2D and 3D hybrid PIC-fluid modeling of electric thruster plumes

IEPC-2017-209

Presented at the 35th International Electric Propulsion Conference
Georgia Institute of Technology – Atlanta, Georgia – USA
October 8–12, 2017

Adrián Domínguez*, Filippo Cichocki†, Mario Merino‡, Pablo Fajardo§ and Eduardo Ahedo¶
Space Propulsion and Plasma group (EP2), Universidad Carlos III de Madrid, Leganés, Spain

The new 2D and 3D hybrid particle-in-cell (PIC) fluid codes under development by the EP2 group are presented and compared in this manuscript through the simulation of an unmagnetized gridded ion thruster (GIT) plasma plume as benchmark case. The main innovations focused on improving PIC statistics are presented including a new population control algorithm acting through the charge-exchange (CEX) and ionization collisions along the plume expansion. Monte Carlo Collision (MCC) and Direct Simulation Monte Carlo (DSMC) based algorithms have been implemented to treat the effect of CEX collisions, especially important in the near plume region, contributing to the ion backscattering and the ion sputtering on the external thruster walls. Considering a simple quasineutral polytropic closure for the electrons, a good agreement is found between 2D and 3D simulation results. A comparison of the computational cost is also provided.

*PhD. Student, Bio-Engineering and Aerospace Engineering Department, addoming@ing.uc3m.es.
†PhD. Student, Bio-Engineering and Aerospace Engineering Department, fcichock@ing.uc3m.es.
‡Assistant professor, Bio-Engineering and Aerospace Engineering Department, mario.merino@uc3m.es.
§Professor, Bio-Engineering and Aerospace Engineering Department, pfajardo@ing.uc3m.es.
¶Professor, Bio-Engineering and Aerospace Engineering Department, eahedo@ing.uc3m.es.

I. Introduction

The increasing use of electric thrusters onboard modern satellites for a wide range of space missions has made essential to understand and predict any integration issues such as the damage caused by the plasma plume-S/C interaction through ion back-scattering and direct sputtering on onboard sensors or solar panels.[1,2] Thus, the simulation of electric thruster plumes is becoming extremely important for assessing those issues early in the design process. Besides, new applications such as the ion beam shepherd (IBS)[3–6] concept, whose aim is to relocate a debris by gradually pushing it with an electric thruster plume to a different orbit are currently under investigation and development.

Plasma plumes generated by electric thrusters such as GITs or Hall effect thrusters (HET)[7,8] are weakly collisional with very different dynamics of ions and electrons. The latter are the major responsible for the self-consistent electric field and are strongly affected by the presence of an external magnetic field while massflow and momentum are mostly due to ions. As it is shown in the present work, two different regions can be distinguished along the plasma plume expansion:[9,10] a 'near region' extending from the thruster exit up to few thruster radii downstream, where CEX collisions[11,12] between ions and neutrals, plasma quasineutralization and the presence of the applied electric and magnetic fields produce a non-homogeneous plasma with non-negligible 3D effects in the plume, and a 'far region' where, in general, a smoother and more rarefied plasma plume profile expands axisymmetrically. Both near and far regions of unmagnetized plumes are studied in this paper.

In order to numerically simulate such plasma plumes, hybrid PIC-fluid codes[13–24] represent a good compromise between fully kinetic models (including Boltzmann equation solvers in 6-dimensinal phase-space[25,26] and full PIC codes[27,28]) and multi-fluid codes,[9,10,29–32] as they allow to achieve a reasonably good accuracy in the physics description, while maintaining a moderate computational cost. In such codes, the ion and neutral species are treated as macro-particles and their corresponding macroscopic magnitudes are obtained at dedicated mesh nodes through a particle weighting process, while the electrons are modeled as a fluid with the corresponding conservation equations (continuity, momentum and energy). The EP2 research group has been active in this research field for over 10 years, as it has been involved in the development of 2D Hall Effect thruster simulation hybrid codes such as HPHall2[33] and HallMA,[34] both of them based on the original code HPHall.[35]

Recently, the group has started the development of two new hybrid codes. The first is a Cartesian 3D (x,y,z) code named EP2PLUS (Extensible Parallel Plasma PLUme Simulator) devoted to the simulation of electric thruster plasma plumes and their interaction with objects including the S/C and, possibly, a debris object.[23] A first version of this code[36] has already been finalized within the European Union funded project LEOSWEEP.[37] The second is a 2D (z,r) code under development which represents a versatile multi-thruster discharge simulation platform for simulating plasma physics inside and in the near plume of different plasma thrusters such as HETs, high efficient multistage plasma thrusters (HEMPT), helicon plasma thrusters (HPT) and electron cyclotron resonance accelerators (ECRA).[38] This new code replaces the aforementioned HET codes incorporating improved algorithms and extending their capabilities in both the PIC and the electron-fluid sides. Besides, it can be used for 2D simulations of plasma plumes as it is shown in the present work.

Considering an unmagnetized GIT plasma plume as the benchmark case, the aim of this manuscript is to compare both codes focusing on the PIC algorithms, which have seen important improvements with respect to previous codes. Innovations on the treatment of the heavy species are focused on improving the population control and PIC statistics in order to reduce numerical noise. Moreover, new collision algorithms based on either Monte Carlo Collisions (MCC) or Direct Simulation Monte Carlo (DSMC) have been implemented to treat the effect of CEX collisions, especially important in the near plume region, contributing to the ion backscattering and sputtering on the external thruster walls. Regarding the electron fluid model, a simple polytropic closure for the electrons is considered here, as this is a particularly good assumption for HET and GIT plumes, for which several experiments have indicated that the electron thermodynamics can be well approximated with a polytropic law.[39] Besides, a comparison of the computational cost of the two codes is provided.

Section II describes the overall structure and main algorithms of both 2D and 3D codes, including the quasineutral fluid closure. Simulation results are presented in Sec. III. Finally the conclusions are drawn in Sec. IV.

II. 2D and 3D hybrid codes

A. Overall structure and main simulation loop

Both 2D and 3D codes are modular and share the same overall architecture, data types, structure, parallelization logic (Open MP) and input/ouput interfaces, including common baseline modules and dedicated subroutines whenever possible.[23,38] As depicted in Fig. 1, the two central modules of both hybrid simulators are the PIC module and the electron module. The former, taking as inputs the electric potential ϕ and the electron temperature T_e, propagates the heavy species (neutrals and ions) one PIC time step forward obtaining the plasma density n and particle fluxes $\vec{g}$. The latter, knowing those values from the PIC module and applying quasineutrality, solves for the electric potential and the electron temperature, thus closing the loop. Concerning the electron module of the 2D code, a new two-dimensional description of the plasma is considered by solving the electron-fluid equations in the magnetic reference system (defined through the local parallel and perpendicular directions to the magnetic field lines, and the azimuthal direction).[38,40] However, as commented in Sec. I, for the analysis of unmagnetized plasma thrusters plumes (and therefore for the results included in this paper), both codes make use of a simplified polytropic electron closure of the electrons:

$$p_e = nT_e; \qquad \frac{T_e}{T_{e0}} = \left(\frac{n}{n_0}\right)^{\gamma-1}, \tag{1}$$

where γ is the constant polytropic coefficient, and n_0, T_{e0} are the plasma density and electron temperature at a reference node where $\phi = 0$, being ϕ the electric potential. Neglecting inertial terms and using Eq. (1), the electron momentum equation yields:

$$\frac{e\phi}{T_{e0}} = \begin{cases} \ln\left(\frac{n}{n_0}\right) & \text{if} \quad \gamma = 1 \\ \frac{\gamma}{\gamma-1}\left[\left(\frac{n}{n_0}\right)^{\gamma-1} - 1\right] & \text{if} \quad \gamma > 1 \end{cases} \tag{2}$$

Ion and neutral macro-particles are grouped into different populations with a dedicated computational particle list for each of them storing all necessary particle data such as current and previous position and velocity, elementary particle mass and charge status, and particle weight (or number of elementary particles represented by each simulation particle). This subdivision of the different species facilitates the population control during the simulation and the treatment of the various particle collisions between different particle lists, thus contributing to reduce the PIC associated numerical noise. Both codes use a structured PIC mesh. In the general case, the 3D PIC mesh is composed of prismatic cells. On the other hand, the 2D mesh boundaries represent 3D annular surfaces and the quadrilateral cells correspond to 3D annular volumes. In both cases it is possible to identify a uniform computational mesh featuring squared elements (cubic in 3D case). Every point in the physical domain has unique computational coordinates in the computational domain. As shown in Fig. 2, considering the 2D case for the sake of simplicity, a given physical point $\vec{x} = (z, r)$ has computational coordinates $\vec{\chi} = (\xi, \eta)$, 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. The 3D is equivalent with an additional coordinate.

Figure 1 shows the general simulation PIC loop. For each particle population, at the beginning of each simulation time step the particle collider algorithm carries out all macro-particles collisions (refer to Sec. II.B). Next, a 3D particle mover algorithm based on Boris' CYLRAD algorithm[41] updates the particles velocity (v_x, v_y, v_z) and position (x, y, z) taking into account the electro-magnetic fields. In the 2D code the particles are projected to the 2D (z,r) axisymmetric plane before weighting them to the PIC mesh nodes so that they are naturally reflected at the axis $r = 0$, thus avoiding the singularity problem of a 2D cylindrical particle mover.[42]

Once the macro-particles of each population have been moved, a crossing check algorithm determines if the particles have exited the domain. If so, they are stored in a dedicated computational list to be later removed. On the other hand, all those particles contained in the simulation domain are sorted to the corresponding PIC mesh cells which are identified as the integer part of the particles computational coordinates. New particles are injected into the domain through corresponding boundary faces. The values of the injection particle flux g_{inj}, fluid velocity $\vec{u}_{inj}$ and temperature T_{inj} are given at the nodes of each injection face. For each species to be injected, taking into account the generation macro-particle weight at

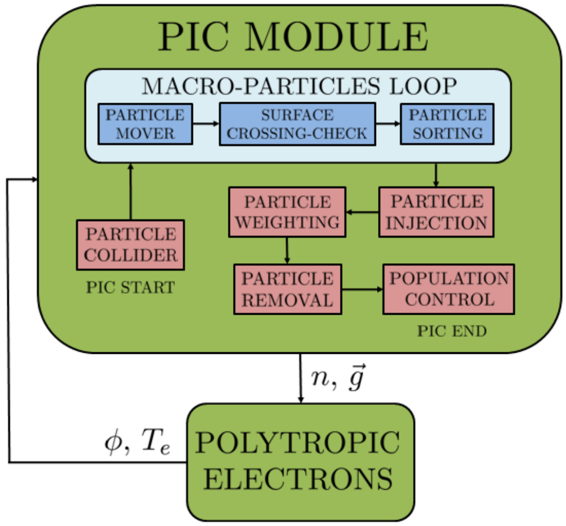

Figure 1. Main simulation loop including PIC and electron fluid modules.

the injection cell W_{gen} (refer to Sec. II.C), the number of macro-particles to be injected in a given time step is rounded appropriately from the expression

$$N_{inj} = \frac{\overline{g}_{inj}\Delta S \Delta t}{W_{gen}}, \tag{3}$$

where ΔS is the injection face area, Δt is the PIC time step and $\overline{g}_{inj}$ is the mean injection flux at the face center. The injected macro-particles are uniformly distributed along the injection face and their macro-particle weight is then obtained from the actual injection particle flux interpolated to the particles random position $g_{inj,\,p}$ as

$$W_{inj} = \frac{g_{inj,\,p}}{\overline{g}_{inj}} W_{gen}. \tag{4}$$

Since the injected macro-particles are uniformly distributed along the injection face, an additional correction on the macro-particle weight is required in the axisymmetric code to cancel out cylindrical effects. The generation weight in the cell is then considered as the mean injection weight at the center of the injection face, so that the actual injected particle weight is obtained by multiplying W_{inj} in Eq. (4) by the ratio $r_p/\overline{r}_{inj}$ where r_p is the random radial coordinate of the injected particle and $\overline{r}_{inj}$ is the radius of the injection face center. On the other hand, the particles injection velocity is obtained from the probability distribution function[43]

$$f_{inj}(\vec{v}) \propto v_{\perp} \exp\left(-\frac{m|\vec{v}-\vec{u}_{inj}|^2}{2T_{inj}}\right), \tag{5}$$

being $v_{\perp}$ the macro-particle velocity perpendicular to the injection face. The values $\vec{u}_{inj}$ and T_{inj} are interpolated from the injection face nodes to the particle position. A continuous injection is simulated by advancing each macro-particle along its injection velocity direction through a random fraction of the PIC time step. Macroscopic magnitudes of a given population are obtained at the PIC mesh nodes through a weighting algorithm using a first order cloud-in-cell (CIC) shape.[44] As an example, the particle density of a given population is computed at a given PIC mesh node as

$$n = \frac{1}{\Delta V}\sum_p W_p S(\vec{\chi}_p) \tag{6}$$

where the sum is extended to all particles contained in those cells sharing the considered node, ΔV is the weighting volume associated to the node, W_p is the macro-particle weight and S is the first order weighting function defined in the computational domain. In the 2D case, for a particular node with computational

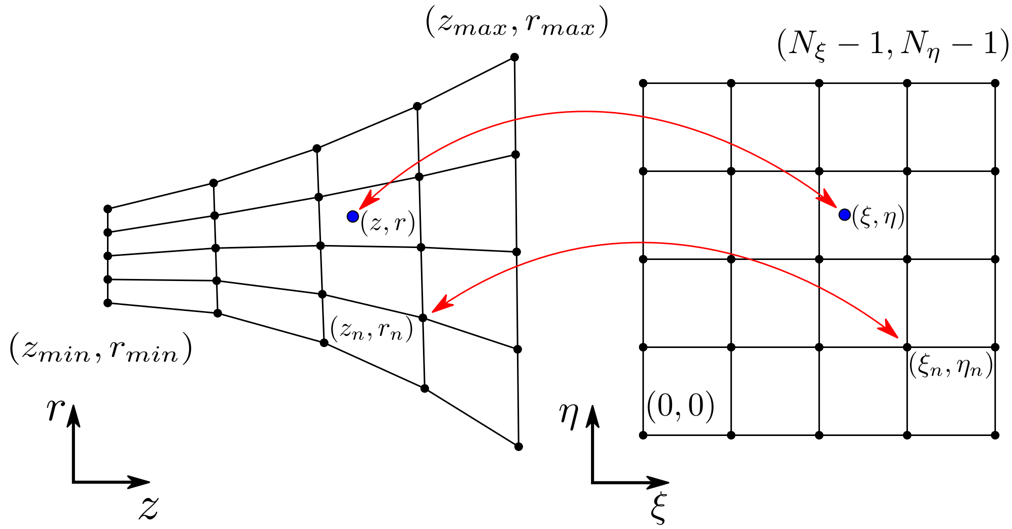

Figure 2. Physical and computational PIC meshes in 2D case.

coordinates $\vec{\chi}_n = (\xi_n, \eta_n)$ this function is evaluated considering the particle computational position $\vec{\chi}_p = (\xi_p, \eta_p)$ as

$$S(\vec{\chi}_p) = (1 - |\xi_p - \xi_n|)(1 - |\eta_p - \eta_n|). \tag{7}$$

Cylindrical effects are taken into account in the 2D case by considering the corrected nodal weighting volumes.[45] For a particular node, its associated volume is

$$\Delta V = \iint_{\Omega(\xi,\eta)} 2\pi r(\xi,\eta) S(\xi,\eta) \left| \frac{\partial(z,r)}{\partial(\xi,\eta)} \right| (\xi,\eta) d\xi d\eta, \tag{8}$$

where the integral is performed in the computational domain. The integration domain comprises the node influence area according to the shape function S. The radial coordinate $r(\xi,\eta)$ and the Jacobian $|\partial(z,r)/\partial(\xi,\eta)|\,(\xi,\eta)$ are conveniently interpolated from the nodes in terms of the function S. For what concerns the 3D mesh considered here, it is Cartesian with fixed spacing along (x, y, z) coordinates. In this simplified case, the weighting volume for inner nodes coincides with the physical cell volume.

Finally, the population control algorithm (see Sec. II.C) is called after removing exited particles from the simulation domain.

B. Macro-particles collisions

Among the large amount of possible collisional processes taking place along the plasma plume, the most relevant ones and thus considered by both codes are ionization collisions up to second degree (generating doubly charged ions) and resonant symmetric CEX without momentum exchange.[9,11,12,46] A cell-wise particle collider algorithm is considered for all collisions.

1. Ionization collisions

Based on the particular method developed in HPHall,[35] the approach implemented is described for the case of the ionization process $A + e \rightarrow A^+ + 2e$. On a given time step Δt, the total mass of new singly-charged ions to be generated in a particular cell of volume ΔV is

$$\Delta m_i = m n n_n R_{01}(T_e) \Delta V \Delta t, \tag{9}$$

where m is the propellant atom mass, n and n_n are the plasma and neutral densities, respectively and $R_{01}(T_e)$ is the ionization rate for the reaction $0 \rightarrow 1$ evaluated at the cell center, which depends on the electron temperature T_e according to the Drawin model[47] (Drawin and Bell[48] models are used respectively for the other ionization reactions considered $A + e \rightarrow A^{++} + 3e$ and $A^{+} + e \rightarrow A^{++} + 2e$). The number of ion macro-particles to be generated in cell is rounded appropriately from the expression $N_{ion} = \Delta m_i/(mW_{gen})$ being W_{gen} the generation macro-particle weight corresponding to the current cell and output ion population (see Sec. II.C). The ion macro-particles velocity is sampled from a Maxwellian distribution with the neutral mean velocity and temperature in the corresponding cell while particles positions are uniformly distributed within the cell. Finally, all neutrals in the cell suffer a macro-particle weight reduction proportional to their current weight, so that the total neutral mass loss in the cell equals Δm_i.

2. CEX collisions

Through resonant symmetric charge-exchange collisions a neutral transfers one or more electrons to an ion. The CEX collisions considered are:

$$A^{+}(\text{fast}) + A(\text{slow}) \rightarrow A^{+}(\text{slow}) + A(\text{fast}) \quad (10)$$

$$A^{++}(\text{fast}) + A(\text{slow}) \rightarrow A^{++}(\text{slow}) + A(\text{fast}) \quad (11)$$

As shown in Fig. 3, the CEX collisions algorithm samples two input heavy populations (e.g. fast ions and slow neutrals) and generates the resulting macro-particles in two different output populations (slow ions and fast neutrals).

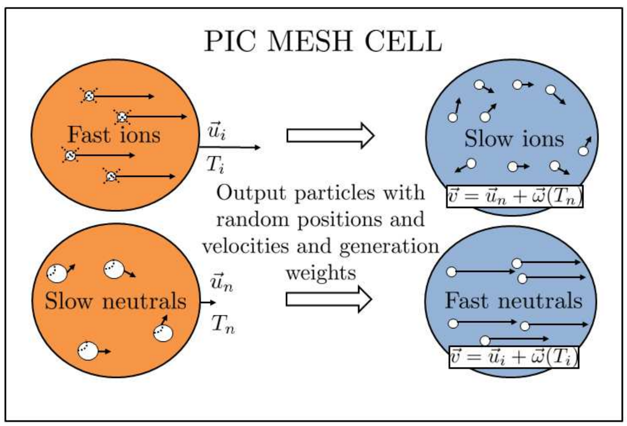

Figure 3. CEX collision in a PIC mesh cell showing the sampling of input species (left) and the output particles generation (right). The input fast ions undergoing the collision are removed from the simulation while the slow neutrals weights are reduced correspondingly. The output slow ions velocity is sampled from corresponding Maxwellian distribution of the input slow neutrals with a mean velocity $\vec{u}_n$ and a temperature T_n, being the latter responsible for the slow ions thermal velocity $\vec{\omega}(T_n)$. The same applies to the output fast neutrals considering the corresponding input fast ions population.

Input macro-particles pairs (each of them constituted by one fast ion and one slow neutral) are sampled following an efficient Direct Simulation Monte Carlo (DSMC) method[43] so that a given pair collides if $U < p_c/p_c^{max}$ being U a random uniform number between 0 and 1 and p_c the collision probability given by

$$p_c = \frac{1 - \exp\left(-\sigma(v_r) n_n v_r \Delta t\right)}{N_n}, \quad (12)$$

where $\sigma(v_r)$ is the collision cross section depending on the relative velocity between the two macro-particles of the pair v_r, and n_n and N_n are the input neutral population particle density and number of macro-particles in cell, respectively. The number of possible input pairs undergoing a collision in a given cell and time step is limited to $p_c^{max} N_i N_n$ with $p_c^{max} = p_c(|v_r \sigma(v_r)|_{max})$ and N_i the number of input ion macro-particles in cell. The CEX cross section $\sigma(v_r)$ is taken from Miller's model.[11] The new generated particles in output populations (slow ions and fast neutrals) are uniformly distributed in the cell and have the corresponding generation macro-particle weight (refer to Sec. II.C). Their velocities are sampled from a local Maxwellian distribution function with mean velocity and temperature of the corresponding input population. After generating the output particles, the weight of the heavier particle of each input pair is reduced by an amount equal to that of the lighter one, while the lighter macro-particle is removed from the simulation thus ensuring mass conservation.

It is worth to note here that after generating the output particles in both ionization and CEX collisions, a macro-particle weight correction similar to that performed by the injection algorithm (see Sec. II.A) is carried out in the 2D code to cancel out cylindrical effects.

C. Population control

In order to improve the PIC-derived statistics and reduce its associated noise, a minimum number of macro-particles per cell of each simulated population is needed. On the other hand, this number must be below a given maximum so as to limit the computational time and memory required by the simulation. The population control algorithm implemented in both codes maintains the number of macro-particles per cell of each simulated particle population within a specied range $[N_{min}, N_{max}]$ in stationary conditions while minimizing the macro-particle weights dispersion. For each cell and particle population, whenever the number of macro-particles per cell is out of the above range, it modifies the generation weight W_{gen} used by the injection and collisional algorithms (see Secs. II.A, and II.B.1 and II.B.2, respectively) as

$$W_{gen} = \overline{W} \frac{N_c}{N_{tg}} \tag{13}$$

where $N_{tg} \in [N_{min}, N_{max}]$ is the desired number of macro-particles per cell and N_c and $\overline{W}$ are the current number of macro-particles and average macro-particle weight in cell, respectively. This algorithm makes N_c tend asymptotically to the desired value N_{tg} in steady conditions. In order to avoid the generation of too many particles with negligible macro-particle weight, a minimum generation weight is also set for each population, which may lead the number of macroparticles per cell to drop below N_{min}.

III. Simulations

A. Simulations settings

The plasma plume of the NASA's NSTAR gridded ion thruster[49,50] is simulated by both codes. A 3D (x, y, z) 2m side cubic domain is considered for the simulation. The equivalent 2D (z, r) domain is a cylinder with 1m radius and 2m long. The grid spacing and number of nodes along each direction and the main simulation parameters are listed in Table 1. The thruster injection area is circular with 14 cm radius. In the 3D code, only injection faces with a center distance from the plume axis (axis z) lower than the injection surface radius are considered. The different particle populations considered in the simulations are listed in Table 2 including the corresponding subscripts used for identifying their associated magnitudes. Different particle computational lists are considered for the injected populations and those species which are generated through collisional processes (ionization and CEX), thus facilitating the population control. Three different populations are injected. A flat injection density profile is considered for the neutrals, which are injected with sonic conditions. Singly and doubly charged ions are injected following a Parks-Katz self-similar model (SSM)[9,29] with polytropic coefficient $\gamma = 1.2$. The radius of the outermost streamline (enclosing a plasma tube carrying 95% of the plasma current) is $R_0 = 14$ cm while the corresponding divergence angles (α_0) are 20.5 and 30 deg for singly and doubly charged ions, respectively. A ratio between doubly and singly charged injected ion currents of 9.1% is considered.[49,50] The electron thermodynamics follow the polytropic closure described in Eqs. (1) and (2) with a polytropic coefficient $\gamma = 1.2$ and $T_{e0} = 3.5$ eV at the potential reference node, located at 6 cm downstream the thruster exit plane. In order to reach stationary conditions, the simulation duration is sufficiently long to let a slow CEX ion of 5 eV cross axially the whole simulation domain. The time step is set so that, on average, a fast doubly charged ion crosses a maximum of one PIC mesh cell per simulation step. Regarding the population control algorithm, the target number of macro-particles per cell for all populations is 500 except for the fast neutrals from CEX collisions, whose value is set to 100 to limit the computational time. A control range of $\pm 10\%$ of the targeted value is set for each population.

Simulation parameter	Units	Value
3D (x, y, z) mesh number of nodes	N/A	$201 \times 201 \times 101$
2D (r, z) mesh number of nodes	N/A	101×101
Grid spacing ($\Delta x = \Delta y \equiv \Delta r$, Δz)	cm	1, 2
Simulation time step	s	3×10^{-7}
Number of simulation steps	N/A	10000
Total simulation time	ms	3
Injected Xe profile	N/A	Flat
Injected Xe velocity	m/s	247 (sonic)
Injected Xe temperature	eV	0.05
Injected Xe^{+} profile	N/A	SSM with $R_0 = 14$ cm, $\alpha_0 = 20.5$ deg and $\gamma = 1.2$
Injected Xe^{+} kinetic energy	eV	1040
Injected Xe^{+} temperature	eV	0.1
Injected Xe^{++} profile	N/A	SSM with $R_0 = 14$ cm, $\alpha_0 = 30$ deg and $\gamma = 1.2$
Injected Xe^{++} kinetic energy	eV	2080
Injected Xe^{++} temperature	eV	0.2
Electron temperature at the reference node	eV	3.5
Electron polytropic coefficient	N/A	1.2

Table 1. Main simulation parameters.

Pop. subscript	Type	Origin
$i1$	Singly charged ions	Slow ions generated through ionization and CEX
$i2$	Singly charged ions	Fast injected ions
$i3$	Doubly charged ions	Slow ions generated through ionization and CEX
$i4$	Doubly charged ions	Fast injected ions
$n1$	Neutrals	Slow injected neutrals
$n2$	Neutrals	Fast neutrals generated through CEX

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

B. Simulation results

The SSM solution for singly charged ions with the parameters of Table 1 is first compared to the 2D and 3D codes simulation results. In order to reduce the PIC related noise, all macroscopic magnitudes are time-averaged over 100 simulation time steps. Figure 4(a) shows the ion particle density isocontours. Regarding the simulation results, the contributions of the singly charged ion populations 1 and 2 are added. Along the plume centerline, a good agreement between both simulations and the SSM solution is found. In the near plume region, a low energy CEX ion cloud with non-negligible density is generated. This ion population is not considered by the non-collisional self-similar model. On the other hand, electric potential isocontours are depicted in Fig. 4(b). The 2D and 3D codes results feature a larger potential fall along the plume centerline. The presence of slow ion populations (singly and doubly charged) due to the collisional processes taking place mainly in the near plume region increases the total plasma density of the reference node. The injected doubly charged ions have a larger divergence angle so that, in the far plume, the plasma density is mainly due to singly charged ions. Therefore, for the same plasma density in the far plume region, the lower density ratio n/n_0 yields a lower electric potential.

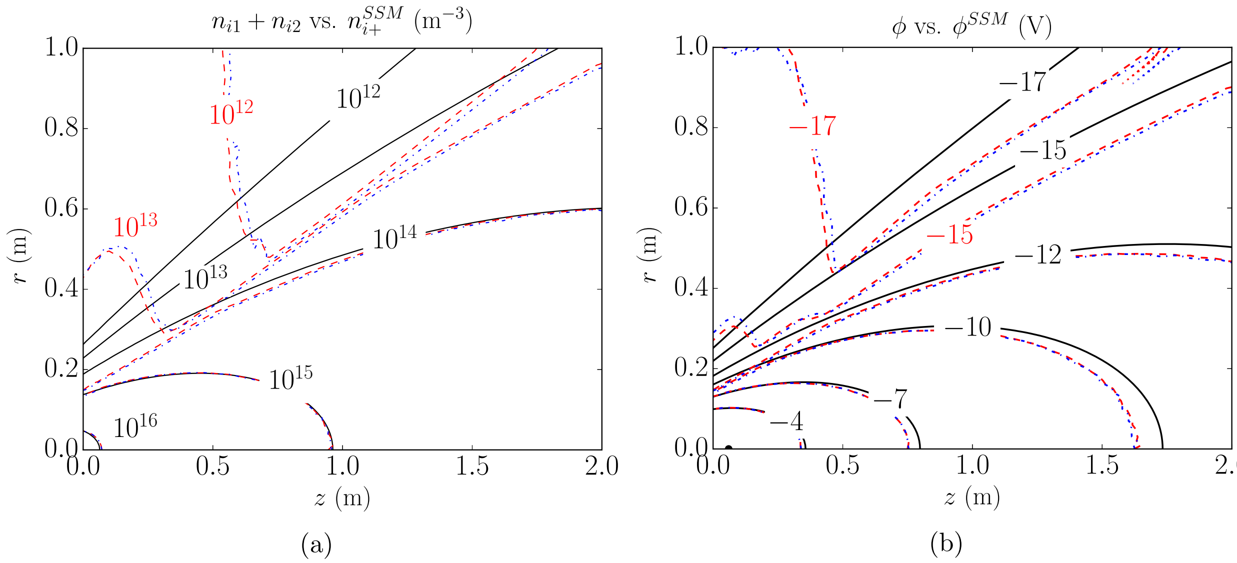

Figure 4. Comparison between 2D (red dashed lines), 3D (blue dot dashed lines) simulation results and equivalent SSM solution for singly charged ions (black solid lines). (a) Particle density. Contributions of ion populations 1 and 2 are considered here. (b) Electric potential. The black dot indicates the location of the potential reference node (at the plume centerline, 6cm downstream from the thruster exit plane).

As shown on Fig. 5(a), both 2D and 3D plasma density isocontour lines match quite well. It is worth noting that this happens regardless of the differences in the weighting algorithms in cylindrical 2D and 3D codes. Regarding the total neutral density depicted in figure 5(b), more noisy results are obtained in the far plume region due to the lower number of particles per cell, as it is usual in PIC simulations. Figures 5(c)-(d) and 5(e)-(f) show, respectively, the particle density and the number of particles per cell for slow singly and doubly charged ions. In order to limit the number of particles generated and thus reduce the computational time, a larger minimum generation weight has been considered in the 3D simulation, thus yielding a lower number of particles per cell. However, apart from the higher noise in the far plume region, the particle density isocontours from the 3D simulation shown in Figs. 5(c) and 5(e) are in good agreement with those of the 2D simulation considering a larger number or particles per cell.

Regarding the computational cost, the 2D code completes a total simulation time of 3ms in several hours while the corresponding computational time is around a day for the 3D code. Table 3 shows the number of macro-particles per simulated population in stationary conditions.

Population	**2D code**	**3D code**
$i1$	2.0	43.7
$i2$	0.7	29.7
$i3$	1.4	7.9
$i4$	0.7	28.4
$n1$	0.3	14.4
$n2$	0.4	14.4
Total	5.5	138.5

Table 3. Number of macro-particles (in millions) of each population in stationary conditions in 2D and 3D cases.

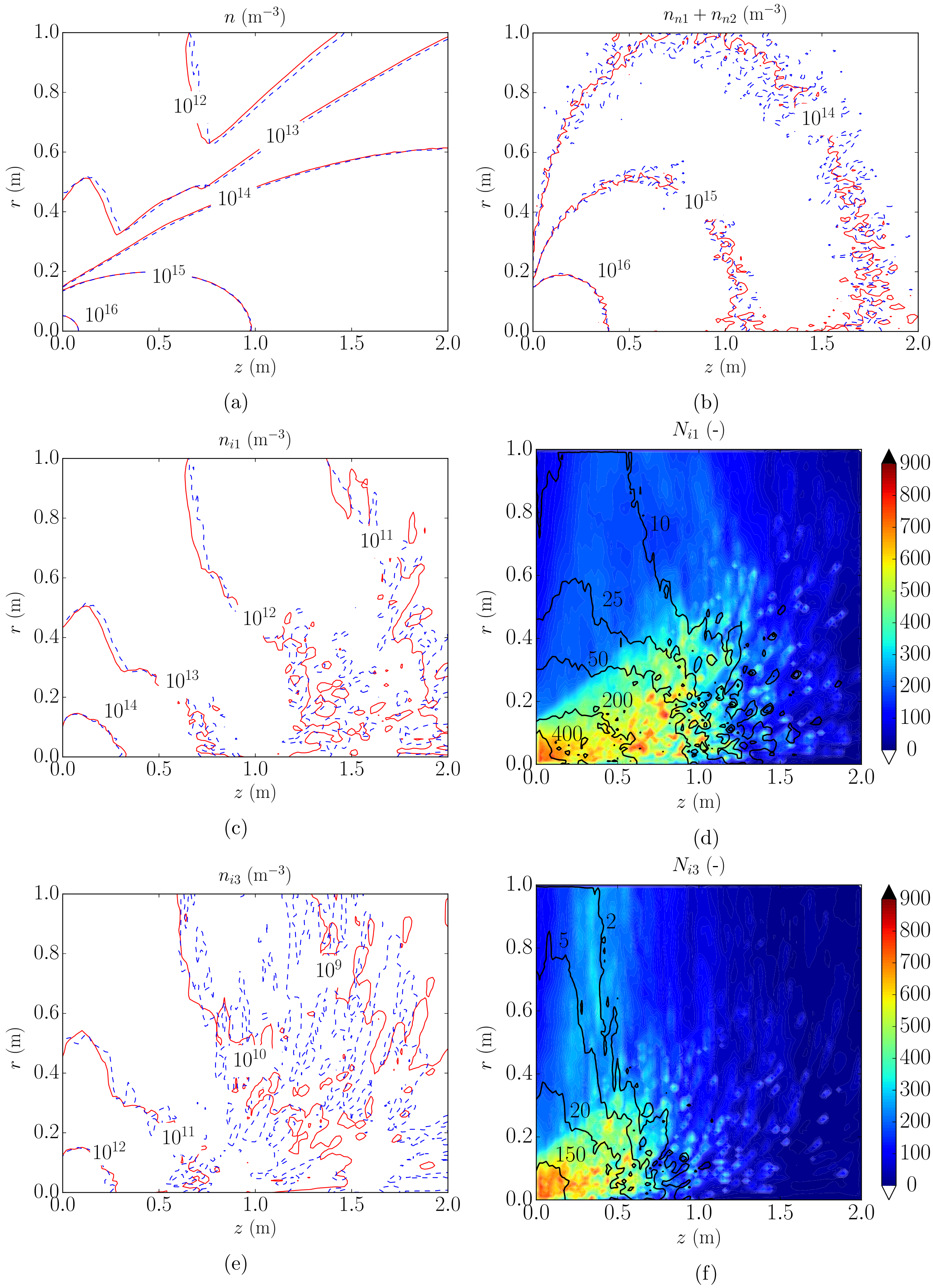

Figure 5. Comparison of 2D and 3D simulation results. (a) Plasma particle density. (b) Total neutral particle density. (c) and (e) Slow singly and doubly charged ions particle density, respectively. (d) and (f) Slow (generated by CEX collisions) singly and doubly charged ions number of macro-particles per cell, respectively. Red solid and blue dashed isocontour lines in figures (a), (b), (c) and (e) corresponds to 2D and 3D simulation results, respectively. In figures (d) and (f) the coloured contour refers to the 2D simulation while the black solid isocontour lines correspond to the 3D simulation.

IV. Conclusions

The main characteristics of the new 2D and 3D hybrid PIC-fluid codes under development by the EP2 group have been presented. The main innovations on the PIC segment focused on improving the PIC statistics have been described including new population control and collisional algorithms to treat the effect of CEX collisions, especially important in the near plume region, since they contribute to the ion backflow and induced sputtering on the external thruster walls. Considering a simple quasineutral polytropic closure for the electrons, the simulation of NASA's NSTAR GIT plasma plume has been considered as a benchmark case to show that both 2D and 3D hybrid PIC-fluid codes results are in good agreement. In addition, an expected deviation from the SSM solution has been found and it is mainly attributed to the non-negligible slow ion density generated by charge-exchange collisions in the near plume region. The implemented population control allows to reduce the results noise while limiting the computational cost of the simulation.

Acknowledgments

This work has been partially supported by the CHEOPS project, that received funding from the European Unions Horizon 2020 research and innovation programme, under grant agreement No 730135. Additional support came from Project ESP2016-75887, funded by the National research and development programme of Spain.

References

[1]H.B. Garrett. The charging of spacecraft surfaces. *Reviews of Geophysics*, 19(4):577–616, 1981.

[2]I.D. Boyd and A. Ketsdever. Interactions between spacecraft and thruster plumes. *Journal of Spacecraft and Rockets*, 38(3):380–380, 2001.

[3]C. Bombardelli and J. Peláez. Ion beam shepherd for contactless space debris removal. *Journal of Guidance, Control and Dynamics*, 34(3):916–920, May 2011.

[4]C. Bombardelli, H. Urrutxua, M. Merino, E. Ahedo, and J. Peláez. Relative dynamics and control of an ion beam shepherd satellite. In James V. McAdams, David P. McKinley, Matthew M. Berry, and Keith L. Jenkins, editors, *Spaceflight mechanics 2012*, volume 143 of *Advances in the Astronautical Sciences*, pages 2145–2158. Univelt, 2012.

[5]C. Bombardelli, H. Urrutxua, M. Merino, E. Ahedo, and J. Peláez. The ion beam shepherd: A new concept for asteroid deflection. *AA*, 90(1):98 – 102, 2013.

[6]M. Merino, E. Ahedo, C. Bombardelli, H. Urrutxua, and J. Peláez. Ion beam shepherd satellite for space debris removal. In Luigi T.DeLuca, Christophe B., Oskar J.H., and Sergey M.F., editors, *Progress in Propulsion Physics*, volume IV of *EUCASS Advances in Aerospace Sciences*, chapter 8, pages 789–802. Torus Press, 2013.

[7]E. Ahedo. Plasmas for space propulsion. *Plasma Phys. Control. Fusion*, 53(12):124037, 2011.

[8]D. M. Goebel and I. Katz. *Fundamentals of electric propulsion: Ion and Hall thrusters*. Willey, New York, 2008.

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

[10]F. Cichocki, M. Merino, and E. Ahedo. Modeling and simulation of EP plasma plume expansion into vacuum. In *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Cleveland, OH*, AIAA 2014-3828, 2014.

[11]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.

[12]D. Rapp and W. E. Francis. Charge exchange between gaseous ions and atoms. *Journal of Chemical Physics*, 37(11):2631–2645, 1962.

[13]J. Forest, L. Eliasson, and A. Hilgers. A new spacecraft plasma simulation software, picup3d/spis. In *Spacecraft Charging Technology*, volume 476, page 515, 2001.

[14]M. Masselin. Development of a hybrid pic code for the simulation of plasma spacecraft interactions, 2012.

[15]M. Wartelski, C. Theroude, C. Ardura, and E. Gengembre. Self-consistent simulations of interactions between spacecraft and plumes of electric thrusters. Technical report, IEPC-2013, 2013.

[16]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, Toulouse, France*, IEPC-03-134, 2003.

[17]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.

[18]B. Korkut and D.A. Levin. Three dimensional coupled PIC and DSMC simulations of ion thruster plumes with SUGAR. In 50^{th} *AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, page 3447, 2014.

[19]D.L. Kahnfeld. Hybrid plume modeling. Master's thesis, Mathematisch-Naturwissenschaftliche Fakultt Ernst-Moritz-Arndt-Universitt Greifswald, Greifswald, Germany, 2015.

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

[21]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 AIAA/SAE/ASEE Joint Propulsion Conference*, page 4939, 2016.

[22]I. Boyd and J. Yim. Hall thruster plume simulation using a detailed hybrid model. In 40^{th} *AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 3952, 2004.

[23]F. Cichocki, A. Domínguez, M. Merino, and E. Ahedo. A 3D hybrid code to study electric thruster plumes. In *Space Propulsion Conference*, number 3124968, Rome, Italy, 2016.

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

[25]M. Martínez-Sánchez, J. Navarro-Cavallé, and E. Ahedo. Electron cooling and finite potential drop in a magnetized plasma expansion. *Physics of Plasmas*, 22:053501, 2015.

[26]S. Correyero, J. Navarro, and E. Ahedo. Collisionless electron cooling on magnetized plasma expansions: advances on modelling. In 34^{th} *International Electric Propulsion Conference*, number IEPC-2015-117, Fairview Park, OH, 2015. Electric Rocket Propulsion Society.

[27]J. Wang, D. Han, and Y. Hu. Kinetic simulations of plasma plume potential in a vacuum chamber. *IEEE Transactions on Plasma Science*, 43(9):3047–3053, 2015.

[28]Yuan Hu and Joseph Wang. Fully kinetic simulations of collisionless, mesothermal plasma emission: Macroscopic plume structure and microscopic electron characteristics. *Physics of Plasmas*, 24(3):033510, 2017.

[29]D.E. Parks and I. Katz. A preliminary model of ion beam neutralization. In 14^{th} *International Electric Propulsion Conference*, Fairview Park, OH, 1979. Electric Rocket Propulsion Society.

[30]J. Ashkenazy and A. Fruchtman. Plasma plume far field analysis. In 27^{th} *International Electric Propulsion Conference*, Fairview Park, OH, 2001. Electric Rocket Propulsion Society.

[31]A.G. Korsun and E.M. Tverdokhlebova. The characteristics of the EP exhaust plume in space. In 33^{rd} *AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Washington DC, 1997. AIAA.

[32]A.L. Ortega, I. Katz, I.G. Mikellides, and D.M. Goebel. Self-consistent model of a high-power hall thruster plume. *IEEE Transactions on Plasma Science*, 43(9):2875–2886, 2015.

[33]F. I. Parra, E. Ahedo, M. Fife, and M. Martínez-Sánchez. A two-dimensional hybrid model of the Hall thruster discharge. *Journal of Applied Physics*, 100(2):023 304–1–023 304–11, 2006.

[34]D. Escobar and E. Ahedo. Two-dimensional electron model for a hybrid code of a two-stage Hall thruster. *IEEE Transactions on Plasma Science*, 36(5):2043–2057, 2008.

[35]J. M. Fife. *Hybrid-PIC modeling and electrostatic probe survey of Hall thrusters*. PhD thesis, Aeronautics and Astronautics Dept., Massachuchusetts Institute of Technology, Boston, September 1998.

[36]F. Cichocki, A. Domínguez, M. Merino, and E. Ahedo. Hybrid 3D model for the interaction of plasma thruster plumes with nearby objects. *Submitted to Plasma Sources Science and Technology*, 2017.

[37]M. Ruiz, I. Urdampilleta, C. Bombardelli, E. Ahedo, M. Merino, and F. Cichocki. The FP7 LEOSWEEP project: Improving low earth orbit security with enhanced electric propulsion. In *Space Propulsion Conference*, number 2980908, Cologne, France, 2014.

[38]A. Domínguez, D. Pérez-Grande, P. Fajardo, and E. Ahedo. NOMADS: development of a versatile plasma discharge simulation plarform for electric propulsion. In *Space Propulsion Conference*, number 3124869, Rome, Italy, 2016.

[39]B.E. Beal, A. Gallimore, and W.A. Hargus. Plasma properties downstream of a low-power Hall thruster. *Physics of plasmas*, 12(123503):1, 8, 2005.

[40]D. Pérez, J. Zhou, A. Domínguez, P. Fajrdo, and E. Ahedo. Development of a fully two-dimensional electron fluid model for plasma thrusters. In 35^{th} *International Electric Propulsion Conference*, Georgia Institute of Technology, USA, 2017. Electric Rocket Propulsion Society.

[41]R. W. Hockney and J. W. Eastwood. *Computer Simulations Using Particles.* Adam Hilger, Bristol and New York, 1988.

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

[43]G. A. Bird. *Molecular gas dynamics and the direct simulation of gas flows.* Clarendon Press, Oxford, 1994.

[44]C.K. Birdsall and D. Fuss. Clouds-in-clouds, clouds-in-cells physics for many-body plasma simulation. *Journal of Computational Physics*, 135(2):141–148, 1997.

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

[46]K. Nanbu and Y. Kitatani. An ion-neutral species collision model for particle simulation of glow discharge. *Journal of Physics D: Applied Physics*, 28(2):324, 1995.

[47]M. Mitchner and C.H. Kruger Jr., editors. *Partially ionized gases.* Wiley, 1973.

[48]E.W. Bell, N. Djurić, and G.H. Dunn. Electron-impact ionization of in^+ and xe^+. *Physical Review A*, 48(6):4286, 1993.

[49]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*, page 2274, 1999.

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