A plasma-vacuum interface tracking algorithm for magnetohydrodynamic simulations of coaxial plasma accelerators

Vivek Subramaniam and Laxminarayan L. Raja
Department of Aerospace Engineering and Engineering Mechanics,
The University of Texas, Austin

Abstract

The resistive Magneto-hydrodynamic (MHD) model describes the behavior of a strongly ionized plasma in the presence of external electric and magnetic fields. For problems involving the plasma with an open boundary to very low pressure/vacuum regions the continuum assumption is no longer valid in the entire domain of interest. For example, this is seen in a plasma accelerator where a dense plasma expands into a vacuum background. A common practice to deal with this issue is to assign small values of density and pressure to the vacuum regions and proceed to solve the continuum based MHD equations throughout the domain. We show that this approach fails to produce solutions consistent with the physics of the problem and can give rise to unacceptable artifacts such as spurious shocks. We develop a plasma-vacuum interface tracking approach to mitigate this problem. The plasma-vacuum interface is tracked in a physically consistent manner and the MHD equations are solved only in the regions that contain the plasma. The interface tracking is achieved using a face-flux formulation derived from the theoretical solution to a 1D free expansion problem. Coupled with a threshold based approach, the interface tracking is implemented for both explicit and implicit time stepping frameworks on generalized unstructured grids. Simulations of magnetized thermal plasma jets expanding into a vacuum background indicate plume profiles devoid of the unphysical shock obtained using the small background density approach. In the context of resistive MHD simulations, the interface tracking approach overcomes the numerical stiffness induced by specifying a large background resistivity in the vacuum regions, resulting in significant wall-clock time gains.

KEYWORDS: Interface-tracking; Implicit schemes; Unstructured grids; Magneto-hydrodynamics; Finite volume methods;

1. Introduction

The Magneto-hydrodynamics (MHD) governing equations are a continuum based description of electrically conducting gas flow (i.e. plasmas) subjected to external or induced electric and magnetic fields. The system of equations consists of the compressible Navier-Stokes equations that describe the flow behavior of the plasma coupled with the Maxwell's equations that describe the coupled electromagnetic fields. The electromagnetic fields affect the flow properties through the Lorentz force, which causes an acceleration of the plasma and Ohmic heating which increases the bulk plasma temperature. These electromagnetic fields are in turn generated by the plasma currents resulting in a tightly coupled non-linear equation system. The MHD equations model a wide variety of physical phenomena, e.g. in astrophysics [1], hypersonic aerodynamics [2], and in electric propulsion devices [3].

This paper is concerned with the development of a simulation approach for the study of MHD phenomena where a relatively dense plasma is sustained or accelerated into a vacuum or near-vacuum environment. An important application where such phenomena is encountered is the coaxial plasma accelerator. These devices utilize the Lorentz forcing caused by a self-induced magnetic field to accelerate the plasma to large velocities (~10's km/s). The finite volume and discontinuous Galerkin formulations have been used successfully to solve the MHD equations, discretized on a domain representative of the accelerator geometry [3,4]. The MHD equations for the plasma accelerator problem present a high degree of mathematical stiffness owing to the disparate time scales associated with the different sub-physics such as flow convection and magnetic field diffusion. This has motivated the design of numerical schemes with time-stepping algorithms that overcome stiffness imposed by the operator with the smallest characteristic time scale. Sankaran et al. [5] used a flux splitting method with artificial diffusion and an explicit fractional time-stepping scheme to simulate a Magnetoplasmadynamic (MPD) thruster device. Their emphasis was the steady state simulation of the device, which motivated design of the fractional time stepping scheme. The time-stepping algorithm partially overcame restrictions imposed by the explicit integration of a stiff diffusion operator by modifying the number of times the convective flux was evaluated. Sitaraman and Raja [4] used a semi-implicit time stepping scheme for both the convective and diffusive fluxes to simulate the unsteady plasma acceleration process in two dimensions. The semi-implicit treatment of the fluxes was achieved by analytically deriving flux Jacobians and the resultant linear system was solved using a matrix free LU-SGS method [6]. The two-dimensional Multiblock Arbitrary Coordinate Hydromagnetic (MACH2) code [7,8] has been used to perform steady state simulations of the MPD thruster by explicitly resolving the fast plasma waves arising from the convective operator and implicitly treating the stiff diffusive terms [9]. Recently, Xisto et al. [10] developed an MHD extension of the PISO method [11] where an AUSM-MHD flux was used to perform steady state simulations of an MPD thruster.

In addition to the challenge of integrating operators with disparate time scales, the specific plasma accelerator operation mode that is the focus of this study poses special challenges. Here we are interested in the so-called 'deflagration mode', that involves the acceleration of a magnetized plasma into a hard vacuum background [12]. This is a highly unsteady phenomenon characterized by a smooth rarefaction of the dense plasma into vacuum. The increase in the Knudsen number (ratio of particle mean free path to the gradient length scale of the plasma) as one moves towards the rarefied regions of the plasma, leads to a breakdown of the continuum based model [13]. Consequently, an MHD numerical scheme that is derived from the continuum based formulation predicts a plasma jet profile that is inconsistent with the physics of free expansion [4]. Therefore, in addition to dealing with the disparate time scales, our application necessitates the design of an MHD numerical scheme that incorporates a physically consistent treatment of the plasma-vacuum expansion.

One approach is to use a hybrid continuum/particle solver where the rarefied regions are treated using a particle approach while the high density regions are modelled using a continuum formulation [14]. This method can be exceedingly expensive and the physics of interest in the plasma accelerator occurs at the continuum level [12] with an accurate resolution of the rarefied region being superfluous. An alternative approach that is commonly followed [4] is to replace the vacuum with a low density, low pressure gas followed by a continuum based numerical treatment of the entire domain. It is assumed that the expansion of the high density gas/plasma into the rarefied background would mimic free expansion into vacuum [4]. This approach has been consistently used [3-10] to perform steady state simulations of plasma accelerators. Here, the values of the background density/pressure are inconsequential towards the final solution since the transients associated with the plasma-background interaction eventually convect out of the simulation domain. However, in the context of unsteady simulations of plasma jets expanding into a vacuum background, the choice of a low background density/pressure as a substitute for vacuum results in unphysical plume profiles. Specifically, this approach leads to the formation of a strong shock at the region where the plasma jet interacts with the background gas. Such a behavior of the numerical solution is inconsistent with physics of free expansion, a process characterized by a single rarefaction wave. This issue was previously identified for a pure gas dynamic expansion into vacuum by Munz et al. [15], where they proposed a gas-vacuum interface tracking algorithm for the one-dimensional inviscid Euler's equations using an explicit time integration method. This approach is termed the Vacuum Riemann Solver (VRS). In this work, we extend the treatment of Munz to the 2D resistive MHD system in the context of a fully implicit time integration scheme. We present a threshold based plasma-vacuum interface tracking algorithm for the MHD simulation of coaxial plasma accelerators. The interface tracking scheme uses the underlying structure of the computational mesh to track the expanding plasma-vacuum interface in a physically consistent manner.

The remainder of this paper is organized as follows. Section 2 presents the resistive MHD model. Section 3.1 presents an overview of the spatial discretization schemes for the convective and magnetic diffusion operators followed by section 3.2 that outlines the fully-implicit time integration scheme. Sections 4.1 and 4.2 illustrate the failure of the low background density approach in capturing free expansion in a 2D plasma accelerator and a 1D gas-dynamic shock tube setup respectively. The shock formation is identified in section 4.2 as being associated with the convergence properties of the underlying convective flux scheme. The threshold based plasma-vacuum interface tracking algorithm is proposed in section 4.3 and is compared against a theoretical solution to a 1D free expansion problem. A 2D interface-tracking algorithm for generalized unstructured grids is presented in section 4.4 in the context of an explicit time stepping scheme for the Euler's equations. This approach is used to perform simulations of unmagnetized thermal plasma jets expanding into a vacuum background to illustrate the absence of a shock-front as compared to the low background density approach. Section 4.5 presents an extension of the interface tracking formulation in the context of an implicit time stepping scheme. Section 4.6 extends the fully-implicit interface tracking framework towards the ideal MHD and resistive MHD systems. Ideal MHD simulations of magnetized plasma jets are shown to exhibit a front devoid of the shock and resistive MHD simulations indicate considerable gains in wall-clock times as compared to the background density approach. Finally, section 5 closes with conclusions.

2. Resistive MHD equations

The MHD model is a 'single-fluid' description of plasma with a single temperature under the local thermodynamic equilibrium (LTE) approximation used to define plasma composition and its thermodynamic properties [16]. Furthermore, the plasma is assumed to have a high enough density that the quasi-neutral approximation is valid throughout the domain of interest [17]. The resistive MHD equations comprise two parts. The hydrodynamic part which consists of the compressible Navier-Stokes equations with magnetic Lorentz force and Joule heating source terms [18]. These are coupled with the Maxwell's equations; namely the Faraday's and Ampere's laws. The entire system is closed by an equation of state and the Ohm's law. A detailed derivation can be found in [19]. Of interest here is the development of a consistent MHD model to study plasma acceleration in coaxial plasma accelerators. Shown in Fig. 1 is the basic operating principle of the accelerator. A radial current induces an azimuthal magnetic field generating an axial Lorentz force that accelerates the plasma to large axial velocities.

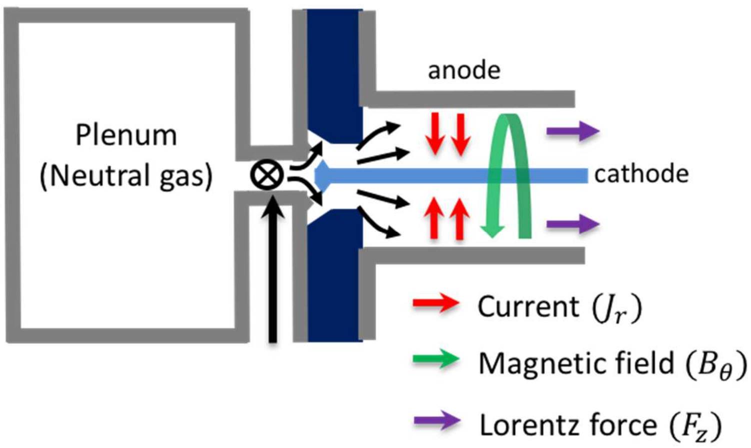

Figure 1. Schematic illustrating the working principle of the coaxial plasma accelerator. A radial current induces an azimuthal magnetic field generating an axial Lorentz force.

The coaxial geometry of the device motivates the use of a cylindrical polar coordinate system. The plasma acceleration is assumed to be axisymmetric thereby transforming the 3D model into a 2D resistive MHD system, the focus of this paper. The primary challenge associated with designing a numerical scheme for the MHD system is the satisfaction of the zero-divergence constraint for the magnetic induction vector. Several hyperbolic divergence cleaning techniques have been proposed to ensure that the magnetic induction vector is divergence free [22]. However, in the specific case of the coaxial plasma accelerator, the axisymmetric cylindrical polar coordinate system leads to the divergence-free criterion being trivially satisfied. In the plasma accelerator simulations, the magnetic induction vector is convected/diffused into the simulation domain from the boundaries [4,5]. A net radial current (Fig. (1)) is simulated by imposing a purely azimuthal magnetic field at the inlet boundary of the device. An examination of the time-evolution equation for the magnetic induction vector indicates that a zero-magnetic-field initial condition coupled with a purely azimuthal field boundary condition ensures that the magnetic induction vector is oriented in the azimuthal direction at all times. The contribution of such an initial field topology, $\vec{B} = (0, B_\theta(r,z), 0)$ (here θ, r, z denote the azimuthal, radial and axial directions) towards the production of radial and axial magnetic field components are essentially zero. This ensures that the magnetic induction vector will always be given by $\vec{B}(t,r,z) = (0, B_\theta(t,r,z), 0)$, where t denotes time. Such a profile will always satisfy the divergence-free criterion. This enables a simplification of the equation system since the radial and axial magnetic field components need not be solved for, thereby reducing the number of unknowns in the axisymmetric resistive MHD equations from 7 to 5.

The resistive MHD equations in conservative form are written as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \vec{V}\right) = 0 \tag{1}$$

$$\frac{\partial}{\partial t}\left(\rho \vec{V}\right) + \nabla \cdot \left(\rho \vec{V} \vec{V}\right) - \nabla\left(P + \frac{B_\theta^2}{2\mu_0}\right) = 0 \tag{2}$$

$$\frac{\partial}{\partial t}(\vec{B}) + \nabla \cdot \left(\vec{V}\vec{B} - \vec{V}\vec{B}\right) = -\nabla \times \left(\eta \frac{\nabla \times \vec{B}}{\mu_0}\right) \tag{3}$$

$$\frac{\partial}{\partial t}\left(\rho e_t + \frac{\vec{B} \cdot \vec{B}}{2\mu_0}\right) + \nabla \cdot \left(\left(\rho e_t + \frac{B_\theta^2}{\mu_0} + P\right)\vec{V} - \frac{\vec{V} \cdot \vec{B}}{\mu_0}\vec{B}\right) = -\frac{1}{\mu_0}\nabla \cdot \left(\eta\left(\nabla \times \vec{B}\right) \times \vec{B}\right) \tag{4}$$

$$\nabla . \vec{B} = 0 \tag{5}$$

The large Reynold's number and fast plasma transient allow for the viscous and conductive heat transfer effects to be ignored. ρ, P, $\vec{V}$ and $\vec{B}$ represent the plasma density, pressure, velocity and magnetic induction vector, respectively. Additionally, η represents the plasma resistivity and e_t the specific total energy that is composed of the specific internal energy e and the specific kinetic energy $\frac{|\vec{V}\cdot\vec{V}|}{2}$, that is $e_t = e + \frac{|\vec{V}\cdot\vec{V}|}{2}$. The plasma resistivity η is a function of the plasma pressure and temperature. A tabulation based on the Spitzer resistivity model [20] is used to calculate the plasma resistivity as a function of the temperature and pressure. The Spitzer resistivity [36] model is given by

$$\eta = \frac{(5e-5)ln(\Lambda)}{T_e^{\frac{3}{2}}}. \tag{6}$$

The tabulation uses the hydrogen resistivity data from [21] to back out the parameter Λ. The real gas formulation with a variable gas constant is used to define the equation of state [18]. This modified equation of state also gives the relationship between the internal energy density and the temperature. These are given by

$$e(T) = \frac{R(T)T}{\gamma - 1} \tag{7}$$

$$P = \rho R(T)T \tag{8}$$

The specific internal energy data from [21] is used to find the gas constant as a function of temperature. Eq. (1) - (5) with the closure relations given by Eq. (6) – (8) provide a consistent description of the dense thermal plasma in coaxial plasma accelerators.

3. Numerical formulation

3.1 Spatial discretization

In order to perform the spatial discretization, the MHD equation system is written in the following conservative form:

$$\frac{\partial U}{\partial t} + \nabla \cdot F^c = \nabla \cdot F^r + S \, . \tag{9}$$

Here, U, F^c, F^r and S represent the vectors of conservative variables, convective (hyperbolic) flux, magnetic resistive flux and source term respectively. Furthermore, the equation system Eqs. (1) - (4) presented in the previous section is non-dimensionalized using a length scale l_s, a pressure scale p_s and a temperature scale t_s. The scaling eliminates the permeability constant that appears in the resistive diffusion term thereby making the equations more suitable for subsequent numerical treatment. The vectors in Eq. (9) are written out in a cylindrical polar coordinate system as follows:

$$U = \begin{bmatrix} \rho \\ \rho V_r \\ \rho V_z \\ B_\theta \\ \rho e_t + \frac{B_\theta^2}{2} \end{bmatrix}, F^c = \begin{bmatrix} \rho(V_r\hat{r} + V_z\hat{z}\,) \\ \rho V_r(V_r\hat{r} + V_z\hat{z}) - \left(P + \frac{B_\theta^2}{2}\right)\hat{r} \\ \rho V_z(V_r\hat{r} + V_z\hat{z}) - \left(P + \frac{B_\theta^2}{2}\right)\hat{z} \\ B_\theta(V_r\hat{r} + V_z\hat{z}) \\ (\rho e_t + B_\theta^2 + P)(V_r\hat{r} + V_z\hat{z}) \end{bmatrix}, F^r = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \eta \nabla B_\theta \\ \eta B_\theta \nabla B_\theta \end{bmatrix} \tag{10}$$

Here, V_r is the radial velocity, V_z the axial velocity, $\hat{r}$ and $\hat{z}$ denote the unit vectors in the radial and axial coordinate directions respectively. The source term is given by

$$S = \begin{bmatrix} -\frac{\rho V_r}{r} \\ -\frac{\rho V_r^2}{r} \\ -\frac{\rho V_r V_z}{r} \\ -\frac{V_r B_\theta}{r} \\ -\frac{e_t + P + B_\theta^2}{r} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{r}\frac{\partial}{\partial r}(\eta B_\theta) - \frac{\eta B_\theta}{r^2} \\ \frac{3}{r}\eta B_\theta \frac{\partial B_\theta}{\partial r} + \frac{1}{r}\frac{\partial \eta}{\partial r} B_\theta^2 \end{bmatrix}. \tag{11}$$

The non-dimensionalized equation system (Eq. (9)) is spatially discretized using a cell-centered finite volume formulation on generalized unstructured grids. The resulting ordinary differential equation (ODE) system obtained after the discretization is performed is as follows:

$$\frac{\partial U_i}{\partial t}\Omega_i + \sum_{n_{faces}} F_f^c \cdot \hat{n}_f s_f = \sum_{n_{faces}} F_f^r \cdot \hat{n}_f s_f + S_i\Omega_i \tag{12}$$

Here, U_i denotes the vector of conservative variables associated with the finite volume cell indexed using the subscript “i”, that has a volume Ω_i. Within the finite-volume framework it is assumed that the vector of conservative variables U_i and source terms S_i are piecewise constant over the cell “i”. The convective and resistive diffusion fluxes F_f^c and F_f^r are computed at a face f using the data from the two cells that lie adjacent to the face. Since face f belongs to the cell “i”, one of these adjacent cells will be cell “i”. The other adjacent cell, indexed using the subscript “j”, is a cell that shares face f with cell “i”. A schematic of the finite volume flux computation is shown in Fig. 2.

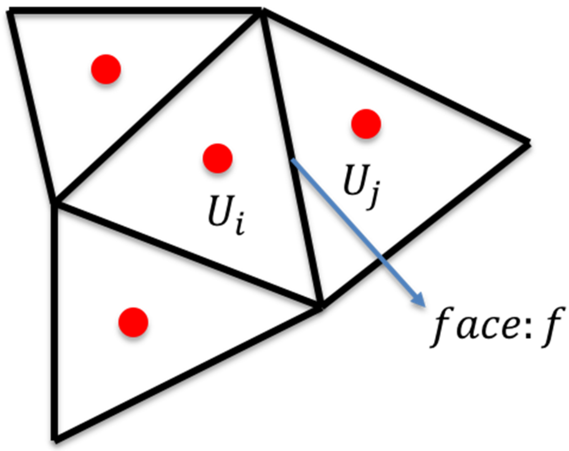

Figure 2. Schematic of the finite volume framework on unstructured grids showing cells “i” and “j” sharing a face “f” between them.

The convective flux is computed using the Harten-Lax-Van Leer (HLL) scheme [23]. The HLL approximate Riemann solver uses the integral form of the conservation law to define a single state between the fastest waves perturbing the initial data. This is achieved by integrating the conservation law between the fastest left and right running waves and spatially averaging the integral to obtain a single intermediate state [24]. The face flux is then obtained by applying the integral conservation law to a control volume having one of its boundaries at the face center. The face-flux is given by

$$F_f^c = \begin{cases} F_i^c & if\ S_l \geq 0 \\ F_j^c & if\ S_r \leq 0 \\ \dfrac{(S_r F_i^c - S_l F_j^c + S_r S_l (U_j - U_i))}{S_r - S_l} & if\ S_l < 0 < S_r \end{cases} \tag{13}$$

Here, F_i^c and F_j^c denote the cell-centered convective fluxes that are calculated using the data from the cells "i" and "j" respectively. S_l, S_r represent estimates for the fastest left-running and right-running waves respectively. In the case of the hyperbolic MHD system these wavespeeds approximate the left and right moving fast-magnetosonic waves. They are given by $S_l = \tilde{v}^n - \tilde{c}^f$ and $S_r = \tilde{v}^n + \tilde{c}^f$, where c^f is the fast-magnetosonic velocity given by $c^f = \sqrt{\frac{\gamma P}{\rho} + \frac{B_\theta^2}{\rho}}$ and v^n denotes the velocity component normal to the face f. The wave speed estimates are derived as Roe-averaged estimates for the magnetosonic speed and flow velocity using the left and right state data of the face-centered Riemann problem. These are the wave-speed estimates as proposed by Einfeldt [13] and Davis [25] and are recommended for use in practical applications by Toro [24] as

$$\tilde{v}^n = \frac{\sqrt{\rho_i} v_i^n + \sqrt{\rho_j} v_j^n}{\sqrt{\rho_i} + \sqrt{\rho_j}} \tag{14}$$

$$\tilde{c}_f = \frac{\sqrt{\rho_i} c_i^f + \sqrt{\rho_j} c_j^f}{\sqrt{\rho_i} + \sqrt{\rho_j}} \tag{15}$$

Here, v_i^n, v_j^n , c_i^f, c_j^f denote the face normal velocities and magnetosonic speeds obtained using the data at the cells "i" and "j".

On unstructured grids, reconstruction of the diffusive flux involves a form of cell-centered gradient reconstruction procedure such as the least-squares reconstruction, followed by an averaging of the adjacent cell-centered gradients to obtain the face gradient. However, it is known that this procedure could lead to checker-boarding effects that pollute the final solution [26]. This issue was addressed by Haselbacher and Blazek [26] who proposed the addition of an extra term while performing the face reconstruction, to combat any possible checker-boarding. This scheme is notionally given by:

$$
\begin{gathered}
\nabla\phi_{ave} = \frac{\nabla\phi_i + \nabla\phi_j}{2} \\
\nabla\phi_f = \nabla\phi_{ave} - (\nabla\phi_{ave}.\hat{d}_{ij})\hat{d}_{ij} + \frac{\phi_j - \phi_i}{d_{ij}}\hat{d}_{ij}
\end{gathered}
\tag{16}
$$

Here, ϕ represents a general variable that is subjected to gradient reconstruction, $\nabla\phi_{ave}$ denotes the gradient obtained by averaging the cell-centered reconstructed gradients on the cells "i" and "j" that lie adjacent to the face. $\hat{d}_{ij}$ represents the unit vector from the centroid of cell "i" to cell "j", and d_{ij} represents the distance between the respective cell centroids. A schematic illustrating this reconstruction scheme is shown in Fig. 3.

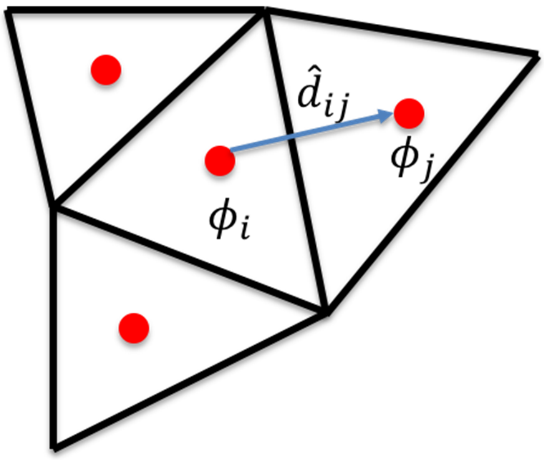

Figure 3. Schematic illustrating the elements of the Hasselbacher diffusion scheme.

The Haselbacher scheme is used to obtain the resistive diffusion flux which is given as follows:

$$
F_f^r = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \left(\frac{\eta_i + \eta_j}{2}\right)\nabla B_f \\ \left(\frac{\eta_i + \eta_j}{2}\right)\left(\frac{B_{\theta i} + B_{\theta j}}{2}\right)\nabla B_f \end{bmatrix}
\tag{17}
$$

3.2 Temporal discretization

In order to setup the temporal discretization, the system ODE is formulated. This is the ODE which governs the time evolution of conservative variables in all the cells associated with the finite-volume discretization. The spatially discretized equations are written in residual form by summing the fluxes over all the faces and collecting them into corresponding residual operators. Since the fluxes are obtained using data from the two cells adjacent to a face, the residual operators are a function of the cell data and the data of the cells which neighbor (share a common face) the cell "i". Symbolically this is given by

$$\frac{\partial U_i}{\partial t}\Omega_i + R_i^c(U_i, \boldsymbol{U}_{nbs}) = R_i^r(U_i, \boldsymbol{U}_{nbs}) + R_i^s(U_i) \tag{18}$$

Here, $R_i^c = \sum_{n_{faces}} F_f^c \cdot \hat{n}_f s_f$ is the convective residual, $R_i^r = \sum_{n_{faces}} F_f^r \cdot \hat{n}_f s_f$ is the magnetic diffusion residual, and $R_i^s = S_i \Omega_i$ is the source residual. $\boldsymbol{U}_{nbs}$ are the conservative variables that belong to the cells neighboring the cell "i". The ODE system is obtained by collecting the residual form ODE's and combining them into one single system given by

$$\frac{\partial \boldsymbol{U}}{\partial t} + R^c(\boldsymbol{U}) = R^r(\boldsymbol{U}) + R^s(\boldsymbol{U}), \tag{19}$$

where

$$\boldsymbol{U} = \begin{bmatrix} U_1\Omega_1 \\ \vdots \\ U_N\Omega_N \end{bmatrix}, \quad \boldsymbol{R}^c = \begin{bmatrix} R_1^c \\ \vdots \\ R_N^c \end{bmatrix}, \quad \boldsymbol{R_r} = \begin{bmatrix} R_1^r \\ \vdots \\ R_N^r \end{bmatrix}, \quad \boldsymbol{R_s} = \begin{bmatrix} R_1^s \\ \vdots \\ R_N^s \end{bmatrix}. \tag{20}$$

Here, N denotes the total number of cells in the domain. A large disparity in the characteristic timescales associated with the various residual operators results in a stiff ODE system. For the dense thermal plasma typically observed in the coaxial plasma accelerator experiment, convection takes place at a characteristic timescale determined by the fast magnetosonic speed that is $\sim 10^{-9}$s whereas the resistive diffusion occurs at a timescale $\sim 10^{-15}$s. The disparity in timescales renders explicit methods (constrained to follow diffusion timescale) infeasible due to the large number of timesteps $\sim 10^9$ required to resolve a microsecond acceleration transient. This leaves the implicit time integration scheme as the method of choice for performing the time integration. The implicit scheme is written as follows

$$\frac{\boldsymbol{U}^{n+1}-\boldsymbol{U}^{n}}{\Delta t}+R^{c}(\boldsymbol{U}^{n+1})=R^{r}(\boldsymbol{U}^{n+1})+\ R^{s}(\boldsymbol{U}^{n+1}). \tag{21}$$

The scheme produces an algebraic system of equations at every time step that can be symbolically represented as

$$\begin{gathered} G(\boldsymbol{U})=0 \\ G(\boldsymbol{U})=\frac{\boldsymbol{U}-\boldsymbol{U}^{n}}{\Delta t}+R(\boldsymbol{U}) \quad ;R(\boldsymbol{U})=R^{c}(\boldsymbol{U})-R^{r}(\boldsymbol{U})-R^{s}(\boldsymbol{U}). \end{gathered} \tag{22}$$

The Newton method is used to solve the system of non-linear equations and is given by

$$G(\boldsymbol{U}^{k})+\frac{\partial G(\boldsymbol{U}^{k})}{\partial \boldsymbol{U}^{k}}(\boldsymbol{U}^{k+1}-\boldsymbol{U}^{k})=0. \tag{23}$$

The vector of conservative variables at the new iterate $\boldsymbol{U}^{k+1}$ is obtained by a matrix inversion, that involves the solution of a large system of sparse linear equations. Expanding out the function G as given by Eq. (22) yields the functional form of the linear system that is solved at every iterate:

$$\boldsymbol{U}^{k+1}=\boldsymbol{U}^{k}-\left[\frac{I}{\Delta t}+\frac{\partial R_c}{\partial \boldsymbol{U}}-\frac{\partial R_r}{\partial \boldsymbol{U}}-\frac{\partial R_s}{\partial \boldsymbol{U}}\right]^{-1}\left(\frac{\boldsymbol{U}^{k}-\boldsymbol{U}^{n}}{\Delta t}+R(\boldsymbol{U}^{k})\right). \tag{24}$$

where

$$\frac{\partial G(\boldsymbol{U}^{k})}{\partial \boldsymbol{U}^{k}}=\frac{I}{\Delta t}+\frac{\partial R_c}{\partial \boldsymbol{U}}-\frac{\partial R_r}{\partial \boldsymbol{U}}-\frac{\partial R_s}{\partial \boldsymbol{U}}$$

In Eq. (24) the partial derivatives of the residual operators with respect to the vector of conservative variables are referred to as the residual Jacobians. The residual Jacobians are analytically derived by differentiating the underlying fluxes (Eq. (12)) that constitute each residual operator. The convergence of the Newton iteration is defined by a relative tolerance criteria, the system being deemed as converged if the l^2 norm of the function G reduces by four orders of magnitude. This relative tolerance criterion employs the fact that the solution iterate $\boldsymbol{U}^{k+1}$ converges to $\boldsymbol{U}^{n+1}$ through the course of the newton iterations (as k increases). The sparse linear system inversion at each Newton step is carried out using the Generalized Minimum Residual (GMRES) algorithm using the block-Jacobi and ILU preconditioners implemented in the Portable and Extensible Toolkit for Scientific Computing (PETSc) library [27].

4. Plasma-vacuum interface tracking algorithm

4.1 Motivation

The algorithm described in this paper is motivated with the goal of performing physically consistent and scalable simulations of electromagnetic acceleration devices that operate in rarefied/vacuum backgrounds. A nominal simulation set up indicating the initial and boundary conditions prescribed to simulate a coaxial plasma accelerator is shown in Fig. 4.

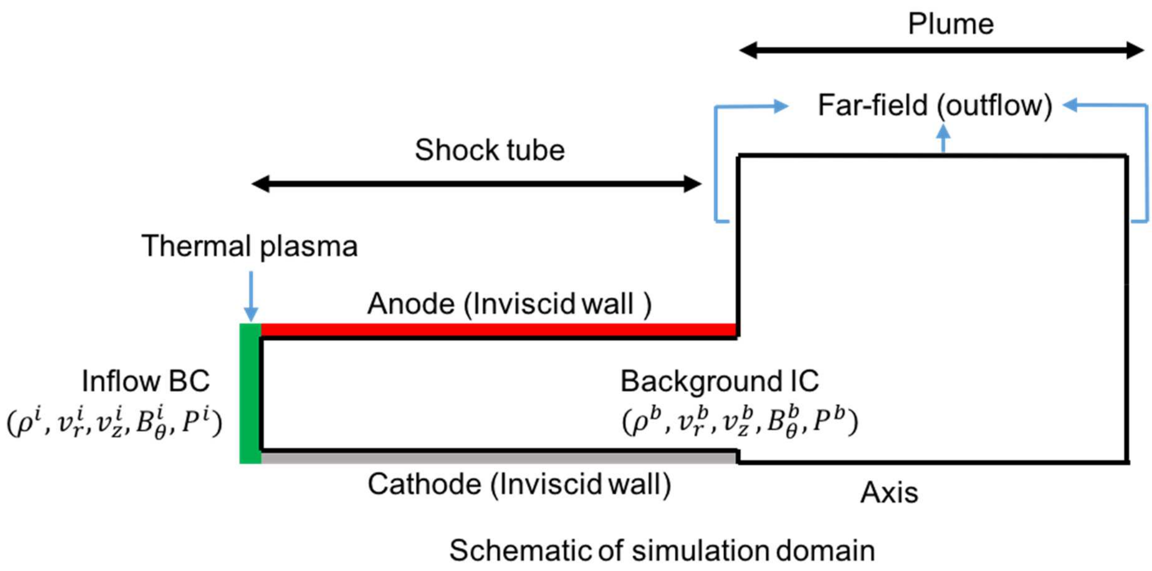

Figure 4. Schematic of simulation domain. The simulated domain (black boundary) includes all regions starting from the inlet where the thermal plasma is formed and includes the subsequent

The simulation domain consists of the coaxial electrode geometry and an additional plume region to capture the jet emanating from the accelerator. In the experiments [28] the device operates in the deflagration mode which is achieved practically by completely evacuating the interelectrode volume before introducing the plasma from the inflow boundary. Earlier attempts to simulate this mode of operation involve specifying a low density $\rho^b \sim 10^{-7} kg/m^3$ and low pressure $P^b \sim 1Pa$ initial condition ($v_r^b = 0, v_z^b = 0, B_\theta^b = 0$) to mimic the vacuum background [4]. It was assumed that the expansion of the high density $\rho^i \sim 10^{-3} kg/m^3$, high pressure

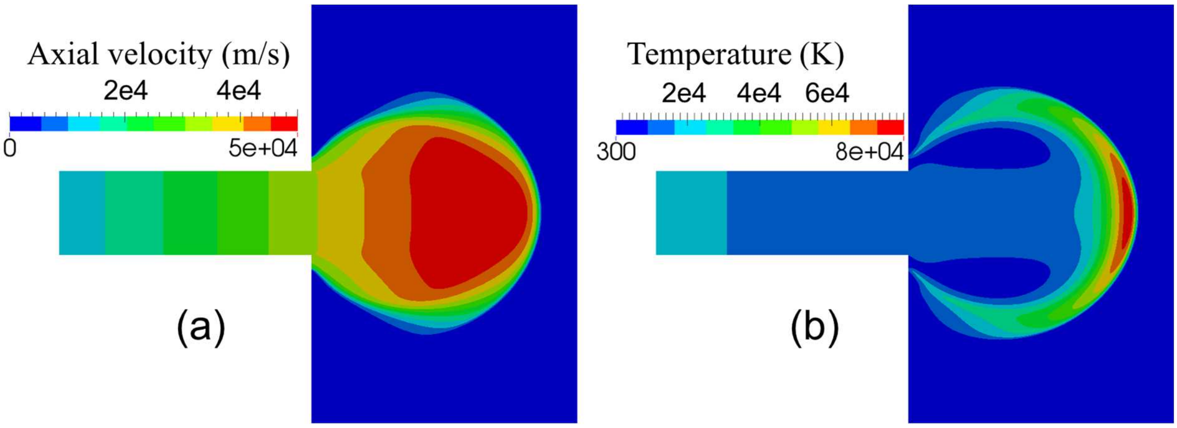

Figure 5. Coaxial plasma accelerator simulations of the deflagration mode 5μs after discharge initiation indicating abrupt shock at the jet-tip (a) Axial velocity (b) Temperature

$P^i \sim 10^5 Pa$ thermal plasma into the low-density background would approximate free expansion into vacuum [15].

However, contrary to the intuition behind this approach, the low background values generate plume profiles that are inconsistent with physics of free expansion into vacuum. Free expansion physics dictates that a plume expanding into vacuum exhibits a smooth transition in density, pressure, and temperature as one moves from the core of the jet towards its periphery [29]. On the other hand, the simulations depict temperature and velocity profiles (Fig. 5) that indicate the presence of an abrupt shock in the region where the plasma jet interacts with the background. The simulation shown in Fig. 5 was performed using quadrilateral mesh elements with a uniform mesh element size of 0.00625 cm in the radial direction and 0.015 cm in the axial directions. The inflow conditions employ a supersonic inlet boundary condition at a pressure and density of $P^i = 10^5 Pa$ and $\rho^i = 10^{-3} kg/m^3$. The initial conditions involve specifying a background density and pressure of $\rho^b = 10^{-7} kg/m^3$ and $P^b = 1Pa$. As shown in Fig. 1 an inviscid wall boundary condition is imposed on the electrodes, axisymmetric boundary condition on the axis and a supersonic outflow boundary condition on the far-field boundaries. A CFL value of 0.9 was used to determine the timestep based on the maximum convective wavespeed.

Prior efforts to simulate these devices do not encounter this issue because the accelerator operated in a background pre-filled with a low-density gas [5,7,8]. This mode of operation, referred to as the 'detonation mode', is characterized by the formation of a shock that travels down the length of the accelerator [30]. The specification the pre-filled gas density along with the thermal plasma inlet boundary condition (Fig. 4) sets up the Initial Boundary Value Problem (IBVP) to capture this moving shock.

As the background gas is evacuated, the mode of operation transitions from the detonation to the deflagration mode [30,31]. The mode transition produces a significant difference in the acceleration physics within the accelerator. As opposed to the detonation mode that exhibited a moving shock, the deflagration mode is characterized by a free expansion into vacuum. However, efforts to simulate this mode using the same IBVP setup as the detonation mode but with a significantly lower background density to mimic vacuum are unable to capture this mode transition. As seen in Fig. 5, the moving shock associated with the detonation mode persists in these low background density deflagration simulations. Capturing the acceleration physics associated with the deflagration mode requires a reformulation of the MHD numerical scheme. In this regard, the objective of this paper is to present a plasma-vacuum interface tracking algorithm, based completely on the continuum formulation, that resolves the expanding plasma-vacuum front in a physically consistent manner.

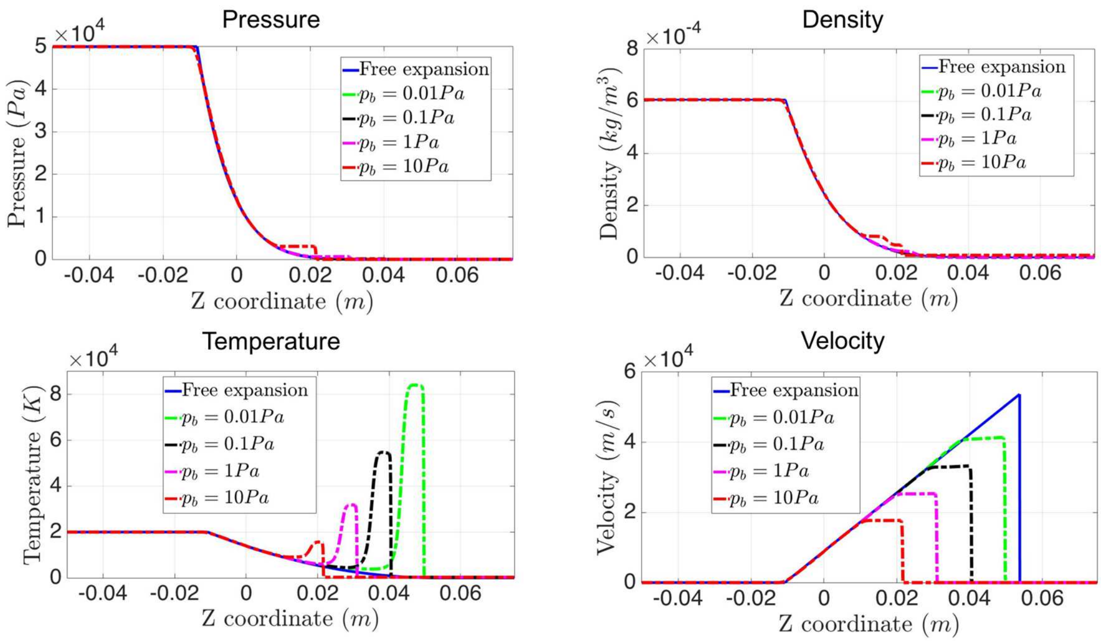

Figure 6: Comparison of the theoretical free expansion solution with the low-pressure background simulations at $t = 1\ \mu s$ for 4 different test cases with progressively lower background pressures.

4.2 Failure of HLL solver in low density backgrounds

To isolate the reason behind the shock production, we consider numerical tests on a simplified 1D shock-tube problem. The shock-tube is designed to capture purely gas-dynamic effects. The objective is to demonstrate the failure of the low background density approach in approximating free expansion into vacuum. To this end, a shock-tube problem is simulated with a fixed left state representing the driver gas and a low background density driven/right state approximating the vacuum background. The driver gas in this test consists of hydrogen at a pressure of 50kPa and a temperature of 20000K. These are nominal inlet boundary conditions used to simulate plasma accelerators [4]. The pressure and temperature values produce a driver gas density of $6.06 \times 10^{-4}\ \mathrm{kg/m^3}$. Four different values of background densities/pressures are considered as approximations to the vacuum state on the right. The background temperature is fixed at 300 K [4] and the background density is varied from 10Pa to 10^{-2} Pa. This gives rise to four different sets of background pressure/density states with the following values:

$(p_b, \rho_b) = (10\ \mathrm{Pa}, 8.08 \times 10^{-6}\ \mathrm{kg/m^3})$, $(1\ \mathrm{Pa}, 8.08 \times 10^{-7}\ \mathrm{kg/m^3})$, $(0.1\ \mathrm{Pa}, 8.08 \times 10^{-8}\ \mathrm{kg/m^3})$, $(0.01\ \mathrm{Pa}, 8.08 \times 10^{-9}\ \mathrm{kg/m^3})$. The theoretical solution [15] consists of a single rarefaction/expansion fan that processes the driver gas, bringing about a smooth transition of pressure, density and temperature to the zero background values. A comparison between the numerical test cases and the theoretical free expansion solution is shown in Fig. 6.

Progressively lowering the background pressure/density produces density/pressure profiles that show good agreement with the theoretical solution. The temperature and velocity profiles however, indicate the presence of a shock that is contrary to the smooth transition predicted by the free expansion solution. The observed behavior of the numerical solution occurs due to the convergence properties of the underlying flux scheme. The HLL scheme constructs the face flux at the interface between two cells, by averaging the theoretical solution to the face-centered Riemann problem between the fastest left and right-running waves [23,24]. Hence, in the limit of the cell sizes going to zero, the HLL solution converges to the theoretical solution [24]. Each of computational test cases presented here have an equivalent theoretical solution. The theoretical solution to a shock tube problem with a high density/pressure gas as the left state and low density/pressure gas as the right will always give rise to a solution that has a rarefaction wave, a contact discontinuity and a shock as one moves from left to right in the domain. This structure is observed as long as the right state has a finite density. Hence, if the initial value problem to capture free expansion is set up using a low background density vacuum approximation, the HLL scheme will predict a solution that by construction contains a shock. The shock strength, estimated by the temperature rise, increases with decreasing background densities. The behavior of the numerical scheme, while consistent with the shock-containing theoretical solution, exhibits a large deviation from the desired free expansion theoretical solution. In this context, the objective is to develop a flux scheme and an algorithmic framework that predicts solutions that resemble a free expansion as opposed to the unphysical shock obtained using the current implementation of the HLL scheme.

4.3 Gas-Vacuum Interface tracking algorithm

The numerical problem illustrated in section 4.2 was identified by Munz [15] who proposed a numerical approach for the 1D gas dynamic shocktube problem with vacuum as one of the initial states. This approach does not employ a low pressure/low density background to approximate the vacuum state. Instead, a zero pressure/density background is used and the face flux reconstruction at the gas-vacuum interface is carried out by approximately solving a Vacuum Riemann problem (VRP) at the interface. The main feature of the algorithm proposed is a physically consistent tracking mechanism for the vacuum-gas interface, which is a point in 1D. The point divides the Eulerian grid into two types of cells, those containing gas (left of the point) and those containing vacuum (right of the point). During the course of the simulation the interface point moves in the fixed grid filling up the cells initially containing vacuum with gas, a process representative of free expansion of gas into vacuum. The conservative variables of the newly filled gas cell are updated using a gas-vacuum face flux [15].

While the above approach works well in 1D, it is not suitable for multi-dimensional MHD simulations of gas-vacuum expansion. The point tracking method becomes cumbersome to implement in 2D and 3D. The tracked points are always associated with particular cells and faces. Maintaining this association in 2D and 3D for an arbitrarily shaped time evolving interface becomes complex and time consuming. For instance, the simple problem of simulating the 2D expansion of a kernel of gas into vacuum, requires the generation of a series of new points and tracking their associated faces and cells. At the same time there exists the situation of two points moving into the same cell which requires a destruction of one of the points. Tracking an arbitrarily shaped interface within this framework of dynamically creating and destroying tracking points becomes quite inconvenient from an implementation perspective. This issue is resolved in our algorithm by realizing that the interface doesn't need to be tracked to a level below grid resolution. The structure of the grid itself can be used to track the time evolving gas-vacuum interface. The algorithm divides the computational domain into three types of cells, those which contain gas referred to as gas cells, cells that contain vacuum but share a face with a gas cell referred to as interface cells and non-interface cells that contain vacuum referred to as vacuum cells. The gas-vacuum interface is resolved by the faces in the mesh as the collection of all those faces which have one adjacent cell as a gas cell and another as an interface cell. This gives rise to three types of faces, ones that have gas cells on both sides referred to as gas faces, ones which have a gas cell on one side and interface cell on another referred to as gas-interface faces and faces that have vacuum cells on both sides referred to as vacuum faces. The proposed algorithm loops though the gas faces computing the face flux using the standard HLL scheme as outlined in section 3.1. This computation is valid since both the cells adjacent to the gas face have a continuum gas/plasma and give wave speeds which have a physical meaning. Vacuum faces are never looped over. Lastly, the gas-interface faces have their face flux computed by approximately solving a vacuum Riemann problem [15], centered at the face, with the gas cell data as the left data.

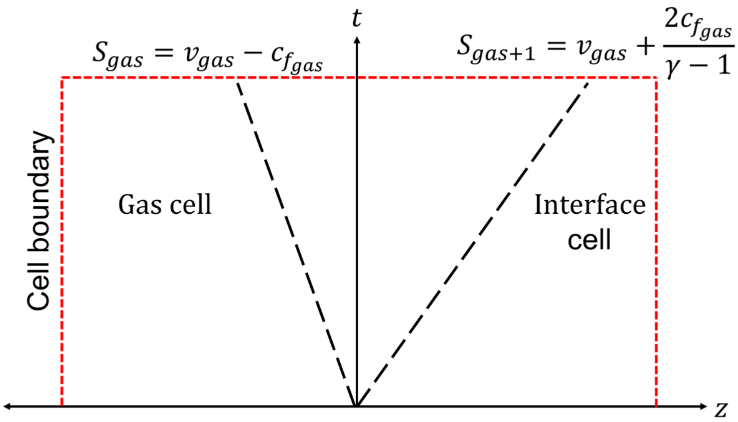

Figure 7: Approximate solution to the gas-interface centered Vacuum Riemann Problem with a gas cell to the left and an interface cell to the right, the cell boundary is indicated by red dashed lines and the fastest wave speeds by black dashed lines.

The construction of the approximate solution is similar to the HLL scheme. However, since the interface cell contains conservative and flux variables that are identically zero, these vectors do not appear in the flux reconstruction. An illustration of a gas-interface face and the wave speeds that constitute the approximate Riemann solution are shown in Fig. 7. The fastest right-moving wave speed is obtained as the tip speed of the rarefaction that processes the gas as it expands into the interface cell [15]. Using the wave speed estimates shown in Fig. 7, the flux reconstruction at the gas-interface face given by:

$$F_{gi}^{c} = \begin{cases} F_{gas}^{c} & if\ S_l \geq 0 \\ \dfrac{(S_r F_{gas}^{c} - S_r S_l U_{gas})}{S_r - S_l} & if\ S_l < 0 \end{cases} \tag{25}$$

Here, $S_l = v_n^{gas} - c_f^{gas}$ and $S_r = v_n^{gas} + \frac{2c_f^{gas}}{\gamma - 1}$. The subscript *"$gi$"* is used on the face-flux expression in Eq. (25) to refer to the flux computed on the gas-interface face. It is useful to point out here that the derivation of the vacuum Riemann flux as given in Eq. (25), with the wave speed estimates used above, is only valid when the magnetic field is oriented in a purely azimuthal direction. Under this condition, the solution has the same wave structure as a 1D unmagnetized free expansion problem with the acoustic velocities replaced by the magnetosonic velocities. For the case where the magnetic field has poloidal components, the analytical vacuum Riemann solution would exhibit Alfven waves. In this case, the wavespeeds would be different from the estimates obtained for the purely azimuthal case. Hence, for the poloidal field orientation the analytical vacuum Riemann problem would have to be solved to obtain the fastest left and right moving waves that would then be used to construct the approximate vacuum Riemann flux. Since this is out of the scope of our application, the derivation of an approximate flux with poloidal fields has been temporarily omitted. The flux scheme obtained in Eq. (25) would remain the same for the HLLC scheme as well. Since the analytical vacuum Riemann solution is constructed by merging the tip of the rarefaction with the contact surface, the fastest right moving wave and the contact wave coalesce into a single wave. Hence the HLLC scheme, that explicitly introduces a separate wavespeed estimate for the contact surface, would yield the same result as the HLL scheme for the approximate vacuum Riemann flux. Using the formulation in Eq. (25), the time update is illustrated in the context of an explicit time-stepping scheme. Here, the subscript f on the convective flux as used in Eq. (13) has been dropped and replaced by the type of face (gas/gas-interface) on which the flux is evaluated. The magnetic diffusion terms are temporarily ignored. The algorithm updates the gas cells as follows,

$$\frac{U_{gas}^{n+1} - U_{gas}^{n}}{\Delta t} + \sum_{n_{faces}} F_f^c \cdot \hat{n}_f s_f = 0 \tag{26}$$

The face flux F_f^c is chosen as F_g^c (Eq. (13)) if the face is a gas-face and F_{gi}^c (Eq. (25)) if the face is a gas-interface face. The flux F_{gi}^c fluxes gas into the interface cells that lie adjacent to the gas cells. This layer of interface cells are the only vacuum cells that are updated in the time-stepping algorithm. The interface cell update is given by

$$\frac{U_{int}^{n+1} - U_{int}^{n}}{\Delta t} + \sum_{n_{faces}} F_{gi}^{c} \cdot \hat{n}_f s_f = 0 \tag{27}$$

The flux from the gas cell into the interface cell changes the values of its conservative variables which physically represents the filling up of the interface cell with gas. Though it would seem logical to assume that this would necessarily result in the transition of the interface cell to a gas cell, initial implementation of this idea results in very small values of the conservative variables. A similar issue arises with the point tracking algorithm of Munz. The movement of the point into a vacuum cell at every new iteration causes the filling up of this cell with gas. The fluxes responsible for this fill up produce small values of density and pressure giving rise to incorrect wave speeds and consequently over-predicting the interface location. In order to avoid this issue, our algorithm implements a threshold for the interface cell to transition to a gas cell. The interface cell accumulates flux, as given by Eq. (27), and transitions to a gas cell only when its density is above a certain threshold value. During the course of the computation, the density within the interface cells is tracked and even though some of these cells accumulate a non-zero density they transition to a gas cell only when their density exceeds the threshold. The threshold is calibrated using the theoretical free expansion solution as a reference.

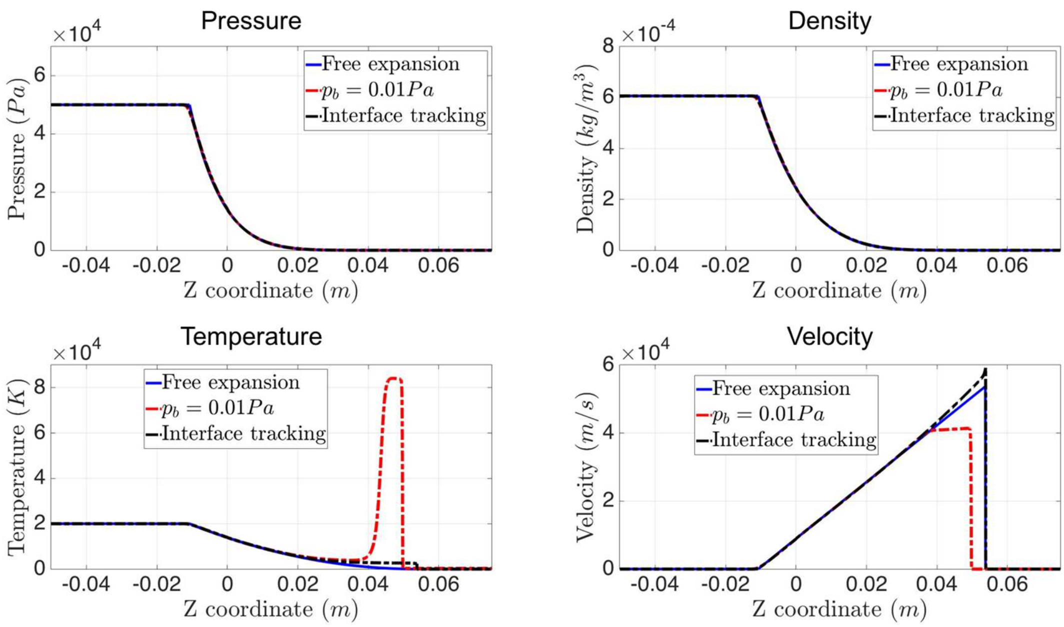

Figure 8: Comparison of the theoretical free expansion solution at $t = 1\mu s$ with the low pressure background simulation and the interface tracking approach using a density threshold of $2 \times 10^{-9} kg/m^3$

A comparison between the theoretical free expansion solution and the interface tracking algorithm presented above is shown in the Fig. 8. The threshold is calibrated by matching the front location predicted using the interface tracking approach with the theoretical free expansion solution. In order to calibrate the threshold, 1D simulations are carried out using the interface tracking approach for a wide range of threshold values. Then at a time of $1\mu s$, the velocity contours in solutions obtained using these values are compared to the theoretical free expansion solution. The threshold value that gives the closest location of the plasma-vacuum front, defined as the location associated with the sudden jump in axial velocity, is chosen as the value of the calibrated threshold. For the driver section density used in the shock tube problem in section 4.2 the threshold is found to be equal to $2 \times 10^{-9} kg/m^3$. The solution obtained using a low background density approach is also shown as reference. It is found that the solution is a strong function of the choice of the density threshold. The absence of a density threshold leads to the over prediction of the tip speeds and the tracking of exceedingly small densities whereas high threshold values lead to an under prediction of the front location. The optimal threshold choice gives a front location close to that predicted by the theoretical solution while simultaneously predicting comparable tip speeds. The temperature and velocity profiles represent the solution well in the high density regions all the way up to the tail of the rarefaction. However, in a small region near the tail of the rarefaction, the temperature profile terminates with a non-zero value and the velocity is slightly over predicted. These features are a consequence of the thresholding, the choice of a density threshold ensures that the new gas cell has a minimum density/pressure and as a consequence of the equation of state, a minimum temperature. Hence, unlike the theoretical solution where the temperature profile smoothly goes to zero towards the interface, the numerical solution terminates with a non-zero temperature at the gas-vacuum interface. However, the monotonic decrease in temperature and a monotonic increase in velocity in this region is still in qualitative agreement with the theoretical solution. Moreover, as compared to the background density approach, the algorithm gives a physically consistent prediction of gas-vacuum expansion without the formation of a spurious shock.

4.4 2-D Interface tracking

Comparing the Vacuum Riemann Solver (VRS) to the 1D theoretical solution gives an approximate method to calibrate the threshold. Once calibrated, the threshold value is used to carry out 2D simulations of thermal plasma jets. The simulated geometry is shown in Fig. 9 below; it consists of a shock tube setup that opens up into a plume region.

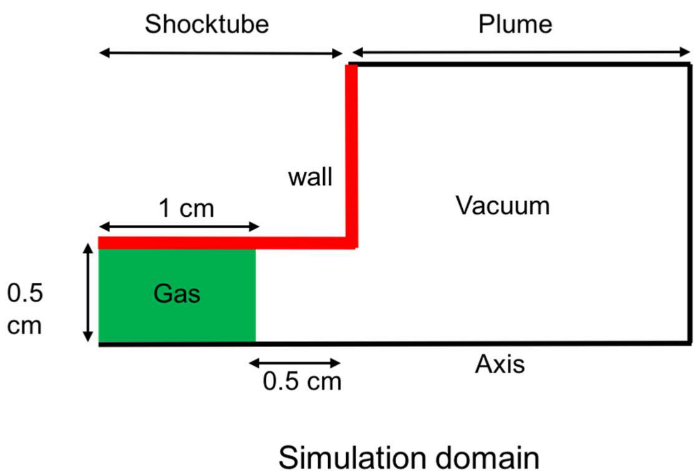

Figure 9: Schematic of simulation domain to study two-dimensional vacuum expansion. The gas expands through a shock tube that eventually opens up into a plume.

The interface tracking algorithm tracks a subset of the total number of cells and faces in the mesh (those tagged as containing the gas). This subset increases as the gas (plasma) expands to fill the domain. For a generalized unstructured mesh, the interface tracking is performed using a tagging scheme based on assigning flags. Gas cells are assigned an integer flag of 1, interface cells i.e. vacuum cells that border the gas cells are assigned a flag 0 and non-interface vacuum cells are assigned a flag -1. An example illustrating the initial set of tagged cells at a small cross-section near the interface is shown in Fig. 10 (a). The face classification in 2D, similar to the 1D case, involves identifying the gas-gas (1,1) faces, and gas-interface (1,0) faces. Additionally, the interface-interface (0,0) and interface vacuum (0, -1) faces are also tracked. Each of the classified groups (faces and cells) are tracked using separate lists that are updated during the course of the time-step update. The interface tracking involves looping over the gas-gas faces and gas-interface faces to update the conservative variables in the gas cells as given by Eq. (26).

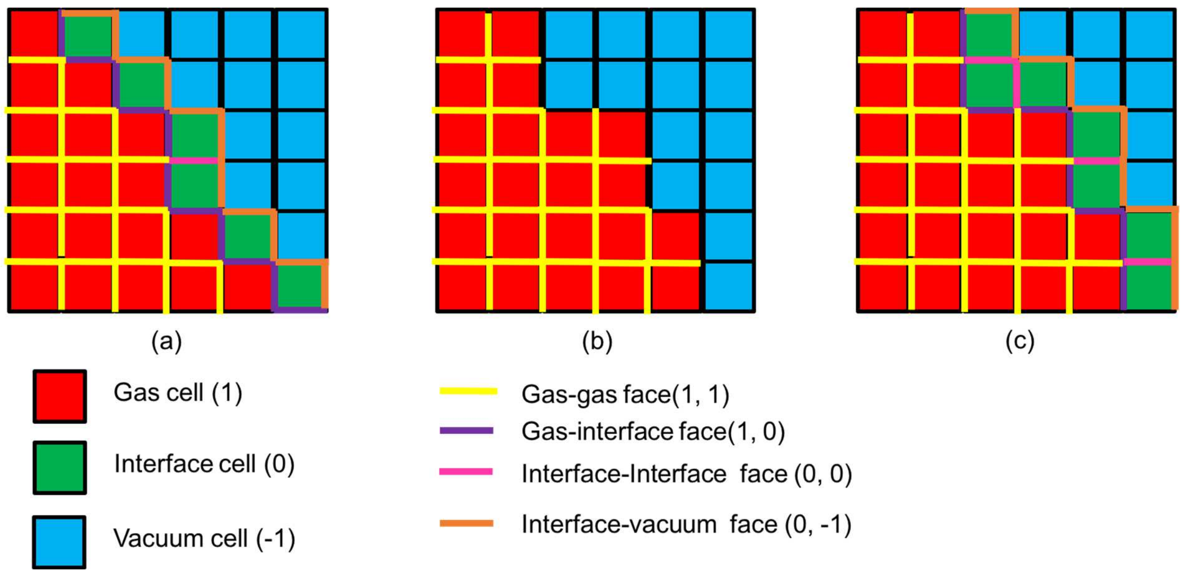

Figure 10: Schematic indicating the change in cell tags over the course of a single time step update. (a) Gas/Interface/Vacuum cells along with gas-interface/interface-interface faces before the time step update (b) Gas/Vacuum cells after the thresholding has been performed (c) New set of interface cells and faces after performing the interface identification

Subsequently, the gas-interface, interface-interface and interface-vacuum faces are looped over to update the conservative variables in the interface cells (Eq. (27)). The interface-interface and interface-vacuum faces give a non-zero flux only if either of the interface cells have a non-zero density. This occurs when the interface cells have not transitioned to a gas cell. Once the conservative variables in the interface cells are updated, the density in these newly filled cells is compared to the threshold. If the density exceeds the threshold, the cell is assigned the integer flag of 1 and pushed into the list of gas cells. If the density does not exceed the threshold it is assigned a flag of -1. The last step of the 2D algorithm involves identifying the new interface cells for the next iteration of the time-stepping scheme. After the thresholding, the cells are tagged as either gas cells (1) or vacuum cells (-1). This state is represented in Fig. 10 (b) and it can be seen that there are no cells tagged as representing the interface (0). A straightforward implementation of interface identification would involve looping over all the faces of the unstructured mesh and classifying the faces at the gas-vacuum boundary (1, -1) as gas-interface faces and tagging the vacuum cell at this boundary as an interface cell (0). This would involve looping over all the faces of the unstructured mesh which is a computationally expensive process. This limitation is improved upon by the realization that the changes in the integer flags only occur at the interface layer. Once the thresholding is performed, the gas-interface faces (Fig. 10 (a)) at the previous timestep are looped over to check if the interface cell has transitioned to a gas cell. If this transition has taken place, the face is pushed into the list of gas (1,1) faces. If the transition doesn't occur the vacuum cell (-1) is assigned a flag 0 and pushed into the list of new interface cells. A similar loop is performed over the interface-interface and interface-vacuum faces where the presence of a (1, -1) adjacent cell flag index involves assigning the vacuum cell (-1) an interface flag (0) and pushing the corresponding cell into the list of new interface cells. The last step of this process involves identifying the new gas-interface, interface-interface and interface-vacuum faces. This process is performed by looping over the faces of the new interface cells to identify the new gas-interface (1,0), interface-interface (0,0) and interface-vacuum (0, -1) faces. The new layer of interface cells and faces are shown in figure 10 (c). Hence, the entire process of identifying the new interface shape involves looping interface data from the previous timestep. Typical meshes used to simulate plasma accelerators have close to a million faces/cells as opposed to approximately thousand faces/cells that are used to track the plasma-vacuum interface (gas-interface, interface-interface, interface-vacuum). This is expected since the interface is essentially a 1D line in a 2D mesh. Hence, in an unstructured mesh the interface tracking predicts the new interface shape by looping over a thousand faces/cells as opposed to a million as proposed in the straightforward implementation.

The 2D interface tracking algorithm outlined above is used to carry out simulations of thermal plasma jets. The VRS algorithm is compared against a simulation with the same driver state but a low background density/pressure initial condition. The simulations shown in Fig. 11 are performed using quadrilateral mesh elements with a uniform mesh element size of 0.00625 cm in the radial direction and 0.015 cm in the axial directions. The the pressure and density of the gas in the shocktube is set to $P^i = 10^5 Pa$ and $\rho^i = 10^{-3} kg/m^3$. The vacuum conditions in the low background density approach are approximated using a density of $\rho^b = 10^{-7} kg/m^3$ and a pressure $P^b = 1 Pa$. For the interface tracking simulations, the vacuum is assigned a zero density and pressure. An inviscid wall boundary condition is imposed on the electrodes, axisymmetric boundary condition on the axis and a supersonic outflow boundary condition on the far-field boundaries. A CFL value of 0.9 was used to determine the timestep based on the maximum convective wavespeed. The results of this test are shown in Fig. 11. It is observed that the interface tracking approach predicts a plasma plume profile that is consistent with free expansion physics. The temperature contours shown in Fig. 11 (c), (d) indicate that the spurious shock obtained using the background density/pressure approach is absent in the interface tracking solution. The velocity contours shown in Fig. 11 (a), (b) show a significantly higher tip velocity obtained using the VRS approach as compared to low density background approach. This is indicative of the fact that the use a low background density/pressure initial condition inhibits the free expansion of the jet.

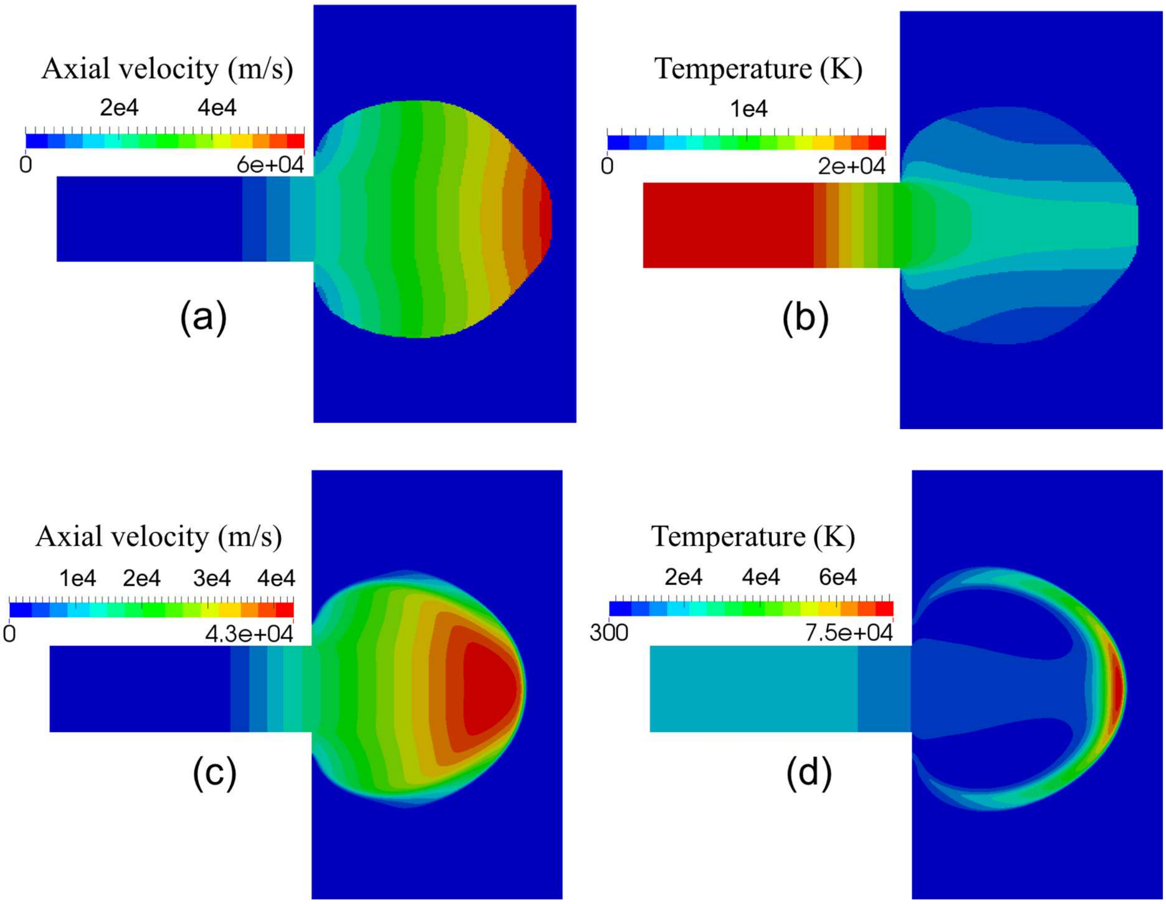

Figure 11: Comparison of velocity and temperature profiles of a 2D vacuum expansion performed using the interface tracking approach (a), (b) and the low background density approach (c), (d) and at $t = 0.5\mu s$

4.5 Implicit interface tracking

The VRS algorithm presented in the previous section used an explicit time-stepping scheme to perform the conservative variable update (Eqs. (26) and (27)). The maximum allowable time step in this scheme, as given by the CFL criterion, is bounded above by the size of the smallest mesh element in the domain and the fastest wave speed passing through this element [24]. For the explicit VRS plume simulations in the previous section this time step is found to be around $5 \times 10^{-10} s$. For larger values of Δt, numerical oscillations are observed near the tip of the rarefaction leading to the generation of cells with negative pressures/temperatures. In order to carry out simulations of plasma accelerators, fine mesh elements are required in the regions close to the inner electrode in order to resolve phenomena such as the pinching effect [32]. The elements in these regions can severely limit the maximum allowable time step when the system is evolved using an explicit time stepping scheme. This motivates the development of an implicit interface tracking scheme. The implicit tracking scheme is formulated using the system ODE notation (Sec. 3.2). The system ODE is obtained by collecting the cell centered conservative variables and residual vectors into a global conservative/residual vector. The global conservative vectors corresponding to the gas/interface cells are differentiated using corresponding subscripts. In this notation the explicit update algorithm outlined in section 4.2 is given by:

$$\frac{\boldsymbol{U}_{gas}^{n+1} - \boldsymbol{U}_{gas}^{n}}{\Delta t} + \boldsymbol{R}_{gas}^{c}(\boldsymbol{U}_{gas}^{n}) = 0. \tag{28}$$

$$\frac{\boldsymbol{U}_{int}^{n+1} - \boldsymbol{U}_{int}^{n}}{\Delta t} + \boldsymbol{R}_{gas}^{c}(\boldsymbol{U}_{gas}^{n}, \boldsymbol{U}_{int}^{n}) = 0. \tag{29}$$

The notation indicates that the conservative variables in the gas cells change due to a residual operator that depends on the data in the gas cells. The conservative variables on the interface cells change due to the fluxes flowing in from the gas cells and due to the fluxes between interface cells that contain non-zero densities (less than the threshold). The above system has a maximum allowable Δt limited by the CFL criteria. To overcome this limitation, the following implicit system is solved:

$$\frac{\boldsymbol{U}_{gas}^{n+1} - \boldsymbol{U}_{gas}^{n}}{\Delta t} + \boldsymbol{R}_{gas}^{c}(\boldsymbol{U}_{gas}^{n+1}) = 0. \tag{30}$$

$$\frac{\boldsymbol{U}_{int}^{n+1} - \boldsymbol{U}_{int}^{n}}{\Delta t} + \boldsymbol{R}_{gas}^{c}(\boldsymbol{U}_{gas}^{n+1}, \boldsymbol{U}_{int}^{n+1}) = 0. \tag{31}$$

This produces a system of algebraic equations which can be written in a similar form as presented in Eq. (21).

$$\frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^{n}}{\Delta t} + \boldsymbol{R}(\boldsymbol{U}^{n+1}) = 0, \tag{32}$$

$$G(\boldsymbol{U}^{n+1}) = 0.$$

Here,

$$\boldsymbol{U}^{n+1} = \left[\boldsymbol{U}_{gas}^{n+1}, \boldsymbol{U}_{int}^{n+1}, \boldsymbol{U}_{vac}^{n+1}\right],$$
$$\boldsymbol{R} = \left[\boldsymbol{R}_{gas}^{c}\left(\boldsymbol{U}_{gas}^{n+1}\right), \boldsymbol{R}_{gas}^{c}\left(\boldsymbol{U}_{gas}^{n+1}, \boldsymbol{U}_{int}^{n+1}\right), 0\right]. \tag{33}$$

In this notation $\boldsymbol{U}_{vac}$ represents the vector of conservative variables on all the vacuum cells and the zero residual in the corresponding global residual vector represents the fact that these variables are not updated in the timestepping scheme. The algebraic system represented by Eq. (32) is solved using the Newton method outlined in section 3. The functional form of the linear system that is solved as a part of the non-linear iterative scheme is written out in a manner that illustrates the decomposition of the global conservative variables into gas, interface and vacuum cells. This leads to a corresponding decomposition of the Jacobian given by,

$$\begin{bmatrix} \Delta\boldsymbol{U}_{gas}^{k} \\ \Delta\boldsymbol{U}_{int}^{\boldsymbol{k}} \\ \Delta\boldsymbol{U}^{\boldsymbol{k}}{}_{vac} \end{bmatrix} = \begin{bmatrix} \dfrac{I}{\Delta t} + \dfrac{\partial \boldsymbol{R}^{c}{}_{gas}}{\partial \boldsymbol{U}_{gas}} & 0 & 0 \\ \dfrac{\partial \boldsymbol{R}_{int}^{c}}{\partial \boldsymbol{U}_{gas}} & \dfrac{I}{\Delta t} + \dfrac{\partial \boldsymbol{R}_{int}^{c}}{\partial \boldsymbol{U}_{int}} & 0 \\ 0 & 0 & \dfrac{I}{\Delta t} \end{bmatrix}^{-1} \begin{bmatrix} \dfrac{\boldsymbol{U}_{gas}^{n} - \boldsymbol{U}_{gas}^{k}}{\Delta t} - \boldsymbol{R}_{gas}^{c}(\boldsymbol{U}_{gas}^{n+1}) \\ \dfrac{\boldsymbol{U}_{int}^{n} - \boldsymbol{U}_{int}^{k}}{\Delta t} - \boldsymbol{R}_{gas}^{c}(\boldsymbol{U}_{gas}^{n+1}, \boldsymbol{U}_{int}^{n+1}) \\ 0 \end{bmatrix}. \tag{34}$$

The convergence of the Newton iteration is defined by a relative tolerance criteria, the system being deemed as converged if the l^2 norm of the function G (Eq. (32)) reduces by four orders of magnitude. The sparse linear system inversion at each Newton step is carried out using the Generalized Minimum Residual (GMRES) algorithm using a block-Jacobi preconditioner implemented in the Portable and Extensible Toolkit for Scientific Computing (PETSc) library [34]. The block-Jacobi preconditioner is a Schwarz-type block-Jacobi method with a ILU(0) factorization. The last step in the design of the implicit tracking scheme involves specifying the frequency at which the interface tracking update is performed within a given non-linear iteration. The interface tracking update process, as outlined in section 4.4, involves comparing the density in the newly filled interface cells to the threshold value and changing the cell flags based on the result of this comparison. In the case of the explicit time stepping scheme, the interface tracking update is called each time the conservative variables (Eqs. (26) and (27)) in the domain are updated. The Newton method on the other hand, involves multiple updates of the conservative variables within a given time step. The conservative variables are updated each time the linear system (Eq. (34)) is solved. One choice is to call the interface tracking update after every linear solve. In practice, this approach leads to the formation of jagged front shapes and over predicts the front position when compared to the explicit tracking scheme. Another choice is to call the interface tracking routine after all the Newton iterations within a time step have been performed. Such an implementation grossly under predicts the front location for large time steps. However, the shape of the front is smooth and is similar to that obtained using the explicit scheme. As a compromise between the two approaches, the current implicit VRS algorithm calls the interface tracking update after solving the linear system two consecutive times within a non-linear iteration.

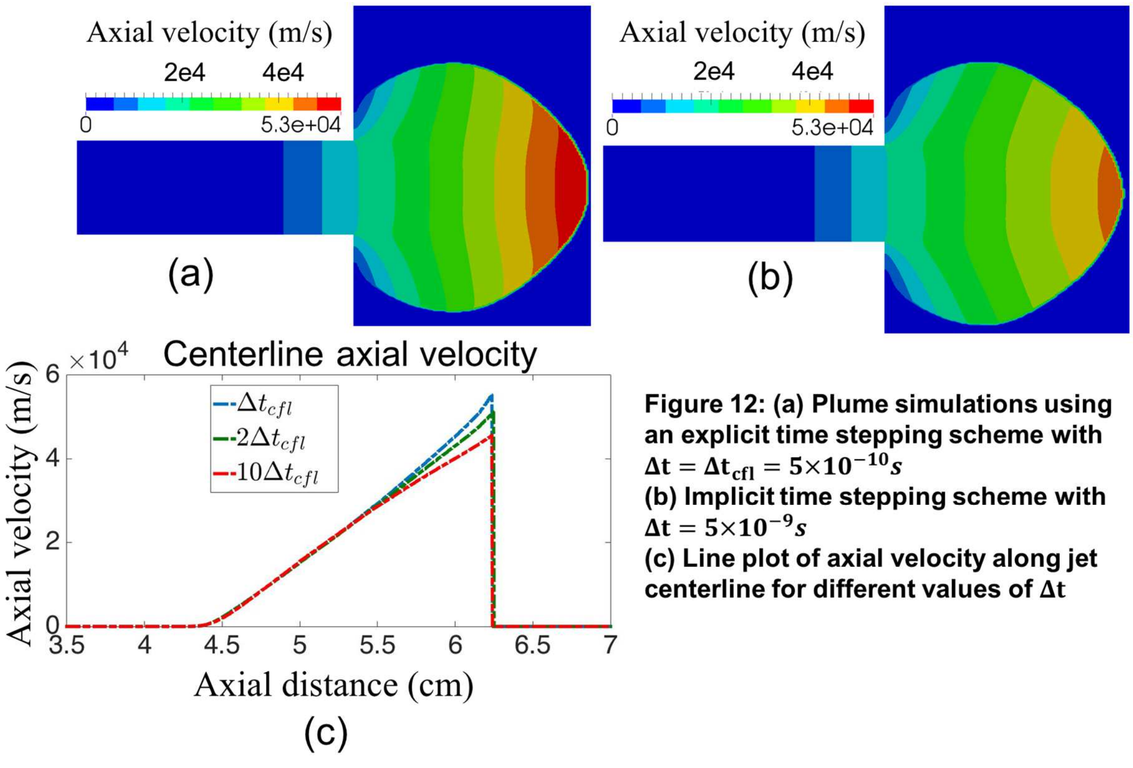

Figure 12: (a) Plume simulations using an explicit time stepping scheme with $\Delta t = \Delta t_{cfl} = 5{\times}10^{-10} s$
(b) Implicit time stepping scheme with $\Delta t = 5{\times}10^{-9} s$
(c) Line plot of axial velocity along jet centerline for different values of Δt

The front shape and axial velocity along the centerline obtained using the implicit time stepping scheme is compared to the explicit scheme 0.3 μs into the simulation. The implicit simulation is carried out using a time step of $5 \times 10^{-9} s$, ten times the maximum allowable time step limitation associated with the explicit scheme. This corresponds to 600 time steps of in the explicit scheme as opposed to 60 time steps of the implicit scheme. The results of this comparison are shown in Fig. 12. It is seen that the implicit scheme produces a qualitatively similar front shape with similar solutions obtained in the high density regions. In regions near the tip of the 2D rarefaction, the implicit scheme suffers from errors generated as a consequence of taking a larger time step. It is seen that the axial velocity is under predicted towards the tip region. However, as seen in Fig. 12 (c) the implicit interface tracking scheme is able to predict a front location similar to that shown by the explicit scheme. Moreover, from the perspective of carrying out simulations of coaxial plasma accelerators, the ability to operate at larger time steps and a consistent prediction of the front shape outweigh the errors observed in these low density regions.

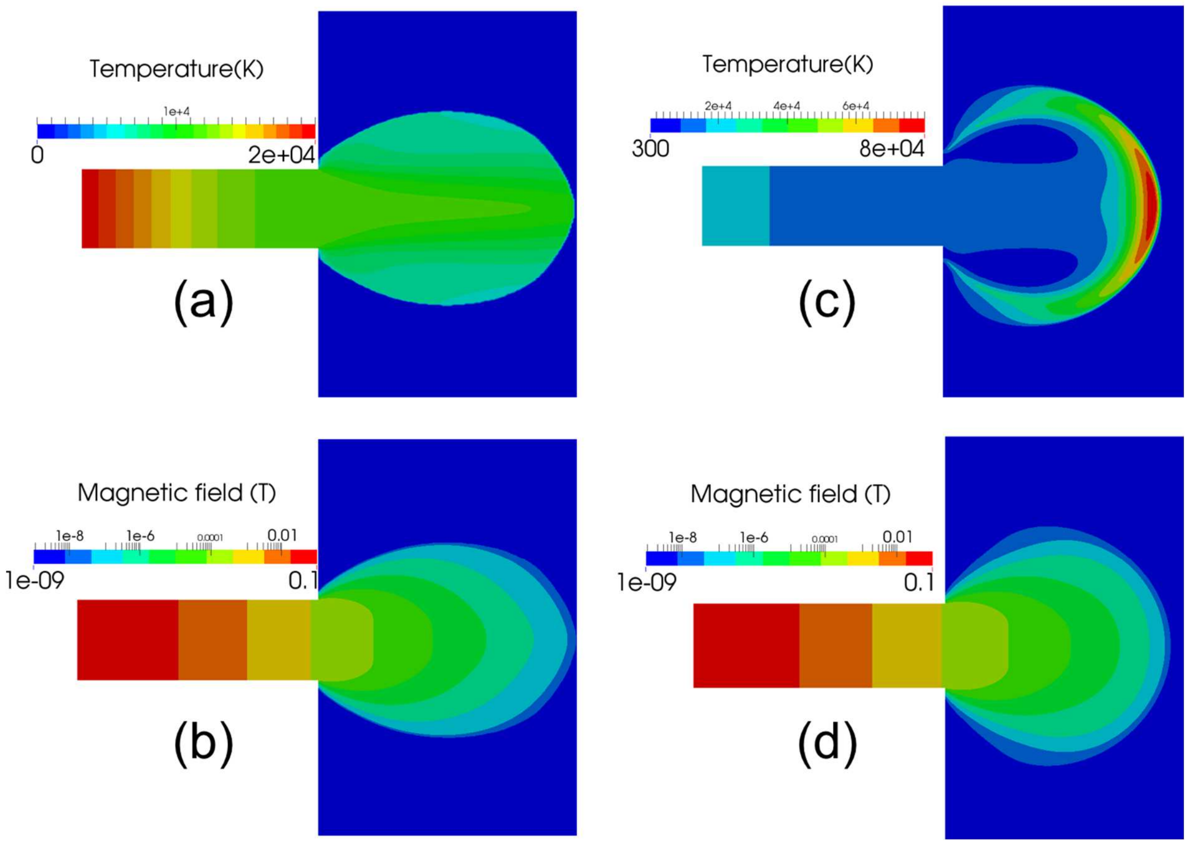

Figure 13: Implicit ideal-MHD simulations of magnetized plasma jets performed using the interface tracking approach (a), (b) and the background density approach (c), (d) for the coaxial plasma accelerator geometry shown in Fig. 1

4.6 Extension to ideal and resistive MHD

The implementation of the VRS tracking framework into an explicit and implicit time stepping scheme in the previous sections exclusively involved integrating the convection operator. The magnetic fields were set to zero in order to obtain these results. Generalizing this approach to a convective operator that includes the transport equation for the magnetic induction vector involves replacing the acoustic wave speed used in section 4.3 $c_f = \sqrt{\frac{\gamma P}{\rho}}$ with the fast magnetosonic speed $c_f = \sqrt{\frac{\gamma P}{\rho} + \frac{B_\theta^2}{\rho}}$. This modification extends the VRS tracking framework to the ideal MHD equations. The implicit ideal-MHD interface tracking scheme is compared to the background density method for plasma accelerator simulations. The simulated geometry is shown in Fig. 1. The gradients in the magnetic field (Fig. 13 (b), (d)) accelerate the plasma as it expands into the vacuum background. The strong shock observed at the jet-boundary (Fig. 13 (c)) using the background density/pressure approach is eliminated in the interface tracking solution (Fig. 13 (a)). The magnetic field contours indicate an extended plume profile (Fig. 13 (b)) as opposed to the shock-inhibited expansion predicted by the background density/pressure approach (Fig. 13 (d)).

Extending the formulation to the resistive MHD system requires the specification of an interface boundary condition for the diffusion operator. The gas-interface flux used in Eq. (25) is the Neumann-type interface boundary condition for the convective operator. Extending the formulation to include the resistive diffusion operator requires the derivation of an equivalent physically consistent interface boundary condition. This is obtained by integrating Ampere's law. Since the plasma accelerator problem consists solely of an azimuthal magnetic field component, Ampere's law for the radial current density is given by,

$$\frac{\partial B_\theta(r,z)}{\partial z} = -\mu_0 j_r(r,z)\,. \tag{35}$$

Here, $j_r(r,z)$ represents the radial current density. Integrating Eq. (35) over the surface of a fictitious cylinder (illustrated in Fig. 14) that extends from the interface boundary to infinity with a radius equal to the distance of the interface cell from the axis gives,

$$-(B_\theta(r_{int},\infty) - B_\theta(r_{int},z_{int})2\pi r_{int} = \mu_0 \int_{z_{int}}^{\infty} j_r(r_{int},z)2\pi r_{int}dz \tag{36}$$

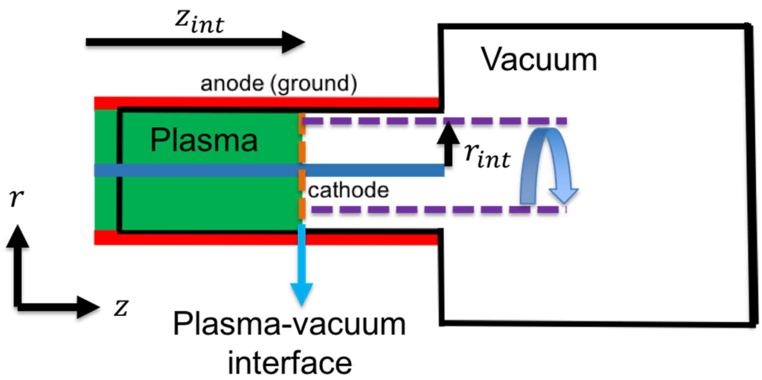

Figure 14: Schematic illustrating a cross section of the fictitious cylindrical integration surface (purple dashed line) used to obtain the magnetic field at a location on the plasma vacuum interface (yellow dashed line) with axial coordinate z_{int} and radial coordinate r_{int}.

Here, z_{int} is the axial location of the interface cell (Fig. 14). Since the magnetic field far away from the accelerator goes to zero and the total current downstream of the interface is also zero (vacuum), the azimuthal magnetic field at the interface is zero. This gives rise to the following expression for the magnetic diffusion flux at the gas-interface boundary.

$$F^r_{gi} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \eta_g \left(\dfrac{B_g}{d_{gi}}\right) \hat{d}_{gi} \\ \eta_g\, B_g \left(\dfrac{B_g}{d_{gi}}\right) \hat{d}_{gi} \end{bmatrix} \tag{38}$$

Here, η_g denotes the resistivity of the gas cell, B_g the cell centered magnetic field at the gas cell, d_{gi} the scalar distance between the gas and interface cell centers and $\hat{d}_{gi}$ the unit vector from the gas cell center to the interface cell center. The terms in the brackets denote the magnitude of the magnetic field gradient at the gas-interface face. Since the magnetic field at the interface cell is set to zero, the gradient, that is equal to the difference between the magnetic field at the gas and interface cells, takes the expression as shown in Eq. (38). The specification of the magnetic diffusion interface flux enables the construction of a residual operator identical to that formed in section 4.3 (Eq. (30), (31)). The resistive diffusion operator update is exclusively performed in the cells that are classified as containing gas (plasma). This is justified since the magnetic diffusion flux does not induce density transport into the interface cells and as a result does not affect the interface growth rate. Within the density thresholding framework described in this paper, the interface growth is governed solely by convective phenomena. The algebraic system obtained as a consequence of the implicit time-discretization of the resistive MHD equations is given by,

$$\frac{\boldsymbol{U}_{gas}^{n+1} - \boldsymbol{U}_{gas}^{n}}{\Delta t} + \boldsymbol{R}_{gas}^{c}\left(\boldsymbol{U}_{gas}^{n+1}\right) = \boldsymbol{R}_{gas}^{r}(\boldsymbol{U}_{gas}^{n+1}) \tag{39}$$

$$\frac{\boldsymbol{U}_{int}^{n+1} - \boldsymbol{U}_{int}^{n}}{\Delta t} + \boldsymbol{R}^{c}{}_{int}(\boldsymbol{U}_{gas}^{n+1}, \boldsymbol{U}_{int}^{n+1}) = 0 \tag{40}$$

The algebraic system is solved using a Jacobian decomposition that includes the contribution from the resistive diffusion flux. The functional form of the linear system that is solved as a part of the non-linear iterative scheme is written as:

$$\begin{bmatrix} \Delta \boldsymbol{U}_{gas}^{k} \\ \Delta \boldsymbol{U}_{int}^{k} \\ \Delta \boldsymbol{U}^{\boldsymbol{k}}{}_{vac} \end{bmatrix} = \begin{bmatrix} \frac{I}{\Delta t} + \frac{\partial \boldsymbol{R}_{gas}^{c}}{\partial \boldsymbol{U}_{gas}} - \frac{\partial \boldsymbol{R}_{gas}^{r}}{\partial \boldsymbol{U}_{gas}} & 0 & 0 \\ \frac{\partial \boldsymbol{R}_{int}^{c}}{\partial \boldsymbol{U}_{gas}} & \frac{I}{\Delta t} + \frac{\partial \boldsymbol{R}_{int}^{c}}{\partial \boldsymbol{U}_{int}} & 0 \\ 0 & 0 & \frac{I}{\Delta t} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\boldsymbol{U}_{gas}^{n} - \boldsymbol{U}_{gas}^{k}}{\Delta t} - \boldsymbol{R}_{gas}^{c}\left(\boldsymbol{U}_{gas}^{k}\right) + \boldsymbol{R}_{gas}^{r}(\boldsymbol{U}_{gas}^{n+1}) \\ \frac{\boldsymbol{U}_{int}^{n} - \boldsymbol{U}_{int}^{k}}{\Delta t} - \boldsymbol{R}_{int}^{c}(\boldsymbol{U}_{gas}^{k}, \boldsymbol{U}_{int}^{k}) \\ 0 \end{bmatrix} \tag{41}$$

The extension of the interface tracking approach to the resistive MHD system leads to significant wall-clock time gains. The predominant cost of carrying out resistive MHD simulations arises from performing the sparse linear system inversion. The solution vector $\boldsymbol{\Delta U}$ (Eq. (41)) is found using the GMRES algorithm, a method that belongs to the family of iterative linear solvers known as Krylov Subspace (KSM) Methods [33]. The net wall-clock time associated with carrying out resistive MHD simulations is proportional to the number of linear solver iterations. Also known as the KSM iterations, these are the number of iterations taken by the GMRES algorithm to find a solution vector $\boldsymbol{\Delta U}$ that approximately satisfies Eq. (41) within some relative error tolerance criterion [27]. The number of KSM iterations are a known to be a function of the matrix condition number and can be significantly reduced by using an effective preconditioner [34]. The results presented in this paper use the ILU and block Jacobi preconditioners. For resistive MHD simulations, the condition number of the linear system is a function of the the initial boundary value problem setup. Prior efforts to simulate plasma accelerators using the resistive MHD equations, in addition to assigning a low density to the vacuum background, use a high background resistivity to mimic the poor conductivity of vacuum [4]. It was shown in section 4.1 that the low background density produces a shock. The high background resistivity on the other hand, leads to a significant increase in the condition number of the linear system and consequently the number of KSM iterations taken for the linear solve to converge. The large condition number of the resistive MHD linear system represents the numerical stiffness associated with the underlying equation system [35]. This stiffness arises due to a disparity in the characteristic timescales associated with the convection and magnetic diffusion operators. For the convection operator, this timescale is given by Δ/c_f where c_f is the estimate of the largest magnetosonic wavespeed in magnitude and for magnetic diffusion operator this is give by Δ^2/η where η is the largest resistivity in the domain. Here, Δ is an estimate for the typical mesh element size. These estimates are also the maximum allowable time step for the explicit integration of the corresponding residual operators. The MHD simulation setup [3,4] that involves specifying a high background resistivity in order to consistently mimic the poor conductivity properties of the vacuum background generates linear systems that have a large condition number. The condition number increases with the increase the background resistivity, since this leads to a lower diffusion timescale and a larger disparity in the characteristic timescales (higher stiffness). Typical plasma accelerator simulations [4] specify a ceiling for the background resistivity as a tradeoff between accurately representing the vacuum background and limiting the number of KSM iterations. Table 1 shows the condition number of the Jacobian matrix for different values of the background resistivity ceiling. Also shown are the average number of KSM iterations within a Newton solve.

Background resistivity	Jacobian condition number	Avg. KSP iterations/newton solve
$0.1\ m/S$	1.087×10^4	517
$0.04\ m/S$	5.241×10^3	312
$0.02\ m/S$	3.156×10^3	187
$0.01\ m/S$	1.581×10^3	128
Interface tracking	54.88	12

Table 1: Jacobian condition numbers and average KSM iterations per Newton step for different values of the background resistivity ceiling

It is seen than an increase in the resistivity ceiling increases the matrix condition number and the number of KSM iterations. The ILU and block-Jacobi preconditioners are used in conjugation with the GMRES algorithm to perform the iterative linear solves reported in Table 1. Within the context of a background resistivity approach, Table 1 indicates that moving towards a consistent representation of the vacuum background (higher resistivity) results in significantly higher KSM iterations and wall-clock times. The interface tracking approach, on the other hand, by identifying the plasma (gas) cells and by solving for the resistive diffusion exclusively in these cells, overcomes the issue of specifying a background resistivity. Here, the timescale disparity is produced exclusively by the resistivity in the plasma cells that tends to be significantly lower than the ceiling values used in background resistivity simulations. Consequently, the linear system produced as a result of this method has a condition number that is three orders of magnitude smaller than that obtained from using the background resistivity approach. This results in significantly lower KSM iterations as shown in Table 1. The lower KSM iterations lead to significant wall-clock time gains as shown in Fig. 11 below.

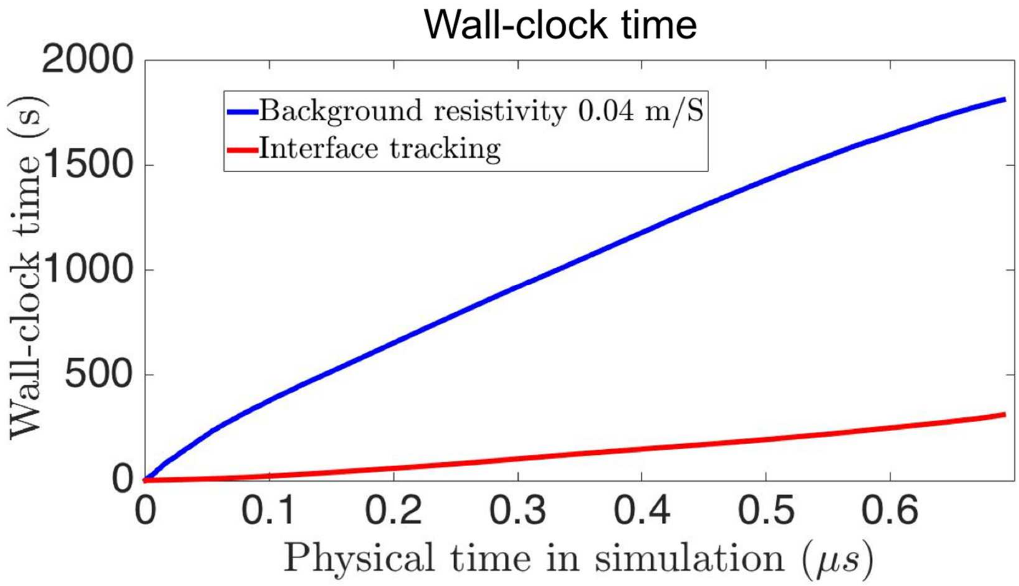

Figure 15: Wall-clock time vs. physical time in a resistive MHD plasma accelerator simulation for the background resistivity (blue) and interface tracking (red) approach

6. Conclusion

In the context simulating plasma accelerators operating in a vacuum background, it is seen that solving the MHD governing equations in the vacuum background is both physically inconsistent and computationally expensive. For the convection operator, approximating the vacuum using a background density and solving for convection over the entire (plasma and vacuum) simulation domain produces an unphysical shock. The shock was shown to be produced as a consequence of the properties of the underlying convective flux scheme. This motivated the design of a threshold based plasma-vacuum interface tracking scheme that was shown to eliminate the shock and produce plume profiles that are physically consistent with free expansion physics. The density threshold was calibrated using a theoretical 1D free expansion solution as reference as shown in Sec. 4.3 Fig. (5). The interface tracking algorithm was implemented in a generalized unstructured mesh setting using cell based tags and extended to be incorporated within an implicit time stepping scheme. The implicit scheme suffers from numerical diffusion effects at the gas-vacuum interface but is able to operate at ten times the maximum allowable explicit CFL time step. The implicit ideal-MHD based interface tracking method was used to demonstrate physically consistent plume profiles in magnetically accelerated jets. For the resistive diffusion operator, it was shown that approximating the background using a high resistivity ceiling and solving for the magnetic diffusion through this background generates stiff linear systems. Within this formulation, the computational resources are predominantly devoted to solving the magnetic diffusion through the vacuum background. From a perspective of physical interpretability, solving for this diffusion is meaningless, since within the MHD formulation, the vacuum background cannot conduct currents and consequently cannot sustain magnetic fields. The interface tracking approach, by identifying and solving for the magnetic diffusion exclusively within the plasma cells, produces significantly less stiff systems that exhibit faster convergence of the linear solve and consequently smaller wall clock time.

Acknowledgement

The authors thank the U.S. Department of Energy, SSAA Grant DE-NA0002011 for support of this work.

References:

[1] K.Watanabe and T. Sato, Global simulation of the solar wind-magnetosphere interaction: The importance of its numerical validity, Journal of Geophysical Research: Space Physics (1978–2012) 95(A1) (1990) 75-88

[2] J.S. Shang, P.W. Canupp, and D.V. Gaitonde, Computational Magneto—Aerodynamic Hypersonics, AIAA 99-4903 (1999) Vol. 1 No. 5

[3] K. Sankaran, L. Martinelli, S.C. Jardin, and E.Y. Choueiri, A flux-limited numerical method for solving the MHD equations to simulate propulsive plasma flows, International journal for numerical methods in engineering, 53(6) (2002) 1415-1432

[4] H. Sitaraman and L.L. Raja, Magneto-hydrodynamics simulation study of deflagration mode in co-axial plasma accelerators, Physics of Plasmas 21 (2014) 012104

[5] K. Sankaran, E.Y. Choueiri , and S. C. Jardin, Comparison of simulated magnetoplasmadynamic thruster flowfields to experimental measurements, Journal of propulsion and power 21(1) (2005) 129-138

[6] H. Luo, J.D. Baum, and R. Löhner, A fast, matrix-free implicit method for compressible flows on unstructured grids, Journal of Computational Physics 146(2) (1998) 664-690.

[7] P.G. Mikellides, P.J. Turchi, and N.F. Roderick, Applied-field magnetoplasmadynamic thrusters, Part 1: Numerical simulations using the MACH2 code, Journal of Propulsion and Power, 16(5) (2000) 887-893.

[8] J.T. Cassibry, F. Thio, T.E. Markusic, and S.T. Wu, Numerical modeling of a pulsed electromagnetic plasma thruster experiment, Journal of propulsion and power, 22(3) (2006) 628-636.

[9] M.H. Frese, MACH2: A Two-Dimensional Magnetohydrodynamic Simulation Code for Complex Experimental Configurations (No. AMRC-R-874) Mission Research Corp Albuquerque NM (1987)

[10] C.M. Xisto, J.C. Páscoa, and P.J. Oliveira, Numerical Modelling of Electrode Geometry Effects on a 2D Self-Field MPD Thruster, ASME 2013 International Mechanical Engineering Congress and Exposition (2013) pp. V001T01A051-V001T01A051

[11] R.I. Issa, A.D. Gosman, and A.P. Watkins, The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme, Journal of Computational Physics 62(1) (1986) 66-82.

[12] F.R. Poehlmann, M.A. Cappelli, and G.B. Rieker, Current distribution measurements inside an electromagnetic plasma gun operated in a gas-puff mode, Physics of Plasmas 17.12 (2010) 123508.

[13] B. Einfeldt, C.D. Munz, P.L. Roe, B. Sjögreen, On Godunov-type methods near low densities, Journal of computational physics, 92(2) (1991) 273-295

[14] Q. Sun, I.D. Boyd, and G.V. Candler, A hybrid continuum/particle approach for modeling subsonic, rarefied gas flows. Journal of Computational Physics 194(1) (2004) 256-277

[15] C.-D. Munz, A tracking method for gas flow into vacuum based on the vacuum Riemann problem, Mathematical methods in the applied sciences 17 (1994) 597-612.

[16] J.R. Goldston and P.H. Rutherford, Introduction to plasma physics, CRC Press (1995)

[17] U.S. Inan and M. Gołkowski, Principles of plasma physics for engineers and scientists, Cambridge University Press, 2010.

[18] H. Sitaraman, Magneto-hydrodynamics simulation study of high density thermal plasmas in plasma acceleration devices, PhD Thesis, University of Texas at Austin (2013)

[19] J.A. Bittencourt, Fundamentals of plasma physics, Springer Science & Business Media, 2013

[20] L. Spitzer and R. Harm, Transport phenomena in a completely ionized gas, Physical Review, vol. 89, no. 5, (1953), 977-981.

[21] M. I. Boulos, P. Fauchais, and E. Pfender, Thermal plasmas: fundamentals and applications, Springer Science & Business Media (2013).

[22] A. Dedner, F. Kemm, D. Kröner, C.D. Munz , T. Schnitzer, and M. Wesenberg, Hyperbolic divergence cleaning for the MHD equations, Journal of Computational Physics, 175(2) (2002) 645-673.

[23] A. Harten, P.D. Lax, and B. van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM review 25.1 (1983) 35-61

[24] E.F. Toro, Riemann solvers and numerical methods for fluid dynamics: a practical introduction, Springer

Science & Business Media (2009)

[25] S.F. Davis, Simplified second-order Godunov-type methods, SIAM Journal on Scientific and Statistical Computing 9.3 (1988) 445-473

[26] A. Haselbacher and J. Blazek, Accurate and efficient discretization of Navier-Stokes equations on mixed grid, *AIAA journal* 38.11 (2000), 2094-2102.

[27] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L.C McInnes, B. Smith, and H. Zhang, PETSc users manual, revision 3.3. Computer Science Division, Argonne National Laboratory, Argonne, IL (2012).

[28] K.T Loebner, T.C. Underwood, T. Mouratidis, and M.A Cappelli, Radial magnetic compression in the expelled jet of a plasma deflagration accelerator, Applied Physics Letters, 108(9), (2016), 094104.

[29] J. D. Anderson, Modern compressible flow: with historical perspective, Vol. 12. New York: McGraw-Hill, (1990)

[30] F.R. Poehlmann, Investigation of a plasma deflagration gun and magnetohydrodynamic Rankine-Hugoniot model to support a unifying theory for electromagnetic plasma guns. Dept. of Mechanical Engineering, Stanford University, PhD Thesis (2010)

[31] D.Y Cheng, Plasma deflagration and properties of a coaxial plasma deflagration gun, Nuclear Fusion, vol. 10, (1970).

[32] U. Shumlak, R. P. Golingo, B.A. Nelson, and D.J. Hartog, Evidence of stabilization in the Z-pinch, Physical Review Letters, 87(20), (2001), 205005.

[33] Y. Saad, Iterative methods for sparse linear systems, Society for Industrial and Applied Mathematics, 2003.

[34] R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H. Van der Vorst, Templates for the solution of linear systems: building blocks for iterative methods, Society for Industrial and Applied Mathematics, 1994.

[35] J.N Shadid, R. P. Pawlowski, J. W. Banks, L. Chacón, P. T. Lin, and R. S. Tuminaro, Towards a scalable fully-implicit fully-coupled resistive MHD formulation with stabilized FE methods, Journal of Computational Physics 229.20 (2010): 7649-7671.

[36] L. Spitzer Jr,and R. Härm, Transport phenomena in a completely ionized gas, Physical Review 89.5 (1953): 977.