on intervals bounded with constant limits, therefore a implicit assumption on the tensor product basis functions for higher dimension is that coordinates in two and three dimensional regions are bounded between constant limits. But in two or three dimension thats not the case in general, e.g. triangle. To overcome this difficulty we define a collapsed coordinate system for two and three dimensions which maps elements without this property to the element bounded by constant limits, e.g. quadrilateral. Reference quadrilateral element is given by,

\[
\hat{Q}_2 = \{ (\xi_1, \xi_2) \mid -1 \leq \xi_1, \xi_2 \leq 1 \}
\]
Figure 4.1: Reference element local coordinate system.

is bounded by the constant limits, however a reference triangle element

\[ \hat{T}_2 = \{ (\xi_1, \xi_2) \mid -1 \leq \xi_1, \xi_2, \xi_1 + \xi_2 \leq 0 \} \]

is not bounded by the constant limits. A suitable coordinate system, describing the triangular region between constant independent limits, is given by the transformation,

\[
\begin{align*}
\eta_1 &= \frac{2(1 + \xi_1)}{1 - \xi_2} - 1 \\
\eta_2 &= \xi_2
\end{align*}
\]

whose inverse transformation is,

\[
\begin{align*}
\xi_1 &= \frac{(1 + \eta_1)(1 - \eta_2)}{2} - 1 \\
\xi_2 &= \eta_2
\end{align*}
\]

Under this local transformation \( \hat{T} \) is,

\[ \hat{T}_2 = \{ (\eta_1, \eta_2) \mid -1 \leq \eta_1, \eta_2 \leq 1 \} \]

So, (4.8) are the suitable collapsed coordinate system for the triangular region. Using these collapsed coordinate system we now define the basis functions for two dimensional elements.

The two-dimensional basis functions have the form,

\[ \phi_{pq}(\xi_1, \xi_2) = \psi_p^a(\xi_1)\psi_q^b(\xi_2), \]
for quadrilaterals and,

$$\phi_{pq}(\xi_1, \xi_2) = \psi^a_p(\eta_1)\psi^b_{pq}(\eta_2),$$

for triangles. The Cartesian coordinates \(\xi_1, \xi_2\) are defined on the reference element as shown in Fig. 4.1, while \(\eta_1, \eta_2\) are collapsed coordinate system and are given by Eqn. (4.8). The functions \(\psi^a_p(z)\) and \(\psi^b_{pq}(z)\) are,

$$\psi^a_p(z) = P^a_0(z), \quad \psi^b_{pq}(z) = \left(\frac{1 - z}{2}\right)^p P^{2p+1,0}_q(z) \quad (4.10)$$

where \(P^{\alpha,\beta}_n\) is the \(n\)th-order Jacobi polynomial with weights \(\alpha\) and \(\beta\).

### Three dimensional Basis functions

<table>
<thead>
<tr>
<th>Element Type</th>
<th>Upper Limits</th>
<th>Local Collapsed Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hexahedron</td>
<td>(\xi_1, \xi_2, \xi_3 \leq 1)</td>
<td>(\xi_1) \quad \xi_2 \quad \xi_3</td>
</tr>
<tr>
<td>Prism</td>
<td>(\xi_1 \leq 1, \xi_2 + \xi_3 \leq 0)</td>
<td>(\bar{\eta}_1 = \frac{2(1+\xi_1)}{1-\xi_2} - 1) \quad \xi_2 \quad \xi_3</td>
</tr>
<tr>
<td>Pyramid</td>
<td>(\xi_1 + \xi_3, \xi_2 + \xi_3 \leq 0)</td>
<td>(\bar{\eta}_1 = \frac{2(1+\xi_1)}{1-\xi_2} - 1) \quad \eta_2 = \frac{2(1+\xi_2)}{1-\xi_2} - 1 \quad \eta_3 = \xi_3</td>
</tr>
<tr>
<td>Tetrahedron</td>
<td>(\xi_1 + \xi_2 + \xi_3 \leq -1)</td>
<td>(\eta_1 = \frac{2(1+\xi_1)}{-\xi_2-\xi_3} - 1) \quad \eta_2 = \frac{2(1+\xi_2)}{1-\xi_2} - 1 \quad \eta_3 = \xi_3</td>
</tr>
</tbody>
</table>

Table 4.1: Local collapsed coordinates for three dimensional elements.

Similar to the two dimensional case, for three dimensional elements that are not bounded by the constant limits, we introduce transformations to the local collapsed coordinates. These coordinates for various type of elements are given in Table 4.1. Under these transformed coordinates, three dimensional
4.2 Runge-Kutta Discontinuous Galerkin method for ideal MHD

elements are bounded by constant limits. For example tetrahedron $\hat{T}_3$ which in Cartesian coordinates is given by,

$$\hat{T}_3 = \{-1 \leq \xi_1, \xi_2, \xi_3, \xi_1 + \xi_2 + \xi_3 \leq -1\}$$

is transformed to,$s$

$$\hat{T}_3 = \{-1 \leq \eta_1, \eta_2, \eta_3, \leq 1\}$$

in local collapsed coordinates. For the three dimensional basis function we define function,

$$\psi_{pqr}^c(z) = \left(\frac{1-z}{2}\right)^{p+q} P_r^{2p+2q+2,0}(z). \quad (4.11)$$

Then using local collapsed coordinates the three dimensional basis functions for various elements are given in Table 4.2

<table>
<thead>
<tr>
<th>Hexahedron Basis</th>
<th>$\phi_{pqr}(\xi_1, \xi_2, \xi_3) = \psi_p^a(\xi_1)\psi_q^a(\xi_2)\psi_r^a(\xi_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prism Basis</td>
<td>$\phi_{pqr}(\xi_1, \xi_2, \xi_3) = \psi_p^a(\bar{\eta}_1)\psi_q^a(\xi_2)\psi_r^b(\xi_3)$</td>
</tr>
<tr>
<td>Pyramid Basis</td>
<td>$\phi_{pqr}(\xi_1, \xi_2, \xi_3) = \psi_p^a(\bar{\eta}<em>1)\psi_q^a(\eta_2)\psi^c</em>{pqr}(\eta_3)$</td>
</tr>
<tr>
<td>Tetrahedron Basis</td>
<td>$\phi_{pqr}(\xi_1, \xi_2, \xi_3) = \psi_p^a(\eta_1)\psi_{pq}^b(\eta_2)\psi^c_{pqr}(\eta_3)$</td>
</tr>
</tbody>
</table>

Table 4.2: Basis functions for Three dimensional Elements

These basis functions are orthogonal in the Legendre inner product over each element, resulting in a diagonal mass matrix. The functions are polynomial in both the Cartesian and non-Cartesian co-ordinates. It was proved in [28] that the coefficients of the basis functions in a solution decay exponentially with polynomial order, thus the numerical solution converges exponentially as the maximum polynomial order of the approximation is increased.
4 Discontinuous Galerkin Methods for Magnetohydrodynamics

4.2.3 Approximation of integrals

To evaluate the integrals in Eqn.(4.6) we first use the mapping $\Phi_K$ to transform the integral on a element to the reference element of the same type, i.e.

$$\int_K f(x_1, x_2, x_3)dx_1dx_2dx_3 = \int_K f(\xi_1, \xi_2, \xi_3)|J_K|d\xi_1d\xi_2d\xi_3,$$

(4.12)

where $J_K$ is jacobian for the change of variable introduced by mapping $\Phi_K$. On the reference element we use the local collapsed coordinates, to transform integral in to the integral with constant limits, using

$$\int_K f(\xi_1, \xi_2, \xi_3)|J_K|d\xi_1d\xi_2d\xi_3 = \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} f(\eta_1, \eta_2, \eta_3)|J_K||J_c|d\eta_1d\eta_2d\eta_3$$

(4.13)

where $J_c$ is jacobian of the mapping used for the local collapsed coordinates.

To compute the integral we Gauss-Legendre type quadrature formula in each variable which has following form for three dimensional integrals,

$$\int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} f(\eta_1, \eta_2, \eta_3)|J_K||J_c|d\eta_1d\eta_2d\eta_3$$

(4.14)

$$\approx \sum_{i=0}^{N_1-1} \left\{ w_i \sum_{j=0}^{N_2-1} \left\{ w_j \sum_{k=0}^{N_3-1} w_k(f|J_K||J_c|(\eta_{1i}, \eta_{2j}, \eta_{3k})) \right\} \right\}$$

Here $w_m$’s are weights and $\eta_{mn}$’ are the quadrature points. Note that after expansion in to basis functions, all the integrals are just integrals of polynomials. Although any type of Gauss-Legendre quadrature formula can be used, we prefer the one which include the end points as in this case boundary conditions can be easily imposed.

Using these approximations we can rewrite Eqn.(4.6) as a system of semi-discrete ODEs,

$$\frac{du_h}{dt} = L_h(u_h).$$

(4.15)
4.2 Runge-Kutta Discontinuous Galerkin method for ideal MHD

<table>
<thead>
<tr>
<th>order</th>
<th>$\alpha_{il}$</th>
<th>$\beta_{il}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>1/2 1/2</td>
</tr>
<tr>
<td></td>
<td>1/2 1/2</td>
<td>0 1/2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3/4 1/4</td>
</tr>
<tr>
<td></td>
<td>1/3 0 2/3</td>
<td>0 0 2/3</td>
</tr>
</tbody>
</table>

Table 4.3: Parameters for Runge-Kutta time marching schemes.

4.2.4 Runge-Kutta Time Discretization

To advance solutions in time, the RKDG method uses a Runge-Kutta (RK) time marching scheme. Here we present the second-, third- and fourth-order accurate RKDG schemes. For second- and third-order simulations, we present the TVD RK schemes of Shu (see [29]). For fourth-order simulations we use the classic scheme.

Consider the semi-discrete ODE,

$$\frac{du_h}{dt} = L_h(u_h).$$

Let $u_h^n$ be the discrete solution at time $t^n$, and let $\Delta t^n = t^{n+1} - t^n$. In order to advance a numerical solution from time $t^n$ to $t^{n+1}$, the RK algorithm is as follows:

1. Set $u_h^{(0)} = u^n_h$.
2. For $i = 1, ..., k + 1$, compute,

   $$u_h^{(i)} = \sum_{l=0}^{i-1} \alpha_{il} u_h^{(l)} + \beta_{il} \Delta t^n L_h(u_h^{(l)}).$$

3. Set $u_h^{n+1} = u_h^{(k+1)}$. 
The values of the coefficients used are shown in Table 4.3. For the linear advection equation, it was proved by Cockburn et al. in [30] that the RKDG method is $L^\infty$-stable for piecewise linear ($k = 1$) approximate solutions if a second-order RK scheme is used with a time-step satisfying,

$$c \frac{\Delta t}{\Delta x} \leq \frac{1}{3},$$

where $c$ is the constant advection speed. The numerical experiments in [42] show that when approximate solutions of polynomial degree $k$ are used, an order $k + 1$ RK scheme must be used, which simply corresponds to matching the temporal and spatial accuracy of the RKDG scheme. In this case the $L^\infty$-stability condition is

$$c \frac{\Delta t}{\Delta x} \leq \frac{1}{2k + 1}.$$

For the nonlinear case, the same stability conditions are used but with $c$ replaced by the maximum eigenvalue of the system.

### 4.3 MHD Riemann Solvers

Riemann solvers are essential components in finite volume (FV) or discontinuous Galerkin (DG) methods. The function of a Riemann solver is to compute the flux across an interface that initially separates two uniform states:

$$U(x, 0) = \begin{cases} U_l, & x \leq x_0 \\ U_r, & x > x_0 \end{cases}$$

where $x_0 \in \mathbb{R}$. Such problems arise at the cell boundaries of a computational grid at every timestep. In recent years a wide variety of Riemann solvers have been developed for ideal MHD (see e.g. [63, 31, 32, 33]). In this section we will give an overview of the various Riemann solvers we have implemented in Nektar code. We use these solvers for various test cases to find an optimal Riemann Solvers in Chapter 6.
4.3 MHD Riemann Solvers

Lax-Friedrichs Flux

The Lax-Friedrichs flux is a simple numerical flux that is used in many RKDG methods. Only the exact flux $F$ must be known to compute this flux:

$$g_{lf}(U_l, U_r) = \frac{1}{2}(F(U_l) + F(U_r)) - \frac{\Delta x}{2\Delta t}(U_r - U_l). \quad (4.16)$$

Roe Solver

The Roe flux is computed from a local linearization of the system about what is known as the Roe average state $\hat{U}$. The Roe average state should be selected such that the Roe matrix $A(U_l, U_r) \equiv DF(\hat{U}(U_l, U_r))$ satisfies:

$$F(U_l) - F(U_r) = A(U_l, U_r)(U_l - U_r).$$

The Roe flux is then given by:

$$g_{roe}(U_l, U_r) = \frac{1}{2}(F(U_l) + F(U_r)) - \frac{1}{2}\Sigma j=1^n |\lambda_j(\hat{U})|\alpha_j r_j(\hat{U}) \quad (4.17)$$

where $\alpha_j = l_j(\hat{U})(U_r - U_l)$. The eigenvalues and left and right eigenvectors of the ideal MHD system are as defined in Section 3.4. For $\gamma = 2$, the Roe average state for the ideal MHD equations is given in [64],

$$\hat{\rho} := \sqrt{\rho_l \rho_r}, \quad \hat{u}_i := \frac{\sqrt{\rho_l} u_{il} + \sqrt{\rho_r} u_{ir}}{\sqrt{\rho_l} + \sqrt{\rho_r}}, \quad \hat{H} := \frac{\sqrt{\rho_l} H_l + \sqrt{\rho_r} H_r}{\sqrt{\rho_l} + \sqrt{\rho_r}}, \quad \hat{B}_i = \frac{B_{il}}{\sqrt{\rho_l}} + \frac{B_{ir}}{\sqrt{\rho_r}}$$

For general $\gamma$, construction of the Roe average state is much more complicated and has been pursued, for example, in [34] and [35]. In practice the arithmetic average state is used instead of the Roe average state for $\gamma \neq 2$, which has been shown to work well [36].

It must be noted that there are issues with the robustness of the Roe solver. It has been shown that any linearization of certain Riemann problems for the
4 Discontinuous Galerkin Methods for Magnetohydrodynamics

Euler equations result in negative densities and pressures, thus no linearized Riemann solver is guaranteed to be positivity preserving for the Euler equations [37]. For the MHD equations, linearized Riemann solvers become even more problematic as negative pressure may be produced by numerical error in the magnetic energy as well as the kinetic energy [33].

A further complication with the Roe solver arises if we choose to use the method of [38] to control divergence errors in the magnetic field. In this approach the MHD system is augmented with a RHS proportional to $\nabla \cdot B$ in order to minimize divergence errors:

$$\frac{\partial U}{\partial t} + \nabla \cdot F = S \rightarrow \frac{\partial U}{\partial t} + A \cdot \nabla U = 0.$$ 

A Roe solver is then usually constructed based on the eigensystem of $A$. However, the usual expression for the Roe flux is no longer appropriate, as $A \cdot \nabla U \neq \nabla \cdot F$ since $A \cdot \nabla U$ includes the source term. One approach to make the Roe solver consistent is to first decompose the jump in conserved variables into eigenmodes,

$$\alpha_j = \lambda_j(U_R - U_L),$$

then compute the upwind solution to the linearized Riemann problem using,

$$U^* = \frac{1}{2}[U_l + U_r - \sum_{j=1}^{8} \text{sign}(\lambda_j)\alpha_j r_j].$$

Finally, we obtain the numerical flux by substituting $U^*$ into the exact expression for the flux:

$$\hat{F} = F(U^*).$$

HLLE Solver

Across any wave propagating with the velocity $b$ and separating states $U^*$ and $U$, the Rankine-Hugoniot (RH) relations hold,

$$F^* - F = b(U^* - U) \quad (4.18)$$

The HLL Riemann solver [39] is constructed by assuming a solution that consists of a single intermediate state $U^*$ between the fastest and slowest
4.3 MHD Riemann Solvers

(most negative) waves. An expression for the intermediate state can be easily be derived from the RH relations,

\[ U_{hll} = U^* = \frac{b_r U_r - b_l U_l - F_r + F_l}{b_r - b_l}, \] (4.19)

where \( b_l \) and \( b_r \) are approximations of the minimum and maximum signal speeds, respectively. Applying the RH relations across the entire Riemann fan gives an expression for the intermediate flux, which is the HLL flux,

\[ g_{HLLE}(U_l, U_r) = \frac{b_r F(U_l) - b_l f F(U_r)}{b_r - b_l} + \frac{b_r b_l}{b_r - b_l} (U_r - U_l). \] (4.20)

One highly effective choice of signal speed estimates is due to [37],

\[ b_l = \min_i (\lambda_i(U_l), \lambda_i(U_m), 0) \quad \text{and} \quad b_r = \max_i (\lambda_i(U_r), \lambda_i(U_m), 0) \]

where \( U_m \) is the Roe mean value. This choice of signal speeds is used, the solver is referred to as the HLLE solver. The HLLE solver is extremely robust as it is both positivity preserving [37] and satisfies and entropy inequality automatically [40]. However, due to the single intermediate state approximation, the HLLE solver cannot resolve non-fast isolated discontinuities, making it quite dissipative.

**HLLEM Solver**

The MHD HLLEM solver was developed by Wesenberg [63]. In this solver, an anti-diffusion term is added to the usual HLLE flux:

\[ g_{HLLEM} := g_{HLLE}(U_l, U_r) + a_0(U_l, U_r), \] \hspace{1cm} (4.21)

where,

\[ a_0(U_l, U_r) = -\frac{b_r^+ b_l^-}{b_r^+ - b_l^-} \sum_{i=2}^6 \delta_i \alpha_i r_i. \]

Here,

\[ \delta_2 := \delta_6 := \delta_a, \quad \delta_3 := \delta_5 := \delta_s, \quad \delta_4 := \delta_c, \]

55
where,
\[ \delta_e := \phi_1 \frac{v_f(v_m)}{v_f(v_m) + |u_x(v_m)|}, \]
\[ \delta_s := \phi_1 \frac{v_f(v_m)}{v_f(v_m) + (1 - \phi_2)|u_x(v_m)| + v_s(v_m)}, \]
\[ \delta_e := \phi_1 \frac{v_f(v_m)}{v_f(v_m) + (1 - \phi_2)|u_x(v_m)| + v_{ax}(v_m)}, \]
\[ \phi_1 := \Phi(v_f(v_m) - v_{ax}(v_m)), \]
\[ \phi_2 := \Phi(B_x^2 + B_y^2(v_m) + B_z^2(v_m)), \]
and,
\[ \Phi(x) = \begin{cases} 
0, & \text{for } (x \in (-\infty, \epsilon)), \\
\frac{x-\epsilon}{\delta-\epsilon}, & \text{for } (x \in [\epsilon, \delta)), \\
1, & \text{for } (x \in [\delta, \infty)).
\end{cases} \]

**HLLC Solvers**

The HLLC family of solvers are derived by assuming an approximate solution to the Riemann problem in which the Riemann fan is separated into two intermediate states, \( U_l^* \) and \( U_r^* \), by a contact discontinuity. The normal velocity is assumed constant within the Riemann fan and is given by the HLL average (see Eq. 4.19),
\[ u_{x_l}^* = u_{x_r}^* = \frac{(\rho u_{x})_{hll}}{\rho_{hll}} \equiv u^* \]  
(4.22)

The standard procedure in a HLLC solver is to then solve for the remainder of the intermediate states using the RH relations across each wave. For intermediate total pressure and densities, this yields,
\[ p_{T_l}^* = p_{T_r}^* = p_{T_l} + \rho_l(b_l - u_{x_l})(u^* - u_{x_l}) \equiv p_T^*, \]  
(4.23a)
\[ \rho_{\alpha}^* = \rho_{\alpha} b_{\alpha} - u_{x_{\alpha}} \]  
(4.23b)

where \( \alpha \) can be \( l \) or \( r \). Gurski [32] observed that when \( B_x \neq 0 \), the transverse magnetic field and velocity must be continuous across the contact (and therefore the entire Riemann fan), in addition to the total pressure and normal
velocity. These are then set to the HLL average,

\[ u^*_{yl} = u^*_{yr} = \frac{\rho u_y}{\rho_{hl}} \equiv u^*_y, \] (4.24a)

\[ u^*_{zl} = u^*_{zr} = \frac{\rho u_z}{\rho_{hl}} \equiv u^*_z, \] (4.24b)

\[ B^*_{yl} = B^*_{yr} = B_{yhll} \equiv B^*_y, \] (4.24c)

\[ B^*_{zl} = B^*_{zr} = B_{zhll} \equiv B^*_z. \] (4.24d)

The following expression for the total energies of the intermediate states can then be derived from the RH relations:

\[ e^*_\alpha = \left( b_\alpha - u_{x\alpha} \right) e_\alpha - p r_\alpha u_{x\alpha} + p^*_T u^* + B_x (u_\alpha \cdot B_\alpha - u^* \cdot B^*) \] (4.25)

With the intermediate states completely determined, the intermediate fluxes \( F^*_\alpha \) are computed from the RH relations across the left and right waves. The numerical flux is then given by,

\[
g_{hl} = \begin{cases} 
F_l & \text{if } b_l > 0, \\
F^*_l & \text{if } b_l \leq 0 \leq u_s, \\
F^*_r & \text{if } u_s \leq 0 \leq b_r, \\
F_r & \text{if } b_r < 0.
\end{cases}
\] (4.26)

This is referred to as the HLLC-G flux. Note that this flux does not tend to the Euler HLLC flux in the limit of vanishing magnetic field. An alternative MHD HLLC solver was developed independently in [31], which is referred to as the HLLC-L solver. In this solver the following forms are used for the intermediate tangential velocities:

\[ u^*_{y\alpha} = u_{y\alpha} + \frac{B_x (B_{y\alpha} - B^*_y)}{\rho_\alpha (u^* - u_{x\alpha})} \] (4.27a)

\[ u^*_{z\alpha} = u_{z\alpha} + \frac{B_x (B_{z\alpha} - B^*_z)}{\rho_\alpha (u^* - u_{x\alpha})}. \] (4.27b)

These satisfy the RH relations over the entire Riemann fan, but not necessarily over the contact alone. This scheme does, however, have the property that it tends to the Euler HLLC flux in the limit of zero magnetic field(see [33]).
In order to resolve more discontinuities exactly, Miyoshi et al. in [33] formulated a more complicated HLL type MHD Riemann solver in which the Riemann fan is assumed to consist of four intermediate states, $U_l^*, U_l^{**}, U_r^{**},$ and $U_r^*$, from left to right. These states are separated by an Alfvén wave, followed by the contact discontinuity, then another Alfvén wave, which have velocities $b_l^*, u^*, b_r^*$, respectively. Similarly to the HLLC solver, the normal velocity is assumed constant throughout the Riemann fan and is given by the HLL average, which is,

$$u^* = \frac{(b_r - u_{xr})\rho_r u_{xr} - (b_l - u_{xl})\rho_l u_{xl} - p_{Tr} + p_{Tl}}{(b_r - u_{xr})\rho_r - (b_l - u_{xl})\rho_l}. \quad (4.28)$$

The constant total pressure within the Riemann fan and $\rho_\alpha^*$ are given by Eq. (4.23b), while the following expressions for the remaining variables in $U_\alpha^*$ can be derived from the RH relations across the outermost waves:

$$u_{y\alpha}^* = u_{y\alpha} - B_x B_{y\alpha} \rho_\alpha (b_\alpha - u_{x\alpha})(b_\alpha - u^*) - B^2_x$$  \hspace{1cm} (4.29a)

$$B_{y\alpha}^* = B_{y\alpha} - B_x \rho_\alpha (b_\alpha - u_{x\alpha})(b_\alpha - u^*) - B^2_x$$  \hspace{1cm} (4.29b)

$$u_{z\alpha}^* = u_{z\alpha} - B_x B_{z\alpha} \rho_\alpha (b_\alpha - u_{x\alpha})(b_\alpha - u^*) - B^2_x$$  \hspace{1cm} (4.29c)

$$B_{z\alpha}^* = B_{z\alpha} - B_x \rho_\alpha (b_\alpha - u_{x\alpha})(b_\alpha - u^*) - B^2_x$$  \hspace{1cm} (4.29d)

$$e_\alpha^* = \frac{(b_\alpha - u_{x\alpha}) e_\alpha - p_{T\alpha} u_{x\alpha} + p_{T} u^* + B_x (u_\alpha B_\alpha - u^* B^* - B^2_x (b_\alpha - u^*))}{b_\alpha - u^*}. \quad (4.29e)$$

The inner states, $U_\alpha^{**}$, are then computed from the RH relations for the Alfvén waves, which have propagation velocities,

$$b_l^* = u^* - \frac{|B_x|}{\sqrt{\rho_l}}, \quad b_r^* = u^* + \frac{|B_x|}{\sqrt{\rho_r}}. \quad (4.30)$$

Note that the density and total pressure are continuous across the Alfvén waves, while the magnetic field, velocity and total pressure are continuous.
4.3 MHD Riemann Solvers

across the contact, hence,

\[ \rho_\alpha^{**} = \rho_\alpha^* , \]  
\[ p_{T\alpha}^{**} = p_{T\alpha}^* , \]  
\[ u_{y\alpha}^{**} = u_{y\alpha}^* \equiv u_{y}^{**} , \]  
\[ u_{z\alpha}^{**} = u_{z\alpha}^* \equiv u_{z}^{**} , \]  
\[ B_{y\alpha}^{**} = B_{y\alpha}^* \equiv B_{y}^{**} , \]  
\[ B_{z\alpha}^{**} = B_{z\alpha}^* \equiv B_{z}^{**} . \]  

Miyoshi et al.\(^{[33]}\) obtained the following expressions for the remaining variables:

\[ u_{y}^{**} = \frac{\sqrt{\rho_\alpha^* u_{y\alpha}^* + \sqrt{\rho_\alpha^* u_{z\alpha}^*}} + (B_{y\alpha}^* - B_{y\alpha}^*) \text{sign}(B_x)}{\sqrt{\rho_\alpha^* + \sqrt{\rho_\alpha^*}}} , \]  
\[ u_{z}^{**} = \frac{\sqrt{\rho_\alpha^* u_{z\alpha}^* + \sqrt{\rho_\alpha^* u_{z\alpha}^*}} + (B_{z\alpha}^* - B_{z\alpha}^*) \text{sign}(B_x)}{\sqrt{\rho_\alpha^* + \sqrt{\rho_\alpha^*}}} , \]  
\[ B_{y}^{**} = \frac{\sqrt{\rho_\alpha^* B_{y\alpha}^* + \sqrt{\rho_\alpha^* B_{y\alpha}^*}} + \sqrt{\rho_\alpha^* \rho_\alpha^* (u_{z\alpha}^* - u_{y\alpha}^*) \text{sign}(B_x)}}{\sqrt{\rho_\alpha^* + \sqrt{\rho_\alpha^*}}} , \]  
\[ B_{z}^{**} = \frac{\sqrt{\rho_\alpha^* B_{z\alpha}^* + \sqrt{\rho_\alpha^* B_{z\alpha}^*}} + \sqrt{\rho_\alpha^* \rho_\alpha^* (u_{z\alpha}^* - u_{y\alpha}^*) \text{sign}(B_x)}}{\sqrt{\rho_\alpha^* + \sqrt{\rho_\alpha^*}}} , \]  
\[ e_\alpha^{**} = e_\alpha^{**} \pm \sqrt{\rho_\alpha^*} (u_{\alpha}^{**} \cdot B_{\alpha}^{**} - u_{\alpha}^* \cdot B_{\alpha}^*) \text{sign}(B_x) . \]  

The intermediate fluxes can then be obtained from the RH relations summed over multiple waves, for example,

\[ F_l^* = F_l + b_l(U_l^* - U_l) , \]  
\[ F_l^{**} = F_l + b_l^* U_l^{**} - (b_l^* - b_l)U_l^* - b_l U_l . \]  

In general the HLLD flux is given by,

\[ g_{\text{HLLD}} = \begin{cases} 
F_l & \text{if } b_l > 0 , \\
F_l^* & \text{if } b_l \leq 0 \leq b_l^* , \\
F_l^{**} & \text{if } b_l^* \leq 0 \leq u^* , \\
F_r^{**} & \text{if } u^* \leq 0 \leq b_r^* , \\
F_r^* & \text{if } b_r^* \leq 0 \leq b_r , \\
F_r & \text{if } b_r < 0 . 
\end{cases} \]
This solver resolves isolated rotational discontinuities exactly, along with contact discontinuities, and reduces to the Euler HLLC solver when $B_x = 0$ (see [33]). In [33] it is also proved that the HLLD solver is positivity preserving if

$$b_l < u_{xl} - \sqrt{\frac{\gamma - 1}{2\gamma} c_{fl}}, \quad b_r > u_{xr} + \sqrt{\frac{\gamma - 1}{2\gamma} c_{fr}},$$

which are clearly satisfied by the choice of signal speeds from the HLLE method.

### 4.4 Limiters

In order for the RKDG method to be stable in the nonlinear case, we require an entropy inequality and the uniform boundedness of the total variation of the discrete solution $u_h$. In general, a limiter function is required for the second condition to hold. We present the limiter functions used in the RKDG method in the context of a one-dimensional nonlinear scalar equation. In order for the total variation of the means to be nonincreasing for a forward Euler timestep, the discrete solution must satisfy (see [42] for details):

\[
\text{sign}(\bar{u}_{j+1} - \bar{u}_j) = \text{sign}(p_{j+1/2}(u_h|_{K_{j+1}}) - p_{j+1/2}(u_h|_{K_j})), \quad (4.36a)
\]

\[
\text{sign}(\bar{u}_j - \bar{u}_{j-1}) = \text{sign}(u^-_{j+1/2} - u^-_{j-1/2}), \quad (4.36b)
\]

\[
\text{sign}(\bar{u}_{j+1} - \bar{u}_j) = \text{sign}(u^+_{j+1/2} - u^+_{j-1/2}), \quad (4.36c)
\]

where $\bar{u}_j$ is the average solution on element $K_j$, $u^-_{j+1/2}$ denotes the limit of the solution at edge $j + 1/2$ taken from the left, $u^+_{j+1/2}$ denotes the limit of the solution at edge $j + 1/2$ taken from the right and,

\[
p_{j+1/2}(u_h|_{K_{j+1}}) = \bar{u}_j - \frac{\Delta t}{\Delta j_{j+1}} f^+(u^-_{j+1/2}) + \frac{\Delta t}{\Delta j_j} f^-(u^+_{j-1/2}).
\]

Here, $\Delta t$ is the forward Euler timestep, $\Delta j_j$ is the size of $E_j$ and we assume that the flux can be written in the form,

\[
F(a, b) = f^+(a) + f^-(b).
\]

There is no guarantee that the numerical solution will satisfy Eqn. (4.36a)-Eqn. (4.36c), thus it is necessary to enforce them by means of what Cockburn in
4.4 Limiters

[42] refers to as a generalized slope limiter. These limiters alter the numerical solution on an element in such a way that Eqn. (4.36a)-Eqn. (4.36c) are satisfied, while preserving the element average of the solution and, as far as possible, the accuracy of the method.

We will first introduce TVDM generalized slope limiters for the one-dimensional case, then show how these are modified to obtain schemes which are total variation bounded in the means (TVBM). We will then present the extension of these limiters to the two-dimensional case.

4.4.1 One-dimensional TVDM generalized slope limiters

A simple set for sufficient conditions on \( u_h \) that imply Eqn. (4.36a)-Eqn. (4.36c) are satisfied can be stated in terms of the minmod function \( m \),

\[
m(a_1, ..., a_n) = \begin{cases} 
  s \min_{1 \leq n \leq N} |a_n|, & \text{if } s = \text{sign}(a_1) = ... = \text{sign}(a_N) \\
  0, & \text{otherwise}
\end{cases}
\]

It can be easily shown (see e.g. [42]) that Eqn. (4.36a)-Eqn. (4.36c) are satisfied if, for all \( j = 1, ..., N \), we have,

\[
\begin{align*}
  &u^-_{j+1/2} - \bar{u}_j = m(u^-_{j+1/2} - \bar{u}_j, \bar{u}_j - \bar{u}_{j-1}, \bar{u}_{j+1} - \bar{u}_j), \quad (4.37a) \\
  &\bar{u}_j - u^+_{j-1/2} = m(\bar{u}_j - u^+_{j-1/2}, \bar{u}_j - \bar{u}_{j-1}, \bar{u}_{j+1} - \bar{u}_j). \quad (4.37b)
\end{align*}
\]

In the following, let \( v_h \) denote the numerical solution prior to limiting, and let \( \Pi_h \) denote the limiter. Thus the limited numerical solution is given by \( u_h = \Pi_h(v_h) \). For the case of piecewise linear approximate solutions,

\[
v_h|_{K_j} = \bar{v}_j + (x - x_j)v'_j, \quad j = 1, ..., N,
\]

we can apply slightly modified versions of the limiter functions from finite volume schemes. The generalized slope limiter based on the minmod limiter, \( \Pi^m_h \), is,

\[
u_h|_{K_j} = \Pi^m_h(v_h|_{K_j}) = \bar{v}_j + (x - x_j)m(v'_j, \frac{\bar{v}_{j+1} - \bar{v}_j}{\Delta_j}, \frac{\bar{v}_j - \bar{v}_{j-1}}{\Delta_j}). \quad (4.38)
\]
This satisfies Eqn.\((4.37a)\) and Eqn.\((4.37b)\). The modification to the standard minmod limiter is the additional input of \(v_j'\) into the minmod function. The minmod generalized slope limiter is equivalent to,

\[
\begin{align*}
  u_{j+1/2}^- &= \bar{v}_j + m(v_{j+1/2}^- - \bar{v}_j, \frac{\bar{v}_j - \bar{v}_{j-1}}{2}, \frac{\bar{v}_{j+1} - \bar{v}_j}{2}), \\
  u_{j-1/2}^+ &= \bar{v}_j - m(\bar{v}_j - v_{j-1/2}^+, \frac{\bar{v}_j - \bar{v}_{j-1}}{2}, \frac{\bar{v}_{j+1} - \bar{v}_j}{2}).
\end{align*}
\]

(Cockburn proposed the following less restrictive (LR) slope limiter, \(\Pi_{lr}^h\), that also satisfies Eqn.\((4.37a)\) and Eqn.\((4.37b)\):

\[
\begin{align*}
  u_h|_{K_j} = \Pi_{lr}^h(v_h|_{K_j}) = \bar{v}_j + (x - x_j)m(v_j', \frac{\bar{v}_{j+1} - \bar{v}_j}{\Delta_j/2}, \frac{\bar{v}_j - \bar{v}_{j-1}}{\Delta_j/2}).
\end{align*}
\]

Compared to the minmod limiter, the LR limiter permits larger values of the internal gradient on an element before limiting occurs. As limiting decreases the accuracy of the solution on an element, this implies that use of the LR limiter will produce more accurate results. The LR slope limiter is equivalent to,

\[
\begin{align*}
  u_{j+1/2}^- &= \bar{v}_j + m(v_{j+1/2}^- - \bar{v}_j, \bar{v}_j - \bar{v}_{j-1}, \bar{v}_{j+1} - \bar{v}_j), \\
  u_{j-1/2}^+ &= \bar{v}_j - m(\bar{v}_j - v_{j-1/2}^+, \bar{v}_j - \bar{v}_{j-1}, \bar{v}_{j+1} - \bar{v}_j).
\end{align*}
\]

Both the minmod and the LR limiter are defined for piecewise linear solutions. They cannot be directly applied to solutions of higher polynomial order as \(v_h'|_{K_j}\) is in general non-constant. Instead, the limiting procedure is broken into two stages: detection and truncation. To detect if an element requires limiting the following procedure is used in [42]:

1. Compute \(u_{j-1/2}^+\) and \(u_{j+1/2}^-\) using either Eqn.\((4.39a)\) and Eqn.\((4.39b)\) (minmod limiter) or Eqn.\((4.41a)\) and Eqn.\((4.41b)\) (LR limiter).

2. If \(u_{j-1/2}^+ = v_{j-1/2}^+\) and \(u_{j+1/2}^- = v_{j+1/2}^-\), no limiting is required so set \(u_h|_{K_j} = v_h|_{K_j}\), otherwise carry out the truncation stage.

In the truncation stage, we set \(u_h|_{K_j}\) equal to a limited lower order polynomial that satisfies Eqn.\((4.37a)\) and Eqn.\((4.37b)\). The simplest way to do this is to limit to a piecewise constant by setting \(u_h|_{K_j} = \bar{v}_j\). We refer to this as the first-order limiter. The second-order limiter results from limiting to a
4.4 Limiters

piecewise linear function. Following [42], this is done by applying Eqn.(4.38) or Eqn.(4.40) to the $P^1$ part of the solution,

$$v^1_h = \sum_{l=0}^{1} \hat{v}_l \phi_l(x).$$

Thus $u_h|_{\Gamma_j} = \Pi^m_h(v^1_h)$ or $u_h|_{\Gamma_j} = \Pi^{\mu}_h(v^1_h)$. It is important to note that this is more accurate than, for example, a direct use (4.40) with $m(v_{j+1/2}^{-}\tilde{v}_j, \tilde{v}_j-v_{j-1/2}^+)$ as the first argument to the minmod function, which uses quantities already computed in the detection phase. The reason for this is that if limiting is needed due to the high-order modes only, then limiting $v^1_h$ will allow the original $P^1$ part of the solution to be preserved, while using the differences computed in the detection step will not.

4.4.2 One-dimensional TVBM generalized slope limiters

At local extrema, the solution is non-monotone thus any limiter based on the minmod function will limit the solution to a piecewise constant no matter how smooth the solution is. This results in a loss in accuracy at local extrema. Following [41], this can be avoided by replacing the minmod function in the limiters with the TVB corrected minmod function $\bar{m}$ which is defined as,

$$\bar{m}(a_1, ..., a_n) = \begin{cases} 
    a_1, & \text{if } |a_1| \leq M \Delta x^2 \\
    m(a_1, ..., a_n), & \text{otherwise},
\end{cases}$$

where $M$ is a constant that is an upper bound for the absolute value of the curvature of the solution at local extrema.

Provided that the value of $M$ is appropriate, the scheme will not suffer from loss of accuracy at local extrema. The penalty for this is that the scheme is no longer TVD. It can, however, be shown to be total variation bounded in the means (TVBM)(see [41]). Moreover, provided the CFL condition is satisfied and coefficients $\alpha_{il}$ from the Runge-Kutta scheme are nonnegative and satisfy,

$$\sum_{l=0}^{i-1} \alpha_{il} = 1,$$
then it can be shown that solutions produced by the RKDG scheme converge in the nonlinear case (see [42]).

4.4.3 Two-dimensional TVBM generalized slope limiters

To extend the generalized slope limiters presented in the previous subsections to the multi-dimensional case, Cockburn in [42] relies on the assumption that spurious oscillations are present in $u_h$ only if they are present in its $P^1$ part, $u^1_h$. Thus if,

$$u^1_h = \Pi_h u^1_h, \quad (4.42)$$

indicating that limiting is not required for $u^1_h$, then it is assumed that no limiting is required for the full solution $u_h$, hence,

$$\Pi_h u_h = u_h$$

Conversely, if limiting is required for $u^1_h$, the solution is truncated to the limited $P^1$ part,

$$\Pi_h u_h = \Pi_h u^1_h.$$  

This approach has the drawback that it does not reduce to the one-dimensional scheme if the multi-dimensional scheme is applied to simulate a one-dimensional problem. In fact Eqn.(4.37a) and Eqn.(4.37b) are not guaranteed to be satisfied if only the $P^1$ part of the solution is used to detect where limiting is required.

The multi-dimensional limiter that we propose for quadrilateral discretizations uses Eqn.(4.41a) and Eqn.(4.41b) in each direction to detect whether limiting is required, with the exception that the point values at edges are replaced by edge average values. This has the advantage that it reduces to the one-dimensional scheme if it is used to simulate a one-dimensional problem. In detail, we compute,

$$\bar{u}^-_{i,j+1/2} = \bar{v}_{i,j} + \bar{m}(\bar{v}_{i,j+1/2} - \bar{v}_{i,j}, \bar{v}_{i,j} - \bar{v}_{i,j-1}, \bar{v}_{i,j+1} - \bar{v}_{i,j}), \quad (4.43a)$$

$$\bar{u}^+_{i,j-1/2} = \bar{v}_{i,j} - \bar{m}(\bar{v}_{i,j} - \bar{v}^+_{i,j-1/2}, \bar{v}_{i,j} - \bar{v}_{i,j-1}, \bar{v}_{i,j+1} - \bar{v}_{i,j}), \quad (4.43b)$$

$$\bar{u}_{i+1/2,j} = \bar{v}_{i,j} + \bar{m}(\bar{v}_{i+1/2,j} - \bar{v}_{i,j}, \bar{v}_{i,j} - \bar{v}_{i-1,j}, \bar{v}_{i+1,j} - \bar{v}_{i,j}), \quad (4.43c)$$

$$\bar{u}^+_{i-1/2,j} = \bar{v}_{i,j} - \bar{m}(\bar{v}_{i,j} - \bar{v}^+_{i-1/2,j}, \bar{v}_{i,j} - \bar{v}_{i-1,j}, \bar{v}_{i+1,j} - \bar{v}_{i,j}). \quad (4.43d)$$
4.5 Divergence free condition

If \( \mathbf{u}_{i,j}^{+} = \mathbf{u}_{i,j}^{+} = \mathbf{u}_{i,j-1}^{+} = \mathbf{u}_{i+1,j}^{-} = \mathbf{u}_{i+1,j}^{-} = \mathbf{u}_{i-1,j}^{-} \), we assume that no limiting is required so,

\[
\mathbf{u}_h|_{K_{i,j}} = \mathbf{v}_h|_{K_{i,j}}.
\]

Otherwise \( \mathbf{u}_h|_{K_{i,j}} \) is set to the limited \( P^1 \) part of \( \mathbf{v}_h|_{K_{i,j}} \). This is computed by setting,

\[
\bar{\mathbf{u}}_{i,j} = \bar{\mathbf{v}}_{i,j},
\]

\[
\mathbf{u}_x = \bar{v}_{x} \left( \bar{v}_{i,j} - \bar{v}_{i,j-1} - \bar{v}_{i,j+1} + \bar{v}_{i,j} \right),
\]

\[
\mathbf{u}_y = \bar{v}_{y} \left( \bar{v}_{i,j} - \bar{v}_{i-1,j} + \bar{v}_{i+1,j} - \bar{v}_{i,j} \right),
\]

for \( K_{i,j} \).

4.5 Divergence free condition

Due to the presence of \( \nabla \cdot \mathbf{B} = 0 \), constraint the equations of ideal MHD do not have strict hyperbolic character. It has been shown in [44] that even a small divergence in the magnetic fields can dramatically change the character of results from numerical simulations. An effective method to solve this problem is proposed by Powell in [43]. The idea is to reformulate the Jacobian matrix to include an 'extra wave' which corresponds to divergence mode with velocity \( \mathbf{v} \).

This amount to adding a source terms to the system of equations on the RHS,

\[
\mathbf{S}_{\text{powell}} = \nabla \cdot \mathbf{B}(0, B_x, B_y, B_z, v_x, v_y, v_z, v \cdot \mathbf{B})^\top
\]

So final equations reads,

\[
\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{F}(\mathbf{U}) + \mathbf{F}^{\text{visc}}(\mathbf{U})) = \mathbf{S}_{\text{powell}}
\]
4 Discontinuous Galerkin Methods for Magnetohydrodynamics

4.6 Local Discontinuous Galerkin Method for Viscous Terms

Similar to the case of ideal case it is enough to consider the scalar advection diffusion equation,

\[
\frac{\partial u}{\partial t} + \sum_{1 \leq i \leq n} \frac{\partial}{\partial x_i} \left( f_i(u) - \sum_{i \leq j \leq n} a_{ij} \frac{\partial}{\partial x_j} u \right) = 0, \tag{4.47}
\]

where \( f_i \) is convection flux, \( a_{ij} \) are diffusion coefficient with condition that matrix \((a_{ij})_{ij}\) is symmetric and semi-positive definite. Note that MHD equation can be written in this form component wise (see Section 3.5). We introduce a new variable \( q_l = \sum_{1 \leq j \leq n} \frac{\partial u}{\partial x_j} \). and rewrite the Eqn.(4.47)

\[
\frac{\partial u}{\partial t} + \sum_{1 \leq i \leq n} \frac{\partial}{\partial x_i} \left( f_i(u) - \sum_{i \leq l \leq n} a_{il} q_l \right) = 0, \tag{4.48a}
\]
\[
q_l - \sum_{1 \leq j \leq n} \frac{\partial g_{lj}}{\partial x_j} = 0, \text{ for } l = 1, \cdots, n \tag{4.48b}
\]

where \( g_{lj} = \int_0^u ds = u \). To obtain the LDG method we apply the DG method to Eqns. (4.48). We set \( w = (u, q_1, q_2, \cdots, q_n) \), and introduce flux,

\[
h_i(w) = (f_i(u) - \sum_{1 \leq l \leq n} a_{il} q_l, -g_{l1}, \cdots, -g_{ln}). \tag{4.49}
\]

Multiplying with test function and integrating by parts results in,

\[
\int_K \frac{\partial u}{\partial t} v_u dx - \sum_{1 \leq i \leq n} \int_K h_{iu} \frac{\partial}{\partial x_i} v_u dx + \int_{\partial K} \hat{h}_u(w, n) v_h dx = 0, \tag{4.50a}
\]
\[
\int_K q_l v_{q_l} dx - \sum_{1 \leq j \leq n} \int_K h_{jql} \frac{\partial}{\partial x_j} v_{q_l} dx + \int_{\partial K} \hat{h}_{q_l}(w, n) v_h dx = 0. \tag{4.50b}
\]

This is the variational formulation which we need to approximate. The only remaining part for implementation is choice of the flux \( \hat{h}(w, n) \). This is done by dividing it into two part,

\[
\hat{h} = \hat{h}_{\text{conv}} + \hat{h}_{\text{diff}}
\]
where convective flux is given by,

\[ \hat{h}_{\text{conv}}(w^-, w^+, n) = (\hat{f}(u^+, u^-, n), 0), \]

here \( \hat{f} \) is given by approximated Riemann solvers described in Section 4.3. The diffusion flux is given by the Bassi-Rebay flux (see [45]) i.e. the averages of variables and fluxes across the interface. This leads to a stable and efficient scheme.

### 4.7 Algorithm

Here we describe the RKDG algorithm as it is implemented in Nektar code. A feature of Nektar [28], is that time integration is performed on the solution values at the quadrature points rather than the modal coefficients. The algorithm utilizes two sets of quadrature points on element boundaries: The set of interface quadrature points \( Q^I \), where a Riemann solver is used to compute the numerical fluxes, and the set of edge quadrature points \( Q^E \), where the fluxes must be evaluated to advance the solution in time. \( Q^I \) corresponds to the Gauss quadrature points along each edge, which do not include the vertexes where there are no clearly defined left and right states to be input to the Riemann solver. The set \( Q^E \) depends on the type of elements used and typically includes the vertexes. The basic steps in the RKDG algorithm implemented in Nektar are as follows:

1. Read in the initial condition \( U(x, 0) \) and evaluate it at all quadrature points, set \( n = 0 \).

2. Limit the solution to ensure that the method is TVB (see Section 4.4), if computing for the solutions with shocks.

3. Compute the numerical fluxes \( \hat{F}_n \) at \( Q^I \) using a Riemann solver (see Section 4.3). At the domain boundaries, use the prescribed boundary conditions as the exterior state that is input to the Riemann solver. Interpolate the resulting fluxes onto \( Q^E \). Scale the fluxes with the ratio of the edge Jacobian to the volume Jacobian. This increases the efficiency of the scheme by allowing the fluxes to be added to the volume terms evaluated at the quadrature points (see [28] for details).
4. Compute the exact fluxes $F$ at the element quadrature points. Utilizing the known gradients of the basis functions, $\nabla \cdot F$ is readily computed via a mapping.

5. Subtract $F$ from $\hat{F}_n$ at the appropriate quadrature points and add this to the result of step 4.

6. Determine the modal coefficients of the result of step 5 by evaluating its inner product with each orthogonal basis function. Evaluate the resulting polynomial representation at the quadrature points and store the result in $U_f$.

7. Similarly compute the viscous fluxes (see Section 4.6), if calculating for viscous MHD and add it to $U_f$ by following step 6 for viscous flux.

8. Carry out the first substep of the RK time integration scheme:

$$U^{(1)} = U(x, t^n) + \beta_{10} \Delta t U_f(x, t^n).$$

9. Repeat steps 2 to 6 for the remaining substeps in the RK scheme.

10. Increment $n$ by one and $t^n$ by $\Delta t$. Continue steps 2 to 9 until the desired simulation time is reached.
5 Three Dimensional Nektar Code

5.1 Introduction

In Chapter 4 we have presented the complete algorithm for RKDG discretization of the viscous MHD equations. This algorithms is implemented in Nektar code which was developed at Brown University [47] [46]. The Nektar code can solve the fluid equations (Euler and Navier Stokes) as well as the ideal and real MHD equations on unstructured triangle or quadrilateral meshes for two dimensional domains. Three dimensional version of the code can handle tetrahedral, prismoid, pyramidal and hexahedral meshes. As DG algorithms allow computation on each element separately, a mesh can include mixed elements in both two and three dimensions. The Nektar code has both serial and parallel version based on MPI.

This code is modified extensively for MHD computations. We have added various Riemann solvers (see Section 4.3) for computation of advection flux of MHD equations. For computing the solution with discontinuities, limiters are implemented for two dimensional simulations(see Section 4.4). Time stepping using TVD-RK methods(see Section 4.2.4) is also implemented. Finally code was installed and tested on ETH supercomputing cluster BRUTUS.

In this chapter we present an overview of the Nektar code. A description of the two dimensional code can be found in [53]. Here we are focused on the three dimensional code. In Section 5.2 we briefly describe the source code for Nektar. In Section 5.3 we illustrate the installations and compilations process. Next two sections(Sections 5.4, 5.5) are devoted to the description of generating the input files using GID. Section 5.6 provides a description of the input files. In the last Section (5.7), we describe the post processing routines.
5 Three Dimensional Nektar Code

5.2 Source code Description of Nektar Code

In this section we give a brief description of the source code. The top directory for the Nektar code is Hybrid.michele, which hence forth is the working directory for us. This directory contains all the source code in various subdirectories containing various part of the code. This directory Hybrid.michele contains following subdirectories,

1. LAPACK
2. include
3. Utilities
4. VecLib
5. Hlib
6. gs
7. Flags
8. Compressed3d
9. Compressed2d
10. P_Compressed2d
11. GID

We will now describe each of these subdirectories.

- **LAPACK** This directory contains the standard LAPACK package based on BLAS libraries. See [http://www.netlib.org/lapack/](http://www.netlib.org/lapack/) for the information about LAPACK.

- **include** This directory contains the header files (h-files) used in the Nektar code. These files contains function declarations and data structure definitions used in the code. Some of the important files are,

  - *hotel.h*
    This file contains the data structure definitions and function declarations related to the parallel version of the code. Also see *nstruct.h*.

  - *gs.h*
    This file consist of declarations of the functions used in ./gs directory.
5.2 Source code Description of Nektar Code

- Tet.h
  This file contains declarations of the functions defined in ./Hlib/src/Tet.C. These functions are related to the geometric properties of Tet type elements. Similarly there are other files related to the different type of elements e.g. Tri.h for Tri elements.

- nekcomp.h, nekstruct.h, nektar.h
  These files contain the definitions of the data structures and declarations of the functions used in MHD part of the code(see ./Compress3d/src). These are important files for getting information about the data structures used in MHD simulations. See individual file for more details.

- Utilities
  This directory contains source code for the pre- and post-processing routines in the subdirectory ./Utilities/src. The routines that are useful to us are following,

  - rea2metis.C
    This pre-processing routine is used to generate METIS file from .rea file, used for the partition of a mesh using METIS routines(See Section 5.5), for parallel runs.

  - p2sfl1d.C
    This post-processing routine is used to combine output files from different nodes into a single output file.

  - nek2tec.C
    This routine convert Nektar output file into a Tecplot readable file.

  To compile these routines see Sections 5.5 and 5.7

- Veclib
  This directory contains the routines implemented in the Veclib library. This library is divided functionally into three areas: vector operations, linear systems, and memory management. The linear systems routines are from LINPACK(See http://www.netlib.org/linpack/). For most routines, there are a variety of precisions available. For more specific documentation on an individual routine, see the README file in the directory. For specific function, try checking the source file. Library libvec.a is generated here during compilation process (see Section 5.3 for compilation).

- Hlib
  This directory consists of the make files for various OS, used for generat-
Three Dimensional Nektar Code

The source code for the \texttt{libhybridopt.a} is in subdirectory \\texttt{./Hlib/src}. This subdirectory contains files for various type of elements. e.g. \texttt{Tet.C}. The important files are,

- **Tet.C**
  Operations related to the element type \texttt{Tet} are defined here (see \texttt{Tet.h} in \texttt{./include} for declarations). Similarly there are different files for different type of elements.

- **Element_List.C**
  This file contains the functions needed to map global data to each individual node and vice-versa. The important functions \texttt{set_con_info} is fixed for the bugs. It maps global elements to the local ones. Function \texttt{set_pllinfo} needs to be modify for setting up the input file for the parallel computation(see Section \ref{sec:parallel}). Structure \texttt{pllinfo} contains the information related to the parallel runs.

The library \texttt{libhybridopt.a} is generated in subdirectory \texttt{./Hlib/Linux} during the compilation process.

- **gs** This directory is needed only for the parallel compilation and runs of the \texttt{Nektar} code. It contains functions related to the MPI implementation. See source code for the more details in subdirectory \texttt{./gs/src}. The library \texttt{libgs.a} is generated here during the compilation process.

- **Flags** Contains the make files for setting up the compiler and library details for compilation. We shall describe it in details in next Section \ref{sec:flags}.

- **Compress3d** The subdirectory \texttt{./Compress3d/src} contain the source code for the MHD simulation. Here we provide detailed description of the important files.

  - **MakeNek and Makfile**
    These are the makefiles for compiling the \texttt{Nektar} code and generating executable \texttt{nektarC} (see Section \ref{sec:makefiles}).

  - **drive.C**
    This file contains the top level \texttt{main()} function. In this function data structure \texttt{Omega} (see file \texttt{./include/}) is initialize using function \texttt{PreProcess} (see \texttt{prepost.C}). This includes setting up the initial conditions and dividing the data on different nodes and memory allocation. After initialization the time stepping is implemented...
5.2 Source code Description of Nektar Code

using number of steps and time step size prescribed in the input file filename.rea. During the time stepping, the function check for variable EQTYPE, and if it is MHD(Ideal MHD) or MHD_NS (viscous MHD) then it choose the specified case. Here the time-stepping based on RK-methods(see Section 4.2.4) is implemented. During each sub time- step, the data structure Omega->Us (see file nekstruct.h) is updated using function filter defined in limit.C (only for 2d code in directory ./Compress2d/src, ignored for the three dimensional simulations) that uses the limiters discussed in Section 4.4. The advection fluxes are then updated using function Conv_Step defined in advection.C. Boundary conditions for viscous fluxes are setup using function Set_Dir_BCs_MHD defined in file visc_mhd.C. Viscous fluxes are then, computed using function Diff_Step_MHD_NS defined in file visc_mhd.C. After the time update, function Analyser is called which is defined in the file analyser.c. It checks for the blow up in the data and write the output files at specified time steps.

- prepost.C
  This file contains functions that decide the equation type, read the corresponding initial conditions given in function ReadICs (in file io.C) and allocate memory depending on equation type and the mesh. This is implemented in the function PreProcess, invoked in drive.C. This file also contains functions related to the data exchange needed between different nodes.

- io.C
  The initial conditions are defined in function Read_ICs, invoked in PreProcess. The initial condition case is selected based on the switch defined in here and specified in last part of the input file filename.rea (see Section 5.6).

- ./Compress2d/limit.C
  Limiters discussed in Section 4.4 are implemented in here. They are only available for two dimensional runs.

- advection.C
  This file contains functions related to computation of the advection fluxes for MHD simulations. The top level function is Conv_Step, invoked in drive.C. In this function we invoke Fill_edges used
5 Three Dimensional Nektar Code

to setup the boundary conditions and exchange data on different nodes. The flux terms in Eqn. (4.6) are evaluated using the function \texttt{Upwind\_edge\_flux\_mhd}. In this function Riemann solver functions are called (see Section 4.3), that are implemented for MHD simulations. The remaining terms containing flux are calculated in function \texttt{Calc\_DivF\_mhd}. The Powell’s source term for divergence cleaning (see Section 4.5) is applied by calling function \texttt{add\_powell\_sources}. This file is extensively modified to implement various Riemann solver routines (Also see ./\texttt{Compress2d/src/advection.C} for two dimensional code) and real gas data to calculate gas constant $\gamma(T)$ for $SF_6$ gas (see [53] for the gas data approximation functions implemented in the code).

• \texttt{visc\_mhd.C}
  This file is used to calculate the viscous flux defined in Section 3.5 using the LDG method (see Section 4.6). The function \texttt{Diff\_Step\_MHD\_NS} (called in \texttt{drive.C}) consists of the top level implementation and is extensively modified to calculate various coefficients (e.g. Reynolds number), that depends on the gas data. Other important function in this file is \texttt{Set\_Dir\_BCs\_MHD} used to set the boundary conditions before calculating the viscous fluxes.

• \texttt{anslyser.C}
  The important function is \texttt{Analyser}, called in \texttt{drive.C} and used to look for blowups in the code. This function also generates the output files if specified in \texttt{filename.rea} using IOSTEP argument.

• \texttt{mpinektar.C}
  This file contains the code for exchanging the data of the shared faces (or edges) between various nodes, by changing the information at the edges of the elements. The function used for this is \texttt{exchange\_edges}, called in functions \texttt{Fill\_edges} in \texttt{advection.C} and \texttt{Diff\_Step\_MHD\_NS} in \texttt{visc\_mhd.C}.

• \texttt{SF6.h}
  The file consist of the function declarations for the gas data function defined in \texttt{io.C}, \texttt{advection.C} and \texttt{visc\_mhd.C}. It also contains the reference values of the variables used in these files.

◊ \texttt{Compress2d} and \texttt{P\_Compress2d} Directory \texttt{./Compress2d} contains the source code for the two dimensional serial runs. The files and functions in this
5.3 Installation and Compiling

directory are very similar to that of ./Compress3d directory and have same structure.

Similarly, ./P_Compress2d contains the parallel version of the two dimensional code (For details See [53]).

- **GID** This directory contains file g2n_3d.c and used to convert GID files to .rea input file. Another file BC_g2n_3d.c is used to specify the boundary conditions (see Section 5.4).

5.3 Installation and Compiling

In this section we describe the installation and compilation process for the Nektar code. Currently Nektar is installed on BRUTUS, ETH supercomputing cluster which has 2200 processing cores on 756 nodes. More information about the cluster can be found here [http://clusterwiki.ethz.ch/wiki/index.php/Brutus](http://clusterwiki.ethz.ch/wiki/index.php/Brutus) The code require the following libraries for compilation:

1. **MPI**: Message passing interface for parallel simulations.
2. **acml** AMD math kernel library
3. **math** library
4. **g2c**
5. **blas**
6. **lapack**
7. **fortran** libraries.
8. **atlas**

One can also use standard Intel math libraries for C and FORTRAN. In addition to this, we need C, C++ and FORTRAN compilers. On **BRUTUS** we use intel compilers for **MPI**.

The compiler and libraries path have to be correctly set in the corresponding system file found in Hybrid/Flags. For our case it is Linux.inc. Make sure that all the corresponding libraries and compiler are in your path. Here is an example of file Linux.inc.
5 Three Dimensional Nektar Code

CXX = mpiCC
CC = mpicc
FC = f77
ifdef MPIPRG
  # CFM default rule
  message += CFM default uses MPICH, chp_4, sun4u.
  MPICXX = $(CXX) -I/users/build/msgpass/mpich/include
  MPICC = $(CC) -I/users/build/msgpass/mpich/include
  MPILIB = -L/users/build/msgpass/mpich/lib/solaris/ch_p4 -lmpi
endif
........
........
ifdef FFTPRG
  FFTLIB = -lfftw
endif
LIBS = -lvec $(FFTLIB)$MPILIB -lacml -lm -lg2c -latlas -lgs

First three lines of this file should be modified incase of different compilers are used. In case library paths are not set in the environment then they can also be prescribed by changing the LIBS argument in the last line.

Before compiling the code, path of the input files has to be set correctly to where code look for input files. This can be done by changing the function set_pllinfo in the file Hlib/src/Element_List.C, for example:

```
sprintf(fname,"/cluster/work/math/kumarh/3D/runs/parallel/parallel.part.%d",pllinfo.nprocs);
```

The file `parallel.part.%d` must be in the corresponding directory. Then the 3D code can be compiled using the following commands,

- Generating libvec.a library (See Section 5.2 for description):
  Goto directory `.veclib` and delete the file libvec.a if it exists then use following command,

  `gmake DIM=3 PARALLEL=1 COMPRESS=1 MHD=1`

  Here argument DIM is corresponds to dimension of the code. Use PARALLEL=1 is you want to compile for parallel runs using MPI libraries. Now copy the generated library libvec.a into directories `.gs` (only if code is compiled for parallel run), and `.Hlib/Linux` (or the corresponding operating system directory).
5.4 Mesh Generation using GID and generating input files

- **Generation of libgs.a** (only for parallel run. See Section 5.2)
  In the directory ./gs, delete libgs.a if exists and then use
  `>>gmake DIM=3 PARALLEL=1 COMPRESS=1 MHD=1 opt`
  Copy the generated libgs.a into the directory ./Hlib/Linux.

- **Generation of libhybridopt.a** (See Section 5.2 for description)
  In the directory ./Hlib/Linux delete libhybridopt.a if it exists and then use,
  `>>gmake DIM=3 PARALLEL=1 COMPRESS=1 MHD=1 opt`
  Note that one can also copy all the required libraries in the directories Hlib/Linux if you they are not set in your parth. If successful then this directory contains libraries libgs.a, libhybridopt.a, libvec.a.

- **Generation of the final executable nektarC**
  In the directory ./Compress3d/src use command
  `>> gmake opt`
  This will generate the executable file nektarC.

To run the code on single machine (note that for this you need to remove argument PARALLEL from the compilation commands) use,

`>>nektarC -n2 -V -chk -S ../runs/filename.rea`

The option -n2 stands for the interpolation order, however the option -V will force the code to first order interpolation (piecewise constant polynomial), the option -chk is used to write the output of the code, and -S means that an output file will be written at each IOSTEP steps, specified in the input .rea file(See Section 5.6). To submit a job on BRUTUS on 64 processors for 24 hours use,

`>>bsub -W 24:00 -n64 -o output.out -e error.out prun ./nektarC -n2 -V -chk -S ../runs/filename.rea`

The files output.out and error.out are useful to check the output of the code run. See [http://clusterwiki.ethz.ch/wiki/index.php/Brutus](http://clusterwiki.ethz.ch/wiki/index.php/Brutus) for checking the job status and other details about the cluster.

### 5.4 Mesh Generation using GID and generating input files

**GID** is mesh generation software developed by International Center for Numerical Methods in Engineering (ICNME) Barcelona, Spain(see [http://gid.](http://gid.))
5 Three Dimensional Nektar Code

cimne.upc.es/index.html for details). The main features of GID are,

- **Mehes**: GID is ideal for generating all the information e.g., structured and unstructured meshes, boundary and loading conditions, material types, visualisation of results, etc., required for the numerical methods.

- **Adaptivity**: GID is extremely easy to adapt to any numerical simulation code. In fact, GID can be modified by the user to read and write data in any format. We use post processing routine to convert files from GID to Nektar.

One can easily generate two and three dimensional adaptive mesh. This can be done by first drawing a contour for the surfaces we need and then generate surfaces. Form surface that forms a volume, a volume can be generated, that can be meshed adaptively with structured or unstructured mesh. The boundary conditions can also be prescribed if needed on each boundary surfaces or boundary curves. See [http://gid.cimne.upc.es/support_team/](http://gid.cimne.upc.es/support_team/) for the more details and manual of GID.

For our computations, we needs to assign the boundary conditions on each boundary surface. A file containing the boundary types is needs to be created with extension .cnd, that have the same name as the .gid file, e.g. filename.cnd. Copy this into the directory filename.gid, generated after we have defined the volume in GID and saved the file. An example of the boundary condition file .cnd on surfaces is given below.

```
NUMBER: 1 CONDITION: W
CONDTYPE: over surfaces
QUESTION: X-Constraint
VALUE: 0

END CONDITION
NUMBER: 2 CONDITION: V
CONDTYPE: over surfaces
QUESTION: X-Constraint

END CONDITION
NUMBER: 3 CONDITION: O
```

78
5.4 Mesh Generation using GID and generating input files

CONDTYPE: over surfaces
.............
END CONDITION
NUMBER: 4 CONDITION: P
CONDTYPE: over surfaces
............... 
END CONDITION

Here we have modified the file presented in [53] by replacing variable over lines to over surfaces.

As one can see that file .cnd defines four types of boundary conditions: W for wall, O for outflow, V for inflow and P for periodic. Once this file is copied inside the filename.gid directory, we need to reopen the filename.gid, to load the boundary conditions. Note that after opening the filename.gid again the comment "4 conditions read" should appear at the bottom left of the command window.

Now the boundary conditions can be assign to the geometry using : Data → Conditions → AssignCond. Finally the mesh can be generated using Mesh → Generate mesh and then specifying the reference mesh width.

In order to generate the filename.rea input file for Nektar code we need to export the mesh data into two files namely filename.msh and filename.txt. These files can be generated as follows: File → Export → GID mesh and save the file as filename.msh. Similarly File → Export → Text data report and save as filename.txt.

Before we proceed further we need to modify the filename.txt so that only information related to the boundary conditions is left. This can be achieved by removing the first part of the file, result in the filename.txt:

W
6 face= 1 Conds: 0 0 0 0 0 0
2 face= 1 Conds: 0 0 0 0 0 0
.............
4797 face= 1 Conds: 0 0 0 0 0 0
End
V
3580 face= 2 Conds: 0 0 0 0 0 0
Three Dimensional Nektar Code

        431 face= 3 Conds: 0 0 0 0 0 0
        End

        517 face= 1 Conds: 0 0 0 0 0 0
        ..............
        End

The input file filename.rea for Nektar can now be generated using executable programs: g2n_3d and BC_g2n_3d (See GID directory in [5.2]),

        >> g2n_3dl filename.msh filename.rea.tmp
        >> BC_g2n_3d filename.rea.tmp filename.txt filename.rea

The boundary conditions can be specified by modifying the file BC_g2n_3d.c suitably.

The input file filename.rea is then generated. We modify this file to add parameters like time step. We also modify it to add the initial conditions we want to use. See section [5.6] for the complete description of the input file.

### 5.5 Mesh partitioning for parallel computation

For a parallel simulation, we have to partition the mesh. As the processor normally have same processing power so it important that each processor has roughly the same number of elements associated with them. At the same time we want to minimize the inter node-communication as that can lead to data bottleneck. We use METIS program, developed by George Karypis and Vipin Kumar, in ([51], [52], [49], [50], [48]) (see http://glaros.dtc.umn.edu/gkhome/views/metis) for partitioning. A parallel version of METIS is also available.

The program kmetis, that is based on k-way partitioning and is give better results when the number of processors is more than 8 is used. The related program pmetis uses multilevel recursive bisection and is better when the number of processors is less or equal to 8.

To partition the mesh, we need to convert input file filename.rea, into a METIS format file. We use executable rea2metis3d for this, can be found in directory ./Utilities/src and compiled using,
5.6 Description of input file

```
>> gmake rea2metis3opt.
This generates the executable rea2metis3d. Using,
>> rea2metis3d filename.rea > Graph
results in a file Graph, the input file for METIS. We can now use executables
pmetis or kmetis for partitioning the mesh using, for example
>> kmetis Graph 64
which results in the file, Graph.part.64. Here 64 is the number of processors.
We rename this file to parallel.part.64 and copy it in the directory specified
during compilation in the function set_pllinfo in ./Hlib/src/Element_List.C (see
Section 5.3). This file contains numbers from 0 to 63 each corresponds to the
processor rank, and having lines, equal to the number of elements.
```

5.6 Description of input file

In this section we describe the structure of the input file filename.rea. A
description of two dimensional file can be found in [53]. Here we consider
the three dimensional case. The file can be divided in four sections. First
section contains various parameters used in the simulations. Second contains
the details of the elements, and its nodes. This is followed by the details
of connectivity of neighboring elements and boundary conditions. In the last part
initial conditions are prescribed. We describe these sections one by one.

◊ Initial Parameters

First Part of the input file filename.rea has the form,
```
****** PARAMETERS *****
REA file
3 DIMENSIONAL RUN
12 PARAMETERS FOLLOW
5.95    beta
500000  NSTEPS
0.001   DT
6       EQTYPE
1.000000 INTYPE
4       MODES
5000000 HISSTEP
1       ISDIFF
```
5 Three Dimensional Nektar Code

2000    IOSTEP
1.0     WTEMP
200     ReMag
1./107308  KINVIS

0 Lines of passive scalar data follows2 CONDUCT; 2RHOCP

..................
The parameter DIMENSIONAL RUN describes the dimension of the simulation, followed by PARAMETERS FOLLOW that specify number of the parameters follows. In this case it is 12. NSTEPS is the number of time steps to be computed, DT (see file ./Compress3d/src/drive.C) is the size of time step and EQTYPE is type of equation will be simulated (see files ./Compress3d/src/drive.C and ./Compress3d/src/prepost.C). Here 6 represent viscous MHD case. For ideal MHD simulations this number is 5. INTYPE is the order of integration and also used for memory allocation in prepost.C. The RK methods based on this parameter are described in drive.C. IOSTEP is number of time step after that output is written (see file ./Compress3d/src/analyser.C). In this case Nektar will write the output after every 2000 steps. WTEMP is wall temperature can be used in function Set_Dir_BC_MHD in file ./Compress3d/src/visc_mhd.C.

◊ Elements This second part of filename.rea is,

..................

0 Lines of passive scalar data follows2 CONDUCT; 2RHOCP

0 LOGICAL SWITCHES FOLLOW

**MESH DATA** 1st line is X of corner 1,2,3,4. 2nd line... is Y, 3rd line is Z
101044 3 101044 NEL,NDIM,NELV
ELEMENT Tet 1
-3.551119 -5.209383 -7.661370 -4.845650
146.932391 150.000000 146.810816 148.647399
-29.171127 -29.544244 -28.345018 -25.786158
ELEMENT Tet 2

..................

Parameter NEL is the number of elements in the mesh. Tet is the element type and for this element functions defined in ./Hlib/src/Tet.C are used. This structure allow meshes with mixed elements. The numbers below this represent the four nodes of the tetrahedron (each column). First line represent x coordinates, second y coordinates and third
Connectivity List and Boundary conditions

***** CURVED SIDE DATA *****
0 Number of curve types
0 Curved sides follow IEDGE, IEL, CURVE(I), I=1,5, CCURVE

***** BOUNDARY CONDITIONS *****

E 1 1 52774.000000 1.000000 0.000000
E 1 2 2.000000 1.000000 0.000000
E 1 3 13181.000000 1.000000 0.000000
E 1 4 3.000000 1.000000 0.000000

E 3233 1 3231.000000 4.000000 0.000000

W 3233 4 0.000000 0.000000 0.000000

E 13478 1 13476.000000 3.000000 0.000000
E 13478 2 27779.000000 2.000000 0.000000
s 13478 3 0.000000 0.000000 0.000000
R=1.0
U=0.0
V=0.0
W=0.0
E=91.571169
B=0.0
C=0.0
D=0.0
E 13478 4 17253.000000 2.000000 0.000000

P 40000 1 22.000000 1.000000 0.000000

E 40000 4 3.000000 3.000000 0.000000

E 44800 4 4799.000000 3.000000 0.000000

Here E represent the connectivity of the faces e.g. first line represent that
the face 1 of element 1 is also face 1 of element 52774. The line starting with 0 represent that face 4 of element 3233 is a wall and 0 represents that it has no neighbor. $s$ represent the inflow or outflow boundary conditions and following lines specify boundary condition for each of the variable(see ./Compress3d/src/prepost.C). The line starting with $P$ represent the periodic boundary conditions e.g. face 1 of element 40000 is periodic with face 1 of element 22. These boundary conditions are used in advection.C and visc_mhd.C(see Section 5.2) and can be modified there, if needed.

- **Initial Conditions**

  1 INITIAL CONDITIONS

  Arc

  ***** DRIVE FORCE DATA ***** PRESSURE GRAD, FLOW, Q

  0 Lines of Drive force data follow

  ***** Variable Property Data ***** Overrides Parameter data.

  Line starting with 1 represent the number of rows used to specify the initial conditions. Arc represent the case of initial condition Nektar will read in function Read.ICs in ./Compress3d/src/io.C

### 5.7 Post Processing and Output files

The simulations with Nektar code with input file filename.rea results in outputs files: filename.s.chk.hdr.n, where $s$ is an integer, sepctify the time steps after that output was written and $n$ is the rank of processor(See file analyser.C for more details). The files filename.fld.hdr.n are also generated at the beginning of the simulation.

To visualize in Tecplot, the output files need some post-processing:

```
>> p2sf13d -pn -r filename.rea filename_s.chk > filename_s.rst
```

In case of serial run we ignore this step and just rename the file with extension .chk to .rst. The program p2sf13d is in the directory ./Utilities/src (see Section 5.2). This result in the file filename_s.rst. The file filename_s.rst either can be used to restart the computation or converted into a .plt file that can be open in Tecplot. We use,
5.7 Post Processing and Output files

```bash
>> nek2tec2d -f -np -r filename.s.rea filename.s.rst > output_s.dat
>> n2t3d filename.msh output_s.dat.dat output_s_opt.dat
>> preplot output_s_opt.dat output_s_opt.plt to result in Tecplot readable file output_s_opt.plt. Executables p2sfld3d, and nek2tec3d are in directory ./Utilities/src and can be generated using
   >> gmake p2sfld3dopt
   >> gmake nek2te3dopt

n2t3d is a post-processing routine implemented to rewrite the output_s.dat file in optimal format for Tecplot, into the file output_s_opt.dat. The executable can be generated by compiling n2t3d.c in directory ./Utilities/src. preplot is another Tecplot optimizer routine available in Tecplot.
```
5 Three Dimensional Nektar Code
Performance of Runge-Kutta Discontinuous Galerkin Methods for Ideal MHD

6.1 Introduction

At present, it is believed that the particular numerical flux or Riemann solver used does not have a significant effect on the results of high-order RKDG simulations (see [54]). Such a conclusion is supported by numerical evidence such as that shown in Fig. 6.1. This figure shows the results of simulations of the MHD shock tube problem described in Section 6.2 with three different numerical fluxes (see Section 4.3 for descriptions), along with the exact solution to the problem. The results of first- and second-order simulations are shown in Figs. 6.1(a) and 6.1(b) respectively. Comparing these results, it appears that the absolute error in the numerical solution is far less sensitive to the choice of numerical flux when the second-order scheme is used. This seems to indicate that as the order of a simulation increases, the choice of numerical flux becomes less significant. This view has lead to the simple and highly dissipative Lax-Friedrichs (LF) flux being used within many RKDG methods, e.g. [54]. Our goal in this chapter is to rigorously examine the effect of more accurate numerical flux calculators in high-order RKDG methods, with particular emphasis on high-order simulations featuring discontinuities. The influence of accurate flux calculators in high-order RKDG methods is intimately tied to the performance of the limiters in the method. For this reason, we also examine the performance of limiters in high-order simulations.

For the Euler equations, the effect of Riemann solvers has been previously evaluated in [25]. One and two dimensional numerical simulations were carried out to compare various Riemann solvers based on performance measures such as numerical error, resolution of discontinuities and CPU times. The LF
Figure 6.1: Density profiles at $t = 0.4$ from (a) first-order and (b) second-order accurate simulations of the MHD shock tube problem described in Section 6.2. The exact solution to the problem is shown along with numerical results using the LF, HLLE and Roe fluxes described in Section 4.3.)
flux was shown to require the least CPU time among all the fluxes that were compared, but it also produced the largest numerical errors. Whereas second order fluxes such as Lax-Wendroff (LW) and Warming-Beam (WB) were found to be unstable. The Harten, Lax and van Leer (HLL) ([39]), HLLC ([55]) and MUSTA ([56]) fluxes were proposed as good choices for RKDG simulations. However, the data generated was not correlated to demonstrate which scheme is the most computationally efficient, or if the benefits of using more accurate Riemann solvers are dependent on the order of accuracy. In addition, error norms were only computed for a smooth linear problem, while we anticipate that the use of accurate Riemann solvers will be most significant in discontinuous nonlinear problems.

It was mentioned in Section [4.1] that to accurately simulate flows with fine scale features such as turbulence, it is not sufficient that a scheme be of high-order, it must also have good spectral properties. In particular, it is desired that the scheme has low dissipation for a broad range of wavenumbers. One quantity that can be used to assess the spectral properties of a scheme is the modified wavenumber. The imaginary part of the modified wavenumber provide information related to the spectral dissipation, whereas the real part provide information about the spectral dispersion of the numerical scheme.

While the modified wavenumber contains a wealth of information on the spectral properties of a spatial discretization scheme, it is not always possible to analytically compute the modified wavenumber for all schemes. Notably, shock capturing schemes use conservative approximations for the spatial derivative,

$$u'_j = \frac{1}{h}(\tilde{u}_{j+1/2} - \tilde{u}_{j-1/2}),$$

that require the definition of numerical fluxes $\tilde{u}_{j\pm1/2} = \tilde{u}(u_{j-q+1}, \ldots, u_{j+r})$. The flux function $\tilde{u}$ is generally a nonlinear function of its arguments even for linear equations, due to limiting. This nonlinearity prohibits the analytical calculation of $\Phi(\phi)$. For nonlinear schemes, Pirozzoli in [60] has devised a method to compute the approximate modified wavenumber behavior numerically.

This chapter is organized in two parts. In the first part we present performance of the Riemann solvers and limiters for RKDG methods. In Section [6.2] we test various Riemann solvers described in Section [4.3] for the Riemann problem.
for the first and second order Finite Volume Methods (FVM). This leads to the selection of the optimal choice of Riemann solvers. In Section 6.3 we test the performance of the limiters described in Section 4.4 for smooth solutions and solutions which are dominated by the discontinuities.

Part two of this chapter describes the spectral performance of the RKDG methods. First we present the details of approximated wavenumber in Section 6.4 followed by Section 6.5 where we discuss the numerical results for the various Riemann solvers and limiters. We also compare our results with well known schemes based on FVM. This is a joint work with V. Wheatley and P. Huguenot (see [61], [62]).

6.2 Performance of Riemann Solvers

Due to the number of solvers available, selection of the optimal solver for a given simulation is difficult. This issue was addressed by Wesenberg in [63], who assessed the computational time required to reach a fixed error bound for finite volume simulations of a suite of test cases using six different Riemann solvers. This was done for both first and second order accurate finite volume schemes. Wesenberg concluded that his MHD-HLLEM was the cheapest for achieving a fixed error bound. However, since the time of Wesenberg’s paper, several attractive new Riemann solvers of the HLLC family have been developed for MHD, e.g. [31, 32, 33]. Our goal here is to assess the performance of these new schemes. We will use a different test case to Wesenberg that more completely demonstrates the capabilities of the Riemann solvers.

In addition to being accurate and efficient, we must also consider the robustness of these Riemann solvers. Physical solutions of the MHD equations possess a positivity preserving property such that positive densities and pressures are always retained. However, nonphysical negative densities and pressures can be generated in numerical solutions. Thus it is important that the MHD Riemann solver selected leads to a positivity preserving scheme.

The primary test case we have selected to evaluate the Riemann solvers described in the Section 4.3 is a shock tube problem from [57]. This shock tube problem was selected as it includes all types of admissible MHD waves.
6.2 Performance of Riemann Solvers

(according to [58]) within one solution: fast- and slow-mode rarefactions, rotational discontinuities, fast shocks and slow shocks. Thus it will thoroughly exercise the wave resolving capabilities of the Riemann solvers. Torrihon’s shock tube problem is computed on the domain \(-1 < x < 1, 0 \leq t \leq 0.4\) with the following initial conditions:

\[
(\rho, p, u, v, w, B_x, B_y, B_z) = \begin{cases} 
(3, 3, 0, 0, 1.5, 1, 0), & x < 0 \\
(1, 1, 0, 0, 1.5, \cos(1.5), \sin(1.5)), & x > 0 
\end{cases}
\]

(6.1)

The boundary conditions for the problem are extrapolated outflow at \(x = \pm 1\), and periodicity in \(y\). This test case is computed for an ideal plasma with \(\gamma = 5/3\).

To facilitate comparison to published results (e.g. [63]), the Riemann solvers are tested within a standard finite volume method. First-order simulations use piecewise constant reconstruction in space and forward Euler time-stepping with a CFL number of 0.45. Second-order simulations use a MINMOD limited piecewise linear reconstruction in space and predictor-corrector time-stepping. For this test case we assess the performance of the schemes via the \(L^2\)-norm of the error in the cell averages, which in this case gives a measure of the scheme’s overall ability to resolve discontinuities, and the CPU-time required to simulate the problem up until the final time, \(T = 0.4\).

Fig. 6.2 shows the results of this test case. It shows the final density profiles produced by first-order finite volume simulations on a grid of 200 cells, along with the exact solution. Examining the density allows us to observe the performance of the schemes in the vicinity of the contact discontinuity, where the velocity and magnetic field do not vary. It does not, however, show the rotational discontinuities as these do not alter the density. In terms of resolution of discontinuities on a fixed grid, the Riemann solvers can be divided into three groups; the LF flux which is highly dissipative, the Roe, HLLC, HLLD, and HLLE-M solvers which capture all the discontinuities reasonably well even on this coarse grid, and the HLLE solver, which lies in between. In the Fig. 6.2(a) there appears to be little difference between the most accurate group of solvers. The Fig. 6.2(b) shows the solutions around the slow shock in greater detail. This allows us to see that the HLLD solution lies closest to the Roe solution, while the HLLE-M solution is even sharper (the shock is smeared over less cells) than Roe solution around the slow shock, due to the
Figure 6.2: Density profiles at $t = 0.4$ from first-order solutions of Torrilhon’s MHD shock tube problem on a grid of 200 cells using various Riemann solvers.
Figure 6.3: $L^2$-error in element average densities versus normalized CPU time for first-order simulations of Torrilhon’s MHD shock tube problem on a grids of 50, 100, 200, and 400 elements using various Riemann solvers.

action of the anti-diffusion terms in the HLLE-M solver.

To assess the performance of the schemes more rigorously, we plot the $L^2$-norms of the error in element average density versus the computational effort (normalized CPU time) in Fig. 6.3. Curves that lie lower on this plot require less computation effort to reach a fixed error bound, and are therefore said to perform better. From Fig. 6.3 we see that the LF-flux has the worst performance, followed by the HLLE solver. The performance of the HLLC-L, HLLC-G, HLLD, HLLE-M and Roe solvers is very similar, with HLLC-L being the best for this performance measure, followed by HLLE-M, HLLD, Roe, then HLLC-G. These rankings appear to be reasonably independent of resolution. Note, however, that the error in density does not assess the performance of the schemes around the rotational discontinuities, where we expect the HLLD and Roe solvers to be superior. To address this problem, in Fig. 6.4 we present the $L^2$-norms of the error in cell average $B_z$ versus the CPU-time for the first-
Performance of Runge Kutta Discontinuous Galerkin Methods for Ideal MHD

Figure 6.4: $L^2$-error in element average $B_z$ versus normalized CPU time for first-order simulations of Torrilhon’s MHD shock tube problem on a grids of 50, 100, 200, and 400 elements using various Riemann solvers.

order simulations. From this figure, we see that for this performance measure the Roe, HLLE-M and HLLD solvers are superior to the HLLC solvers, with the HLLE-M solver requiring the minimum computational effort to achieve a given error bound, followed by the HLLD solver.

In Fig. 6.5 we plot error norms versus the computational effort for second-order accurate simulations. The ordering of the schemes in terms of computational efficiency is mostly unchanged from the first-order results. The main exceptions are that in the second-order results, the Roe solver requires marginally less computational effort to produce a given error in $B_z$ than the HLLD solver, while in terms of the error in $\rho$, the performance of the Roe, HLLD, HLLE-M, and HLLC-L solvers is practically identical.

So far we have assessed the accuracy and efficiency of our set of Riemann solvers. The robustness of the Riemann solvers is another important consideration. There are no known issues with the robustness of HLLE and HLLC-
Figure 6.5: $L^2$-error in element average $\rho$ (a) and $B_z$ (b) versus normalized CPU time for second-order simulations of Torrilhon’s MHD shock tube problem on a grids of 50, 100, 200, and 400 cells using various Riemann solvers.
Figure 6.6: Partial density profile at $t = 0.4$ from second-order solutions of Torrilhon’s MHD shock tube problem on a grid of 400 cells using the HLLE-M Riemann solver.

type Riemann solvers, provided appropriate estimates for the signal speeds are used, as discussed in Section 4.3. There are however, multiple issues with the robustness of the Roe solver. In addition to the fact that the solver is not guaranteed to be positivity preserving (see [37]), in multiple dimensions it has also been shown to produce what is known as the carbuncle phenomenon (see [65]) when used to simulate shocks that are nearly stationary on the grid.

From the results presented so far, the HLLE-M solver appears to be the optimal choice of solver as it is gives the lowest error for a given computational cost. The source of the low error is the anti-diffusion terms in the solver. These cause the numerical representation of the discontinuities in the solution to be more sharply resolved than when the other solvers are used. This increased sharpness comes at a cost, however, as it results in the scheme being over-compressive which can lead to oscillations in the vicinity of discontinuities. Such oscillations are seen in the results of second-order accurate finite volume simulations, an example of which is shown in Fig. 6.6.
6.3 Performance of Limiters

6.2.1 Selection of Solvers

From the results presented in this section, the optimal choices of Riemann solver appears to be the MHD HLLD solver (see [33]) or the HLLC-L solver (see [31]). These solvers exhibit high computational efficiency and have similarly low numerical dissipation to the Roe solver, but do not suffer from the same robustness issues. They are not as computationally efficient as the MHD HLLE-M solver, in terms of the error in the magnetic field, but we observe that they do not generate numerical oscillations around discontinuities that we see when the HLLE-M solver is used. Additionally, both solvers have the property that they reduce to the Euler HLLC solver when the magnetic field vanishes, unlike the HLLC-G solver.

The HLLD solver requires slightly less computational effort than HLLC-L to produce the same error in the magnetic field, but the latter has the advantage that it is easier to implement. For this reason, we use the HLLC-L solver in our high-order RKDG simulations. In Section 6.3 we examine the impact of using this accurate Riemann solver in such simulations, compared to the LF flux.

6.3 Performance of Limiters

6.3.1 Simple Wave Problem

Landau et al. in [59] present an exact, non-linear smooth solution to the one-dimensional Euler equations (no magnetic field). The solution only exists if the initial conditions satisfy the following relations:

\[
\rho(x, 0) = \rho_0 \left(1 + \frac{(\gamma - 1)u_x}{2a_0}\right)^{\frac{2}{\gamma - 1}},
\]

\[
p(x, 0) = p_0 \left(1 + \frac{(\gamma - 1)u_x}{2a_0}\right)^{\frac{2\gamma}{\gamma - 1}},
\]

\[u_y = u_z = B_x = B_y = B_z = 0.
\]

The particular case we use the initial velocity,

\[u_x(x, 0) = u_0 \sin(\pi x),\]
on the domain $-1 < x < 1$ with the boundary condition $u(1, t) = u(-1, t)$. The exact solution is initially smooth, but the sinusoidal waves steepen non-linearly until a shock develops at $t_{\text{shock}}$, and the exact solution breaks down. The fact that the solution is smooth prior to $t_{\text{shock}}$ implies that the numerical methods we are interested in testing should reproduce the solution to their theoretical order of accuracy, even though this is a nonlinear problem. This is not the case with any solution that involves discontinuities, such as a shock tube problem, because around discontinuities these methods revert to first-order accuracy. The existence of an exact solution to the Euler simple wave problem allows us to study the convergence of RKDG schemes for a nonlinear problem. We will examine the results at $t = 0.9$, somewhat prior to the formation of a shock at $t \approx 0.95$.

In Fig. 6.7 the solution from limited second-order simulations are compared to the exact solution. The HLLC-L Riemann solver was used for both simulations. During the simulation, the less restrictive (LR) limiter did not detect that any limiting was required, thus it is unnecessary to show a separate curve for an unlimited simulation. On the other hand the minmod limiter detected that limiting was required on the majority of elements. This indicates a major
Figure 6.8: $L^2$ errors in element average density versus $N$ for the Euler simple wave problem at $t = 0.5$. The least-squares fitted order of convergence, $n$, for each scheme is indicated in the legend.

short-coming of the minmod limiter for high-order simulations: The minmod limiter selects the input gradient with the minimum modulus. If this is not the gradient of the polynomial representation of the field internal to the element, then limiting is carried out and the coefficients of the higher order modes are set to zero. In the RKDG method, the input gradients are the internal gradient, and slope estimates based on the difference between the element average fields and the average fields on the adjacent elements. In general, if the solution has a finite curvature, the internal gradient will not have the minimum modulus, thus the solution will be limited and will revert to (at most) second order accuracy. This will occur even if limiting is unnecessary, as is the case with the solution we are currently examining. From this we must conclude that the minmod limiter is unsuited for use in a high-order RKDG method. Examining the numerical solutions shown in Fig. 6.7 we see that the LR limited solution approximate the exact solution very well, while the minmod limited solution under-predicts the peak value.

In Fig. 6.8 we show how the $L^2$ error in element average density converges with
N for various RKDG schemes. The least squares fitted order of convergence for each scheme is indicated in the figure legend. Note that in these simulations, the LR limiter does not detect that any limiting is required, so there is no need to present separate LR limited and unlimited results. The solutions from the LR limited third- and fourth-order RKDG schemes converge at the expected rate. On the other hand, the results from the minmod limited fourth-order RKDG scheme show an order of convergence of between one and two. This is due to the solution being limited to piecewise linear or piecewise constant on the majority of elements, once again highlighting the short-comings of this limiter.

To further investigate the importance of using accurate Riemann solvers in high-order schemes, in Fig. 6.8 we compare the convergence of solutions from LR limited third- and fourth-order RKDG schemes using LF and HLLC-L Riemann solvers. For the third-order scheme, the solutions produced using the HLLC-L solver have lower errors and converge more rapidly, as might have been anticipated from our low-order results. However, for the fourth-order scheme the errors in the LF and HLLC-L results are indistinguishable on the scale of the plot for all discretizations used. This shows that high-order accurate solutions to the simple wave problem are insensitive to the choice of Riemann solver over a broad range of resolutions, supporting the view of Cockburn et al. (see [54]), at least for smooth solutions.

6.3.2 Behavior of Limited Solutions for Problems Dominated by Discontinuities

To investigate the effect of low-order limiting and accurate Riemann solvers in problems dominated by discontinuities, we compare numerical solutions to Torrilhon’s MHD Riemann problem from second-, third- and fourth-order RKDG schemes. These are shown in Fig. 6.9. Let us first examine the impact of the Riemann solver on solutions from the LR limited fourth order scheme. When the LF flux is used the $L^2$-error in the element average density is 0.0220. Using the more accurate HLLC-L solver, this error is reduced to 0.0172, which is significant. This is in contrast to our earlier results for smooth solutions indicates that when discontinuities are present, the choice of Riemann solver is still significant for high-order methods.
Figure 6.9: Numerical solutions to Torrilhon’s Riemann problem at $t = 0.4$ using second- and third- and fourth-order RKDG schemes. The HLLC-L flux solver was used in all simulations other than one. A fourth-order result using the LF flux is shown for comparison. The $L^2$ error in the element average densities for each simulation is also indicated in the legend.
We now turn our attention to the results of differing order produced using the HLLC-L solver. From Fig. 6.9 we observe that the solution from the third-order scheme is the least accurate overall, with an $L^2$ error in the element average density of 0.0272. The behavior of the second- and fourth-order LR limited schemes is similar, other than in the vicinity of the contact discontinuity where the second-order scheme is more accurate. This results in the error in the solution from the fourth-order scheme (0.0172) being greater than that from the second-order scheme (0.0138). Both of these two solutions are more accurate than that from the minmod limited second-order scheme, which has a error of 0.0188.

The reason why we do not observe an increase in accuracy as the order of the scheme is increased, and why the LR limiting procedure results in more accurate solutions, can be explained by examining where the limiter functions alter the polynomial order of the solution. For $t = 0.4$, this is shown in Fig. 6.10 for LR and minmod limited second-order RKDG schemes, and in Fig. 6.11 for LR limited third- and fourth-order RKDG schemes. From Fig 6.10 we see that in the second-order solution, the minmod limiter carries out limiting on almost all elements where the solution is non-constant, resulting in 117 of 200 elements being limited. The LR limiter limits far fewer elements, 32 of 200 in this example, mostly those on the edge of discontinuities. This results in the minmod limited solution being more diffuse than the LR limited solution, as can be seen in Fig. 6.9, and therefore less accurate, as the $L^2$ errors revealed. From Fig. 6.11 we see that 43 and 56 elements are limited to lower order polynomials in the solutions from the LR limited third- and fourth-order schemes, respectively. Also, the unlimited elements are in general located where the solution is close to linear, so that a higher-order representation of the data has little impact on the overall accuracy of the solution. The cumulative result is that the error caused by additional limiting outweighs any gain in accuracy due to the increased order of the scheme, resulting in the third- and fourth-order schemes producing less accurate solutions than the LR limited second-order scheme. For this discontinuity dominated problem, these results indicate that if elements are limited to a maximum polynomial order of one, and the overall scheme is at least second-order accurate, then the $L^2$ error in the solution is influenced more by the extent to which the solution must be limited than by the order of accuracy of the scheme.

The remaining issue to be addressed from the results shown in Fig. 6.10 and
6.3 Performance of Limiters

Figure 6.10: Distribution of limited elements in solutions to Torrilhon’s MHD Riemann problem at $t = 0.4$ from second-order RKDG schemes. The HLLC-L flux solver was used along with the minmod (top) and LR (bottom) limiters.
Figure 6.11: Distribution of limited elements in solutions to Torrilhon’s MHD Riemann problem at $t = 0.4$ from third-order (top) and fourth-order (bottom) RKDG schemes. The HLLC-L flux solver was used along with the LR limiter.
Fig 6.11 is why the third-order scheme produces the solution with the largest $L^2$ error in element average densities. The reason appears to be that the limiter function interacts poorly with the piecewise quadratic representation of the data, in that it limits the data to be piecewise constant on a far greater number of elements than for the other schemes, which locally reduces the order of the scheme to one. It is important to note that the distributions of limited elements shown in Fig 6.11 are instantaneous. These distributions change significantly with time. In particular, the third-order scheme was observed to require limiting on a far greater number of elements after the first and second sub-steps of the RK scheme than after the third and final sub-step, which is what we see in Fig 6.11. For example after the second sub-step at $t = 0.4$, the solution is limited to piecewise constants on 27 elements and piecewise linear functions on 29 elements. The fact that the scheme is reduced to first-order on this number of elements results in the solution it produces being less accurate than even the second-order minmod limited solution.

In conclusion, for high-order RKDG simulations of Torrilhon’s MHD Riemann problem, the polynomial representation of the data is limited to be at most piecewise linear around the discontinuities. The remaining elements where the representation is piecewise quadratic (for third-order) or cubic (for fourth-order) lie in regions where the exact solution is linear and would thus be represented equally well by a lower-order function. This implies that, at best, a high-order RKDG scheme will behave in the same manner as a second-order scheme the problem studied here. The only remedy for this would be to develop a limiter function that produces higher-order limited polynomials. Our results imply that for any problem featuring by discontinuities, the error in a solution produced by a high-order RKDG method is likely to be dominated by the low-order errors in the vicinity of the discontinuities. Such errors have been show to be significantly reduced by the use of accurate Riemann solvers in Section 6.2. This explains the improved performance of the fourth-order scheme when the HLLC-L solver was used instead of the LF flux.
6.4 Modified Wave Number

Consider the one-dimensional linear advection equation on an infinite domain with a sinusoidal initial condition with wavenumber \( k \).

\[
\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0, \quad -\infty < x < \infty, \quad u(x,0) = \hat{u}_0 e^{ikx}.
\] (6.2)

Here, we assume \( a > 0 \). Applying separation of variables, it is clear that the exact solution to Eqn.(6.2) has the form,

\[
u(x,t) = \hat{u}(t)e^{ikx}.
\] (6.3)

Inserting this into Eqn.(6.2) we obtain,

\[
\frac{d\hat{u}(t)}{dt} + iak\hat{u}(t) = 0, \quad \hat{u}(0) = \hat{u}_0,
\] (6.4)

which has the exact solution,

\[
u(x,t) = \hat{u}_0 e^{ik(x-at)}.
\] (6.5)

Let us now consider the semi-discrete approximation of Eqn.(6.2) on a uniform grid with nodes given by \( x_j = jh \),

\[
\frac{\partial u_j}{\partial t} + au_j' = 0, \quad u_j(0) = \hat{u}_0 e^{ikx_j},
\] (6.6)

where \( u_j(t) \approx u(x_j,t) \) and \( u_j' \) is a discrete approximation of the spatial derivative. If an explicit, linear finite difference approximation of \( u_j' \) is used,

\[
u_j' = \frac{1}{h} \sum_{l=-q}^{r} a_l u_{j+l},
\] (6.7)

then the exact solution \( u_j(t) = \hat{u}(t)e^{ikx_j} \) applies. Inserting this into Eqn.(6.6) we obtain,

\[
\frac{d\hat{u}(t)}{dt} + a \frac{1}{h} \sum_{l=-q}^{r} a_l e^{iklh} \hat{u}(t) = \frac{d\hat{u}(t)}{dt} + iak_{mod(k)}\hat{u}(t) = 0, \quad \hat{u}(0) = \hat{u}_0,
\] (6.8)
6.4 Modified Wave Number

This is identical to the continuous semi-discrete form given in Eqn. (6.4) except that the wavenumber $k$ is replaced by the modified wavenumber,

$$k_{\text{mod}}(k) = \frac{1}{ih} \sum_{l=-q}^{r} a_l e^{iklh}.$$ 

Hence, the exact solution to the semi-discrete equation is,

$$u_j(t) = \hat{u}_0 e^{i(kx_j - k_{\text{mod}}(k)at)}.$$ 

For convenience we introduce the reduced wavenumber $\phi \equiv kh$ and reduced modified wavenumber $\Phi(\phi) \equiv k_{\text{mod}}(k)h$. Rewriting the above equation in terms of reduced wavenumbers, we obtain,

$$u_j(t) = \hat{u}_0 e^{i(\phi j - \Phi at/h)} \cdot \hat{u}_0 e^{i(\Phi \cdot \Phi(\phi) \cdot \Phi at/h)}.$$ 

This illustrates the value of the modified wavenumber. $\Im(\Phi(\phi))$ contains information on the spectral dissipation properties of the scheme. If $\Im(\Phi(\phi)) > 0$, the scheme is unstable at that wavenumber, while if $\Im(\Phi(\phi)) < 0$, the scheme is dissipative. $\Re(\Phi(\phi))$ contains information on the spectral dispersion properties of the scheme. If $\Re(\Phi(\phi)) = \phi$, disturbances with that wavenumber will propagate at the correct speed in the semi-discrete evolution.

6.4.1 The Approximate Modified Wavenumber for Nonlinear Schemes

Following [60], consider a numerical simulation with the initial condition $u_j(0) = \hat{u}_0 e^{ij\phi_n}$ that is advanced a single, very small, timestep $\Delta t$ such that the error in time integration is negligible. If the linear, semi-discrete scheme were solved exactly, Eqn. (6.9) can be rearranged to give the following expression for $\Phi(\phi_n)$:

$$\Phi(\phi_n) = \frac{h}{i\alpha \tau} \log \left( \frac{\hat{u}(\phi_n; \tau)}{\hat{u}_0} \right), \quad (6.10)$$

For a non-linear scheme, we can use a similar expression. However, we must account for the fact that the nonlinearities produce additional Fourier modes in the numerical solution at time $\tau$. Thus, the amplitude of the mode associated with the reduced wavenumber $\phi_n$, $\hat{u}(\phi_n; \tau)$, must be extracted from the
solution by means of a discrete Fourier transform (DFT). On the a finite grid with nodes \(j = 0, ..., N\), the supported Fourier modes have reduced wavenumbers \(\phi_n = 2\pi n/N\) in the range \(0 \leq \phi_n \leq \pi\), and the DFT of the solution at \(\phi_n\) is given by,

\[
\hat{u}(\phi_n; \tau) = \frac{1}{N} \sum_{j=0}^{N-1} e^{-ij\phi_n}.
\]

Using this in Eqn.(6.10), the following expression is obtained for the approximate modified wavenumber corresponding to \(\phi_n\):

\[
\Phi(\phi_n) = \frac{h}{ia\tau} \log \left( \frac{\hat{u}(\phi_n; \tau)}{\hat{u}(\phi_n; 0)} \right) . \tag{6.11}
\]

It is important to note that \(\hat{u}(\phi_n; 0)\) is the DFT of the numerical initial condition, which may differ from the exact initial condition if, for example, the initial values were converted to cell averages to initialize a finite volume method. In order to obtain the modified wavenumber behavior for the complete spectrum, simulations must be carried out for all \(\phi_n\).

It is important to note that in the general case, several Fourier modes of \(O(1)\) amplitude will be present in a simulation. Due to the nonlinearity of shock capturing schemes, these modes will interact and thus their evolution will not be governed precisely by the approximate modified wavenumber.

### 6.5 Approximate Modified Wavenumber Behavior of DG Methods

In order to compute the approximate modified wavenumber behavior of DG methods for MHD, a series of simulations were carried out where a sinusoidal transverse magnetic field was advected with a constant velocity. The initial conditions were as follows:

\[(\rho, p, u, v, w, B_x, B_y, B_z) = (1, 1, 1, 0, 0, 0, \cos(2\pi nx), 0) .\]

This differs from the approach of Pirozzoli(see [60]), who carried out simulations of the linear advection equation. Although the exact solution to our problem is a linearly advected \(B_y\) profile, the numerical solution will differ
6.5 Approximate Modified Wavenumber Behavior of DG Methods

from that produced by a discretization of the linear advection equation due to the presence of additional characteristic speeds. These alter the timestep as well as the output of most of the Riemann solvers tested. Our approach should give a more accurate representation of the behavior of DG methods for MHD. Our simulations were carried out on a mesh with 100 elements in the $x$-direction, with element centers $x_j = jh$ and $h = 0.01$. Simulations were run for $1 < n < 49$.

Let us first examine the resulting modified wavenumber behavior for unlimited second-, third- and fourth-order DG methods, this demonstrates how the scheme behaves in regions where the solution is smooth and is shown in Fig. 6.12. In the figure, the results for unlimited DG methods are compared to the exact behavior of a spectral method and that of a first order upwind method. The behavior of the third- and fourth-order methods was computed using both the LF flux and the HLLC flux. All the DG methods are far superior to the first order upwind scheme. The third- and fourth-order DG schemes have much better spectral properties than the second order scheme: $\Re(\Phi)$ departs significantly from spectral behavior at a much higher wavenumber, and $\Im(\Phi)$ is significantly less negative, indicating lower dissipation, particularly at high wavenumbers. In particular, it can be seem that the dispersive performance of the fourth-order schemes is remarkably good, with no visible deviation of $\Re(\Phi)$ from the spectral line on the scale of the plot. The main value of these plots, however, is to observe the effect that the Riemann solver has on the spectral properties of the method. As discussed in Section 6.1 it has been claimed that the choice of Riemann solver is unimportant for high order methods. In the behavior of $\Re(\Phi)$, this claim appears to be valid; there is minimal difference in the dispersion behavior of the third order methods with the HLLC and LF fluxes. However, when we examine the imaginary part of the modified wavenumber, we see that when the HLLC flux is used, there is a very significant decrease in the dissipation at most wavenumbers for both the third- and fourth-order schemes, indicating that the choice of Riemann solver is important. Next, we investigate the effect of limiting on the spectral behavior of DG methods. In Fig. 6.13 the modified wavenumber behavior of minmod and less restrictive limited (see Section 4.4) second-order DG schemes are compared to the unlimited behavior. Once again the behavior of spectral and first-order upwind schemes are plotted for reference. Initially, at low wavenumbers, the behavior of the limited schemes coincides with that of the unlimited schemes.
Figure 6.12: Real (top) and imaginary (bottom) parts of the approximate modified wavenumber for various RKDG schemes.
6.5 Approximate Modified Wavenumber Behavior of DG Methods

Figure 6.13: Real (top) and imaginary (bottom) parts of the approximate modified wavenumber for second order limited DG schemes.
Figure 6.14: Real (top) and imaginary (bottom) parts of the approximate modified wavenumber for the first-order upwind method (UW1), second-order TVD finite volume methods using the minmod (TVD-MM), Superbee (TVD-SB) and van Leer (TVD-VL) limiters, the third-order centered scheme (CS3), and third- (WENO3), fifth- (WENO5) and seventh-order (WENO7) WENO schemes [60].
6.5 Approximate Modified Wavenumber Behavior of DG Methods

It can be seen that the effect of the limiters is to force the scheme towards the behavior of the first-order scheme at high wavenumbers. The choice of limiter has a significant effect on the modified wavenumber of the scheme. The minmod limited scheme departs significantly from the behavior of the unlimited scheme at a significantly lower wavenumber than the less restrictive limited scheme. Interestingly, for the scheme with the less restrictive limiter, \( \Re(\Phi) > 0 \) at the lowest wavenumber, although its value is so small that it is not visible in the figure. This indicates a linear instability of the scheme at the lowest wavenumber. Pirozzoli [60] observes similar behavior for the Superbee limited second-order finite volume scheme, and claims that the nonlinear stability properties of the scheme compensate for this.

The approximate modified wavenumber behavior obtained in [60] for various schemes is shown in Fig 6.14. As expected, the second-order minmod-limited finite volume and DG schemes exhibit very similar behavior. The behavior of the less restrictive-limited DG scheme resembles that of the Superbee-limited finite volume scheme. The third-order WENO behavior begins to depart significantly (by 2%) from spectral at approximately \( \phi = 1.25 \) for the real part, and \( \phi = 0.6 \) for the imaginary part. The unlimited third-order DG scheme performs much better than this, with significant departures from ideal behavior beginning at \( \phi = 1.8 \) and \( \phi = 1.25 \) for the real and imaginary parts, respectively.

The computation of the approximate modified wavenumber behavior for limited third- and fourth-order DG schemes was also attempted. However, the analysis predicted that these schemes would produce very high dissipation, even at low wavenumbers, despite the fact that the schemes should have, at worst, the same performance as the second-order scheme. The reason for this is the fact that the limiter function reduces the polynomial order of the representation of the data to unity where ever limiting is required. For the second-order scheme, this does not cause any problems as the representation of the data is already piecewise linear. For the third- and fourth-order schemes, the data is represented by piecewise quadratic and cubic polynomials, respectively. For wavenumbers higher than \( k = 1 \), limiting occurs, locally reducing the polynomial order of the data from that in the numerical initial condition to unity, which constitutes a large amount of dissipation in a single timestep. For this reason we believe that the method of Pirozzoli (see [60]) does not produce meaningful results for schemes where maximum polynomial order of limited
data does not match the order of the unlimited data.

Despite the fact that we cannot obtain meaningful results for high-order limited DG schemes, we can make the following statement: Due to the nature of the limiter used in our RKDG method, in the vicinity of discontinuities the method exhibits high numerical dissipation, similar to that of second-order finite volume methods. In this region they are inferior to high-order WENO schemes that can be seen in Fig. 6.14 to provide improved modified wavenumber behavior. This is because the reconstructed numerical solution is always a high-order polynomial in the WENO schemes, it is not limited to a piece-wise linear function in the vicinity of discontinuities. This situation can only be improved by the development of high-order limiter functions for RKDG methods.

6.6 Conclusions

To investigate the importance of accurate Riemann solvers in RKDG simulations, we examined the results of first-, second-, third-, and fourth-order simulations of both smooth and non-smooth problems. In the presence of discontinuities, the influence of Riemann solver selection is directly tied to the performance of limiters. Thus we also examined the performance of common limiters in these simulations. It was noted that a particular extension of one-dimensional limiters to multiple dimensions can result in the TVD condition being violated.

We first examined the computational efficiency of several recently developed MHD Riemann solvers, in terms of the computing time required to achieve a fixed error bound for a given problem. The performance of these solvers was compared to that of the Roe solver, the LF flux and the MHD HLLE-M solver, which had previously been identified as the most efficient by [63]. This was done for first- and second-order schemes. We find that all of the accurate Riemann solvers are more computationally efficient than the LF flux, with the HLLE-M, HLLC-L, HLLD and Roe solvers exhibiting similar efficiency. The HLLD and HLLC-L solvers were identified as good candidates for use in high-order simulations as of the most efficient group of solvers, these have the most desirable robustness properties.
From our third- and fourth-order results for smooth solutions, we conclude that the minmod limiter is unsuited for use in high-order RKDG methods. The reason for this is that in regions of finite curvature, this limiter tends to restrict the polynomial order of the trial space, reducing the scheme to at most second-order accuracy, even when limiting is not strictly required to keep the scheme TVD. The less-restrictive limiter does not exhibit this behavior. For third-order simulations of smooth problems, we find that using the HLLC-L solver results in lower errors and faster convergence than when the LF flux is used. However, for fourth-order simulations of the same problem, we observe negligible differences in the solutions when the Riemann solver is varied. This supports the view of [54] that the results of high-order RKDG simulations are insensitive to the choice of Riemann solver, at least for smooth solutions.

From our simulations of problems dominated by discontinuities, we find that in the vicinity of discontinuities, high-order RKDG methods behave in a similar manner to the second-order method due to the use of a piecewise linear limiter. The error in numerical solutions to such problems is dominated by the error in these regions. Thus, for such solutions, the choice of Riemann solver used in a high-order method has a similar significance as for a second-order method. Our analysis of second-order methods indicates that the choice of Riemann solvers is very significant, with the more accurate Riemann solvers having the best performance. This results in fourth-order simulations of discontinuous solutions using the HLLC-L solver having considerably lower error than those using the LF flux, in contrast to the result for smooth solutions.

The spectral properties of RKDG schemes where investigated by computing their approximate modified wavenumber behavior. The modified wavenumber behavior of high-order unlimited RKDG schemes was found to be excellent. In particular, the dispersive performance of the fourth-order scheme is remarkably good, with very little deviation from spectral behavior over the complete range of numerically resolved wavenumbers. The dissipation of this scheme is also very low, even at the highest wavenumbers. This indicates that in regions away from discontinuities, where no limiting is required, RKDG schemes provide the high accuracy and low numerical dissipation needed, for example, to carry out LES simulations. These demonstrated that the third- and fourth-order RKDG schemes produce solutions with very little dissipation at all wavenumbers that converge at the theoretical rate. When limiting is required, however, the spectral performance of RKDG schemes tends to that...
of the first-order method at high wavenumbers. Thus in the vicinity of discontinuities, high-order RKDG methods exhibit high numerical dissipation, similar to that of second-order finite volume methods. In this region they are inferior to high-order WENO schemes that have been shown to provide improved modified wavenumber behavior. This is because the reconstructed numerical solution is always a high-order polynomial in the WENO schemes, it is not limited to a piecewise linear function in the vicinity of discontinuities. This situation can only be improved by the development of high-order limiter functions for RKDG methods.
7 High Current Axis Symmetric Arc Simulations with External Magnetic Field

7.1 Introduction

As discussed in Section 1.5 of Chapter 1 we use MHD equations for high current arc simulations. This was referred to the strongly coupled model in Chapter 1. These equations were derived in Chapter 2 from the statistical description of plasma described by Boltzmann equation by assuming high collision frequency and quasi neutrality. One can easily see that these assumptions are easily satisfied by the plasma arc that occurs during the current interruption in circuit breakers. This chapter is based on the results obtained by P. Huguenot (see [53], [66]).

To carry out the computations for the axis symmetric arc, we convert these equations into the cylindrical coordinates system. Using assumptions of axis symmetry we drop all the terms containing derivatives with respect to angular axis $\theta$. This result in a system of MHD equations with radial and axial fluxes only. We then modify the RKDG methods described in Chapter 4 to calculate these fluxes and their derivatives with respect to radial variable $r$ and $z$ variable. Details about the RKDG methods for the axis symmetric code can be found in [53].

For the real gas ($SF_6$) arc computations, we have implemented several routines in Nektar code to compute various non-dimensional coefficients. These coefficients are functions of temperature and pressure but we have ignored their dependence on the pressure for simplicity. Details of these coefficients and their implementation can be found in [53].
7 High Current Axis Symmetric Arc Simulations with External Magnetic Field

At high temperatures, gas starts to radiate. Hence we need to take radiative transfer into account. This was done by adding Stefan’s radiation term:

\[ S_d = sT^4, \]

into the energy equation as a source term. Here \( s \) is the radiation constant.

To test the effects of the external magnetic field on the axis symmetric arc, we first need to generate the arc. This is done by first assuming that an arc initially exists, by considering a current and conductivity profile (see Section 7.2). This profile is then evolved till it reaches steady state with suitable boundary conditions. This steady state is then considered to be the actual arc, on which we then test the effects of external magnetic field. This chapter is organized as follows:

In Section 7.2 we first present the details of the axis symmetric computation for the arc generation and then discuss the numerical results. Next Section 7.3 contains the implementation details and result for the effects of external magnetic fields on the arc which is followed by the conclusions in Section 7.4.

7.2 Arc generation

To generate the arc in axis symmetric setting, we consider the geometry shown in Fig. 7.1 with two arc roots \( C_1 \) and \( C_2 \), where the arc is initially attached. The electrodes (C1 and C2) have the width of 10 mm each. The height of the geometry is 200 mm and width is 150 mm. The outflow (O) widths are 10 mm and 30 mm for lower and upper boundary respectively. We have considered the arc which is not parallel to the \( z \) axis because we want to test the effect of external magnetic field applied in axial (\( z \)) direction.

To calculate the non-dimensional coefficients, we introduce the reference variables. We consider a reference length of \( L_0 = 10^{-3} \) m. The reference pressure \( P_0 = 6 \times 10^5 \) Pa and reference temperature \( T_0 = 300K \). At this temperature and pressure the \( SF_6 \) has density 35.13 \( kg/m^3 \) which is our reference density \( \rho_0 \). The reference velocity \( V_0 \) can be calculated using the relation \( V_0 = \sqrt{P_0/\rho_0} \) and this results in \( V_0 = 130.6 \) m/s. Similarly we can calculate reference magnetic field \( B_0 \) using Eqn. (3.10), and get \( B_0 = 0.8683 \) T.
7.2 Arc generation

Figure 7.1: Geometry for axis symmetric computations. Axis of rotation is $A$ with outflow boundary conditions at $O$ and wall boundary conditions at $W$. Arc roots are at the electrodes $C_1$ and $C_2$.

The main computational difficulty we face in these simulations is the dependence of coefficient on temperature, which results in very large variation in their values. In particular, in the low temperature regions we have low conductivity which results in a very low Lundquist number. This make the evolution problem stiff and results in very small time step for explicit time stepping due to CFL condition. To overcome this difficulty we assume that the permeability constant $\mu_0 = 1.0$ in our computations. We do realize that this can have significant quantitative effect on these results, but we believe that qualitatively these results still hold.

To generate the *numerical* arc, we first prescribe a current profile that has a constant thickness of 5 mm and it is centered in the middle of arc root $C_1$ and $C_2$. In this region we consider a large artificial electric conductivity profile given in Fig. 7.2(a). The artificial conductivity decays exponentially with time so that when dimensional time reaches $t = 0.75 ms$, the value of artificial conductivity is very low compare to that of actual conductivity. This profile is necessary as we want the current to stay near to a region where it is initially prescribed. At this state current heats up the gas, and we have the new profile of conductivity based on the temperature and then the current is
7 High Current Axis Symmetric Arc Simulations with External Magnetic Field

(a) Initial artificial conductivity $\sigma(1/\Omega m)$ (b) Initial magnetic field $B_\theta$ corresponding to constant current in the initial arc

Figure 7.2: Profile of initial conductivity and angular magnetic field $B_\theta$

free to move based on this new conductivity.

We consider the following Initial conditions in non-dimensional variables,

- Initial conditions:

$\rho = 1.0$ \hspace{1cm} (7.1a)

$v = 0.0$ \hspace{1cm} (7.1b)

$p = 1.0$ \hspace{1cm} (7.1c)

$B_r = 0.0$ \hspace{1cm} (7.1d)

$B_z = 0.0$ \hspace{1cm} (7.1e)

Initial conditions for $B_\theta$ can be calculated by using the Biot Savart Law. The resultant profile for $B_\theta$ from constant total current $I = 100kA$ is shown in Fig. 7.2(b)

For the outflow boundary conditions we consider following values of the non-dimensional variables,
7.2 Arc generation

Figure 7.3: Result: Non-Dimensional axial and radial velocities respectively

(a) Non-Dimensional axial velocity at time $t = 1.8\ ms$

(b) Non-Dimensional radial velocity at time $t = 1.8\ ms$

Figure 7.4: Result: Non-dimensional angular magnetic field and temperature respectively at time $t = 1.8\ ms$

(a) Non-dimensional angular magnetic field at time $t = 1.8\ ms$

(b) Temperature(in Kelvin) at time $t = 1.8\ ms$
7 High Current Axis Symmetric Arc Simulations with External Magnetic Field

Figure 7.5: Result: Pressure and non-dimensional current density with current lines respectively

- Boundary conditions: outflow $O$

\begin{align*}
\rho &= 1.0, & (7.2a) \\
\mathbf{v} &= 0.0, & (7.2b) \\
p &= 1.0, & (7.2c) \\
\mathbf{B} &= 0.0. & (7.2d)
\end{align*}

Wall boundary conditions are implemented for velocity by inverting the normal component of velocity at the wall. Magnetic field conditions for the wall are implemented by assuming a no current condition.

### 7.2.1 Numerical Results

For computation we use 6000 triangular elements. We use Lax-Friedrichs flux for computation of the fluxes in the simulations. For the time stepping first order forward Euler explicit scheme is used. All the results presented here are first order. After dimensional time $t = 1.8 \text{ ms}$, we get the following results:

Figs. 7.3(a) and 7.3(b) are the resulting non-dimensional axial and radial velocity profiles respectively. As one can see from the radial velocity profile,
that the gas is moving away from the arc, which is expected. Also note that radial velocity is maximum at the contact C_1. Figs. 7.4(a) and 7.4(b) are the non-dimensional angular magnetic field and dimensional temperature profiles respectively. Temperature profile represents the actual position of the arc as higher temperature implies higher conductivity, which governs the current profile. Figs. 7.5(a) and 7.5(b) represent the dimensional pressure and non-dimensional current density profile respectively. Comparing the current and temperature we see that current density profile is governed by the temperature profile.

### 7.3 Effects of External magnetic field

In a rotating arc circuit breaker, the current that flows inside the arc also goes through a coil located around the arc chamber. This coil generates a magnetic field in the whole region. This external magnetic field interacts with the arc through Lorentz forces terms in the momentum conservation equation and results in a velocity profile. Designers can use this to build pressure in a chamber and use it to blow the arc when current goes through zero phase. This could results in a significant decrease in operating energy of a circuit breaker.

In this section, we will test the effect of an external magnetic field applied

![Figure 7.6: Results: Non-dimensional axial and radial velocities respectively](image)

(a) Non-dimensional axial velocity (b) Non-dimensional radial velocity with external magnetic field at time $t = 3.0 \text{ ms}$
7 High Current Axis Symmetric Arc Simulations with External Magnetic Field

on the arc in axial($z$) direction. When this magnetic field will interact with non parallel arc current (that is why we have chosen arc current $J$ such that it is not parallel to axial direction), it should result in a velocity component in direction $J \times B$ which is the angular($\theta$) direction.

We consider external magnetic field $B_z = 0.05$. It corresponds to a dimensional magnetic field of $0.043T$. This magnetic field is added to the arc obtain in the Section 7.2.1. Also we keep $B_z$ constant during the computation.

7.3.1 Numerical Results

After a dimensional time of $t = 3.0\ ms$, following results were obtained:

Figure 7.7: Results: Non-Dimensional angular magnetic field and dimensional temperature.

Figs. 7.6(a) and 7.6(b) are the non-dimensional axial and radial components of the velocity with external magnetic field respectively. Figs. 7.7(a) and 7.7(b) are the non-dimensional angular magnetic field and dimensional temperature respectively. One can see that there is a significant change in the temperature profile due to external magnetic field. Fig. 7.8(a) is the dimensional pressure with external magnetic field. Note that maximum pressure is increased from 12 bars to 20 bars, which is a significant increase, due to external magnetic
7.4 Conclusion

(a) Pressure (in Bars) with external magnetic field at time $t = 3.0 \text{ ms}$

(b) Non-dimensional current density and current lines with external magnetic field at time $t = 3.0 \text{ ms}$

Figure 7.8: Results: Pressure and current density with current lines

A change in the temperature profile results in a corresponding change in the non-dimensional current density profile (see Fig. 7.8(b)).

However as we have discussed in the start of this Section that due to the Lorentz force term $J \times B$, this external magnetic field should give rise to the angular component of velocity. This resulting non-dimensional angular velocity is shown in Fig. 7.9.

The maximum non-dimensional velocity is $V_\theta = 0.07$, which corresponds to the dimensional velocity of 9.15 $m/s$ using the reference velocity $V_0 = 130.6 \text{ m/s}$ from the previous Section. Considering the arc radius of 10 $mm$, we find that the rotation frequency of the arc due to the external magnetic field is 147 $Hz$.

7.4 Conclusion

We have shown that MHD equations (strongly coupled system) are suitable for the high current arc simulations. Axis-symmetric arc is generated by assuming that an arc already exists with total current of 100 $kA$. This arc was then evolved for its steady state which was at time 1.8 $ms$. We then applied external
Figure 7.9: Non-dimensional angular velocity $V_\theta$ generated by axial magnetic field $B_z$

magnetic field $B_z$ in the axial direction and evolve it till time 3.0 ms where flow is almost steady. This result in the angular component of velocity which rotate the arc. This rotating arc can be used to generate pressure, which can result in a significant decrease in the operating energy of the high current generator circuit breakers.
8 High Current Arc Simulations in Three Dimensions and Effects of External Magnetic Field

8.1 Introduction

A full three dimensional simulation of the plasma arc is a challenging objective. This is due to many factors like complicated geometry, highly varying gas data, presence of multiple scales etc. One needs to make several simplifications and choose suitable numerical methods, in order to simulate an arc. These simplifications can involve, ignoring some physical phenomena, simplifying the geometry or ignoring some uninteresting scales. As discussed in the Chapter 6 we use MHD equations for our simulations.

For the numerical simulations, one needs methods that can handle complicated geometry, are adaptive, easily parallelizable and are of high order of accuracy. As we have discussed in Chapter 4 RKDG methods have all these properties. So we choose these methods for our simulations using the Nektar code. This code was described in Chapter 5.

As for the radially symmetric code, the three dimensional code is modified to implement the real gas data for $SF_6$ gas. This was done by adding routines for calculating the gas constant, Reynolds number, Lundquist number and heat transfer coefficients. We use approximate gas data functions depending on the temperature. We have also implemented radiative transfer using Stefan’s radiation. This results in a source term

$$S_d = sT^4$$

in the energy Eqn. (3.14d). See Section 5.2 and file ./Compress3d/src/visc_mhd.C for more details.
8 High Current Arc Simulations in Three Dimensions and Effects of External Magnetic Field

(a) Three dimensional geometry for the Arc  (b) XY plane cut of the geometry simulations

Figure 8.1: Geometry of the Arc chamber

To generate the numerical arc in three dimensions, we consider a domain with high temperature. An arc is assumed to exist between the two contacts. The key idea is that with time, gas will radiate which will result in temperature reduction everywhere except where gas is heated by the current. We simulate the arc for the sufficient time so that arc reached a steady state and consider this as an actual arc. In the next Section 8.2, we will describe computational details of this simulation, followed by a discussion of the simulation results.

As we have discussed in Chapter 6, we have chosen the arc which is not parallel to y axis. Effects of the external magnetic fields are then tested by applying magnetic field $B_y$. This is similar to the setting in Chapter 6 except we have changed axial axis z in the last chapter to y-axis here. Section 8.3 contains the computational details. This results in a rotation of arc, which is then discussed. In the end we conclude in Section 8.4.
8.2 Arc generation

We consider the geometry shown in Figs. 8.1. In Fig. 8.1(a) we have the three dimensional domain for the computation which is arc chamber of the circuit breaker. Fig. 8.1(b) is XY plane cut of the three dimensional geometry. The domain is axial symmetric along the y-axis, and boundary conditions setup is exactly same as in the last Chapter. The radius of domain is 70 mm and y-axis length is 200 mm long. We have arc attached to both electrodes which are 10 mm wide.

The reference length we consider is $L_0 = 10^{-3} \text{ m}$. Reference pressure is $P_0 = 10^6 \text{ Pa}$, and reference temperature is $T_0 = 5000 \text{ K}$. Using the gas data, we have reference density $\rho_0 = 0.506 \text{ kg/m}^3$. We now calculate the reference velocity $V_0 = \sqrt{P_0/\rho_0} = 1405.8 \text{ m/s}$. Similarly we calculate reference magnetic field $B_0 = \sqrt{P_0\mu_0} = 1.121 \text{ T}$.

Using these reference variables and assuming that the minimum conductivity is $\sigma_{\text{min}} = 6000$, Lundquist number $S_r = \mu_0 V_0 L_0 \sigma_{\text{min}} = 1.06 \times 10^{-2}$ that is too small and we have a stiff problem. So we multiply this with constant
8 High Current Arc Simulations in Three Dimensions and Effects of External Magnetic Field

(a) Current density contour for $J = 10 \text{ A/m}^2$.

(b) $XY$ plane cut of current density

Figure 8.3: Current density profile at $t = 0.569 \text{ ms}$

(a) Absolute Magnetic field and directional vector

(b) $B_z$ Magnetic field in $YZ$ plane.

Figure 8.4: Magnetic field at $t = 0.569 \text{ ms}$
8.2 Arc generation

Figure 8.5: Magnetic field at $t = 0.569$ ms

(a) $B_y$ magnetic field in $YZ$ plane

(b) $B_z$ magnetic field in $XY$ plane

Figure 8.6: $\|v\|$ field of the arc at $t = 0.569$ ms

(a) $\|v\|$ Contour for $\|v\| = 350 \text{ m/s}$

(b) $\|v\|$ field in $XY$ plane
Similarly we multiply $G_r$ with constant 20. We have the following initial condition for the simulations:

- Initial conditions:

\[
\begin{align*}
p &= 0.0829 \quad (8.1a) \\
T &= 20000 \, K \quad (8.1b) \\
v &= 0.0 \, m/s \quad (8.1c) \\
p &= 10^6 \, Pa \quad (8.1d) \\
B_y &= 0.0 \, T \quad (8.1e)
\end{align*}
\]

$B_x$ and $B_z$ are calculated using Biot-Savart Law, as in the last Chapter. They correspond to the total current of $I = 100 \, kA$ and the initial arc joins arc roots, with width 10 mm.

Wall boundary condition for the wall are the same as in the previous chapter. Wall temperature is $T = 10000 \, K$ except at the arc roots where we put $T = 20000 \, K$. For outflow boundaries we extrapolate all the variable at the boundary.

We use 101044 tetrahedron elements in our computations. For these simulations we use Lax-Friedrichs flux for flux calculations and explicit Euler forward scheme is used for the time stepping. Computational time is 24 hours with 64 processors on the BRUTUS cluster. At time $t = 0.569 \, ms$ we have the following results:

Figs. 8.2(a) and 8.2(b) are the temperature profile of the arc. Fig. 8.2(a) is surface contour of the temperature $T = 12000 \, K$. Combining both figures we can see that we have high temperature at the arc roots and the most of the heating is takes place at the center of the domain. Figs. 8.3(a) and 8.3(b) are the current density profile of the arc with the current lines and the current moving downward. We can see that current has moved to the center of the domain from its initial position due to high temperature there. Corresponding to the current profile we have magnetic field profiles in Figs. 8.4 and 8.5. One can observe from Figs. 8.4(a), 8.4(b), and 8.5(b) that magnetic field is axis symmetric. On the other hand in Fig. 8.5(a), $B_y$ is really small compared to $B_x$ and $B_z$. Fig. 8.6(b) gives the profile of the velocity field. We also note that the gas is pushed away from the arc, through the outflow boundaries. There is also a bifurcation near the lower end of the domain.
8.3 Effects of external magnetic fields

As discussed in Chapter 6, we shall use an external magnetic field to rotate the arc in an arc chamber. To do so here we apply magnetic field $B_y = 0.5\, T$. This is achieved by adding this to the arc generated in Section 8.2, as the initial conditions and then modifying the boundary conditions for magnetic field with $B_y = 0.5\, T$. Note that during the simulations axial component $B_y$ of magnetic field is also evolved. The computational time was another 24 hours with 64 processors. After $t = 1.138\, ms$, we obtain the following results:

Fig. 8.7 is the temperature profile of the arc. We can see from the $T = 12000\, K$ contour that temperature is comparatively less than it was in without external magnetic field. Fig. 8.8 represent the current profile of the arc. Comparing with the simulation with Fig. 8.3 we can see that there is a change in the shape of the current near the lower electrodes. Figs. 8.9 and 8.10 represent the corresponding magnetic field profile. The most important result for us is in Fig. 8.11, where we can see that the arc is rotating. Infact one can see that the profile of velocity is changed completely when compared to the Fig. 8.6. From the flow lines we can see that the arc is rotating. Note that there is significant jump in the absolute value of the velocity. Without external magnetic field
8 High Current Arc Simulations in Three Dimensions and Effects of External Magnetic Field

(a) Current density contour for $J = 10 \, A/m^2$.

(b) $XY$ plane cut of current density

Figure 8.8: Current density profile with external magnetic field after time $t = 1.138 \, ms$

(a) $B_x$ magnetic field in $YZ$ plane

(b) $B_z$ magnetic field in $XY$ plane

Figure 8.9: Magnetic field with external magnetic field after time $t = 1.138 \, ms$
8.3 Effects of external magnetic fields

Figure 8.10: $B_y$ with external magnetic field after time $t = 1.138 \, ms$

(a) $\|v\|$ Contour for $\|v\| = 350 \, m/s$  
(b) $\|v\|$ field in XY plane

Figure 8.11: $\|v\|$ field with external magnetic field after time $t = 1.138 \, ms$
we had the maximum velocity of 490 m/s, compared to 689 m/s. Also the maximum velocity is now at the arc roots, instead of at the center of the domain.

### 8.4 Conclusion

We have shown that the equations of MHD are suitable for the full three dimensional computations of the plasma arc for the high current circuit breakers. We have used these equations to generate the arc for a total current 100 kA. We applied an external magnetic field by adding it into the arc and applying it as the boundary conditions. This results in the rotation of the arc due to external magnetic field with significant increase in the velocity.

One of the major obstacle to simulating the arc with the real gas coefficients is the stiffness due to the low values of the conductivity. This forced us to change the the Lundquist number significantly. One way of dealing with this issue can be by using implicit schemes for the MHD simulations.
Bibliography


Bibliography


Bibliography


[53] **Huguenot, P.**, Axisymmetric high current arc simulations in generator circuit breakers based on real gas magnetohydrodynamics models, Diss., Eidgenossische Technische Hochschule ETH Zrich (2008), No. 17625.
Bibliography


# Curriculum Vitae

## Personal details

<table>
<thead>
<tr>
<th>Name</th>
<th>Harish Kumar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date of birth</td>
<td>October 30, 1980</td>
</tr>
<tr>
<td>Place of birth</td>
<td>Rohtak, India</td>
</tr>
<tr>
<td>Citizenship</td>
<td>India</td>
</tr>
</tbody>
</table>

## Education

<table>
<thead>
<tr>
<th>Date</th>
<th>Qualification</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/2004–06/2009</td>
<td>PhD studies in Mathematics at ETH Zürich, Zurich, Switzerland</td>
</tr>
<tr>
<td>08/2001–06/2004</td>
<td>MS in Mathematical Sciences, IISc., Bangalore, India</td>
</tr>
<tr>
<td>07/1998–07/2001</td>
<td>B.Sc.(Hons.) in Mathematics, MD University, Rohtak, Haryana, India</td>
</tr>
<tr>
<td>07/1996–06/1998</td>
<td>High School at JNV Kalo, Jhajjar, Haryana, India</td>
</tr>
</tbody>
</table>