Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 226 (2007) 1532–1549
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A numerical algorithm for MHD of free surface flows at low magnetic Reynolds numbers

Roman Samulyak [a,*], Jian Du [b], James Glimm [a,b], Zhiliang Xu [a]

[a] *Computational Science Center, Brookhaven National Laboratory, Upton, NY 11973, USA*
[b] *Department of Applied Mathematics and Statistics, SUNY at Stony Brook, Stony Brook, NY 11794, USA*

Received 7 September 2006; received in revised form 6 April 2007; accepted 1 June 2007
Available online 14 June 2007

Abstract

We have developed a numerical algorithm and computational software for the study of magnetohydrodynamics (MHD) of free surface flows at low magnetic Reynolds numbers. The governing system of equations is a coupled hyperbolic–elliptic system in moving and geometrically complex domains. 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 numerical algorithm has been implemented as an MHD extension of FronTier, a hydrodynamic code with free interface support. The code is applicable for numerical simulations of free surface flows of conductive liquids or weakly ionized plasmas. The code has been validated through the comparison of numerical simulations of a liquid metal jet in a non-uniform magnetic field with experiments and theory. Simulations of the Muon Collider/Neutrino Factory target have also been discussed.

Keywords: Magnetohydrodynamics; Free surface flows; Front tracking; Embedded boundary

1. Introduction

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 (see [18–20] and references therein) 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

* Corresponding author. Tel.: +1 631 344 3304.
E-mail address: rosamu@bnl.gov (R. Samulyak).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.06.005

tokamak refueling devices [24,28], laser ablation in magnetic fields [12], 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 [20] 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) [23,24], laser–plasma interaction, and the interaction of a liquid mercury jet with proton pulses in target devices for future advanced accelerators [22] are among numerous examples of such MHD problems. For the simulation of such processes, we propose in this paper a 3D numerical algorithm and describe its implementation.

The algorithm solves the compressible equations for fluid flows in the low magnetic Reynolds number approximation [18] for electromagnetic forces. Mathematically, the governing system of equations is a coupled hyperbolic–elliptic system in geometrically complex and evolving domains. We use the method of front tracking [9] for the propagation of fluid interfaces. Our FronTier code is capable of tracking and resolving topological changes of large numbers of interfaces in two and three-dimensional spaces [10]. In the method of front tracking, the interface is a Lagrangian mesh moving through a volume filling rectangular mesh according to solutions of Riemann problems for interface points. High resolution solvers based on second order Godunov methods are used to update hyperbolic states in the interior away from interfaces. The embedded boundary method [15] is used for solving the elliptic problem in geometrically complex domains bounded by fluid interfaces. The explicit treatment of interfaces typical for the method of front tracking is especially advantageous for multiphysics problems involving phase transitions. It allows not only to solve accurately the Riemann problem for the phase boundary, but also to apply different mathematical approximations in the regions separated by interfaces to account for different material properties and, if necessary, eliminate fast time scales in numerical simulations.

We validate the method by simulating the distortion of the liquid mercury jet entering a non-uniform magnetic field and comparing simulation results with analytical solutions in terms of expansion series [21]. The problem is also significantly important for the Neutrino Factory/Muon Collider, a future advanced accelerator [22]. The target has been proposed as a liquid mercury jet entering a 15 Tesla solenoid and interacting with an intense proton pulse in the solenoid center. Results of this paper are relevant to the first process in the target – the entrance of the mercury jet in the magnetic solenoid. Simulations of the mercury jet expansion, cavitation, and surface instabilities due to the interaction with proton pulses have already been performed using our MHD code in 2D axisymmetric approximation [25,26]. Full 3D studies are in progress and will be reported in a forthcoming paper.

The paper is organized as follows. In Section 2, we introduce the system of governing equations and discuss mathematical approximations. The numerical algorithm, its implementation in the FronTier code, and validation are described in Section 3. Applications of FronTier to the numerical simulation of a mercury jet entering a non-uniform magnetic field and relevance to design of the Neutrino Factory/Muon Collider target are presented in Section 4. Finally, we conclude the paper with a summary of our results and perspectives for future work.

2. Governing equations

The system of MHD equations [14] 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:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{u}), \tag{1}$$

$$\rho\left(\frac{\partial}{\partial t}+\mathbf{u}\cdot\nabla\right)\mathbf{u}=-\nabla P+\rho\mathbf{g}+\frac{1}{c}(\mathbf{J}\times\mathbf{B}), \tag{2}$$

$$\rho\left(\frac{\partial}{\partial t}+\mathbf{u}\cdot\nabla\right)e=-P\nabla\cdot\mathbf{u}+\rho\mathbf{u}\cdot\mathbf{g}+\frac{1}{\sigma}\mathbf{J}^2, \tag{3}$$

$$\frac{\partial\mathbf{B}}{\partial t}=\nabla\times(\mathbf{u}\times\mathbf{B})-\nabla\times(\frac{c^2}{4\pi\sigma}\nabla\times\mathbf{B}), \tag{4}$$

$$\nabla\cdot\mathbf{B}=0, \tag{5}$$

$$P=P(\rho,e). \tag{6}$$

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:$\mathbf{B}=\mu\mathbf{H}$. In the system (1)–(4), we neglected effects of the heat conduction and viscosity. Eq. (5) is the solenoidal property of the magnetic field, and (6) is the equation of state (EOS). EOS models for compressible conducting media are briefly discussed in Section 3.1. We write electrodynamic equations in Gaussian units throughout the paper. But we compare our results with other studies using more convenient units for some quantities, such as Tesla for the magnetic field and m/s for the velocity.

If the magnetic Reynolds number,

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

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)–(5) can be simplified. Namely, the time evolution of the magnetic field (4) can be neglected, and the current density distribution can be obtained from Ohm's law

$$\mathbf{J}=\sigma\left(-\nabla\varphi+\frac{1}{c}\mathbf{u}\times\mathbf{B}\right), \tag{7}$$

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

$$\nabla\cdot(\sigma\nabla\varphi)=\frac{1}{c}\nabla\cdot\sigma(\mathbf{u}\times\mathbf{B}). \tag{8}$$

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

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

Eq. (5) is automatically satisfied for an external magnetic field created by a realistic source. The low magnetic Reynolds number approximation is applicable to moderately conductive media, such as liquid metals or salts, and weakly ionized plasmas. Its application to the problem of the pellet fueling of tokamaks was justified in [28].

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:

$$\Delta P|_{\Gamma}=T\left(\frac{1}{r_1}+\frac{1}{r_2}\right); \tag{9}$$

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

$$\left.\frac{\partial\varphi}{\partial\mathbf{n}}\right|_{\Gamma} = \frac{1}{c}(\mathbf{u}\times\mathbf{B})\cdot\mathbf{n}, \tag{10}$$

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

In this paper, we propose a numerical algorithm for the MHD system of equations in low Re^M approximation 1, 2, 3, (7), (8) for free surface flows. A similar technique can be applied for the full system of MHD equations (1)–(5) for free surface flows.

3. Numerical algorithm and implementation

The governing system of Eqs. (1)–(3), (7), (8), a coupled hyperbolic–elliptic system in a geometrically complex moving domain, is solved using operator splitting. 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 [3,9] is performed at the beginning of a time step. Then interior states are updated by second order hyperbolic solvers such as the Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL) [30]. At the end of the time step, the elliptic system is solved using a finite volume discretization with interface constraints (the embedded boundary method [15]), and the interior states are updated by adding electromagnetic source terms. In the next two sections, we describe numerical algorithms for the hyperbolic and elliptic subsystems and their implementation in the FronTier code.

3.1. Hyperbolic problem and free surface propagation

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 [3,9]. Front tracking is implemented in FronTier [10,11], 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.

For free surface MHD flows, we are interested in tracking only fluid interfaces which are contact discontinuity hypersurfaces of the corresponding Riemann problem [6]. 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 [11], 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 [11]. 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

$$\begin{aligned}
&\frac{\partial\rho}{\partial t}+\frac{\partial\rho v_N}{\partial\mathbf{n}}+\frac{\alpha n_0}{r}\rho v_N = 0,\\
&\frac{\partial\rho v_N}{\partial t}+\frac{\partial(\rho v_N^2+P)}{\partial\mathbf{n}}+\frac{\alpha n_0}{r}\rho v_N^2 = \rho g_N+\frac{1}{c}(\mathbf{J}\times\mathbf{B})_N,\\
&\frac{\partial\rho v_T}{\partial t}+\frac{\partial\rho v_N\mathbf{v}_T}{\partial\mathbf{n}}+\frac{\alpha n_0}{r}\rho v_N\mathbf{v}_T = 0,\\
&\frac{\partial\rho E}{\partial t}+\frac{\partial(\rho E v_N+P v_N)}{\partial\mathbf{n}}+\frac{\alpha n_0}{r}(\rho E v_N+P v_N) = \rho g_N v_N.
\end{aligned} \tag{11}$$

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, 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 [11].

A 5-point stencil for the point propagation algorithm is schematically shown in Fig. 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, [11] and references therein). In the prediction step (see Fig. 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_I at the end of the time step. The middle states of the solution of the Riemann problem also provide the interface states s_{-I} and s_{+I} at the new interface position x_I.

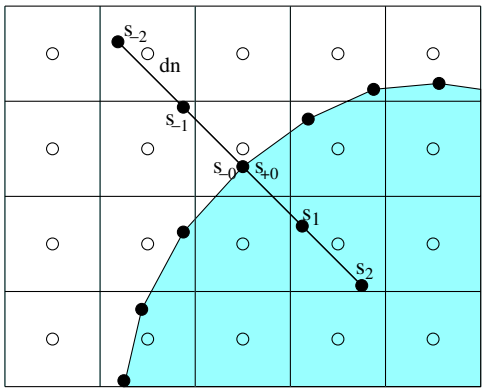

Fig. 1. Schematic of a stencil for the normal point propagation algorithm.

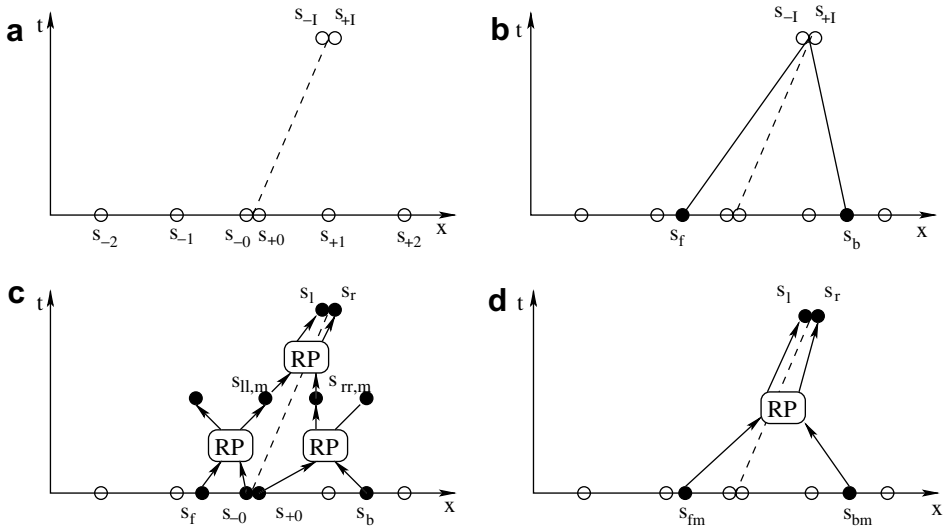

Fig. 2. Schematic of the normal point propagation algorithm. RP denotes the procedure of solving the Riemann problem and extracting of middle states from the solution.

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 (Fig. 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 [11], can be replaced by the following sequences of Riemann problems. The Riemann problem with the input states $s_{\mathbf{f}}$ and s_{-0} is solved and the right wave state of the solution, s_{ll}, approximates the wave incoming on the contact at time $t_0 + \mathrm{d}t$ from the left. Correspondingly, the Riemann problem with the input states s_{b} and s_{+0} is solved and the left wave state of the solution, s_{rr}, approximates the wave incoming on the contact at time $t_0 + \mathrm{d}t$ from the right. The states s_{ll} and s_{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 $t_0 + \mathrm{d}t$. The procedure is illustrated in Fig. 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 Fig. 2d. This reduces the computational time by eliminating two relatively expensive Riemann problem solving steps for every interface point.

Two techniques for the redistribution of interfaces and resolving their topological changes, the grid free and grid based tracking, have been developed [10]. 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. 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 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 [30] type schemes such as the Piecewise Linear, Piecewise Parabolic Method [4], a second order MUSCL scheme developed by I. L. Chern, and an unsplit MUSCL scheme [5]. An exact and several approximate Riemann solvers are available for use by these methods.

In general, the electrical conductivity changes the equation of state of a compressible medium. Suitable for the FronTier hyperbolic solvers and complete in the sense of [17] EOS for weakly ionized plasmas was developed in [28]. The EOS deals with gases (plasmas) for which the dissociation and ionization fractions are given by the corresponding Saha equations and therefore are functions of the state. This EOS has been implemented in the FronTier code. The test problems presented in this paper deal with mercury in weakly compressible regime. We assume that the number density of free electrons and the electrical conductivity are constant. As it can be seen from formulas of the weakly ionized EOS model of [28], the functional form of the pressure equation is not changed compared to the standard polytropic gas EOS model in the case of constant dissociation and ionization fractions, and the energy equation is changed only by a constant. Therefore a non-conducting fluid or gas EOS model can be used in the case of constant conductivity. We use a stiffened polytropic EOS for mercury that describes a single phase fluid with tension [6,17]. In the example of a heterogeneous modeling of multiphase fluid (mercury with vapor bubbles), discussed briefly in Section 4, the stiffened polytropic EOS is used for the liquid mercury, and the polytropic EOS is used for the vapor, which is assumed non-conducting.

3.2. Elliptic problem for irregular domains

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 [15]. 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 Eq. (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

$$(L\varphi)_i = \frac{1}{V_i}\left(\sum_j \mathbf{F}_j \cdot \mathbf{n}_j S_j\right), \tag{12}$$

where $L = \nabla \cdot \sigma\nabla$ and 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 [15,29], the interpolation of fluxes through edges of partial cells is necessary for stability and second order accuracy. The flux interpolation method is illustrated in Fig. 3. 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 $\mathbf{F}_{j+1}$, which are the finite differences of values in the centers of the corresponding regular grid blocks. 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. For instance, the normal flux through the point G is $(\varphi(i+1,j,k) - \varphi(i,j,k))/h_x$. The flux at the domain boundary (interface) is given by the Neumann condition.

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 [15].

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

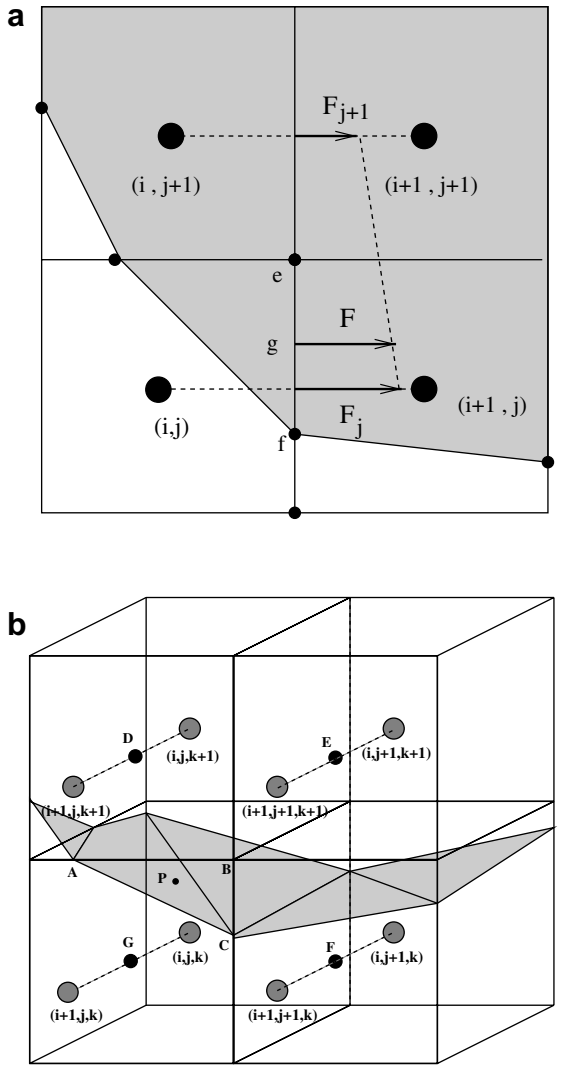

Fig. 3. Schematic of the linear and bilinear flux interpolation. (a) Linear flux interpolation in 2D and (b) bilinear flux interpolation in 3D.

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. A 9-point stencil is set to calculate fluxes across the control volume BADEF, as shown in Fig. 4, 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 potential 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 Fig. 4, 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:

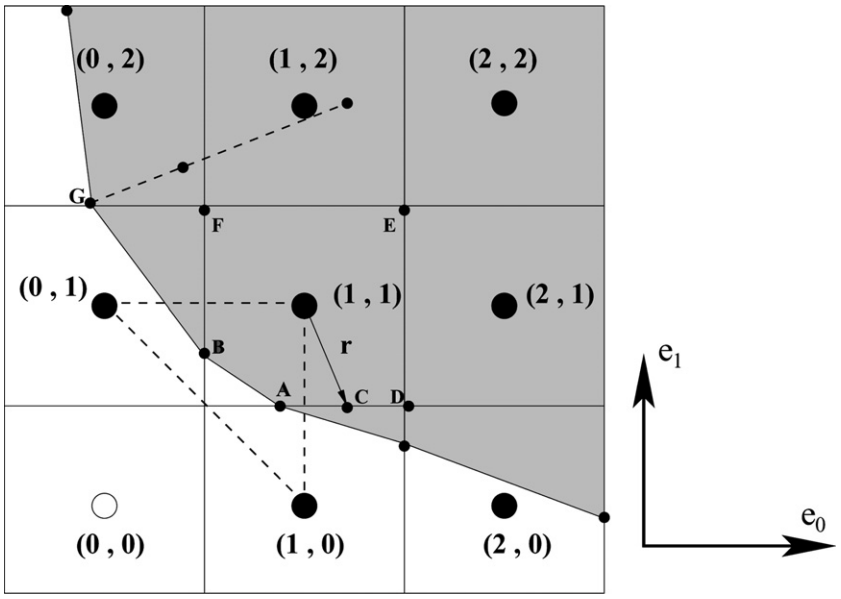

Fig. 4. Stencil setting for a partial grid block.

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

where d',$d = 0,1$ and $d' \neq d$, h_d is the grid spacing in the direction d, and $a = \frac{|\mathbf{r}\cdot bfe_{d'}|}{h_{d'}}$ is the block element aperture.In 3D, a 27-point stencil is used and the coefficients form a $3\times3\times3$ matrix. Using the similar notations as above, the bilinearly interpolated flux in the direction d has the stencil values:

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

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

(4) 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 Eq. (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 Eq. (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 Eq. (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.

(5) The resulting linear system $A\mathbf{x} = \mathbf{b}$ is solved. We use preconditioners and iterative solvers implemented in PETSc [1] and HYPRE [13] 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 Fig. 4), 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 Fig. 4), 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.

(6) 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.

3.3. Validation of the elliptic technique

An extensive theoretical analysis of the method of front tracking for hyperbolic systems of conservations laws has already been performed, and the method has been validated and tested on problems of Rayleigh–Taylor and Richtmyer–Meshkov surface instabilities (see for example [7,8] for the comparison of theoretical, numerical, and experimental data of Rayleigh–Taylor mixing rates). 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

$$\Delta\varphi = \mathbf{f},$$
$$\left.\frac{\partial\varphi}{\partial\mathbf{n}}\right|_{\Gamma} = g,$$

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, both shown in Fig. 5.

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

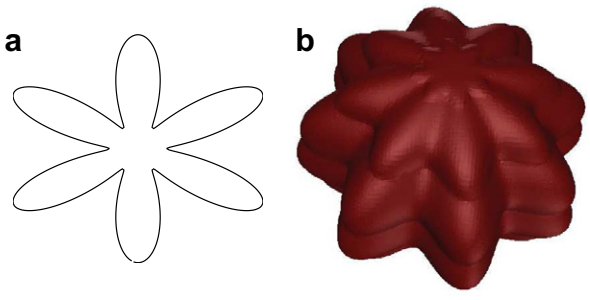

Fig. 5. Computational domains for the elliptic problem validation. (a) 2D domain and (b) 3D domain.

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

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 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. The distribution of the gradient error in the 2D problem is shown in Fig. 6. 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.

We would like to comment on the computational cost of the MHD algorithm. 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

Table 1
Convergence and timing results for the x-derivative of the solution in 2D

Mash size	Error	Convergence 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

Error is measured by L_2 norm of φ_x, and the relative tolerance for the iterative linear solver is 10^{-5}.

Table 2
Convergence and timing results of the gradient of the solution in 2D

Mesh size	Error	Convergence 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

Error is measured by L_2 norm of $\nabla\varphi$, and the relative tolerance for the iterative linear solver is 10^{-5}.

Table 3
Convergence results of the gradient of the solution in 3D

Mesh size	Error	Convergence 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

Error is measured by L_2 norm of $\nabla\varphi$, and the relative tolerance for the iterative linear solver is 10^{-5}.

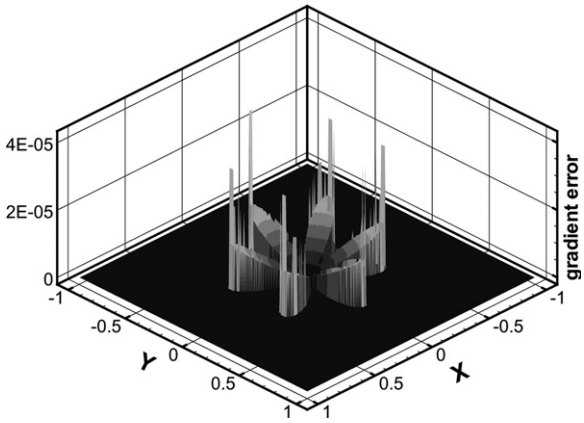

Fig. 6. Norm of the gradient error by the EB method on a 256×256 grid.

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 can 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, approaches $O(N \log N)$ [2]. We use parallel libraries of preconditioners and iterative solvers implemented in THE PETSc [1] and HYPRE [13] packages. The elliptic problem becomes dominant at some sufficiently fine grid which is $136 \times 136 \times 136$ for the numerical example illustrated in Fig. 7. To reduce the total computational cost, we often perform one elliptic step per several hyperbolic time steps.

4. Application: mercury jet in a 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

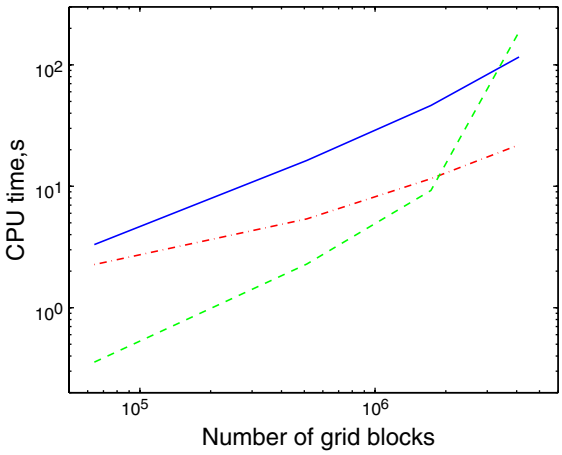

Fig. 7. CPU time spent by the interface propagation algorithm (dot-dashed line), hyperbolic solver (solid line), and the elliptic solver (dashed line). The elliptic problem starts to dominate at the grid size $136 \times 136 \times 136$. Here the elliptic problem is solved for conducting fluid occupying 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.

solutions in terms of expansion series [21] were 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 were 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.

The setup of the problem is as follows. A free mercury jet with the diameter of 0.8 cm is oriented in the z-direction. It moves along the z-axis, and 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 Fig. 8.

In the theoretical analysis [21], 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 (8), (10). Therefore a relatively short jet with edges completely within the computational domain was simulated (see Fig. 9a). The jet length was chosen to be long enough compared to the jet diameter, but sufficiently short to avoid a large 3D computational domain. The jet velocity was set to zero in the laboratory frame, and the effect of the jet motion was 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 Fig. 9b and c), and compared the jet deformation with theoretical results of [21]. The theory itself has been experimentally validated.

Past experiments have been performed [21] 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 [21], the theory is in a good agreement with experiments on the width of

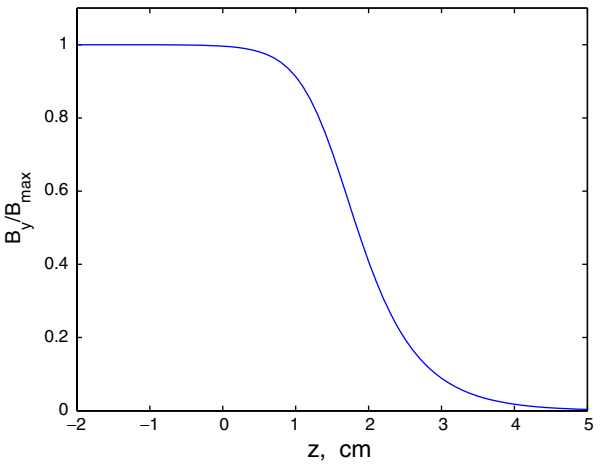

Fig. 8. Distribution of the applied magnetic field

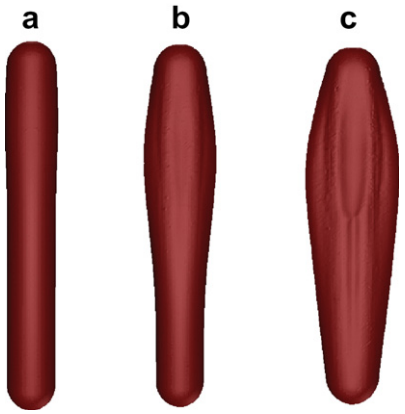

Fig. 9. Change of the mercury jet shape as it enters the magnetic field. (a) Initial time, (b) $t = 1.0$ ms, (c) $t = 1.5$ ms.

the flattened jet. Simulation results of the relative width change of the jet cross-section in magnetic fields ranging from 5.5 T to 12 T and jet velocities of 50 m/s and 60 m/s are shown in Fig. 10. The jet width is in a good agreement with theoretical predictions. As we see from Fig. 11, the jet width linearly depends on B^2/u at a fixed distance from the magnetic field center.

We have also observed a quasi-steady state of the jet shape in the sense that jet deformations remained unchanged at some fixed longitudinal displacements with respect to the magnetic field at different times. The quasi-steady state lasted until the longitudinal position of the jet cross-section approached the jet edge.

We have also performed the convergence study and found that the simulation error reduces with the mesh refinement (see Table 4). We would like to note that since theoretical calculations were 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. Since such detailed information on Oshima's experiments is not available to us we have chosen to compare simulations with the theory, and clearly observed the agreement and convergence with the mesh refinement.

The described problem is significantly important to the design of the Muon Collider/Neutrino Factory Target. The target is shown schematically in Fig. 12. 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 25–30 m/s into a 15 T magnetic field at a

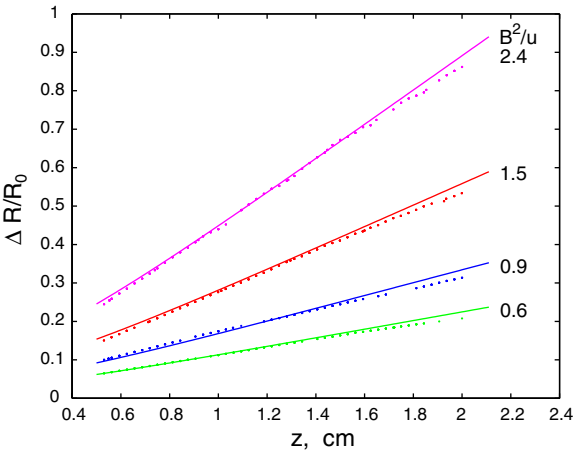

Fig. 10. Dependence of the relative change of the width of the jet cross-section on the longitudinal coordinate with respect to the magnetic field center at different values of B^2/u, [Tesla^2 · s/m]. Solid line is the theoretical result and dots are measurements of simulated jet shapes.

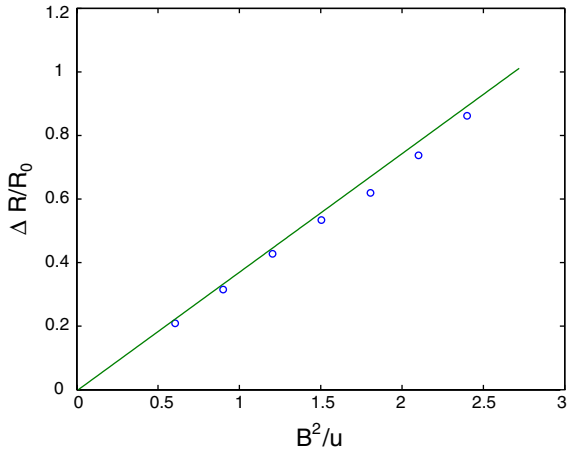

Fig. 11. Dependence of the relative change of the width of the jet cross-section on B^2/u, [Tesla^2 s/m], at the fixed distance ($z = 2$ cm) from the magnetic field center. Solid line is the theoretical result and dots are measurements of simulated jet shapes.

Table 4
Convergence of simulations to theoretical results at $B = 12$ Tesla and $u = 60$ m/s

Mesh size	$Z = 1$ cm		$Z = 2$ cm	
	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

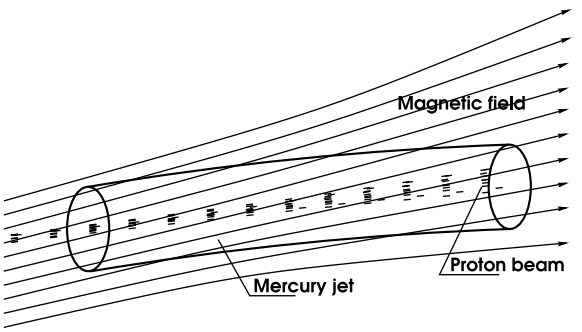

Fig. 12. Schematic of the target for the Muon Collider/Neutrino Factory.

small angle (0.033 rad) to the axis of the field. When the jet reaches the center of the magnet, it will interact with a 3 ns proton pulse depositing about 100 J/g of energy in the mercury.

We have shown that in the first phase of the jet evolution, the entrance of the mercury jet into the solenoid, the jet deformation may be significant at some design parameters, and that in turn may reduce the effective cross-section of the jet interaction with the proton pulse. The result of the study was the optimization of the jet entrance angle that does not lead to a large reduction of the interaction cross-section.

The state of the target after the interaction with a pulse of protons depositing a large amount of energy into mercury is also of major importance to the accelerator design. Simulations of the mercury jet expansion, cavitation, and surface instabilities due to the interaction with proton pulses have already been performed using our code in 2D axisymmetric approximation [25,26]. The previous work dealt with cavitation of mercury without a magnetic field. We have performed recently the direct numerical simulation of cavitation in the mercury

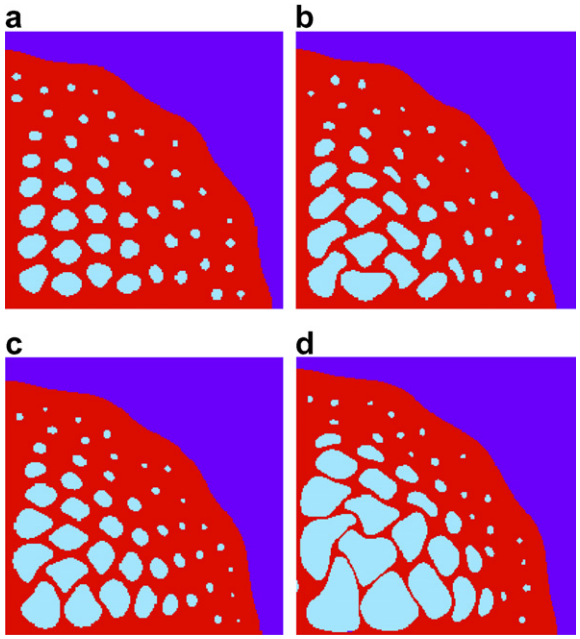

Fig. 13. Direct 2D numerical simulation of cavitation in the mercury jet after the interaction with a proton pulse depositing 50 J/g (top images) and 100 J/g (bottom images) of energy into the jet. Density distribution in the jet cross-section is shown: red is mercury, light blue is the rarefied gas in cavitation bubbles, and dark blue is the ambient gas. Time frames correspond to 20 μs (left) and 40 μs (right). (a) $E_{max} = 50$ J/g, $t = 20$ μs; (b) $E_{max} = 50$ J/g, $t = 40$ μs; (c) $E_{max} = 100$ J/g, $t = 20$ μs and (d) $E_{max} = 100$ J/g, $t = 40$ μs.

jet in the presence of a magnetic field. Omitting physics modeling and details of the numerical algorithm for the cavitation simulation presented in [27], we illustrate here only the geometric complexity of the problem. Fig. 13 shows cross-sections of the cavitating jet, and interfaces between the mercury jet, ambient gas, and cavitation bubbles. Our code is capable of propagating and resolving such complex interfaces, and solving of the Poisson problem in the domain occupied by mercury. Full 3D studies of the mercury target using the algorithm described in this paper are in progress and will be reported in a forthcoming paper.

5. Conclusions

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 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 solvers for hyperbolic problems and iterative solvers for linear systems implemented in the PETSc and HYPRE packages.

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. 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 study of a similar problem is important to the design of the future Muon Collider/Neutrino Factory target, and has led to the optimization of the jet entrance angle into the magnetic solenoid. Numerical

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 using the 2D axisymmetric version of our code. The full 3D simulations are in progress and will be presented in a forthcoming paper. Another current application of the MHD algorithm is the simulation of flows of weakly ionized plasmas associated with the fueling of magnetically confined nuclear fusion devices by the injection of frozen deuterium-tritium pellets.

Acknowledgements

We thank members of the Muon Collider/Neutrino Factory Collaboration for useful discussions and for partial support of work presented here.

This manuscript has been authored in part by Brookhaven Science Associates, LLC, under Contract No. DE-AC02-98CH10886 with the US Department of Energy. The United States Government retains, and the publisher, by accepting the article for publication, acknowledges, a world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purpose.

References

[1] S. Balay, W. Gropp, L. McInnes and B. Smith, PETSc Users Manual, ANL-95/11 – Revision 2.1.1, Argonne National Laboratory, 2001.
[2] W. Briggs, V.E. Henson, S. McCormic, A Multigrid Tutorial, 2nd ed., SIAM, publications, 2000.
[3] I.R. Chern, J. Glimm, O. McBryan, B. Plohr, S. Yaniv, Front tracking for gas dynamics, J. Comput. Phys. 62 (1986) 83–110.
[4] P. Colella, P. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, J. Comput. Phys. 54 (1984) 174–201.
[5] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, J. Comput. Phys. 87 (1990) 171–200.
[6] R. Courant, K. Friedrichs, Supersonic Flows and Shock Waves, Interscience, New York, 1948.
[7] E. George, J. Glimm, Self similarity of Rayleigh–Taylor mixing rates, Phys. Fluids 17 (2005), 054101-054101-13.
[8] E. George, J. Glimm, X.L. Li, A. Marchese, Z.L. Xu, A comparison of experimental, theoretical, and numerical simulation of Rayleigh–Taylor mixing rates, Proc. National Academy of Sci. 99 (2002) 2587–2592.
[9] J. Glimm, J. Grove, X.L. Li, K.L. Shyue, Q. Zhang, Y. Zeng, Three dimensional front tracking, SIAM J. Sci. Comput. 19 (1998) 703–727.
[10] J. Glimm, J. Grove, X.L. Li, D.C. Tan, Robust computational algorithms for dynamic interface tracking in three dimensions, SIAM J. Sci. Comput. 21 (2000) 2240–2256.
[11] J. Glimm, J. Grove, Y. Zhang, Interface tracking for axisymmetric flows, SIAM J. Sci. Comput. 24 (2002) 208–236.
[12] S.S. Harilal, M.S. Tillack, B.O Shay, C.V. Bindhu, F. Najmabadi, Confinement and dynamics of laser produced plasma expanding across a transverse magnetic field, UCSD-ENG-106, 2003.
[13] Hypre: high performance preconditioners: user's manual, http://www.llnl.gov/CASC/hypre/software.html.
[14] D.J. Jackson, Classical Electrodynamics, Wiley, New York, 1974.
[15] H. Johansen, P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, J. Comput. Phys. 147 (1998) 60–85.
[17] R. Menikoff, B.J. Plohr, The Riemann problem for fluid flow of real materials, Rev. Mod. Phys. 61 (1989) 75–130.
[18] S. Molokov, C.B. Reed, Review of free-surface MHD experiments and modeling, Technical Report ANL/TD/TM99-08, Argonne National Laboratory, 1999.
[19] N. Morley, S. Smolentsev, L. Barleon, I. Kirillov, Liquid magnetohydrodynamics - recent progress and future directions for fusion, Fusion Eng. Design 51-52 (2000) 701–713.
[20] N. Morley, S. Smolentsev, R. Munipalli, M.-J. Ni, D. Gao, M. Abdou, Progress on the modeling of liquid metal, free surface MHD flows for fusion liquid walls, Fusion Eng. Design 72 (2004) 3–34.
[21] S. Oshima, R. Yamane, Y. Moshimaru, T. Matsuoka, The shape of a liquid metal jet under a non-uniform magnetic field, JCME Int. J. 30 (1987) 437–448.
[22] Ozaki, S., Palmer, R., Zisman, M., Gallardo, J. (editors): Feasibility Study- II of a Muon-Based Neutrino Source, BNL-52623 (2001).
[23] P. Parks, R. Turnbull, Effect of transonic flow in the ablation cloud on the lifetime of a solid hydrogen pellet in plasma, Phys. Fluids 21 (1978) 1735–1741.
[24] P. Parks, Magnetic-field distortion near an ablating hydrogen pellet, Nuclear Fusion 20 (1980) 311–320.
[25] R. Samulyak, Numerical simulation of hydro- and magnetohydrodynamic processes in the Muon Collider target, Lecture Notes in Computational Science, 2331, Springer Verlag, Berlin, 2002, 391–400.
[26] R. Samulyak, Y. Prykarpatskyy, Richtmyer-Meshkov instability in liquid metal flows: influence of cavitation and magnetic fields, Mathematics and Computers in Simulations 65 (2004) 431–446.

[27] R. Samulyak, T. Lu, Y. Prykarpatskyy, J. Glimm, Z. Xu, M.N. Kim, Comparison of heterogeneous and homogenized numerical models of cavitation, Int. J. Multiscale Comput. Eng. 4 (2006) 377–389.
[28] R. Samulyak, T. Lu, P. Parks, A magnetohydrodynamic simulation of pellet ablation in the electrostatic approximation, Nucl. Fusion 47 (2007) 103–118.
[29] P. Schwartz, M. Barad, P. Colella, T. Ligocki, A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions, J. Comput. Phys. 211 (2006) 531–550.
[30] B. van Leer, Towards the ultimate conservative difference scheme: V.A second order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.