Step-by-step verification of particle-in-cell Monte Carlo collision codes

Pietro Parodi, Federico Petronio

Step-by-step verification of Particle-in-Cell Monte-Carlo Collision codes

Pietro Parodi[1, 2] and Federico Petronio[3, 4]

[1)]*Aeronautics and Aerospace Department, von Karman Institute for Fluid Dynamics, Waterloosesteenweg 72, 1640 Sint-Genesius-Rode, Belgium.*

[2)]*Department of Mathematics, KU Leuven, Celestijnenlaan 200B, 3001 Leuven, Belgium.*

[3)]*Laboratoire de Physique des Plasmas (LPP), CNRS, Sorbonne Université, École Polytechnique, Institut Polytechnique de Paris, 91120 Palaiseau, France.*

[4)]*COMHET common laboratory by Safran Spacecraft Propulsion, CNRS, Ecole Polytechnique, 91120 Palaiseau, France.*

(*Electronic mail: federico.petronio@lpp.polytechnique.fr)

(*Electronic mail: pietro.parodi@vki.ac.be)

(Dated: 21 November 2024)

Step-by-step verification of PIC-MCC codes

The Particle-in-Cell (PIC) method with Monte-Carlo collisions (MCC) is widely used in the simulation of non-equilibrium plasmas for electric propulsion and laboratory applications. Due to the simplicity of the basic PIC algorithm and the specific modeling needs of the different research groups, many codes have been independently developed. Verification of these codes, i.e. ensuring that the computational code correctly implements the intended mathematical models and algorithms, is of fundamental importance. Different benchmark cases, such as one from Turner *et al.*, Charoy *et al.*, and Villafana *et al.*, have been published in recent years. These have consisted of a complex physical setup, in which many computation modules interact to yield the final result. Although this approach has the advantage of testing the code in a realistic case, it may hide some implementation errors. Moreover, in case of disagreement, the previous works do not provide an easy way to identify the faulty code modules. In this work, we propose a step-by-step approach for the verification of PIC-MCC codes in a 2D-3V electrostatic setup. The criteria for the test cases are (i) they should highlight possible implementation errors by testing the modules separately, whenever possible (ii) they should be free from physical instabilities to avoid chaotic behavior, and (iii) the numerical result should be accompanied by analytical calculations, for confirmation purposes. The 7 test cases identified all show excellent agreement between the authors' codes.

I. INTRODUCTION

With advancements in computational capabilities, numerous codes have been developed for various types of plasma, including space plasmas, fusion plasmas, and laboratory plasmas. Ideally, each code should be validated against experimental measurements, but this task is often extremely challenging. Consequently, in recent years, significant efforts have been made in code-to-code verification to assure code users about the reliability of their tools, both in the plasma physics community and in affine ones[1–3]. In particular, the verification of engineering codes has been identified as one of the fundamental challenges in assisting the development of Electric Propulsion devices[4]. This work proposes several tests to rigorously verify 2D Particle-in-Cell Monte-Carlo collision (PIC-MCC) simulations of low-temperature, low-pressure plasmas.

In the last decades, several efforts of benchmarking 2D PIC codes have been carried out by the low-temperature plasmas community[5,6], as part of the LANDMARK[7] project. However, these efforts proposed rather complex simulation setups, i.e. mimicking the behavior of a Hall Thruster (HT) along different planes. This approach, which has the merit of reproducing physical conditions similar to the ones of experimental devices, does not provide a straightforward way to understand each code's elementary inaccuracies. This is why we have sought to propose a different approach, one that lies between unit testing and mezzanine testing, as previously described by Turner[8].

The growing complexity of numerical models requires a conscientious assessment of the validity of the final results, linked to the correctness of algorithms and models used or to the numerical uncertainty and noise. The pioneering work of Surendra[9] and the more recent of Turner *et al.*[10] dealt with a helium RF discharge, benchmarking it in different pressure conditions. Turner and collaborators, in particular, were able to put together five different codes and show that their results were statistically indistinguishable. Also in helium, Carlsson *et al.*[11] performed a two-code comparison in a parallel-plate glow discharge. Over the past few years, two 2D-PIC benchmark studies in xenon in HT configuration have been published. In the work of Charoy *et al.*[5], seven different codes have been compared in an axial-azimuthal HT configuration. Villafana *et al.*[6], on their side, studied the magnetized discharge inside the channel of such devices, comparing the results of seven groups in a radial-azimuthal configuration. These test cases have shown a good agreement among the different codes in the selected configuration. In particular, the effect of changing the macroparticle's statistical weight has been investigated in both publications. Even though they have demonstrated to correctly capture the instabilities' physics, some important physical mecha-

nisms, e.g. the collision processes, were neglected. Moreover, the final results are the combination of the interaction of a large number of modules and physical phenomena. This can lead to an undesired disguise of implementation errors, that could emerge elsewhere when the simulation setup is changed.

The aforementioned benchmark studies[5,6] demonstrate that simulation outcomes are highly sensitive to the selected parameters. Specifically, these works examine code convergence with respect to the number of macro-particles used. In the same way, the discretization of time and space can significantly influence the final simulation results, potentially leading to inaccurate estimations of macroscopic plasma parameters. Notably, higher order moments of the distribution function, such as the heat flux, are particularly sensitive to the chosen simulation parameters when calculated from the discrete particle distribution. Ensuring convergence is therefore even more critical to obtain accurate estimates of these quantities.

The present work proposes a step-by-step incremental verification of a PIC-MCC code. When designing the test cases, we aimed to provide the reader with some simple and logical tests to challenge their code from basic to more complex phenomena, both physically and numerically. Moreover, whenever possible, we provided analytical calculations to strengthen the confidence in the numerical result. In this way, one can easily identify the defective code parts and avoid error propagation in long simulations. Although the present work cannot remedy the regrettable lack of experimental validation for low temperature plasmas, we believe that having an accurate verification tool for all parts of a PIC-MCC code is fundamental to building confidence in future validation studies and predictive simulations.

Agreement among several codes tested on the same problem is one way to identify the correct solution to that problem. However, the obtained solution is valid only within the numerical framework chosen for the simulations. As mentioned, the previous LANDMARK works[5,6] have started investigating the effect of tightening the numerical constraint in PIC-MCC simulations, particularly analyzing the effect of the macroparticles' weight. Nevertheless, a precise analysis of the error has not been performed. By progressively refining the simulation's numerical parameters, one can apply methods derived from the theory of Richardson extrapolation to estimate the numerical error. In the present work, we apply these techniques to understand the evolution of the solution as the numerical parameters of the simulations are refined. In particular, to verify that the numerical result is converging, and to estimate the numerical error with respect to a fully converged solution.

In this paper, we first in Section II discuss the main features of a PIC-MCC algorithm and present the codes used to perform the simulations. Then, we describe the basic simulation domain and the selected test cases. The results of the simulations are presented and discussed in Section III, while the error quantification procedure is described in Section IV. Finally, in Section V, we provide some conclusions and comments.

II. METHODOLOGY AND DESCRIPTION OF THE TEST CASES

In this section, we briefly describe the basic PIC-MCC algorithm and we provide some details of its implementation in the LPPIC and PANTERA codes, developed respectively at the Laboratoire de Physique des Plasmas (LPP) and at the von Karman Institute for Fluid Dynamics (VKI). Then, we describe the different test cases used for step-by-step verification.

A. The PIC-MCC algorithm

Figure 1 summarizes the algorithm of the typical electrostatic, explicit PIC-MCC code, common to the VKI and LPP implementations. Once the particles are initialized, the charge of the particles is deposited on the grid as ρ_g, then Poisson's equation is solved on the grid for the potential ϕ_g, from which the field $\mathbf{E}_g$ is computed. This is then interpolated onto the particles, giving $\mathbf{E}_p$. The magnetic field $\mathbf{B}_p$ acting on the particles is also computed. Finally, particle positions $\mathbf{x}_p$ and velocities $\mathbf{v}_p$ are updated using a discrete version of Newton's equation of motion with the Lorentz force. At this point of the cycle, collisions are computed using the MCC algorithm. Further details on the PIC and MCC algorithms relevant for the present work are available in the appendices.

The results shown in this work are obtained using the MCC algorithm of Vahedi & Surendra[12], described in Appendix A, for the selection of the colliding particles, since it appears to be the most widely adopted and therefore more appropriate for this benchmarking work. Particles undergoing collisions are always scattered isotropically (i.e. in a hard-sphere fashion), except for backscatter (charge-exchange) interactions, where we simply exchange the particles' velocities.

Different choices can be made in terms of grid type, spatial and temporal discretization of the equations. We will describe the peculiarities of the two codes used in the present work in Sections II A 1 and II A 2.

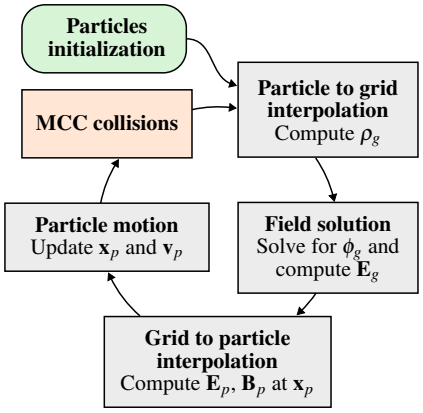

FIG. 1: Flowchart of the typical PIC-MCC algorithm.

1. PANTERA, *the VKI code*

PANTERA is the PIC-MCC-DSMC code whose development started in 2019 at VKI for the simulation of different types of rarefied gas and plasma flows including collisional hypersonic flows[13], the plasma plume generated by laboratory plasma sources[14], and electric propulsion devices[15]. The code uses unstructured grids in 2D and 3D. Poisson's equation (A5) is discretized using the Finite Element Method (FEM) with linear elements for the electric potential. The solution is obtained using the GMRES solver of the PETSc library[16]. Particles are Dirac deltas in both velocity and configuration space ($S_v = S_x = \delta$), which results in linear weighting functions, coincident with the basis functions of the finite elements. For the present work, an explicit time integration is used, where the equations of motion are discretized using the popular Boris-leapfrog scheme[17]. Parallelization of particles and domain is achieved through the Message Passing Interface (MPI). The code uses the Mersenne Twister (MT19937) pseudorandom number generator (PRNG) proposed by Matsumoto & Nishimura[18].

2. LPPIC, *the LPP code*

LPPIC is the PIC-MCC code under development at LPP since 2014. It features a structured Cartesian mesh and solves the Poisson equation, discretized using the finite-differences (FD) method, using an iterative parallel multigrid solver (PFMG) from the open source HYPRE

library[19]. The weighting functions for the particles are of the cloud-in-cell (CIC) type[20]. The particle pusher uses the classical leapfrog scheme with Boris algorithm[17,21]. The PRNG employed is the Fortran 2003 RNG, which is initialized using each CPU's internal clock. The code is parallelized via MPI through domain decomposition. It has been verified with the 1D helium benchmark of Turner *et al.*[10], and also used in the 2D PIC LANDMARK benchmarks[5,6]. The code has been successfully used to simulate the radial-azimuthal[22] and axial-azimuthal[23–26] planes of HTs. Moreover, the code has been also employed to simulate inductively coupled discharges[27].

B. Test cases description

In this section, we give an overview of the setup of all the proposed test cases. First, we describe the computational domain, initial, and boundary conditions that are common to all test cases. All test cases are performed in a 2D square domain of dimension 1 m×1 m, as shown in Figure 2. In all cases where an electric potential is present, the walls of the domain are set at zero potential. The simulation is always initialized with electrons and H^+ ions at a uniform density $n_i = n_e = 5 \times 10^{11}$ m^{-3} with velocities following Maxwell-Boltzmann distributions at temperatures $T_i = 300$ K and $T_e = 11600$ K ≈ 1 eV. All simulations are performed up to a physical time of 5 µs. Particles always vanish upon reaching the boundary of the domain. This represents a conductive wall where electrons are conducted away and ions recombine with an electron and become neutrals. Simulation properties common to all test cases are summarized in Table I, while an overview of all simulations is given in Table II. Physical constants used in the codes are summarized in Table III, and the resulting plasma properties are shown in Table IV. For the results shown in Section III, the grids are discretized with a cell size $\Delta x = 5 \times 10^{-3}$ m, and the time step is set to $\Delta t = 2.5 \times 10^{-9}$ s for both codes. These parameters are sufficiently refined to resolve the Debye length and the electron plasma frequency, respectively, as listed in Table IV. The macroparticle weight F_p is set to 1.25×10^5 m^{-1} in PANTERA, and 1×10^5 m^{-1} in LPPIC. A detailed study of the effect of the statistical weight in this kind of simulation is left for future work. For both codes, we perform 10 simulations of each test case with different PRNG seeds, in order to evaluate the effect of the stochastic scatter.

Notice that the results reported in Section III are extracted at the beginning of each timestep. Moreover, we highlight that no half-timestep is used in plasma diagnostics. If the half-timestep particles' push is used in the diagnostics, the results do not agree with the ones presented in the

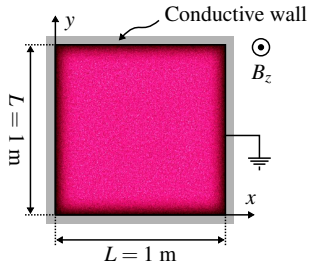

FIG. 2: The 2D simulation domain. The snapshot shows the electron density after 5 µs in a simulation with electric field only.

current paper.

TABLE I: Simulation parameters.

Quantity	Symbol	Value	Unit
Grid size	Δx	5×10^{-3}	m
Time step	Δt	2.5×10^{-9}	s
Electron temperature	T_e	11600	K
Ion temperature	T_i	300	K
Initial electron density	$n_{e,0}$	5×10^{11}	m^{-3}
Initial ion density	$n_{i,0}$	5×10^{11}	m^{-3}
Background gas density	n_g	3×10^{19} or 3×10^{20}	m^{-3}
Total simulation time	t_f	5×10^{-6}	s
Domain size	L	1	m
Magnetic field magnitude	$\lvert\mathbf{B}\rvert$	2×10^{-5} or 2×10^{-4}	T

1. Test case 1 - free effusion

This represents the simplest case, where neither the charge of the particles nor collisions are considered. Particles are therefore free to effuse to the wall according to their initial velocity. Here, we check that particle velocities are correctly initialized, that the mover behaves correctly in the absence of the Lorentz force, and that particles are correctly deleted at the boundary. In addition,

TABLE II: Overview of all simulation test cases.

#	Description	E field	B field	MCC collisions
1	Free effusion	no	$B_z = 0$	no
2	Electric field only	yes	$B_z = 0$	no
3a	Low external magnetic field	yes	$B_z = 2 \times 10^{-5}$ T	no
3b	High external magnetic field	yes	$B_z = 2 \times 10^{-4}$ T	no
4a	Collisions, low pressure	yes	$B_z = 0$	yes, $n_g = 3 \times 10^{19}$ m^{-3}
4b	Collisions, high pressure	yes	$B_z = 0$	yes, $n_g = 3 \times 10^{20}$ m^{-3}
5	Ionization, high-pressure	yes	$B_z = 0$	yes, $n_g = 3 \times 10^{20}$ m^{-3}

TABLE III: Physical constants.

Constant	Symbol	Value	Unit	Source
Electron mass	m_e	$9.1093837015 \times 10^{-31}$	kg	NIST[28]
Hydrogen ion mass	m_i	$1.67262192369 \times 10^{-27}$	kg	NIST[28]
Boltzmann constant	k_B	1.380649×10^{-23}	J/K	NIST[28]
Elementary charge	e	$1.602176634 \times 10^{-19}$	C	NIST[28]

the diagnostics for the number of particles, total momentum, and kinetic energy can be verified.

The initial flux of particles of species α to the wall can be analytically computed, using the

TABLE IV: Derived initial plasma properties.

Constant	Symbol	Value	Unit
Debye length	λ_D	1.0511×10^{-2}	m
Electron plasma frequency	ω_{pe}	3.9891×10^{7}	rad/s
Ion plasma frequency	ω_{pi}	9.3432×10^{5}	rad/s
Electron thermal velocity	$v_{th,e}$	6.6912×10^{5}	m/s
Ion thermal velocity	$v_{th,i}$	2.5112×10^{3}	m/s
Ion acoustic velocity	c_s	9.7853×10^{3}	m/s

formula for the flux of particles from a Maxwell-Boltzmann distribution:

$$\Gamma_\alpha = \frac{1}{4} n_{\alpha,0} \sqrt{\frac{8 k_B T_\alpha}{\pi m_\alpha}} A, \tag{1}$$

where A is the area of the wall, $n_{\alpha,0}$ is the initial number density, T_α is the initial temperature, and m_α is the molecular mass of species α. Inserting the corresponding initial values, we obtain $\Gamma_i = 1.256 \times 10^{15}$ s^{-1} and $\Gamma_e = 3.346 \times 10^{17}$ s^{-1}. This value should correspond to the initial slope of the plot of the total number of particles in time.

2. Test case 2 - electric field only

In this test case, we activate the electric field, therefore we consider the effect of the particles' charge on their motion. Initially, since the plasma is quasineutral and with uniform density, the electric field is very small and only due to stochastic noise from the random initial particle positions. The behavior at the very beginning of the simulation is therefore similar to that of test case 1. In a few electron oscillation periods (ω_{pe}^{-1}), the higher flux of electrons to the wall causes a depletion of electrons and the formation of a sheath at the boundary, which preserves quasineutrality in the plasma bulk. This fast transient in proximity to the walls causes the formation of Langmuir waves that propagate from the sheaths toward the center of the domain.

This test case verifies, in addition to the procedures of the previous test case, the correct implementation of the deposition of the particles' charge on the grid, the correct solution of Poisson's equation, the interpolation of the force on the particles, and the mover with the addition of the electric force.

3. Test case 3a - low external magnetic field, magnetized electrons

In this test case, we have the addition, with respect to test case 2, of an external uniform magnetic field in the direction perpendicular to the simulation plane, $B_z = 2 \times 10^{-5}$ T. Symmetry considerations make it irrelevant to define the direction (positive or negative along z) of the magnetic field vector. The magnetic field amplitude has been chosen such that the electrons start being magnetized ($r_e < L$), while ions are unmagnetized ($r_i > L$). Here, r_j is the gyroradius of the species, which can be computed as

$$r_j = \frac{m_j v_{\perp,j}}{|q| B_z}, \tag{2}$$

with $v_{\perp,j}$ taken as the thermal velocity of species j. Notice that, although being non-magnetized, the Lorentz force including the magnetic field is considered also for the ions. This test case verifies the correct implementation of the mover (i.e., both codes are using the Boris rotation scheme[17]) with the addition of a uniform magnetic field.

4. *Test case 3b - high external magnetic field, magnetized ions*

In this test case, the magnetic field is increased to $B_z = 2 \times 10^{-4}$ T. With this value electrons are well magnetized, and ions start being magnetized ($r_i < L$). The physical consequence is that electrons move in orbits around an almost fixed center of gyration. Those electrons that are within a distance of approximately r_e from the wall are therefore lost within the first gyroperiod. This number of electrons can be estimated as:

$$N_e = [L^2 - (L - 2r_c)^2] n_{e,0} = 3.728 \times 10^{10}. \tag{3}$$

This test case verifies the correct behavior of the mover for highly magnetized electrons and weakly magnetized ions.

5. *Test case 4a - collisions with a low-pressure background gas*

This test case is identical to test case 2, with the addition of MCC between the simulated particles and a background of neutral H gas. The properties of the gas are fixed at $n_g = 3 \times 10^{19}$ m^{-3}, $T_g = 300$ K. We simulate elastic and charge exchange $H^+ - H$, elastic $e - H$, and two electron impact excitation processes, although the latter have almost no effect due to their high activation energy. The cross-section for elastic $e - H$ and electron impact excitation are taken from the Morgan database on LXCat[29], and are shown in Figure 3. The cross sections for $H^+ - H$ elastic and charge exchange collisions are taken from the tabulated data in Schultz *et al.*[30], and are shown in Figure 4. This test case verifies the correct implementation of elastic and charge exchange collisions, which consists of interacting particles with the correct probability and computing correctly the velocities of the scattered particles.

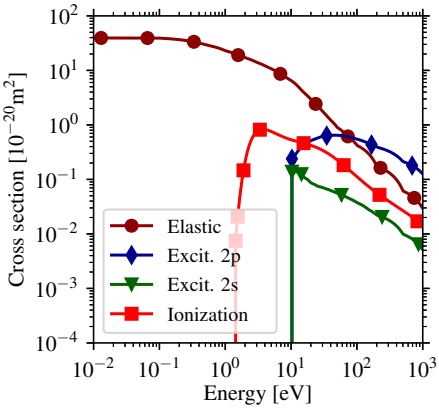

FIG. 3: Cross sections for electron impact processes with the background H gas. The energy is measured as the kinetic energy of both colliding particles in the center of mass frame. The ionization energy has been scaled down by a factor of 10 with respect to the one of Janev *et al.*[31].

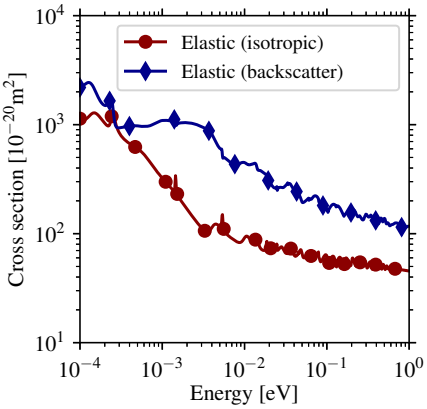

FIG. 4: Cross sections for H^+ ion impact processes with the background H gas. The energy is measured as the kinetic energy of both colliding particles in the center of mass frame.

6. Test case 4b - collisions with a high-pressure background gas

In this test case, the density of the background gas is increased by one order of magnitude, to $n_g = 3 \times 10^{20}$ m^{-3}. This is expected to have a significant effect on the discharge physics, changing

the collision frequencies. It also tests the collision algorithm in a regime where the collision probability approaches the accuracy limit of the algorithm of Vahedi & Surendra[32] (V&S), set at $P_j = 0.1$ (see Appendix A).

7. *Test case 5 - ionization reactions with a high-pressure background gas*

This test case corresponds to the setup of test case 2 with the addition of the electron impact ionization reaction. The cross-section corresponds to the fit of Janev *et al.*[31], where the energy has been scaled down by a factor of 10, as shown in Figure 3. The same scaling has been applied to the activation energy of the reaction, which is also the ionization energy, resulting in $E_{iz} = 1.43$ eV. This has been done so that sufficient reactions can occur without modifying the initial conditions of the test case with respect to test case 2. With this test case, we verify the correct implementation of a reaction with two reactants and three products, where the ionization energy has to be deducted from the reactants and the remaining energy redistributed among the products. In this case, it is always equally shared between the product electrons.

III. CODE-TO-CODE COMPARISON AND ANALYSIS

In this section, we present the simulation outputs of the code-to-code comparison on some fundamental plasma parameters, using the test cases described in Section II B. Firstly, we identify the most relevant plasma parameters to consider. The choice of these parameters has been made taking into account the final benchmarking goal of this paper.

The primary parameter we naturally opted to assess is the total particle count within the system, or equivalently, the average plasma density. One should notice, that the average plasma plasma density must be carefully accounted for at the domain boundaries. This parameter provides insights into the accuracy of particle initialization using a Maxwellian velocity distribution, as well as the efficacy of the particle pusher (and associated modules). The second parameter under consideration is the total kinetic and potential energies of the species, which are presented both individually for each species and collectively. Analyzing the evolution of these energy parameters will enable us to confirm that the system does not undergo any non-physical heating and that the codes experience the same energy relaxation. The potential energy is computed in the two codes

as the sum of the potential energy of each particle, e.g.:

$$\mathscr{E}_p^{\alpha} = \sum_i q_\alpha \phi_i(\mathbf{x}_i), \tag{4}$$

where $\phi_i(\mathbf{x}_i)$ is the electric potential at the position $\mathbf{x}_i$ of the i^{th} particle and α indicates the particle type. In PANTERA, the total potential energy can equivalently be computed as the one stored by the field over the domain:

$$\mathscr{E}_p = \sum_\alpha \mathscr{E}_p^{\alpha} = \frac{1}{2}\varepsilon_0 \int |\mathbf{E}|^2 \, dV, \tag{5}$$

thanks to the properties of the FEM discretization, while the equivalence is not exact for the FD discretization in LPPIC.

Among the parameters that have been inquired, we mention the electrons' and ions' momentum. By analyzing the time evolution of these quantities we noticed that they strongly depend on the initial random velocities of the particles. When averaging over several simulations, the dispersion around the mean value was significant. For this reason, we decided not to include any analysis of the total momentum in this work. However, we provide in the supplementary data in Zenodo[33] the temporal evolution of these quantities. The reader is expected to obtain a similar temporal evolution and order of magnitude. The repository also contains flow field data, including bulk velocity, translational temperatures, and electric potential, of the last time step for all test cases at different levels of refinement, as obtained with the VKI code.

In Figure 5 we show the results for test case 1 (free effusion), where electrons and ions are two non-interacting gases. In (a) and (b) we show the temporal evolution over the simulated interval of the ions' and electrons' normalized densities, respectively. The dotted line represents the theoretical results calculated using Eq. (1). The slow ion motion makes the predicted theoretical evolution valid all along the simulated interval. Conversely, the fast electron loss makes the theoretical predictions of Eq. (1) valid only at the beginning of the simulation, since the hypotheses of uniform distribution in space and Maxwellian velocity distribution from which Eq. (1) is derived become inaccurate for electrons. In this free effusion case, where no wall-sheath can form, within 5 µs only around 3.5% of the initial electrons remain inside the simulation box. The insets show the different results obtained in 10 simulations. The random initialization is responsible for a deviation smaller than 1‰ if compared to the initial density.

Figure 5 (c) and (d) report the temporal evolution of the total energy for ions and electrons, respectively. As one can see, the ion energy decreases linearly. The electron energy experiences

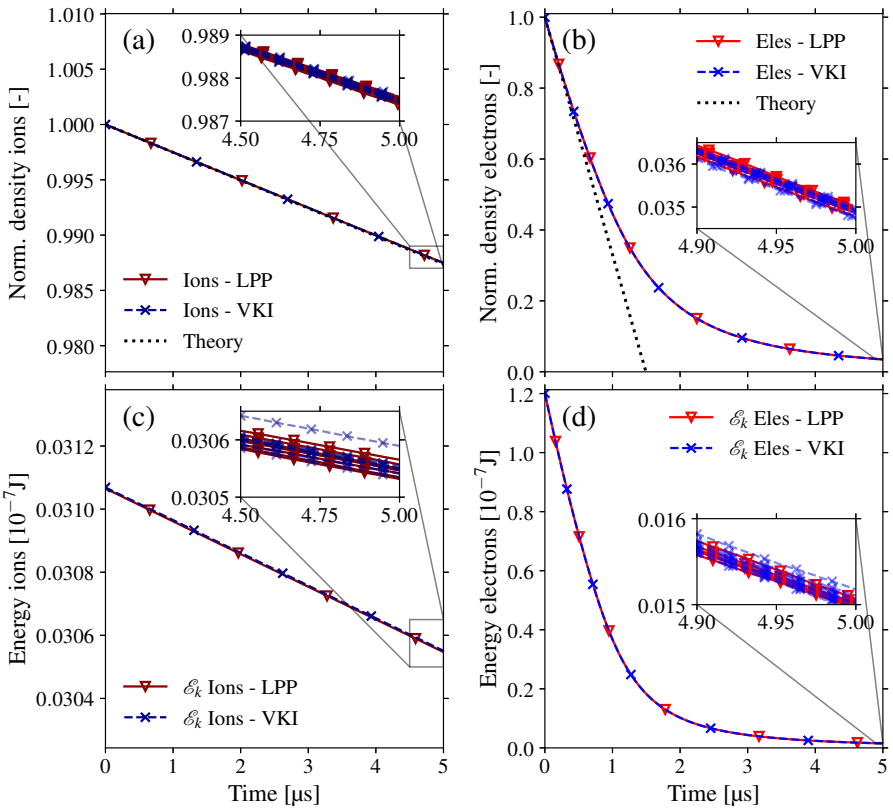

FIG. 5: **Test-case 1**. Normalized mean density for ions (a) and electrons (b). Total kinetic energy of ions (c) and electrons (d). The black dotted line represents the losses from Eq. (1).

a different behavior, since after around 2 μs only 20% of the initial electrons remain inside the simulation box. As before, the dispersion for the energy is lower than 1‰.

In Figure 6 we observe the results for test case 2 (electric field only). Notice that, for the sake of clarity, in this and in the following plots of plasma parameters we show the data averaged over the 10 simulations. One can see in (a) that the presence of the electric field, which causes the formation of the sheaths at the boundaries, increases the ion losses and greatly decreases the electron ones with respect to test case 1. One should notice from (c) and (d) that the potential energy is exchanged between ions and electrons at the electron plasma frequency, i.e. $\omega_{pe} = 3.9891 \times 10^7\,\mathrm{rad/s}$ or $\tau_{pe} = 2\pi/\omega_{pe} = 1.5751 \times 10^{-7}\,\mathrm{s}$. Figure 7 shows the value of the electric

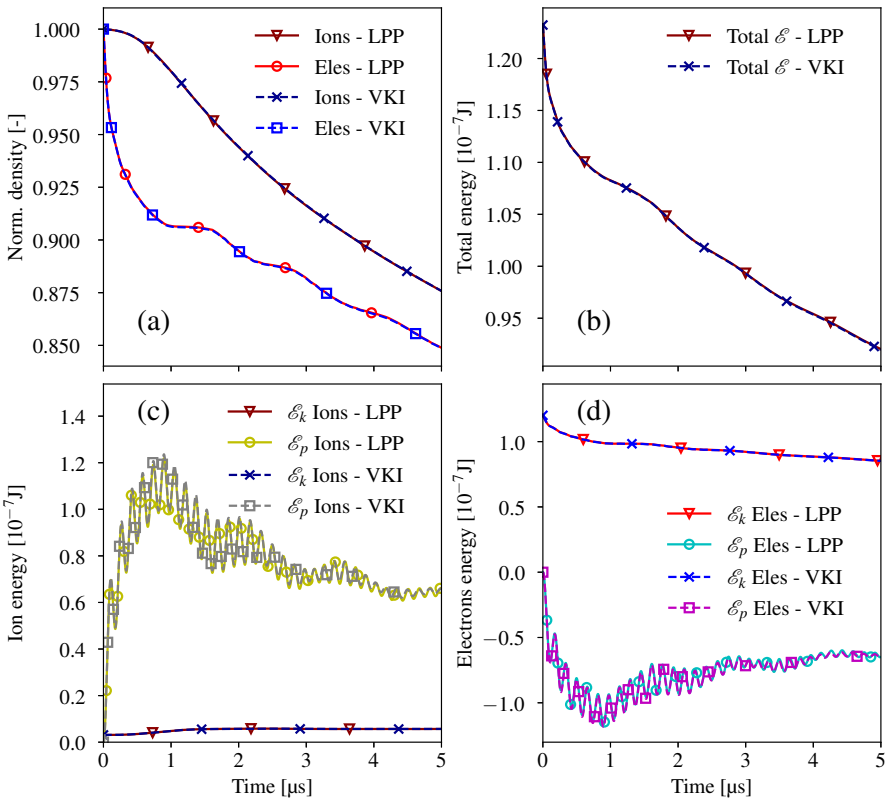

FIG. 6: **Test-case 2**. Normalized mean density for ions and electrons (a). The sum of the total (kinetic + potential) energy for electrons and ions (b). Total kinetic energy of ions (c) and electrons (d).

field E_x along the horizontal line at $y = 0.5$ m in time. One can observe the sheaths forming and increasing in size during time. By applying a 2D discrete Fourier transform (DFT) algorithm to the data in Figure 7, one obtains what is shown in Figure 8. Here, the dispersion relation for Langmuir waves (or electron plasma waves),

$$\omega = \omega_{pe}^2 + \frac{3k_B T_e}{m_e} k^2, \tag{6}$$

is shown with the dashed line. The wave dispersion perfectly superposes the spectrum calculated with the DFT.

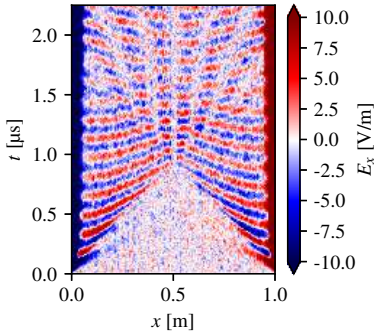

FIG. 7: Plot of the x component of the electric field, E_x, along the horizontal line at $y = 0.5\,\mathrm{m}$ in time. Waves are seen to propagate from the boundaries where the plasma sheath forms towards the interior of the domain, eventually intersecting and forming stationary waves, while decreasing in intensity.

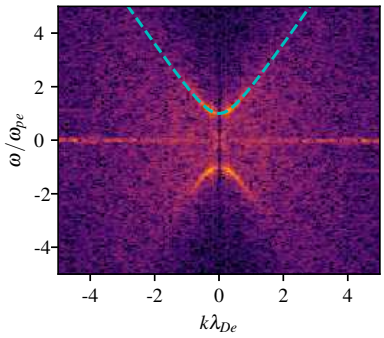

FIG. 8: Numerical dispersion map obtained by computing the 2D Discrete Fourier Transform (DFT) of the same data of Figure 7, after application of the Hann filter. The axes are normalized with the electron plasma frequency, ω_{pe}, and the Debye length λ_{De}. The dashed line represents the dispersion relation for Langmuir waves in Eq. (6).

In Figures 9 and 10 we show the evolution of the plasma parameters when we add to the simulation a low and high magnetic field, respectively (test cases 3a and 3b). Test case 3a shows some not extremely significant differences in the parameters evolution with respect to test case 2. This is consistent with the low magnetization of the plasma in these conditions. On the contrary,

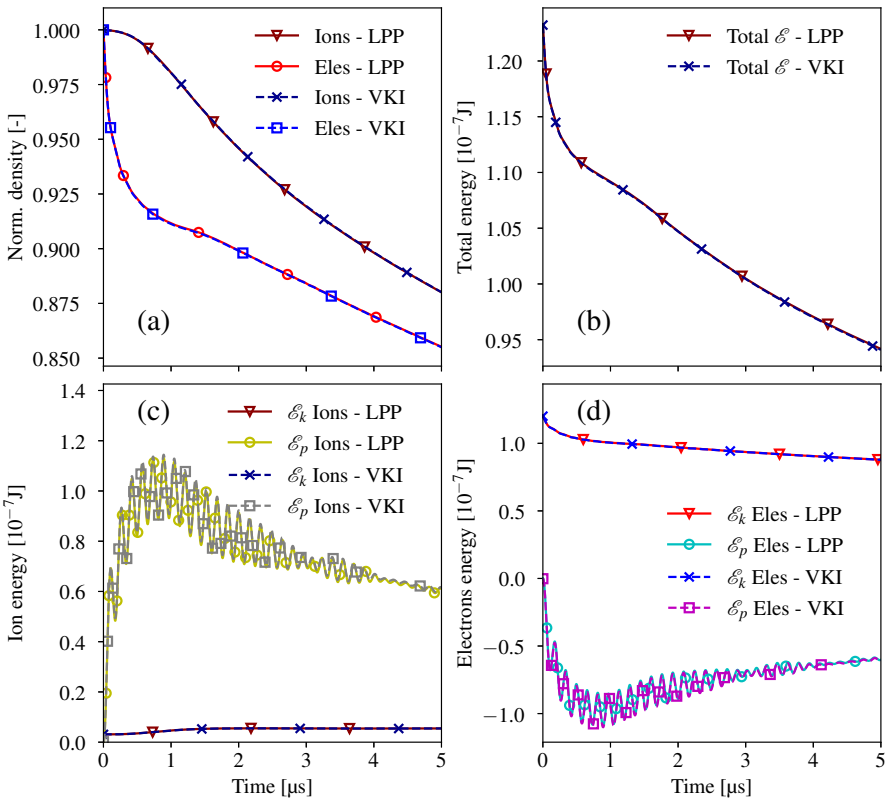

FIG. 9: **Test-case 3a**. Normalized mean density for ions and electrons (a). The sum of the total (kinetic + potential) energy for electrons and ions (b). Total kinetic energy of ions (c) and electrons (d).

when we use a high magnetic field (test case 3b), some strong differences are present. The field causes a decrease in the particle losses since more particles are orbiting around the magnetic field lines, and so they are retained inside the system. We verify that the initial jump in the number of electrons matches well with the analytical calculation of Eq. (3).

As discussed in Section A, collisions play an important and challenging role when simulating a plasma, since any difference in the algorithms may affect the final result. The results of test cases 4a and 4b are reported in Figures 11 and 12. In both cases, the plasma density is affected by the presence of collisions. However, the most visible effect of the collisions is on the particle's

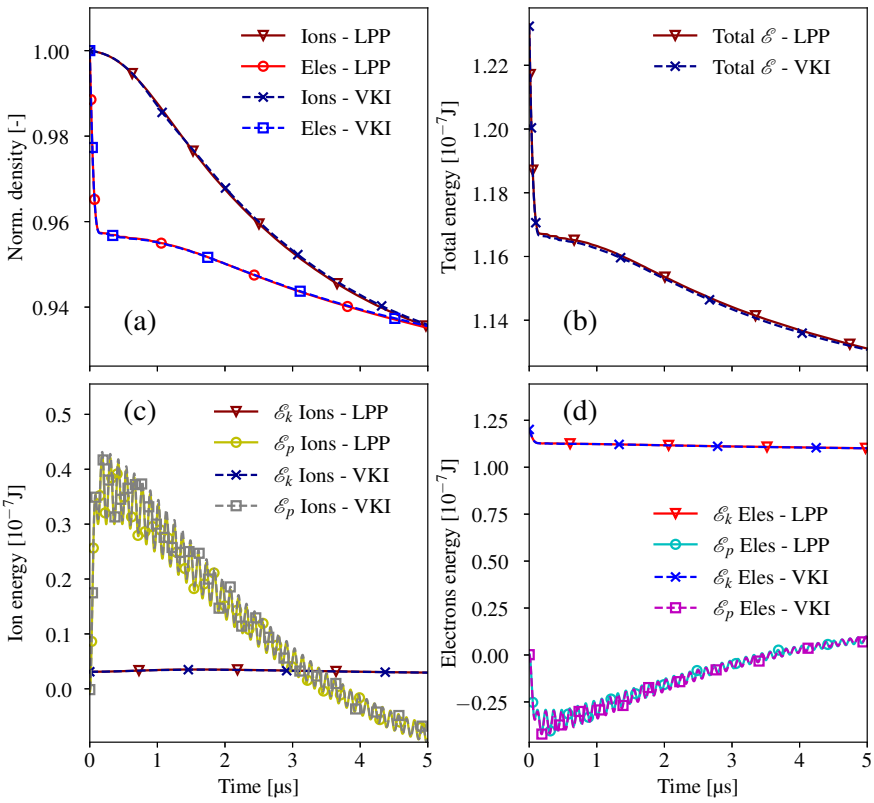

FIG. 10: **Test-case 3b**. Normalized mean density for ions and electrons (a). The sum of the total (kinetic + potential) energy for electrons and ions (b). Total kinetic energy of ions (c) and electrons (d).

potential energy. Even at lower pressure, the plasma frequency oscillations are smoother than in the previous cases. At high pressure, the collisions completely damp these oscillations.

By introducing ionization reactions, we observe some macroscopic variations in the plasma parameters, as shown in Figure 13. Indeed, both the ion and electron densities, after a small drop in the first tens of nanoseconds due to the particle absorption at the walls, increase significantly. The total energy in the system, consistently with the significant part of the energy lost to ionize the background gas, is lower than in the previous cases. Since the elastic and excitation collisions have been switched off, we see that the high-frequency plasma fluctuations reappear in the potential

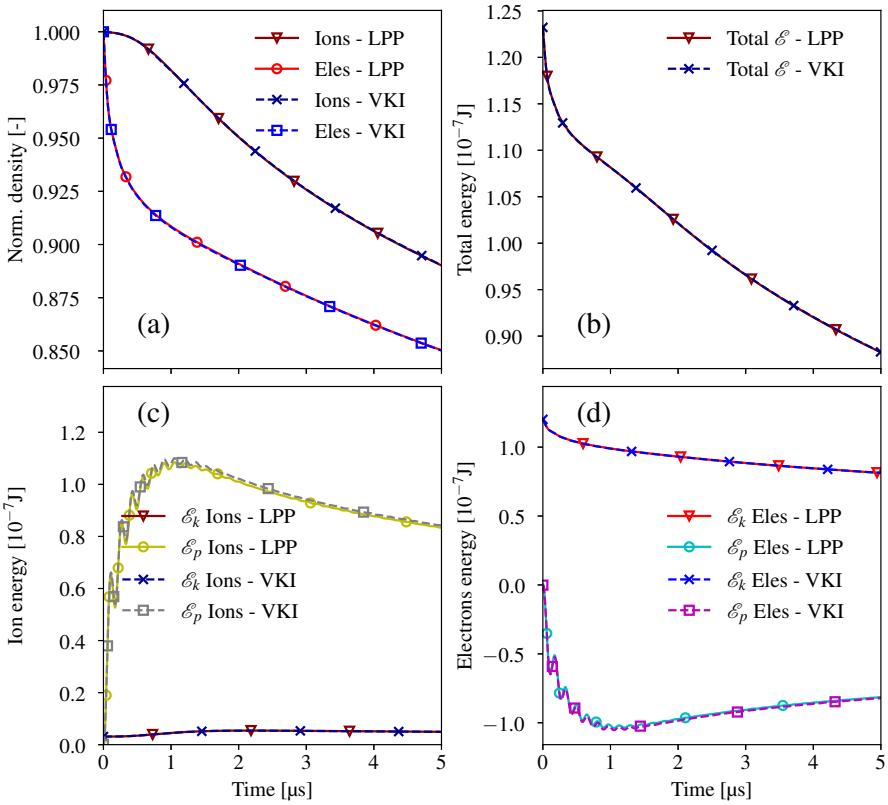

FIG. 11: **Test-case 4a**. Normalized mean density for ions and electrons (a). The sum of the total (kinetic + potential) energy for electrons and ions (b). Total kinetic energy of ions (c) and electrons (d).

energy evolution.

In Figure 14 we show the density profiles of ions and electrons near the walls. These profiles are obtained by taking the last snapshot (not time-averaged) of each test case and by averaging along y between $y = 0.4\,\mathrm{m}$ and $y = 0.6\,\mathrm{m}$. Consistently with the observation made earlier in this section, we observe that the profiles of test cases 2, 3a, and 4a are almost superimposed. Conversely, the sheath dimension is affected by a strong magnetic field and by a higher pressure. In the case of free effusion, we do not have a sheath: the electron density is equally low along all the domain. In case 5, ionization generates a significant increment of the density in the bulk, for both electrons

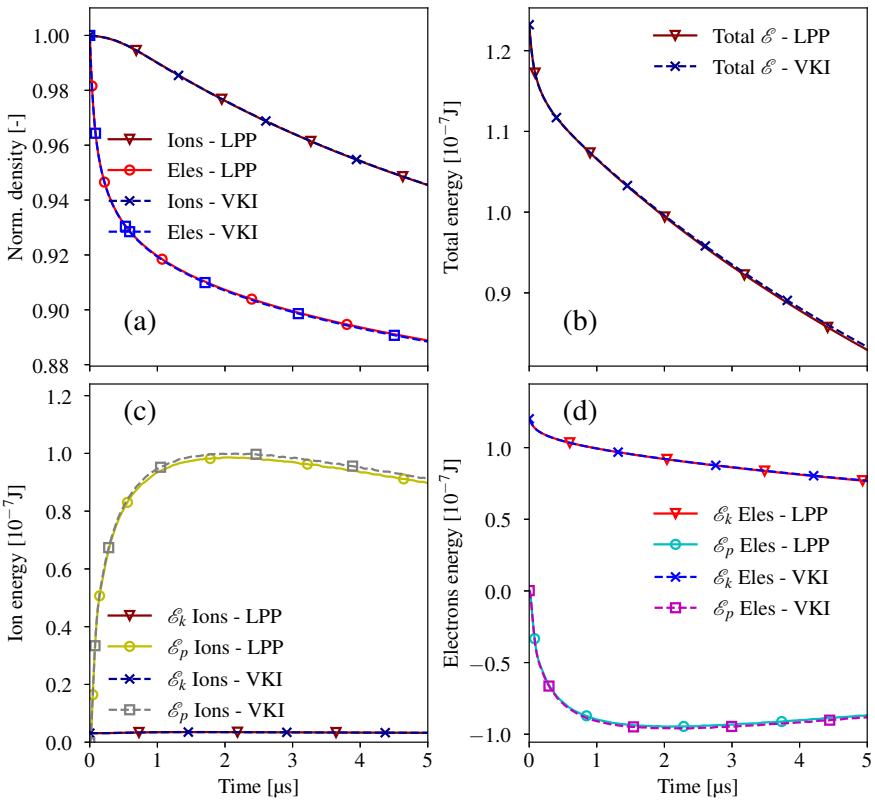

FIG. 12: **Test-case 4b**. Normalized mean density for ions and electrons (a). The sum of the total (kinetic + potential) energy for electrons and ions (b). Total kinetic energy of ions (c) and electrons (d).

and ions.

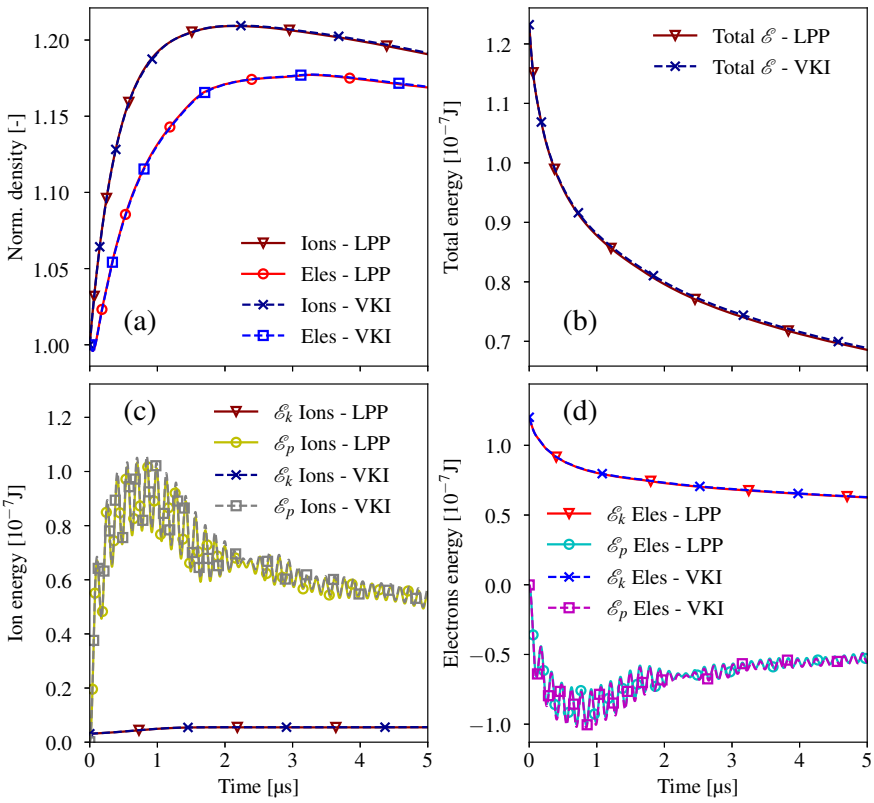

FIG. 13: **Test-case 5**. Normalized mean density for ions and electrons (a). The sum of the total (kinetic + potential) energy for electrons and ions (b). Total kinetic energy of ions (c) and electrons (d).

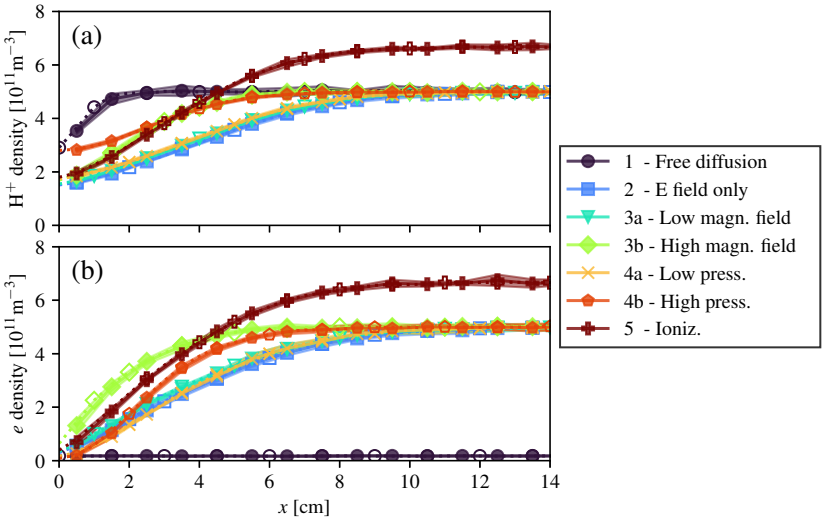

FIG. 14: Density profiles for ions (a) and electrons (b) in the sheath region for the different test cases. These profiles are calculated by averaging in space, and not in time, the final density snapshot between $y = 0.4\,\mathrm{m}$ and $y = 0.6\,\mathrm{m}$. The dotted line reports the results from LPP, while the dashed shows the VKI ones. The shaded area represents a confidence interval around the mean value, obtained as described in Section IV.

IV. NUMERICAL ERROR QUANTIFICATION

In this section, we evaluate the accuracy of the simulation results and their convergence toward the exact solution. It is common, for benchmarks and numerical validation studies for PIC, to consider the stochastic scatter between different realizations of the same simulation to form a confidence band to compare different codes. This is reasonable if the models are implemented similarly in all codes and if the benchmarks are performed with the same numerical parameters. In principle, however, it should also be verified that the results converge towards the exact spatial and temporal profiles, which should not depend on the chosen numerical discretization. In PIC simulations, we identified three numerical parameters that determine the error: the spatial discretization of the grid, Δx, the temporal discretization, Δt, and the macroparticles' weight, F_p.

In classic computational fluid dynamics (CFD), an estimate of the numerical error is typically obtained by performing simulations at multiple discretization levels. The most common method for estimating the fully-converged solution is Richardson extrapolation, and error quantification methods are typically based on it. One such method is the grid convergence index (GCI) method by Roache[34], further refined in the works of Stern *et al.*[35], Wilson *et al.*[36], Xing & Stern[37], and Eça & Hoekstra[38]. Researchers are starting to tackle the additional complications presented by the stochastic nature of PIC simulations: Riva *et al.*[39] developed a methodology based on the method of manufactured solutions, and Radtke *et al.*[40] developed a method based on robust verification. Both methods are adapted for stochastic code outputs. However, the former requires a modification of the model equations, which is outside the scope of the present work. The latter provides detailed statistics of code response, but requires a large number of simulations with different combinations of discretization parameters. In the present work, we will focus on obtaining a sufficiently conservative error estimate using the GCI method and building confidence in the convergent behavior of the simulation response.

As a generalization of Richardson Extrapolation, the GCI method assumes an error model of the form[34]:

$$Y^\star = Y(h_j) + Ch_j^p + \mathcal{O}(h_j^{p+1}). \tag{7}$$

Here, Y is a local or integrated flow quantity. With $Y(h_j)$ we indicate the code response corresponding to the discretization parameter h_j at level j. Quantity $Y^\star$ is the exact solution, which is unknown, p is the effective convergence rate, and C is a constant. The error estimate $\bar{\varepsilon}(h_i) =$

$Y^\star - Y(h_i)$ can be obtained by performing simulations at three different levels of refinement:

$$\bar{\varepsilon}(h_i) = Y^\star - Y(h_i) \approx C h_i^p \approx \frac{Y(h_i) - Y(h_j)}{r^p - 1}, \tag{8}$$

where $r = h_k/h_j = h_j/h_i > 1$ is the discretization parameter ratio, such that $h_k > h_j > h_i$, i.e. h_i corresponds to the finest and h_k to the coarsest grid discretization. The effective convergence rate can be determined from:

$$r^p = \frac{Y(h_j) - Y(h_k)}{Y(h_i) - Y(h_j)}. \tag{9}$$

Roache[34] recommended as an error estimate for $Y(h_j)$, the result at the intermediate level of refinement:

$$\bar{\varepsilon}(h_j) = \frac{r^p}{r^p - 1}\left[Y(h_i) - Y(h_j)\right]. \tag{10}$$

In this work, we refine the time step and particle weight together with the spatial discretization, such that the numerical stability constraints are respected:

$$\Delta x = \Delta x_0 r^\ell, \quad \Delta t = \Delta t_0 r^\ell, \quad F_p = F_{p,0} r^{2\ell}. \tag{11}$$

We choose a grid ratio $r = 2$, and three level of refinement: $\ell \in \{0, 1, 2\}$. The first one, $\ell = 1$, corresponds to the numerical settings described in Section II. We perform 5 replications ($m \in \{1, .., 5\}$) with different PRNG seeds of each test case at the 3 different levels of refinement. Only the VKI code has been used to perform this error quantification study, due to the large number of simulations required.The responses for any given local or integrated flow variable are denoted as $Y_{\ell,m}$. Following the same method as Radtke *et al.*[40], we compute 100 averages (indexed using $b \in \{1, ..., 100\}$) of bootstrap samples, as:

$$\bar{Y}_\ell^b = \frac{1}{5}\sum_{w=1}^{5} Y_{\ell,w'}, \tag{12}$$

where samples are drawn with replacement from the available simulations, i.e. $w' \sim \mathscr{U}[1, ..., 5]$, where $\mathscr{U}$ is the discrete uniform distribution. First, we analyze the effective convergence rate of the total number of ions and electrons in time, by computing Eq. (9) for each of the bootstrapped responses, as

$$r_b^p = \frac{\bar{Y}_1^b - \bar{Y}_0^b}{\bar{Y}_2^b - \bar{Y}_1^b}. \tag{13}$$

As an example, the result for test case 2 is shown in Figure 15. An analogous plot for all test cases is available in the supplementary material in Zenodo[33]. In all the test cases, the response does not

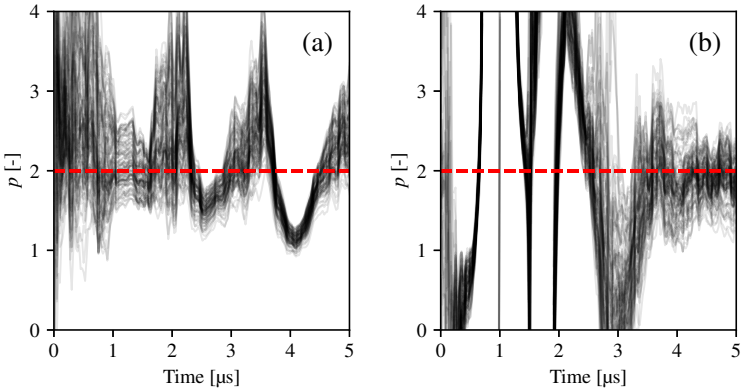

FIG. 15: Effective convergence rate as a function of time, computed from (9) for the mean density of electrons (a) and ions (b) for test case 2, using 100 bootstrap samples. The same plots for all test cases are available in Zenodo[33].

show a clear monotonic convergence behavior until $t = 4\,\mu s$, when the observed convergence rate p tends to 2. Test case 1 (i.e., free effusion) represents an exception, in fact, in the absence of field-particle interaction, the code response only contains stochastic scatter around the mean, and the convergence rate is undetermined. Assuming an effective convergence rate $p = 2$, we compute a confidence interval $\pm\bar{\varepsilon}$ for the mean electron and ion density using Eq. (10), where the maximum difference among all bootstrap samples is chosen, thus incorporating the statistical scatter in the confidence interval, i.e.:

$$\bar{\varepsilon} = \frac{r^p}{r^p - 1} \max_b \left| \bar{Y}_2^b - \bar{Y}_1^b \right|. \tag{14}$$

The error estimate for the mean density tends to increase with time, and does not exceed $2.5 \times 10^{-3}\, n_0$ in all test cases, where $n_0 = n_{e,0} = n_{i,0}$ is the initial number density of electrons or ions. Figure 16 shows the difference between the mean of LPP and VKI code responses for all test cases, and the uncertainty bands show the error estimates. We find that the results from the two codes are in agreement within the computed uncertainty band, with small exceptions in the first part of the simulation, where in fact the convergence rate is not well determined and the error may be slightly underestimated. The same procedure has been applied to the spatial density profiles, for which the confidence bands are shown in Figure 14. Error quantification for the other quantities that have been compared in Section III is left for future work.

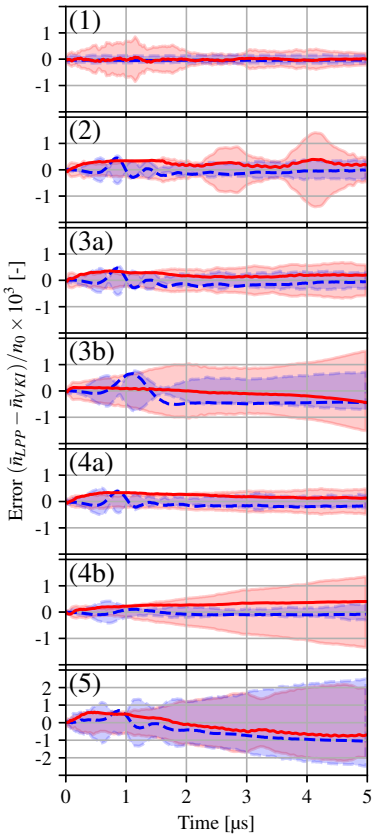

FIG. 16: Plot of the error between the two codes of the mean density of electrons (red solid lines) and ions (blue dashed lines), normalized by the initial number density n_0, as a function of time for all test cases. The error estimate (shaded area) is computed using (9) and reported in red and blue, for electrons and ions, respectively.

V. DISCUSSION AND CONCLUSION

In this work, we presented a general approach to test and verify 2D PIC-MCC codes for low-temperature, low-pressure plasmas. The proposed methods do not require any particular code configuration: they remain very general and can be easily applied to any code. We defined clearly the working framework (physical constants, dimensions, etc.) to avoid discrepancies issued by

slightly different input data. The proposed incremental approach will help the reader identify any failing part of their implementation. Especially, the proposed schemes allow the user to test the majority of a magnetized low-temperature (LTP) code. We highlight the fact that the ones presented here are not simply unit tests: the reader will notice that once a part of the code is verified, it is used in the subsequent test case. This approach helps to verify not only the modules individually but also their coupling.

Even though we have not used these test cases to compare the execution time and performance of each implementation precisely, we emphasize that the computational cost of such an approach remains minimal. Being the execution time of some minutes on a few tens of CPUs, one can easily deduct that these test cases can be run on personal computers or small servers. The present test cases could be used to compare implementation performances. This is, per contra, out of the scope of the current paper and will be left for future work.

The results discussed in Section III show a very good agreement between the codes and the proposed models. Even though the implementation of PANTERA and LPPIC are different (e.g., the grid is unstructured in the former and Cartesian in the latter), we noticed that the time evolution of the analyzed spatially-integrated plasma parameters is substantially the same, i.e. the difference is $\ll 1\%$. We also verified that the random-number-seeds chosen in every single simulation have a limited effect on the plasma evolution, confirming the non-chaotic nature of the test cases.

By applying an error quantification procedure derived from the GCI method, as shown in Section IV, we estimated the error deriving from the numerical discretization. The procedure, commonly applied in computational fluid dynamics, is complicated to some degree by the presence of stochastic noise due to the nature of PIC-MCC. Bootstrapping has been used here to account for the effect of random variations on the distribution of derived quantities such as the rate of convergence. The procedure has shown that the results for the densities of electrons and ions for the two codes are in agreement within the computed uncertainty bounds. Given the small magnitude of the latter, we deem the selected level of refinement sufficient to face the verification goal, which remains the main ambition of this work.

Undeniably, even though we believe to have proposed some tests for the major modules of a 2D PIC-MCC code for LTPs, some physical features remain out of the picture. In particular, we only considered limited plasma chemistry and we did not discuss effects related to magnetic field gradients or plasma heating procedures. However, we believe that those effects could be later added in such a framework to pursue the work started in the current paper.

VI. ACKNOWLEDGMENTS

The research of PP is founded by an FWO Strategic Basic PhD fellowship (reference 1S24022N). The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. for providing computing time on SuperMUC-NG at Leibniz Supercomputing Centre through the project "Heat flux regulation by collisionless processes in heliospheric plasmas – ARIEL". The work of FP was partially supported by Agence de l'Innovation de Defense – AID - via Centre Interdisciplinaire d'Etudes pour la Défense et la Sécurité – CIEDS - (project 2023 – validHETion). The authors thank Nicolas Lequette, Anne Bourdon, Pascal Chabert, Thierry Magin, and Aldo Frezzotti for the fruitful discussion and support, and Alexis Launois for the preliminary work on this topic during his internship at VKI and LPP.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Pietro Parodi: Conceptualization (equal); Investigation (equal); Methodology (equal); Software (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Federico Petronio:** Conceptualization (equal); Investigation (equal); Methodology (equal); Software (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in Zenodo, at http://doi.org/10.5281/zenodo.13839148, reference number 33.

Appendix A: Details on the PIC-MCC algorithm

Particles and field representation

The idea behind the PIC method is that real particles are represented by a smaller number of finite-sized particles, which interact due to their electric charge (i.e. electrostatically in the current work) through the field computed on a grid. The PIC method can be derived from physical intuition, as shown in the books of Birdsall & Langdon[20] and Hockney & Eastwood[41]. Different types of discretization in space and in time have been adopted by the developers of PIC codes, however, the underlying principles remain the same. The distribution function f_α for particles of species α is represented by *macroparticles* with a certain *shape function* S_v in velocity space and S_x configuration space, such that:

$$f_\alpha(\mathbf{x},\mathbf{v},t) = \sum_{p\in\alpha} F_p S_x(\mathbf{x}-\mathbf{x}_p) S_v(\mathbf{v}-\mathbf{v}_p), \tag{A1}$$

where the sum is performed over all macroparticles p of species α, with position $\mathbf{x}_p$, velocity $\mathbf{v}_p$, and particle weight F_p. Alternatively, the particle weight factor can be defined as

$$F_p = \frac{N_{\text{real}}}{N_{\text{macro}}},$$

with N_{macro} the number of macroparticles, corresponding to a number N_{real} of real particles. Typically, a Dirac delta function is used in velocity space $S_v = \delta$, while b-splines of various orders are typically used in configuration space. This family of functions with compact support includes the Dirac delta function as well as the multi-dimensional equivalent of "hat" functions. One can show that the evolution of the distribution function according to the Vlasov equation corresponds to particles moving according to the Newton-Lorentz equations of motion,

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_p(t) \tag{A2}$$

$$\frac{d\mathbf{v}_p}{dt} = \frac{q_p}{m_p}[\mathbf{E}(\mathbf{x}_p) + \mathbf{v}_p \times \mathbf{B}(\mathbf{x}_p)]. \tag{A3}$$

Where $\mathbf{E}(\mathbf{x}_p)$ is computed as the value of the electric field on grid element g interpolated to the particle p through a weighting function W^E_{pg}, as

$$\mathbf{E}_p = \sum_g W^E_{pg} \mathbf{E}_\mathbf{g}. \tag{A4}$$

The electric potential ϕ is computed on the grid, by discretizing and solving Poisson's equation, which reads:

$$\nabla \cdot (\varepsilon_r \nabla \phi) = -\frac{\rho}{\varepsilon_0}. \tag{A5}$$

The charge density ρ in the previous expression is accumulated on grid nodes also through a weighting function W_{pg}^{ρ},

$$\rho_g = \sum_p W_{pg}^{\rho} F_p q_p. \tag{A6}$$

Typically, these weighting functions have the same functional form (i.e., $W_{pg}^{E} = W_{pg}^{\rho}$), so to minimize the force that a particle exerts on itself due to interpolation errors. Within this framework, collisions and reactions between the particles and a background gas can be included by MCC techniques.

MCC collision algorithm

In weakly ionized plasmas, collisions between particles and the neutral background gas are relevant since they represent the mechanism through which the discharge is sustained. Collisions are usually included in PIC codes by a specific module, which mimics the interaction of simulated particles with a background gas. The usual MCC algorithm used in the low-temperature plasma community[10,42,43] is the one proposed by Vahedi & Surendra[12] (V&S), appearing also in the work by Birdsall[44]. Similar algorithms were proposed by Nanbu[45]. Notice that an entire set of algorithms sharing many similarities with the previous ones, but meant for collisions between two simulated particles, originates from the Direct Simulation Monte-Carlo (DSMC) community. These are, for instance: the Time-Counter, No-Time-Counter (NTC), Majorant frequency scheme, and Bernoulli trials algorithms. A summary of these algorithms and their origin is given by Roohi & Stefanov[46]. We discuss below a few details of the MCC algorithm that could potentially have a significant effect on the accuracy of the results.

We consider a certain number of collisional and reactive processes with their cross sections $\sigma_1, ..., \sigma_N$, and particles that can undergo at most one collision per time step. In the algorithm of V&S[12], first, a "null" collision frequency ν' is estimated as an upper bound for the combination of all collisional processes, as:

$$\nu' = \max_x(n_g(x)) \max_{\mathscr{E}}(\sigma_T(\mathscr{E})v), \tag{A7}$$

with

$$\sigma_T(\mathscr{E}) = \sum_j \sigma_j(\mathscr{E}) \tag{A8}$$

the total cross section, n_g the neutral target density, and v the relative velocity of the colliding particles in the center-of-mass reference frame (i.e. $v = \sqrt{2\mathscr{E}/m_r}$, with $m_r = m_1 m_2/(m_1+m_2)$ the reduced mass of the colliding particles, which have mass m_1 and m_2). This collision frequency is then used to select particles to be tested for collisions with a probability:

$$P_{\text{null}} = 1 - \exp(-\nu'\Delta t). \tag{A9}$$

Then, every single collisional process occurs with a probability:

$$P_i(\mathscr{E}) = \frac{n_g \sigma_i(\mathscr{E}) v}{\nu'}. \tag{A10}$$

Therefore, at each time step, the number of particles undergoing process i is, on average:

$$N_{\text{MCC}} = N_p[1 - \exp(-\nu'\Delta t)]\frac{n_g\langle\sigma_i(\mathscr{E})v\rangle}{\nu'}. \tag{A11}$$

As noted in the original paper[12], the algorithm is accurate in the limit of $\nu'\Delta t \ll 1$. This essentially imposes a limit on the time step. For this verification work it is especially relevant to notice that, using the algorithm as originally described, the number of collisions depends on the chosen value of ν'. Different codes may use various methods, more or less conservative, to estimate this value. This can introduce differences in the simulated collision frequency of $\mathscr{O}(\nu'\Delta t)$ between codes. Therefore, we replace Eq. (A10) with the following expression:

$$P_i = \frac{n_g \sigma_i(\mathscr{E}) v \Delta t}{P_{\text{null}}}, \tag{A12}$$

such that the number of particles undergoing process i does not depend on the chosen value for ν', and is, on average:

$$N_{\text{MCC}} = N_p n_g \langle\sigma_i(\mathscr{E})v\rangle\Delta t. \tag{A13}$$

This way, the value of ν' becomes merely a factor that optimizes the efficiency of the algorithm, without affecting its result.

Appendix B: Notes on collision algorithms of Vahedi & Surendra and the classic DSMC

We aim here to clarify some fundamental discrepancies between the MCC algorithm of V&S typically employed in combination with PIC and the NTC algorithm typically employed in DSMC,

which became evident during the preparation of this work. A collisional process i for a particle that traverses a background gas at a constant velocity can be considered a Poisson point process. Collisions are events that can occur at any time, but with a definite frequency, that can be determined considering a long enough time interval:

$$\nu_i = \lim_{t\to\infty} \frac{N_{\mathrm{coll}}(t)}{t}, \tag{B1}$$

where $N_{\mathrm{coll}}(t)$ is the number of collisional events in the time interval $[0,t)$. Considering a finite time interval $[0,T)$, the probability that a certain number n of collision events has occurred is[47]:

$$P\{N_{\mathrm{coll}}(T)=n\} = \frac{(\nu_i T)^n}{n!}\exp(-\nu_i T). \tag{B2}$$

If the same time interval is divided into M time steps of duration $\Delta t = T/M$, the probability of exactly n collision events occurring in the NTC and V&S algorithms are:

$$P^{\mathrm{NTC}}\{N_{\mathrm{coll}}(T)=n\} = (\nu_i\Delta t)^n(1-\nu_i\Delta t)^{M-n}\binom{M}{n}, \tag{B3}$$

$$P^{\mathrm{V\&S}}\{N_{\mathrm{coll}}(T)=n\} = [1-\exp(-\nu_i\Delta t)]^n[\exp(-\nu_i\Delta t)]^{M-n}\binom{M}{n}. \tag{B4}$$

These show that the two algorithms have two different but important properties: the algorithm of V&S always reproduces the correct probability of a particle not incurring into a collision, $P\{N_{\mathrm{coll}}(T)=0\}$, while the NTC algorithm always reproduces the expected value of the number of collisions of the true distribution, $E[N_{\mathrm{coll}}]$ (and therefore the physical collision frequency), of course as long as this is smaller than the number of time steps M. The difference between the two algorithms may not be negligible, especially when $\nu_i\Delta t \ll 1$ is not verified.

REFERENCES

[1]American Institute of Aeronautics and Astronautics, "AIAA Guide for the Verification and Validation of Computational Fluid Dynamics Simulations," American Institute of Aeronautics and Astronautics (1998).

[2]B. Bagheri, J. Teunissen, U. Ebert, M. M. Becker, S. Chen, O. Ducasse, O. Eichwald, D. Loffhagen, A. Luque, D. Mihailova, J. M. Plewa, J. Van Dijk, and M. Yousfi, "Comparison of six simulation codes for positive streamers in air," Plasma Sources Science and Technology **27**, 095002 (2018).

[3]E. N. Tinoco, "CFD Uncertainty and Validation for Commercial Aircraft Applications," in Meeting Proceedings RTO-MP-AVT-147, Paper 20. Neuilly-sur-Seine, France (NATO, 2007) pp. 20–1 – 20–36.

[4]A. Bourdon, T. E. Magin, J.-L. Cambier, M. Torrilhon, and G. Lapenta, "TR-AVT-294, Towards Improved Computational Tools for Electric Propulsion," Tech. Rep. (NATO Science and Technology Organization, 2022).

[5]T. Charoy, J. P. Boeuf, A. Bourdon, J. A. Carlsson, P. Chabert, B. Cuenot, D. Eremin, L. Garrigues, K. Hara, I. D. Kaganovich, A. T. Powis, A. Smolyakov, D. Sydorenko, A. Tavant, O. Vermorel, and W. Villafana, "2D axial-azimuthal particle-in-cell benchmark for low-temperature partially magnetized plasmas," Plasma Sources Science and Technology **28**, 105010 (2019).

[6]W. Villafana, F. Petronio, A. C. Denig, M. J. Jimenez, D. Eremin, L. Garrigues, F. Taccogna, A. Alvarez-Laguna, J. P. Boeuf, A. Bourdon, P. Chabert, T. Charoy, B. Cuenot, K. Hara, F. Pechereau, A. Smolyakov, D. Sydorenko, A. Tavant, and O. Vermorel, "2D radial-azimuthal particle-in-cell benchmark for E × B discharges," Plasma Sources Science and Technology **30**, 075002 (2021).

[7]J.-P. Boeuf and A. Smolyakov, "Landmark, low temperature magnetized plasma benchmarks," `https://jpb911.wixsite.com/landmark/contact` (2017).

[8]M. M. Turner, "Verification of particle-in-cell simulations with Monte Carlo collisions," Plasma Sources Science and Technology **25**, 054007 (2016).

[9]M. Surendra, "Radiofrequency discharge benchmark model comparison," Plasma Sources Science and Technology **4**, 56 (1995).

[10]M. M. Turner, A. Derzsi, Z. Donkó, D. Eremin, S. J. Kelly, T. Lafleur, and T. Mussenbrock, "Simulation benchmarks for low-pressure plasmas: Capacitive discharges," Physics of Plasmas

20, 013507 (2013).

[11]J. Carlsson, A. Khrabrov, I. Kaganovich, T. Sommerer, and D. Keating, "Validation and benchmarking of two particle-in-cell codes for a glow discharge," Plasma Sources Science and Technology **26**, 014003 (2016).

[12]V. Vahedi and M. Surendra, "A Monte Carlo collision model for the particle-in-cell method: applications to argon and oxygen discharges," Computer Physics Communications **87**, 179–198 (1995).

[13]S. Boccelli, P. Parodi, T. E. Magin, and J. G. McDonald, "Modeling high-Mach-number rarefied crossflows past a flat plate using the maximum-entropy moment method," Physics of Fluids **35**, 086102 (2023).

[14]P. Parodi, D. Le Quang, G. Lapenta, and T. E. Magin, "Particle-in-Cell simulation of the VKI DRAG-ON facility," in AIAA SCITECH 2024 Forum (2024).

[15]P. Parodi, G. Lapenta, and T. E. Magin, "Particle-based Simulation of an Air-Breathing Electric Propulsion System," in Proceedings of the 15th Symposium of VKI PhD Research (2024).

[16]S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, and J. Zhang, "PETSc Web page," `https://petsc.org/` (2022).

[17]J. P. Boris, "Relativistic plasma simulation-optimization of a hybrid code," in Proc. Fourth Conf. Num. Sim. Plasmas (1970) pp. 3–67.

[18]M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator," ACM Transactions on Modeling and Computer Simulation **8**, 3–30 (2998).

[19]R. D. Falgout and U. M. Yang, "hypre: A library of high performance preconditioners," in International Conference on Computational Science (Springer, 2002) pp. 632–641.

[20]C. K. Birdsall and A. B. Langdon, Plasma Physics via Computer Simulation (McGraw-Hill, New York, 1985).

[21]H. Qin, S. Zhang, J. Xiao, J. Liu, Y. Sun, and W. M. Tang, "Why is Boris algorithm so good?" Physics of Plasmas **20**, 084503 (2013).

[22]V. Croes, T. Lafleur, Z. Bonaventura, A. Bourdon, and P. Chabert, "2D particle-in-cell simulations of the electron drift instability and associated anomalous electron transport in Hall-effect thrusters," Plasma Sources Science and Technology **26**, 034001 (2017).

[23]T. Charoy, T. Lafleur, A. Tavant, P. Chabert, and A. Bourdon, "A comparison between kinetic theory and particle-in-cell simulations of anomalous electron transport in E × B plasma discharges," Physics of Plasmas **27**, 063510 (2020).

[24]F. Petronio, T. Charoy, A. Alvarez Laguna, A. Bourdon, and P. Chabert, "Two-dimensional effects on electrostatic instabilities in hall thrusters. ii. comparison of particle-in-cell simulation results with linear theory dispersion relations." Physics of Plasmas (2023), 10.1063/5.0119255.

[25]F. Petronio, Plasma instabilities in Hall Thrusters: a theoretical and numerical study, Ph.D. thesis, Ecole Polytechnique (2023).

[26]F. Petronio, T. Charoy, A. Alvarez Laguna, A. Bourdon, and P. Chabert, "Two-dimensional effects on electrostatic instabilities in hall thrusters. i. insights from particle-in-cell simulations and two-point power spectral density reconstruction techniques." Physics of Plasmas (2023), 10.1063/5.0119253.

[27]R. Lucken, V. Croes, T. Lafleur, J.-L. Raimbault, A. Bourdon, and P. Chabert, "Edge-to-center plasma density ratios in two-dimensional plasma discharges," Plasma Sources Science and Technology **27**, 035004 (2018).

[28]E. Tiesinga, P. J. Mohr, D. B. Newell, and B. N. Taylor, "The NIST Reference on Constants, Units, and Uncertainty," https://physics.nist.gov/cuu/Constants/index.html.

[29]L. C. Pitchford, L. L. Alves, K. Bartschat, S. F. Biagi, M.-C. Bordage, I. Bray, C. E. Brion, M. J. Brunger, L. Campbell, A. Chachereau, et al., "Lxcat: An open-access, web-based platform for data needed for modeling low temperature plasmas," Plasma Processes and Polymers **14**, 1600098 (2017).

[30]D. Schultz, R. Wang, P. Stancil, and N. Cariatore, "Data for proton transport through atomic hydrogen environments: Elastic scattering," Atomic Data and Nuclear Data Tables **154**, 101601 (2023).

[31]R. K. Janev, W. D. Langer, K. J. Evans, and D. E. J. Post, Elementary Processes in Hydrogen-Helium Plasmas (Springer-Verlag, 1987).

[32]V. Vahedi, M. A. Lieberman, G. DiPeso, T. D. Rognlien, and D. Hewett, "Analytic model of power deposition in inductively coupled plasma sources," Journal of Applied Physics **78**, 1446–1458 (1995).

[33]P. Parodi and F. Petronio, "Simulation data for the step-by-step verification of PIC-MCC codes from VKI and LPP," Zenodo, https://doi.org/10.5281/zenodo.13839148 (2024), version 1.

[34]P. J. Roache, "Perspective: A Method for Uniform Reporting of Grid Refinement Studies," Journal of Fluids Engineering **116**, 405–413 (1994).

[35]F. Stern, R. V. Wilson, H. W. Coleman, and E. G. Paterson, "Comprehensive Approach to Verification and Validation of CFD Simulations—Part 1: Methodology and Procedures ," Journal of Fluids Engineering **123**, 793–802 (2001).

[36]R. Wilson, J. Shao, and F. Stern, "Discussion: Criticisms of the "Correction Factor" Verification Method 1," Journal of Fluids Engineering **126**, 704–706 (2004).

[37]T. Xing and F. Stern, "Factors of Safety for Richardson Extrapolation," Journal of Fluids Engineering **132**, 061403 (2010).

[38]L. Eça and M. Hoekstra, "A procedure for the estimation of the numerical uncertainty of cfd calculations based on grid refinement studies," Journal of Computational Physics **262**, 104–130 (2014).

[39]F. Riva, C. F. Beadle, and P. Ricci, "A methodology for the rigorous verification of Particle-in-Cell simulations," Physics of Plasmas **24**, 055703 (2017).

[40]G. A. Radtke, N. Martin, C. H. Moore, A. Huang, and K. L. Cartwright, "Robust verification of stochastic simulation codes," Journal of Computational Physics **451**, 110855 (2022).

[41]R. W. Hockney and J. W. Eastwood, Computer Simulation Using Particles (1st ed.) (CRC Press, 1988).

[42]J. P. Verboncoeur, "Particle simulation of plasmas: review and advances," Plasma Physics and Controlled Fusion **47**, A231 (2005).

[43]D. Tskhakaya, K. Matyash, R. Schneider, and F. Taccogna, "The particle-in-cell method," Contributions to Plasma Physics **47**, 563–594 (2007).

[44]C. Birdsall, "Particle-in-cell charged-particle simulations, plus Monte Carlo collisions with neutral atoms, PIC-MCC," IEEE Transactions on Plasma Science **19**, 65–85 (1991).

[45]K. Nanbu, "Probability Theory of Electron–Molecule, Ion–Molecule, Molecule–Molecule, and Coulomb Collisions for Particle Modeling of Materials rocessing Plasmas and Gases," IEEE Transactions on Plasma Science **28**, 971–990 (2000).

[46]E. Roohi and S. Stefanov, "Collision partner selection schemes in DSMC: From micro/nano flows to hypersonic flows," Physics Reports **656**, 1–38 (2016).

[47]J. F. C. Kingman, "Poisson processes revisited," Probability and Mathematical Statistics **26**, 77–95 (2006).