Stony Brook University

Simulation of Magnetohydrodynamic Multiphase Flow

A Dissertation Presented

by

Jian Du

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

August 2007

Stony Brook University

The Graduate School

Jian Du

We, the dissertation committee for the above candidate for the Doctor of Philosophy degree, hereby recommend acceptance of this dissertation.

James Glimm - Dissertation Advisor
Professor
Department of Applied Mathematics and Statistics

Xiaolin Li - Chairperson of Defense
Professor
Department of Applied Mathematics and Statistics

Yongmin Zhang - Member
Professor
Department of Applied Mathematics and Statistics

Roman Samulyak - Outside Member & Co-advisor
Scientist
Computational Science Center
Brookhaven National Laboratory

This dissertation is accepted by the Graduate School.

Lawrence Martin

Dean of the Graduate School

Abstract of the Dissertation

Simulation of Magnetohydrodynamic Multiphase Flow

by

Jian Du

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

2007

We propose a numerical algorithm for the study of magnetohydrodynamics (MHD) of free surface flows at low magnetic Reynolds numbers. The numerical algorithm employs the method of front tracking and the Riemann problem for material interfaces, second order Godunov-type hyperbolic solvers, and the embedded boundary method for the elliptic problem in complex domains. The code is applied for the numerical simulations of free surface/multiphase conductive flows with two different equation of state (EOS) models. First, numerical results of mercury jet distortion within non-uniform magnetic field are compared with theory and good agreements are achieved. Then we simulate the hydro and MHD processes in the liquid metal target for the Muon

Collider/Neutrino Factory, which include mercury jet interacting with proton pulse in longitudinal magnetic field. Strong restriction effect of the magnetic field on the jet expansion speed is observed. We also use the multiscale approach to include the effect of bubble collapsing and proposed that it is one of the causing factors for the surface instabilities of the target. At last, we simulate the expansion of laser-generated plasma plumes in magnetic field and confirmed quantitatively that the plume expansion is directed along magnetic field lines and impeded in other directions.

Key Words: MHD, embedded boundary, front tracking, phase transition.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In Section 1.1 and 1.2, the background of my research work and the governing system of MHD equations are introduced. Two approaches to the simulation of multiphase systems, the heterogeneous and the homogenized models, are briefly described in Section 1.3. The last section is about the Muon Collider Target introduction.

1.1 Background of Research

Computational magnetohydrodynamics, greatly inspired over the last decades by magnetic confinement fusion and astrophysics problems, has achieved significant results. The major research effort has been in the area of highly ionized plasmas. Numerical methods and computational software for MHD of weakly conducting materials such as liquid metals or weakly ionized plasmas have not been developed to such an extent despite the need for fusion research and industrial technologies. Theoretical, computational, and experimental studies of liquid metal MHD have been driven by potential applications of

flowing liquid metals or electrically conducting liquid salts as coolant in magnetic confinement fusion reactors as well as some industrial problems. Weakly ionized plasmas have been studied with respect to their application to tokamak refueling devices [3–5], laser ablation in magnetic fields [6], and other processes in laboratories and in nature.

The existence of moving free material interfaces or complex geometries in many important MHD problems creates major complications for numerical algorithms. The majority of numerical studies of free surface MHD flows are based on semi-analytical treatment of simplified flow regimes. To the best of our knowledge, the only fully numerical treatment of general free surface incompressible liquid flows is implemented in the HIMAG code [7] using the level set algorithm for fluid interfaces, the electric potential formulation for electromagnetic forces, and the incompressible fluid flow approximation. However, strong linear and nonlinear waves and other compressible fluid phenomena such as cavitation are typical features of many practically important free surface MHD regimes in both weakly ionized plasmas and liquid metals interacting with intense sources of external energies. The ablation of solid hydrogen pellets in tokamaks (a proposed tokamak fueling technology) [3, 4], laser - plasma interaction, and the interaction of a liquid mercury jet with proton pulses in target devices for future advanced accelerators [8] are among numerous examples of such MHD problems. To simulate such processes, we propose in the thesis a numerical algorithm applicable to both 2D and 3D and describe its implementation.

1.2 MHD System of Equations & Approximations

The system of MHD equations [9] contains a hyperbolic system of the mass, momentum, and energy conservation equations for the fluid, and a parabolic equation for the evolution of the magnetic field:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} &= -\nabla \cdot (\rho \mathbf{u}), && (1.1)\\
\rho \left(\frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla \right) \mathbf{u} &= -\nabla P + \rho \mathbf{g} + \frac{1}{c}(\mathbf{J} \times \mathbf{B}), && (1.2)\\
\rho \left(\frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla \right) e &= -P \nabla \cdot \mathbf{u} + \rho \mathbf{u} \cdot \mathbf{g} + \frac{1}{\sigma}\mathbf{J}^2, && (1.3)\\
\frac{\partial \mathbf{B}}{\partial t} &= \nabla \times (\mathbf{u} \times \mathbf{B}) - \nabla \times (\frac{c^2}{4\pi\sigma} \nabla \times \mathbf{B}), && (1.4)\\
\nabla \cdot \mathbf{B} &= 0, && (1.5)\\
P &= P(\rho, e). && (1.6)
\end{aligned}$$

Here $\mathbf{u}$, ρ and e are the velocity, density, and the specific internal energy of the fluid, respectively, P is the pressure, $\mathbf{g}$ is the gravitational acceleration, $\mathbf{B}$ is the magnetic field induction, $\mathbf{J} = \frac{c}{4\pi} \nabla \times \mathbf{H}$ is the current density distribution, σ is the fluid conductivity, and c is the speed of light. The magnetic field $\mathbf{H}$ and the magnetic induction $\mathbf{B}$ are related by the magnetic permeability coefficient μ: $\mathbf{B} = \mu \mathbf{H}$.

If the magnetic Reynolds number,

$$Re = \frac{4\pi u \sigma L}{c^2},$$

where L is the typical length scale, is small, and the eddy current induced

magnetic field δB is small compared to the external field B, the full system of MHD equations 1.1 - 1.5 can be simplified. Namely, the time evolution of the magnetic field 1.4 can be neglected, and the current density distribution can be obtained from Ohm's law:

$$\mathbf{J} = \sigma \left(-\nabla\varphi + \frac{1}{c}\mathbf{u} \times \mathbf{B} \right), \tag{1.7}$$

where φ is the electric field potential. Due to the charge neutrality, the potential φ satisfies the following Poisson equation:

$$\nabla \cdot (\sigma \nabla\varphi) = \frac{1}{c} \nabla \cdot \sigma(\mathbf{u} \times \mathbf{B}). \tag{1.8}$$

For a numerical computation, such an approach effectively removes fast time scales associated with the magnetic field diffusion,

$$\tau = \frac{4\pi\mu\sigma L^2}{c^2}.$$

Equation 1.5 is automatically satisfied for an external magnetic field created by a realistic source.

The following boundary conditions must be satisfied at the interface Γ of a conducting fluid with a dielectric medium:

i) the normal component of the velocity field is continuous across the interface;

ii) the pressure jump at the interface is defined by the surface tension T

and main radii of curvature:

$$\Delta P|_{\Gamma} = T\left(\frac{1}{r_1} + \frac{1}{r_2}\right); \tag{1.9}$$

iii) the normal component of the current density vanishes at the interface giving rise to the Neumann boundary condition for the electric potential:

$$\left.\frac{\partial \varphi}{\partial \mathbf{n}}\right|_{\Gamma} = \frac{1}{c}(\mathbf{u} \times \mathbf{B}) \cdot \mathbf{n}, \tag{1.10}$$

where $\mathbf{n}$ is a normal vector at the fluid free surface Γ.

The governing equations 1.1 - 1.3, 1.8, 1.10 form a coupled hyperbolic - elliptic system. It has to be solved in geometrically complex and moving domains since our interest is in free surface flow. The fluid interface is represented as an explicit co-dimension one Lagrangian mesh moving through a volume filling Eulerian mesh. The propagation and redistribution of the interface using the method of front tracking [10, 11] are performed at the beginning of a time step. Then the interface tangling is resolved by grid based reconstruction and interior states are updated by second order hyperbolic solvers such as the Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL) [12]. At the end of the time step, the elliptic system is solved using a finite volume discretization with interface constraints (the embedded boundary method [13]). At last, the interior and front states are updated by adding electromagnetic source terms.

The low magnetic Reynolds number approximation is applicable to moderately conductive media, such as liquid metals or salts, and weakly ionized

plasmas. The tokamak fueling and plasma disruption mitigation through the ablation of solid deuterium pellets (or argon killer pellet for plasma disruptions) produce cold, weakly ionized ablation flow. The applicability of the low magnetic Reynolds number approximations to such problems was justified in [5]. In that paper, a-posteriori estimate of the magnetic Reynolds number, magnetic beta, and eddy current was performed. It was found that the electrostatic (low magnetic Reynolds number) approximation applies to most of the tokamak fueling regimes.

1.3 Models for Multiphase Flow

1.3.1 General Information

An accurate description of cavitation and wave propagation in cavitating and bubbly fluids is a key problem in modeling and simulation of hydrodynamic processes in a variety of applications ranging from marine engineering to high-energy physics. We've used two different equation of state models, the homogeneous model and the heterogeneous model for our simulations. The homogeneous model treats the bubbly flow as a single component mixture and suitable averaging is performed over the length scale which is large compared to the distance between bubbles. In the heterogeneous model, on the other hand, we model a liquid/vapor or liquid-nondissolvable gas mixture as a system of one phase domains (vapor bubbles in a liquid) separated by free interfaces [14]. Next section will give detailed information about the two models.

1.3.2 Homogeneous and Heterogeneous Model

The homogeneous EOS model consists of three branches. The pure vapor branch is described by the polytropic EOS:

$$P = (\gamma_v - 1)E\rho, \tag{1.11}$$

where P is the pressure, ρ is the density, E is the specific internal energy, and γ_v is the adiabatic exponent for the vapor. The liquid branch is described by the corresponding reduction of the stiffened polytropic EOS model [15]

$$P = (\gamma_l - 1)\rho(E + E_\infty) - \gamma_l, \tag{1.12}$$

where γ_l is the adiabatic exponent for the liquid, and P_∞ and E_∞ are two model parameters defining the maximum tension (the maximum value of the "negative pressure" achievable in the liquid) and the energy shift constant, correspondingly. E_∞ can be used to obtain the quantitative agreement of the internal energy of the liquid at normal conditions with experimental data. The two branches are connected by a model for the liquid-vapor mixture:

$$P = P_{sat,l} + P_{vl}\log\left[\frac{\rho_{sat,v}{a_{sat,v}}^2(\rho_{sat,l} + \beta(\rho_{sat,v} - \rho_{sat,l}))}{\rho_{sat,l}(\rho_{sat,v}{a_{sat,v}}^2 - \beta(\rho_{sat,v}{a_{sat,v}}^2 - \rho_{sat,l}{a_{sat,l}}^2))}\right], \tag{1.13}$$

where $\rho_{sat,v}$, $\rho_{sat,l}$, $a_{sat,v}$, $a_{sat,l}$ are the density and the speed of sound of vapor and liquid in saturation points, respectively, $P_{sat,l}$ is the liquid pressure in the

saturation point, β is the void fraction

$$\beta = \frac{\rho - \rho_{sat,l}}{\rho_{sat,v} - \rho_{sat,l}}, \tag{1.14}$$

and the parameter P_{vl} is

$$P_{vl} = \frac{\rho_{sat,v} {a_{sat,v}}^2 \rho_{sat,l} {a_{sat,l}}^2 (\rho_{sat,v} - \rho_{sat,l})}{\rho_{sat,v}{}^2 {a_{sat,v}}^2 - \rho_{sat,l}{}^2 {a_{sat,l}}^2}. \tag{1.15}$$

The expression 1.13 was derived by integrating an experimentally validated model for the sound speed in bubbly mixture. A set of the EOS input parameters, most of which are measurable quantities, allows to fit the two-phase EOS to thermodynamics data for real fluids. And the most important feature of the homogenized isentropic EOS model is the correct behavior of the sound speed in liquid at void fractions ranging from the pure liquid to pure vapor (gas) phases. Figure 1.1 displays a plot of the pressure vs. specific volume for the model.

Since liquid/vapor mixture is treated as a pseudo-fluid that obeys an equation of state of a single component flow, conductivity model is needed for the mixed phase domain. Bruggeman's method (Effective Medium Theory [16]) is used in our simulations. Also the linear model, in which conductivity is linearly proportional to the mass fraction of the liquid phase, is used for the comparison. Suppose the liquid phase has conductivity of σ_l. Vapor phase has conductivity of σ_g and volume fraction of β. Then if the condition $\sigma_l >> \sigma_g$

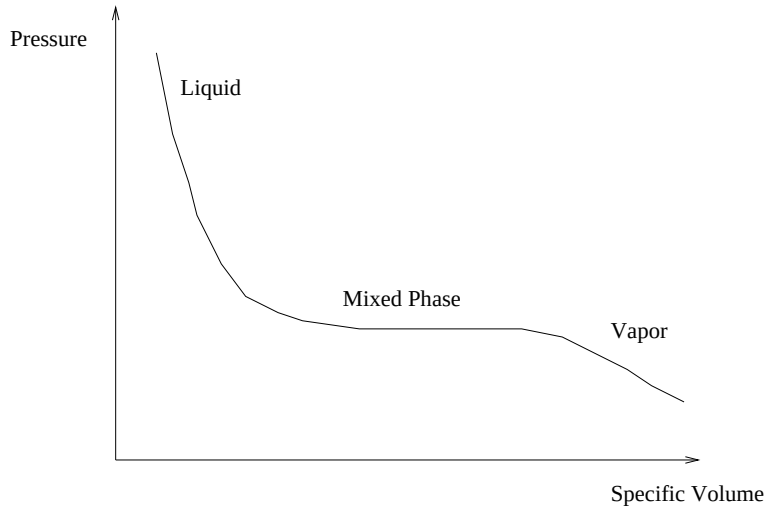


Figure 1.1: Pressure vs. specific volume for the homogeneous model

is satisfied, the conductivity of the mixed phase can be calculated by:

$$\sigma_m = \begin{cases} 0 & \text{if } 1 \geq \beta \geq \frac{2}{3}; \\ \frac{1}{2}(2 - 3\beta) & \text{if } \beta \leq \frac{2}{3} \ . \end{cases} \tag{1.16}$$

The dependence of conductivity for the mixture on vapor volume fraction is shown in Figure 1.2.

One of the main disadvantages of the homogenized EOS model for multi-phase flows is its inability to resolve spatial scales comparable to the distance between bubbles. Averaging of fluid properties will result in an unresolved fine structure of waves that may be critical for understanding the important features of the flow dynamics such as surface instabilities, bubble collapse induced pressure peaks, etc. The heterogeneous model eliminates this deficiency and improves many other thermodynamic and hydrodynamic aspects of the modeling of cavitating and bubbly flows.

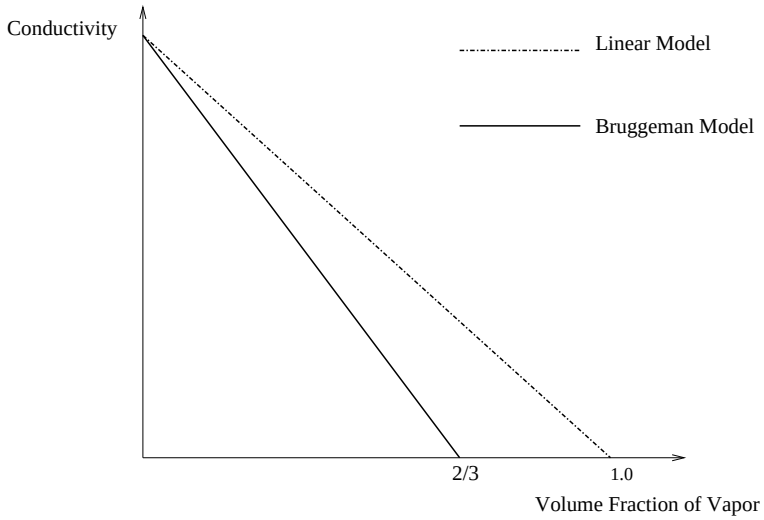


Figure 1.2: Conductivity vs. vapor volume fraction

In the heterogeneous method, we model a liquid – vapor or liquid – non-dissolvable gas mixture as a system of one phase domains (vapor bubbles in a liquid) separated by free interfaces, as shown in Figure 1.3. FronTier, a compressible hydrodynamics code with free interface support, is used to model the behavior of bubble interfaces. The dynamics of the interface comes from the mathematical theory of Riemann solutions, which are idealized solutions of single jump discontinuities for a conservation law. The method makes it possible to resolve spatial scales smaller than the typical distance between bubbles and to model some non-equilibrium thermodynamics features such as finite critical tension in cavitating liquids.

Though computationally intensive, the heterogeneous model is potentially a very accurate technique, limited by only numerical errors. It allows to account for drag, surface tension, and viscous forces as well as the phase transition induced mass transfer. Since characteristic time scales of the

hydrodynamics processes in such a jet are small, we assume that the evolution of cavitation bubbles is mainly due to the expansion (contraction) of the bubble content (mix of vapor and gas). According to the equilibrium thermodynamics approximation, liquid will vaporize when the pressure falls below the corresponding vapor pressure at given temperature. Liquids are able to sustain some amount of tension which depends on their purity. The critical radius of the cavitation bubble is:

$$R_c = \frac{2\sigma}{\Delta P_c}, \tag{1.17}$$

where σ is the surface tension coefficient and ΔP_c is the critical strength of the tensile pressure in the liquid. Initial cavitation bubble sizes in real liquids (for instance, $R_c = 1$ micron for mercury at $P_c = 10$ bar) are close to numerically resolved limits as the FronTier code is equipped with the adaptive mesh refinement. However we frequently use larger initial bubble size, especially for coarser grid computations in large domains. This is effectively equivalent to the insertion of a cavitation bubble at later time.

To create such a nucleus with critical radius R_C, a critical energy $E_C = \frac{16\pi\sigma^3}{3\triangle P_C^2}$ [17] must be deposited into the liquid to break the barrier against nucleation. This critical energy E_C accounts only for surface energy and the gain in volume energy. The energy needed to convert liquid to vapor (heat of vaporization) is neglected, because it is relatively small. Within the homogeneous nucleation theory [17], one can write the nucleation rate J:

$$J = J_0 \exp^{E_C/(k_b T)}, \tag{1.18}$$

per unit volume and per unit time. Here k_b is the Boltzmann's constant, T is the liquid temperature, and J_0 is a factor of proportionality defined as:

$$J_0 = N\left(\frac{2\sigma}{\pi m}\right)^{1/2}, \tag{1.19}$$

where N is the number density of the liquid (molecules/m^3) and m is the mass of a molecule. Thus the nucleation probability Σ in a volume V during a time period t is [18]:

$$\Sigma = 1 - \exp^{\left(-J_0 V t \exp^{(-E_C/(k_b T))}\right)}. \tag{1.20}$$

Equations (1.17) - (1.20) allow to relate a typical volume V, in which a nucleation bubble appears with probability Σ during time t, with the critical pressure. For $\Sigma = 0.5$, the critical pressure is:

$$P_C \cong -\left(\frac{16\pi\sigma^3}{3k_b T \ln(J_0 V t/\ln 2)}\right)^{1/2} \tag{1.21}$$

The numerical time step defines a natural time scale t in equation (1.21), and $V^{1/3}$ defines the spacing between cavitation bubbles. Expressions (1.17) - (1.21), however, agree with experiments only in a relatively narrow thermodynamic region [17]. Therefore, there is some uncertainty in calculating the cavitation rate in numerical simulations if experimental data of the cavitation threshold and concentration of cavitation centers is unavailable for specific experimental conditions, especially if experiments deal with liquids with large amount of impurities due to interactions with intense beams of high energy particles typical for the Neutrino Factory/Muon Collider target.

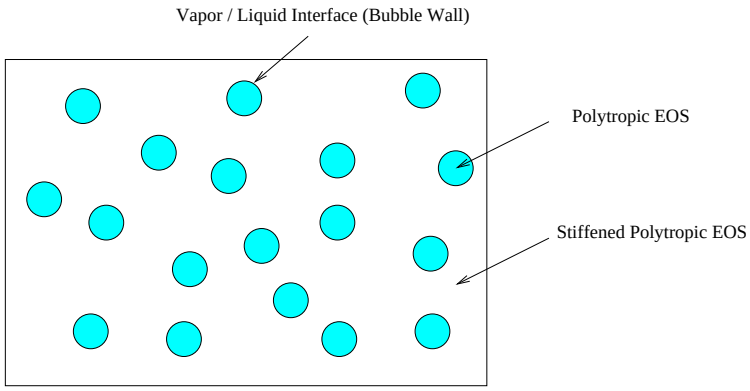


Figure 1.3: Direct numerical simulation of cavitation bubbles

1.3.3 EOS and Conductivity Model of Weakly Ionized Plasma

Partially ionized ablated materials occur in a variety of applications. Some examples include the ablation of deuterium/tritium pellets injected into the magnetically confined plasma (tokamak) and the expansion of laser generated plasma plumes in nanotube synthesis. Magnetic fields introduced in such applications strongly affect the ablation process. The flow of the ablated material is directed along the magnetic field lines and confined in the transverse direction. The ablation rate can also be influenced by the magnetic field. The simulation of the magnetohydrodynamics (MHD) in partially ionized ablated material presents a new challenge in both the modeling and the numerical algorithm. Specifically, the conductivity model and the equation of state are described below.

The processes of dissociation and ionization in diatomic gases in the presence of external energies, resulting in the formation of weakly ionized plasmas, introduce energy sinks and therefore strongly affect the plasma temperature.

This in turn influences the electrical conductivity and MHD processes. In the one temperature, one pressure MHD model, the pressure and specific internal energy of a partially dissociated and partially ionized diatomic gas can be written [19, 20] as:

$$P = \left(\frac{1}{2} + \frac{1}{2}f_d + f_i\right)\frac{\rho k T}{m} \tag{1.22}$$

$$e = \left(\frac{1 - f_d}{2(\gamma_m - 1)} + \frac{f_d + f_i}{\gamma - 1}\right)\frac{kT}{m} + \frac{1}{2}f_d\frac{k\epsilon_d}{m} + f_i\frac{k\epsilon_i}{m}, \tag{1.23}$$

where γ_m and $\gamma = 5/3$ are specific heat ratios for molecules and atoms, respectively, k is the Boltzmann constant, m is the mass of the atom (ion). ϵ_d and ϵ_i are energy associated with dissociation and ionization, respectively. The dissociation $f_d(\rho, T)$ and ionization $f_i(\rho, T)$ fractions are defined as:

$$f_d = (n_a + n_i)/n_t,$$

$$f_i = n_i/n_t,$$

in which $n_t \equiv 2n_g + n_a + n_i = \rho/m$ stands for the total number density of *nuclei*, and n_g, n_a, and n_i denote, respectively, the number densities of gas D_2 molecules, D atoms, and D^+ ions. The dissociation and ionization fractions can be found from Saha equations [21] which can be written (in eV units) as [19, 20]

$$\frac{f_i^2}{1 - f_i} = 3.0 \times 10^{21}\frac{T^{\alpha_i}}{n_t}\exp\left(-\frac{\epsilon_i}{T}\right), \tag{1.24}$$

$$\frac{f_d^2}{1 - f_d} = 1.55 \times 10^{24}\frac{T^{\alpha_d}}{n_t}\exp\left(-\frac{\epsilon_d}{T}\right), \tag{1.25}$$

where $\alpha_i = 3/2$ and the parameter α_d is material dependent parameter. The system of equations (1.22) - (1.25) can be used as an EOS closure for a hydrodynamic system of equations written in $\rho, \mathbf{u}, T$ independent variables, as well as for a simple finite difference discretization of such a system. However most of advanced numerical discretization algorithms, including second order MUSCL type schemes and interface propagation algorithms implemented in the FronTier code [22], are based on solutions of Riemann problems. Solving numerically a Riemann problem requires an ability to calculate the sound speed and integrals of Riemann invariant type expressions along characteristics. For this purpose, expressions for entropy and other thermodynamic properties of the system (1.22) - (1.25) based on the second law of thermodynamics have been derived [5]. To satisfy the requirement of FronTier's hyperbolic solvers and interface propagation routines, the complete set of thermodynamic functions in terms of different pairs of independent variables such as (ρ, e), (ρ, P), (ρ, T) has been used. The corresponding algorithms use numerical solvers for complicated nonlinear algebraic equations. The direct use of such algorithms in hydrodynamic simulations is prohibitively expensive. To speedup the code, tabulated data of thermodynamic functions and their integrals on a fine mesh in the specific domain of interest is created before simulations. Then the table is used for the look-up and interpolation algorithms during the run.

We use the following model for the electrical conductivity of weakly ionized plasma. Values of the conductivity in the transverse and parallel direction to the magnetic field are the same since the ablation electron collision frequency is much higher than their gyro-frequency [4]. Thus σ is given by the

expression [4, 23]

$$\sigma[s^{-1}] = \frac{8.70 \times 10^{13}}{\ln \Lambda \cdot T_e^{-3/2} + 0.054 T_e^{-0.059}(1/f_i - 1)}, \tag{1.26}$$

where f_i is the ionization fraction, T_e is the electron temperature in eV unit, and

$$\Lambda = \frac{1.55 \times 10^{10} T_e^{3/2}}{n_e^{1/2}}, \tag{1.27}$$

where n_e is the electron number density. In the limit $f_i \to 1$, $\sigma \to \sigma_\perp$, which is the transverse Spitzer conductivity.

1.4 Applications: Muon Collider Target Simulation

In order to understand the fundamental structure of matter and energy, an advance in the energy frontier of particle accelerators is required. Advances in high-energy particle physics are paced by advances in accelerator facilities. The aim of the multi-institutional research group Neutrino Factory/Muon Collider Collaboration is to explore the feasibility of a high-energy, high-luminosity muon-muon collider and a neutrino factory (http://www.cap.bnl.gov/mumu). However, several challenging technological problems remain to be solved in the collider design in order to achieve potential advantages of greatly increased particle energies over traditional electron-positron machines (linear colliders). One of the most important problems is to create an effective target able to generate high-flux muon beams. The need to operate high atomic number material targets in particle accelerators that will be able to withstand intense

Figure 1.4: Schematic picture of muon collider target setup

thermal shock has led to the exploration of free liquid jets as potential target candidates for the proposed Muon Collider. The target will contain a series of mercury jet pulses of about 1 cm in diameter and 30 cm in length. Each pulse will be shot at a velocity of 20 m/s into a 15 Tesla magnetic field at a small angle to the axis of the magnetic field. When the jet reaches the center of the magnet it will be hit with a 2 ns proton pulse. Every proton pulse will deposit about 80 J/g of energy in the mercury. The large tension developed within the target induced by the proton pulse greatly exceeds the threshold value which mercury can withstand and cavitation will occur. The schematic setup of the experiment is as shown in Figure 1.4. Without magnetic field, the jet of mercury quickly breaks up after interacting with proton pulse. (Figure 1.5)

Numerical simulations of hydro and MHD processes in the target can reduce the amount of costly experiments and help to optimize target parameters. Such simulations present a challenging problem of computational science. They require mathematical modeling of complex flows undergoing phase

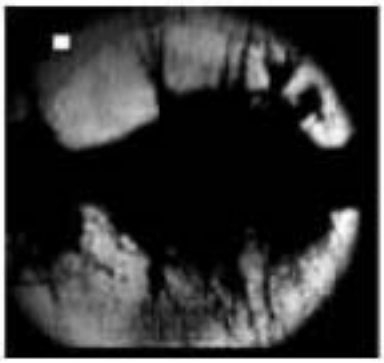


Figure 1.5: Mercury jet breakup after interaction with proton pulse [2]

transitions (cavitation) and numerical methods for solving MHD equations in moving domains with complex geometries. The application of our numerical algorithm to the target simulation will be presented in the following chapters.

Chapter 2

Numerical Algorithm and Implementations

2.1 Grid Based Front Tracking

Front tracking is an adaptive computational method in which a lower dimensional moving grid is fit to and follows distinguished waves in a flow. Tracked waves explicitly include jumps in the flow state across the waves and keep discontinuities sharp. A key feature is the avoidance of finite differencing across discontinuity fronts and thus the elimination of interfacial numerical diffusion including mass and vorticity diffusion [10, 11]. Front tracking is implemented in FronTier [22, 24], a multiphysics code which is capable of tracking and resolving topological changes of geometrically complex interfaces in two and three space dimensions. Details of the front tracking method and the structure of the FronTier code are described in the above mentioned papers. In this section, we will describe only details of the algorithm specific to the MHD system.

2.1.1 Interface Propagation

For free surface MHD flows, we are interested in tracking only fluid interfaces which are contact discontinuity curves of the corresponding Riemann problem [25]. The interface propagation consists of normal and tangential propagation of each interface point. Since the tangential propagation in the MHD case is essentially equivalent to the algorithm described in [22], we will concentrate here only on the algorithm for the normal propagation of interface points.

In the operator splitting scheme, the system of equations (1) - (3) can be considered as a pure hydrodynamic system in an external field given by the Lorentz force. Since this force depends on the material dynamics and properties, the algorithm for the propagation of the interface in the normal direction is slightly different from that for the gravity force [22]. The algorithm solves a generalized Riemann problem for the projection of the flow equations onto the direction normal to the front at the point being propagated. The projection of the system (1) - (3) into the normal direction $\mathbf{n}$ yields the following one dimensional system

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} + \frac{\partial \rho v_N}{\partial \mathbf{n}} + \frac{\alpha n_0}{r}\rho v_N = 0, \\
&\frac{\partial \rho v_N}{\partial t} + \frac{\partial \left(\rho v_N^2 + P\right)}{\partial \mathbf{n}} + \frac{\alpha n_0}{r}\rho v_N^2 = \rho g_N + \frac{1}{c}(\mathbf{J}\times\mathbf{B})_N, \\
&\frac{\partial \rho v_T}{\partial t} + \frac{\partial \rho v_N \mathbf{v}_T}{\partial \mathbf{n}} + \frac{\alpha n_0}{r}\rho v_N \mathbf{v}_T = 0, \\
&\frac{\partial \rho E}{\partial t} + \frac{\partial \left(\rho E v_N + P v_N\right)}{\partial \mathbf{n}} + \frac{\alpha n_0}{r}\left(\rho E v_N + P v_N\right) = \rho g_N v_N.
\end{aligned} \tag{2.1}$$

Here n_0 is the r component of the normal vector $\mathbf{n}$, $\partial/\partial\mathbf{n} = \mathbf{n} \cdot \nabla$ is the directional derivative in the direction $\mathbf{n}$, $A_N = \mathbf{A} \cdot \mathbf{n}$ is the normal component of a vector field $\mathbf{A}$ and $A_T = \mathbf{A} - A_N\mathbf{N}$ is the corresponding tangential component. The parameter α is equal to 1 for a cylindrical coordinate system, axially symmetric with respect to z axis, 2 for a spherical coordinate system, and 0 otherwise. The implementation of geometric source terms corresponding to the cylindrical and spherical systems of coordinates is given in [22].

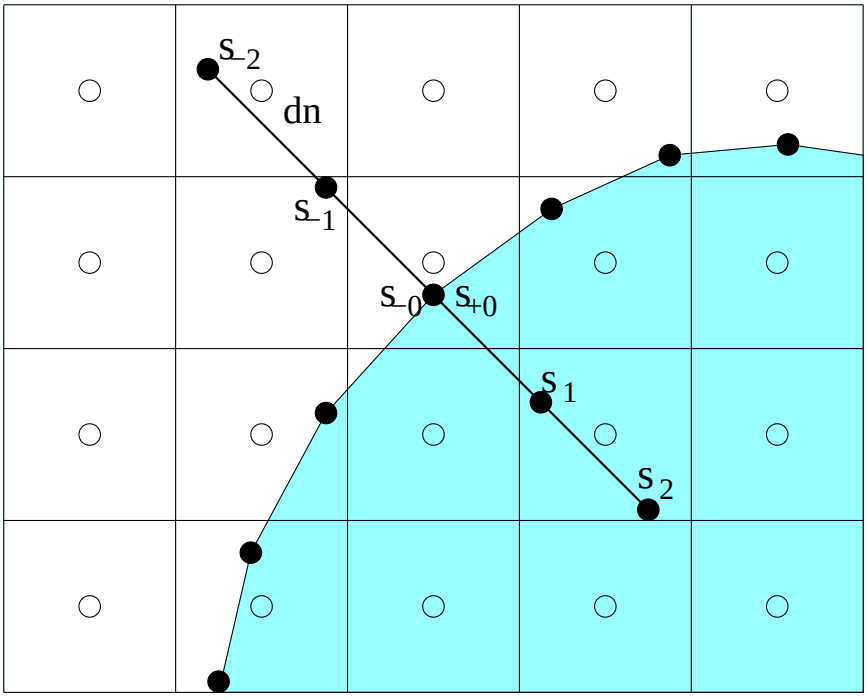


Figure 2.1: Schematic of a stencil for the normal point propagation algorithm.

A 5-point stencil for the point propagation algorithm is schematically shown in Figure 2.1. The algorithm has three main steps: slope reconstruction to compute approximations to the flow gradients along the normal line, prediction using the Riemann problem solution, and correction to account for

the flow gradients on both sides of the front and to include geometric and body terms. The reconstruction step is standard and used in many shock capturing methods (see, for example, [22] and references therein). In the prediction step (see Figure 2a), we solve the Riemann problem with states s_{-0} and s_{+0} to calculate the interface velocity W_0 at the beginning of the time step. Using this velocity, we can estimate the position of the interface $x_{\rm I}$ at the end of the time step. The middle states of the solution of the Riemann problem also provide the interface states $s_{-\rm I}$ and $s_{+\rm I}$ at the new interface position $x_{\rm I}$.

The correction step starts with obtaining states connected by characteristics with the states at the predicted interface position. Namely, we trace back the incoming characteristics from the predicted new front position using the velocity and sound speeds computed from the Riemann problem, and use the slope reconstruction algorithm to approximate the states s_f and s_b at the feet of characteristics (Figure 2b). The correction to the final interface states can be obtained by integrating iteratively the characteristic form of the system (11) along both characteristics coming to the interface. This procedure, as it was shown in [22], can be replaced by the following sequences of Riemann problems. The Riemann problem with the input states $s_{\rm f}$ and s_{-0} is solved and the right wave state of the solution, $s_{\rm ll}$, approximates the wave incoming on the contact at time $t_0 + dt$ from the left. Correspondingly, the Riemann problem with the input states $s_{\rm b}$ and s_{+0} is solved and the left wave state of the solution, $s_{\rm rr}$, approximates the wave incoming on the contact at time $t_0 + dt$ from the right. The states $s_{\rm ll}$ and $s_{\rm rr}$ are then modified by the action of the Lorentz force which was computed at the end of the previous time step.

The resulting states s_{llm} and s_{rrm} approximate the state of the flow incoming on the interface at the end of the time step. In most practical calculations only one, the left or right state has to be modified by the electromagnetic terms since the conducting fluid is usually only on one side of the interface.

Finally, the Riemann problem with the input state s_{llm} and s_{rrm} is solved to obtain approximations of the left and right states at the front, s_{l} and s_{r}, and the front velocity V at the time $t_0 + dt$. The procedure is illustrated in Figure 2c. Assuming that the acceleration of the interface is constant during the time step, the interface velocity *during the time step* is approximated as $W = (W_0+V)/2$. However if strong waves are not present in the vicinity of the interface, we simplify the algorithm and approximate the final interface states by solving the Riemann problem with the input states s_{fm} and s_{bm} obtained from states s_{f} and s_{b} by the action of the Lorentz force, as shown in Figure 2d. This reduces the computational time by eliminating two relatively expensive Riemann problem solving steps for every interface point.

2.1.2 Interface Reconstruction

Two techniques for the redistribution of interfaces and resolving their topological changes, the grid free and grid based tracking, have been developed [24]. In the first method, interface points are always independent of the rectangular grid while in the second method, the interface points are formed by the intersection of the interface with the rectangular grid lines. Since the first algorithm is more accurate and sufficiently robust, especially in 2D, we use it for the interface propagation in the hyperbolic part of the MHD algorithm.

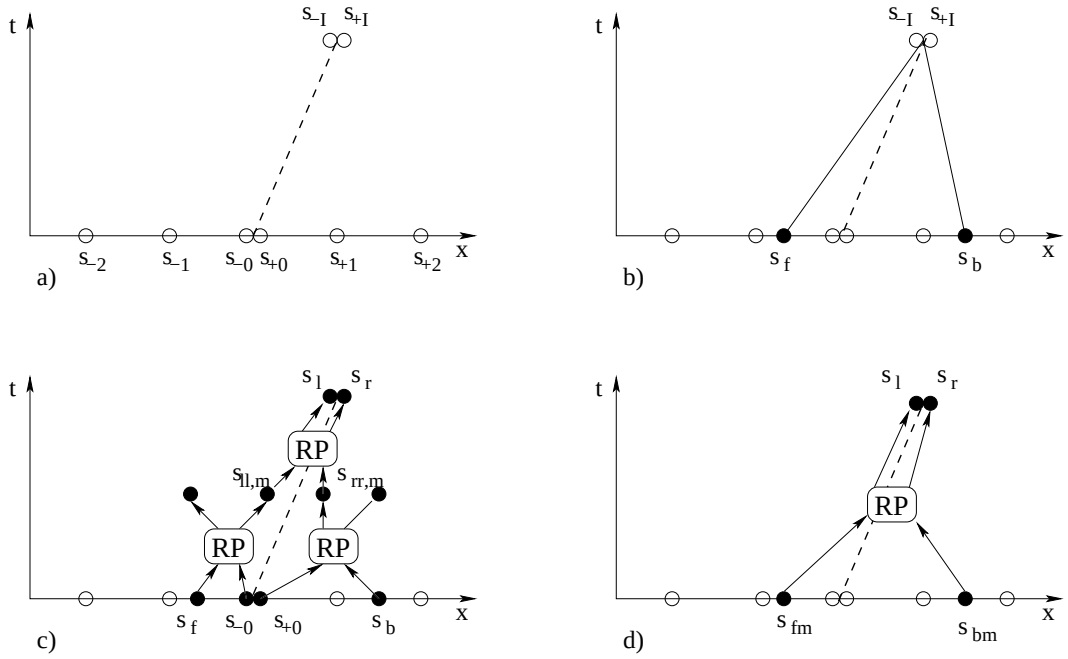


Figure 2.2: Schematic of the normal point propagation algorithm. RP denotes the procedure of solving the Riemann problem and extracting of middle states from the solution.

We always transform the interface to the grid based one at the beginning of the elliptic time step since such an interface ideally suits the finite volume discretization technique for the Poisson equation described in the next section. In many practical applications, it is sufficient to solve the elliptic problem once per several hyperbolic time steps. The reconstruction employs the micro-topology within each rectangular grid block cell on the specified lattice. The grid based reconstruction is divided into three steps [24]:

1. Compute the crossings between the interface and the grid cell edges.

2. Determine components at the grid block corners and eliminate inconsistent crossings.

3. Reconstruct a new interface by using the remaining consistent crossings.

The reconstruction is based on the following three hypotheses:

1. At most two fluid components intersect any individual cell in the reconstruction lattice.

2. Each cell edge has at most one interface crossing.

3. The portion of each cell edge that lie on one side of the reconstructed interface forms a connected set.

The first hypothesis is valid for the two fluid flows. The second says that the reconstructed interface cross each cell edge at most once. The third one implies that if two corners of a cell lie in the same fluid component, then the entire edge connecting those corners also remains in the same fluid.

2.1.3 Interior States Update

The final phase of the hyperbolic time step update consists of computing new states on the rectangular spatial grid. Several different shock capturing methods have been implemented in FronTier. They include both directionally split MUSCL [12] type schemes such as the Piecewise Linear, Piecewise Parabolic Method [26], a second order MUSCL scheme developed by I. L. Chern, and an unsplit MUSCL scheme [27]. An exact and several approximate Riemann solvers are available for use by these methods.

In FronTier code, a connected region in the domain separated by the interface is represented by a component type. Therefore, each grid node is associated with a specific component in addition to the state variables. Computations near the fluid interface use ghost cells [28, 29] to avoid stencils crossing the interface, keeping the different fluid computations entirely separate. The ghost cell method avoids the interpolation across the interface and therefore effectively keep the sharp front profile.

At the end of the elliptic step, the interior states within the region occupied by the conducting materials are modified by adding the Lorentz force term.

2.2 Embedded Boundary Elliptic Solver

2.2.1 Principles

The embedded boundary method is based on the finite volume discretization in grid blocks defined by the rectangular Cartesian grid and the interface.

The solution is treated as a regular block centered quantity, even when these centers are outside of the domain. However the gradient of the potential and the right hand side are located in geometrical centers (centroids) of partial grid blocks cut by the interface [13]. This treatment has the advantages of dealing with geometrically complex domains and ensures second-order accuracy of the solution.

We will describe the method and implementation for the MHD elliptic problem, namely the Poisson's equation (8) with Neumann boundary condition (10) for both 2D and 3D. The regular grid block is a square in 2D and a cube in 3D, and the component elements of each block are either 2D block edges or 3D surfaces. Using the divergence theorem and integrating $\nabla\varphi$ over the control volume, the differential operator can be discretized as

$$(L\varphi)_{\Delta_i} = \frac{1}{V_i}(\sum_j \mathbf{F}_j \cdot \mathbf{n}_j S_j) \tag{2.2}$$

V_i is the control volume. Each block element has size S_j and unit normal $\mathbf{n}_j$, with flux $\mathbf{F}_j$ crossing its geometric center. For full block elements (not cut by the boundary), $\mathbf{F}_j$ is obtained by the centered difference while the flux across partial block elements is obtained using a linear interpolation (in 2D) or bilinear interpolation (in 3D) between centered difference fluxes in adjacent blocks. As it was shown in [13, 30], the interpolation of fluxes through edges of partial cells is necessary for stability and second order accuracy near the interface. In 2D, the flux across the center g of the partial edge ef is obtained using the linear interpolation between the fluxes $\mathbf{F}_j$ and $\mathbf{F}_{j+1}$, the finite differences

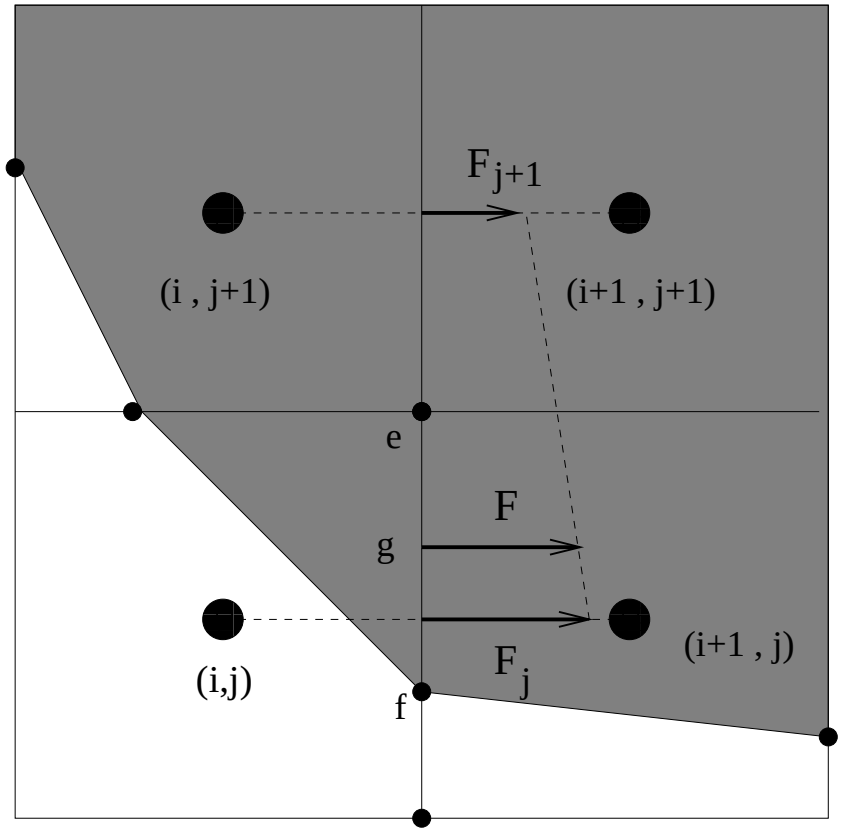


Figure 2.3: Linear flux interpolation

of values in the centers of the corresponding regular grid blocks(Figure 2.3). In 3D, the flux across the centroid P of the surface ABC, which is the partial front surface element of the grid block (i, j, k) (the computational domain is above the interface), is obtained using the bilinear interpolation between fluxes across the points G,D,E and F, which are centroids of the regular block surface elements(Figure 2.4). The flux at the domain boundary (interface) is given by the Neumann condition.

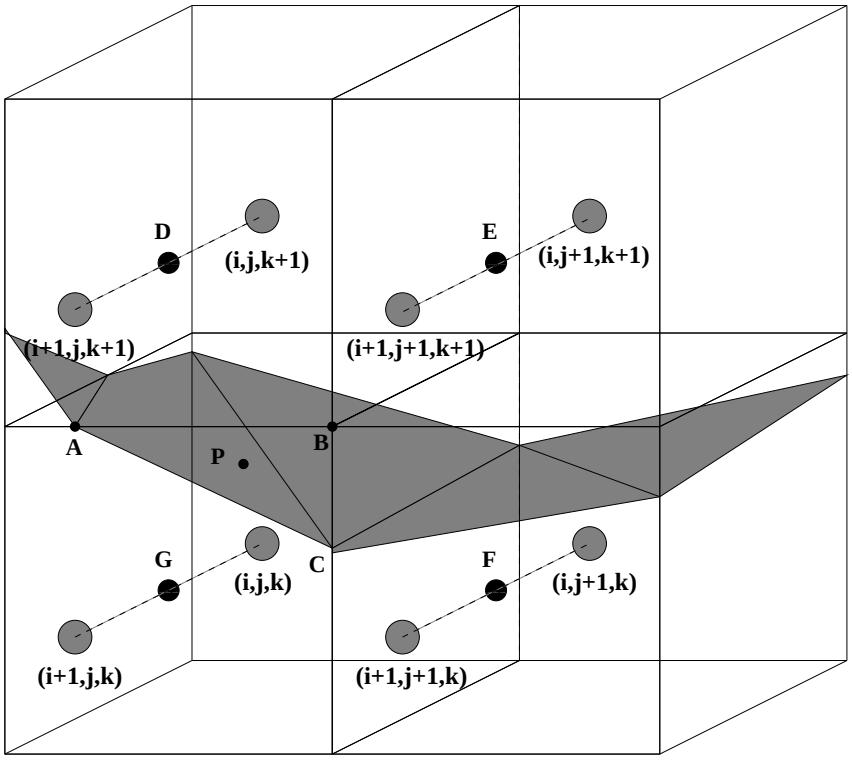


Figure 2.4: Bilinear flux interpolation

2.2.2 Code Structure

In order to implement the embedded boundary method, the interface is reconstructed using its intersections with grid lines. The following assumptions and simplifications are made:

1. The maximum number of intersection of each grid block edge with the boundary curve or surface is one.
2. The elliptic problem domain within each grid block forms a connected set.
3. The positions of the boundary points are adjusted to remove partial blocks with volumes less than a certain preset value.

The first and second assumptions are generally satisfied when the curvature of the interface is not too large or the mesh is sufficiently refined. The third one is necessary since blocks of arbitrary small volumes introduce large numerical errors and increase the condition number of the linear system resulting from the discretization [13].

The summary of the algorithm implementation is as follows.

(1) The elliptic domain boundary is constructed using intersection points of the grid free interface with grid lines. Using information from the reconstruction, types of material components of both grid points and block centers are also properly set. All regular grid blocks are divided into three types: INTERNAL, PARTIAL, and EXTERNAL, which means completely within, partially within (cut by the interface), and completely outside of the computational domain for the elliptic problem.

(2) The number of blocks marked as PARTIAL or INTERNAL is counted, and the total size of the linear system is set. A 2D or 3D matrix is set to record the global indices of the counted blocks, while the indices of the EXTERNAL blocks are set to be negative. In parallel computing, two buffer layers of the index matrix are passed between neighboring processors for two purposes: to form a local to global index mapping and to do a quadric interpolation of the potential gradient near the intersection of the subdomain boundary with the interface.

(3) For each block marked as PARTIAL, all block elements are also divided into three types similar as above. The edge centers and lengths are stored for 2D block elements, and surface centroids and areas are stored in 3D. Take the 3D case for example. Based on the four corner components type on each of the six block surfaces, they are marked as type PARTIAL(1 3 black filled corners), INTERNAL(four black filled corners), and EXTERNAL(no black filled corner), as shown in Figure 2.5. For the surfaces marked as PARTIAL, only three configurations are possible based on previous assumption: triangle, trapezoid, and pentagon. Then with the information of the positions of the corners and the crossings, the surface area and centroid coordinates is stored.

(4) A 9-point stencil is set to calculate fluxes across the control volume $BADEF$, as shown in Figure 2.6, where the elliptic problem domain is the shaded region, and filled circles represent locations where the potential is defined. According to the expression of flux $\mathbf{F}_j = \sum_{m,n} c(m,n)\varphi(m,n), (m,n = 0,1,2)$, we define a 3×3 matrix C with matrix elements $c(m,n)$ representing the coefficient of φ centered at (m,n). Therefore $\varphi(1,1)$ is always the po-

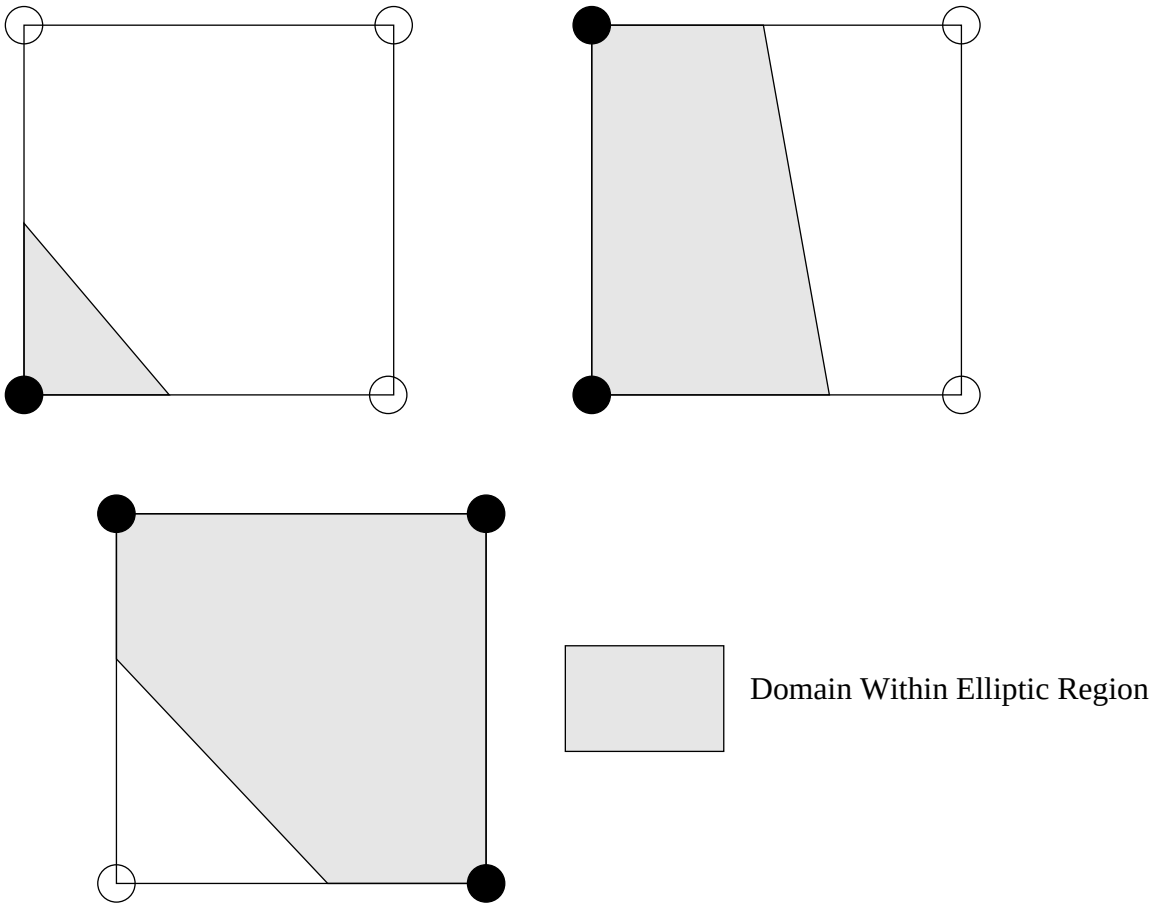


Figure 2.5: 3D block surface configurations

tential located within the control volume. Suppose a Cartesian coordinate is constructed with basis vector $\mathbf{e}_i (i = 0, 1)$ and origin $(0, 0)$ as shown in Figure 2.6, we further denote $c(m, n)$ as $c(\mathbf{V})$, where the vector $\mathbf{V}$ has components m and n. The vector $\mathbf{r}$ is drawn from the regular block center containing the control volume to the center of the block element on which the flux is to be integrated. Then $\mathbf{e'}_i = sign(\mathbf{r} \cdot \mathbf{e}_i)\mathbf{e}_i$ gives orientational information of $\mathbf{r}$. Let $\mathbf{e}$ be the vector whose entries are all ones, the stencil values for the linearly interpolated flux in d-direction are:

$$c(\mathbf{e}) = \frac{a-1}{h_d}; \quad c(\mathbf{e} + \mathbf{e'}_d) = \frac{1-a}{h_d}$$

$$c(\mathbf{e} + \mathbf{e'}_{d'}) = \frac{-a}{h_d}; \quad c(\mathbf{e} + \mathbf{e'}_d + \mathbf{e'}_{d'}) = \frac{a}{h_d}$$

where $d', d = 0,1$ and $d' \neq d$, h_d is the grid spacing in the direction d, and $a = \frac{|\mathbf{r} \cdot \mathbf{e}_{d'}|}{h_{d'}}$ is the block element aperture.

In 3D, a 27-point stencil is used and the coefficients form a $3 \times 3 \times 3$ matrix. Using the similar notations as above, the bilinearly interpolated flux in the direction d has the stencil values:

$$c(\mathbf{e}) = \frac{-(1-a)(1-b)}{h_d}; \quad c(\mathbf{e} + \mathbf{e'}_d) = \frac{(1-a)(1-b)}{h_d}$$

$$c(\mathbf{e} + \mathbf{e'}_{d'}) = \frac{a(b-1)}{h_d}; \quad c(\mathbf{e} + \mathbf{e'}_{d''}) = \frac{(a-1)b}{h_d}$$

$$c(\mathbf{e} + \mathbf{e'}_d + \mathbf{e'}_{d'}) = \frac{a(1-b)}{h_d}; \quad c(\mathbf{e} + \mathbf{e'}_d + \mathbf{e'}_{d''}) = \frac{(1-a)b}{h_d}$$

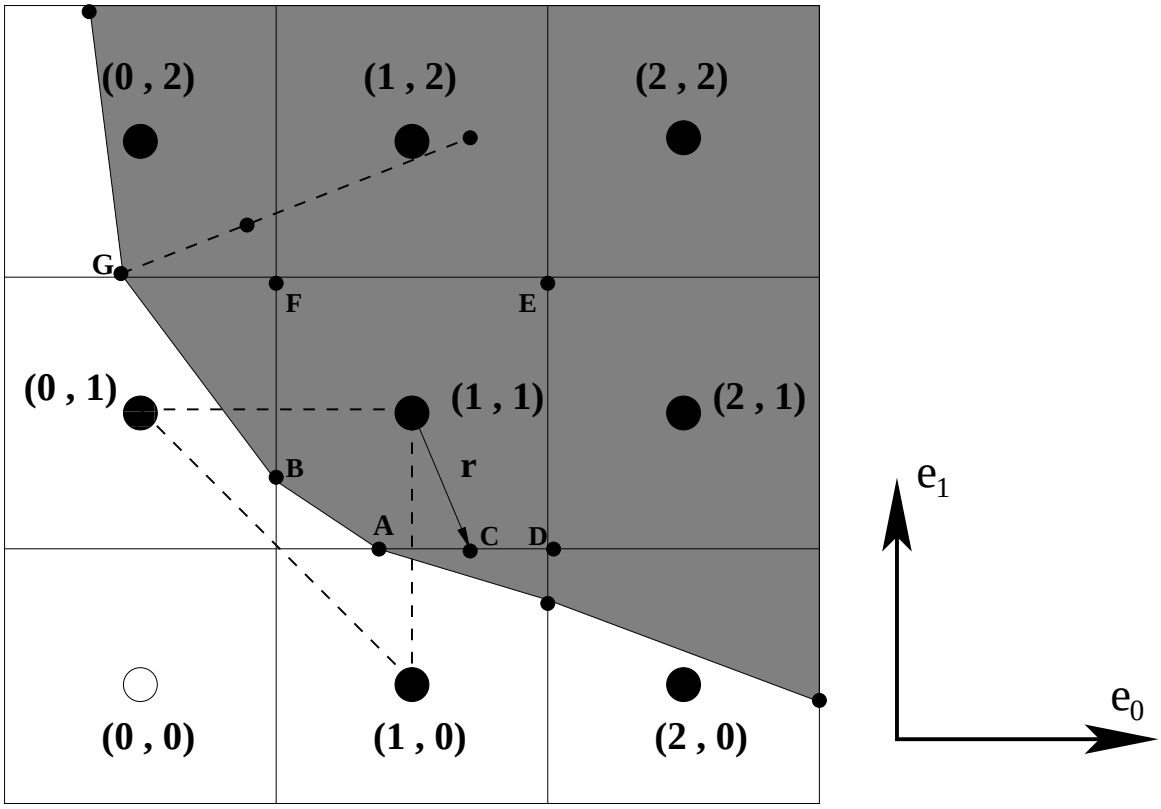


Figure 2.6: Stencil setting for a partial grid block.

$$c(\mathbf{e}+\mathbf{e'}_{d'}+\mathbf{e'}_{d''})=\frac{-ab}{h_d};\quad c(\mathbf{e}+\mathbf{e'}_{d}+\mathbf{e'}_{d'}+\mathbf{e'}_{d''})=\frac{ab}{h_d};$$

where $d, d', d'' = 0, 1, 2$ and $d \neq d' \neq d''$. $a = \frac{|\mathbf{r}\cdot\mathbf{e}_{d'}|}{h_{d'}}, b = \frac{|\mathbf{r}\cdot\mathbf{e}_{d''}|}{h_{d''}}$.

(5) Substituting $\mathbf{F}_j = \sum_{m,n} c(m,n)\varphi(m,n)$ or $\sum_{m,n,l} c(m,n,l)\varphi(m,n,l)$ into the equation (12) and summing up fluxes through all elements of each PARTIAL block, the coefficient at each stencil point is set and added to the global matrix. Since the right hand side in equation (10), which must be evaluated at the centroid of the partial block, has the divergence form of a vector field ($\nabla\cdot(\mathbf{u}\times\mathbf{B})$), the divergence theorem can also be applied to replace the divergence with the finite volume integration of the flux of ($\mathbf{u}\times\mathbf{B}$). This cancels the gradient of potential and the flux of $\mathbf{u}\times\mathbf{B}$ in the normal direction to the boundary since they are equal by the boundary condition. Note that from equation (8), both the stencil values and the right hand side is multiplied by the fluid conductivity σ, which is evaluated at the center of block elements. There is no need to calculate the size of the control volume since it appears as a denominator on both sides. Five or seven point finite differences are used for the INTERNAL grid blocks.

(6) The resulting linear system $A\mathbf{x} = \mathbf{b}$ is solved. We use preconditioners and iterative solvers implemented in PETSc [31] libraries. Then the gradient of the potential is calculated at all PARTIAL and INTERNAL block centers, even if these centers are outside of the elliptic domain. Either the centered difference or quadric interpolation is used to maintain the second order accuracy. For example, the x-derivative of the potential in the point $(1,1)$, $\varphi_x(1,1)$ (see Figure 2.6), is easily calculated by the centered differences of $\varphi(0,1)$ and

$\varphi(2,1)$. However the quadric fitting is required to calculate φ_x in the point $(0,2)$ or $(0,1)$: $\varphi_x(0,1)$ is obtained as the x-derivative of the quadric curve, which interpolates potential values $\varphi(0,1)$, $\varphi(1,1)$, and $\varphi(2,1)$. We also calculate the gradient of the potential at the interface points as it is needed for the Riemann solver described in the previous section. If an interface point is located inside a triangle between regular grid block centers for which the gradient of the potential is known (such as the interface points A and B in Figure 2.6), a triangular interpolation is used to calculate the gradient of the potential in the interface point. If the interface point is outside of such a triangular (point G), a normal to the interface is constructed, and the gradient of the potential in the interface point is obtained similarly to the quadric interpolation procedure described above.

(7) The interior momentum states are modified by adding the Lorentz force term. Notice that if the hyperbolic system is written in terms of conserved variables, namely the density, momentum, and total energy density, the last variable remains unchanged. It is easy to verify that the external magnetic field does not change the total energy of the system, and the increase of the internal energy due to Joule's heat is canceled by the decrease of the kinetic energy due to the Lorentz force.

2.3 Validation and Performance

2.3.1 Elliptic Problem Validation

Since the described elliptic technique is new to the method of front tracking and the FronTier software, we have validated it using analytical solutions of a simple elliptic problem. Namely, we solve numerically the Neumann problem:

$$\Delta\varphi = \begin{cases} \mathbf{f} \\ \frac{\partial\varphi}{\partial\mathbf{n}}\big|_{\Gamma} = g \end{cases}, \tag{2.3}$$

assuming that the exact solution is $\varphi = e^{k_1x^2+k_2y^2+k_3z^2}$. $\mathbf{f}$ and g are obtained by differentiating the exact solution. The problem is solved in the irregular 2D domain and a perturbed spherical 3D domain, as shown in Figure 2.7 and Figure 2.8.

We analyze the convergence of the gradient of the solution as the Neumann boundary problem contains an arbitrary constant. $\nabla\varphi$ also corresponds to physically measurable quantities in applications. The convergence rate R is

$$R = \log\Big(\frac{\parallel e_{n+1} \parallel}{\parallel e_n \parallel}\Big) \Big/ \log\Big(\frac{h_{n+1}}{h_n}\Big),$$

where e_{n+1} and e_n are error vectors corresponding to the grid spacing h_{n+1} and h_n. The L_2 norm is used in our calculations. Tables 1 - 3 contain data on the solution error, convergence rate, CPU time, and the number of iterations of the linear solver necessary to obtain the required tolerance. We observe that

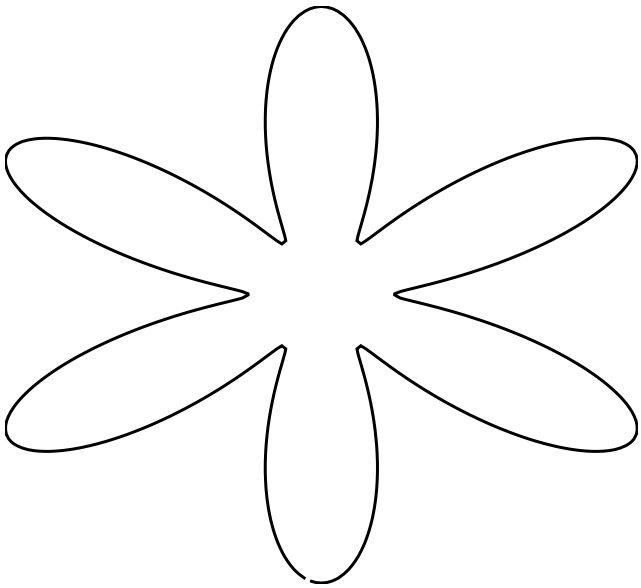


Figure 2.7: 2D Computational domains for the elliptic problem

Figure 2.8: 3D computational domains for the elliptic problem

Mash size	Error	Conv. Rate	CPU Time, s	Iterations
32×32	1.110e-3	N/A	0.012	24
64×64	2.477e-4	2.164	0.025	51
128×128	5.332e-5	2.190	0.144	112
256×256	1.339e-5	2.124	1.321	283

Table 2.1: Convergence and timing results for the x-derivative of the solution in 2D. Error is measured by L_2 norm of φ_x, and the relative tolerance for the iterative linear solver is 10^{-5}.

Mesh size	Error	Conv. Rate	CPU time, s	Iterations
64×64	9.094e-05	N/A	0.087	44
128×128	2.013e-05	2.175	0.389	98
256×256	4.798e-06	2.122	2.223	264
512×512	1.776e-06	1.893	15.445	500

Table 2.2: Convergence and timing results of the gradient of the solution in 2D. Error is measured by L_2 norm of $\nabla\varphi$, and the relative tolerance for the iterative linear solver is 10^{-5}.

the solution gradient is second order accurate. With the setting of a Dirichlet point and eliminating the solution constant associated with the Neumann boundary, we find that the computed solution is also second order accurate. All 2D test calculations were performed on 4 processors using a 2×2 domain decomposition, and 3D tests used a $2 \times 2 \times 2$ domain decomposition on 8 processors.

Mesh size	Error	Conv. Rate	Iterations
$32 \times 32 \times 32$	1.316e-03	N/A	42
$64 \times 64 \times 64$	3.179e-04	2.050	76
$128 \times 128 \times 128$	8.046e-05	2.016	144

Table 2.3: Convergence results of the gradient of the solution in 3D. Error is measured by L_2 norm of $\nabla\varphi$, and the relative tolerance for the iterative linear solver is 10^{-5}.

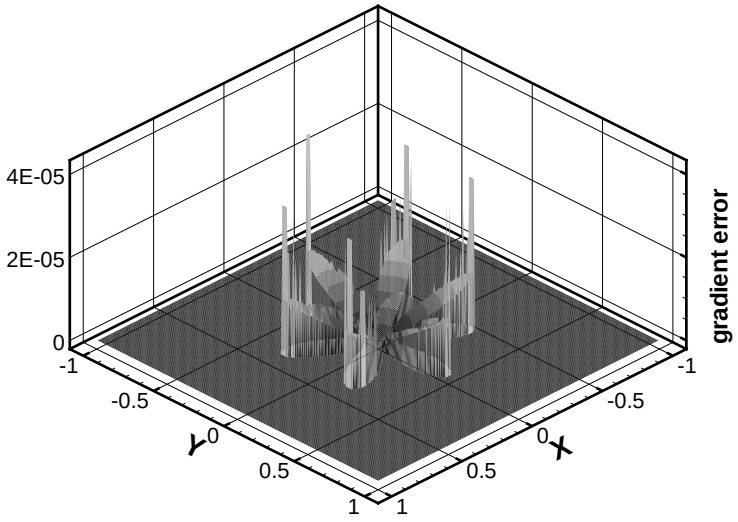


Figure 2.9: Norm of the gradient error by the EB method on a 256×256 grid.

2.3.2 Comparison with Mixed Finite Element Method [1]

The Mixed Finite Element Method (MFEM) uses an unstructured triangular mesh and instead of solving the second order elliptic equation, it solves two first order equations and gives the potential and flux at the same time. Thus, it solves

$$\vec{q} = -a\nabla\phi$$

$$\nabla \cdot \vec{q} = f$$

For the mixed finite element method, two function spaces are needed: one scalar space for the potential ϕ and one vector space for the flux $\vec{q}$.

The unknowns are potentials on the elements and the flux on the edges. To reduce the problem to a smaller one, the mixed-hybrid finite element is modified by introducing a Lagrangian multiplier on the edges. Chavent and Roberts [32] give in detail an implementation using rectangle elements. The same 2D test problem as in Section 2.3.1 are solved by the mixed-hybrid finite element with three different basis functions for flux: the RT0 (Raviart-Thomas space of degree zero), the RT1 (Raviart-Thomas space of degree one) and BDM1(Brezzi-Douglas-Marini space of degree one). A mesh generation method similar to Quadtree/Octree based algorithm is used to generate the unstructured mesh[33]. Refer to [34] for more details. We did a comparison study of the two method: Embedded Boundary Method and Mixed Finite Element Method for the solution of elliptic boundary value problems. Both the solution errors and performances of the two methods are compared (Table 2.4 shows gradient error for the whole domain while Table 2.5 shows gradient only on the domain boundary). Figure 2.11 and Figure 2.12 shows the L2 error of the solution gradient from EBM and BDM1 respectively. And Figure 2.10 is part of the unstructured mesh.

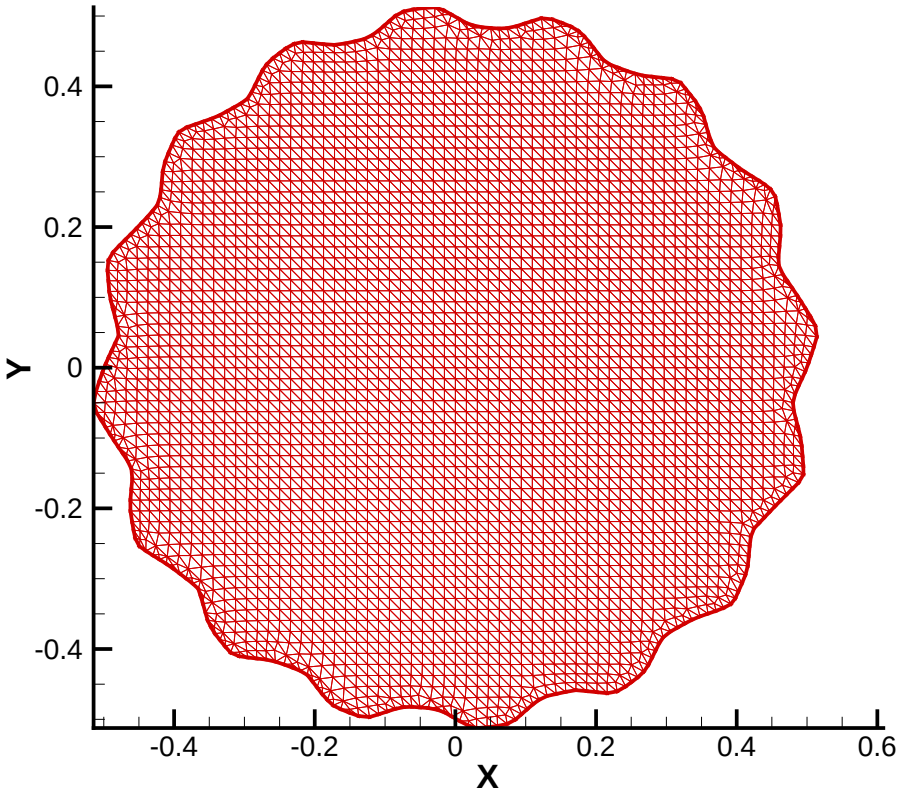


Figure 2.10: Details of unstructured mesh for 128x128 grid

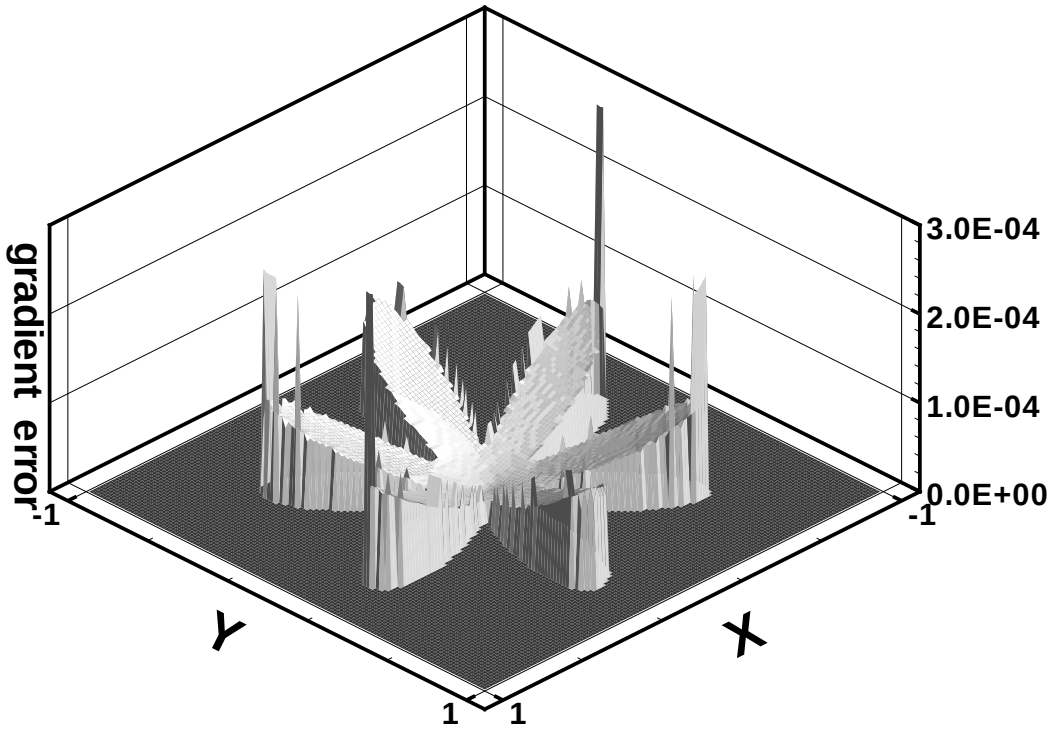


Figure 2.11: Norm of gradient error by EBM using 128x128 grid

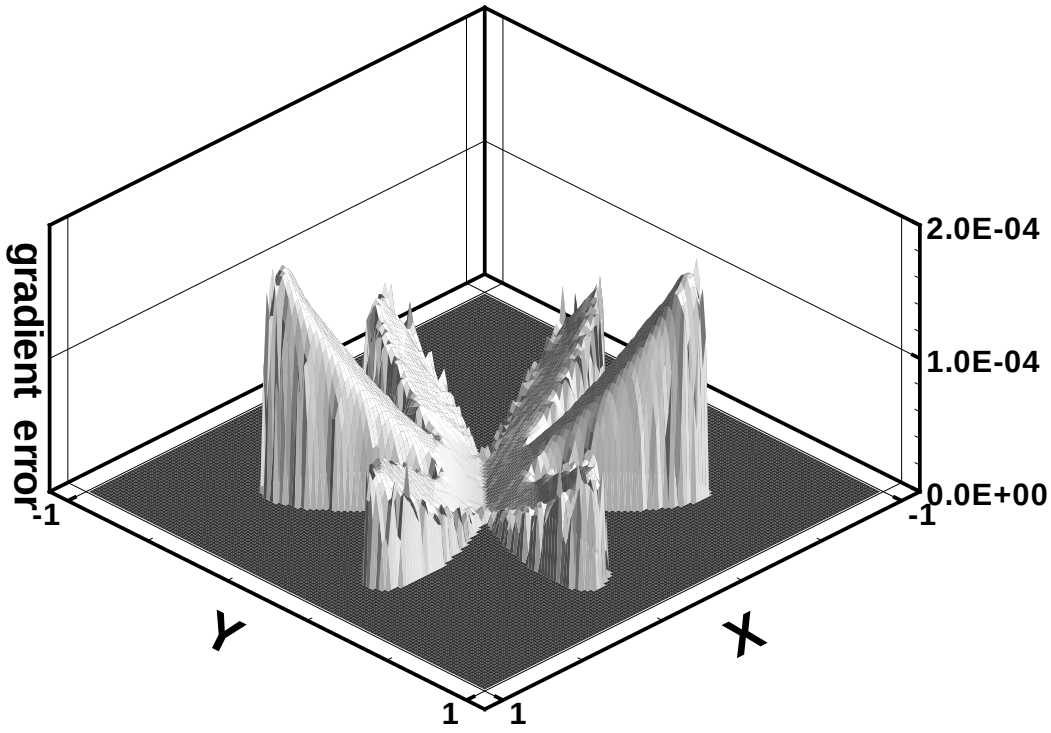


Figure 2.12: Norm of gradient error by BDM1 using 128x128 grid

Table 2.4: Convergence and Timing Study using Uniform Mesh for the domain in fig. 2.7

Mesh Size	EBM				
	error	ratio	time	iterations	unknowns
64×64	2.110753e-04	N/A	0.022319	43	1008
128×128	5.779287e-05	1.869	0.164115	91	4008
256×256	1.472989e-05	1.920	1.438516	209	15738
512×512	3.641386e-06	1.952	10.398294	360	61967
Mesh Size	RT0				
	error	ratio	time	iterations	unknowns
64×64	1.806532e-03	N/A	0.283352	115	3177
128×128	1.142133e-03	0.661	1.624428	218	12341
256×256	6.140341e-04	0.895	11.236136	415	47870
512×512	3.166839e-04	0.955	79.363582	770	187229
Mesh Size	BDM1				
	error	ratio	time	iterations	unknowns
64×64	1.695040e-04	N/A	0.832989	151	6354
128×128	5.857866e-05	1.533	5.360611	278	24682
256×256	1.641488e-05	1.835	33.627868	461	95740
512×512	4.311759e-06	1.929	320.587307	1185	374458
Mesh Size	RT1				
	error	ratio	time	iterations	unknowns
64×64	2.203143e-05	N/A	1.212332	151	6354
128×128	7.323467e-06	1.589	7.185103	296	24682
256×256	2.032626e-06	1.849	45.390103	549	95740
512×512	5.312876e-07	1.936	312.086512	1055	374458

From these results, we draw the following conclusions. Since the embedded boundary method uses a structured cartesian grid, it is relatively easy to implement while it is harder to write the mesh generation program. But after the mesh is given, the discretization is simpler for the mixed finite element method. Also FEM could have higher accuracy if high order basis function is used. To save computational resources when solving large problems, we could use EBM with automatic mesh refinement since it has better performance for

Table 2.5: Maximum gradient errors on the boundary by different methods

Size	EBM	RT0	BDM1	RT1
64×64	2.459720e-03	9.367834e-03	7.927028e-04	3.102073e-04
128×128	6.567893e-04	6.797467e-03	1.274594e-04	4.007023e-05
256×256	1.755489e-04	3.626596e-03	2.486447e-05	1.321007e-05
512×512	4.614643e-05	1.754357e-03	6.351849e-06	2.751578e-06

the same order of accuracies.

With the same mesh size, EBM has the fewest number of unknowns for the linear system and therefore, the fastest speed. From table 2.4, EBM and BDM1 have close solution errors while the time difference is huge. There are two reasons for this. One is that the EBM uses a structured grid and the finite volume/central finite difference has super convergence in the mesh. The MFM uses an unstructured grid, and to achieve the same order of accuracy, a higher order basis function space is needed, which means more unknowns. The other reason is that the unknowns for EBM are cell centered and those for the MFM are edge centered. Since the approximate ratio of the vertices to faces to edges is 1:2:3 for a simple large triangle mesh, we know the ratio of the unknowns for the EBM, RT0, BDM1, RT1 is approximately 1:3:6:6. Thus the EBM problem is smaller, which explains why it is much faster.

Finally, since interface reconstruction on the computational grid is needed for the EBM to do flux discretization, it is well consistent with the grid based Front Tracking method described in Section 2.1.2 . This makes the implementation of the algorithm more straightforward.

2.3.3 Timing Separation

Since the elliptic solver is only one part of the MHD algorithm, we would like to comment on the detailed computational cost. It is the sum of costs of three major components:

(i) interface propagation and reconstruction algorithms;

(ii) solving the hyperbolic system in the interior domains;

(iii) solving the linear system of equations corresponding to the elliptic problem.

Since the interface is a co-dimension one hypersurface, the computational cost of (i) is usually much smaller than the cost of other components. However this cost can be comparable with that of (ii) for very complex interfaces. The cost of (ii) is $O(N)$, where N is the total number of grid cells. For given N, the actual cost depends on the hyperbolic scheme, Riemann solver, and the equation of state. The FronTier code has a modular structure which allows the user to choose solvers and algorithms from a list of available options. The cost of iii) depends on the linear solver method the optimal choice of which may depend on the problem size. While solving relatively small linear systems corresponding to 2D domains may be done with $O(N^2)$ direct solvers, the optimal choice for a large 3D problem is a preconditioned iterative algorithm. The cost of the most optimal ones, such as multigrid, is $O(N \log N)$ [35]. We use parallel libraries of preconditioners and iterative solvers implemented in the PETSc [31] packages. For the above reason, the time cost for the elliptic step is minor on the coarse grid and becomes dominant at some sufficiently fine grid which is $136 \times 136 \times 136$ for the numerical example illustrated in

Figure 2.13. To reduce the total computational cost, we often perform one elliptic step per several hyperbolic time steps.

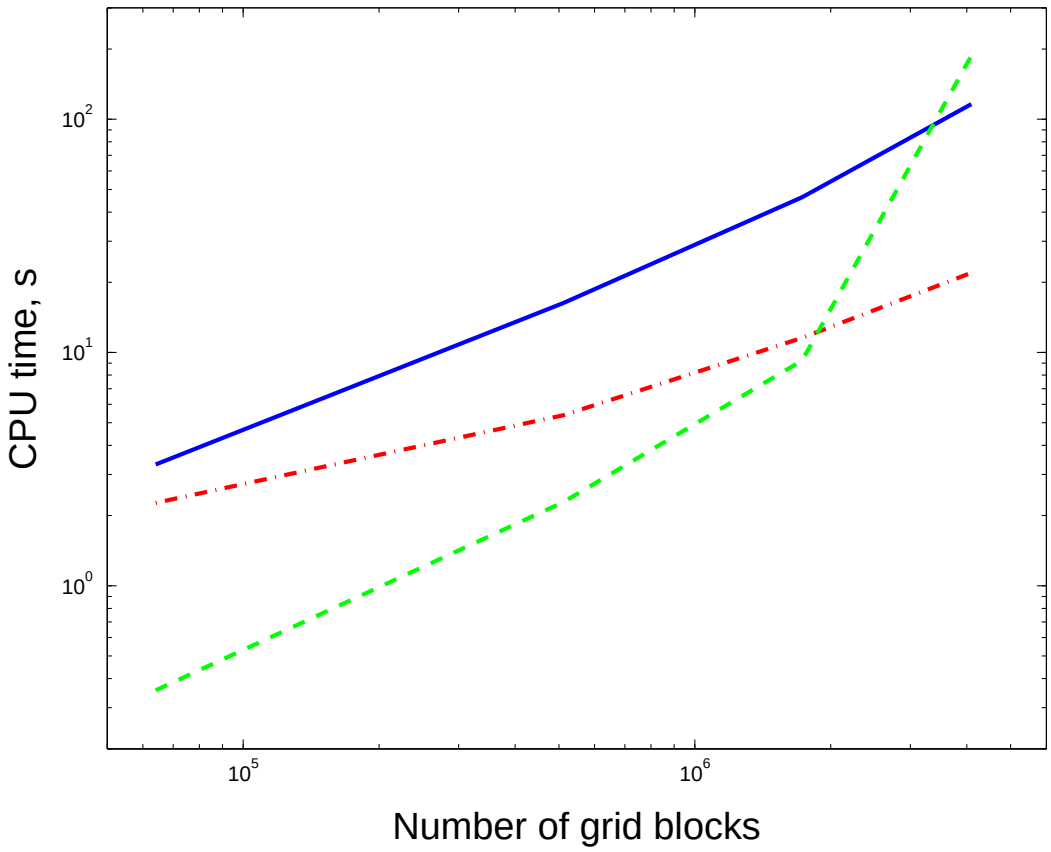


Figure 2.13: CPU time spent by the interface propagation algorithm (dot–dashed line), hyperbolic solver (solid line), and the elliptic solver (dashed line). The elliptic problem starts to dominate at the grid size $136 \times 136 \times 136$. Here the elliptic problem is solved for conducting fluid occupying 34 % of the computational domain. The hyperbolic solver is MUSCL with the exact Riemann solver and stiffened polytropic (conducting liquid) and polytropic (non-conducting gas) EOS models. The elliptic solver is GMRES with the block Jacobi preconditioning, as implemented in PETSc. The calculation was performed on a 2399 MHz Pentium cluster using $2 \times 2 \times 2$ domain decomposition.

Chapter 3

Simulation Results and Discussion

3.1 Jet Distortion in Non-uniform Magnetic Field

In this section, we validate our MHD code through the comparison of the numerical simulation with experiments and asymptotic solutions of a liquid mercury jet entering a non-uniform magnetic field. The simplicity of the problem from a physics point of view, and the presence of experimental data and satisfactory analytical solutions in terms of expansion series [36] are key factors in choosing this problem for the benchmark. Despite the fact that some features of this problem such as an incompressible and steady state flow regime are not well suitable for the simulation with a compressible time dependent code, we are able to obtain a good agreement with experiments and theory using a realistic equation of state for mercury. Our simulations of complex compressible free surface flows in liquids and weakly ionized plasmas in fusion and accelerator target applications often require an analysis of indirect and incomplete experimental data for validation, and therefore are less suitable for the benchmark.

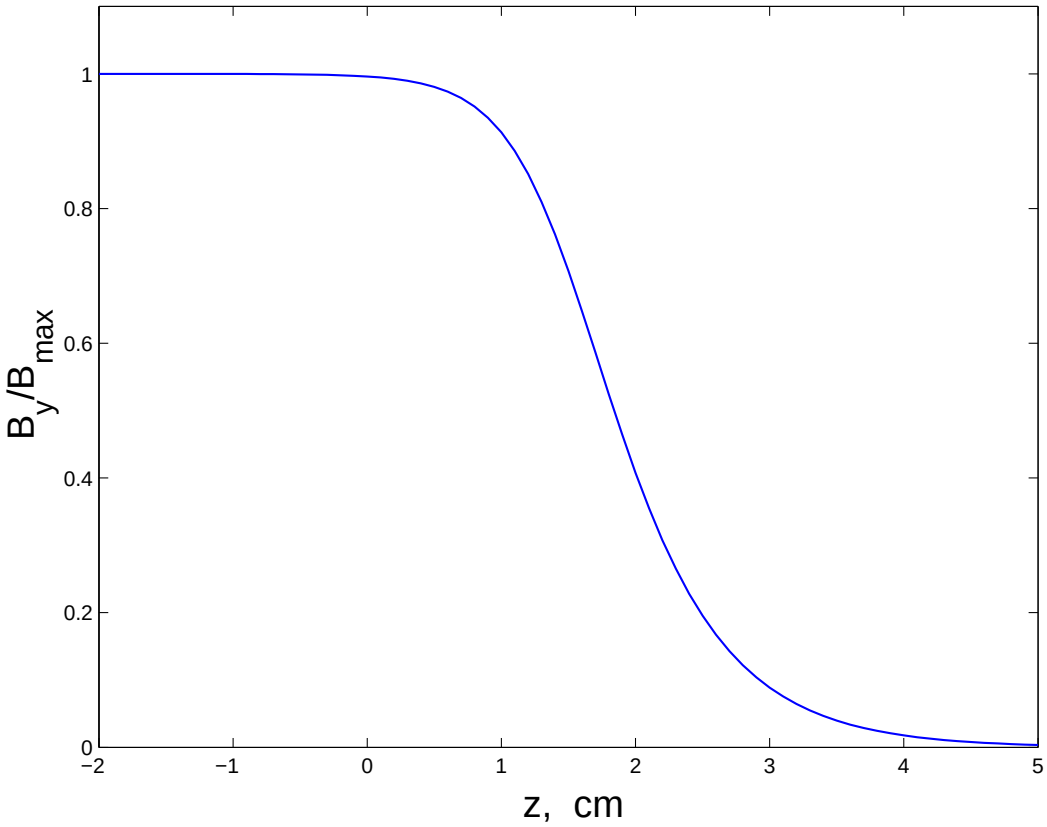


Figure 3.1: Distribution of the applied magnetic field

The setup of the problem is as following. A free mercury jet with diameter of 0.8 cm enters a stationary transverse magnetic field that has only one non-zero component, B_y, with the hyperbolic tangent spatial dependence

$$\left(\frac{B_y}{B_{max}}\right)^2 = \frac{1}{2}\left[1 - \tanh\left(\frac{z - z_0}{L_m}\right)\right],$$

where z_0 is the center and L_m is the characteristic length of the magnetic field. In our simulations, $z_0 = 1.5$ cm and $L_m = 0.62$ cm. The distribution of the magnetic field is shown in Figure 3.1.

In the theoretical analysis [36], the mercury jet going through the solenoid was approximated as a steady state jet of infinite length. The experiments also dealt with long jets. However conventional methods such as periodic or flow through boundary conditions are not applicable to the 3D direct numerical simulations of such very long or infinite jets that exceed the size of the computational domain. A simple analysis shows that the main contribution to the MHD force is provided by currents flowing in the longitudinal direction. It is difficult to accurately approximate the Neumann boundary condition on the edges of a jet slice for the Poisson problem. Therefore a relatively short jet with edges completely within the computational domain was simulated (see Figure 3.2 a). The jet length is chosen to be long enough compared to the jet diameter, but short enough to avoid an unnecessarily large computational domain. The jet velocity is set to zero in the laboratory frame, and the effect of the jet motion is achieved by moving the magnetic field in the computational domain. We have observed the flattening of the jet as it moves through the nonuniform magnetic field (see Figures 3.2 (b) and (c)), and compared the jet deformation with theoretical results of [36]. The theory itself has been experimentally validated.

Accordingly to theoretical calculation, when the effect of the surface tension is small, the relative increase of width of the jet at distance Z from the magnetic field center satisfies the formula:

$$\frac{dR}{R_0} = \frac{N_a}{8}\{Z + \frac{1}{\epsilon_m} log(e^{\epsilon_m Z} + e^{-\epsilon_m Z})\},$$

where R_0 is the initial radius of the jet before entering the magnetic field. $\epsilon_m = \frac{R_0}{L_m}, N_a = \frac{\sigma {B_y}^2 R_0}{\rho u}$ is the Stuart number,with ρ, σ and u represent the density, electrical conductivity and main flow velocity of the jet respectively. Therefore, with all other parameters as constants, the expansion is a function of $\frac{{B_y}^2}{u}$ and Z only. Simulation results of the relative width change of the jet cross-section in magnetic fields ranging from 5.5 Tesla to 12 Tesla and jet velocities of 50 m/s and 60 m/s are shown in Figure 3.3. The relative expansion of jet width is in a good agreement with theoretical predictions. As we see from Figure 3.4, the jet width linearly depends on B^2/u at a fixed distance (2cm) from the magnetic field center. An excellent agreement between simulation and theory is obtained. We have observed a quasi-steady state of the jet shape in the sense that jet deformations remain unchanged at some fixed longitudinal displacements with respect to the magnetic field at different times. The quasi-steady state lasts until the longitudinal position of the jet cross-section approaches the jet edge.

Past experiments have been performed [36] only in a narrow range of parameters (small absolute values and ranges of the velocity and magnetic field). Here we present our studies of the jet deformation at large changes of the velocity and magnetic field, and compare them with the theory. Studies using a wide range of parameters are also important for practical applications, as future experiments with mercury jets at the Neutrino Factory / Muon Collider facility will operate with parameters significantly different than that in Oshima's experiments. As it was shown in [36], the theory is a good agreement with experiments on the width of the flattened jet.

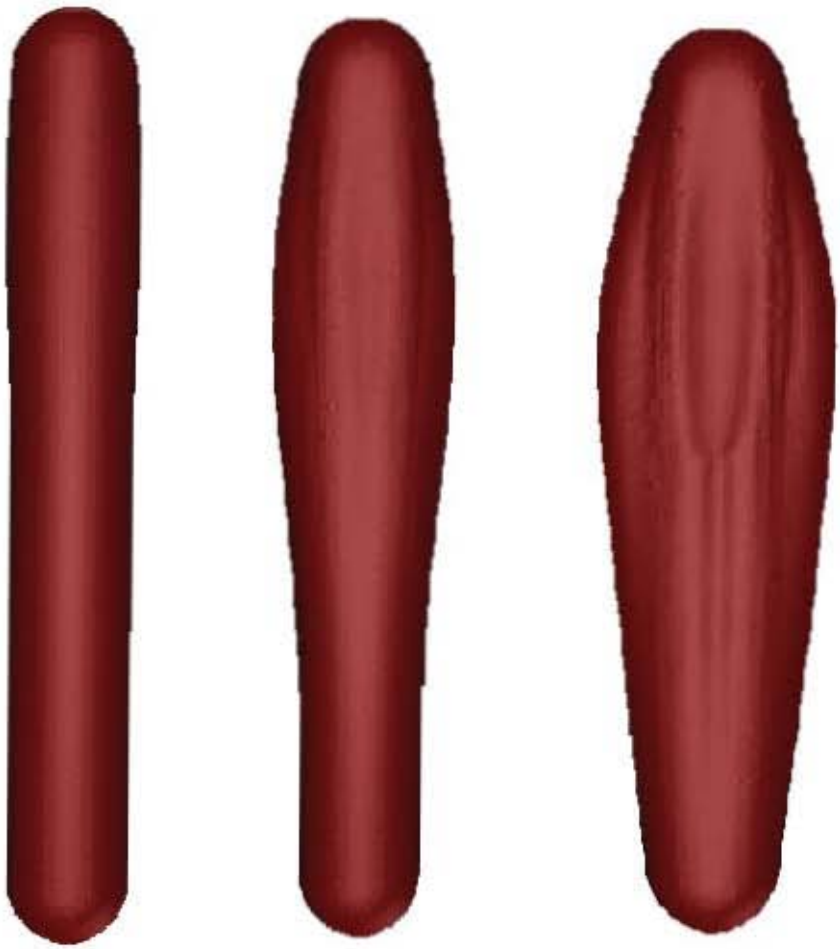


Figure 3.2: Change of the mercury jet shape as it enters the magnetic field. (a) Initial time, (b) t = 1.0 ms, (c) t = 1.5 ms

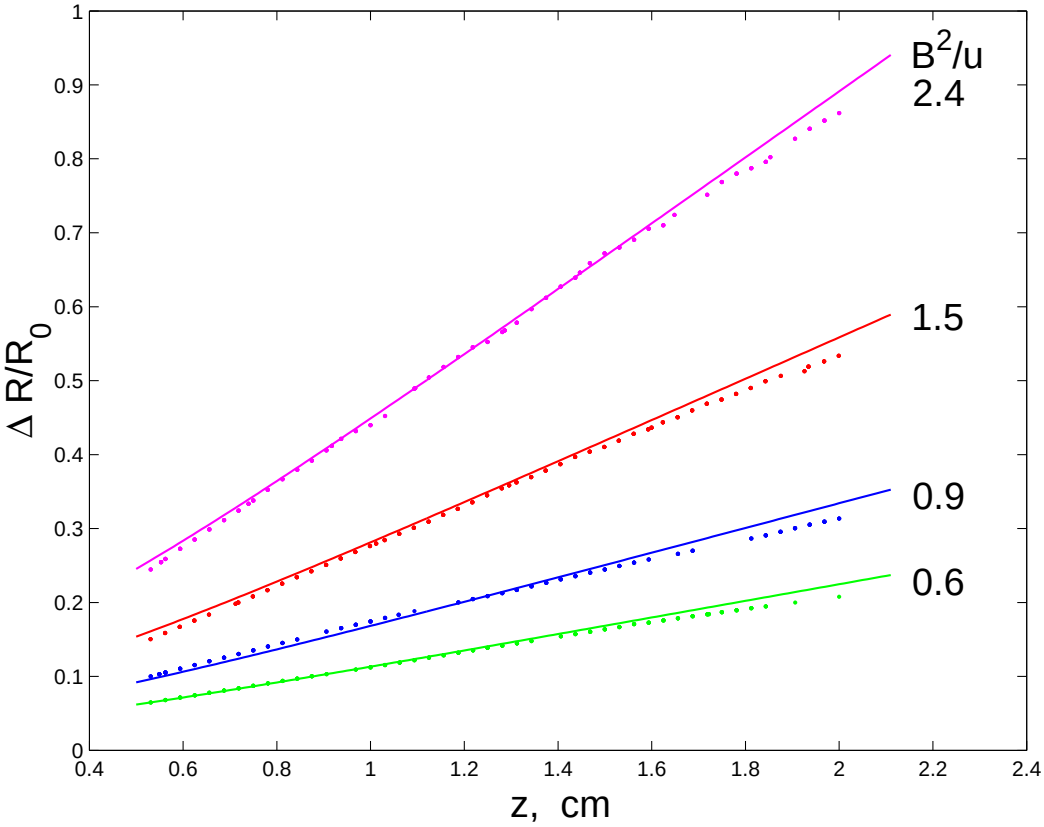


Figure 3.3: Dependence of the relative change of the width of the jet cross-section on the longitudinal coordinate with respect to the magnetic field center at different values of B^2/u, [Tesla^2·s/m]. Solid line is the theorerical result and dots are measurements of simulated jet shapes.

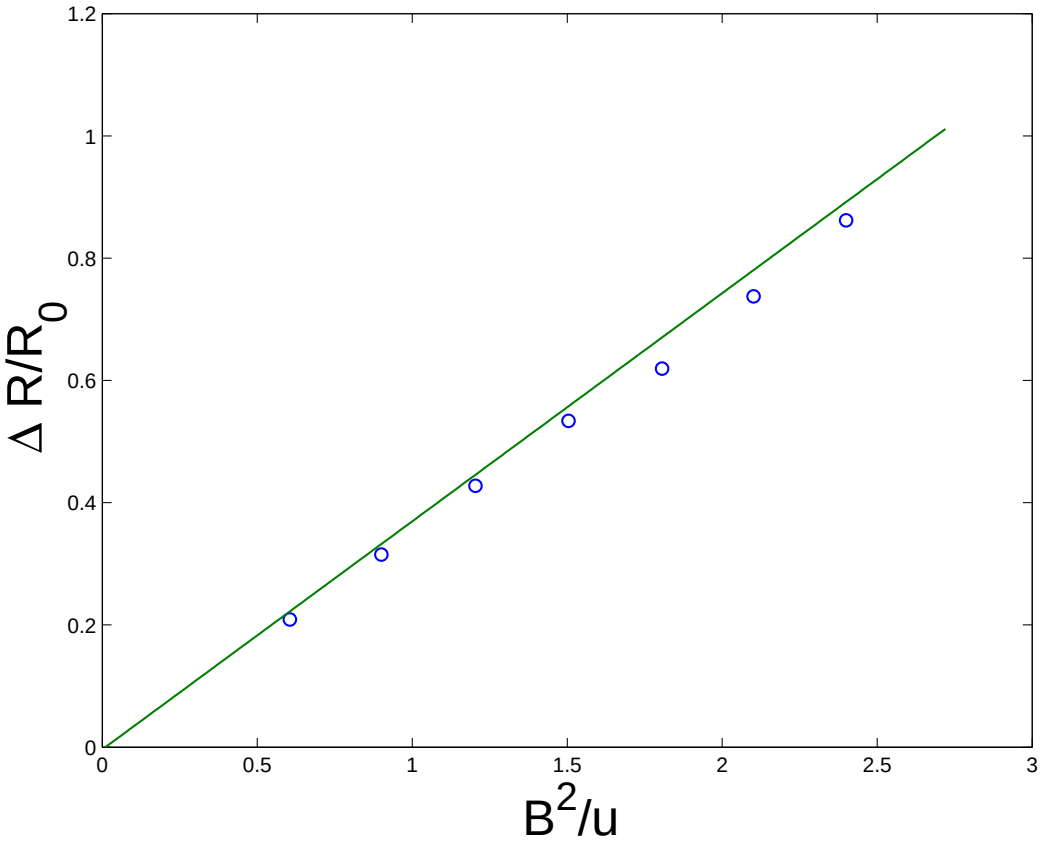


Figure 3.4: Dependence of the relative change of the width of the jet cross-section on B^2/u, [Tesla^2·s/m], at the fixed distance ($z = 2$ cm) from the magnetic field center. Solid line is the theorerical result and dots are measurements of simulated jet shapes.

Mesh Size	Z=1cm		Z=2cm	
	dR	Relative Error	dR	Relative Error
$24 \times 15 \times 78$	0.1227	3.23e-1	0.2163	3.96e-1
$48 \times 30 \times 156$	0.1770	2.37e-2	0.3195	1.08e-1
$72 \times 45 \times 234$	0.1806	3.86e-3	0.3524	1.65e-2

Table 3.1: Convergence of simulations to theoretical results at B=12 Tesla and u=60 m/s.

We have also performed the convergence study and find that the simulation error reduces with the mesh refinement (see Table 3.1). We would like to note that since theoretical calculations are used to quantify the error, some small error will remain at even higher level of the grid refinement due to the unknown error of the theory. Experimental results are also not perfect as fluctuations of the jet velocity, pressure in the nozzle etc. impose significant errors, and the solution can be approximated only in the sense of statistical average. Such detailed information on Oshima's experiments is not available to us, but our experience with the analysis of the targetry experimental data from CERN tells us that the statistical average errorbars could be as high as 20%. Therefore we have chosen to compare simulations with the theory, and clearly observed the agreement and convergence with the mesh refinement.

3.2 Muon Collider Targetry simulation

In this section, we present results of the numerical simulation of a liquid mercury jet interacting with an intensive proton pulse in a 15 Tesla magnetic field. Such a jet will be used as a target in the proposed Muon Collider/Neutrino Factory. The target is shown schematically in Figure 1.4. It

will contain a series of mercury jet pulses of about 0.5 cm in radius and 60 cm in length. Each pulse will be shot at a velocity of 25 m/sec into a 15 Tesla magnetic field at a small angle (0.1 rad) to the axis of the magnetic field. When the jet reaches the center of the magnet, it is hit with a 3 ns proton pulse depositing about 80 J/g of energy in the mercury.

3.2.1 Problem Setup

In all of our simulations, we restrict the jet length to 10 cm to avoid unnecessary computation cost. For the 2D simulations, the domain is chosen as the cross sections along either the radial or the longitudinal direction of the jet. Cylindrical symmetry is used for the longitudinal case and only half of the cylinder is simulated. In 3D simulations, jet surface is initialized as cylinder with the radius of 0.5 cm and the length of 10 cm. Figure 3.5 shows the configurations.

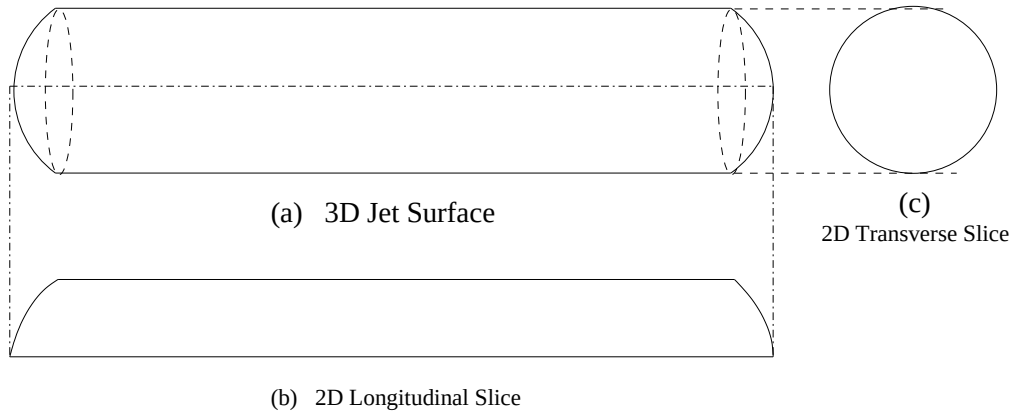


Figure 3.5: 2D and 3D interface profile

The actual energy deposition in the mercury jet due to the interaction with protons was calculated using a Monte-Carlo code MARS [37]. The energy deposition profile can be accurately approximated by a two-dimensional Gaussian function in cylindrical coordinates. Since the angle between the jet axis and the magnetic field lines is small, the off-axial component of the magnetic field can be ignored for the study of short time scale hydrodynamic processes caused by the proton energy deposition (the off-axial magnetic field could not be ignored in the study of some other aspects of this problem, for example the relatively long time scale entrance of the jet into the magnetic field). In the assumption of a uniform magnetic field, only the radial motion of the mercury induces eddy currents.

First, the energy deposition ΔE causes the increase of pressure ΔP in the liquid. SESAME table [38] can be used to look up the data of pressure increase due to the energy deposition specific for mercury. For stiffened polytropic equation of state, γ_l can be calculated from equation:

$$\Delta P = (\gamma_l - 1)\rho\Delta E \tag{3.1}$$

Then P_∞ in the EOS equation 1.12 is obtained by combining above parameters with the sound speed in mercury (145 m/s). This makes stiffened polytropic EOS parameters consistent with the data from SESAME table. As for the homogeneous EOS model mentioned before, suppose the initial density and pressure (before the energy deposition) are (ρ_0, P_0). We calculate what density increase $\Delta\rho$ should be made in order to increase the pressure to $P_1 = P_0 + \Delta P$

by connecting two states (ρ_0, P_0) and $(\rho_0 + \Delta\rho, P_1)$ through adiabatic curve.

The typical initial pressure profile along the radial direction of the jet is shown as Figure 3.6. The pressure value can reach up to 16 kbar in mercury. Previous numerical simulations performed with a single phase equation of state have shown that the strength of rarefaction waves occurring after the energy deposition significantly exceed the mercury cavitation threshold. We believe that the formation of cavities takes place and cavitation bubbles influence the wave dynamics in mercury and the jet surface evolution. Both the homogeneous and heterogeneous models mentioned in previous chapter are used to model the multiphase flow.

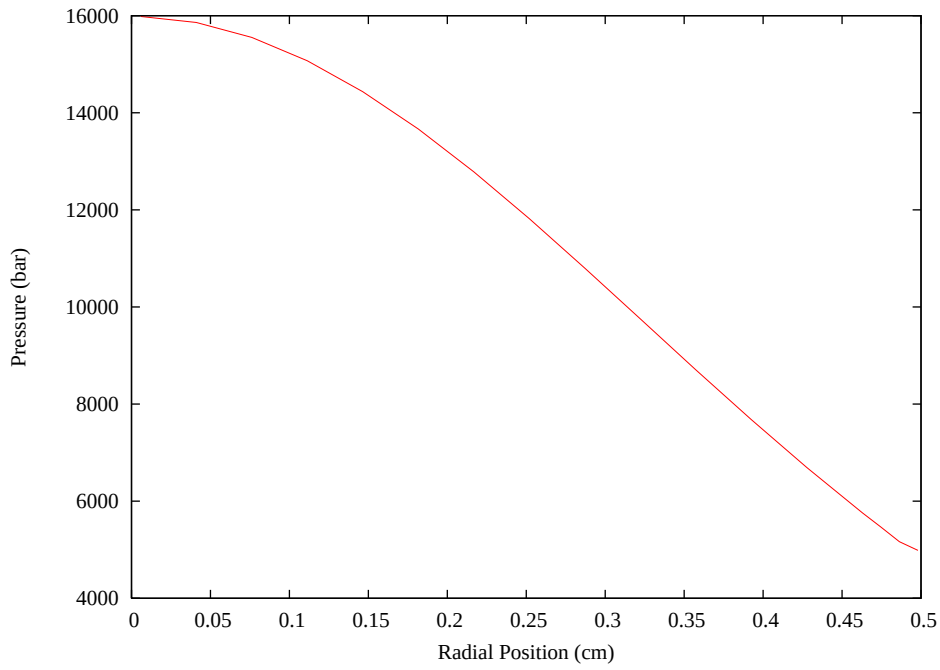


Figure 3.6: Initial pressure profile after energy deposition

3.2.2 Results and Comparison

First we consider the results from the homogeneous model. For the simulations without MHD effects, Figure 3.7 - 3.9 plot the density profile for the 2D radial cross section. The results are obtained with grid 256×256 over domain [-1.5, 1.5]×[-1.5,1.5]. Two phase domain with density much less than pure liquid mercury forms at late time. Figure 3.10 shows the jet radius vs. time from 2D simulations with different grid resolutions. (For the 2D longitudinal case, the radius is always measured at the middle point of the longitudinal axis). It is clearly shown that simulations with different grid size give almost identical jet expansion speed. The first picture in Figure 3.13 is the density plot at $t = 0.15ms$ in the 2D longitudinal cross section. The jet expansion velocity calculated from Figure 3.10 is about 30 m/s and is in good agreement with experimentally measured values [39]. Also as shown in Figure 3.11, 2D and 3D simulations give very close results.

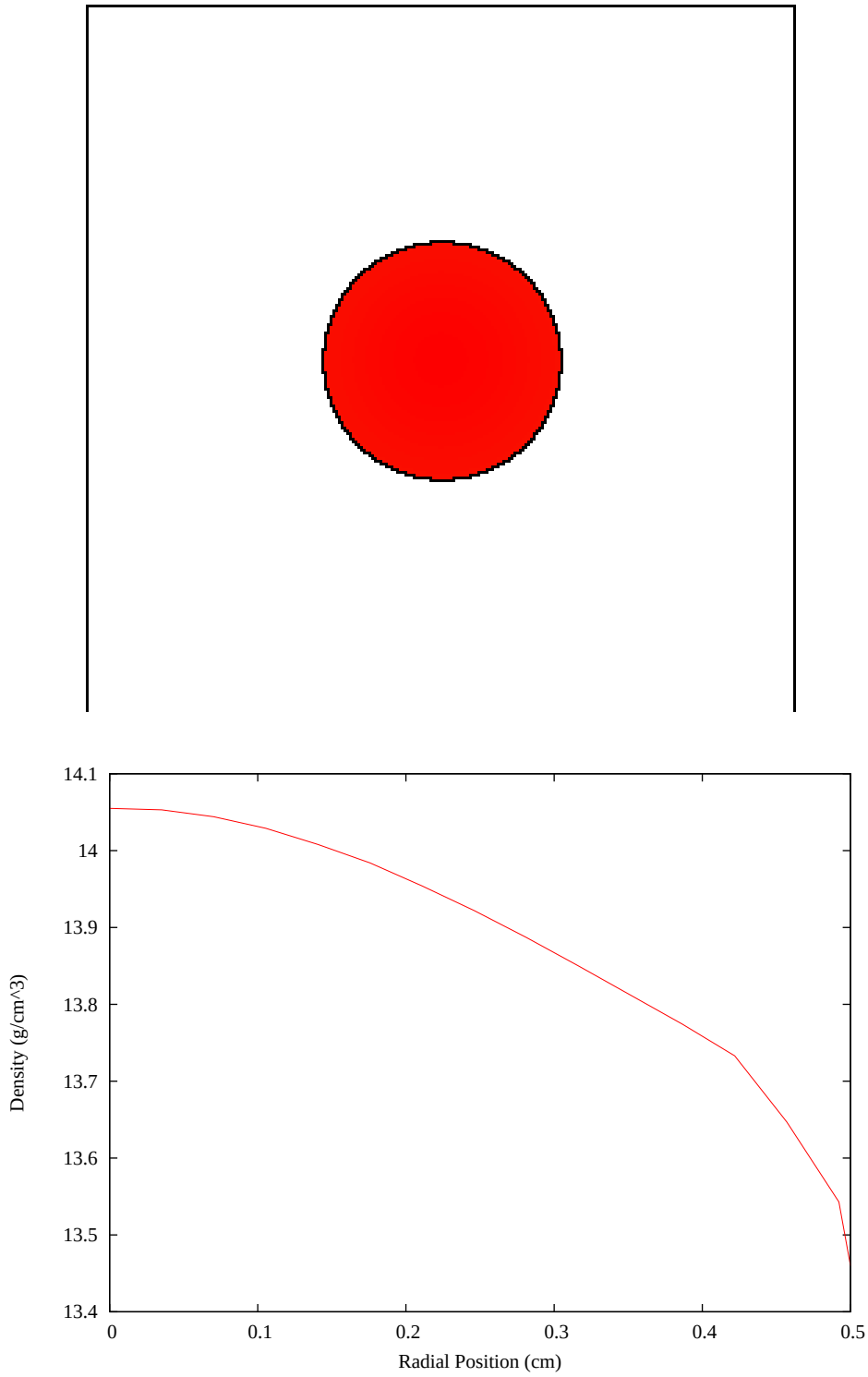


Figure 3.7: Density profile and radial distribution for t = 0

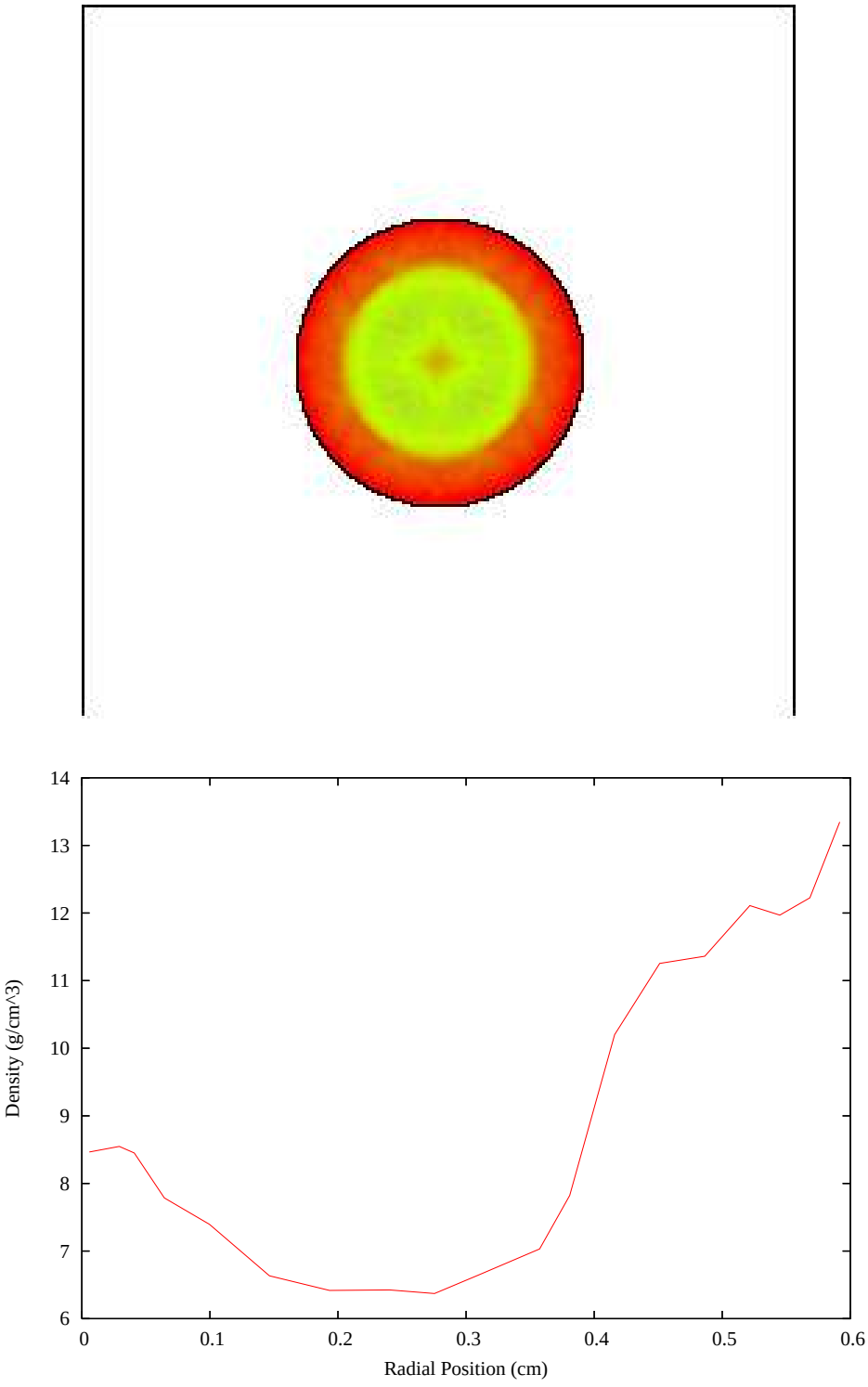


Figure 3.8: Density profile and radial distribution for t = 0.025 ms

Radial Position (cm)

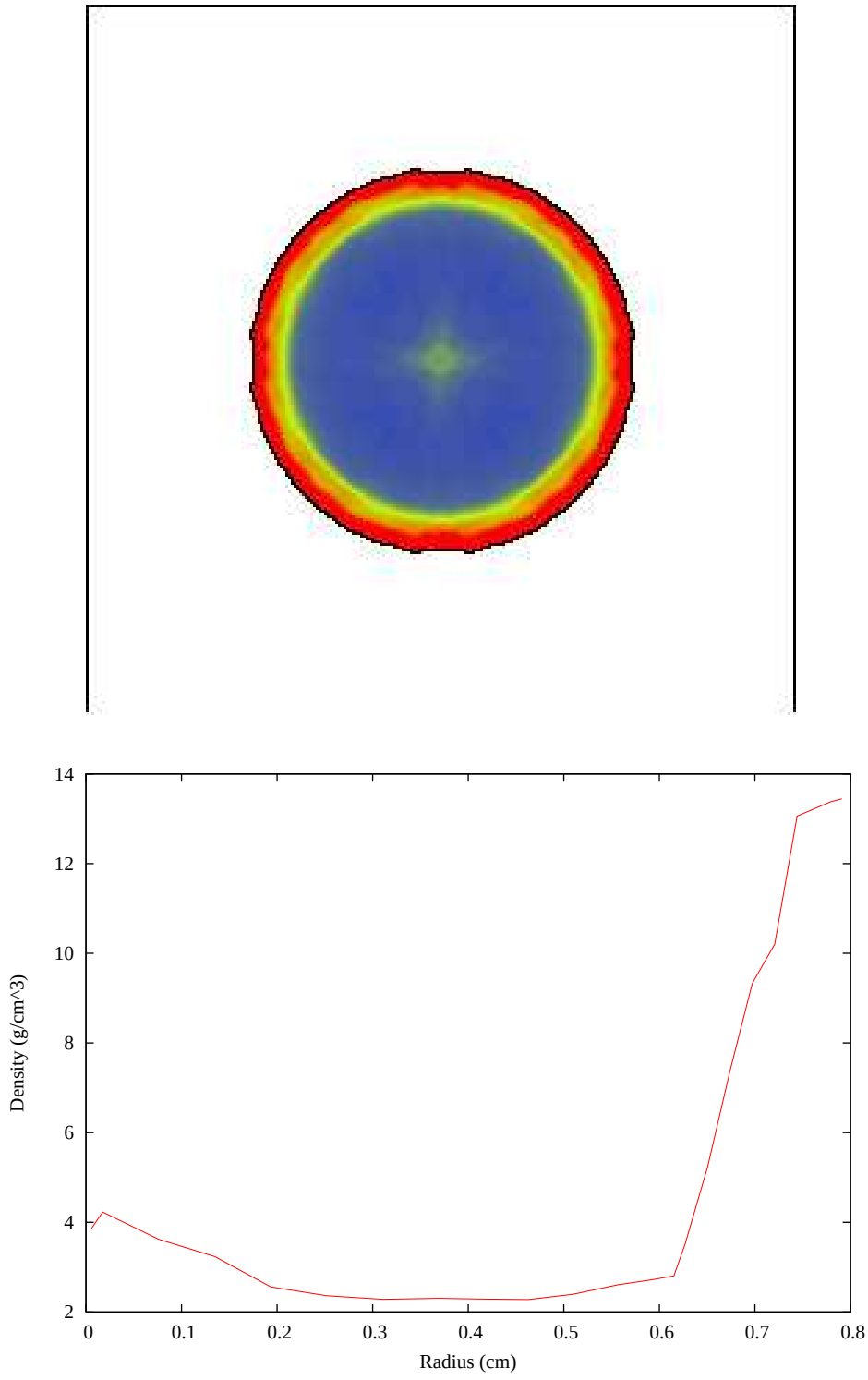


Figure 3.9: Density profile and radial distribution for t = 0.075 ms

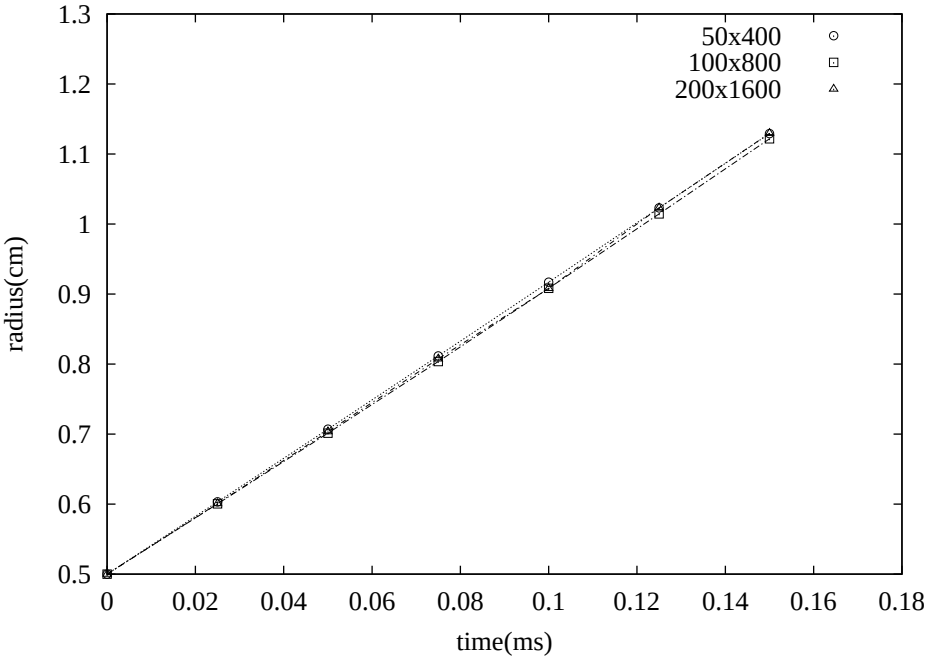


Figure 3.10: Comparison of 2D simulations without MHD effects

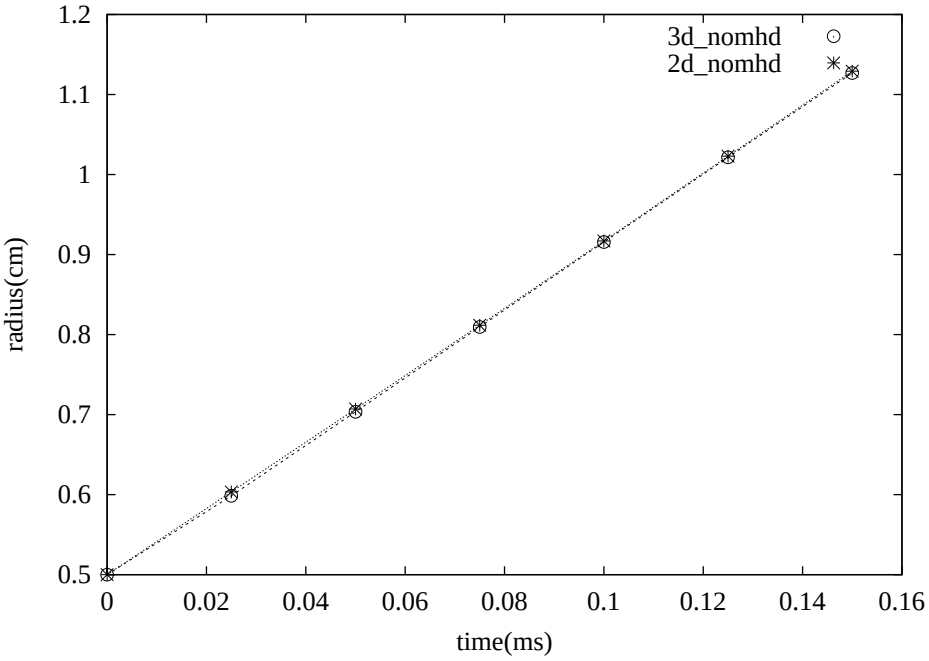


Figure 3.11: Comparison of 2D and 3D simulations without MHD effects

After adding the magnetic field and with MHD effect included, both the Bruggeman and linear conductivity model are used to calculate the electronic current density. By comparing Figure 3.11 with Figure 3.12, we find that 15 Tesla magnetic field can greatly restrict the jet expansion. In the density plot of 2D longitudinal cross section as in Figure 3.13, the growth of the cavitation region is also much slower in the simulations with MHD effects. On the other hand, there is no big difference regarding expansion speed between the two conductivity models, while the running with Bruggeman model has slightly higher expansion velocity. The MHD restriction effect on cavitation region is also stronger for the linearized conductivity model, as shown in the cross section density profile plot in Figure 3.14. This is not difficult to understand since the Bruggeman model has a percolation threshold and thus smaller MHD effects when the volume fraction of vapor is the same. 2D and 3D simulations with homogeneous model give similar jet expansion for the MHD runnings as shown in Figure 3.12. In Figure 3.15, the time evolution of jet radius and cross section density profile under different magnetic field are plotted. It is shown that the jet expansion is almost complete stopped after some time when magnetic field reaches 15 Tesla.

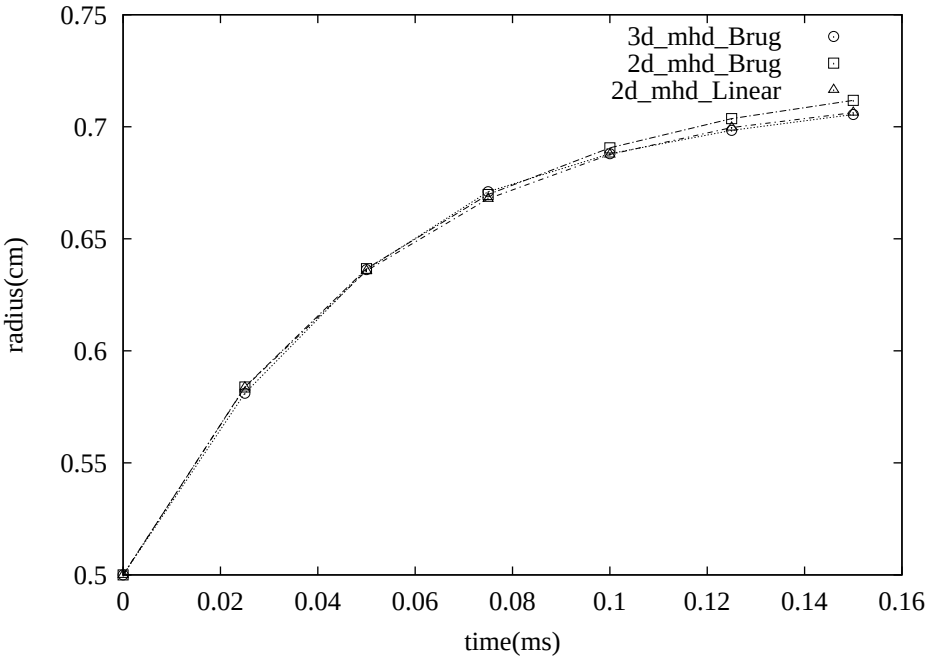


Figure 3.12: Comparison of 2D and 3D simulations with MHD effects

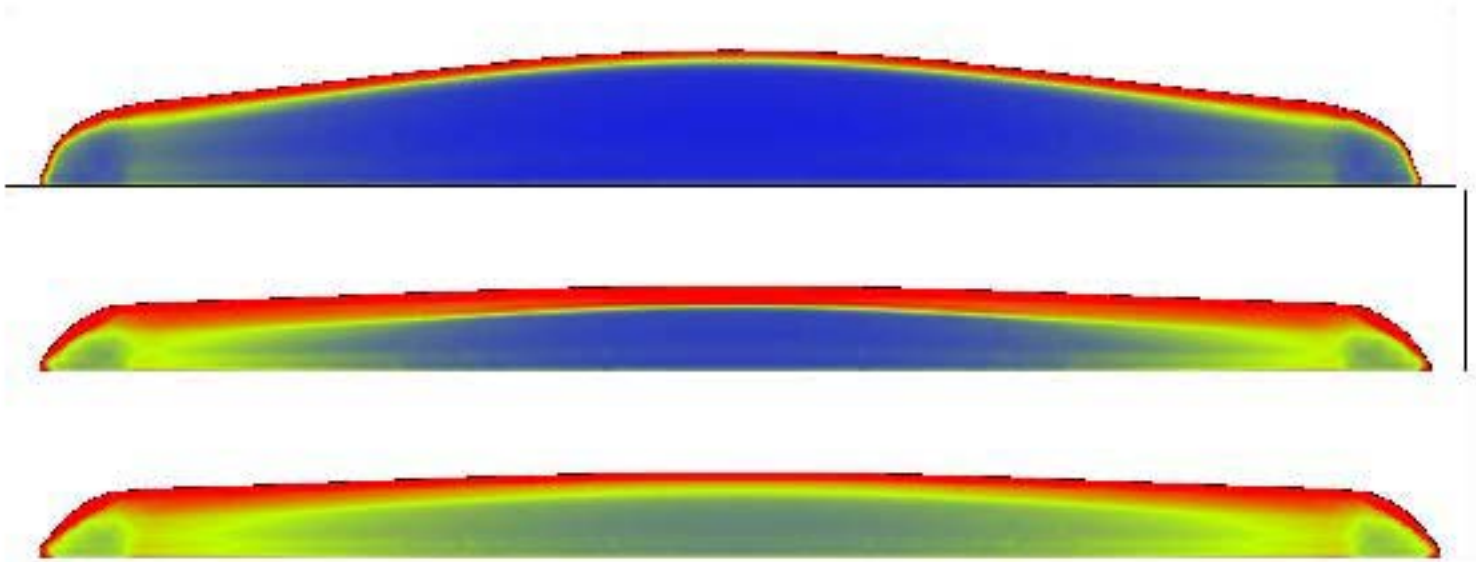


Figure 3.13: Density Profile for 2D simulations of no MHD, MHD with Bruggeman model, and MHD with linear model,from top to bottom at t = 0.15ms

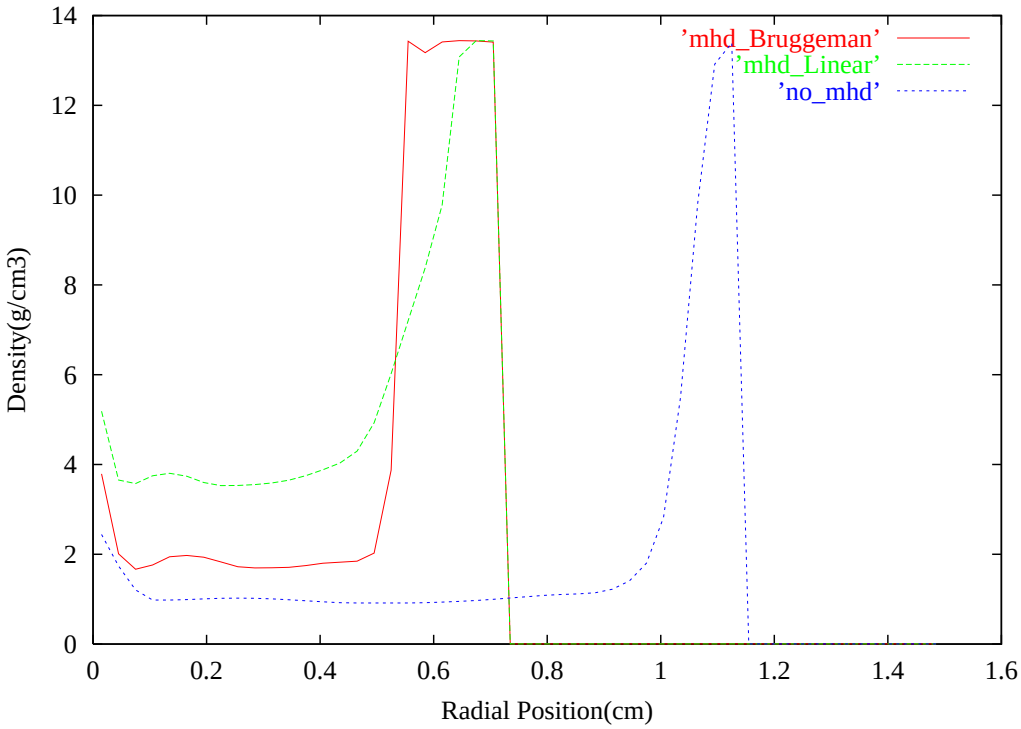


Figure 3.14: Radial density distribution at jet center of no MHD, MHD with Bruggeman model, and MHD with linear model

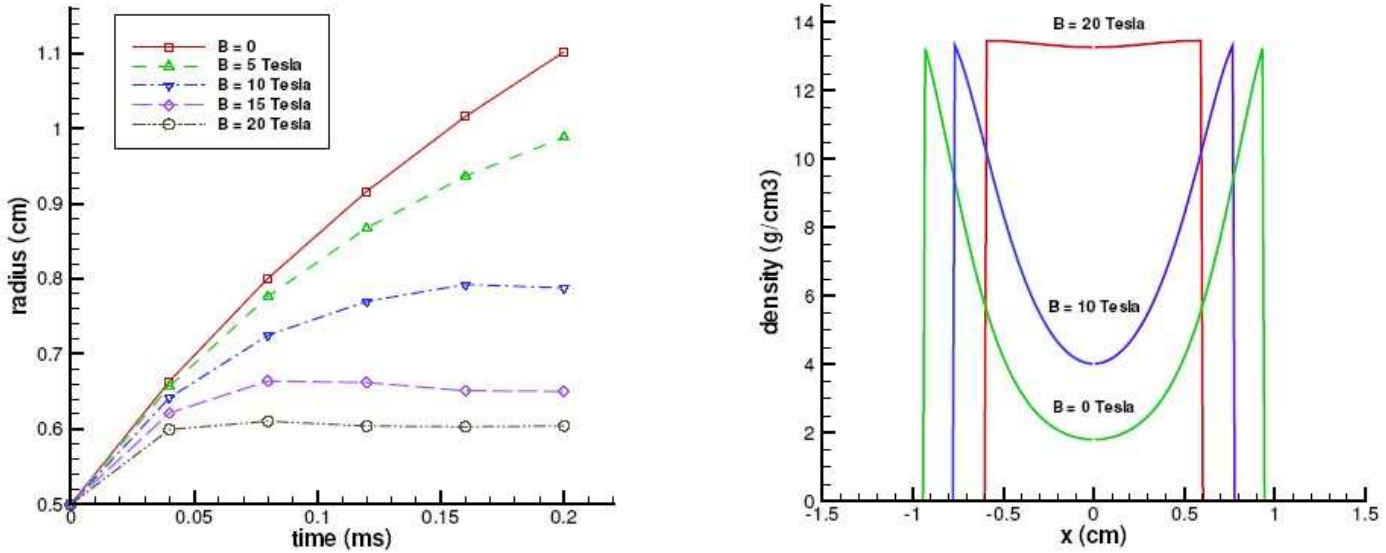


Figure 3.15: Jet Expansion(left) and Radial Density Distribution(right) Under Different Magnetic Field

Simulation	$P_{critical}$(Bar)	Bubble Distance(mesh blocks)
$2D_{b2}$	-100	2
$2D_{b3}$	-100	3
$2D_{b4}$	-100	4

Table 3.2: Different Sets of Simulations with Critical Pressure and Bubble Distance

Besides the homogenous equation of state, we also use heterogenous EOS in which vapor and liquid phase use polytropic and stiffened polytropic EOS respectively. Vapor bubbles are dynamically inserted into the rarefied region in liquid where the pressure is less than some critical value. We use different critical pressure values and also different bubble distances for the simulations to see if they have any impacts on the jet expansion. All the simulations use grid size of 200 $\times$ 1600 over domain [0,1.5] $\times$ [0,12] to resolve the vapor bubbles. The parameters are shown in Table 3.2. From the time evolution of jet radius plotted in Figure 3.18, we find that the related parameters for bubble insertion do not have big influences on the speed of jet expansion, though the final interface profiles are quite different. The homogeneous EOS model gives larger jet expansion than the heterogeneous model. From Figure 3.17 and Figure 3.16, as in homogeneous case, the expansion of the jet as well as the growth of each individual bubble are greatly impeded by the magnetic field. Figure 3.19 plots the electronic current density distribution for the heterogeneous simulations and shows that everywhere the current is tangential to the interface.

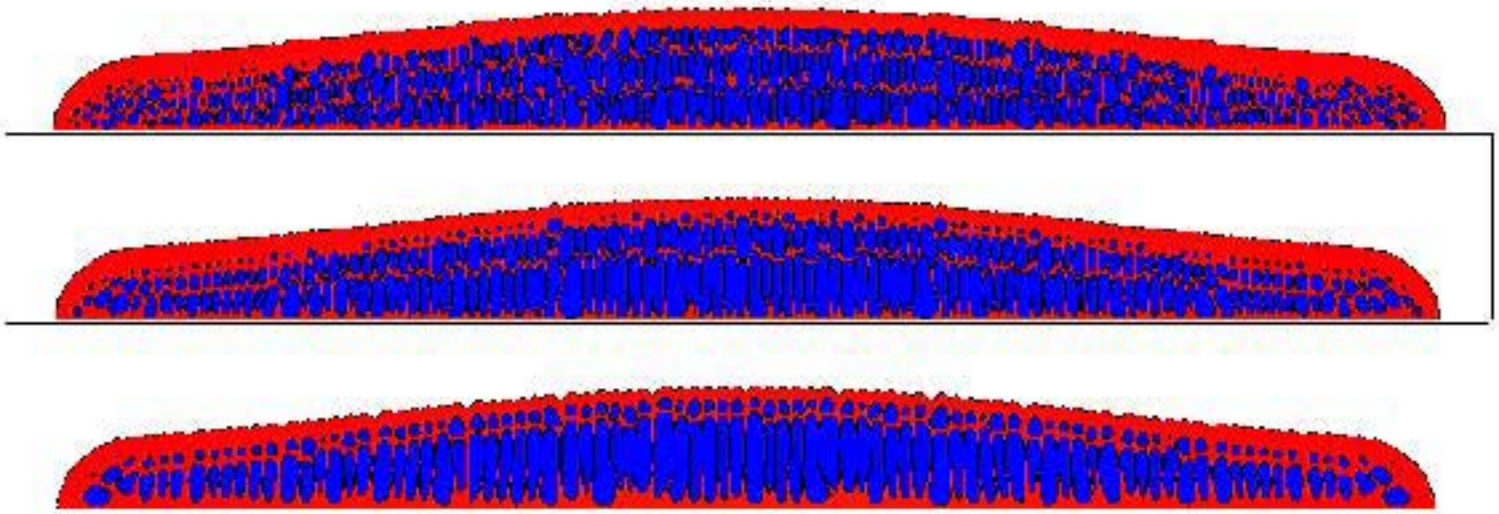


Figure 3.16: Density Profile for 2D simulations of no MHD $2D_{d2}$,$2D_{d3}$,$2D_{d4}$,from top to bottom at t = 0.15ms

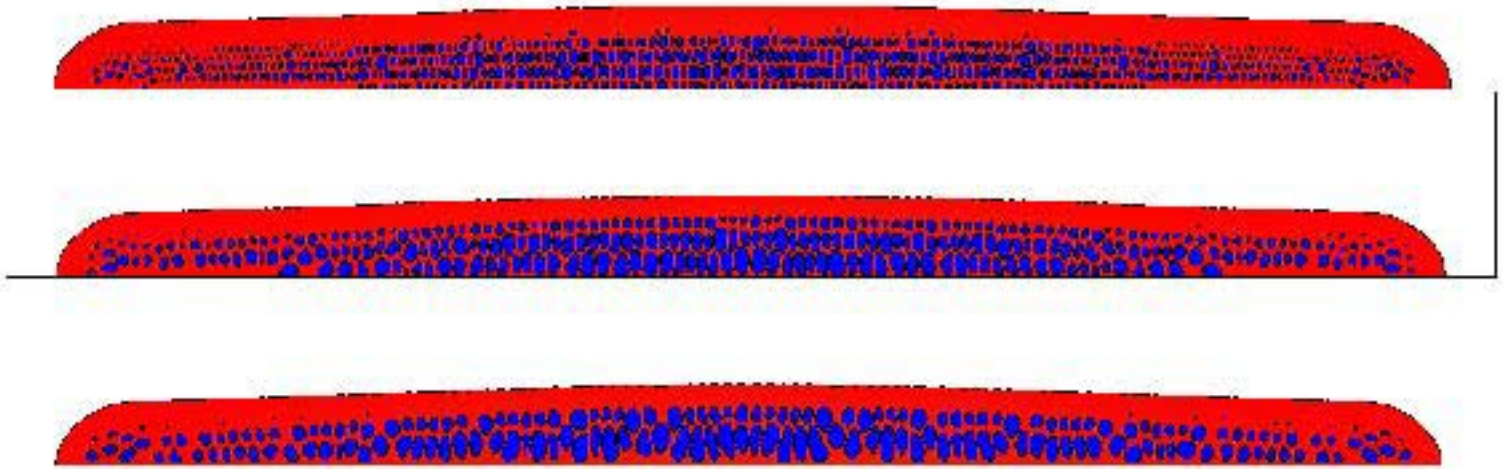


Figure 3.17: Density Profile for 2D simulations of MHD $2D_{d2}$,$2D_{d3}$,$2D_{d4}$,from top to bottom at t = 0.15ms

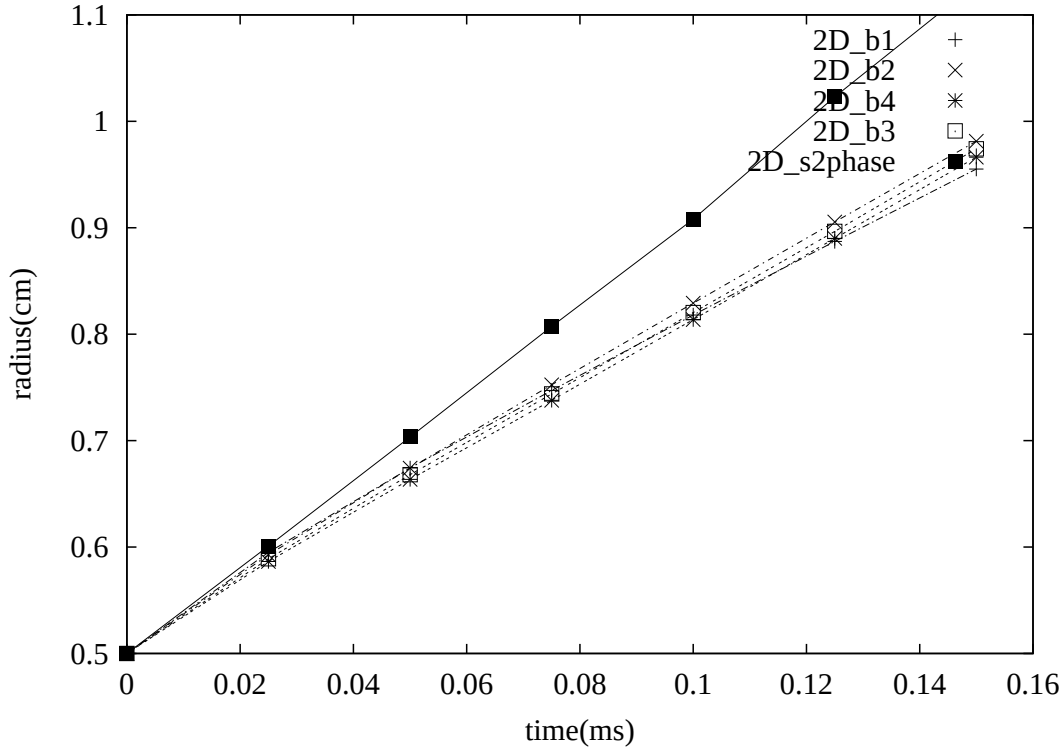


Figure 3.18: Comparison of jet expansion of homogenous and heterogenous eos simulations without MHD effects

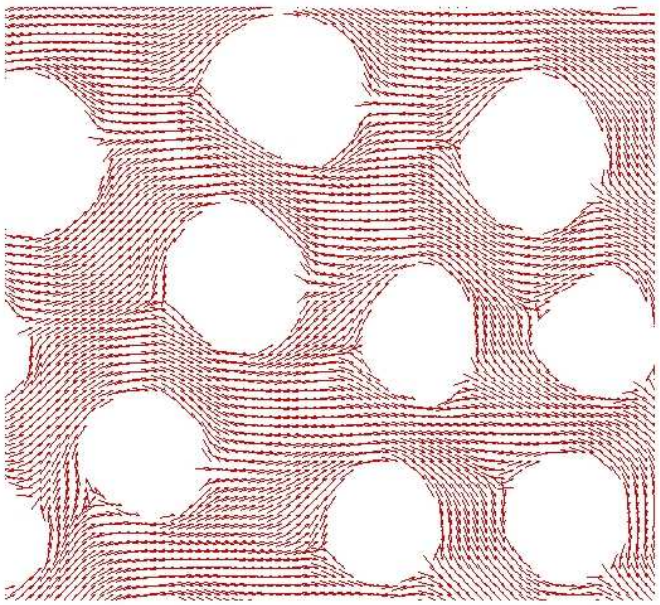


Figure 3.19: 2D electronic current density distribution

3.2.3 Effects of Bubble Collapse

For all above simulations, no obvious surface instabilities are observed, even for the running without MHD effect. This is not consistent with the experimental results, as shown in Figure 3.20. One of the possible causes is that without magnetic field, the mercury jet has already become turbulent before interacting with proton pulse (left side, Figure 3.20) while the jet surface is smooth in all of our simulations. Another reason, we believe, is due to the unresolved process of cavitation bubble evolution. In both the homogenous and heterogenous simulations, the collapsing and rebounding of the cavitation bubbles are not included, which could be of great importance for the jet breakup.

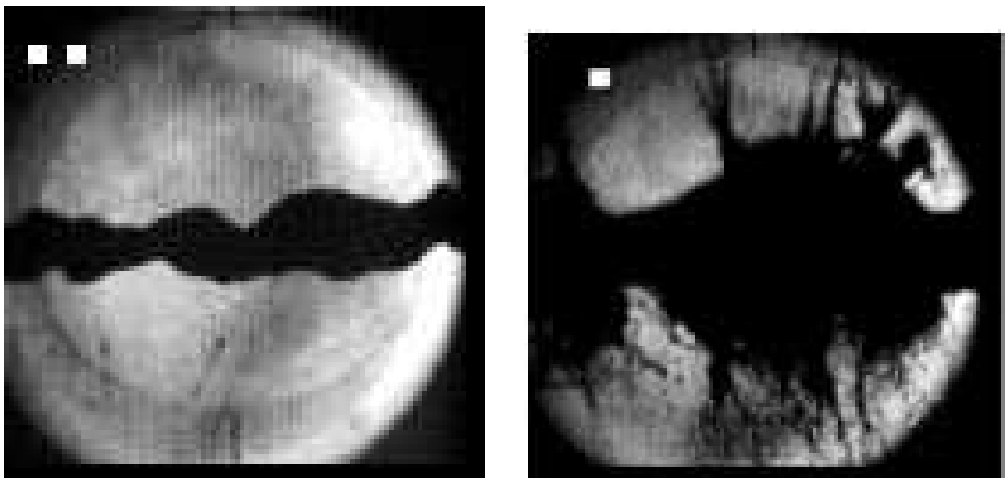


Figure 3.20: Initial and late time surface profile for the mercury target

It is known that the bubble collapse in compressible liquid is governed by the Keller's equation:

$$(1-\frac{1}{c_f}\frac{dR}{dt})R\frac{d^2R}{dt^2}+\frac{3}{2}(1-\frac{1}{3c_f}\frac{dR}{dt})(\frac{dR}{dt})^2=\frac{1}{\rho_f}(1+\frac{1}{c_f}\frac{dR}{dt}+\frac{R}{c_f}\frac{d}{dt})(p_B-p), \quad (3.2)$$

where R is the bubble radius, c_f is the sound speed in liquid, p_B and p are pressure within bubble vapor and ambient liquid respectively. By solving the equation numerically with typical initial data found in mercury target, it can be shown that the cavitation bubbles tend to shrink to less than 10^{-3} of the original sizes. Therefore, it is too expensive to do 2D/3D global simulations to resolve the whole procedure. Also the time step has to be reduced to almost zero for such simulations due to the CFL condition. Therefore, we propose a multiscale approach which contains only portion of the computational domain. The main idea is as following.

First we separate a small region from 2D or 3D global domain which contains a cavitation bubble (top and lower left of Figure 3.21). Our goal is to study locally if single bubble collapse can be the cause of any surface perturbations for the mercury target. In order to make the simulation closer to real situation and enable the bubble to go to small size with less limitations from grid resolution, we initialize the 2D and 3D simulations with late time data (rebounding stage) from corresponding 1D simulation, which has relative large bubble radius. As for 1D case, the domain is initialized as a two phase region separated by an interface (bubble wall). The liquid phase uses a stiffened polytropic EOS and vapor phase uses a polytropic EOS. By setting the relatively

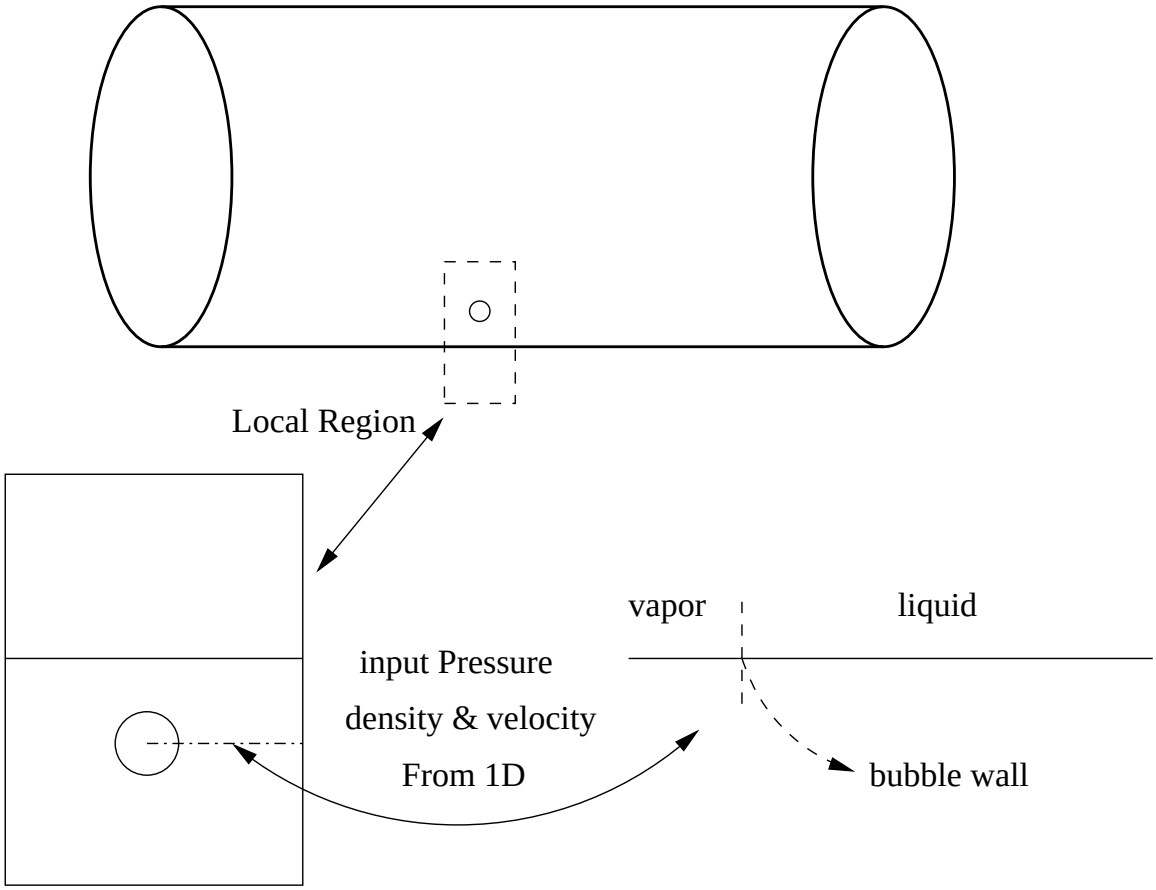


Figure 3.21: Multiscale approach to simulate the bubble collapse

large initial vapor bubble radius (0.01cm), low vapor pressure (0.01bar) and large ambient liquid pressure (100bar), the vapor bubble begins to collapse. When minimum radius is achieved and large pressure accumulated inside, a rebounding stage ensues and shock front develops. As the bubble radius increases, the pressure front propagates outwards from the collapse center. At the late stage of the rebounding when the bubble radius is large enough, the pressure, density and velocity profiles are taken as the initial input condition of a similar 2D/3D simulation.

By solving the Keller's equation numerically with initial conditions mentioned above, we got the time evolution radius profile of the collapsing bubble and compared it with out simulation results. As in Figure 3.22, the simulation and theory matches reasonably well, considering Keller's equation is dealing with bubble collapsing with infinite domain of constant far field pressure while our 1D simulation only use finite domain where pressure changes quickly with bubble collapsing. As from Figure 3.23, the pressure peak in the vicinity of the bubble can reach as high as 8000 bar.

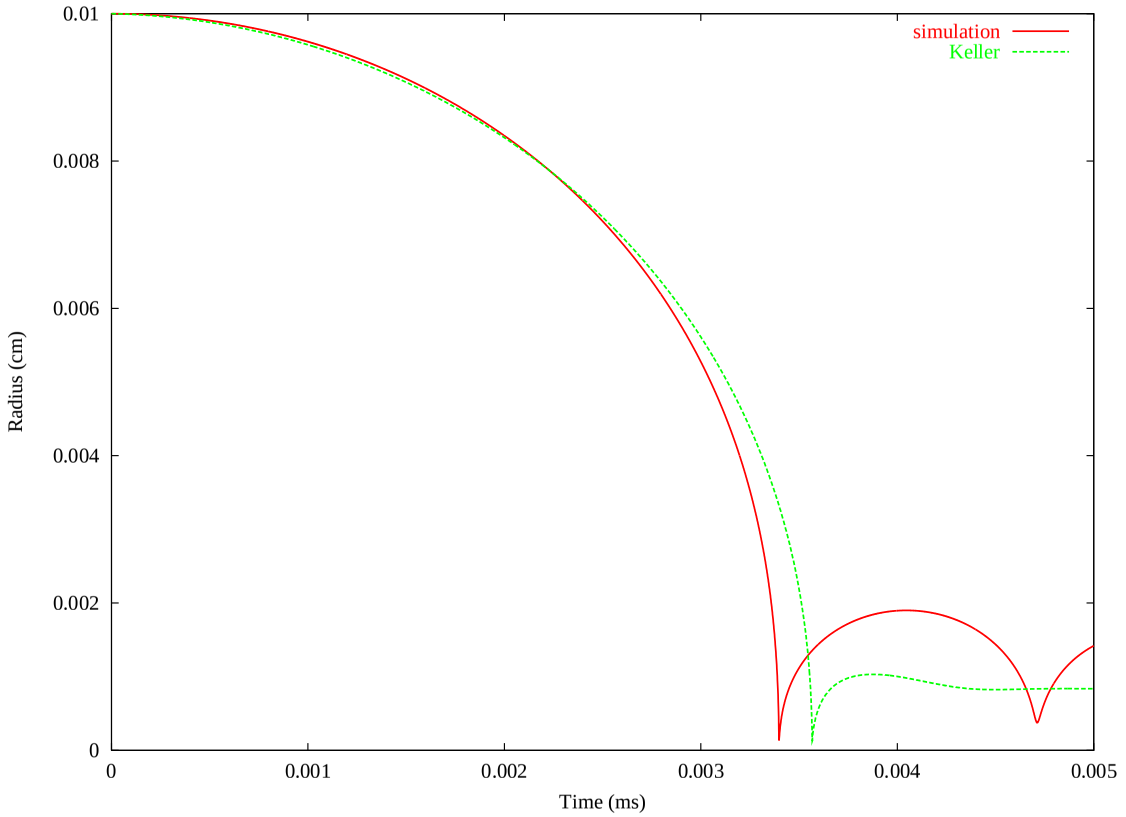


Figure 3.22: Radius vs. Time profile for bubble evolution

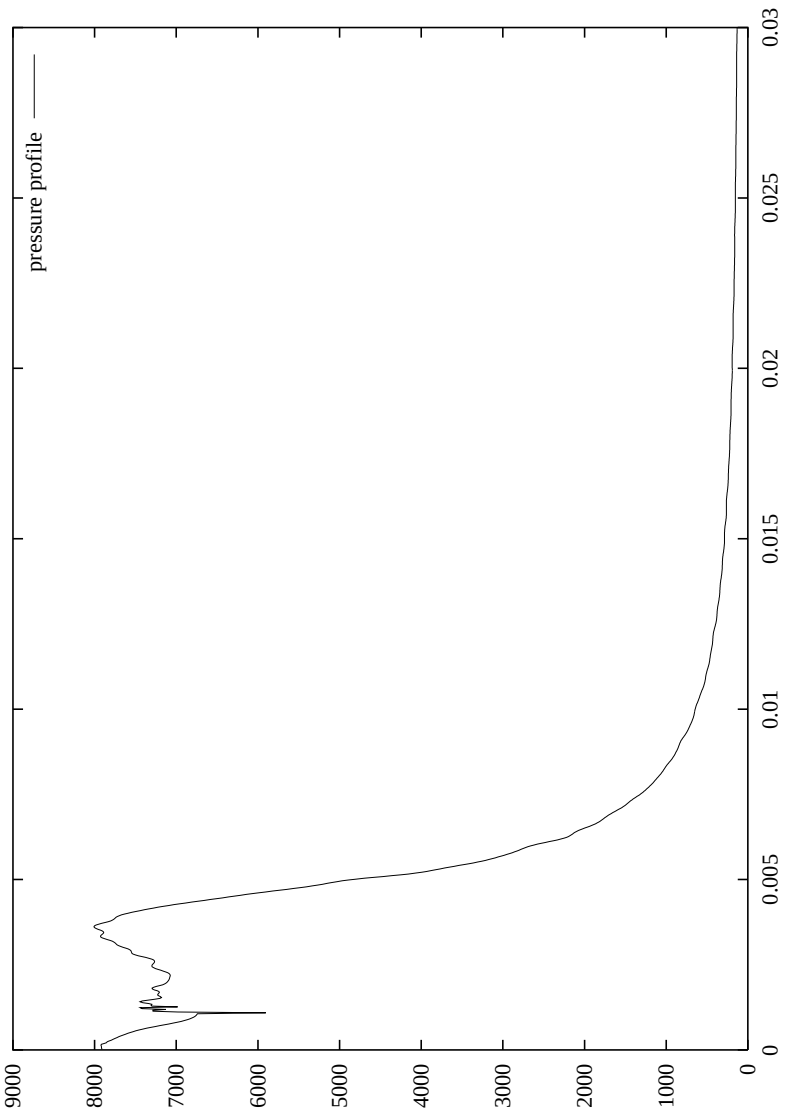
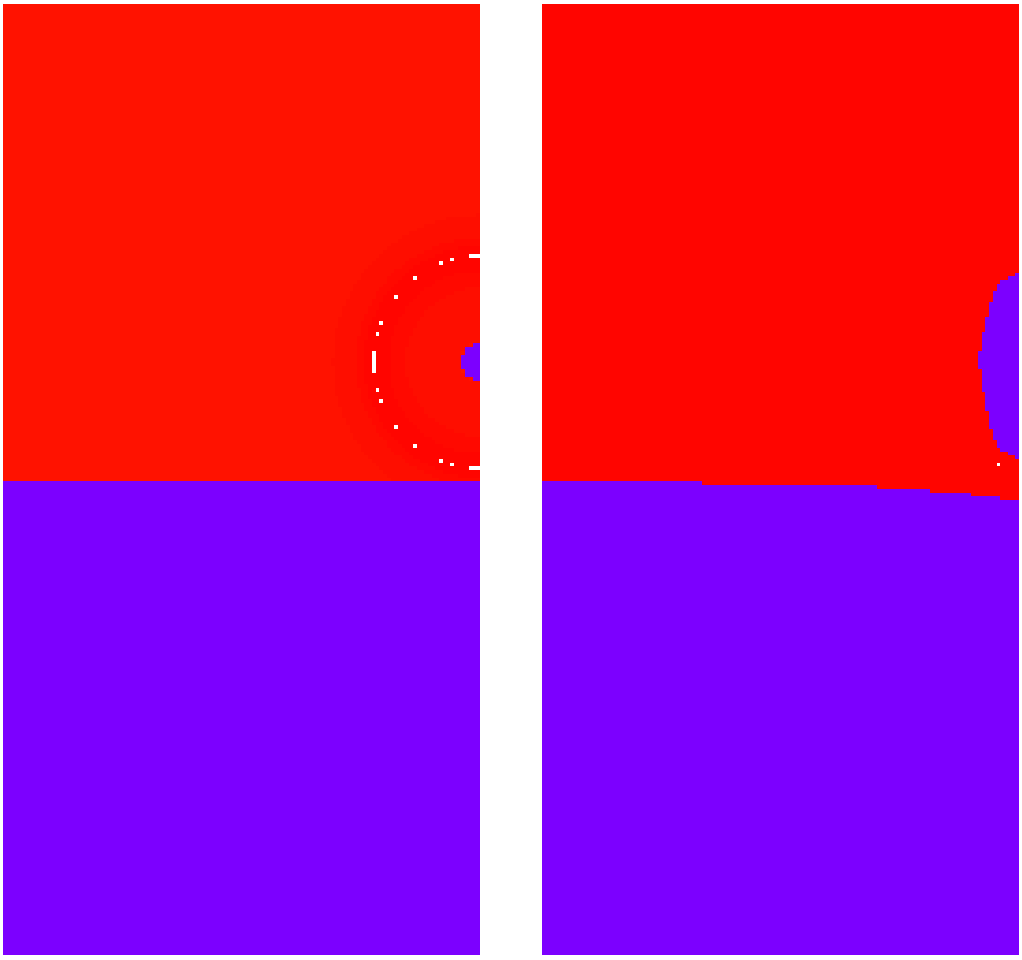


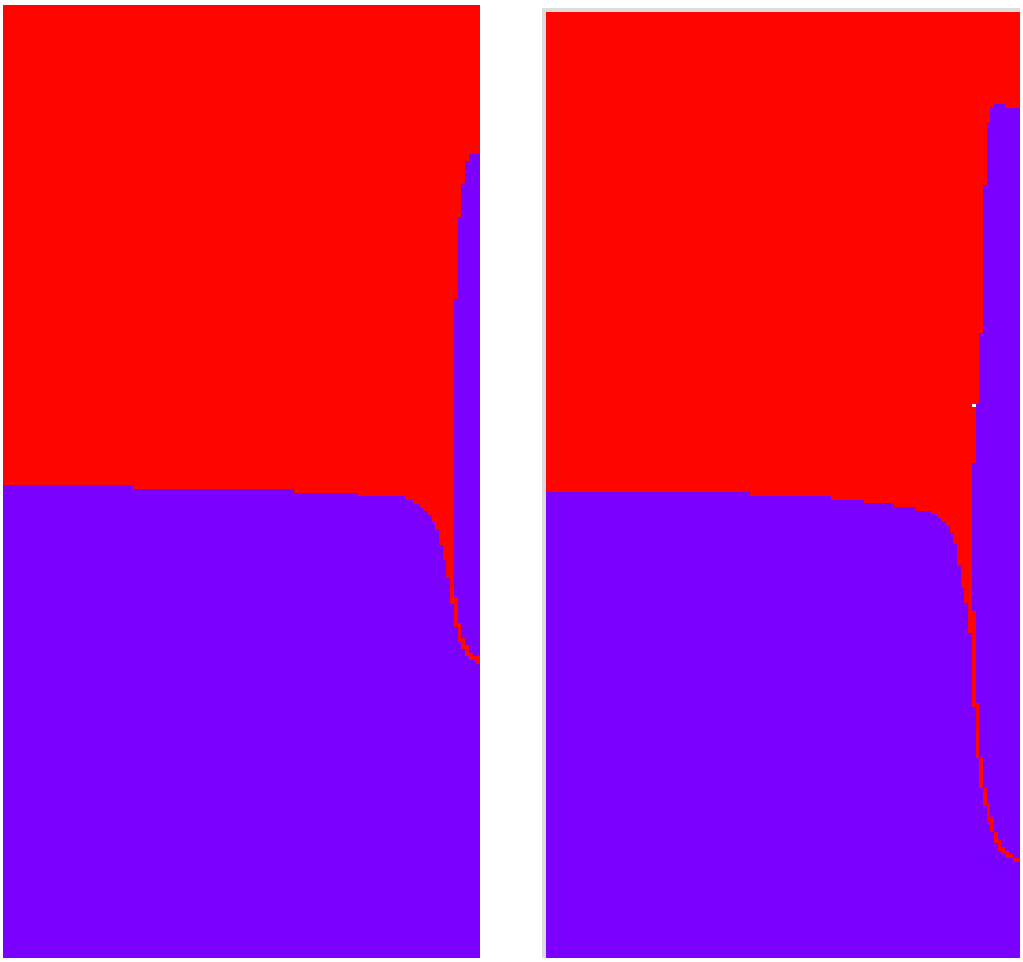
Figure 3.23: Pressure Profile from 1D Simulation

Taking the pressure profile at late time of 1D running (before shock wave hit the boundary) as initial conditions, 2D and 3D simulations are done with tracked bubble near the interface, both of which only contain partial of the global domain. The schematic picture of the approach is as shown in 3.24. The simulations are performed with grid size of 128×256 over domain $[0,0.04] \times [0,0.08]$. The initial radius of the bubble is 0.016 cm and the distance from the bubble center to the liquid/ambient gas interface is 0.01 cm. From the interface evolution profiles in Figure 3.24 and Figure 3.25, we can clearly see that the vapor bubble grows continuously and interface instability appears very quickly. Figure 3.26 shows the tip position of the instability as function of time and it is clear that the velocity can reach as high as 160m/s .



(a) $t = 0ms$ (b) $t = 0.007ms$

Figure 3.24: density profile for bubble distance to interface = 0.01 cm



(a) $t = 0.014ms$ (b) $t = 0.0025ms$

Figure 3.25: density profile for bubble distance to interface = 0.01 cm

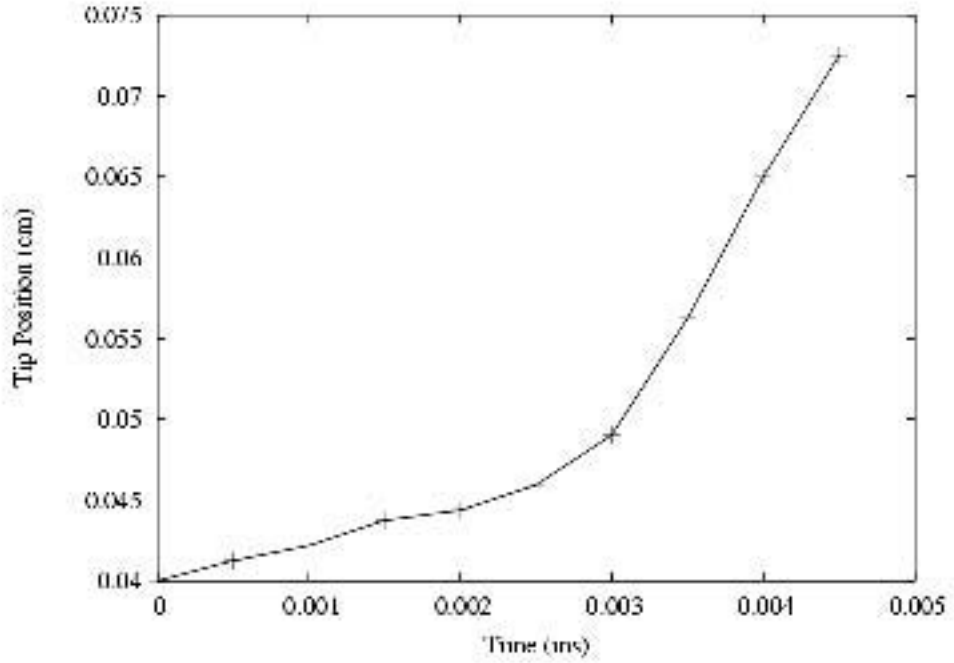
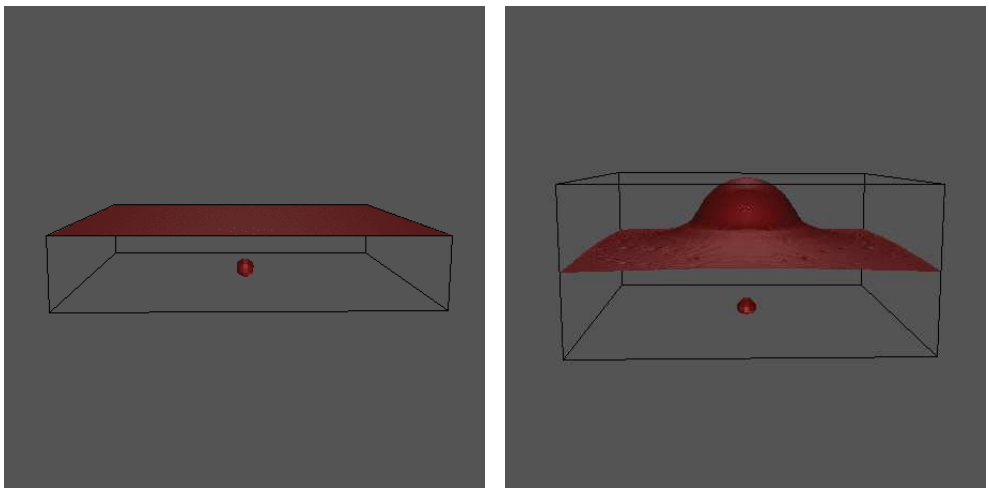
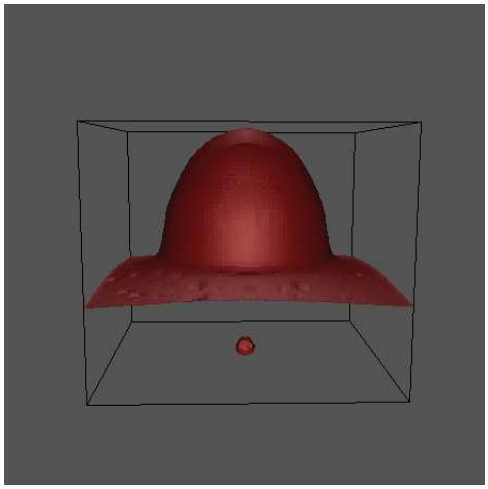


Figure 3.26: Position of interface tip (cm) Vs. Time(ms)

Similarly, 3D simulation is performed with grid 80 × 80× 80 over domain [0,0.04]×[0,0.04]×[0,0.04]. Figure 3.27 clearly shows the development of surface perturbations during the bubble rebounding procedure.



(a) $t = 0$ (b) $t = 0.0035ms$



(c) $t = 0.0075ms$

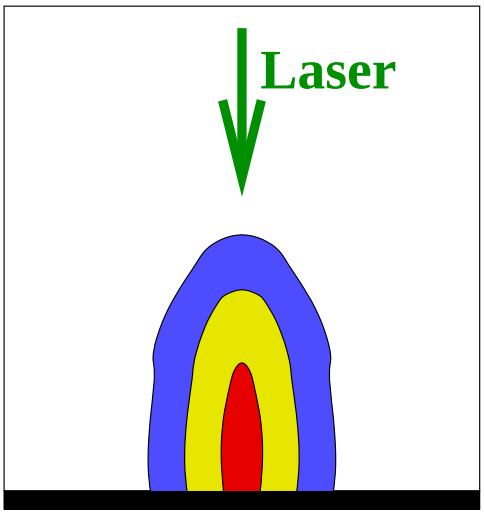
Figure 3.27: Interface profile during bubble rebounding

3.3 Laser Ablated Plasma Plume Simulation

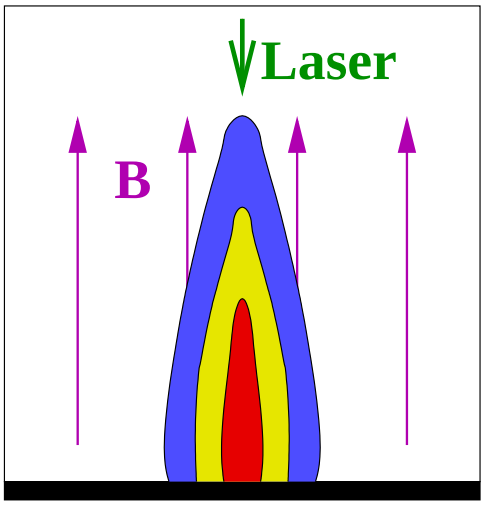
Magnetic fields are used for the control of the dynamic properties of transient plasmas induced by laser ablation [6]. The laser ablation in magnetic fields was studied experimentally in [6] as a method of diverting or extracting energy from the highly ionized expanding plasma generated in IFE target explosions. The plasma conductivity in IFE target explosions is very large and the ablation process is beyond the applicability of low magnetic Reynolds number approximation [6], at least at its initial stage. The plasma created in the process of nano-tube synthesis is usually weakly ionized and the induced magnetic field is usually much smaller than the applied field. We simulated the expansion of laser-generated plumes using the algorithm discussed before and the EOS model for partially ionized gas. The laser and material settings are similar to those mentioned in Povitsky's papers [40, 41]. We investigated the MHD effect and observed changes in plume structure and dynamics in the presence of the magnetic field.

In our computational model, the interface between the ablated plasma plume and the cold nonconducting ambient gas is tracked explicitly (See Figure 3.28). We compared results of 2D axisymmetric simulations and 3D simulations and found a good agreement. In transverse magnetic fields as in Figure 3.28, plumes become wedge-shaped and must be simulated in 3D.

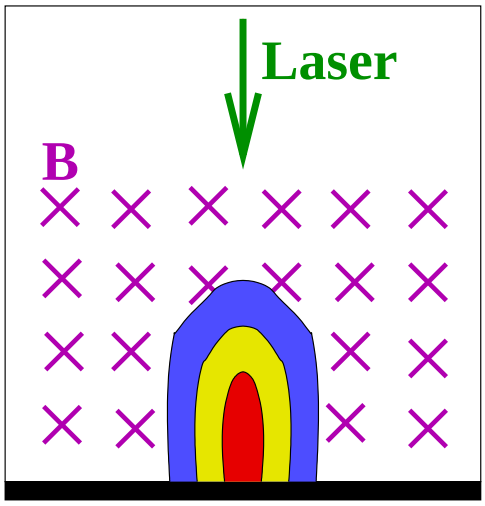
The interaction of the laser beam with the evaporating material leads to the formation of an isothermally expanding plasma and this persists until the end of the laser pulse; after the termination of the laser pulse, no particles are ejected from the target surface [42]. Our simulations were started from



(a) $B = 0$



(b) Longitudinal Field

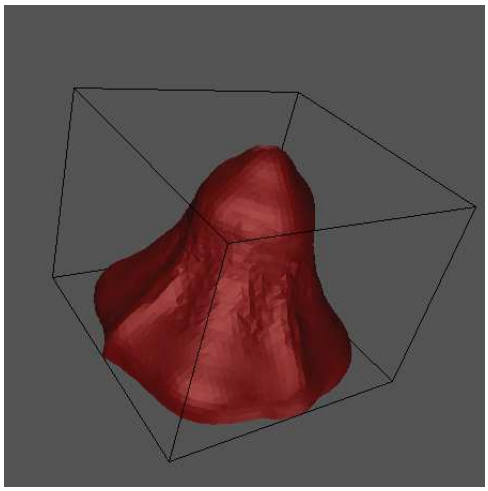


(c) Transverse Field

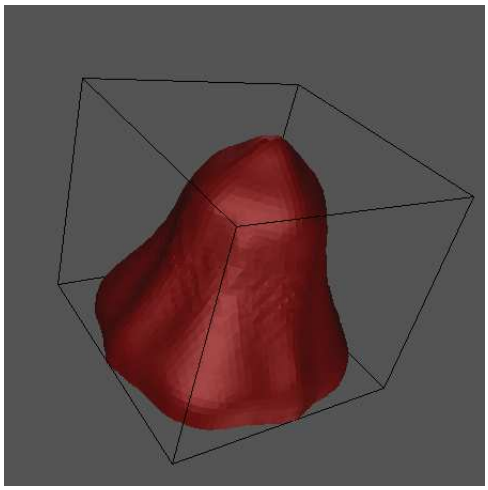
Figure 3.28: Schematic picture of plasma plume within magnetic field

either the beginning or the end of the laser pulse. In the first approach, the laser beam ablates the solid target and brings energy along with mass into the plasma plume, and the ablated material expands during the time of the laser pulse. After the termination of the pulse, the target surface is treated as a reflecting boundary. In the second approach, a thin layer near the target is initialized as the hot ablated material, and the target surface is always taken to be a reflecting boundary. Since the simulated plume expansion time (several microseconds) is much longer than the laser pulse duration (20 nanoseconds), both approaches gave essentially the same results.

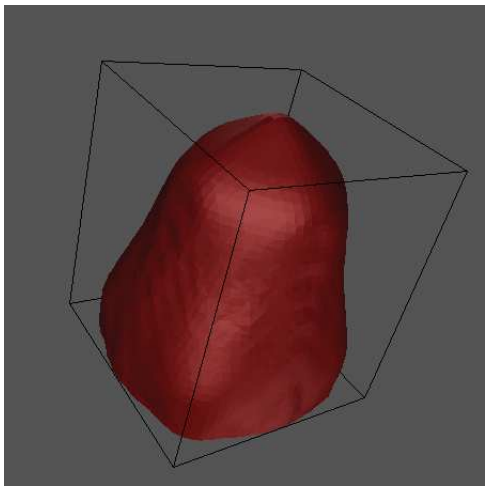
In the simulations presented here, the container is initially filled with the argon gas at temperature of $1500K\,(0.13eV)$ and pressure of 0.01 bar. A laser beam of $2mm$ diameter is focused on the surface of a carbon target, creating the ablated vapor with the initial pressure of 100 bar and temperature of $5eV$. Since the ionization energy of carbon is $11.26eV$, the carbon vapor is only weakly ionized. The injection velocity of the ablated material is set to zero. After the end of the laser pulse, the carbon surface is treated as a fixed wall (normal velocity of vapor on the wall is zero). Flow through condition is applied to all other boundaries of the computational domain to represent a large container that does not interfere with the expansion of the plasma plume.



(a) $B = 0$

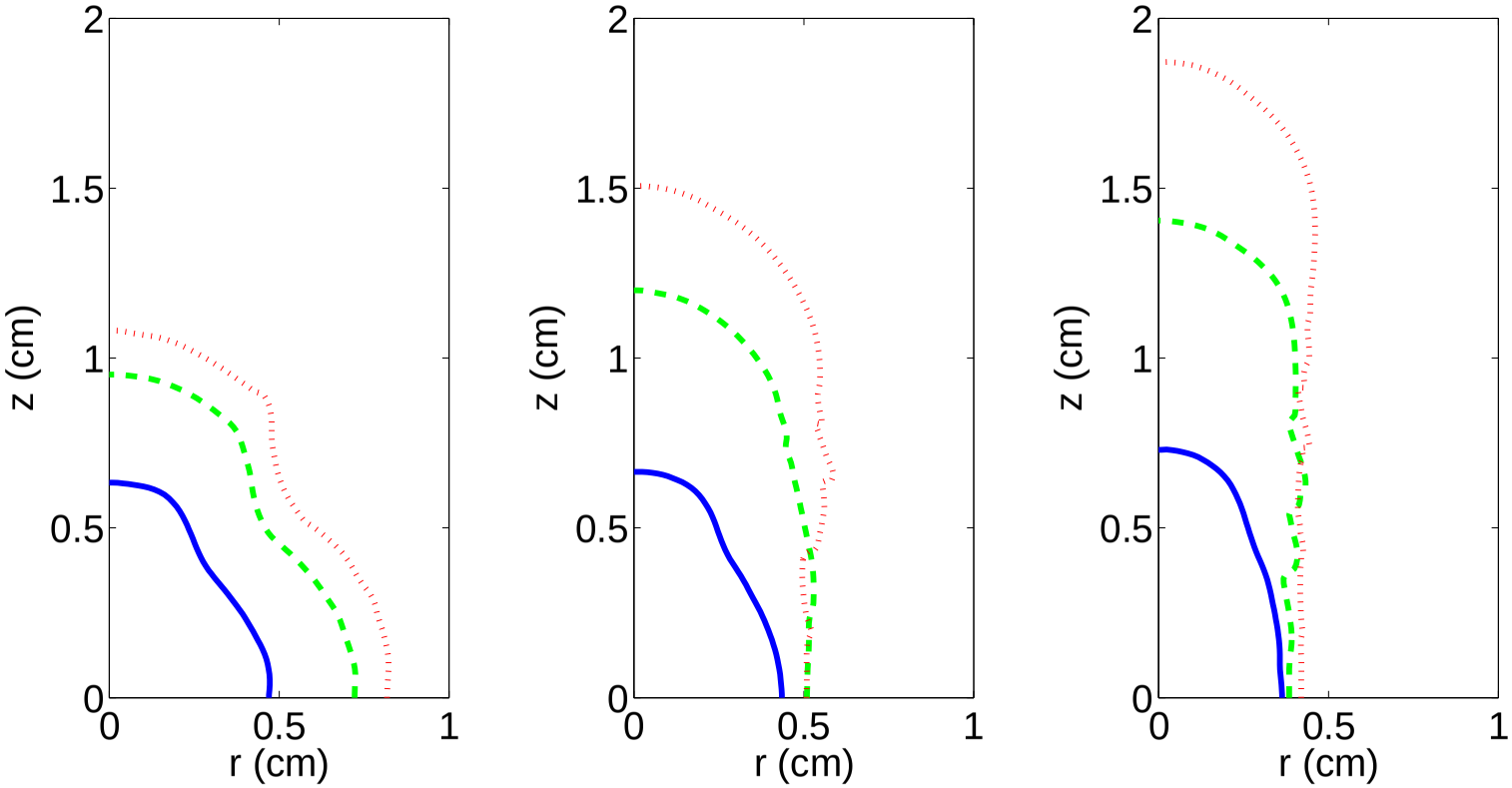


(b) $B_l = 1$Tesla



(c) $B_l = 2$Tesla

Figure 3.29: 3D plume fronts at $t = 1\mu s$ in longitudinal magnetic field

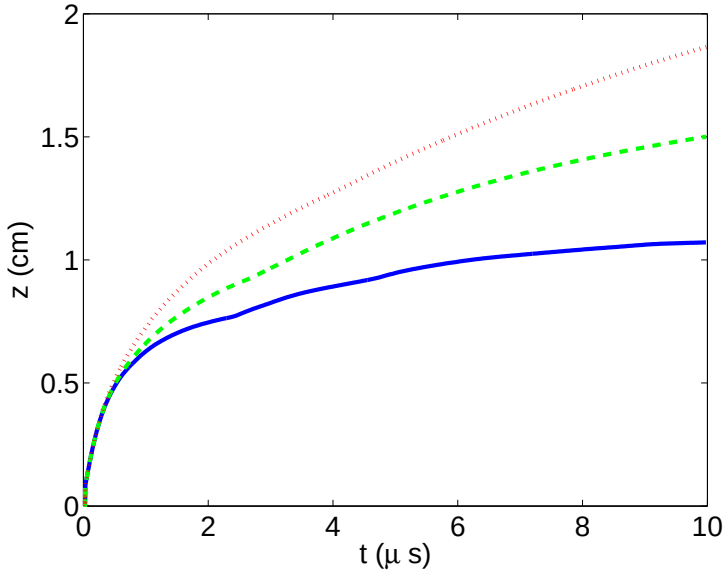


(a) $B = 0$ (b) $B_l = 1$Tesla (c) $B_l = 2$Tesla

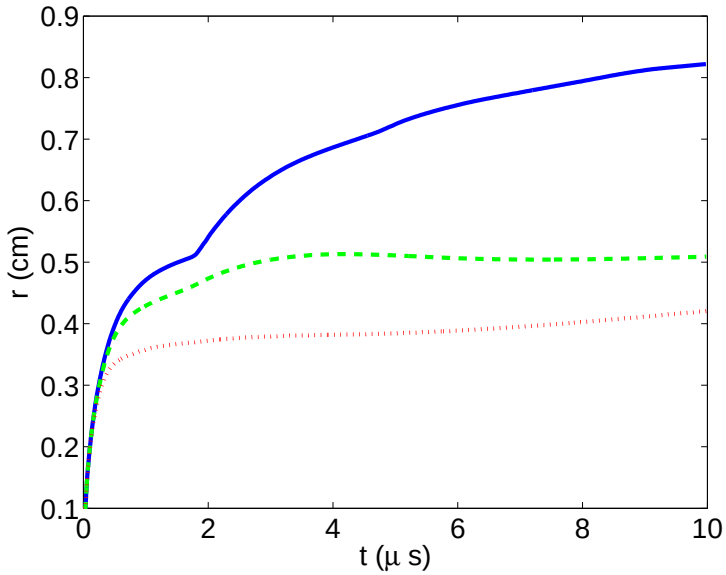
Figure 3.30: Evolution of the plume front in longitudinal magnetic fields. In each figure, the solid curve is for $t = 1\mu s$, the dashed curve is for $t = 5\mu s$, the dotted curve is for $t = 10\mu s$.

First we studied the evolution of the plume front in longitudinal magnetic fields. Both two dimensional axisymmetric and three dimensional simulations were carried out. The grid resolution used in the simulations was 0.02 cm, and the grid size for a typical 3D simulation is $200 \times 200 \times 100$. The 2D and 3D simulation results were essentially the same. The 3D plume fronts in longitudinal magnetic fields are shown in Figure 3.29. Figure 3.30 shows the evolution of the plume front in longitudinal magnetic fields. In the absence of the magnetic field, the plume front is close to spherical as shown in Figure 3.30(a). In magnetic fields, the plume is channelled along the field lines and the longitudinal expansion is boosted while the transverse expansion is suppressed, as shown in Figure 3.30(b) and 3.30(c).

Figure 3.31 shows the MHD effect of longitudinal magnetic fields on plume expansion quantitatively. It is clear from the figure that the spherical expansion in the absence of the magnetic field slows down quickly with increasing time. But in magnetic fields the expansion is almost one-dimensional – there is no significant slowing down of the longitudinal expansion, while the transverse expansion almost completely stops. The peak values of longitudinal and transverse expansion velocities at $t = 10\mu s$ in the absence of the magnetic field are 419 m/s and 439 m/s respectively. In the 2 Tesla field the corresponding values are 1310 m/s and 25 m/s. Besides the change of the plasma plume velocity distribution in the magnetic field, the plasma density and pressure significantly increase. For example, the peak value of the plume density at $10\mu s$ increases from $1.0 \times 10^{-7} g/cm^3$ in the absence of the magnetic field to $2.0 \times 10^{-7} g/cm^3$ in the 2 Tesla field, while the maximum pressure increases



(a) Longitudinal expansion



(b) Transverse expansion

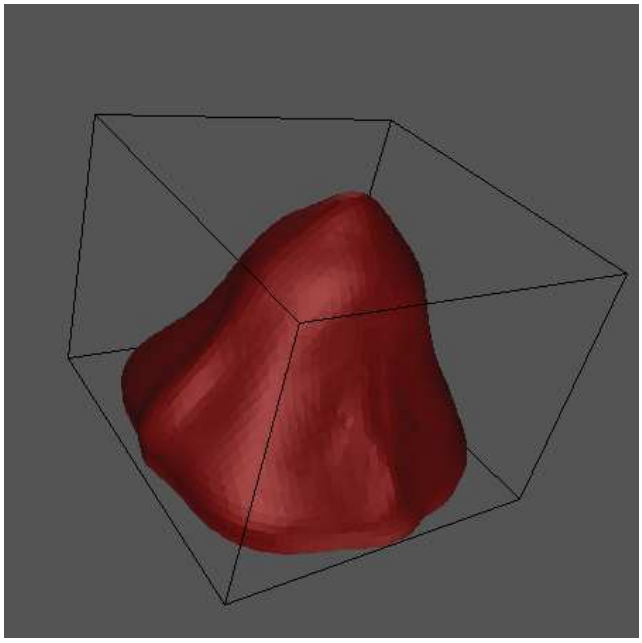
Figure 3.31: Longitudinal and transverse expansion of the plume in the longitudinal magnetic fields. In each figure, the solid curve is for $B = 0$, the dashed curve is for $B = 1$ Tesla, the dotted curve is for $B = 2$ Tesla.

from 9 mbar to 19 mbar.

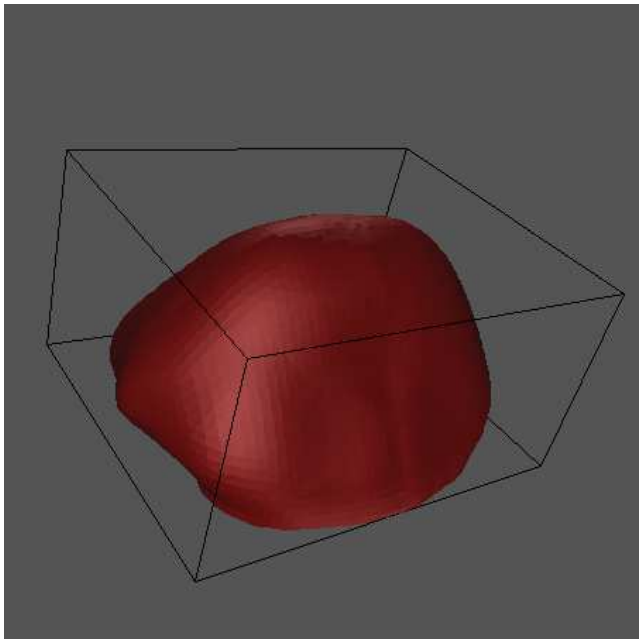
We also studied the effect of transverse magnetic fields on the plasma plume expansion. Due to the loss of axisymmetry, only three dimensional simulations are applicable. The plasma plumes in 1 and 2 Tesla transverse magnetic fields are plotted in the Figure 3.32. It is seen that with the increase of the magnetic field, the plume expansion is extended in one direction (parallel to field lines) and reduced in other directions (perpendicular to the field lines).

Assuming the magnetic field along y-axis, we also plotted the contours of the density distribution in the cross sections of the plume in xz and yz planes respectively in Figure 3.33. In the plane perpendicular to the magnetic field, the plume density is higher in the center as expected. In the plane parallel it is found that the plume density is approximately uniform along the field lines. The dark regions in the bottom of the plume stand for the nearly vacuum region created by the fast expansion of plasma plume. The zigzag boundary of the plume comes from the interpolation used in the calculation of contours, the actual tracked interface is a smoother triangulated two dimensional manifold.

From the MHD simulations of plasma plume expansion, we conclude that the magnetic field is an effective tool to enhance, control, or direct the expansion. Finally, to justify the the low magnetic Reynolds number approximation, we calculated the induced magnetic field ΔB at the center of the plume using Biot-Savart Law. Indeed we found that $\Delta B/B \ll 1$ in our simulations. For the longitudinal magnetic field, no matter whether $B = 1$ Tesla or $B = 2$ Tesla, the ratio $\Delta B/B$ attains maximum at about 90 nanoseconds after the end of the laser pulse, and the maximum value of $\Delta B/B$ is only 0.002.

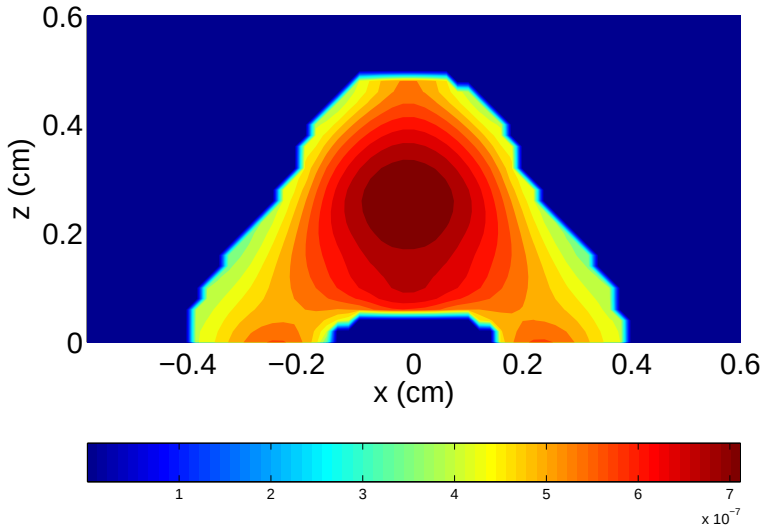


(a) $B_t = 1$ Tesla

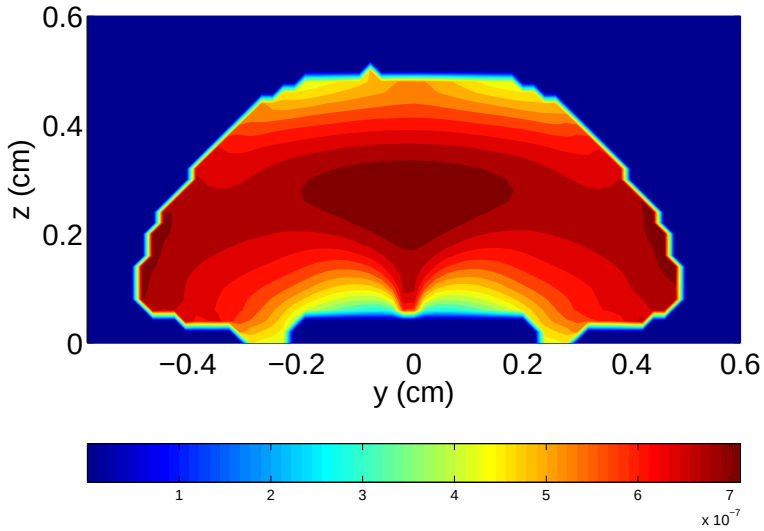


(b) $B_t = 2$ Tesla

Figure 3.32: Three dimensional plume fronts at $t = 1\mu s$ in transverse magnetic fields.



(a) Perpendicular to **B**



(b) Parallel to **B**

Figure 3.33: Density distribution in the cross sections of the plasma plume in 2 Tesla transverse magnetic fields. $t = 1\mu s$. The unit of the density if g/cm^3.

Chapter 4

Conclusions and Future Work

4.1 Conclusions and Future Work

We have developed a numerical algorithm and computational software for the numerical simulation of free surface magnetohydrodynamic flows at low magnetic Reynolds numbers. The software is applicable to the simulation of free surface MHD flows of conducting liquids undergoing phase transitions and weakly ionized gases. The corresponding governing equations constitute a coupled hyperbolic - elliptic system in a geometrically complex and evolving domain. The numerical algorithm includes the interface tracking technique for the hyperbolic problem, a Riemann problem for the material interface, discretization of elliptic equations in irregular domains with interface constraints using the embedded boundary method, and high performance parallel solvers such as MUSCL-type schemes for hyperbolic problems and iterative solvers implemented in the PETSc package.

We have validated the elliptic technique using an exact solution of a Poisson problem with a Neumann boundary condition in geometrically complex

domain and showed that it is second order accurate for the electric potential and its gradient. Comparisons with the mixed finite element methods have also been made. The MHD code has been validated by comparing numerical simulations with analytical solutions in terms of expansion series for the problem a liquid mercury jet moving in a non-uniform magnetic field. A good agreement of simulations with experimentally validated theoretical calculations has been achieved.

The algorithm has been applied to the numerical simulations of the future Muon Collider/Neutrino Factory target. Both 2D axisymmetric and 3D simulations of MHD processes in the mercury jet target interacting with a high intensity proton pulse in a strong magnetic field have already been performed. Different eos models(homogeneous and heterogeneous model) and conductivity models have been applied for the multiphase flow. The restriction effect of the magnetic field on the mercury jet dispersal has been verified qualitatively. Also with the approach of multiscale, we found that bubble collapsing and rebounding may play important roles in jet breakup.

The code has also been applied to the simulation of the expansion of laser-generated plasma plumes in magnetic fields. Both longitudinal and transverse magnetic fields have been studied. We confirmed quantitatively that the longitudinal magnetic field would channel the plasma plume. In the transverse magnetic field, the plume expansion is directed along the magnetic field lines, and the expansion in other directions is reduced. Therefore our MHD simulations confirms the effectiveness of magnetic fields as a method to control the ablation flow, and provides a way to test physical settings without going

through expensive experiments.

Bibliography

[1] J.Glimm R.Samulyak J.Du, S.Wang. A comparision study of two methods for elliptic boundary value problems. 2007.

[2] http://www.physics.princeton.edu/mumu/target/.

[3] P. Parks and R. Turnbull. Effect of transonic flow in the ablation cloud on the lifetime of a solid hydrogen pellet in plasma. *Phys. Fluids*, 21:1735–1741, 1978.

[4] P. Parks. Magnetic-field distortion near an ablating hydrogen pellet. *Nucl. Fusion*, 20:311, 1980.

[5] R. Samulyak, T. Lu, and P. Parks. A magnetohydrodynamic simulation of pellet ablation in the electrostatic approximation. *Nucl. Fusion*, 47:103–118., 2007.

[6] C. V. Bindhu S. S. Harilal and M. S. Tillack. Confinement and dynamics of laser-produced plasma expanding across a transverse magnetic field. *Phys. Rev. E.*, 69:026413, 2004.

[7] N. Morley, S. Smolentsev, R. Munipalli, M.-J. Ni, D. Gao, and M. Abdou. Progress on the modeling of liquid metal, free surface mhd flows for fusion liquid walls. *Fusion Eng. Design*, 72:3–34, 2004.

[8] Palmer R. Zisman M. Gallardo J. (editors) Ozaki, S. Feasibility study-ii of a muon-based neutrino source. Technical Report BNL-52623, BNL, 2001.

[9] D. J. Jackson. *Classical Electrodynamics*. John Wiley & Sons, New York-Toronto-Singapore, 1974.

[10] I. R. Chern, J. Glimm, O. McBryan, B. Plohr, and S. Yaniv. Front tracking for gas dynamics. *J. Comput. Phys.*, 62:83–110, 1986.

[11] J. Glimm, J. Grove, X. L. Li, K. L. Shyue, Q. Zhang, and Y. Zeng. Three dimensional front tracking. *SIAM J. Sci. Comput.*, 19:703–727, 1998.

[12] B. van Leer. Towards the ultimate conservative difference scheme: V. a second order sequel to godunov's method. *J. Comput. Phys.*, 32:101–136, 1979.

[13] H. Johansen and P. Colella. A cartesian grid embedded boundary method for poisson's equation on irregular domains. *J. Comput. Phys.*, 147:60–85, 1998.

[14] R. Samulyak, T. Lu, Y. Prykarpatskyy, J. Glimm, Z. Xu, and M.N. Kim. Comparison of heterogeneous and homogenized numerical models of cavitation. *Int. J. Multiscale Comput. Eng.*, 4:377 – 389, 2006.

[15] R. Menikoff and B.J. Plohr. The riemann problem for fluid flow of real materials. *Rev. Mod. Phys.*, 61:75–130, 1989.

[16] Tuck C. Choy. *Effective medium theory: principles and applications*. Oxford University Press, 1999.

[17] C. E. Brennen. *Cavitation and Bubble dynamics*. Oxford University Press, 1995.

[18] S Balibar and F. Caupin. Metastable liquids. *J. of Physics: Condensed Matter*, 15:s75–s82, 2003.

[19] R. Prater F. S. Felber, P. B. Parks and D. Vlasow. *Nucl. Fusion*, 19:1061, 1979.

[20] N. Nakajima R. Ishizaki, P. B. Parks and M. Okamoto. *Phys. Plasmas*, 11:4064, 2004.

[21] Y. P. Raizer Y. B. Zel'dovich. *Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena*. Academic, New York, vol. i. edition, 1966.

[22] J. Glimm, J. Grove, and Y. Zhang. Interface tracking for axisymmetric flows. *SIAM J. Sci. Comput.*, 24:208–236, 2002.

[23] T. W. Johnston I. P. Shkarofsky and M. P. Bachynski. *The Particle Kinetics of Plasmas*. Addison-Wesley, 1966.

[24] J. Glimm, J. Grove, X. L. Li, and D. C. Tan. Robust computational algorithms for dynamic interface tracking in three dimensions. *SIAM J. Sci. Comput.*, 21:2240–2256, 2000.

[25] R. Courant and K. Friedrichs. *Supersonic Flows and Shock Waves*. Interscience, New York, 1948.

[26] P. Colella and P. Woodward. The piesewise parabolic method (ppm) for gas-dynamical simulations. *J. Comput. Phys.*, 54:174–201, 1984.

[27] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys.*, 87:171–200, 1990.

[28] J. Glimm, E. Isaacson, D. Marchesin, and O. McBryan. Front tracking for hyperbolic systems. *Adv. Appl. Math.*, 2:91–119, 1981.

[29] J. Glimm, D. Marchesin, and O. McBryan. Subgrid resolution of fluid discontinuities II. *J. Comp. Phys.*, 37:336–354, 1980.

[30] P. Schwartz, M. Barad, P. Colella, and T. Ligocki. A cartesian grid embedded boundary method for the heat equation and poisson's equation in three dimensions. *J. Comput. Phys.*, 211:531 – 550, 2006.

[31] S. Balay, W. Gropp, L. McInnes, and B. Smith. Petsc users manual. Technical Report Revision 2.1.1, Argonne National Laboratory, 2001.

[32] G. Chavent and J.E. Roberts. A unified physical presentation of mixed, mixed-hybrid finite elements and standard finite difference approximations for the determination of velocities in waterflow problems. *Adv. Water Resources*, 14(6):329–348, 1991.

[33] Mark A.Yerry and Mark S.Shephard. A modified quadtree approach to finite element generation. *IEEE Comput. Graph. Appl.*, 3(1):39–46, 1983.

[34] Shuqiang Wang. *A Mixed Finite Element Method for Elliptic Interface/Boundary Value Problem*. PhD thesis, SUNY at Stony Brook, December 2006.

[35] W. Briggs, V.E. Henson, and S. McCormic. *A Multigrid Tutorial*. SIAM publications, second edition, 2000.

[36] S. Oshima, R. Yamane, Y. Moshimaru, and T. Matsuoka. The shape of a liquid metal jet under a non-uniform magnetic field. *JCME Int. J.*, 30:437 – 448, 1987.

[37] N.V. Mokhov. Mars code developments, benchmarking and applications. 2000.

[38] http://www.t4.lanl.gov/opacity/sesame.html.

[39] et al. H. Kirk. Target studies with bnl e951 at the ags. In *Particles and Accelerators 2001*, June 18-22 2001.

[40] D. C. Lob ao and A. Povitsky. Modeling of plume dynamics and shock wave in laser ablation with application to nano-technology. *Lecture Notes in Computer Science*, 2668:871, 2003.

[41] D. C. Lob ao and A. Povitsky. Interaction of plume with shock waves in laser ablation. *AIAA Journal*, 43(3):595, 2005.

[42] M. S. Tillack F. Najmabadi S. S. Harilal, C. V. Bindhu and A. C. Gaeris. Internal structure and expansion dynamics of laser ablation plumes into ambient gases. *J. Appl. Phys.*, 93(5):2380, 2003.