A multispecies, multifluid model for laser–induced counterstreaming plasma simulations☆

D. Ghosh [a],[1],[*], T.D. Chapman [a],[2], R.L. Berger [a],[2], A. Dimits [a],[3], J.W. Banks [b]

[a] Lawrence Livermore National Laboratory, Livermore, CA 94550, United States
[b] Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, United States

ABSTRACT

The interpenetration of counterstreaming plasmas is an important phenomenon in several application areas, such as astrophysical flows, design of controlled fusion devices, and laser–induced plasma experiments. Multispecies "single-fluid" codes are unable to model this phenomenon due to the single velocity representation for all the species/fluids. Kinetic codes, though capable of modeling interpenetration, are computationally prohibitive for at-scale simulations. In this paper, we propose a multifluid model that solves the fluid equations for each ion fluid or stream. This allows distinct flows that interact with each other through electrostatic and collisional forces. We introduce and describe our code, EUCLID, that uses a conservative finite-difference formulation to discretize the governing equations in space. The 5th-order Monotonicity-Preserving WENO scheme is used for the upwind approximation of the hyperbolic flux, and the explicit 4th-order Runge–Kutta scheme is used for time integration. The code is verified for several benchmark cases and manufactured solutions. We simulate one- and two-dimensional interactions of counterstreaming plasmas in vacuum as well as in the presence of gas fill, where the setups are representative of laser-induced plasma experiments.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Counterstreaming plasmas often exhibit interpenetration, where they flow through each other, and localized regions may exist where plasma species or populations inhabiting the same physical space may have different velocities. This dynamics plays a significant role in applications such as astrophysical flows, controlled fusion devices [1,2], and experiments involving the interaction of laser-induced plasma streams [3–7]. Multispecies, single-fluid codes [8] fail to capture the interpenetration; the assumption of a single velocity field results in counterstreaming plasmas stagnating and causing density pile-ups. Alternatively, collisional, kinetic codes that solve the Boltzmann equations for each species [9–13] are able to represent this phenomenon; however, they are computationally expensive due to their high-dimensionality, and therefore, impractical for experimental-scale simulations. An effort has also been made to derive species diffusion models [14,15] for multispecies, single-fluid codes. These are empirical in nature, and the introduction of a diffusion term results in a stiff system of partial differential equations (PDEs). The drawbacks of these current approaches motivate the use of a multifluid model.

The multifluid equations are derived by taking the velocity-space moments of the Boltzmann equation for each species [16]. This results in a distinct set of fluid equations for each stream or fluid that is solved along with the Maxwell's equations or some simplification thereof. Over the last few decades, multifluid models have been successfully applied to several plasma applications. In the context of magnetically-confined fusion device simulations, these methods have been applied to θ-pinch [17] and

* Corresponding author.
E-mail address: ghosh5@llnl.gov (D. Ghosh).
[1] Center for Applied Scientific Computing, Computation Directorate
[2] Design Physics, Weapons & Complex Integration Directorate
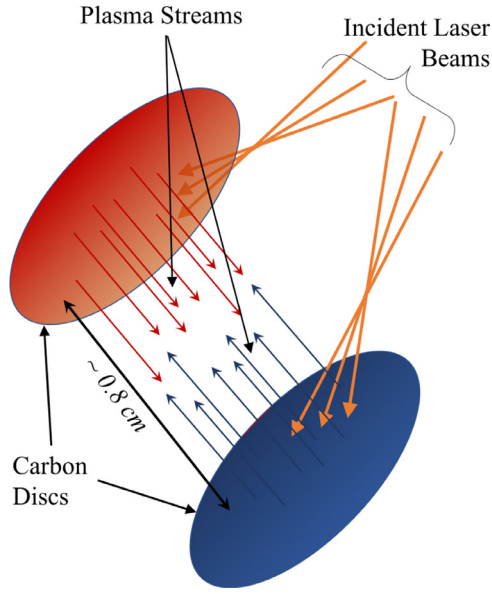[3] Physics Division, Physical & Life Sciences Directorate

Fig. 1. Illustration of the laser-induced plasma interactions in HEDP experiments with counterstreaming plasmas ablating off the carbon discs.

Z-pinch [18,19] simulations. The multifluid approach has been applied to device edge plasma simulations where the modeling of impurities introduced from the device wall is essential [20–25]. In astrophysics, the multifluid approach has been used to study the interactions of the solar wind with the magnetosphere [26] and the local interstellar cloud [22,27], collisionless reconnection [18,19,28–30], and the interactions of weakly/partially-ionized plasmas in the heliosphere and the atmosphere [23,31,32]. Other applications include plasma glow discharge [33,34], ICF capsule detonation [35,36], and plasma-enhanced combustion [37]. A majority of these multifluid approaches use a predefined number of fluids, for example, two-fluid formulations with one ion species and one electron [18,19,25,28,36,38–40], two-fluid formulations with one ion and one neutral species [22,23], and three- and four-fluid formulations with one ion species, one electron species, and neutral species including dust [22,29,32]. Formulations with an arbitrary number of species have also been proposed [21,27,34,41].

This paper focuses on the interactions of laser–induced counterstreaming plasmas in high energy density physics (HEDP) experiments [3–7]. Fig. 1 illustrates a typical setup of such experiments. Two foils/discs made of carbon or a carbon-based compound are irradiated with high-energy laser beams, and the interaction of the counterstreaming plasmas ablating off them is studied. Multifluid models for these applications were proposed [42–44] that solved the fluid equations for two ion species in a one-dimensional domain. A kinetic model was incorporated [42] to represent the friction and heating resulting from plasma instabilities. Overall, these studies showed that a multifluid model was essential to model the interpenetration, and a good agreement with kinetic simulations was observed [44]. In this paper, we present an EUlerian Code for pLasma Interpenetration Dynamics (EUCLID) that extends these previous efforts in developing and implementing a computationally efficient multifluid model for the applications described above. EUCLID solves an electrostatic multispecies, multifluid model that comprises the inviscid Euler equations for each ion fluid or stream, where the number of ion fluids/streams is arbitrary. The fluids interact with each other through the electrostatic forces as well as collisional friction and thermal equilibration terms [44]. We note that in this formulation, each grid cell has

some amount of all fluids. That is to say that there are no interfaces separating different fluids. Furthermore, a given material or plasma species (such as hydrogen or carbon) may be represented by more than one fluid; for example, in 1, the carbon ablated from each disk may be represented by its own distinct fluid. The Debye length in our applications is much smaller than the characteristic length scales (by a factor of the order of 10^5–10^6), and therefore, we use a reduced model for the electrons that simplifies the electrostatic force term. The nondimensionalized governing equations are discretized in a Cartesian domain with the conservative finite-difference formulation [45,46]. The basic data structures in EUCLID are implemented using Chombo [47], a parallel, block-structured adaptive mesh refinement (AMR) library for PDEs, and therefore, EUCLID is multidimensional and supports distributed-memory parallelism.

The primary numerical challenge in our applications stems from flows involving the presence of a fluid that exists only in a localized part of the domain; it doesn't exist elsewhere. Since the fluid density cannot be numerically zero anywhere in the domain (even if other fluids with nonzero densities exist in the same region), we specify the species density corresponding to "numerical vacuum" as 10^{-14}. This results in very steep gradients, where the species density and pressure may transition from $O(1)$ to $O(10^{-14})$ within a few grid cells. Therefore, we use the 5th-order weighted essentially nonoscillatory (WENO) scheme [48] and the characteristic-based reconstruction procedure, which has been shown to be more robust than reconstructing the conserved or the primitive variables for supersonic flows involving strong shocks and gradients. In addition, because the WENO scheme may result in $O(\epsilon)$ oscillations, where ϵ is a WENO parameter that is typically set to 10^{-6}, we implement the monotonicity-preserving (MP) limits [49] to avoid any numerical oscillations. The equations are evolved in time using the four-stage, 4th-order explicit Runge–Kutta (RK4) method.

The primary contributions of this paper are a detailed description of the multifluid model, the assumptions and reformulations that are needed for a robust numerical solution in our application context, and the ingredients of our numerical algorithm. The outline of the paper is as follows. The multifluid model that EUCLID solves is described in Sections 2, and 3 describes the numerical method. Section 4 outlines the code verification using exact solutions, manufactured solutions, and benchmark cases, and the simulation of flows representative of the application areas described above are shown in Section 5. Conclusions are summarized in Section 6.

2. Governing equations

The inviscid Euler equations for each ion fluid are obtained by taking the moments of the collisionless Boltzmann equation and neglecting viscosity and heat conduction [16], and can be expressed in their nondimensional form as:

$$\frac{\partial \rho_\alpha}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = 0, \tag{1a}$$

$$\frac{\partial \rho_\alpha \mathbf{u}_\alpha}{\partial t} + \nabla (\rho_\alpha \mathbf{u}_\alpha \otimes \mathbf{u}_\alpha + P_\alpha) = -Z_\alpha n_\alpha \nabla \phi + \sum_{\beta \neq \alpha} \mathbf{R}_{\alpha,\beta} + \mathbf{R}_{\alpha,e}, \tag{1b}$$

$$\frac{\partial \mathcal{E}_\alpha}{\partial t} + \nabla \cdot \{(\mathcal{E}_\alpha + P_\alpha)\mathbf{u}_\alpha\} = -Z_\alpha n_\alpha \mathbf{u}_\alpha \cdot \nabla \phi + \sum_{\beta \neq \alpha} \left(\mathbf{R}_{\alpha,\beta} \cdot \mathbf{u}_\alpha + Q_{\alpha,\beta} \right) + \mathbf{R}_{\alpha,e} \cdot \mathbf{u}_\alpha + Q_{\alpha,e}, \tag{1c}$$

where $\alpha = 1, \cdots, n_f$ is the fluid index, n_f is the number of fluids, ρ_α is the density, $\mathbf{u}_\alpha$ is the velocity vector, P_α is the pressure, Z_α is

the ionization number, ϕ is the electrostatic potential, $n_\alpha = \rho_\alpha/m_\alpha$ is the number density, m_α is the atomic mass, and

$$\mathcal{E}_\alpha = \frac{P_\alpha}{\gamma_\alpha - 1} + \frac{1}{2}\rho_\alpha \mathbf{u}_\alpha \cdot \mathbf{u}_\alpha \tag{2}$$

is the internal energy. The ideal gas law $P_\alpha = n_\alpha T_\alpha$ is assumed, and the specific heat ratio is specified as $\gamma_\alpha = 5/3$ for all fluids. On the right-hand side (RHS) of (1), the terms $-Z_\alpha n_\alpha \nabla\phi$ and $-Z_\alpha n_\alpha \mathbf{u}_\alpha \cdot \nabla\phi$ represent the electrostatic force and the work done by it, respectively, and the remaining terms represent the collisional interactions between the ion fluids and electrons, as described below. We reiterate here that (1) is solved for each "fluid" or distinct population of ion species; a single species can be partitioned into multiple populations and modeled as separate fluids.

Friction and thermal equilibration terms. The frictional force on fluid α due to fluid β is proportional to the velocity of β relative to that of α, and this can be expressed as

$$\mathbf{R}_{\alpha,\beta} = m_\alpha n_\alpha \nu_{\alpha,\beta}\left(\mathbf{u}_\beta - \mathbf{u}_\alpha\right) = -\mathbf{R}_{\beta,\alpha}, \tag{3}$$

and the total heating of fluid α due to fluid β is

$$Q_{\alpha,\beta} = Q_{\alpha,\beta}^{\text{fric}} + Q_{\alpha,\beta}^{\text{eq}}, \tag{4}$$

where

$$Q_{\alpha,\beta}^{\text{fric}} = m_{\alpha,\beta} n_\alpha \nu_{\alpha,\beta}\left(\mathbf{u}_\alpha - \mathbf{u}_\beta\right) \cdot \left(\mathbf{u}_\alpha - \mathbf{u}_\beta\right) \tag{5}$$

is the frictional heating between two fluids that is proportional to the square of the relative velocity, and

$$Q_{\alpha,\beta}^{\text{eq}} = 3 m_\alpha n_\alpha \frac{\nu_{\alpha,\beta}}{m_\alpha + m_\beta}\left(T_\beta - T_\alpha\right) = -Q_{\beta,\alpha}^{\text{eq}} \tag{6}$$

is the thermal equilibration due to a temperature difference between the two fluids. The collisional coefficient $\nu_{\alpha,\beta}$ is given by [44]

$$\nu_{\alpha,\beta} = \frac{4\sqrt{2\pi}}{3} \frac{Z_\alpha^2 Z_\beta^2 e^4 n_\beta \Lambda_{\alpha,\beta}}{m_\alpha m_{\alpha,\beta}}$$
$$\times \left[r\left(\mathbf{u}_\alpha - \mathbf{u}_\beta\right)\cdot\left(\mathbf{u}_\alpha - \mathbf{u}_\beta\right) + \frac{T_\alpha}{m_\alpha} + \frac{T_\beta}{m_\beta}\right]^{-\frac{3}{2}}\left(\frac{n_{\text{ref}}x_{\text{ref}}}{T_{\text{ref}}^2}\right), \tag{7}$$

where $m_{\alpha,\beta} = m_\alpha m_\beta/\left(m_\alpha + m_\beta\right)$. Similarly, the electron–ion friction and thermal equilibration terms are

$$\mathbf{R}_{\alpha,e} = -\mathbf{R}_{e,\alpha} = -m_e n_e \nu_{e,\alpha}(\mathbf{u}_\alpha - \mathbf{u}_e), \tag{8a}$$

$$Q_{\alpha,e} = Q_{\alpha,e}^{\text{eq}} = -Q_{e,\alpha}^{\text{eq}} = -3 m_e n_e \frac{\nu_{e,\alpha}}{m_\alpha + m_e}(T_\alpha - T_e), \tag{8b}$$

where the electron-ion friction heating $Q_{\alpha,e}^{\text{fric}}$ has been ignored. Since $m_e \ll m_\alpha \Rightarrow m_{\alpha,e} \approx m_e$, we have

$$\nu_{e,\alpha} = \frac{4\sqrt{2\pi}}{3} \frac{Z_\alpha^2 e^4 n_\alpha \Lambda_{ei}}{m_e^2 (T_e/m_e)^{\frac{3}{2}}}\left(\frac{n_{\text{ref}}x_{\text{ref}}}{T_{\text{ref}}^2}\right). \tag{9}$$

as the simplified form for the electron–ion collisional coefficient [44]. In the simulations reported in this paper, the ion–ion log-lambda is $\Lambda_{\alpha,\beta} = 5$ and the electron–ion log-lambda is $\Lambda_{e,\alpha} = 10$, unless otherwise mentioned, and $r = [2/(9\pi)]^{1/3}$ is a constant.

Eq. (1) is expressed in terms of nondimensional variables, and these can be multiplied with the corresponding reference quantity to obtain their physical values. The primary reference quantities and their CGS units are mass m_{ref} (g), number density n_{ref} (cm^{-3}), length x_{ref} (cm), temperature T_{ref} (ergs), and charge e (elementary charge in statcoulomb). The derived reference quantities and their units are:

$$\begin{aligned}
u_{\text{ref}} &= \sqrt{T_{\text{ref}}/m_{\text{ref}}} & \text{(cm/s)}, \\
\rho_{\text{ref}} &= m_{\text{ref}} n_{\text{ref}} & \text{(g/cm}^3\text{)}, \\
t_{\text{ref}} &= x_{\text{ref}}/u_{\text{ref}} = x_{\text{ref}}\sqrt{m_{\text{ref}}/T_{\text{ref}}} & \text{(s)}, \\
P_{\text{ref}} &= n_{\text{ref}} T_{\text{ref}} & \text{(Ba)}, \\
\phi_{\text{ref}} &= T_{\text{ref}}/e & \text{(V)}.
\end{aligned} \tag{10}$$

Electron Model. The fluid equations for electrons can be expressed by removing the $\mathbf{R}_{\alpha,e}$ and $Q_{\alpha,e}$ terms in (1) and replacing the subscript α with the subscript e in the remaining terms, and these equations can be solved along with the ion fluid equations to resolve the electron dynamics, where the Poisson's equation is solved for the electrostatic potential. However, to avoid the stiff time scales arising from the electron thermal velocity (for example, the electron plasma frequency), we make the following assumptions:

$$\frac{\partial \rho_e \mathbf{u}_e}{\partial t} + \nabla(\rho_e \mathbf{u}_e \otimes \mathbf{u}_e) \to 0 \Rightarrow \nabla P_e = n_e \nabla\phi + \sum_\alpha \mathbf{R}_{e,\alpha}, \tag{11a}$$

$$P_e = n_e T_e \qquad (T_e \text{ constant}), \tag{11b}$$

$$n_e = \sum_\alpha n_\alpha Z_\alpha, \qquad \mathbf{u}_e = \frac{1}{n_e}\sum_\alpha Z_\alpha n_\alpha \mathbf{u}_\alpha. \tag{11c}$$

Eq. (11a) and (11b) assume that the electrons are inertialess and isothermal, respectively. Eq. (11c) assumes that the plasma is quasineutral with zero currents, which is a consequence of the Debye lengths being much smaller than the length scales of interest in our applications. Thus, we do not resolve electron dynamics, and our governing equations are deliberately limited to applications without significant charge separation between electrons and ions. Since the electrons are assumed to be isothermal, they act as "infinite" sources or sinks for the ion energy; consequently, the model is not energy-conservative, although this can easily be changed. We will explore higher-fidelity electron models in future publications. The electric field can be obtained by expressing the electron pressure in (11a) as (11b) and rearranging:

$$\nabla\phi = \frac{T_e}{n_e}\nabla n_e - \frac{1}{n_e}\sum_\alpha \mathbf{R}_{e,\alpha}. \tag{12}$$

Substituting (12) in (1) and adding $\nabla(Z_\alpha T_e n_\alpha)$ and $\nabla\cdot(Z_\alpha T_e n_\alpha \mathbf{u}_\alpha)$ on both sides of (1b) and (1c), respectively, the equations for the ion fluids are:

$$\frac{\partial \rho_\alpha}{\partial t} + \nabla\cdot(\rho_\alpha \mathbf{u}_\alpha) = 0, \tag{13a}$$

$$\begin{aligned}
\frac{\partial \rho_\alpha \mathbf{u}_\alpha}{\partial t} + \nabla(\rho_\alpha \mathbf{u}_\alpha \otimes \mathbf{u}_\alpha + P_\alpha^*) &= \frac{Z_\alpha T_e}{n_e}[n_e \nabla n_\alpha - n_\alpha \nabla n_e] \\
&+ \frac{Z_\alpha n_\alpha}{n_e}\sum_\beta \mathbf{R}_{e,\beta} \\
&+ \mathbf{R}_{\alpha,e} + \sum_{\beta\neq\alpha}\mathbf{R}_{\alpha,\beta},
\end{aligned} \tag{13b}$$

$$\begin{aligned}
\frac{\partial \mathcal{E}_\alpha}{\partial t} + \nabla\cdot\{(\mathcal{E}_\alpha + P_\alpha^*)\mathbf{u}_\alpha\} &= \frac{Z_\alpha T_e}{n_e}[n_e\nabla(\mathbf{u}_\alpha n_\alpha) - (\mathbf{u}_\alpha n_\alpha)\nabla n_e] \\
&+ \frac{Z_\alpha n_\alpha}{n_e}\sum_\beta \mathbf{u}_\alpha \cdot \mathbf{R}_{e,\beta} \\
&+ \sum_{\beta\neq\alpha}\left(\mathbf{R}_{\alpha,\beta}\cdot\mathbf{u}_\alpha + Q_{\alpha,\beta}\right) \\
&+ \mathbf{R}_{\alpha,e}\cdot\mathbf{u}_\alpha + Q_{\alpha,e},
\end{aligned} \tag{13c}$$

where

$$P_\alpha^* = P_\alpha + Z_\alpha T_e n_\alpha \tag{14}$$

is the augmented pressure, and $Z_\alpha T_e n_\alpha$ is the "electron pressure". The main motivation behind including the electron pressure on the left-hand-side (LHS) of (13) is to allow its inclusion in the upwinding of the spatial discretization operator described in the next section. The electrostatic force acting through this term is advective in nature, and therefore it should be included along with the hydrodynamic pressure P_α and the convective velocity $\mathbf{u}_\alpha$ in the calculation of the wavespeeds.

Thus, to summarize, (13) with $\alpha = 1, \cdots, n_f$ represents the complete system of equations solved by EUCLID. We note that (13) reduces to the inviscid, compressible Euler equations governing neutral gas dynamics [50] for $n_f = 1, Z_\alpha = 0$.

3. Numerical method

Eq. (13) constitutes a system of N hyperbolic PDEs with source terms that can be expressed as

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{S}(\mathbf{U}), \tag{15}$$

where

$$\mathbf{U} \equiv \begin{bmatrix} \mathbf{U_1}, & \cdots, & \mathbf{U}_\alpha, & \cdots, & \mathbf{U}_{n_f} \end{bmatrix}^\mathrm{T},$$

$$\mathbf{F}(\mathbf{U}) \equiv \begin{bmatrix} \mathbf{F_1}(\mathbf{U_1}), & \cdots, & \mathbf{F}_\alpha(\mathbf{U}_\alpha), & \cdots, & \mathbf{F}_{n_f}(\mathbf{U}_{n_f}) \end{bmatrix}^\mathrm{T},$$

$$\mathbf{S}(\mathbf{U}) \equiv \begin{bmatrix} \mathbf{S_1}(\mathbf{U}), & \cdots, & \mathbf{S}_\alpha(\mathbf{U}), & \cdots, & \mathbf{S}_{n_f}(\mathbf{U}) \end{bmatrix}^\mathrm{T},$$

are the state vector, flux tensor, and source vector for all the fluids with $N = 5n_f$ components (5 components for each set of three-dimensional fluid equations), and

$$\mathbf{U}_\alpha(\mathbf{U}_\alpha) = \begin{bmatrix} \rho_\alpha, & \rho_\alpha \mathbf{u}_\alpha, & \mathcal{E}_\alpha \end{bmatrix}^\mathrm{T}, \tag{16a}$$

$$\mathbf{F}_\alpha(\mathbf{U}_\alpha) = \begin{bmatrix} \rho_\alpha \mathbf{u}_\alpha, & \rho_\alpha \mathbf{u}_\alpha \mathbf{u}_\alpha + P_\alpha^*, & (\mathcal{E}_\alpha + P_\alpha^*)\mathbf{u}_\alpha \end{bmatrix}^\mathrm{T} \tag{16b}$$

are the state vector and flux tensor for fluid α, respectively, and

$$\mathbf{S}_\alpha(\mathbf{U}) = \mathbf{S}_\alpha^\mathrm{elec}(\mathbf{U}) + \mathbf{S}_{\alpha,ii}^\mathrm{coll}(\mathbf{U}) + \mathbf{S}_{\alpha,ei}^\mathrm{coll}(\mathbf{U}),$$

is its source term comprising the electrostatic and collisional terms:

$$\mathbf{S}_\alpha^\mathrm{elec}(\mathbf{U})$$
$$= \frac{Z_\alpha T_e}{n_e} \begin{bmatrix} 0, & n_e \nabla n_\alpha - n_\alpha \nabla n_e, & n_e \nabla \cdot (n_\alpha \mathbf{u}_\alpha) - n_\alpha \mathbf{u}_\alpha \cdot \nabla n_e \end{bmatrix}^\mathrm{T}, \tag{17a}$$

$$\mathbf{S}_{\alpha,ii}^\mathrm{coll}(\mathbf{U}) = \sum_{\beta \neq \alpha} m_\alpha n_\alpha \nu_{\alpha,\beta}$$
$$\times \begin{bmatrix} 0 \\ (\mathbf{u}_\beta - \mathbf{u}_\alpha) \\ (\mathbf{u}_\beta - \mathbf{u}_\alpha) \cdot \mathbf{u}_\alpha + \frac{m_{\alpha,\beta}(\mathbf{u}_\beta - \mathbf{u}_\alpha)^2}{m_\alpha} + \frac{3(T_\beta - T_\alpha)}{m_\alpha + m_\beta} \end{bmatrix}, \tag{17b}$$

$$\mathbf{S}_{\alpha,ei}^\mathrm{coll}(\mathbf{U}) = m_e n_e$$
$$\begin{bmatrix} 0 \\ \nu_{e,\alpha}(\mathbf{u}_e - \mathbf{u}_\alpha) + \frac{Z_\alpha n_\alpha}{n_e} \sum_\beta \nu_{e,\beta}(\mathbf{u}_\beta - \mathbf{u}_e) \\ \left\{ \nu_{e,\alpha}(\mathbf{u}_e - \mathbf{u}_\alpha) + \frac{Z_\alpha n_\alpha}{n_e} \sum_\beta \nu_{e,\beta}(\mathbf{u}_\beta - \mathbf{u}_e) \right\} \cdot \mathbf{u}_\alpha + \frac{3\nu_{e,\alpha}(T_e - T_\alpha)}{m_\alpha + m_e} \end{bmatrix}. \tag{17c}$$

We note that the flux tensor of fluid α depends only on the state vector of fluid α, while the source term for each fluid is a function of the other fluids as well since it contains the inter-fluid interactions. In subsequent discussions, we refer to the following equation:

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{S}^\mathrm{elec}(\mathbf{U}), \quad \mathbf{S}^\mathrm{elec}(\mathbf{U}) \equiv \begin{bmatrix} \mathbf{S}_\alpha^\mathrm{elec}(\mathbf{U}) \big| \alpha = 1, \cdots, n_f \end{bmatrix}^\mathrm{T} \tag{18}$$

as the "collisionless" multifluid equations, where the collisional source terms have been excluded.

Spatial discretization. Eq. (15) is discretized on a three-dimensional Cartesian grid, with $\mathbf{i} \equiv \{i, j, k\}$ as the grid index, and the conservative finite-difference formulation [45,46] can be written at a grid point as:

$$\frac{d\mathbf{U_i}}{dt} = \mathbf{L}(\mathbf{U_i}) \equiv -\sum_{d=1}^{3} \left(\frac{\hat{\mathbf{f}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(d)} - \hat{\mathbf{f}}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}^{(d)}}{\Delta x} \right) + \mathbf{S_i}, \tag{19}$$

where $\mathbf{U_i}$ and $\mathbf{S_i} \equiv \mathbf{S}(\mathbf{U_i})$ are the cell-centered values of the state and source vectors, d is the dimension index, and $\mathbf{e}_d$ is the unit vector along d (its d-th component is 1 and the other components are 0). The grid spacing, Δx, is taken as uniform in all dimensions in our implementation. The high-order approximation to the flux primitive, $\hat{\mathbf{f}}^{(d)}$, is

$$\hat{\mathbf{f}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(d)} = \mathcal{S}\left(\left\{ \mathbf{f}_{\mathbf{i}+k\mathbf{e}_d}^{(d)} \bigg| - w_L \leq k \leq w_R \right\} \right) = \mathbf{h}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(d)} + \mathcal{O}(\Delta x^p), \tag{20}$$

where p is the order of the discretization scheme, $\mathcal{S}$ is the finite-difference reconstruction operator, w_L and w_R define its stencil width, and $\mathbf{h}^{(d)}$ is the flux primitive satisfying:

$$\mathbf{f}^{(d)}(\mathbf{x}) \equiv \mathbf{f}^{(d)}(\mathbf{U}(\mathbf{x})) = \frac{1}{\Delta x} \int_{x_d - \frac{\Delta x}{2}}^{x_d + \frac{\Delta x}{2}} \mathbf{h}^{(d)}(\boldsymbol{\xi}) d\boldsymbol{\xi} \tag{21}$$

and $\mathbf{f}^{(d)}(\mathbf{U}) \equiv \mathbf{F}(\mathbf{U}) \cdot \mathbf{e}_d$ is the d-th component of the flux tensor. The numerical flux approximation $\hat{\mathbf{f}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(d)}$ is computed at the cell interfaces $\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d$ independently along each dimension; the following discussion describes this for one spatial dimension.

The vector flux approximation is reconstructed at the cell interface using a characteristic-based approach [45,46,48], where it is computed as:

$$\hat{\mathbf{f}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \left[R^{(d)}\left(\mathbf{U}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^\mathrm{Roe} \right) \right] \boldsymbol{\varphi}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}, \tag{22}$$

where $R^{(d)}$ is the matrix whose columns are the right-eigenvectors of the flux Jacobian evaluated at the Roe-averaged state $\mathbf{U}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^\mathrm{Roe}$ [50]. The Roe-fixed scheme [48] is used to compute the components of the characteristic flux vector, $\boldsymbol{\varphi}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$, at the cell interfaces:

$$\varphi_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(k)} = \begin{cases} \varphi_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(k),\mathrm{L}} & \text{if } \lambda_{k,\mathbf{i}}^{(d)}, \lambda_{k,\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(d)}, \lambda_{k,\mathbf{i}+\mathbf{e}_d}^{(d)} > 0 \\ \varphi_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(k),\mathrm{R}} & \text{if } \lambda_{k,\mathbf{i}}^{(d)}, \lambda_{k,\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(d)}, \lambda_{k,\mathbf{i}+\mathbf{e}_d}^{(d)} < 0 \\ \frac{1}{2}\left[\varphi_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(k),\mathrm{L}} + \varphi_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(k),\mathrm{R}} - \lambda_{k,\max}^{(d)}\left(\varrho_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(k),\mathrm{R}} - \varrho_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(k),\mathrm{L}} \right) \right] & \text{otherwise} \end{cases}, \tag{23}$$

$$\boldsymbol{\varphi}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \begin{bmatrix} \varphi_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(k)} \Big| k = 1, \cdots, N \end{bmatrix},$$

where $\lambda_{k,\{\mathbf{i},\mathbf{i}+\mathbf{e}_d\}}^{(d)}$ are the k-th eigenvalue (characteristic wavespeed) along dimension d evaluated at the cell-centered solution $\mathbf{U}_{\{\mathbf{i},\mathbf{i}+\mathbf{e}_d\}}$, $\lambda_{k,\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(d)}$ is the k-th eigenvalue evaluated at the Roe-averaged state $\mathbf{U}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^\mathrm{Roe}$. The left- and right-biased characteristic flux and solution components at the cell interfaces are computed as:

$$\varphi_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(k),\mathrm{L}} = \mathrm{MPWENO5}\left(\varphi_{\mathbf{i}-2\mathbf{e}_d}^{(k)}, \varphi_{\mathbf{i}-\mathbf{e}_d}^{(k)}, \varphi_{\mathbf{i}}^{(k)}, \varphi_{\mathbf{i}+\mathbf{e}_d}^{(k)}, \varphi_{\mathbf{i}+2\mathbf{e}_d}^{(k)} \right), \tag{24a}$$

$$\varphi_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(k),\mathrm{R}} = \mathrm{MPWENO5}\left(\varphi_{\mathbf{i}+3\mathbf{e}_d}^{(k)}, \varphi_{\mathbf{i}+2\mathbf{e}_d}^{(k)}, \varphi_{\mathbf{i}+\mathbf{e}_d}^{(k)}, \varphi_{\mathbf{i}}^{(k)}, \varphi_{\mathbf{i}-\mathbf{e}_d}^{(k)} \right), \tag{24b}$$

$$\varrho_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(k),\mathrm{L}} = \mathrm{MPWENO5}\left(\varrho_{\mathbf{i}-2\mathbf{e}_d}^{(k)}, \varrho_{\mathbf{i}-\mathbf{e}_d}^{(k)}, \varrho_{\mathbf{i}}^{(k)}, \varrho_{\mathbf{i}+\mathbf{e}_d}^{(k)}, \varrho_{\mathbf{i}+2\mathbf{e}_d}^{(k)}\right), \tag{24c}$$

$$\varrho_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{(k),\mathrm{R}} = \mathrm{MPWENO5}\left(\varrho_{\mathbf{i}+3\mathbf{e}_d}^{(k)}, \varrho_{\mathbf{i}+2\mathbf{e}_d}^{(k)}, \varrho_{\mathbf{i}+\mathbf{e}_d}^{(k)}, \varrho_{\mathbf{i}}^{(k)}, \varrho_{\mathbf{i}-\mathbf{e}_d}^{(k)}\right), \tag{24d}$$

where $\varphi_{(\cdot)}^{(k)}, \varrho_{(\cdot)}^{(k)}$ represent the k-th component of the cell-centered characteristic flux and solution vectors $\boldsymbol{\varphi}_{(\cdot)}, \boldsymbol{\varrho}_{(\cdot)}$, respectively, and MPWENO5$(\cdot)$ is the 5th-order monotonicity-preserving WENO scheme [48,49] summarized in Appendix A. The cell-centered characteristic flux and solution vectors are calculated as

$$\boldsymbol{\varphi}_{(\cdot)} = \left[L^{(d)}\left(\mathbf{U}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{\mathrm{Roe}}\right)\right]\mathbf{f}_{(\cdot)}, \qquad \boldsymbol{\varrho}_{(\cdot)} = \left[L^{(d)}\left(\mathbf{U}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{\mathrm{Roe}}\right)\right]\mathbf{U}_{(\cdot)}, \tag{25}$$

where $L^{(d)}\left(\mathbf{U}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{\mathrm{Roe}}\right)$ is the matrix whose rows are the left-eigenvectors of the flux Jacobian evaluated at the Roe-averaged state $\mathbf{U}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{\mathrm{Roe}}$.

In the above method, the Roe-averaged state is computed independently for each fluid at the cell interfaces, i.e., the state vectors of each fluid $\mathbf{U}_\alpha$ is extracted from $\mathbf{U}$, and the Roe-averaged state is calculated. The complete eigenstructure ($\lambda^{(d)}$, $L^{(d)}$, $R^{(d)}$; $d \in \{x, y, z\}$) for the multifluid system (15) required by the characteristic-based procedure outlined above is provided in Appendix B. We reiterate here that the contribution of the electron pressure, included on the LHS of (13), is incorporated into the derivation of the characteristic wavespeeds, and consequently, the upwind discretization. Thus, to summarize, the following are the steps in the discretization of the hyperbolic flux term in (15) along each dimension d:

1. At each cell interface $\mathbf{i}+\frac{1}{2}\mathbf{e}_d$, compute the Roe-averaged state $\mathbf{U}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{\mathrm{Roe}}$ [50], its eigenvalues $\lambda^{(d)}\left(\mathbf{U}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{\mathrm{Roe}}\right)$, and its left and right eigenvector matrices $L^{(d)}\left(\mathbf{U}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{\mathrm{Roe}}\right), R^{(d)}\left(\mathbf{U}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{\mathrm{Roe}}\right)$ using the expressions given in Appendix B.
2. Compute the characteristic flux and solution vectors $\boldsymbol{\varphi}_{(\cdot)}, \boldsymbol{\varrho}_{(\cdot)}$ at all the grid points needed by the MPWENO5 operator as (25).
3. Reconstruct the left- and right-biased characteristic flux and solution vectors at the cell interface $\mathbf{i}+\frac{1}{2}\mathbf{e}_d$ as given by (24) using the MPWENO5 scheme in Appendix A.
4. Apply the Roe-fixed upwinding (23) procedure to compute the upwind characteristic flux vector at the interface $\boldsymbol{\varphi}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$, and then compute the conserved flux vector using (22).

The source term in (15) comprises the electrostatic and collisional source terms, as given in (17). While the collisional source term does not contain any derivatives, the electrostatic source term (17a) contains gradient and divergence terms. These are computed using the fourth-order central difference operator along each dimension, which can be expressed for an arbitrary grid variable ψ as:

$$\left.\frac{\partial \psi}{\partial x^{(d)}}\right|_{\mathbf{i}} = \frac{1}{12\Delta x}\left(\psi_{\mathbf{i}-2\mathbf{e}_d} - 8\psi_{\mathbf{i}-\mathbf{e}_d} + 8\psi_{\mathbf{i}+\mathbf{e}_d} - \psi_{\mathbf{i}+2\mathbf{e}_d}\right) + \mathcal{O}\left(\Delta x^4\right). \tag{26}$$

This differencing is applied component-wise for vector quantities.

Time integration. Eq. (19) is a ordinary differential equation (ODE) in time, and it is integrated using the fourth-order, four-stage, explicit Runge–Kutta method. Since the time integration method is explicit, the time step Δt is restricted by the CFL criterion, where the CFL number is defined as

$$\sigma = \Delta t \frac{\max_{\Omega,\alpha}(|\mathbf{u}_\alpha| + c_\alpha^*)}{\Delta x}, \tag{27}$$

where Ω is the total domain, α is the fluid index, and $c_\alpha^* = \sqrt{\gamma_\alpha P_\alpha^*/\rho_\alpha}$ is the "augmented" speed of sound that is computed from the augmented pressure in (14). Linear stability typically requires $\sigma < 1$, although this is specific to the spatial discretization method, and it is not a sufficient condition for stability for nonlinear simulations. Eq. (27) does not take into account the time scales of the collisional source terms, (17b) and (17c); consequently, a lower time step that resolves the collisional time scales is needed, and this is reported in the simulations presented in subsequent sections.

4. Verification

EUCLID is verified through several benchmark test problems. Since (13) reduces to the inviscid Euler equations for gasdynamics for $Z_\alpha = 0, n_f = 1$), standard compressible flow test cases are solved and the results verified. Since few benchmark problems exist for our specific applications, we verify our code for charged fluids using the following approach. An analytical, easily-differentiable form is assumed for the solution $\mathbf{U}$, and (19) is modified as

$$\frac{d\mathbf{U_i}}{dt} = \mathbf{L}(\mathbf{U_i}) - \mathbf{L}_{\mathrm{ex}}(\mathbf{U_i}), \tag{28}$$

where $\mathbf{L}_{\mathrm{ex}}(\mathbf{U_i})$ is the exact form of $\mathbf{L}(\mathbf{U_i})$ obtained by analytically evaluating its constituent terms. We then verify that the numerical error in computing $\mathbf{L}(\mathbf{U})$, quantified as

$$\varepsilon = \|\mathbf{L}(\mathbf{U}) - \mathbf{L}_{\mathrm{ex}}(\mathbf{U})\|, \tag{29}$$

converges to zero at the theoretical order of the spatial discretization scheme, where $\|(\cdot)\|$ is an appropriate norm over the entire grid. In all the cases described in this section, we take the discrete 2-norm $\|(\cdot)\|_2$. The analytical solutions used here are the ones that could be derived by Mathematica [51] in a reasonable time while testing as many terms in the governing equations as possible. EUCLID is implemented for the three-dimensional equations; lower-dimensional flows are simulated by specifying a small number of grid points (the minimum required by the finite-difference stencil) and periodic boundary conditions along the inactive dimension(s). These tests are repeated along each dimension to verify the code; for example, a two-dimensional test is repeated in the x–y, y–z, and x–z planes.

4.1. Collisionless multifluid equations

The collisionless multifluid equations are given by (18), and the verification tests in this section demonstrate the accuracy, convergence, and nonoscillatory properties of our algorithm.

Accuracy and convergence. Our implementation is verified for a single charged fluid ($n_f = 1$) by generating a two-dimensional solution that results in all terms in (18) being nonzero (except those along the third dimension). The governing Eqs. (13), contain terms that are either uni-dimensional (for example, ρu, ρv, ρu^2, etc.) or bi-dimensional (for example, ρuv, ρuw, ρvw, etc.), and therefore, a two-dimensional test applied in all three planes (x–y, y–z, and x–z) is sufficient. Since the number of fluids is 1, the fluid index subscript is omitted. The species is set to helium ($m = 4, Z = 2$), and the solution is prescribed as

$$\rho(x, y) = 1 + \frac{1}{5}\cos(2\pi x)\cos(2\pi y), \tag{30a}$$

$$u(x, y) = \frac{1}{2}\cos(2\pi x)\cos(4\pi y), \tag{30b}$$

$$v(x, y) = \frac{1}{4}\cos(4\pi x)\cos(2\pi y), \tag{30c}$$

(a) Single-fluid solution
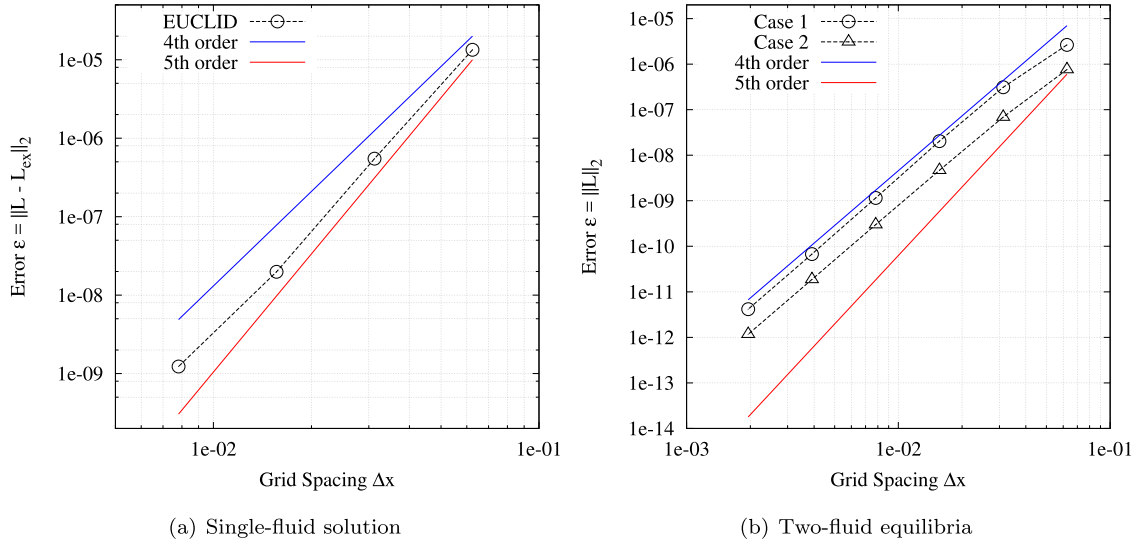
(b) Two-fluid equilibria

Fig. 2. Error vs. grid spacing for (a) a two-dimensional solution with a single fluid, and (b) two two-dimensional 2-fluid equilibria. The order of convergence is at least 4 in both cases; the order is closer to 5 for the single-fluid case because the electrostatic source terms, which are discretized using a 4th order discretization, are very small for this case, and the flux divergence term, discretized by the 5th-order MPWENO scheme, is dominant.

$$w(x, y) = 0, \tag{30d}$$

$$P(x, y) = 2 + \frac{1}{4} \cos(2\pi x) \cos(2\pi y). \tag{30e}$$

The electron temperature T_e is 1, and the specific heat ratio is taken as $\gamma = 5/3$. The analytical term $\mathbf{L}_{ex}(\mathbf{U_i})$ is obtained using Mathematica. A periodic, two-dimensional domain with unit length along x and y is considered. Fig. 2(a) shows the error as a function of the grid spacing Δx for the number of grid points varying from 16×16 to 128×128; the number of grid points is the same along x and y for all the cases. The time step is specified as $\Delta t = 0.001$. The order convergence is observed to be greater than 4; the solutions are almost 5th order, except between the two finest grids (64×64 and 128×128). Examining the electrostatic source term, (17a), we note that $n_e \equiv \sum Z_\alpha n_\alpha = Zn$ for $n_f = 1$, and thus it reduces to

$$\mathbf{S}^{elec}(\mathbf{U}) = \begin{bmatrix} 0, & \mathbf{0}, & ZT_e n \nabla \cdot \mathbf{u} \end{bmatrix}^T. \tag{31}$$

Thus, the error is dominated by the flux-divergence term $\nabla \cdot \mathbf{F}$ in (18), which is discretized using the 5th-order MPWENO scheme described in the previous section.

We now consider a two-fluid case that verifies the implementation of terms that involve multiple fluids. Smooth, two-dimensional equilibria for the two-fluid equations ($n_f = 2$) are derived as follows. The two fluids are two distinct species, and the initial number densities are specified functions while the initial velocities are zero for both the fluids:

$$P_1^{(0)}(x, y) = \begin{cases} -2\cos(2\pi x)\cos(4\pi y) + 3\log[5 + 2\cos(2\pi x)\cos(4\pi y)] & \text{Case 1} \\ -2[-5 + \log\{10 + 3\cos(4\pi x)\cos(2\pi y)\}] & \text{Case 2} \end{cases}, \tag{35a}$$

$$P_2^{(0)}(x, y) = \begin{cases} 10 - 3\log[5 + 2\cos(2\pi x)\cos(4\pi y)] & \text{Case 1} \\ \frac{1}{2}[-3\cos(4\pi x)\cos(2\pi y)] + 2\log[10 + 3\cos(4\pi x)\cos(2\pi y)] & \text{Case 2} \end{cases}. \tag{35b}$$

$$n_{1,2}(x, y, t = 0) \equiv n_{1,2}^{(0)}(x, y), \quad \mathbf{u}_{1,2}^{(0)} = 0, \tag{32}$$

The pressures $P_{1,2}(x, y, t = 0) \equiv P_{1,2}^{(0)}(x, y)$ are derived using Mathematica such that they satisfy the equilibrium conditions:

$$\begin{bmatrix} \partial_x \left(P_\alpha^{(0)} + Z_\alpha T_e n_\alpha^{(0)} \right) \\ \partial_y \left(P_\alpha^{(0)} + Z_\alpha T_e n_\alpha^{(0)} \right) \end{bmatrix} = \frac{Z_\alpha T_e}{n_e^{(0)}} \begin{bmatrix} n_e^{(0)} \partial_x n_\alpha^{(0)} - n_\alpha^{(0)} \partial_x n_e^{(0)} \\ n_e^{(0)} \partial_y n_\alpha^{(0)} - n_\alpha^{(0)} \partial_y n_e^{(0)} \end{bmatrix}, \quad \alpha = 1, 2, \tag{33}$$

where $n_e^{(0)} = Z_1 n_1^{(0)} + Z_2 n_2^{(0)}$. Derivation of the pressure functions involves solving the PDE given by (33), and the constants of integration are taken such that the resulting functions have no nonpositive values. We consider two cases, where the species are helium ($Z_1 = 2, m_1 = 4$) and carbon ($Z_2 = 6, m_2 = 12$), the domain is $x, y \in [.0, 1)$. with periodic boundary conditions, and the number densities are

$$n_1^{(0)}(x, y) = \begin{cases} 1 + \cos(2\pi x)\cos(4\pi y) & \text{Case 1} \\ 1 & \text{Case 2} \end{cases}, \tag{34a}$$

$$n_2^{(0)}(x, y) = \begin{cases} \frac{1}{2} & \text{Case 1} \\ \frac{1}{2} + \frac{1}{4}\cos(4\pi x)\cos(2\pi y) & \text{Case 2} \end{cases}. \tag{34b}$$

We were unable to obtain real-valued pressure functions for a case where the number densities for both the fluids varied spatially, and therefore, these two cases, where one of the number densities is constant, are necessary to verify our implementation. The electron temperature T_e and specific heat ratios for both fluids are 1 and 5/3, respectively. The pressure functions that satisfy (33) are

Since these are equilibrium cases, $\mathbf{L}_{ex}(\mathbf{U_i}) \equiv \mathbf{0}$, and the error can be measured by advancing (19) by one time step with the forward
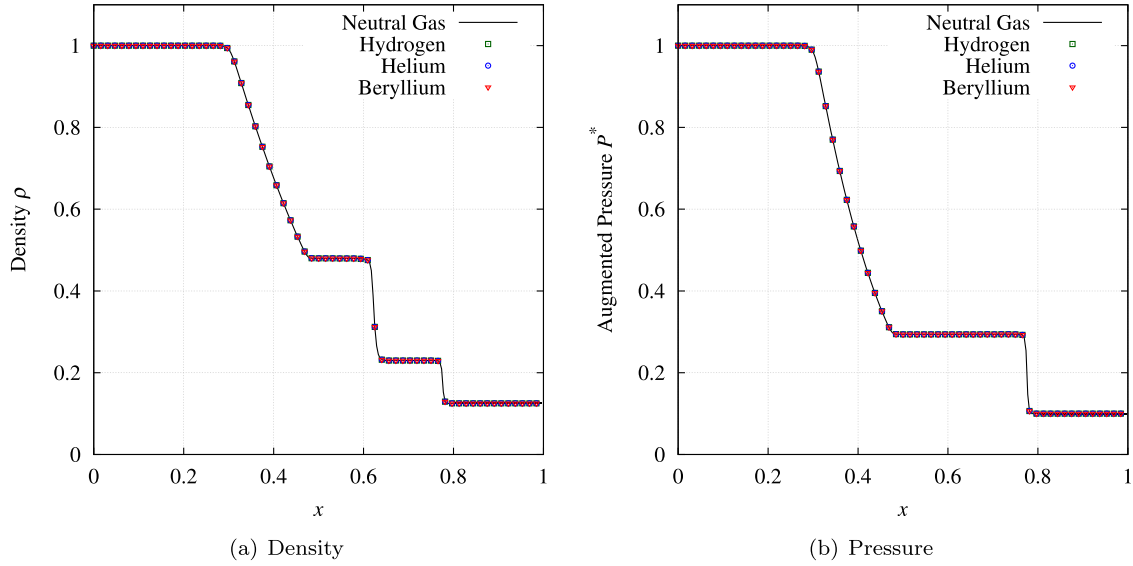
(a) Density

(b) Pressure

Fig. 3. Solution of the electrostatic shock tube problem: The solutions for the charged species (hydrogen, helium, and beryllium) agree well with the neutral gas solution.

Euler scheme:

$$\varepsilon = \|\mathbf{L}(\mathbf{U})\|_2 = \frac{1}{\Delta t}\|\mathbf{U}^{n+1} - \mathbf{U}^n\|_2. \tag{36}$$

Fig. 2(b) shows the error as a function of the grid spacing Δx, starting from a grid with 16×16 points to a grid with 256×256 points. The time step is specified as $\Delta t = 0.00001$. Since the electrostatic source term (17a) has the same magnitude as the flux divergence term $\nabla \cdot \mathbf{F}$, 4th-order convergence is observed.

The two tests discussed above are sufficient to verify our discretization of the multifluid Eqs. (13), except the collisional terms, and demonstrate the high-order convergence of the spatial discretization schemes. The single-fluid test verifies the discretization of the flux term, while the two-fluid cases verify the discretization of the electrostatic source term. These tests are repeated in the y–z and x–z planes to complete the verification of all non-collisional terms in (13).

Nonoscillatory behavior. The Sod shock tube test [50,52] is a one-dimensional Riemann problem that is used to test the nonoscillatory behavior of compressible gasdynamics codes. The initial solution is a discontinuity that has a high-density, high-pressure fluid on one side and a low-density, low-pressure fluid on the other side. As the simulation progresses, the initial discontinuity decomposes into a rarefaction wave, a contact discontinuity, and a shock wave. While the smooth problems discussed above test the accuracy and convergence properties of the spatial discretization, the Riemann problem tests the ability of an upwind discretization scheme to resolve the characteristic wavespeeds correctly and yield nonoscillatory solutions. In this section, we consider an electrostatic shock tube problem by noting that the LHS of (13) is identical to the compressible Euler equations for neutral gases [50], if the hydrodynamic pressure in the latter is replaced by the augmented pressure (sum of the hydrodynamic pressure and electron pressure). Eq. (18) is thus solved with a zero RHS on a one-dimensional domain $x \in [0, 1]$ with inviscid wall boundary conditions at both ends. The number of fluids is $n_f = 1$, and the initial solution is specified as

$$\{\rho, P^*\} = \begin{cases} \{1, 1\} & x \le 0.5 \\ \{0.125, 0.1\} & x > 0.5 \end{cases}, \quad u = 0, \quad P = P^* - \frac{ZT_e}{m}\rho. \tag{37}$$

We consider four cases, each with a different species:

$$\begin{array}{ll} \text{Neutral gas} & Z = 0, m \in \mathbb{R}^+ \\ \text{Hydrogen} & Z = 1, m = 1 \\ \text{Helium} & Z = 2, m = 4 \\ \text{Beryllium} & Z = 4, m = 9 \end{array}, \tag{38}$$

and the electron temperature T_e and the specific heat ratio γ are set to 0.01 and 5/3, respectively. Fig. 3 shows the density and pressure at $t = 0.15$ for solutions obtained on a grid with 256 points. The CFL number for all the simulations are set to $\sigma = 0.9$. The solutions obtained with the charged species (hydrogen, helium, and beryllium) agree well with the neutral gas solution, as expected. This case demonstrates that the characteristic-based MPWENO5 spatial discretization described in the previous section is able to yield nonoscillatory solutions for flows with strong discontinuities.

4.2. Collisional terms

We now verify our implementation of the collisional source terms by comparing them with previous results in the literature. Since these terms do not involve any spatial gradients, "zero-dimensional" cases with uniform flow variables are sufficient for their verification. We consider two setups that are identical to those solved by Rambo and Procassini [44]: The thermal equilibration of two fluids at initially different temperatures, and the slowing down of a fluid stream due to the collisional interactions with a background fluid. The results presented in Rambo and Procassini are in terms of the physical variables (with dimensions), and therefore, the discussion and figures in this section involve these variables (obtained by multiplying the nondimensional variables solved by EUCLID by the corresponding reference quantity). The primary reference quantities for the cases reported here are

$$m_{\text{ref}} = 1.67 \times 10^{-24}\,\text{g (proton mass)}, \quad n_{\text{ref}} = 1 \times 10^{20}\,\text{cm}^{-3}$$

$$x_{\text{ref}} = 0.1\,\text{cm (1 mm)}, \quad T_{\text{ref}} = 1.602 \times 10^{-9}\,\text{ergs (1 keV)},$$

$$e = 4.803 \times 10^{-10}\,\text{statcoulomb (elementary charge)},$$

and, therefore, the derived reference quantities are obtained from (10) are:

$$u_{\text{ref}} = 3.095 \times 10^7\,\text{cm/s}, \quad \rho_{\text{ref}} = 1.67 \times 10^{-4}\,\text{g/cm}^{-3},$$

$$t_{\text{ref}} = 3.23 \times 10^{-9}\,\text{s}, \quad P_{\text{ref}} = 1.602 \times 10^{11}\,\text{Ba}.$$

The specific heat ratio for all fluids is specified as $\gamma_\alpha = 5/3$.

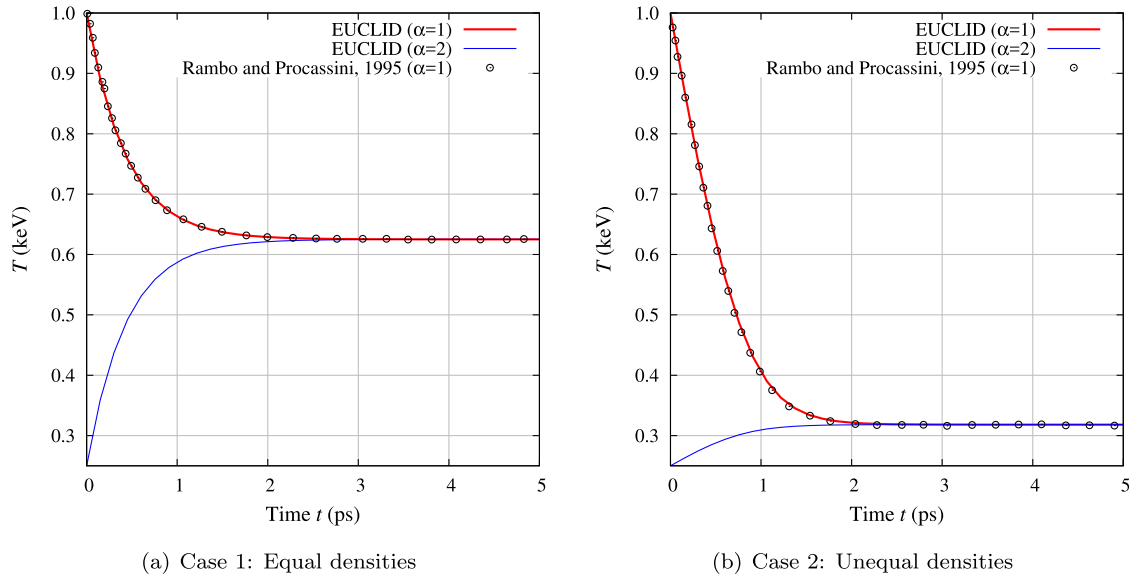(a) Case 1: Equal densities　　　　　　(b) Case 2: Unequal densities

Fig. 4. Thermal equilibration of two fluids at initially different temperatures. An excellent agreement is observed with results in the literature [44].

The first test considers two carbon fluids ($n_f = 2$, $Z_{\{1,2\}} = 6$, and $m_{\{1,2\}} = 12$), and two cases with the following initial solutions (in terms of the nondimensional variables):

Case 1 : $\quad n_1 = n_2 = 1,\quad\quad u_1 = u_2 = 0,\quad T_1 = 1,\ T_2 = 0.25$
Case 2 : $\quad n_1 = 0.1,\ n_2 = 1,\quad u_1 = u_2 = 0,\quad T_1 = 1,\ T_2 = 0.25.$
$$\tag{39}$$

The domain is $x = [0, 1]$ with periodic boundary conditions, and it is discretized by 8 grid points. The final time for the simulation is specified as $t_f = 0.00155$, which corresponds to a physical time of $\sim 5\,\mathrm{ps}$. A CFL of $\sigma = 0.0005$ is specified, where σ is given by (27); this value is obtained by "trial-and-error" to ensure that the equilibration time scale is resolved. Fig. 4 shows the evolution of the temperatures of both the fluids: in both cases, the two fluids equilibrate to the density-weighted mean of the initial temperatures $(n_1 T_1 + n_2 T_2)/(n_1 + n_2)$. An excellent agreement is observed with prior results [44].

The second test considers two carbon fluids as well, and two cases with the following initial solutions (in terms of the nondimensional variables) are solved:

Case 1 : $\quad n_1 = n_2 = 1,\quad\quad u_1 = 2.1166,\ u_2 = 0,\quad T_1 = T_2 = 0.5$
Case 2 : $\quad n_1 = 0.1,\ n_2 = 1,\quad u_1 = 2.1166,\ u_2 = 0,\quad T_1 = T_2 = 0.5.$
$$\tag{40}$$

The nondimensional u specified above corresponds to a physical velocity of $6.55 \times 10^7\,\mathrm{cm/s}$. The domain is $x = [0, 1]$ with periodic boundary conditions, and it is discretized by 8 grid points. The final time is specified as $t_f = 0.031$, which corresponds to a physical time of $\sim 100\,\mathrm{ps}$. A CFL of $\sigma = 0.01$ is specified to resolve the frictional time scale. Fig. 5(a) and (b) show the evolution of the velocities and temperatures of both the fluids for Case 1. The first fluid $\alpha = 1$ slows down and accelerates the second fluid $\alpha = 2$ due to the inter-fluid friction; the velocities for both the fluids converge to the mean value. The frictional forces also result in heating both the fluids, as is observed by the increase in the temperatures by a factor of ~ 10. Fig. 5(c) and (d) show the evolution of the velocities and temperatures for Case 2, where the density of the first fluid is 10 times lower. The velocities of both the fluids converge to the density-weighted mean. The evolution of the temperatures shows the rapid heating of the first (lower density) fluid due to the friction, followed by cooling due to thermal equilibration, while the second (higher density) fluid experiences a more gradual heating

due to friction and thermal equilibration. In both these cases, our results agree well with those previously published [44].

5. Results

In this section, we use EUCLID to simulate multifluid flows that are representative of laser-induced plasma experiments. We consider the interactions of counterstreaming plasmas in vacuum and in the presence of a gas fill. Both one-dimensional and two-dimensional cases are simulated. The simulations reported in these section use the following primary reference quantities that correspond to the physical setup of HEDP experiments [4,6,7,53]:

$m_{\mathrm{ref}} = 1.67 \times 10^{-24}\,\mathrm{g}$ (proton mass),

$n_{\mathrm{ref}} = 9.03 \times 10^{21}\,\mathrm{cm^{-3}}$ (n_{crit}),

$x_{\mathrm{ref}} = 0.1\,\mathrm{cm}$ (1 mm),$\quad T_{\mathrm{ref}} = 1.602 \times 10^{-9}\,\mathrm{ergs}$ (1 keV),

$e = 4.803 \times 10^{-10}\,\mathrm{statcoulomb}$ (elementary charge),

where n_{crit} is the critical plasma density above which the fluid becomes opaque to electromagnetic radiation [54]. The derived reference quantities are obtained from (10), and they are:

$u_{\mathrm{ref}} = 3.095 \times 10^7\,\mathrm{cm/s},\quad \rho_{\mathrm{ref}} = 1.51 \times 10^{-2}\,\mathrm{g/cm^{-3}},$
$t_{\mathrm{ref}} = 3.23 \times 10^{-9}\,\mathrm{s},\quad\quad P_{\mathrm{ref}} = 1.447 \times 10^{13}\,\mathrm{Ba}.$

The subsequent discussions in this section describe the nondimensional variables that are solved for by EUCLID; they can be multiplied by these reference quantities to obtain their physical values with units. The specific heat ratio for all species in the simulations presented in this section is specified as $\gamma_\alpha = 5/3$.

The initial solution in the cases discussed here involve fluids that exist with a specified density only within a part of the domain. This is implemented using a "smoothed slab" function that is defined in one spatial dimension as

$$\Upsilon(x; \delta_x, x_{\min}, x_{\max}) = e^{\frac{x}{\delta_x}} \left[\left(e^{\frac{x_{\min}}{\delta_x}} + e^{\frac{x}{\delta_x}} \right)^{-1} - \left(e^{\frac{x_{\max}}{\delta_x}} + e^{\frac{x}{\delta_x}} \right)^{-1} \right].$$
$$\tag{41}$$

This function attains unit value inside $(x_{\min}, x_{\max})$, smoothly transitions to 0 with a width δ_x, and is zero outside $[x_{\min}, x_{\max}]$. The transition width δ_x is specified as a small number; however we must caution the reader that very low values for δ_x results in numerical overflow/underflow while evaluating $\Upsilon(x)$. In the parts

(a) Case 1 (Equal densities): Velocity

(b) Case 1 (Equal densities): Temperature

(c) Case 2 (Unequal densities): Velocity

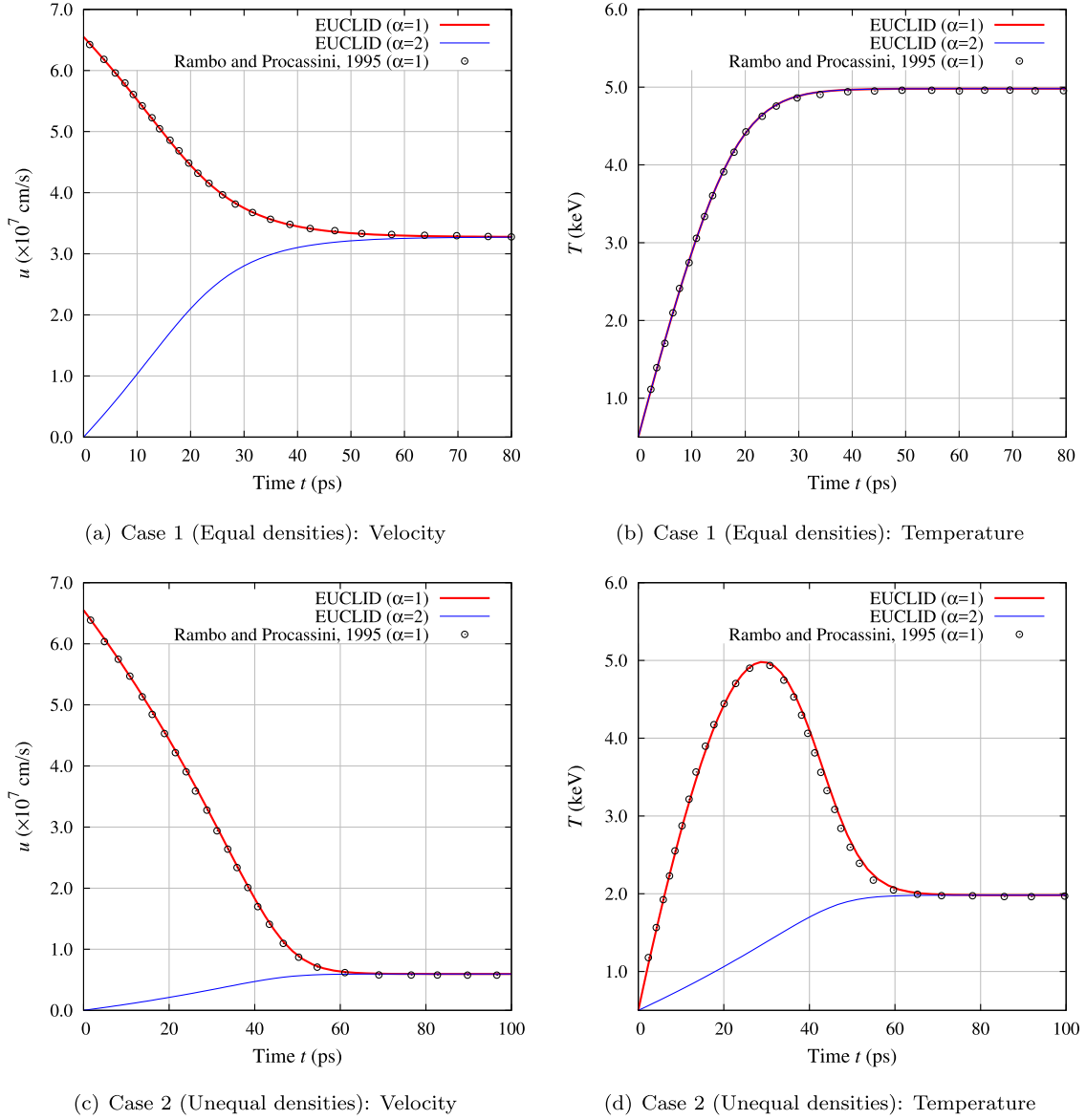(d) Case 2 (Unequal densities): Temperature

Fig. 5. Frictional slowing of one fluid with respect to a background fluid with an initially nonzero relative velocity. An excellent agreement is observed with results in the literature [44].

of the domain where a fluid doesn't exist, its density is specified as a small number ("numerical vacuum") since zero density will result in numerical exceptions. This number, n_{vac}, should be small enough such that the fluid doesn't significantly affect the overall flow physics in these regions (where it does not exist), yet the numerical discretization should be able to maintain the positivity of density and pressure. While the WENO algorithm is not strictly positivity-preserving, our use of the monotonicity-preserving limits on WENO and the robust characteristic-based reconstruction (as described in Section 3) allowed the numerical vacuum to be specified as $n_{vac} = 10^{-14}$.

The first example we consider is the one-dimensional interaction of two counterstreaming plasmas. The domain is $x \in [0, 1]$, and we consider two fluids $n_f = 2$. The initial solution is as follows:

$$n_\alpha(x, t = 0) = n_{vac} + \tilde{n}_\alpha \Upsilon(x; \delta_{x,\alpha}, x_{min,\alpha}, x_{max,\alpha})$$
$$T_\alpha(x, t = 0) = 1, \quad u_\alpha(x, 0) = 0 \qquad \alpha = 1, 2,$$

$$\tag{42}$$

where

$$x_{min,1} = -1, \; x_{max,1} = 0.2, \; x_{min,2} = 0.8, \; x_{max,2} = 2,$$
$$\delta_{x,\{1,2\}} = 0.005,$$

and $n_{vac} = 10^{-14}$. This represents a setup where fluid $\alpha = 1$ is a localized slab in $x \in [0, 0.2]$, and fluid $\alpha = 2$ is a localized slab in $x \in [0.8, 1]$ with vacuum in between (see the top left plots in Figs. 6 and 7). The electron temperature is $T_e = 0.5$, and we consider two symmetric cases (hydrogen–hydrogen and carbon–carbon interactions):

Case 1 (hydrogen – hydrogen) : $m_{\{1,2\}} = 1, \; Z_{\{1,2\}} = 2, \; \tilde{n}_{\{1,2\}} = 0.5$
Case 2 (carbon – carbon) : $m_{\{1,2\}} = 12, \; Z_{\{1,2\}} = 6, \; \tilde{n}_{\{1,2\}} = 0.1$.

The boundary conditions are as follows: An inviscid wall boundary is imposed at $x = 0$, and outflow is imposed at $x = 1$ for fluid $\alpha = 1$, while outflow is imposed at $x = 0$, and inviscid wall boundary is imposed at $x = 1$ for fluid $\alpha = 2$. The domain is discretized by a grid with 256 points, and the final time for the simulations is $t_f = 0.4$. The CFL number for Case 1 (hydrogen–hydrogen)
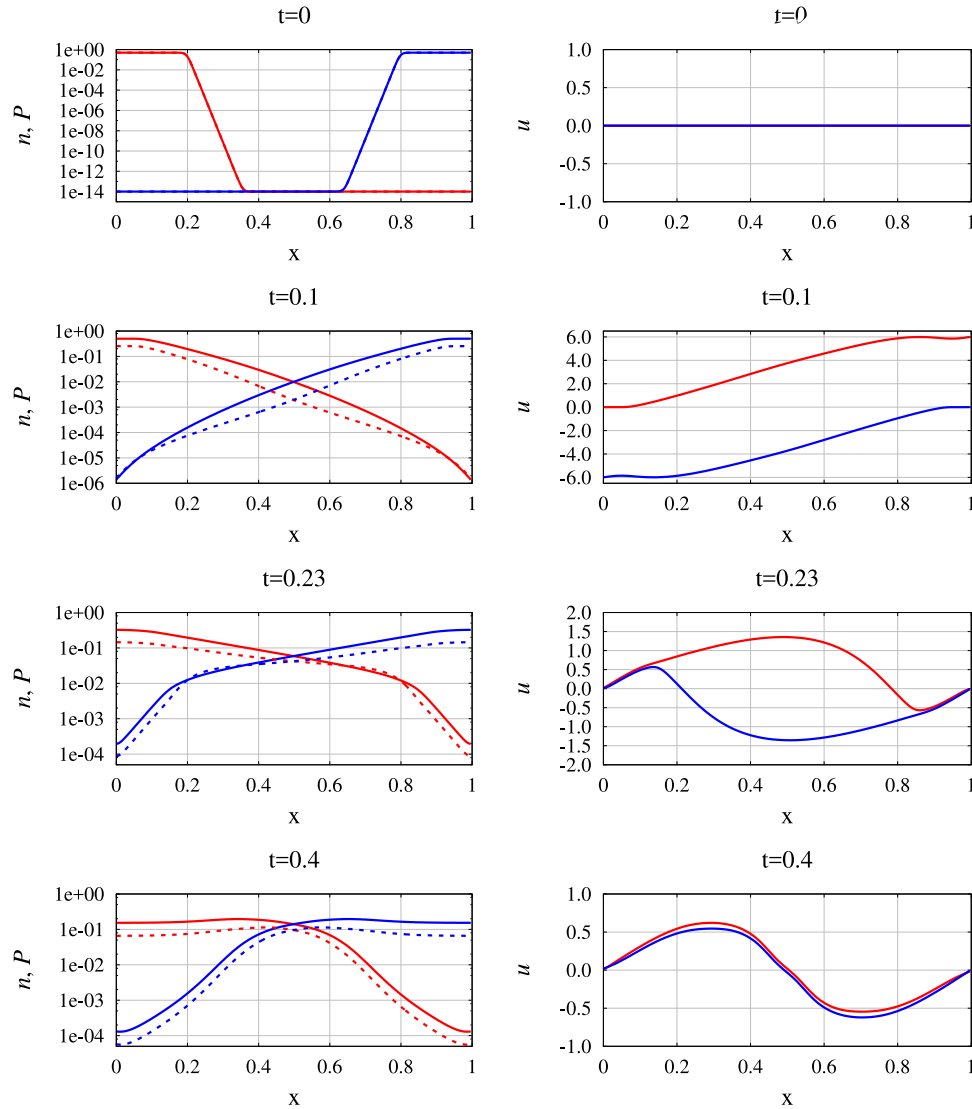
Fig. 6. Interpenetration of two hydrogen streams: The left column shows the number density n_α (solid lines) and the pressure P_α (dashed line), and the right column shows the velocity u_α. The red lines correspond to $\alpha = 1$, and the blue lines correspond to $\alpha = 2$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

was specified as 0.5, while that for Case 2 (carbon–carbon) was specified as 0.05; these values ensure that the collisional source terms are also linearly stable at the computed time step size.

Fig. 6 shows number density $n_{\{1,2\}}$ and pressure $P_{\{1,2\}}$ in the left column and the velocity $u_{\{1,2\}}$ in the right column for the hydrogen–hydrogen interaction (Case 1), where successive rows show the solution at simulation times of 0 (initial solution), 0.1, 0.23, and 0.4 (final time). The interaction can be described as follows. At $t = 0$ (initial solution), each fluid is piled up at either end of the domain. As the simulation progresses, the fluids expand from either side of the domain, and by $t = 0.1$, they interpenetrate each other. At this time, the velocity plot shows unimpeded expansion for each hydrogen fluid as they leave the domain. However, by $t = 0.23$, the friction between the fluids results in the outflowing fluid at each end of the domain getting pulled back by the dominant fluid that is flow in the opposite direction. For example, at $x \to 0$, the fluid $\alpha = 2$ has assumed the positive (rightward) velocity of the dominant fluid $\alpha = 1$ because it is being pulled back by it. At this time, the region between $x \sim 0.3$ and $x \sim 0.7$ exhibits a multifluid counterstreaming flow. The pressure and density

plots show a sharper gradient compared to a smooth expansion at $t = 0.1$ (in the region $x < 0.2$ for $\alpha = 2$ and $x > 0.8$ for $\alpha = 1$) indicating the push-back from the dominant fluid. Finally, at $t = 0.4$, while each fluid has been pushed back by the other at either end of the domain, the friction results in the velocities converging to a single value at each spatial location, and thus, this becomes a "single-fluid" flow that can be solved by a multispecies, single-fluid code.

Fig. 7 shows the solution for the carbon–carbon interaction (Case 2), where the same quantities are plotted at the same simulation times. Qualitatively, a similar behavior is observed; however, since carbon has a higher atomic number Z_α, the interactions (electrostatic and collisional forces) are significantly stronger. As a result, the fluids at either end of the domain are unable to reach and exit the other end of the domain. Even at an early time of $t = 0.1$, while each fluid interpenetrates each other and flow in opposite directions in the region between $x \sim 0.2$ and $x \sim 0.8$, the friction forces result in a small region adjacent to the expansion front where the expanding fluid is getting pushed back by the dominant fluid (for example, in the region $0.15 < x < 0.21$
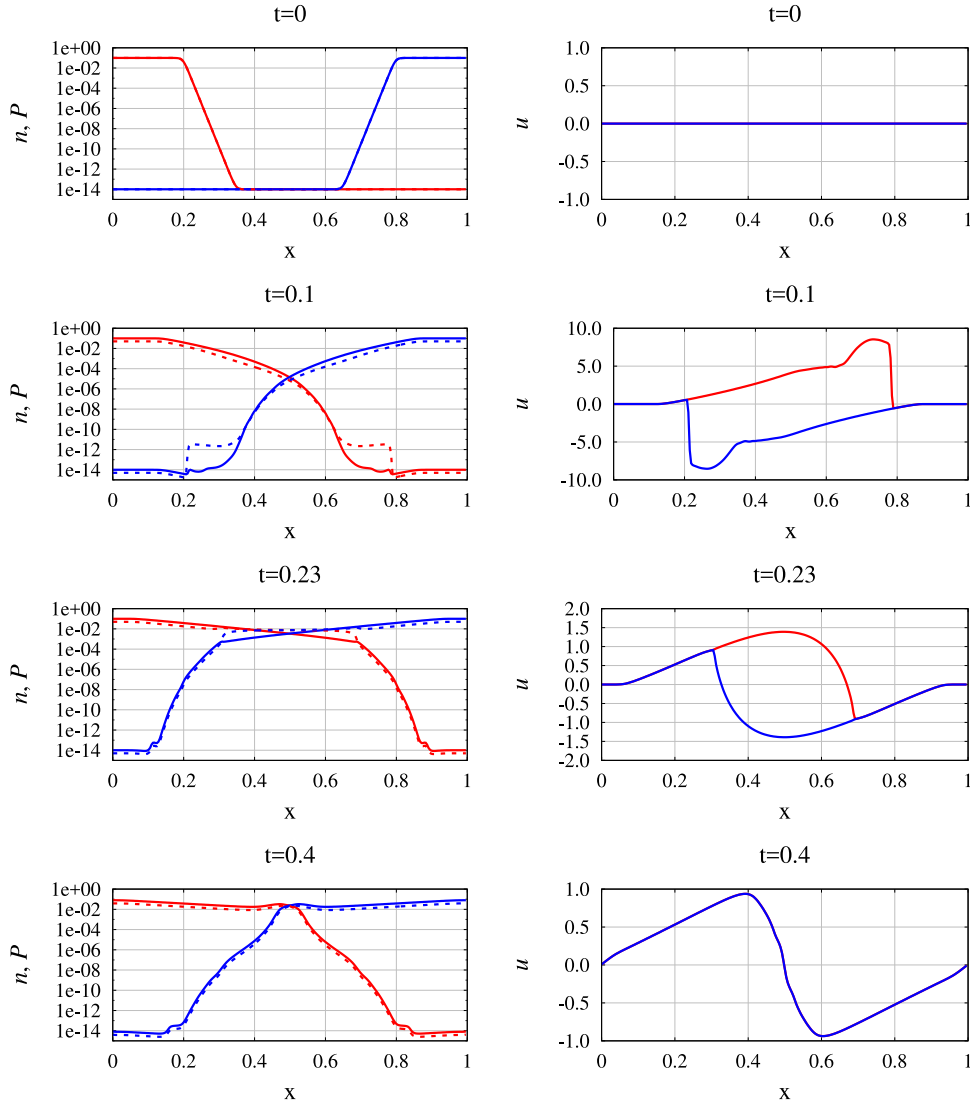
Fig. 7. Interpenetration of two carbon streams: The left column shows the number density n_α (solid lines) and the pressure P_α (dashed line), and the right column shows the velocity u_α. The red lines correspond to $\alpha = 1$, and the blue lines correspond to $\alpha = 2$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

for the fluid $\alpha = 2$). At $t = 0.23$, the pressure and number density gradients and the velocity show the push-back each fluid is facing from the other. Finally, by $t = 0.4$, the fluids have converged to a "single fluid" with identical velocities.

We now consider a two-dimensional interaction of two counter-streaming fluids. The domain is $x \in [0, 1]$, $y \in [0, 2]$, and the initial solution for the two fluids is:

$$n_\alpha(x, y, t = 0) = n_{vac} + \tilde{n}_\alpha \Upsilon(x; \delta_{x,\alpha}, x_{min,\alpha}, x_{max,\alpha})$$
$$T_\alpha(x, y, t = 0) = T_0 + \tilde{T}_\alpha \Upsilon(y; \delta_{y,\alpha}, y_{min,\alpha}, y_{max,\alpha}) \quad , \quad \alpha = 1, 2,$$
$$u_\alpha(x, y, t = 0) = v_\alpha(x, y, t = 0) = 0$$

$$(43)$$

where

$$x_{min,1} = -1, \; x_{max,1} = 0.2, \; x_{min,2} = 0.8, \; x_{max,2} = 2,$$
$$\delta_{x,\{1,2\}} = 0.005,$$
$$y_{min,1} = y_{min,2} = 0.8, \; y_{max,1} = y_{max,2} = 1.2, \; \delta_{y,\{1,2\}} = 0.02,$$
$$\tilde{n}_{\{1,2\}} = 0.08, \; T_0 = 1, \; \tilde{T}_{\{1,2\}} = 4,$$

and $n_{vac} = 10^{-14}$ is the numerical "vacuum". Along the x dimension, the density variation is identical to that in the one-dimensional examples discussed above, and this is shown in Fig. 8(a). Both the fluids are carbon ($Z_{\{1,2\}} = 6$ and $m_{\{1,2\}} = 12$), and we refer to the first fluid $\alpha = 1$ as "carbon-left" (red curve) and the second fluid $\alpha = 2$ as "carbon-right" (blue curve). Along the y dimension, the temperature towards the center of the domain in this dimension ($0.8 \le y \le 1.2$) is specified to be 5 times hotter than the temperature near the boundaries ($y < 0.8$, $y > 1.2$). Fig. 8(b) shows this temperature variation, which is identical for both fluids. This represents a situation where the two localized slabs of plasmas at each end of the domain are hotter in the middle than at the edges. This case is representative of the laser-induced plasma experiments where high-energy laser beams heat the carbon foils around their middle, while the edges remain cooler. The electron temperature is specified as $T_e = 0.5$. The boundary conditions are similar to that of the one-dimensional cases: Inviscid wall boundary is imposed at $x = 0$ for *carbon-left* and at $x = 1$ for *carbon-right*, and outflow is specified at $x = 1$ for *carbon-left* and at $x = 0$

(a) Initial number density
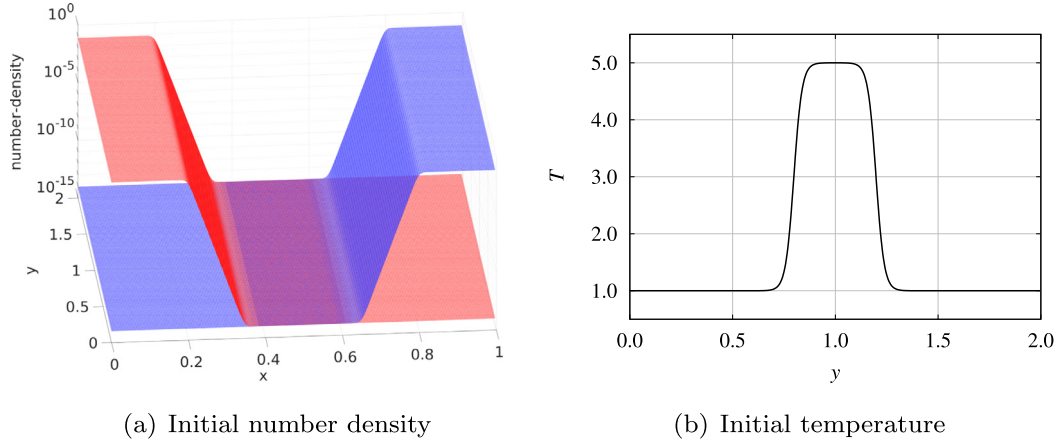
(b) Initial temperature

Fig. 8. Initial solution for the two-dimensional two-fluid (carbon–carbon) interpenetration case: The red mesh in the left figure is the number density of the first carbon fluid $\alpha = 1$ (*carbon-left*), while the blue mesh is the number density of the second carbon fluid $\alpha = 2$ (*carbon-right*). The right figure shows the initial temperature variation along y for both the fluids. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(a) $t = 0.08$

(b) $t = 0.14$

(c) $t = 0.20$

(d) $t = 0.35$
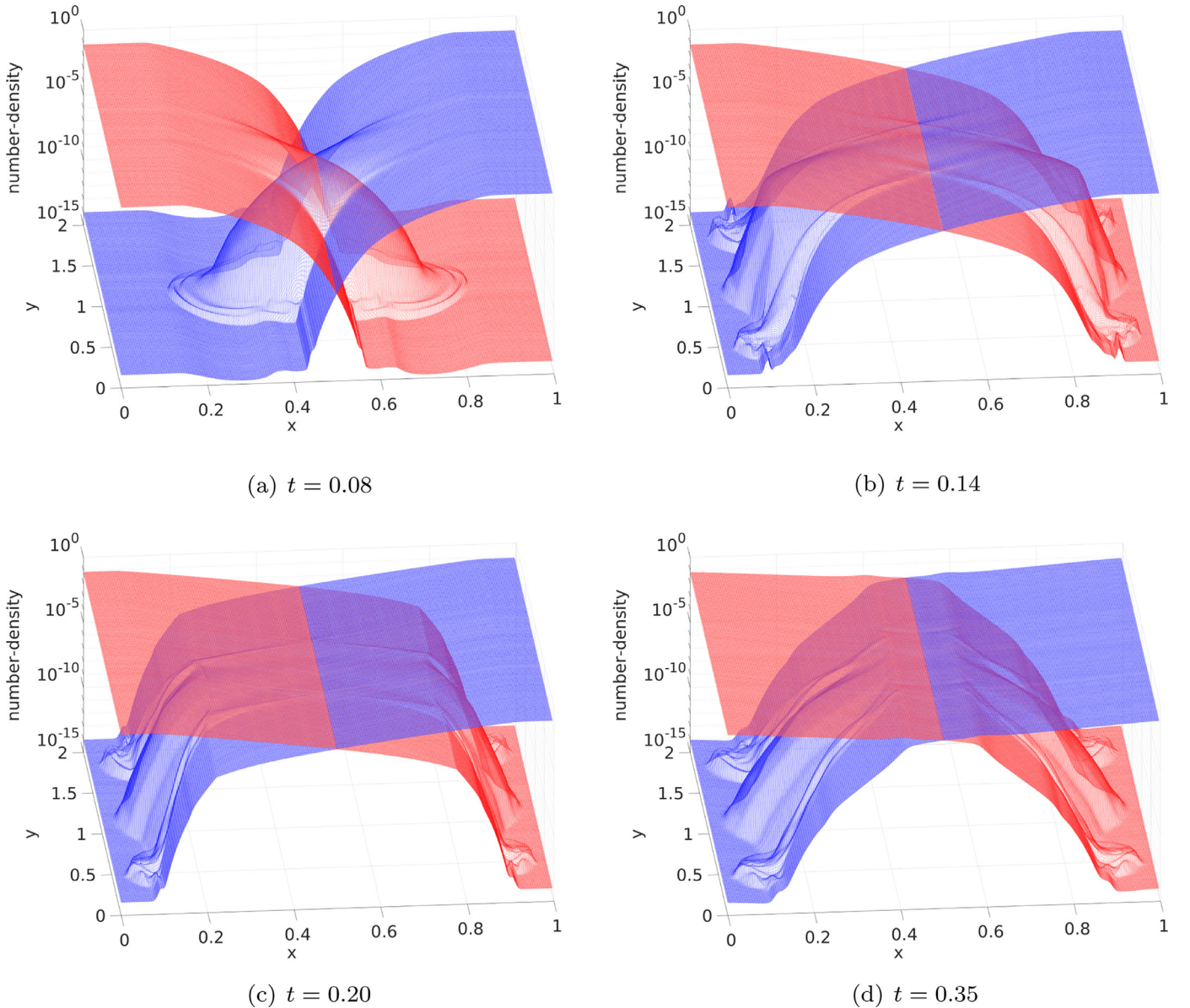
Fig. 9. Evolution of the number density for the two-dimensional carbon–carbon interpenetration case: The red mesh is *carbon-left*, and the blue mesh is *carbon-right*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(a) $t = 0.08$

(b) $t = 0.14$

(c) $t = 0.20$

(d) $t = 0.35$
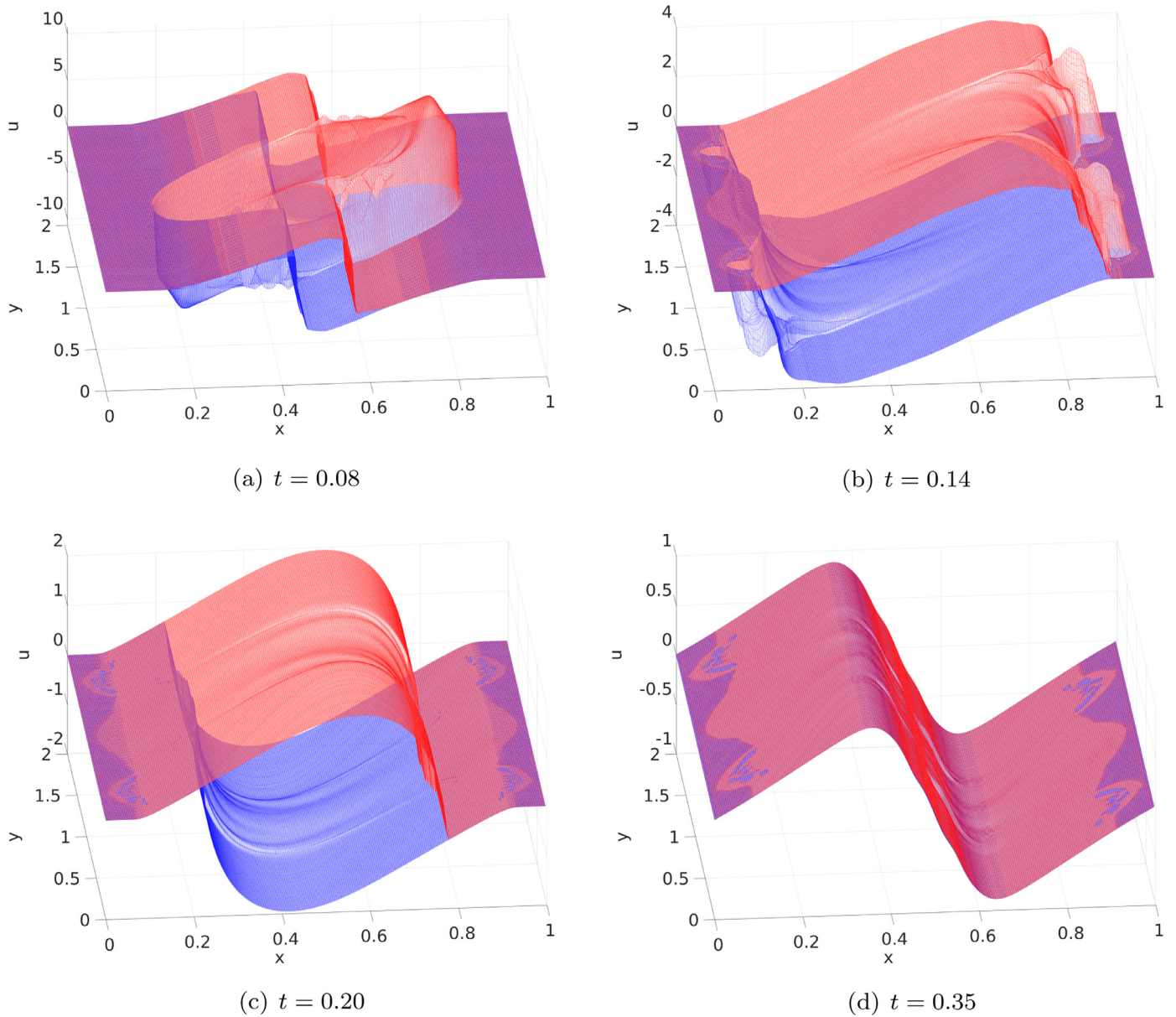
Fig. 10. Evolution of the x-velocity (u) for the two-dimensional carbon–carbon interpenetration case: The red mesh is *carbon-left*, and the blue mesh is *carbon-right*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

for *carbon-right*. Outflow boundaries are specified for both fluids at $y = 0, 2$.

The final time for the simulation is $t_f = 0.35$, and CFL is specified as $\sigma = 0.05$; this ensures that the time step size resolves the collisional terms. Fig. 9 shows the evolution of the number densities for the two fluids at $t = \{0.08, 0.14, 0.20, 0.35\}$, while Fig. 10 shows the evolution of u, the x-component of the velocity. In these figures, the red surfaces correspond to *carbon-left*, and the blue surfaces correspond to *carbon-right*. Along the x-dimension, the overall dynamics is similar to that in the one-dimensional example discussed above. The two fluids start expanding towards each other, as can be observed at $t = 0.08$. The expansion is faster near the center of the domain along y because of the higher temperature specified in that region. In Figs. 9(a) and 10(a), the expansion front in $0.8 \leq y \leq 1.2$ leads that near the y-boundaries. At $t = 1.4$, the two fluids have interpene-

trated into each other, and this is shown in Fig. 9(b). Fig. 10(b) shows that both the fluids are expanding in the large majority of the domain; however, they are also starting to get pushed back by the other fluid. For example, *carbon-left* is being pushed back by *carbon-right* near $x = 1$, and this is more pronounced for $0.8 \leq y \leq 1.2$ due to the higher temperature. The inter-fluid friction prevents each fluid from reaching the other end of the domain, and by $t = 0.2$, they get pushed back. Fig. 9(c) shows the sharp gradient in the number density that develops as a result of this push-back, and Fig. 10(c) shows the velocities converging to the same value near the x-boundaries, while counterstreaming flows exist towards the center of the domain $0.2 < x < 0.8$. For example, near $x = 0$, *carbon-right* is being carried right-ward (towards $x = 1$) by *carbon-left*. The two fluids assume identical velocities by $t = 0.35$, and they essentially become a single fluid.
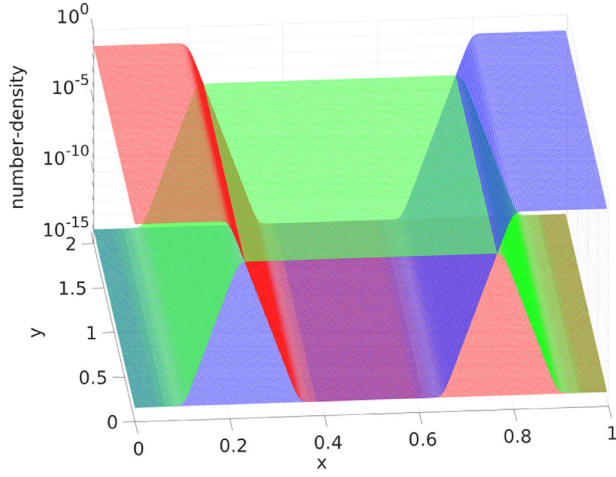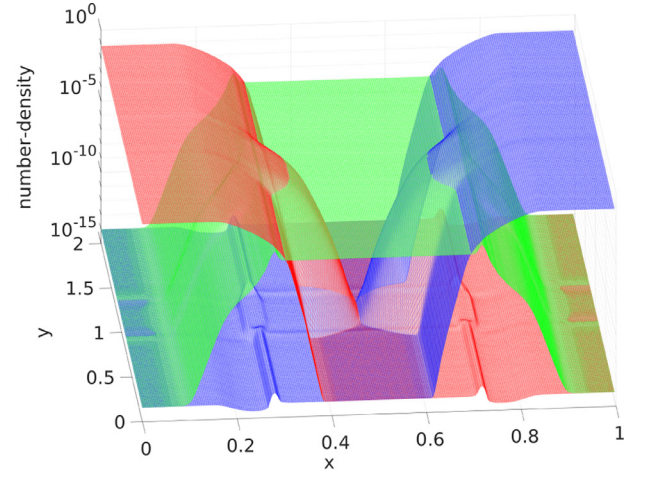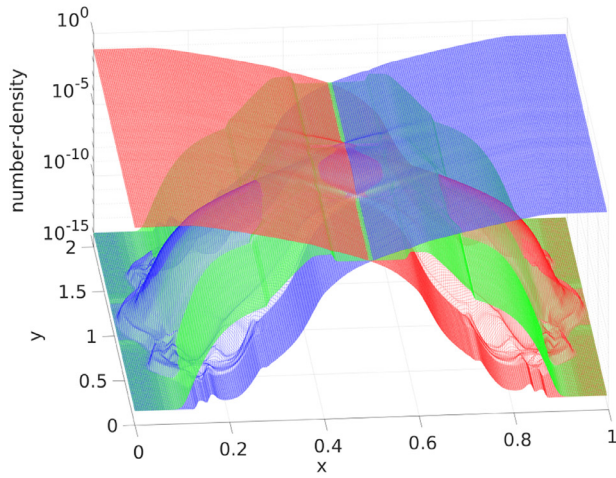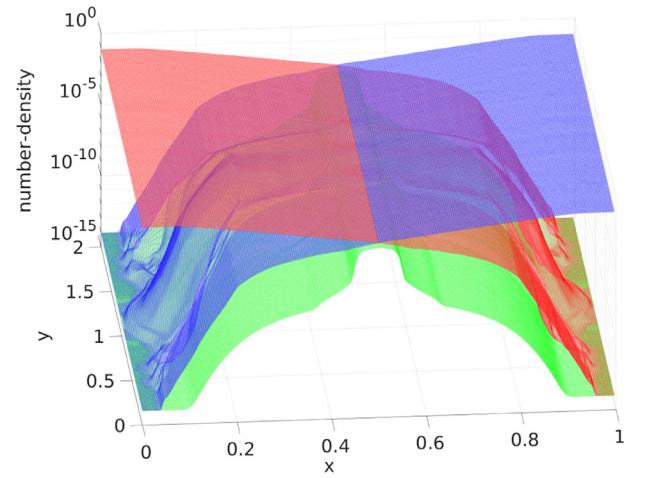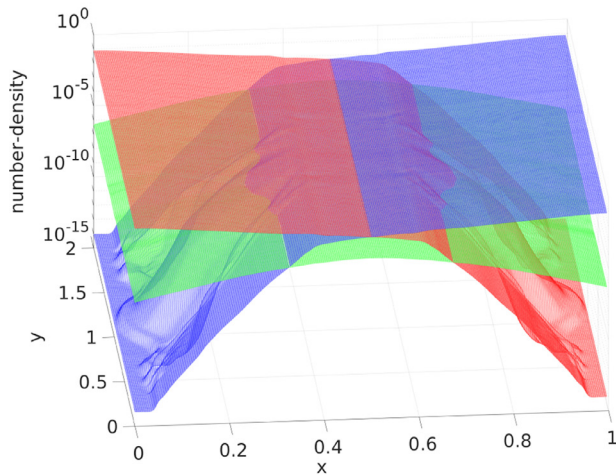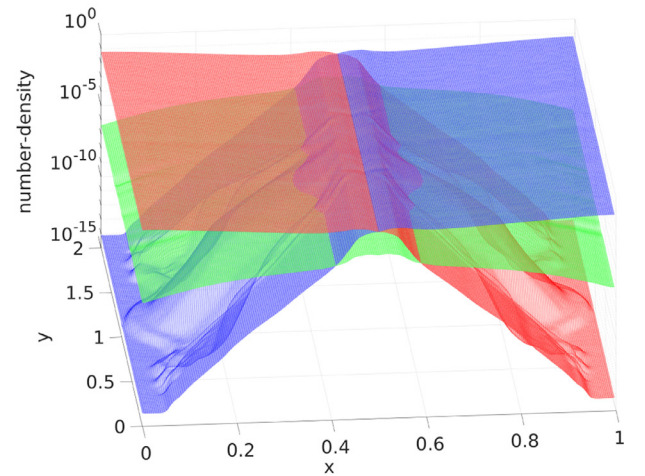
Fig. 11. Evolution of the number density for the two-dimensional interaction of two carbon fluids in the presence of helium gas fill: The red and blue meshes show the number densities for *carbon-left* ($\alpha = 1$) and *carbon-right* ($\alpha = 2$), respectively, while the green mesh shows the number density for *helium-fill* ($\alpha = 3$). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
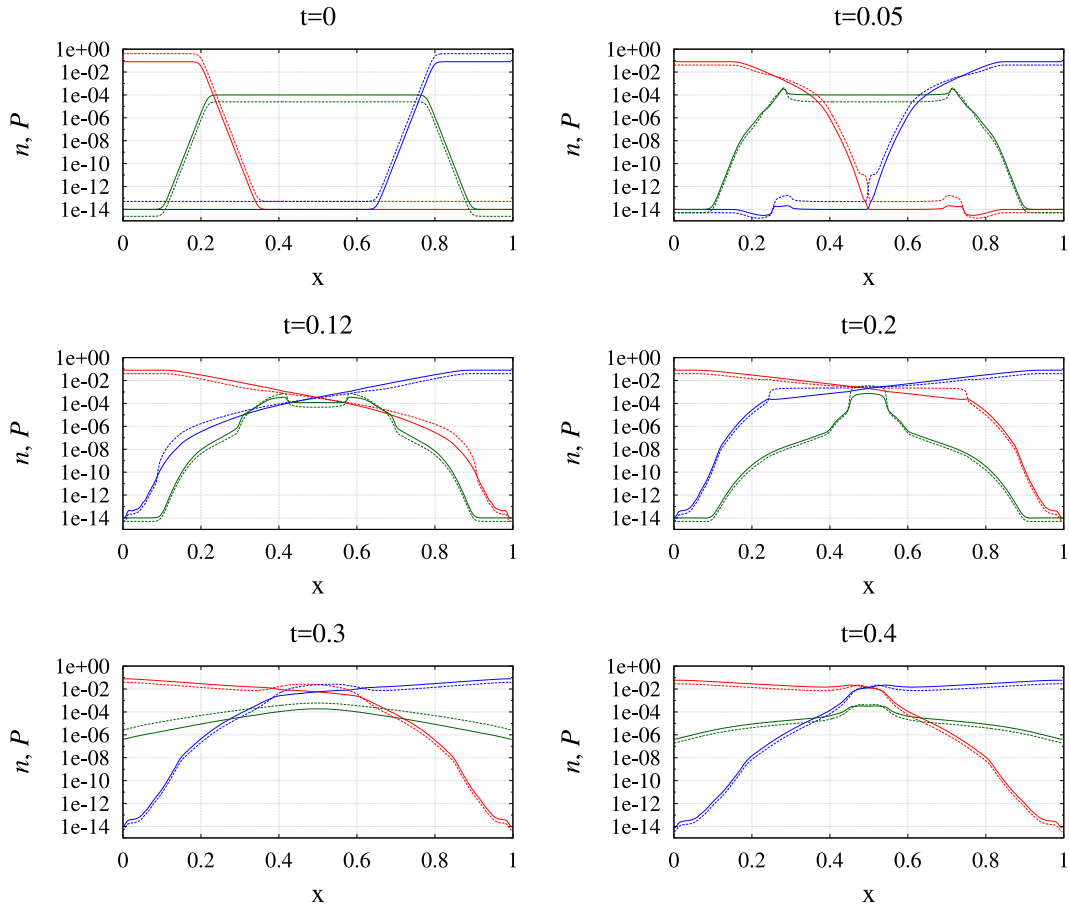
Fig. 12. Evolution of the pressure and number density for the two-dimensional interaction of two carbon fluids in the presence of helium gas fill along the *x*-dimension at *y* = 1: The red and blue curves correspond to *carbon-left* and *carbon-right*, respectively, and the green curve corresponds to *helium-fill*. The solid lines show the number density, and the dashed lines show the pressure. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Finally, we consider the two-dimensional interaction of two counterstreaming fluids in the presence of a gas fill. The details of the setup is identical to that in the previous example of the two-dimensional interaction of two carbon fluids with the addition of helium as the gas fill. Thus, the number of fluids is $n_f = 3$, and the properties of the third fluid are $Z_3 = 2$ and $m_3 = 4$. The initial solution for the third fluid, which we refer to as *helium-fill*, is as follows:

$$n_3(x, y, t = 0) = n_{\text{vac}} + \tilde{n}_3 \Upsilon(x; \delta_{x,3}, x_{\min,3}, x_{\max,3}),$$
$$T_3(x, y, t = 0) = T_{0,3} = 0.25, \quad u_3(x, y, t = 0) = v_3(x, y, t = 0) = 0,$$
$$\tag{44}$$

where $\tilde{n}_3 = 10^{-4}$ is the fill gas density, and

$$\delta_{x,3} = 0.005, \quad x_{\min,3} = 0.22, \quad x_{\max,3} = 0.78.$$

Thus, compared with the previous setup, the vacuum in between the two carbon fluids in the initial solution is filled by a low-density helium gas. The initial temperature for the two carbon fluids vary along the *y*-dimension as before, as shown in Fig. 8(b), and the initial temperature for the helium fluid is uniform. Fig. 11(a) shows the number density at $t = 0$, where the red and blue meshes correspond to *carbon-left* and *carbon-right*, respective, and the green mesh corresponds to *helium-fill*. The final time for the simulation is specified as $t_f = 0.4$, and the CFL is specified as $\sigma = 0.05$. The boundary conditions for *carbon-left* and *carbon-right*

are same as before; outflow boundaries are specified for *helium-fill* everywhere.

Fig. 11 shows the number density for the three fluids at various solution times, while Fig. 12 shows the number density and pressure variation along *x* at *y* = 1. As the two carbon fluids expand from either end of the domain, they drag the helium gas fill along with them, and this is observed in the solutions at $t = 0.05$ and $t = 0.12$. Fig. 13 shows the velocity along the *x*-dimension (*u*) at $y = 1$, and the solution at $t = 0.12$ shows that *helium-fill* assumes the velocity of the dominant carbon fluid (*carbon-left* in $x < 0.5$ and *carbon-right* in $x > 0.5$), while the two carbon fluids are expanding and interpenetrating. By $t = 0.2$, a large amount of *helium-fill* has accumulated in the center of the domain, and the velocity variation indicates that it has started expanding outwards through the two carbon fluids. Meanwhile, similar to the behavior observed in the previous two-fluid cases, the carbon fluids from either end are pushed back from the other boundary by the dominant fluid at that boundary, and although they are counterstreaming around the center of the domain, they have converged to a single velocity near the boundaries ($x < 0.3$ and $x > 0.7$). The two carbon fluids converge completely to a single velocity by $t = 0.3$, and each fluid has been pushed back by the other at each end of the domain; however, *helium-fill* streams through the carbon fluids to expand outward, and the velocity variation at $t = 0.3$ shows significant helium–carbon counterstreaming. Finally, at $t = 0.4$, all three fluids converge to a single velocity field and essentially become one fluid.
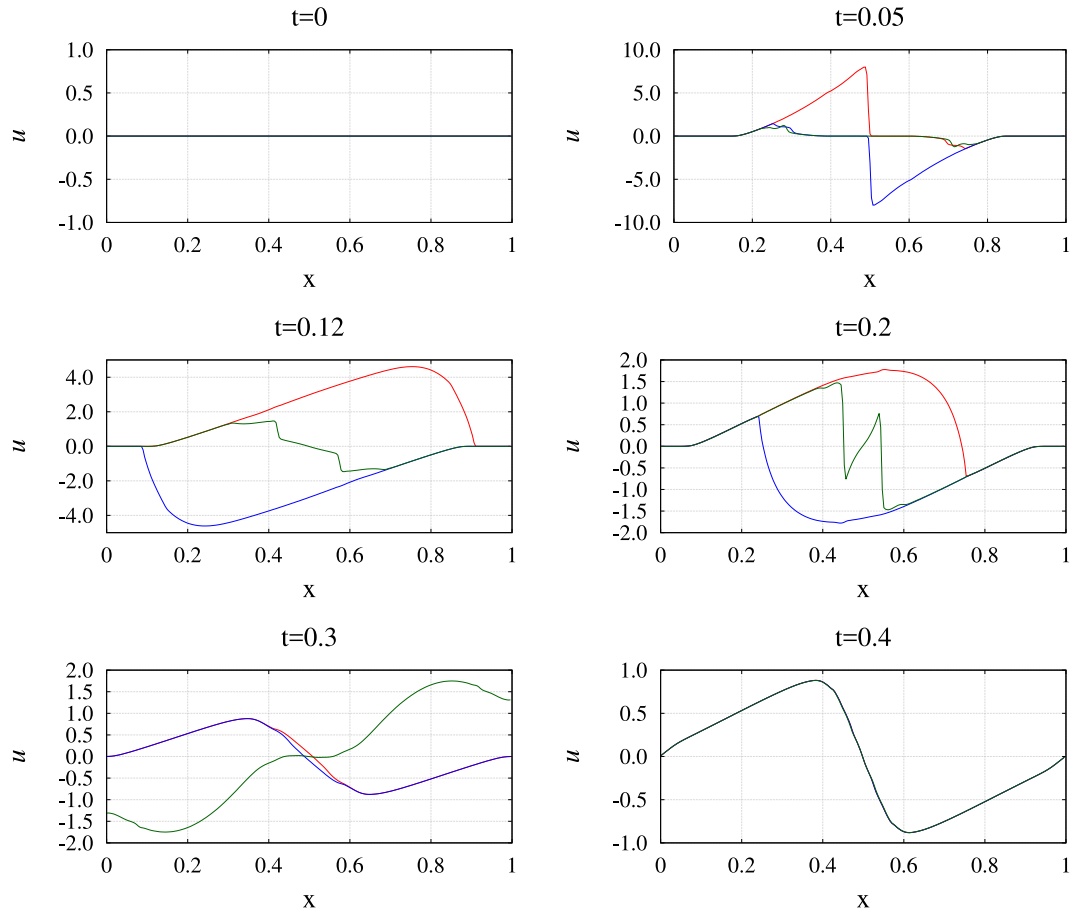
Fig. 13. Evolution of the *x*-velocity (*u*) for the two-dimensional interaction of two carbon fluids in the presence of helium gas fill along the *x*-dimension at $y = 1$: The red and blue curves correspond to *carbon-left* and *carbon-right*, respectively, and the green curve corresponds to *helium-fill*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

6. Conclusions

This paper reports the initial development of EUCLID, a multi-fluid code for simulating plasma interpenetration. The governing equations are the inviscid Euler equations, expressed as the conservation of mass, momentum, and energy, for each ion fluid. The formulation allows the partitioning of the same species into multiple fluids or streams. The fluids interact through electrostatic forces and collisional terms that include friction and thermal equilibration (both ion–ion and electron–ion). The electrons are assumed to be isothermal and inertia-less, and the plasma is assumed to be quasi-neutral; thus, the fluid equations for the electrons and the Poisson equation for the electrostatic potential need not be solved. The governing equations are discretized using a conservative finite-difference method, and the 5th-order Monotonicity-Preserving WENO scheme is used for the upwind discretization of the hyperbolic flux. Currently, the explicit 4th-order Runge–Kutta method is used for time integration. EUCLID is verified and its accuracy, convergence, and nonoscillatory behavior is demonstrated using several benchmark flow problems and the method of manufactured solutions, and these are reported in this paper. Finally, we simulate one- and two-dimensional flows that are representative of laser–induced plasma experiments–the interactions of counterstreaming fluids in vacuum as well as in the presence of a gas fill.

EUCLID is a "work-in-progress", and this paper is intended as a description of the basic algorithm. One of the challenges with our current approach results from using an explicit time integra-tion method. The time scales of the collisional terms (both friction and thermal equilibration) are often faster than the convective and acoustic time scales for our applications. In the results reported in this paper, a very low CFL number was needed to resolve these time scales, especially for interactions between heavier species like carbon. More physically-relevant simulations will involve heavier species such as gold ($Z = 40, m = 197$) resulting in faster colli-sional time scales. Thus, the implementation of implicit–explicit (IMEX) time integration methods [55–57] to integrate the source terms implicitly is being investigated. In addition, several aspects of the physical model and their improvements are areas of active research. Examples include removing the assumption of isother-mal electrons and solving the electron energy equation along with the ion fluid equations and the inclusion of heat source terms in the energy equations to simulate the continued localized heat-ing due to laser beams. Finally, counterstreaming plasmas result in the growth of kinetic instabilities that have a significant effect on the solution. A kinetic simulation of these instabilities within a fluid code is computationally intractable; however, we are in-vestigating the development of collisional kinetic models [58] that will appear as kinetic friction and heating terms in the fluid equations.

Appendix A. Monotonicity-preserving WENO5 (MPWENO5) scheme

This section summarizes the reconstruction procedure along dimension d; the purpose is to provide a brief but sufficient description that allows the exact reproducibility of our algorithm. The readers are encouraged to refer to the original publications [48,49] for a detailed description of this method. The fifth-order monotonicity-preserving WENO (MPWENO5) scheme described here computes a left-biased approximation to the flux primitive at the interface $\mathbf{i} + \frac{1}{2}\mathbf{e}_d$, given the values of the flux function at grid points $\mathbf{i} - 2\mathbf{e}_d, \cdots, \mathbf{i} + 2\mathbf{e}_d$. This same procedure can be used to compute the right-biased approximation by providing the flux functions at the grid points $\mathbf{i} + 3\mathbf{e}_d, \cdots, \mathbf{i} - \mathbf{e}_d$ as the inputs (note the reverse ordering). The fifth-order WENO approximation can be computed as:

$$f^{\text{WENO5}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \sum_{l=1}^{3} \omega_l f^{(l)}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}; \quad \omega_l = \frac{\zeta_l}{\sum_{m=1}^{3} \zeta_m}, \quad l = 1, 2, 3, \tag{A.1}$$

where

$$f^{(1)}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \frac{1}{3} f_{\mathbf{i}-2\mathbf{e}_d} - \frac{7}{6} f_{\mathbf{i}-\mathbf{e}_d} + \frac{11}{6} f_{\mathbf{i}}, \tag{A.2a}$$

$$f^{(2)}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = -\frac{1}{6} f_{\mathbf{i}-\mathbf{e}_d} + \frac{5}{6} f_{\mathbf{i}} + \frac{1}{3} f_{\mathbf{i}+\mathbf{e}_d}, \tag{A.2b}$$

$$f^{(3)}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \frac{1}{3} f_{\mathbf{i}} + \frac{5}{6} f_{\mathbf{i}+\mathbf{e}_d} - \frac{1}{6} f_{\mathbf{i}+2\mathbf{e}_d} \tag{A.2c}$$

are the three third-order approximations of $f_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$, and

$$\zeta_l = \frac{c_l}{\epsilon + \mathcal{S}_l}, \quad l = 1, 2, 3; \quad c_1 = \frac{1}{10}, \quad c_2 = \frac{6}{10}, \quad c_3 = \frac{3}{10}; \tag{A.3a}$$

$$\mathcal{S}_1 = \frac{13}{12} \left(f_{\mathbf{i}-2\mathbf{e}_d} - 2f_{\mathbf{i}-\mathbf{e}_d} + f_{\mathbf{i}} \right)^2 + \frac{1}{4} \left(f_{\mathbf{i}-2\mathbf{e}_d} - 4f_{\mathbf{i}-\mathbf{e}_d} + 3f_{\mathbf{i}} \right)^2, \tag{A.3b}$$

$$\mathcal{S}_2 = \frac{13}{12} \left(f_{\mathbf{i}-\mathbf{e}_d} - 2f_{\mathbf{i}} + f_{\mathbf{i}+\mathbf{e}_d} \right)^2 + \frac{1}{4} \left(f_{\mathbf{i}-\mathbf{e}_d} - f_{\mathbf{i}+\mathbf{e}_d} \right)^2, \tag{A.3c}$$

$$\mathcal{S}_3 = \frac{13}{12} \left(f_{\mathbf{i}} - 2f_{\mathbf{i}+\mathbf{e}_d} + f_{\mathbf{i}+2\mathbf{e}_d} \right)^2 + \frac{1}{4} \left(f_{\mathbf{i}} - 4f_{\mathbf{i}+\mathbf{e}_d} + 3f_{\mathbf{i}+2\mathbf{e}_d} \right)^2, \tag{A.3d}$$

where ζ_l are the WENO weights that are computed by scaling the optimal weights c_l by the smoothness indicators $\mathcal{S}_l$, and $\epsilon = 10^{-6}$ is a small parameter to prevent division by zero. In the regions where the solution is smooth, $\omega_l \to c_l$, and (A.1) becomes

$$f^{\text{WENO5}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \sum_{l=1}^{3} c_l f^{(l)}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \frac{1}{30} f_{\mathbf{i}-2\mathbf{e}_d} - \frac{13}{60} f_{\mathbf{i}-\mathbf{e}_d} + \frac{47}{60} f_{\mathbf{i}} + \frac{27}{60} f_{\mathbf{i}+\mathbf{e}_d}$$
$$- \frac{1}{20} f_{\mathbf{i}+2\mathbf{e}_d}, \tag{A.4}$$

which is the fifth-order approximation of $f_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$. Near discontinuities, the weights ω_l corresponding to the stencils containing the discontinuity go to zero, and a nonoscillatory approximation biased away from the discontinuity is computed.

The MPWENO5 approximation is obtained from $f^{\text{WENO5}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$ by ensuring that the final approximation is within the monotonicity-preserving bounds. Defining the following functions:

$$\text{sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}, \tag{A.5a}$$

$$\text{minmod}(x, y) = \frac{1}{2} (\text{sign}(x) + \text{sign}(y)) \min(|x|, |y|), \tag{A.5b}$$

$$\text{median}(x, y, z) = x + \text{minmod}(y - x, z - x), \tag{A.5c}$$

the monotonicity-preserving approximation is expressed as:

$$f_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \text{median}\left(f^{\text{WENO5}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}, f^{\min}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}, f^{\max}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \right). \tag{A.6}$$

The minimum and maximum bounds are computed as

$$f^{\min}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \max\left[\min\left(f_{\mathbf{i}}, f_{\mathbf{i}+\mathbf{e}_d}, f^{\text{med}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \right), \min\left(f_{\mathbf{i}}, f^{\text{UL}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}, f^{\text{LC}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \right) \right], \tag{A.7a}$$

$$f^{\max}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \min\left[\max\left(f_{\mathbf{i}}, f_{\mathbf{i}+\mathbf{e}_d}, f^{\text{med}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}\right), \max\left(f_{\mathbf{i}}, f^{\text{UL}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}, f^{\text{LC}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}\right)\right],$$ (A.7b)

where

$$f^{\text{med}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \frac{1}{2}\left(f_{\mathbf{i}} + f_{\mathbf{i}+\mathbf{e}_d} - d^{\text{med}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}\right),$$ (A.8a)

$$f^{\text{UL}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = f_{\mathbf{i}} + \upsilon\left(f_{\mathbf{i}} - f_{\mathbf{i}-\mathbf{e}_d}\right),$$ (A.8b)

$$f^{\text{LC}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = f_{\mathbf{i}} + \frac{1}{2}\left(f_{\mathbf{i}} - f_{\mathbf{i}-\mathbf{e}_d}\right) + \frac{\beta}{3}d^{\text{LC}}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}$$ (A.8c)

are the median value of the flux, an upper limit based on υ, and an estimate allowing for a large curvature (LC), respectively. In our implementation, we set $\upsilon = 2$, $\beta = 4$, and $d^{\text{LC}}_{(\cdot)} = d^{\text{med}}_{(\cdot)} = d^{\text{MM}}_{(\cdot)}$, where

$$d^{\text{MM}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \text{minmod}\left(d_{\mathbf{i}}, d_{\mathbf{i}+\mathbf{e}_d}\right); \quad d_{\mathbf{i}} = f_{\mathbf{i}-\mathbf{e}_d} - 2f_{\mathbf{i}} + f_{\mathbf{i}+\mathbf{e}_d}.$$ (A.9)

These choices for the monotonicity-preserving scheme work well for our applications, which involves density and pressure varying from $\mathcal{O}(1)$ to $\mathcal{O}(10^{-14})$, based on the discussion on the role of these parameters and their effect on the solution [49].

Appendix B. Eigenstructure

The complete eigenstructure of (13), derived using Mathematica [51], is provided in this section. The eigenvalues of the flux Jacobian $\partial \mathbf{f}^{(d)}_\alpha / \partial \mathbf{U}_\alpha$; $d \in \{x, y, z\}$ along each dimension for the fluid with index α are:

$$\Lambda^{(x)}_\alpha = \{u_\alpha, u_\alpha, u_\alpha, u_\alpha - c^*_\alpha, u_\alpha + c^*_\alpha\},$$ (B.1a)

$$\Lambda^{(y)}_\alpha = \{v_\alpha, v_\alpha, v_\alpha, v_\alpha - c^*_\alpha, v_\alpha + c^*_\alpha\},$$ (B.1b)

$$\Lambda^{(z)}_\alpha = \{w_\alpha, w_\alpha, w_\alpha, w_\alpha - c^*_\alpha, w_\alpha + c^*_\alpha\},$$ (B.1c)

where $c^*_\alpha = \sqrt{\gamma_\alpha P^*_\alpha / \rho_\alpha}$ is the augmented speed of sound, and u_α, v_α, w_α are the Cartesian components of the velocity vector $\mathbf{u}_\alpha$. The right and left eigenvectors along the three dimensions are provided below. In these expressions,

$$s^2_\alpha = u^2_\alpha + v^2_\alpha + w^2_\alpha, \quad \tau_\alpha = \frac{Z_\alpha T_e}{m_\alpha}, \quad \text{and} \quad \mu_\alpha = \gamma_\alpha - 1.$$

Along X.

$$R^{(x)}_\alpha = \begin{pmatrix} 0 & 0 & 1 & \frac{1}{c^{*2}_\alpha} & \frac{1}{c^{*2}_\alpha} \\ 0 & 0 & u_\alpha & \frac{u_\alpha - c^*_\alpha}{c^{*2}_\alpha} & \frac{u_\alpha + c^*_\alpha}{c^{*2}_\alpha} \\ 0 & \rho_\alpha & v_\alpha & \frac{v_\alpha}{c^{*2}_\alpha} & \frac{v_\alpha}{c^{*2}_\alpha} \\ \rho_\alpha & 0 & w_\alpha & \frac{w_\alpha}{c^{*2}_\alpha} & \frac{w_\alpha}{c^{*2}_\alpha} \\ \rho_\alpha w_\alpha & \rho_\alpha v_\alpha & \frac{1}{2}\left(s^2_\alpha - \frac{2\tau_\alpha}{\mu_\alpha}\right) & \frac{\left(s^2_\alpha - \frac{2\tau_\alpha}{\mu_\alpha}\right)}{2c^{*2}_\alpha} - \frac{u_\alpha}{c^*_\alpha} + \frac{1}{\mu_\alpha} & \frac{\left(s^2_\alpha - \frac{2\tau_\alpha}{\mu_\alpha}\right)}{2c^{*2}_\alpha} + \frac{u_\alpha}{c^*_\alpha} + \frac{1}{\mu_\alpha} \end{pmatrix}$$

$$L^{(x)}_\alpha = \begin{pmatrix} -\frac{w_\alpha}{\rho_\alpha} & 0 & 0 & \frac{1}{\rho_\alpha} & 0 \\ -\frac{v_\alpha}{\rho_\alpha} & 0 & \frac{1}{\rho_\alpha} & 0 & 0 \\ \frac{\left(s^2_\alpha - 2\tau_\alpha - s^2_\alpha \gamma_\alpha\right)}{2c^2_\alpha} + 1 & \frac{\mu_\alpha u_\alpha}{c^2_\alpha} & \frac{\mu_\alpha v_\alpha}{c^2_\alpha} & \frac{\mu_\alpha w_\alpha}{c^2_\alpha} & -\frac{\mu_\alpha}{c^2_\alpha} \\ \frac{\mu_\alpha s^2_\alpha}{4} + \frac{\tau_\alpha}{2} + \frac{u_\alpha c^*_\alpha}{2} & -\frac{1}{2}\left(\mu_\alpha u_\alpha + c^*_\alpha\right) & -\frac{1}{2}\left(\mu_\alpha v_\alpha\right) & -\frac{1}{2}\left(\mu_\alpha w_\alpha\right) & \frac{\mu_\alpha}{2} \\ \frac{\mu_\alpha s^2_\alpha}{4} + \frac{\tau_\alpha}{2} - \frac{u_\alpha c^*_\alpha}{2} & -\frac{1}{2}\left(\mu_\alpha u_\alpha - c^*_\alpha\right) & -\frac{1}{2}\left(\mu_\alpha v_\alpha\right) & -\frac{1}{2}\left(\mu_\alpha w_\alpha\right) & \frac{\mu_\alpha}{2} \end{pmatrix}$$

Along Y.

$$R^{(y)}_\alpha = \begin{pmatrix} 0 & 0 & 1 & \frac{1}{c^{*2}_\alpha} & \frac{1}{c^{*2}_\alpha} \\ 0 & \rho_\alpha & u_\alpha & \frac{u_\alpha}{c^{*2}_\alpha} & \frac{u_\alpha}{c^{*2}_\alpha} \\ 0 & 0 & v_\alpha & \frac{v_\alpha - c^*_\alpha}{c^{*2}_\alpha} & \frac{v_\alpha + c^*_\alpha}{c^{*2}_\alpha} \\ \rho_\alpha & 0 & w_\alpha & \frac{w_\alpha}{c^{*2}_\alpha} & \frac{w_\alpha}{c^{*2}_\alpha} \\ \rho_\alpha w_\alpha & \rho_\alpha u_\alpha & \frac{1}{2}\left(s^2_\alpha - \frac{2\tau_\alpha}{\mu_\alpha}\right) & \frac{\left(s^2_\alpha - \frac{2\tau_\alpha}{\mu_\alpha}\right)}{2c^{*2}_\alpha} - \frac{v_\alpha}{c^*_\alpha} + \frac{1}{\mu_\alpha} & \frac{\left(s^2_\alpha - \frac{2\tau_\alpha}{\mu_\alpha}\right)}{2c^{*2}_\alpha} + \frac{v_\alpha}{c^*_\alpha} + \frac{1}{\mu_\alpha} \end{pmatrix}$$

$$L_\alpha^{(y)} = \begin{pmatrix} -\frac{w_\alpha}{\rho_\alpha} & 0 & 0 & \frac{1}{\rho_\alpha} & 0 \\ -\frac{u_\alpha}{\rho_\alpha} & \frac{1}{\rho_\alpha} & 0 & 0 & 0 \\ \frac{(s^2-2\tau_\alpha-s_\alpha^2\gamma_\alpha)}{2c_\alpha^{*2}}+1 & \frac{\mu_\alpha u_\alpha}{c_\alpha^2} & \frac{\mu_\alpha v_\alpha}{c_\alpha^2} & \frac{\mu_\alpha w_\alpha}{c_\alpha^2} & -\frac{\mu_\alpha}{c_\alpha^2} \\ \frac{\mu_\alpha s_\alpha^2}{4}+\frac{\tau_\alpha}{2}+\frac{v_\alpha c_\alpha^*}{2} & -\frac{1}{2}(\mu_\alpha u_\alpha) & -\frac{1}{2}(\mu_\alpha v_\alpha+c_\alpha^*) & -\frac{1}{2}(\mu_\alpha w_\alpha) & \frac{\mu_\alpha}{2} \\ \frac{\mu_\alpha s_\alpha^2}{4}+\frac{\tau_\alpha}{2}-\frac{v_\alpha c_\alpha^*}{2} & -\frac{1}{2}(\mu_\alpha u_\alpha) & -\frac{1}{2}(\mu_\alpha v_\alpha-c_\alpha^*) & -\frac{1}{2}(\mu_\alpha w_\alpha) & \frac{\mu_\alpha}{2} \end{pmatrix}$$

Along Z.

$$R_\alpha^{(z)} = \begin{pmatrix} 0 & 0 & 1 & \frac{1}{c_\alpha^{*2}} & \frac{1}{c_\alpha^{*2}} \\ 0 & \rho_\alpha & u_\alpha & \frac{u_\alpha}{c_\alpha^2} & \frac{u_\alpha}{c_\alpha^2} \\ \rho_\alpha & 0 & v_\alpha & \frac{v_\alpha}{c_\alpha^2} & \frac{v_\alpha}{c_\alpha^2} \\ 0 & 0 & w_\alpha & \frac{w_\alpha-c_\alpha^*}{c_\alpha^2} & \frac{w_\alpha+c_\alpha^*}{c_\alpha^2} \\ \rho_\alpha v_\alpha & \rho_\alpha u_\alpha & \frac{1}{2}\left(s_\alpha^2-\frac{2\tau_\alpha}{\mu_\alpha}\right) & \frac{(s_\alpha^2-\frac{2\tau_\alpha}{\mu_\alpha})}{2c_\alpha^2}-\frac{w_\alpha}{c_\alpha^*}+\frac{1}{\mu_\alpha} & \frac{(s_\alpha^2-\frac{2\tau_\alpha}{\mu_\alpha})}{2c_\alpha^2}+\frac{w_\alpha}{c_\alpha^*}+\frac{1}{\mu_\alpha} \end{pmatrix}$$

$$L_\alpha^{(z)} = \begin{pmatrix} -\frac{v_\alpha}{\rho_\alpha} & 0 & \frac{1}{\rho_\alpha} & 0 & 0 \\ -\frac{u_\alpha}{\rho_\alpha} & \frac{1}{\rho_\alpha} & 0 & 0 & 0 \\ \frac{(s_\alpha^2-2\tau_\alpha-s_\alpha^2\gamma_\alpha)}{2c_\alpha^2}+1 & \frac{\mu_\alpha u_\alpha}{c_\alpha^2} & \frac{\mu_\alpha v_\alpha}{c_\alpha^2} & \frac{\mu_\alpha w_\alpha}{c_\alpha^2} & -\frac{\mu_\alpha}{c_\alpha^2} \\ \frac{\mu_\alpha s_\alpha^2}{4}+\frac{\tau_\alpha}{2}+\frac{w_\alpha c_\alpha^*}{2} & -\frac{1}{2}(\mu_\alpha u_\alpha) & -\frac{1}{2}(\mu_\alpha v_\alpha) & -\frac{1}{2}(\mu_\alpha w_\alpha+c_\alpha^*) & \frac{\mu_\alpha}{2} \\ \frac{\mu_\alpha s_\alpha^2}{4}+\frac{\tau_\alpha}{2}-\frac{w_\alpha c_\alpha^*}{2} & -\frac{1}{2}(\mu_\alpha u_\alpha) & -\frac{1}{2}(\mu_\alpha v_\alpha) & -\frac{1}{2}(\mu_\alpha w_\alpha-c_\alpha^*) & \frac{\mu_\alpha}{2} \end{pmatrix}$$

The eigenvalues and eigenvector matrices for the entire system (15) are obtained by assembling the eigenvalues and eigenvector matrices for each fluid in a block-wise manner:

$$\Lambda^{(d)} = \left\{\Lambda_1^{(d)}, \cdots, \Lambda_\alpha^{(d)}, \cdots, \Lambda_{n_f}^{(d)}\right\},$$

$$R^{(d)} = \begin{bmatrix} R_1^{(d)} & & & & \\ & \ddots & & & \\ & & R_\alpha^{(d)} & & \\ & & & \ddots & \\ & & & & R_{n_f}^{(d)} \end{bmatrix}, \quad L^{(d)} = \begin{bmatrix} L_1^{(d)} & & & & \\ & \ddots & & & \\ & & L_\alpha^{(d)} & & \\ & & & \ddots & \\ & & & & L_{n_f}^{(d)} \end{bmatrix}$$

for $d \in \{x, y, z\}$.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.compfluid.2019.04.012.

References

[1] Berzak Hopkins LF, Le Pape S, Divol L, Meezan NB, Mackinnon AJ, Ho DD, et al. Near-vacuum hohlraums for driving fusion implosions with high density carbon ablators. Phys Plasmas 2015;22(5):056318. doi:10.1063/1.4921151.

[2] Le Pape S, Berzak Hopkins LF, Divol L, Meezan N, Turnbull D, Mackinnon AJ, et al. The near vacuum hohlraum campaign at the NIF: a new approach. Phys Plasmas 2016;23(5):056311. doi:10.1063/1.4950843.

[3] Bosch RA, Berger RL, Failor BH, Delamater ND, Charatis G, Kauffman RL. Collision and interpenetration of plasmas created by laser illuminated disks. Physics of Fluids B: Plasma Physics 1992;4(4):979–88. doi:10.1063/1.860114.

[4] Kuramitsu Y, Sakawa Y, Morita T, Gregory CD, Waugh JN, Dono S, et al. Time evolution of collisionless shock in counterstreaming laser-produced plasmas. Phys Rev Lett 2011;106:175002. doi:10.1103/PhysRevLett.106.175002.

[5] Park H-S, Ryutov D, Ross J, Kugland N, Glenzer S, Plechaty C, et al. Studying astrophysical collisionless shocks with counterstreaming plasmas from high power lasers. High Energy Density Phys 2012;8(1):38–45. doi:10.1016/j.hedp.2011.11.001.

[6] Ross JS, Glenzer SH, Amendt P, Berger R, Divol L, Kugland NL, et al. Characterizing counter-streaming interpenetrating plasmas relevant to astrophysical collisionless shocks. Phys Plasmas 2012;19(5):056501. doi:10.1063/1.3694124.

[7] Ryutov DD, Kugland NL, Park H-S, Plechaty C, Remington BA, Ross JS. Intra-jet shocks in two counter-streaming, weakly collisional plasma jets. Phys Plasmas 2012;19(7):074501. doi:10.1063/1.4736973.

[8] Marinak MM, Haan SW, Dittrich TR, Tipton RE, Zimmerman GB. A comparison of three-dimensional multimode hydrodynamic instability growth on various national ignition facility capsule designs with HYDRA simulations. Phys Plasmas 1998;5(4):1125–32. doi:10.1063/1.872643.

[9] Dawson JM. Particle simulation of plasmas. Rev Mod Phys 1983;55:403–47. doi:10.1103/RevModPhys.55.403.

[10] Larroche O. Kinetic simulation of a plasma collision experiment. Phys Fluids B 1993;5(8):2816–40. doi:10.1063/1.860670.

[11] Sonnendrücker E, Roche J, Bertrand P, Ghizzo A. The semi-lagrangian method for the numerical resolution of the vlasov equation. J Comput Phys 1999;149(2):201–20. doi:10.1006/jcph.1998.6148.

[12] Markidis S, Lapenta G, Rizwan-uddin. Multi-scale simulations of plasma with iPIC3d. Math Comput Simul 2010;80(7):1509–19. doi:10.1016/j.matcom.2009.08.038. Multiscale modeling of moving interfaces in materials

[13] Filbet F, Sonnendrücker E. Comparison of Eulerian Vlasov solvers. Comput Phys Commun 2003;150(3):247–66. doi:10.1016/S0010-4655(02)00694-X.

[14] Kagan G, Tang X-Z. Electro-diffusion in a plasma with two ion species. Phys Plasmas 2012;19(8):082709. doi:10.1063/1.4745869.

[15] Kagan G, Tang X-Z. Thermo-diffusion in inertially confined plasmas. Phys Lett A 2014;378(21):1531–5. doi:10.1016/j.physleta.2014.04.005.

[16] Braginskii SI. Transport processes in a plasma. Rev Plasma Phys 1965;1:205.

[17] Anderson D. Axisymmetric multifluid simulation of high beta plasmas with anisotropic transport using a moving flux coordinate grid. J Comput Phys 1975;17(3):246–75.

[18] Loverich J, Hakim A, Shumlak U. A discontinuous Galerkin method for ideal two-fluid plasma equations. Commun Comput Phys 2011;9(2):240–68. doi:10.4208/cicp.250509.210610a.

[19] Srinivasan B, Shumlak U. Analytical and computational study of the ideal full two-fluid plasma model and asymptotic approximations for Hall-magnetohydrodynamics. Phys Plasmas 2011;18(9):092113. doi:10.1063/1.3640811.

[20] Vold EL. Multidimensional and multifluid plasma edge modelling: status and new directions. Contrib Plasma Phys 1992;32(3â4):404–21. doi:10.1002/ctpp.2150320336.

[21] Simonini R, Corrigan G, Radford G, Spence J, Taroni A. Models and numerics in the multi-fluid 2-D edge plasma code EDGE2D/U. Contrib Plasma Phys 1994;34(2â3):368–73. doi:10.1002/ctpp.2150340242.

[22] Meier ET, Shumlak U. A general nonlinear fluid model for reacting plasma-neutral mixtures. Phys Plasmas 2012;19(7):072508. doi:10.1063/1.4736975.

[23] Laguna AA, Lani A, Deconinck H, Mansour N, Poedts S. A fully-implicit finite-volume method for multi-fluid reactive and collisional magnetized plasmas on unstructured meshes. J Comput Phys 2016;318:252–76. doi:10.1016/j.jcp.2016.04.058.

[24] Cagas P, Hakim A, Juno J, Srinivasan B. Continuum kinetic and multi-fluid simulations of classical sheaths. Phys Plasmas 2017;24(2):022118. doi:10.1063/1.4976544.

[25] Alvarez-Laguna A, Ozak N, Lani A, Mansour NN, Deconinck H, Poedts S. A versatile numerical method for the multi-fluid plasma model in partially- and fully-ionized plasmas. J Phys Conf Ser 2018;1031(1):012015.

[26] Winglee RM. Multi-fluid simulations of the magnetosphere: the identification of the geopause and its variation with IMF. Geophys Res Lett 1998;25(24):4441–4. doi:10.1029/1998GL900217.

[27] AlexashovD., Izmodenov, V.. Kinetic vs. multi-fluid models of H-atoms in the heliospheric interface: a comparison. Astron Astrophys 2005;439(3):1171–81. doi:10.1051/0004-6361:20052821.

[28] Hakim A, Loverich J, Shumlak U. A high resolution wave propagation scheme for ideal two-fluid plasma equations. J Comput Phys 2006;219(1):418–42. doi:10.1016/j.jcp.2006.03.036.

[29] Leake JE, Lukin VS, Linton MG, Meier ET. Multi-fluid simulations of chromospheric magnetic reconnection in a weakly ionized reacting plasma. Astrophys J 2012;760(2):109.

[30] Wang L, Hakim AH, Bhattacharjee A, Germaschewski K. Comparison of multi-fluid moment models with particle-in-cell simulations of collisionless magnetic reconnection. Phys Plasmas 2015;22(1):012108. doi:10.1063/1.4906063.

[31] O'Sullivan S, Downes TP. A three-dimensional numerical method for modelling weakly ionized plasmas. Mon Not R Astron Soc 2007;376(4):1648–58. doi:10.1111/j.1365-2966.2007.11429.x.

[32] Jones AC, Downes TP. The Kelvin–Helmholtz instability in weakly ionized plasmas – II. multifluid effects in molecular clouds. Mon Not R Astron Soc 2012;420(1):817–28. doi:10.1111/j.1365-2966.2011.20095.x.

[33] Benilov MS. Multifluid equations of a plasma with various species of positive ions and the Bohm criterion. J Phys D Appl Phys 1996;29(2):364.

[34] Benilov MS. Multifluid description of nonequilibrium plasmas. In: IEEE international conference on plasma science; 1996. p. 163–4. doi:10.1109/PLASMA.1996.550702.

[35] Sousa E, Shumlak U. A blended continuous–discontinuous finite element method for solving the multi-fluid plasma model. J Comput Phys 2016;326:56–75. doi:10.1016/j.jcp.2016.08.044.

[36] Srinivasan B, Kagan G, Adams CS. Multi-fluid studies of plasma shocks relevant to inertial confinement fusion. J Phys Conf Ser 2016;717(1):012054.

[37] Polak S, Gao X. A fourth-order finite-volume method with adaptive mesh refinement for the multifluid plasma model. American Institute of Aeronautics and Astronautics; 2018. doi:10.2514/6.2018-2195.

[38] Shumlak U, Loverich J. Approximate Riemann solver for the two-fluid plasma model. J Comput Phys 2003;187(2):620–38. doi:10.1016/S0021-9991(03)00151-7.

[39] Baboolal S. Finite-difference modeling of solitons induced by a density hump in a plasma multi-fluid. Math Comput Simul 2001;55(4):309–16. doi:10.1016/S0378-4754(00)00310-4. Nonlinear Waves: Computation and Theory

[40] Kumar H, Mishra S. Entropy stable numerical schemes for two-fluid plasma equations. J Sci Comput 2012;52(2):401–25. doi:10.1007/s10915-011-9554-7.

[41] Shumlak U, Lilly R, Reddell N, Sousa E, Srinivasan B. Advanced physics calculations using a multi-fluid plasma model. Comput Phys Commun 2011;182(9):1767–70. doi:10.1016/j.cpc.2010.12.048.

[42] Berger RL, Albritton JR, Randall CJ, Williams EA, Kruer WL, Langdon AB, et al. Stopping and thermalization of interpenetrating plasma streams. Phys Fluids B 1991;3(1):3–12. doi:10.1063/1.859954.

[43] Rambo PW, Denavit J. Interpenetration and ion separation in colliding plasmas. Phys Plasmas 1994;1(12):4050–60. doi:10.1063/1.870875.

[44] Rambo PW, Procassini RJ. A comparison of kinetic and multifluid simulations of laser produced colliding plasmas. Phys Plasmas 1995;2(8):3130–45. doi:10.1063/1.871145.

[45] Shu C-W, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. J Comput Phys 1988;77(2):439–71. doi:10.1016/0021-9991(88)90177-5.

[46] Shu C-W, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. J Comput Phys 1989;83(1):32–78. doi:10.1016/0021-9991(89)90222-2.

[47] Adams M, Colella P, Graves DT, Johnson J, Keen N, Ligocki TJ, et al. Chombo software package for amr applications - design document Tech. Rep.. Lawrence Berkeley National Laboratory; 2015. LBNL-6616E

[48] Jiang G-S, Shu C-W. Efficient implementation of weighted ENO schemes. J Comput Phys 1996;126(1):202–28. doi:10.1006/jcph.1996.0130.

[49] Balsara DS, Shu C-W. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. J Comput Phys 2000;160(2):405–52. doi:10.1006/jcph.2000.6443.

[50] Laney CB. Computational gasdynamics. Cambridge University Press; 1998. ISBN 9780521625586.

[51] Wolfram Research, Inc. Mathematica, version 11.3, Champaign, IL, 2018.

[52] Sod GA. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. J Comput Phys 1978;27(1):1–31. doi:10.1016/0021-9991(78)90023-2.

[53] Ross JS, Park H-S, Berger R, Divol L, Kugland NL, Rozmus W, et al. Collisionless coupling of ion and electron temperatures in counterstreaming plasma flows. Phys Rev Lett 2013;110:145005. doi:10.1103/PhysRevLett.110.145005.

[54] Chen FF. Introduction to plasma physics and controlled fusion. 2nd ed. Switzerland: Springer International Publishing; 1984. doi:10.1007/978-3-319-22309-4.

[55] Ascher UM, Ruuth SJ, Spiteri RJ. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. Appl Numer Math 1997;25(2–3):151–67. doi:10.1016/S0168-9274(97)00056-1.

[56] Kennedy CA, Carpenter MH. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. Appl Numer Math 2003;44(1–2):139–81. doi:10.1016/S0168-9274(02)00138-1.

[57] Pareschi L, Russo G. Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. J Sci Comput 2005;25(1–2):129–55. doi:10.1007/BF02728986.

[58] Dimits AM, Banks JW, Berger RL, Brunner S, Chapman T, Copeland D, et al. Linearized coulomb collision operator for simulation of interpenetrating plasma streams. IEEE Trans Plasma Sci 2019:1–7. doi:10.1109/TPS.2019.2897790.