Axisymmetric plasma plume characterization with 2D and 3D particle codes

A. Domínguez-Vázquez, F. Cichocki, M. Merino, P. Fajardo, E. Ahedo

Universidad Carlos III de Madrid, 28911 Leganés, Spain

January 21, 2020

Abstract

The expansion of a rarefied axisymmetric plume emitted by a plasma thruster is analyzed and compared with a 3D Cartesian-type and a 2D cylindrical-type simulation code, both based on a particle-in-cell formulation for the heavy species and a simple Boltzmann-type model for the electrons. The first part of the paper discusses the 2D code numerical challenges in the moving of particles, their generation within the cells, and the weighting to the nodes, caused by the radial non-uniformity and the singular and boundary character of the symmetry axis. The second part benchmarks the 2D code against the 3D one for a high-energy, unmagnetized plume with three major species populations (injected neutrals, singly-charged and doubly-charged ions) and three minor species populations (constituted by particles coming from collisional processes, such as the charge-exchange reactions). The excellent agreement found in the results proves that both plume codes are capable of simulating, with a reasonable noise level, heavy particle populations differing by several orders of magnitude in number density. For simulations with a comparable level of accuracy, the 2D code presents a ten-fold gain in computational cost, although the symmetry axis remains its weakest point, due to particle depletion there and the related weighting noise.

1 Introduction

Nowadays, the number of spacecraft (S/C) with onboard electric propulsion (EP) is increasing rapidly. The physics of the ejected high-energy plasma plumes has therefore become a subject of extreme interest, given the criticality of their interaction with the satellite surfaces like solar arrays, especially from a system engineering point of view. In fact, the surface erosion and contamination due to the impact of secondary ions generated within these plumes is a key issue to take into account at a preliminary stage of the satellite design. This has led to the development of a large number of plasma thruster plume simulators [1–11] featuring a detailed modeling of the physics in the near-region of the plume, where most of the slow ion backflow responsible for S/C sputtering and contamination is generated [12].

One such code is EP2PLUS, [12–14], a 3D code that has already been successfully used to analyze different phenomena, such as the plasma plume interaction with a space debris, in the context of the ion beam shepherd concept [15], or the expansion of the plasma plume of either a Gridded Ion Thruster [12]

or a High-Efficiency Multistage Plasma Thruster [16]. EP2PLUS adopts a hybrid approach, opting for a particle-in-cell (PIC) formulation for the heavy species (e.g. ions and neutrals), while relying on a fluid model for the electrons, with a kinetic-type closure at the level of the pressure tensor. In many aspects and problems with rarefied plumes, hybrid codes offer the best trade-off in terms of computational time, generality of geometries and conditions, and reliability of results [12].

Although the interaction of a plasma plume with the satellite is clearly an asymmetric phenomenon that depends on the complex geometry of the S/C, most plasma plumes are quasi-axisymmetric (except perhaps for a small asymmetry introduced by a non-centered neutralizer), so that the computation of their properties, including the critical ion back-flow at the emission plane, could largely benefit from a 2D (axisymmetric) formulation instead of a 3D one. This work attempts to assess these potential benefits, together with the drawbacks arising from cylindrical effects. To this effect an axisymmetric plume code is developed and its performances and results are compared with those of EP2PLUS. The 2D plume code is an adaptation of a 2D hybrid multi-thruster simulation platform, currently under development, for the analysis of the plasma discharge inside the chamber and the very-near plume of various electromagnetic thrusters, such as the Hall-effect thruster (HET) [17] and the Helicon plasma thruster (HPT) [18].

This paper limits the study to unmagnetized plumes, with the electrons modeled as a near-collisionless polytropic fluid [19], and focuses the analysis on the PIC formulation, which is more affected by the change from 3D to 2D and more open to innovative approaches. In fact, the two codes will be tested to operate satisfactorily with up to six heavy species populations with very different densities and energies. Such benchmarking is also beneficial for the 2D multi-thruster simulator, thus further justifying the development of the 2D plume code.

While the 3D PIC module of EP2PLUS operates naturally in a Cartesian spatial mesh, the 2D PIC module of the axisymmetric code uses a 2D cylindrical mesh, which has several important implications on the particle modeling and management. First of all, the cylindrical mesh consists of ring-like cells and the radial coordinate introduces a non-uniformity that needs to be considered in the numerical algorithms. Then, each macro-particle represents a ring of elementary particles (with a uniform distribution along the azimuthal direction), with one rotational (around the symmetry axis) and two translational degrees of freedom, along the radial r and axial z coordinates. Contrary to a uniform Cartesian mesh, the macro-particle contribution to the species density depends strongly on its radius: as it moves radially, its volume of influence (i.e. the volume of the occupied mesh cell) changes, and so does the represented elementary species density.

The radial non-uniformity has also strong implications on the macro-particle weighting scheme: standard weighting schemes like those employed in 3D Cartesian codes [12] can produce systematic errors on the species density. To prevent them, Ruyten [20] proposed density-conserving PIC and

cloud-in-cell (CIC) shape factors in cylindrical coordinates, while Larson [21] applied correction factors to the nodal weighted density. In order to generalize these approaches to a non-uniform cylindrical mesh, Verboncoeur suggested the use of corrected nodal weighted volumes in a generalized weighting scheme [22], which has later been applied to general non-uniform and unstructured meshes with triangular and quadrilateral elements by Vázquez and Castellanos [23] and Araki and Wirz [24].

Secondly, the symmetry axis ($r = 0$) introduces difficulties in particle moving and affects the noise level of the PIC-related statistics. Regarding the macro-particles trajectory integration, the axis $r = 0$ is a singularity in the radial equation of motion, in cylindrical coordinates, which may lead to unphysical accelerations at low radii [25]. As to the PIC noise, the weighted (macroscopic) magnitudes at the nodes in the axis are determined by the macro-particles of only two neighboring cells, instead of the 8 cells considered in a 3D Cartesian mesh [see Fig.1(a)]. In addition, the lower number of cells surrounding the symmetry axis in the 2D cylindrical domain mesh leads to a quicker macro-particle depletion there.

Section 2 of this paper discusses the most suitable algorithms for an axisymmetric PIC code in order to deal with the above mentioned effects of a cylindrical mesh. Then, Sec. 3 cross-validates the 2D code by comparing its maps of the plume expansion with those of EP2PLUS for a high-energy plume, analyzes the spatial evolution of the number of macro-particles per cell, and compares the respective computational times. Finally, the conclusions are drawn in Sec. 4.

2 The axisymmetric code

Like in EP2PLUS, the axisymmetric code assumes plasma quasineutrality except in the very thin Debye sheaths around material surfaces, treated as discontinuities of the electric potential. In this work, the electron density and temperature follow the simple polytropic law

$$T_\mathrm{e}/T_\mathrm{e0} = (n_\mathrm{e}/n_\mathrm{e0})^{\gamma-1} \tag{1}$$

where n_e0 and T_e0 are respectively the electron density and temperature at the reference location where the electric potential ϕ is zero, and γ is the constant polytropic coefficient. For this equation of state, the collisionless electron momentum balance equation leads to an analytical expression for the electric potential [19]:

$$\frac{e\phi}{T_\mathrm{e0}} = \begin{cases} \ln\left(\frac{n_\mathrm{e}}{n_\mathrm{e0}}\right) & \text{if} \quad \gamma = 1 \\ \frac{\gamma}{\gamma-1}\left[\left(\frac{n_\mathrm{e}}{n_\mathrm{e0}}\right)^{\gamma-1} - 1\right] & \text{if} \quad \gamma > 1 \end{cases} \tag{2}$$

The PIC module of the code follows a Lagrangian description for the heavy species populations (ions and neutrals) [12] and its outputs, after each timestep, are the node-weighted densities, fluxes, and temperatures of all the heavy species under consideration. Then n_e is obtained from plasma quasineutrality and Eq. (2) determines ϕ.

Once the electric potential map has been obtained, the electric field $\mathbf{E}$ is computed at the PIC mesh nodes and fed back to the PIC module for the next time step simulation.

The PIC module features two classes of algorithms: *particle-wise* and *mesh-element-wise* algorithms. The former are applied to every macro-particle and include (*i*) interpolation of electromagnetic fields to the macro-particle position, (*ii*) integration of the macro-particle trajectory with a leap-frog algorithm, and (*iii*) macro-particle sorting to the mesh cells. The latter are run for each PIC mesh volume cell or cell-face and involve: (*i*) the collisional processes between the heavy species particles and with the electrons (i.e. charge exchange and ionization collisions), (*ii*) the injection of new macro-particles through the corresponding boundary cell-faces, (*iii*) the volumetric weighting of sorted macro-particles, and (*iv*) the update of the macro-particle generation weight for each simulated heavy species population to appropriately control the number of macro-particles per cell within a specified interval in stationary conditions. All these algorithms have already been described in detail in Ref. [12] and, in the following, only changes on them due to the axisymmetric geometry are discussed.

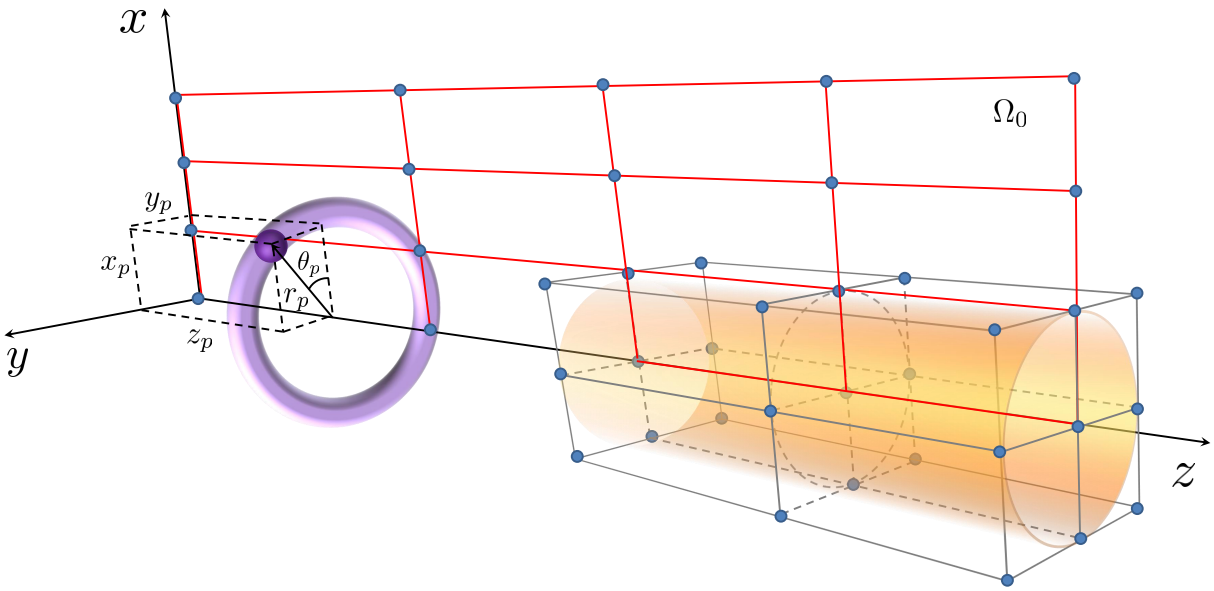
2.1 Axisymmetric mesh

Both 3D and 2D codes use a structured PIC mesh in order to speed up different particle-to-mesh algorithms (such as the particle sorting and weighting to the mesh nodes) and the mesh-to-particle algorithms (such as the interpolation of the electromagnetic fields from the mesh nodes to the particles position). Figure 1(a) shows a rectangular simulation scenario consisting in a 3D (x, y, z) Cartesian prismatic domain and the corresponding cylindrical 2D (z, r) domain identified by the half meridian plane of the 3D circumscribed cylinder, hereinafter referred to as Ω_0. In the general case, the 3D PIC mesh is composed of prismatic cells with quadrilateral cell-faces, while the 2D cylindrical mesh features quadrilateral cells corresponding to 3D annular or conical volumes and their corresponding cell-faces.

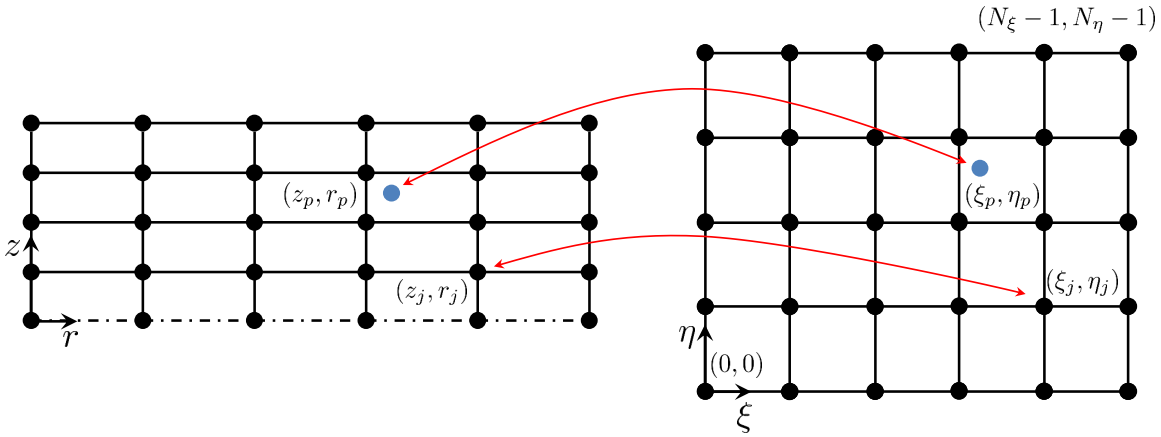
In both cases, it is possible to identify a uniform computational mesh with rectangular elements (cubes in 3D, squares in 2D), with a bijective relation between physical and computational coordinates. As shown in Fig. 1(b) for the 2D case, a given physical point $\boldsymbol{r} = (z, r)$ has computational coordinates $\boldsymbol{\xi} = (\xi, \eta)$, where $\xi \in [0, N_\xi - 1]$ and $\eta \in [0, N_\eta - 1]$ are the computational coordinates taking integer values at the nodes, and N_ξ and N_η are the corresponding number of nodes along each coordinate. Macro-particles are quickly sorted to the PIC mesh nodes from the knowledge of their computational coordinates, since their integer part provides directly the occupied cell indices.

2.2 Particle mover

The Lagrangian description of the heavy species requires computing the trajectory of the p^{th} macro-particle from the integration of Newton equation of motion:



(a)



(b)

Figure 1: (a) Simulation scenario consisting in a 3D (x, y, z) Cartesian prismatic domain and the corresponding cylindrical 2D (z, r) domain identified by the half meridian plane of the 3D circumscribed cylinder, Ω_0. The 3D prismatic cells are compared to the 2D cylindrical ones at the axis $r = 0$. A ring-shaped 2D mesh macro-particle is represented, featuring a 3D Cartesian position (x_p, y_p, z_p) equivalent to the cylindrical position (r_p, θ_p, z_p), being $r_p = \sqrt{x_p^2 + y_p^2}$ and $\cos(\theta_p) = x_p/r_p$, $\sin(\theta_p) = y_p/r_p$. (b) Sketch of the physical and computational 2D structured meshes used for plume simulations. The dot-dashed line indicates the symmetry axis.

$$m_p \frac{d\boldsymbol{v}_p}{dt} = eZ_p \left(\boldsymbol{E}_p + \boldsymbol{v}_p \times \boldsymbol{B}_p \right), \tag{3}$$

where $\boldsymbol{E}_p$ and $\boldsymbol{B}_p$ are respectively the electric and magnetic fields at the macro-particle position ($\boldsymbol{B}_p = \boldsymbol{0}$ in the present work), e is the electron charge, and m_p and Z_p the elementary particle mass and charge number. In EP2PLUS, a 3D particle mover based on the Boris CYLRAD algorithm [26] integrates the ion and neutral macro-particles trajectories, solving Eq. (3) through a generalized second order leap-frog algorithm, which leads to a time shift $\Delta t/2$ between the particles position and velocity, Δt being the PIC integration time step. Therefore, if k is the current time step index, the particles position is known at time k while the particles velocity at time $k-1/2$. Direct integration in cylindrical coordinates yields non-physically large azimuthal accelerations when a particle moves close to the axis $r = 0$ [25]. In order to avoid this problem, the 2D code integrates the particles trajectories using the same 3D Cartesian particle mover algorithm, thus obtaining, for every particle, its 3D Cartesian velocity $\boldsymbol{v} = (v_x, v_y, v_z)$ and position $\boldsymbol{x} = (x, y, z)$. Then, in order to perform the particle sorting and weighting to the PIC mesh nodes, every particle is projected into the 2D plane Ω_0 [see Fig. 1(a)], in which the 2D cylindrical particle position $\boldsymbol{r} = (z, r)$ is obtained taking $r = \sqrt{x^2 + y^2}$).

Thus, the radial turning point of the particle near the axis $r = 0$ is automatically reproduced. The particle computational coordinates $\boldsymbol{\xi} = (\xi, \eta)$ are then computed from its 2D physical position (z, r) and used to sort the particle to the PIC mesh cells and weight it to the corresponding nodes. The computation of the higher order moments of the velocity distribution function (e.g. species fluxes and temperatures) requires to rotate the particle velocity to the plane Ω_0 from the particle actual azimuthal position θ, defined by $\cos\theta = x/r$ and $\sin\theta = y/r$. Note that, at Ω_0, $v_x \equiv v_r$ and $v_y \equiv v_\theta$, being this last velocity component responsible for the azimuthal rotation of the ring-like particles in a 2D cylindrical domain.

2.3 Particle volumetric weighting

The macroscopic magnitudes characterizing the simulated populations, such as the densities, the fluxes, and the temperatures, are obtained by weighting the particle distributions to the nodes of the PIC mesh. This process links the macro-particles with the domain mesh and implies some form of interpolation from the macro-particles to the mesh nodes and viceversa. An ensemble of N macro-particles in a 2D cylindrical domain can be described by the exact macro-particle density

$$n_{\mathrm{d}}(\boldsymbol{r}) = \sum_{p=1}^{N} \frac{W_p}{2\pi r_p} \delta(\boldsymbol{r} - \boldsymbol{r}_p), \tag{4}$$

where $\boldsymbol{r}_p = (z_p, r_p)$ is the position of the p^{th} macro-particle, which actually represents a ring of W_p elementary particles uniformly distributed along the azimuthal direction [see Fig. 1(a)], and $\delta(z, r)$

is the Dirac function (which gives 1 when integrated in a volume containing $\boldsymbol{r}_p$). In PIC codes, the corresponding smoothed particle density at the mesh node j is obtained by integrating in space the exact macro-particle density n_d of Eq. (4) weighted by a nodal shape function $S_j(\boldsymbol{r}')$, and then dividing by the weighting volume ΔV_j associated to the node as

$$n_j = \frac{1}{\Delta V_j} \int_{\mathrm{ROI}} n_\mathrm{d}(\boldsymbol{r}) S_j(\boldsymbol{r}') 2\pi r \, \mathrm{d}r\mathrm{d}z = \frac{1}{\Delta V_j} \sum_p W_p S_j(\boldsymbol{r}'_p), \tag{5}$$

where $\boldsymbol{r}' = \boldsymbol{r} - \boldsymbol{r}_j$ is the physical position relative to the node and the integral extends throughout the nodal region of influence (ROI).

In order to deal with general non-uniform cylindrical meshes, the weighting process is performed in the uniform computational domain considering the bilinear shape function defined in Ref. [27], $S_j(\boldsymbol{\xi}')$, which, for the j^{th} node with computational coordinates $\boldsymbol{\xi}_j = (\xi_j, \eta_j)$, depends on the macro-particles relative computational coordinates $\boldsymbol{\xi}' = (\xi', \eta') = (\xi - \xi_j, \eta - \eta_j)$.

Regarding the weighting volume associated to the nodes, as it was already considered in Ref. [27], systematic errors in cylindrical coordinates are avoided using the corrected weighted nodal volumes of Ref. [22], defined as

$$\Delta V_j = \iint_{\mathrm{ROI}} 2\pi r(\xi, \eta) S(\xi - \xi_j, \eta - \eta_j) |\mathcal{J}(\xi, \eta)| d\xi d\eta, \tag{6}$$

where $|\mathcal{J}(\xi, \eta)| = |\partial(z, r)/\partial(\xi, \eta)|$ is the determinant of the Jacobian matrix, which, along with the radial coordinate $r(\xi, \eta)$, is interpolated from its known values at the nodes using again the same shape function S. [Note that the same integral scheme should be applied to a generic irregular 3D mesh, with the only difference of a now missing $2\pi r$ term, and an extra coordinate ζ.] For the inner nodes of a 3D uniform Cartesian mesh, Eq. (6) (generalized to 3D) yields the constant physical volume of the cells [12] (at the boundaries, the weighting volume reduces with the number of applicable cells).

2.4 Particle distribution and population control

As explained in Ref. [12], new macro-particles are introduced in the simulation domain from the boundary cell-faces (either injection or material wall) and are generated in the bulk domain due to the different collisional processes, the main ones being ionization and charge exchange (CEX) collisions. In both 3D and 2D codes, all these processes involve the generation and distribution of macro-particles inside a given mesh cell, and use an appropriate cell-wise generation weight W_gen.

Let N_gen be the number of new macro-particles to be produced in a given time step inside a given cell, and W_gen the corresponding generation macro-particle weight in the cell. In both 3D and 2D codes, the particles are uniformly distributed inside the regular computational cells, which greatly reduces the computational cost in the general case of non-uniform domain meshes. However, unlike

in the 3D code, in which all particles in the cell are generated with the same macro-particle weight W_{gen}, in the axisymmetric formulation the cylindrical geometry is taken into account by introducing a linearly varying macro-particle weight with the radius within the cell, so that W_{gen} now represents the average macro-particle generation weight in the cell, and the p^{th} macro-particle, generated at the radius r_p, takes the weight:

$$W_p = \frac{r_p}{\overline{r}_{\text{cell}}} W_{\text{gen}} \tag{7}$$

where $\overline{r}_{\text{cell}}$ is the cell mean radius, or the average generation radius for the considered uniform distribution. It is worth mentioning that this radius dependence is considered only for the weight at the generation stage; once generated, the macro-particle conserves its weight until the next collisional event.

In order to limit the PIC noise, especially critical at the symmetry axis of the 2D cylindrical simulation scenario, the EP2PLUS population control is here adapted to the 2D code as well. As explained in Ref. [12], the number of macro-particles per cell is kept within a specified range by updating appropriately the generation weight W_{gen}. This approach can only work properly for those mesh cells featuring a dominant particle generation process (i.e. injection, neutral re-injection due to ion recombination, or collisional processes), and, for a plasma plume simulation like the one shown in Sec. 3, provides satisfactory results with a minimum computational effort (when compared to particle re-sampling or merging/splitting approaches [28, 29]).

The 2D code can then feature different approaches for what concerns the targeted number of macro-particles per cell. While a constant density field n_0 can be reproduced in the Cartesian 3D case with uniform values for the macro-particle weight and number per cell, in a uniform 2D cylindrical mesh, the linearly increasing cells volume with the radius must be taken into account. In order to reproduce the constant density field n_0, two choices are available. The first one, hereinafter named (A), shown in Fig. 2(a), is to set a constant macro-particle weight per cell and let the targeted number of macro-particles per cell vary linearly with the cell volume (or equivalently in this case with the radius). The second approach, referred to as (B), illustrated in Fig. 2(b), considers a constant number of macro-particles per cell in the whole domain, and includes the cylindrical effect on the macro-particle weight.

The performance of both approaches is compared considering two populations: (*i*) a mono-energetic singly charged ion population injected radially with a Gaussian profile [see Eq. (8) below] and with zero temperature, and (*ii*) a thermal neutral population injected with a flat density profile and sonic conditions. Both populations are injected at $z = 0$, between $r = 0$ and $r = 14$cm. The ion injection properties are the same as those of Tab. 1 (except for the zero injection temperature), while the neutrals feature the same injected mass flow as the ions, the same axial fluid velocity and a temperature

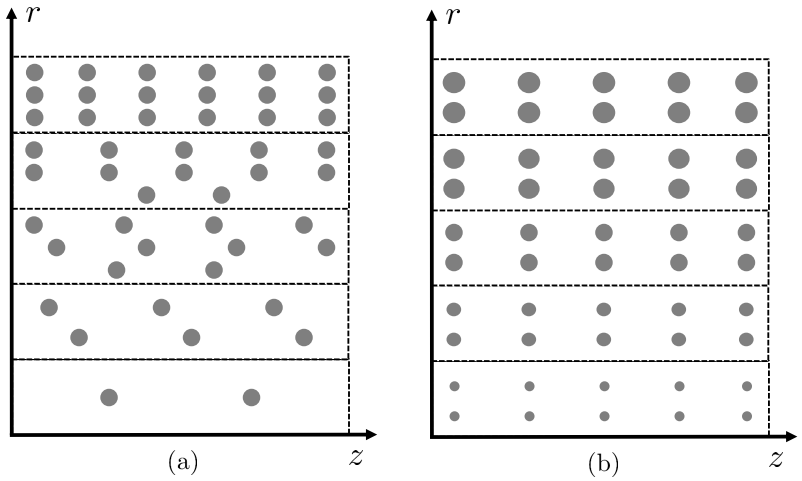


Figure 2: Two different approaches for the global population control. In (a) the macro-particle weight is uniform while the number of macro-particles per cell grows linearly with the cells volume (in this case with the radius). In (b) the number of macro-particles per cell is constant, and the weight grows linearly with the mean cell radius.

$T_\mathrm{n} = (3/5) m_\mathrm{n} u_\mathrm{n}^2$, where m is the neutral elementary mass (xenon is considered) and u_n the injection axial velocity. Both populations and control approaches feature the same average number (5000 here) of macro-particles at injection cells. While this is constant in the approach (B) for all injection cells, the approach (A) features a linearly varying number with the injection cell radius.

The number of macro-particles per cell and the weighted species density obtained with both approaches are shown in Fig. 3. Time-averaged values over 2000 simulation time steps are considered for the number of macro-particles per cell, while instantaneous ones (i.e. at the last simulation time step) are shown for the species densities. As expected, the mono-energetic ion population clearly benefits from the constant number of macro-particles per cell in the approach (B), which, for the same total number of macro-particles in the domain, keeps a higher number of macro-particles in the cells at $r = 0$, and thus helps keeping low the noise level along the axis.

On the contrary, the local particle weight dispersion (i.e. the standard deviation of particle weights in a given cell) in approach (B) leads to a higher noise level in the weighted density at the symmetry axis for the thermal neutrals. Unlike for the injected ions (or ions generated inside an ionization chamber), which are pushed radially outwards from the symmetry axis by the ambipolar electric fields, large neutral macro-particles (injected far from the symmetry axis) can cross radially the domain and reach the symmetry axis, thus producing large variations in the weighted magnitudes there. Contrary to the case of the mono-energetic ions, the number of neutral macro-particles per cell tends to the same value downstream regardless of the considered distribution, albeit with a different weight dispersion. Therefore, the choice of the optimal population control strategy depends on the dynamics of the injected population. Following these conclusions, the results from the 2D code shown in Sec. 3 have

been obtained with the approach (A) for the injected thermal neutral population, and approach (B) for the quasi mono-energetic ion populations.

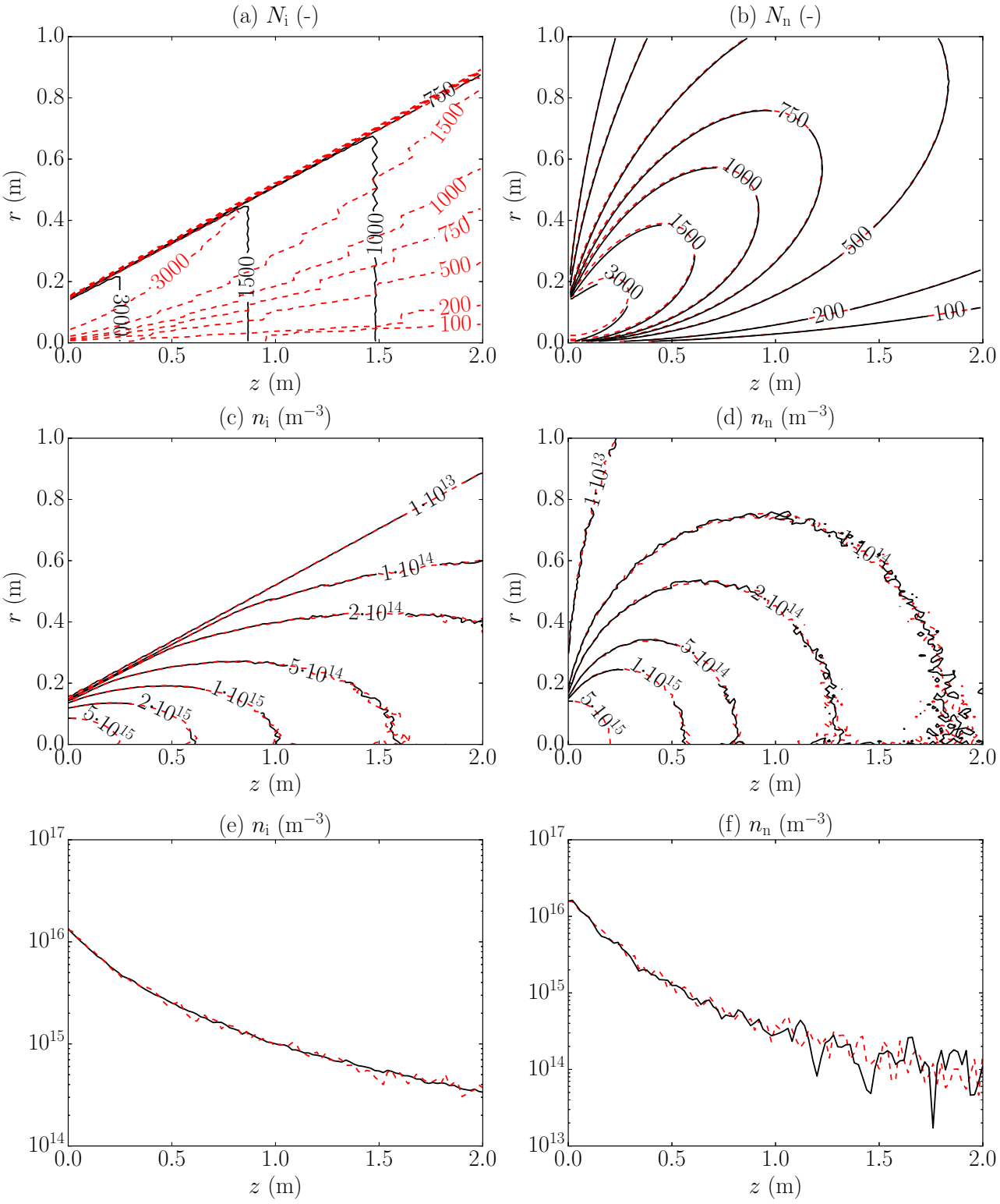


Figure 3: Comparison of population control approaches (A) (red dashed lines) and (B) (black solid lines). Number of particles per cell for (a) mono-energetic ions and (b) thermal neutrals. Weighted particle density for (c) mono-energetic ions and (d) thermal neutrals. Axial evolution of the weighted particle density at the symmetry axis for (e) mono-energetic ions and (f) thermal neutrals.

3 Code benchmarking and discussion of the results

3.1 Simulation settings

The 2D axisymmetric code is validated by simulating an expanding plasma plume similar to that of the NASA's NSTAR gridded ion thruster [30, 31], and comparing its results with those obtained by the 3D EP2PLUS code for the same plume. The simulation domain is a 2 m side cube for the 3D simulation and a cylinder of 1m radius and 2 m length for the 2D one. The mesh spacing, the number of nodes along each direction and the main simulation parameters are listed in Tab. 1. The thruster injection area is circular with a 14 cm radius.

Three different heavy species based on xenon are injected: neutrals, singly-charged and doubly-charged ions. A flat injection density profile and a sonic velocity (based on its own temperature) is considered for the neutrals. On the other hand, the injection profiles of the density n, the axial and radial velocities u_z and u_r of singly- and doubly-charged ions, follow a generalized Parks-Katz radial plume profile [19, 32, 33]:

$$n(r) = n_0 \exp\left(-\frac{C}{2}\frac{r^2}{R_0^2}\right), \quad u_z(r) = u_{z0}\left(\frac{n}{n_0}\right)^{\frac{\gamma-1}{2}}, \quad u_r(r) = u_z \frac{r}{R_0}\tan\alpha_0, \tag{8}$$

where n_0 and u_{z0} are the density and axial velocity at $r = 0$, γ is the electron polytropic coefficient, $C = 5.44$, and α_0 represents the ion streamline divergence angle at the outermost radius $R_0 = 14$ cm. This angle is set to 20.5 and 30 degrees for, respectively, the singly- and doubly-charged ions. While the resulting divergence efficiency matches the values found in the literature, the choice of the divergence angles of both species is not unique: since the ion grid optics of the thruster are presumably optimized for the singly-charged ions, they here feature a lower divergence angle. Regarding the doubly-charged ion current, it amounts to 9.1% of the singly-charged ion current [30, 31].

The following collisional reactions are considered [12]:

- Ionization collisions: $\mathrm{Xe} + e \rightarrow \mathrm{Xe}^{+} + 2e$, $\mathrm{Xe} + e \rightarrow \mathrm{Xe}^{++} + 3e$, and $\mathrm{Xe}^{+} + e \rightarrow \mathrm{Xe}^{++} + 2e$
- Symmetric CEX collisions: $\mathrm{Xe}^{+}(\mathrm{fast}) + \mathrm{Xe}(\mathrm{slow}) \rightarrow \mathrm{Xe}^{+}(\mathrm{slow}) + \mathrm{Xe}(\mathrm{fast})$, and $\mathrm{Xe}^{++}(\mathrm{fast}) + \mathrm{Xe}(\mathrm{slow}) \rightarrow \mathrm{Xe}^{++}(\mathrm{slow}) + \mathrm{Xe}(\mathrm{fast})$

It is underlined that the asymmetric non-resonant CEX reaction [$\mathrm{Xe}^{++}(\mathrm{fast}) + \mathrm{Xe}(\mathrm{slow}) \rightarrow \mathrm{Xe}^{+}(\mathrm{fast}) + \mathrm{Xe}^{+}(\mathrm{slow})$] has not been included since its cross section is much lower than that of the symmetric reactions [34]. The above reactions create populations with very different densities and energies from those of the injected particles. To have an adequate population control and correct macroscopic magnitudes of the minor species, we consider six particle populations, each one with its own computational list. They are detailed in Tab. 2, with their corresponding subscripts.

The time step in Table 1 is set so that, on average, a fast doubly-charged ion crosses no more than one PIC cell per simulation step, while the simulation duration (set to 50000 time steps, equivalent to

15 ms) is sufficiently long to reach stationary conditions for the slowest particle population (injected neutrals, with a residence time of about 8 ms). Regarding the population control algorithm, the targeted number of macro-particles per cell for all populations is 500 (with control range of $\pm 10\%$), except for the fast CEX neutrals, generated deterministically (i.e. directly taking the particle properties of the corresponding colliding fast ion) and therefore featuring no active population control. In fact, the intermittent generation and the mono-energetic distribution of such neutrals makes the monitoring of their macro-particle weight and number nearly ineffective, in terms of noise control. On the other hand, a proper visualization of their density requires an extremely large number of averaging time steps, which is here set to 2000.

Simulation parameter	**Units**	**Value**
3D (x, y, z) mesh number of nodes	-	$201 \times 201 \times 101$
2D (r, z) mesh number of nodes	-	101×101
3D mesh number of cells	-	$4 \cdot 10^6$
2D mesh number of cells	-	10^4
Mesh spacing ($\Delta x = \Delta y \equiv \Delta r$, Δz)	cm	1, 2
Simulation time step	s	$3 \cdot 10^{-7}$
Number of simulation steps	-	50000
Time-averaging steps number	-	2000
Number of macro-particles per cell	-	500
Injected Xe velocity	m/s	247 (sonic)
Injected Xe temperature	eV	0.05
Injected Xe mass flow	mg/s	0.265
Injected Xe^+ kinetic energy	eV	1040
Injected Xe^+ temperature	eV	0.1
Injected Xe^+ mass flow	mg/s	2.40
Injected Xe^{++} kinetic energy	eV	2080
Injected Xe^{++} temperature	eV	0.2
Injected Xe^{++} mass flow	mg/s	0.109

Table 1: Main simulation parameters. The considered meshes are uniform for both 2D and 3D simulations. The number of macro-particles per cell is controlled at injection cells for the injected populations and where collisional effects are not negligible for collisional populations. The reference electron temperature refers to the position $r = 0, z = 6$ cm, slightly downstream of the injection plane, where $\phi = 0$.

Population	Description
i1s	Slow Xe^+ ions from ionization and CEX
i1f	Fast injected Xe^+ ions
i2s	Slow Xe^{++} ions from ionization and CEX
i2f	Fast injected Xe^{++} ions
ns	Slow injected Xe neutrals
nf	Fast Xe neutrals from CEX

Table 2: Different macro-particles populations considered in the simulations, with a dedicated computational particle list for each of them.

Finally, regarding the electrons, we have chosen a polytropic coefficient of $\gamma = 1.2$, consistent with experimental observations and implying a mild cooling along the plume (i.e. the temperature drops by 37% when the density decreases by a factor of 10), and a temperature $T_{e0} = 3.5$ eV at the potential reference node, located 6cm downstream of the injection plane $z = 0$.

3.2 Simulation results and discussion

The results shown in this section refer to time-averaged quantities over 2000 PIC time steps. The electric potential in the meridian plane Ω_0 ($z - r$ plane) is shown for both 2D and 3D simulations in Fig. 4(a). It is monotonically decreasing both axially and radially except for the bump on the side of the main plume, which is due to the formation of CEX ions. Far downstream, when vacuum is reached, the total potential drop is $\gamma T_{e0}/[e(\gamma - 1)] = 21$ V, according to Eq.(2). Such a value is approached in nearly depleted regions of the plasma plume that are located radially outwards at $z = 1$ m. However, at the centerline, 2 m downstream from the injection plane, the potential fall is just around 11 V.

The relative error in the ambipolar electric field (between 2D and 3D simulations) and the field direction are shown in Fig. 4(b). As expected, the electric field vector is directed radially outwards and, at the plume centerline, along the plume expansion. In the areas populated by CEX ions, on the other hand, the axial component of the electric field inverts and produces the well known ion backflow phenomenon [12, 15]. The differences between the 2D and 3D electric field are finally negligible everywhere except at the boundaries of the main plume core, where the number of macro-particles is particularly small, and close to the injection plane boundary at large radii, where the effect of the different simulation box geometry (cylinder versus cube) becomes important.

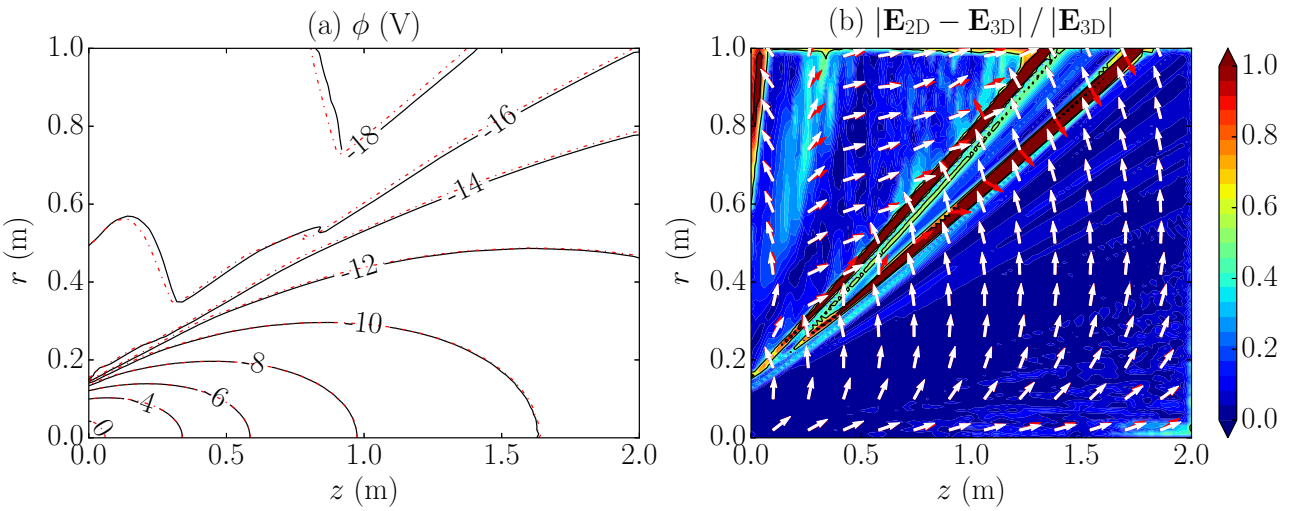


Figure 4: Comparison of (a) electric potential and (b) electric field relative reconstruction error at the Ω_0 plane. In (a), the solid black lines refer to the 3D simulation, while the red dash-dot lines to the 2D simulation. In (b) white and red arrows show the direction of the electric field in respectively the 3D and 2D simulation.

Figure 5 depicts some relevant properties of the six heavy particle populations. All results show a very good agreement between the 2D and 3D codes, and, more importantly, the capability of reproducing with a good level of noise the statistics of populations differing several orders of magnitude in density. In fact the densities of the particle populations produced by collisional events are 2-3 orders of magnitude lower than those of the injected populations.

Xe^+ and the slow CEX Xe^+ densities are shown respectively in Figs. 5(a) and (b). The total Xe^+ density is dominated by the injected ions in most of the domain, except at the lateral plume regions where injected ions are totally absent. Their density is maximum at the injection plane centerline, where it is above 10^{16} m^{-3}, and reduces monotonically downstream because of both the injection divergence angle and the effect of the ambipolar electric field. The CEX singly charged ions are mainly generated in the near-region (within 50 cm axially and 20 cm radially from the injection plane centerline), they reach a peak density above 10^{14} m^{-3} and quickly vanish downstream as they are accelerated radially and axially outwards by the electric fields. At the injection plane, their density decays radially to values as low as $3 \cdot 10^{12}$ m^{-3} at r =1 m.

The total density of doubly charged Xe ions is shown in Fig. 5(c). These ions have a larger injection divergence angle than the singly-charged ones, and, at the outer periphery of the main plume core, they are repelled by the latter, so that they present a local minimum density around 10^{12} m^{-3} at the upper right corner of the simulation domain. It is underlined that this effect has not been observed experimentally, and is clearly related to our choice of the injection divergence angle for the two ion species. The slow Xe^{++} ions, shown in Fig. 5(d), present a peak density of $3 \cdot 10^{12}$ m^{-3} at the centerline, which decreases quickly both radially and axially outwards.

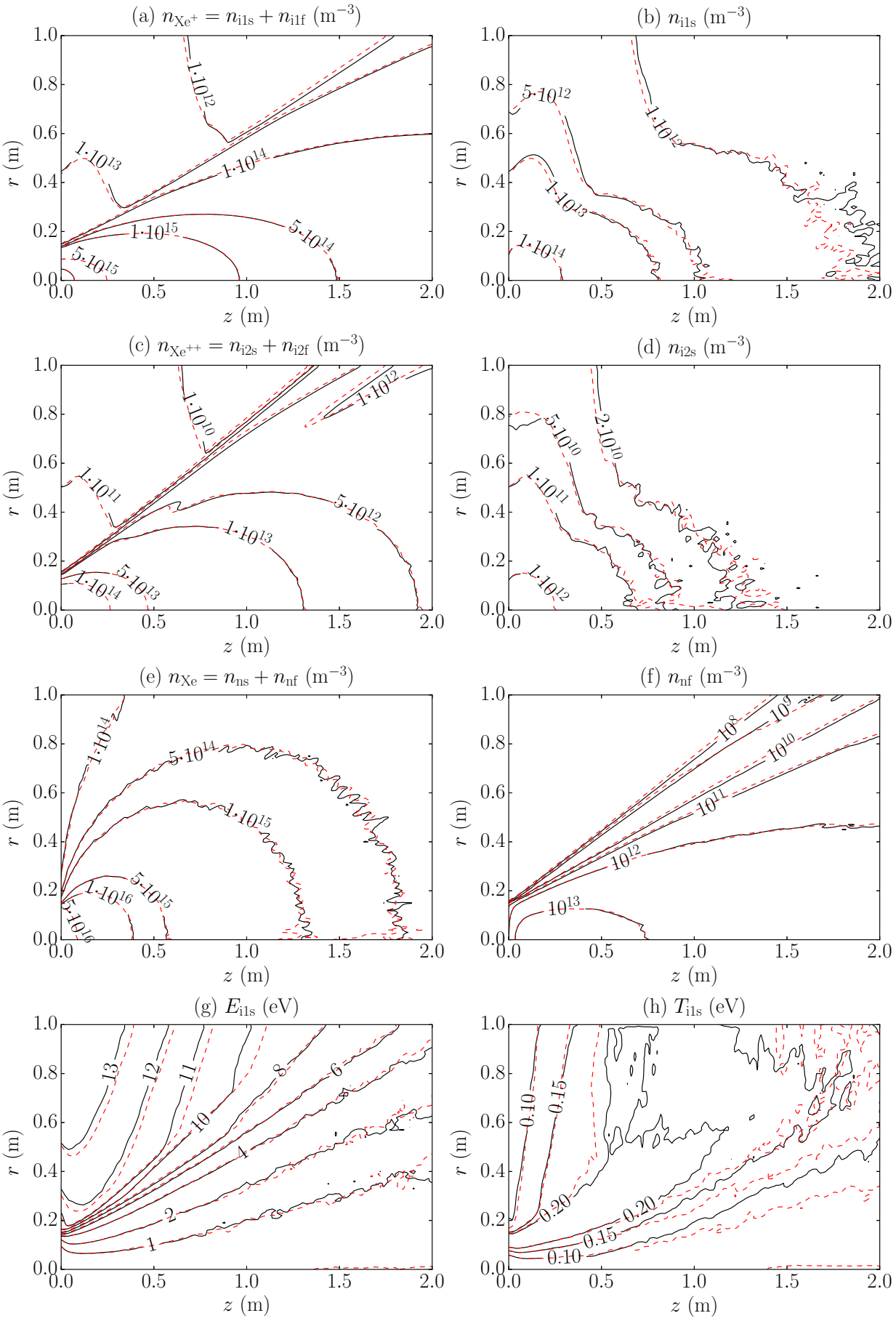


Figure 5: Comparison of (a) total Xe^+, (b) slow Xe^+, (c) total Xe^{++}, (d) slow Xe^{++}, (e) total Xe and (f) fast CEX Xe neutral number densities, and (g) slow Xe^+ average energy (thermal plus average motion) per particle and (h) temperature, at the Ω_0 plane. Both 2D (red dashed lines) and 3D (black solid lines) results are shown.

The total neutral density is shown in Fig. 5(e) and is dominated clearly by the injected sonic population, which expands almost spherically and whose density decays more quickly than that of the injected ions, from $7 \cdot 10^{16}$ m^{-3} at the injection plane centerline, to $4 \cdot 10^{14}$ m^{-3} at the centerline, 2 m downstream. The CEX neutrals, shown in Fig. 5(f), on the other hand, are particularly fast and highly focused so that their density decay is much smaller (a factor of 10 in 2 m axial expansion), and present a peak density above $2 \cdot 10^{13}$ m^{-3} slightly downstream from the injection plane.

Another important property of the plume that the two codes are able to characterize is the average energy of slow ions (thermal plus average motion) generated by CEX collisions, shown in Fig. 5(g). The backflow ions (which produce spacecraft sputtering and contamination) have energies in the order of 12-13 eV on average and tend to migrate quickly radially outwards. In fact, the average ion energy does not increase axially downstream, as it should if the CEX ions were accelerated along the corresponding potential drop [$\sim 5 - 10$ V, as shown in Fig. 4(a)]. This means that the downstream slow ions density is primarily dominated by ions produced in the vicinity, with a very small collision cross section, given the extremely low ion and neutral densities there. Finally, Fig. 5(h) shows the slow ions temperature, which takes relatively low values, below 0.3 eV, compared to their total energy. This means that the CEX ions are a nearly mono-energetic population.

The differences in temperature at larger radii and axial distances can be attributed to a lower number of macro-particles per cell in the 3D case (induced by the setting of a minimum ion generation weight to reduce the total computational cost), which affects the computed statistics.

While the physical plume properties show minimal differences, which validates the 2D simulator, important differences between the 2D and 3D simulations are found in the evolution of the number of macro-particles per cell for both injected populations (ions and neutrals). The number of injected Xe^+ ion macro-particles per cell is compared in Fig. 6(a). While the number of macro-particles per cell is the same at the injection plane (500), the axial decrease in the 3D simulation is much quicker, given the higher dimensionality of the simulation domain (macro-particles have 3 degrees of freedom in position, thus experimenting a larger dispersion). In particular, a number of slightly less than 10 macro-particles per cell is found in 3D, at $z = 2$ m, versus approx. 80 macro-particles per cell in 2D (depending on the considered radius). Nevertheless, the highest value is always found at the centerline for the 3D case, while the 2D case shows a local minimum there. This is also true in Fig. 6(b), which shows the number of macro-particles per cell for the injected neutrals. Since this population is sonic at injection, the expansion is nearly spherical and the number of macro-particles per cell drops much quicker than for the ions. Once again, in 3D we observe a quicker drop in number of macro-particles per cell as the plume expands downstream, and a maximum number always at the symmetry axis. In 2D, on the other and, the axis always features a local minimum in number of macro-particles per cell, with a maximum found along the streamline containing 50% of the total axial particle flow [blue solid

line in Fig. 6(b)].

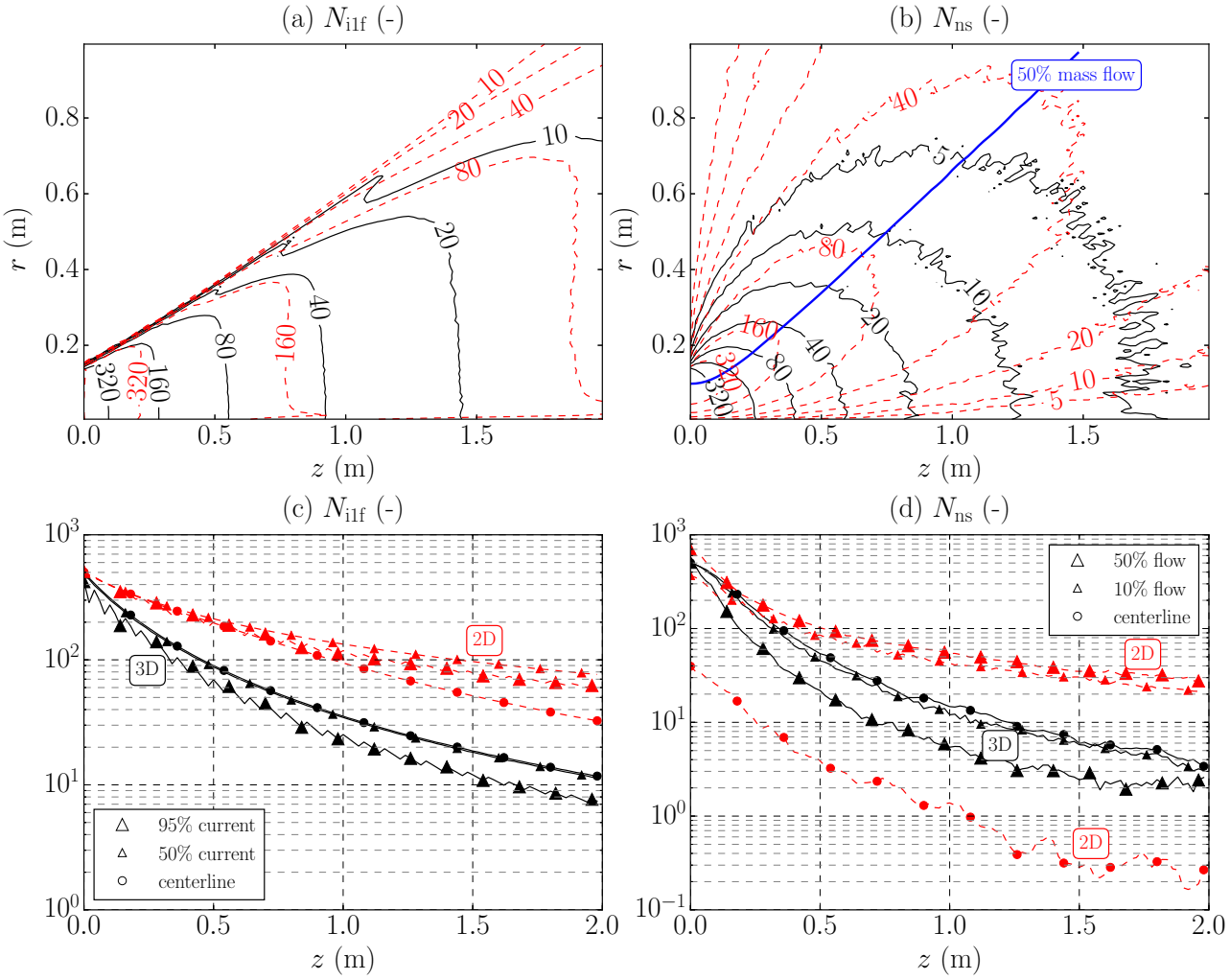


Figure 6: Comparison of the number of macro-particles per cell for (a) injected ions and (b) injected neutrals at the Ω_0 plane, (c) injected ions and (d) injected neutrals along different streamlines. Both 2D (red dashed lines) and 3D (black solid lines) simulation results are shown. In (b), the blue solid line shows the radius of the stream tube containing 50% of the total axial neutral flow. In (c), circles are used for the evolution along the centerline, small upwards triangles for the evolution along the 50% flow streamline and big upwards triangles for the evolution along the 95% flow streamlines. In (d), circles are used for the centerline, small upwards triangles for the 10% axial flow streamline, and big upwards triangles for the 50% axial flow streamline.

The evolution of the number of macro-particles per cell along different ion and neutral streamlines is shown respectively in Figs. 6(c) and (d). The streamlines are identified in terms of the percentage of the total axial particle flow contained in the corresponding streamtube. In the ion case, at $z = 0$ the number of macro-particles per cell is 500, as set by the simulation parameters, for all streamlines. In the neutral case, this is only true for the 3D simulation, since the 2D code follows the population control approach (A), with a constant injection macro-particle weight (see Sec. 2.4), and hence, a linearly variable number of macro-particles per cell (with the cell radius). The centerline depletion of

injected neutrals in the 2D simulation is very clear: the number of macro-particles per cell drops to less than 1 (as time-average), for $z > 1$ m, and this decay is quicker there than along any other streamline, because of the smaller cells volume (which is proportional to the radius in a uniform mesh). Moreover, the 2D simulation axis is more critical than the 3D simulation one because the latter presents a much larger number of injected macro-particles that can potentially cross it downstream. In fact, while the probability of crossing a centerline cell is the same for both 2D and 3D macro-particles (the particle mover algorithm is 3D in both cases), the number of macro-particles injected in the radial interval $\mathrm{d}r$ of the injection plane is much larger in the 3D case (a ring of 3D cells inject macro-particles between r and $r + \mathrm{d}r$, versus only one cell in the 2D case).

Finally, a key feature in the 2D versus 3D comparison is the large difference in computational time, for the same simulation time and comparable PIC noise. The total number of macro-particles of the 3D case is 143 million, versus the 6.48 million particles of the 2D case. This number ratio of 22 is only partially mitigated by the total computational cost per time step per 2D macro-particle, that is approximately twice as large: 6 μs versus 3 μs (using one single thread). Therefore, the 2D code is approximately 10 times quicker in carrying out a plume simulation with a comparable noise level. This time gain is very significant for extensive research activity, considering that each 3D simulation presented here has required a computational time of around a week, using 40 threads in an up-to-date workstation.

4 Conclusions

A new cylindrical axisymmetric code devoted to the simulation of plasma plumes expansion into vacuum has been developed and validated against an already existing 3D plume code. The 2D cylindrical geometry introduces non-trivial difficulties in the particle modeling. The already available 3D Cartesian particle mover is applied to avoid unphysical macro-particles accelerations near the symmetry axis, while the radial expansion is taken into account in both the macro-particles weighting and generation within the domain. Regarding the latter, two different population control approaches have been proposed to limit the noise level, which is especially critical at the 2D domain symmetry axis. It has been found that the optimal algorithm depends on the dynamics of the simulated macro-particle population: mono-energetic populations benefit from a constant target number of macro-particles per cell, while a constant macro-particle generation weight is recommended for thermal populations.

The simulation of a typical plasma plume expansion scenario based on an ion thruster has been considered to compare and benchmark the 2D code against the 3D one, and show their capabilities. An excellent agreement is found between the codes, which are both capable of reproducing, with an acceptable noise level, the properties of heavy particle populations with densities differing by several

orders of magnitude (i.e. the injected and CEX ions populations).

As expected, the simulation of a plasma plume expansion into vacuum greatly benefits from a 2D formulation, which allows for a significant reduction of the computational time (a factor of 10) while keeping a similar PIC statistics noise level. However, the symmetry axis still remains the weak point of the 2D code. Further work will deal with the development of a cell-wise particle rezoning algorithm to better control the noise downstream through the re-sampling of the particle populations from their retrieved distributions. Besides, the particle weighting and population control could benefit from using a domain mesh in the variables (z, r^2), mapped into the computational coordinates (ξ, χ). A bilinear weighting in such computational coordinates would correspond to an area weighting scheme, and the population control could benefit from both a constant number of macro-particles and a constant generation weight per cell. This condition is naturally reproduced with a uniform mesh in (z, r^2), at the cost of a lower resolution at the symmetry axis.

As a final comment, the PIC codes discussed here, with a polytropic closure for electrons, can deal only with unmagnetized plasma plumes, which is the case of ion thrusters and other electrostatic thrusters (e.g. electrospray ones). For Hall-effect thrusters and other electromagnetic thrusters, only the far plume is unmagnetized, while the near-plume and the in-chamber plasma jet present strongly magnetized electrons, but marginally magnetized ions. This means that the 2D-to-3D PIC comparison discussed here continues to be valid in these cases, but the electron model to be coupled with the PIC one, must be changed. Full-2D axisymmetric models for highly magnetized electrons both in the chamber and the near plume of both Hall-effect and Helicon-plasma thrusters are being developed [17, 35, 36]. A 3D model of mildly magnetized electrons, to be matched with EP2PLUS, is under development too [37].

5 Acknowledgments

The work was supported by the Spain's National Research and Development Plan (Project ESP2016-75887). Additional funding came from the CHEOPS project, financed by the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement 730135.

References

[1] M. Celik, M. Santi, S. Cheng, M. Martínez-Sánchez, and J. Peraire. Hybrid-PIC simulation of a Hall thruster plume on an unstructured grid with DSMC collisions. In *28th International Electric Propulsion Conference*, paper 2003-134, Toulouse, France, March 17-21, 2003. Electric Rocket Propulsion Society, Fairview Park, OH.

[2] M.M. Santi. *Hall thruster plume simulation using a hybrid-PIC algorithm.* PhD thesis, Massachusetts Institute of Technology, Boston, MA, 2003.

[3] I. Boyd and J. Yim. Hall thruster plume simulation using a detailed hybrid model. In *40th Joint Propulsion Conference and Exhibit*, paper 2004-3952. AIAA, Reston, VA, 2004.

[4] M. Masselin. Development of a hybrid PIC code for the simulation of plasma spacecraft interactions. Master's thesis, University of Stockholm, Sweden, 2012.

[5] M. Wartelski, C. Theroude, C. Ardura, and E. Gengembre. Self-consistent simulations of interactions between spacecraft and plumes of electric thrusters. In *33rd International Electric Propulsion Conference*, paper 2013-73, Washington D.C., October 7-10, 2013. Electric Rocket Propulsion Society, Fairview Park, OH.

[6] F. Taccogna, D. Pagano, F. Scortecci, and A. Garulli. Three-dimensional plume simulation of multi-channel thruster configuration. *Plasma Sources Science and Technology*, 23(6):065034, 2014.

[7] B. Korkut and D.A. Levin. Three dimensional DSMC-PIC simulations of ion thruster plumes with SUGAR. In *50th Joint Propulsion Conference*, paper 2014-3447, Cleveland, Ohio, July 28–30, 2014. AIAA, Reston, VA.

[8] C. Cai. Numerical studies on plasma plume flows from a cluster of electric propulsion devices. *Aerospace Science and Technology*, 41:134–143, 2015.

[9] D.L. Kahnfeld. Hybrid plume modeling. Master's thesis, Mathematisch-Naturwissenschaftliche Fakultät Ernst-Moritz-Arndt-Universität Greifswald, Greifswald, Germany, 2015.

[10] B. Korkut, Z. Li, and D.A. Levin. 3-D simulation of ion thruster plumes using octree adaptive mesh refinement. *IEEE Transactions on Plasma Science*, 43(5):1706–1721, 2015.

[11] S.J. Araki, R.S. Martin, D. Bilyeu, and J.W. Koo. SM/MURF: Current capabilities and verification as a replacement of AFRL plume simulation tool COLISEUM. In *52nd Joint Propulsion Conference*, paper 2016-4939, Salt Lake City, Utah, July 25-27, 2016. AIAA, Reston, VA.

[12] F. Cichocki, A. Domínguez, M. Merino, and E. Ahedo. Hybrid 3D model for the interaction of plasma thruster plumes with nearby objects. *Plasma Sources Science and Technology*, 26(12):125008, 2017.

[13] F. Cichocki, A. Domínguez-Vázquez, M. Merino, and E. Ahedo. A 3D hybrid code to study electric thruster plumes. In *Space Propulsion Conference*, number 3124968, Rome, Italy, May 2-6, 2016. 3AF, Paris, France.

[14] F. Cichocki. *Analysis of the expansion of a plasma thruster plume into vacuum.* PhD thesis, Universidad Carlos III de Madrid (UC3M), Leganés, Spain, 2017.

[15] F. Cichocki, M. Merino, and E. Ahedo. Spacecraft-plasma-debris interaction in an ion beam shepherd mission. *Acta Astronautica*, 146:216–227, 2018.

[16] D. Kahnfeld, R. Schneider, F. Cichocki, M. Merino, E. Ahedo, J. Duras, and N. Koch. HEMPT thruster discharge and plume simulation with a 2D3v-PIC-MCC and a 3D hybrid fluid-PIC code. In *35th International Electric Propulsion Conference*. Electric Rocket Propulsion Society, Fairview Park, OH.

[17] D. Pérez-Grande, Zhou. J., A. Domínguez-Vázquez, P. Fajardo, and E. Ahedo. Development updates for a two-dimensional axisymmetric hybrid code for plasma thruster discharges. In *35th International Electric Propulsion Conference*, paper 2017-201, Atlanta, Georgia, USA, October 8-12, 2017. Electric Rocket Propulsion Society, Fairview Park, OH.

[18] T. Lafleur. Helicon plasma thruster discharge model. *Physics of Plasmas*, 21(4):043507, 2014.

[19] M. Merino, F. Cichocki, and E. Ahedo. Collisionless plasma thruster plume expansion model. *Plasma Sources Science and Technology*, 24(3):035006, 2015.

[20] W. M. Ruyten. Density-conserving shape factors for particle simulations in cylindrical and spherical coordinates. *Journal of Computational Physics*, 105:224–232, 1993.

[21] D. J. Larson, D. W. Hewett, and A. B. Langdon. Correction factors for PIC accumulation on radial grids. *Computer Physics Communications*, 90:260–266, 1995.

[22] J. P. Verboncoeur. Symmetric spline weighting for charge and current density in particle simulation. *Journal of Computational Physics*, 174:421–427, 2001.

[23] P. A. Vázquez and A. Castellanos. Weighting of charge in PIC codes for unstructured meshes in cylindrical coordinates: application to charged jets. In *CEIDP '05. 2005 Annual Report Conference on Electrical Insulation and Dielectric Phenomena*, pages 499–502, Nashville, Tennessee, USA, October 16-19, 2005. Institute of Electrical and Electronics Engineers, Piscataway, NJ.

[24] S. J. Araki and R. E. Wirz. Cell-centered particle weighting algorithm for PIC simulations in a non-uniform 2d axisymmetric mesh. *Journal of Computational Physics*, 272:218–226, 2014.

[25] C. K. Birdsall and A. B. Langdon. *Plasma physics via computer simulation.* Adam Hilger, Bristol, Philadelphia and New York, 1991.

[26] R.W. Hockney and J.W. Eastwood. *Computer simulation using particles.* CRC Press, Boca Ratón, FL, 1988.

[27] F.I. Parra, E. Ahedo, J.M. Fife, and M. Martínez-Sánchez. A two-dimensional hybrid model of the Hall thruster discharge. *Journal of Applied Physics*, 100(2):023304, 2006.

[28] G. Lapenta. Particle rezoning for multidimensional kinetic particle-in-cell simulations. *Journal of computational physics*, 181(1):317–337, 2002.

[29] R.S. Martin and J.L. Cambier. Octree particle management for DSMC and PIC simulations. *Journal of Computational Physics*, 327:943–966, 2016.

[30] J. Polk, R. Kakuda, J. Anderson, J. Brophy, V. Rawlin, M. Patterson, J. Sovey, and J. Hamley. Validation of the NSTAR ion propulsion system on the Deep Space One mission - overview and initial results. In *35th Joint Propulsion Conference and Exhibit*, paper 1999-2274, Los Angeles, California, June 20-24, 1999. AIAA, Reston, VA.

[31] J.R. Brophy. NASA's Deep Space 1 ion engine (plenary). *Review of Scientific Instruments*, 73(2):1071–1078, 2002.

[32] F. Cichocki, M. Merino, E. Ahedo, D. Feili, and M. Ruiz. Electric propulsion subsystem optimization for "Ion Beam Shepherd" missions. In *34th International Electric Propulsion Conference*, paper 2015-420, Hyogo-Kobe, Japan, July 6-10, 2015. Electric Rocket Propulsion Society, Fairview Park, OH.

[33] D.E. Parks and I. Katz. A preliminary model of ion beam neutralization. In *14th International Electric Propulsion Conference*, paper 1979-2049, Princeton, New Yersey, October 30 - November 1, 1979. Electric Rocket Propulsion Society, Fairview Park, OH.

[34] J.S. Miller, S.H. Pullins, D.J. Levandier, Y. Chiu, and R.A. Dressler. Xenon charge exchange cross sections for electrostatic thruster models. *Journal of Applied Physics*, 91(3):984–991, 2002.

[35] D. Pérez-Grande. *Fluid modeling and simulation of the electron population in Hall effect thrusters with complex magnetic topologies*. PhD thesis, Universidad Carlos III de Madrid (UC3M), Leganés, Spain, 2018.

[36] J. Zhou, A. Domínguez-Vázquez, D. Pérez-Grande, P. Fajardo, and E. Ahedo. An axisymmetric hybrid model for the plasma transport in a Helicon plasma thruster. In *Space Propulsion Conference*, page 00308, Paris, France, 2018. 3AF.

[37] F. Cichocki, M. Merino, and E. Ahedo. A 3d electron fluid model to study magnetic field effects on an expanding plasma thruster plume. In *Space Propulsion Conference*, page 00295, Paris, France, 2018. 3AF.