45th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit
2 - 5 August 2009, Denver, Colorado

AIAA 2009-4914

Accuracy improvements in a Hall thruster PIC/fluid code

Robert Santos* and Eduardo Ahedo†
Universidad Politécnica de Madrid, Madrid 28040, Spain

Improvements on particle algorithms using non-orthogonal structured meshes with refinement in the near-wall region are presented for a Hall thruster hybrid code. As a result, the total mass and energy of heavy species of the plasma is conserved with more precision along the thruster. By the other way, a new recombination model at walls is presented with the aim of controlling better the neutral particle population and reducing the statistical oscillations.

I. Introduction

Hybrid codes are at the present day a standard research tool to simulate plasma physics, and are specially well suited to study the plasma discharge of a Hall thruster. Our hybrid code is an upgraded version of the original HPHall code developed by Fife and Martínez Sánchez.[1,2] HPHall is a time-dependent, axisymmetric code that consists of a particle-in-cell (PIC) subcode[12] for the heavy species of the plasma, an anisotropic fluid subcode for the electrons, and plasma quasineutrality. The PIC subcode models the heavy species as many test particles, where each test particle represents a huge number of neutrals(n), singly-charged ions (i_+), and doubly-charged ions (i_{++}), respectively. Montecarlo collision methods (MCC) are implemented to model the different collisional processes of the heavy species, and a 'leapfrog' integration scheme[12] is used to move the particles over the self-consistent electromagnetic field. The macroscopic magnitudes of heavy species, such as ion density or ion current density, are obtained from weighting at the nodes of the simulation mesh of near particle properties.

Several improvements on the PIC subcode algorithms have been already reported in previous works.[4] Those improvements were related to the conservation of the mass flow of heavy species and a better control of the ion-particle population to achieve good statistics in simulations. But no references about the influence of the computational PIC mesh were made, and neutrals were excluded from the control population algorithm, despite our experience suggests that these issues have an important role in the results of the simulations, specially in the near-plume region.

In this paper we further improve the particle algorithms to include the effect of the non-orthogonal PIC meshes in the computation of the electric field and in the conservation of the mass flow of the heavy species along the thruster. The ion balance energy is also improved to achieve less than 0.5% of error, and a new recombination model at walls is presented in order to better control the neutral population in simulations.

*PhD Candidate, robert.santos@upm.es, Full address: E.T.S. Ingenieros Aeronáuticos, Plaza Cardenal Cisneros, Madrid 28040, Spain, Student AIAA Member
†Professor, eduardo.ahedo@upm.es, Senior AIAA Member.

II. The mesh and the computational grid

The PIC code uses a mesh for the spatial discretization of the physical domain. The electromagnetic field is computed in the nodes of that mesh. This avoids the calculation of the Coulomb interaction forces between the particles, which means much less computational effort in simulations. Once the field forces are known in the nodes of the spatial-mesh, they are interpolated to the particle positions (gather process) to update particle positions and velocities (particle push process) in the next timestep of the simulation. The macroscopic magnitudes of the plasma species are then obtained by averaging the near particle properties within a local volume of influence of the mesh nodes (weighting process).

Most of the current particle codes use orthogonal grids (Cartesian or cylindrical) as a spatial discretization, because of the simplicity and reduced cost in computational effort. But their application is reduced to problems with simple geometries. Since HPHall simulates not only the channel discharge of a Hall thruster, which varies its annular-cylindrical geometry due to sputtering, but also the near-plume region, a more complex geometry of the spatial-mesh is necessary. This suggests two possibilities: the use of unstructured meshes,[14] and the use of non-orthogonal structured meshes.[13]

The HPHall code uses the last one, which consists of a logically connected deformable quadrilateral cells in two dimensions (notice that HPHall invokes axisymmetry). Figure 1(a) shows a typical mesh used by HPHall to simulate a SPT100-type Hall thruster.

The logically Cartesian connection of the quadrilateral cells preserves the nearest-neighbour relationships, which means a fast and trivial localization of every particle in each cell. This is a clear advantadge with respect to the unstructured meshes, since the indirect adressing methods are avoided, and the additional memory references are not required. As a disadvantadge, the cells in the non-orthogonal unstructured meshes may have a large distortion to fit the boundaries, whereas in a unstructured mesh the cells can have different size due to refinement, but not different shape due to distortion.

The non-orthogonal structured mesh of the physical space $\Omega_P = \{z, r\}$ is then mapped to a Cartesian computational grid in the logical space $\Omega_L = \{\xi, \eta\}$, Fig. 1(b). This computational grid is constituted by unit-quadrilaterals, and there is a unique relation between the logical coordinates $\{\xi_p, \eta_p\}$ and the physical coordinates $\{z_p, r_p\}$ of each particle. This allows the utilization of the weighting algorithms developed for Cartesian-grid codes, and avoids the complications of doing such task directly on non-orthogonal grids. For a more flexibility in the mesh-generation process, the physical domain can be split into a set of non-overlapping patches, each with its own transformation in the logical space.

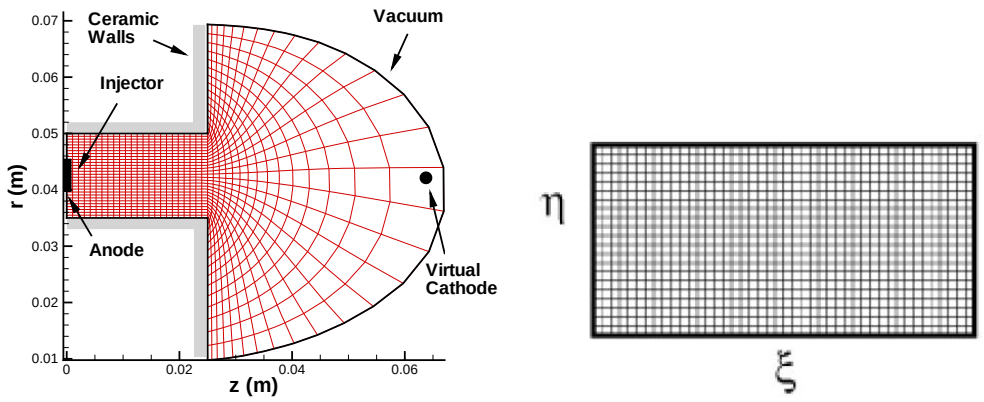
Along this paper, the selected mesh to perform the simulations is constituted by 40 x 30 nodes and 1131 quadrilateral cells, fig. 1(a). The mesh is refined (smaller cells) in the near-wall region of the external walls of the thruster, with the aim of computing with more accuracy the gradient length of plasma inhomogeneities towards the walls, and having enough test particles within these low-density regions, since the PIC code fixes a minimum number of particles-per-cell ($\backsim 20 - 50$).

III. Computing the electric field in a non-orthogonal mesh

For usual operation regimes of Hall thrusters, is generally assumed that the electric field is practically irrotational, that is $\nabla \times \mathbf{E} = 0$, and therefore admits an electric potential function

$$\mathbf{E} = -\nabla\phi. \tag{1}$$

The HPHall code uses a macroscopic fluid subcode to solve this electric potential function at the nodes of the computational grid, $\phi_{j,k}$. The electric field, which moves the particles within



(a) (b)

Figure 1. (a) Sketch of a typical simulation mesh of HPHall. The mesh is consituted by 1131 **quadrilateral cells and is refined in the near-wall region. (b) The cartesian computational grid in the logical space** $\Omega_L = \{\xi, \eta\}$**.**

the physical space in the next timestep, is then computed in two steps. First, the gradient of the electric potential function is computed in the logical space Ω_L using a centered-difference method,

$$\left.\frac{\partial \phi}{\partial \xi}\right|_{j,k} = \frac{\phi_{j+1,k} - \phi_{j-1,k}}{2}$$
$$\left.\frac{\partial \phi}{\partial \eta}\right|_{j,k} = \frac{\phi_{j,k+1} - \phi_{j,k-1}}{2} \tag{2}$$

and second, the resulting vector is transformed into the physical space Ω_P,

$$E_z|_{j,k} = -\left[\frac{\partial \phi}{\partial \xi}\frac{\partial \xi}{\partial z} + \frac{\partial \phi}{\partial \eta}\frac{\partial \eta}{\partial z}\right]_{j,k}$$
$$E_r|_{j,k} = -\left[\frac{\partial \phi}{\partial \xi}\frac{\partial \xi}{\partial r} + \frac{\partial \phi}{\partial \eta}\frac{\partial \eta}{\partial r}\right]_{j,k} \tag{3}$$

where

$$\mathbf{J} = \left[\frac{\partial(\xi, \eta)}{\partial(z, r)}\right] \tag{4}$$

is the jacobian matrix of the mapping $\xi = \xi(z, r), \eta = \eta(z, r)$, also calculated at each node of the computational grid using a centered-difference method in the logical space, but only once at the start of the simulation.

This computation of the electric field does not take into account the non-uniformities of the mesh, because the centered-difference method is applied in the uniform cartesian grid of the logical space. The error associated is related to the characteristic length of the cell, ΔL_{cell}, in the physical space. This error can be large in regions where the distortion of the mesh is important, like the near-wall and near-plume regions, and affects the axial and radial components of the electric field. This may produce incorrect values of the ion current, specially in the near-plume region of the thruster.

A more correct form to compute the electric field takes into account a 5-points finite-difference method, which involves the reference node (j, k) and the nodes connected to it, fig. 2. Notice

that the original scheme used by HPHall is a 4-points finite-defference method. The arc-length parameters χ and σ of the curves $\xi = cont$ and $\eta = const$, respectively, are used now as intermediate variables instead of the logical variables ξ and η. This scheme leads to a weighted average of the electric potential over the connected nodes,

$$\left.\frac{\partial\phi}{\partial\sigma}\right|_{j,k} = w_{j-1,k}\phi_{j-1,k} + w_{j,k}\phi_{j,k} + w_{j+1,k}\phi_{j+1,k}$$
$$\left.\frac{\partial\phi}{\partial\chi}\right|_{j,k} = s_{j,k-1}\phi_{j,k-1} + s_{j,k}\phi_{j,k} + s_{j,k+1}\phi_{j,k+1} \tag{5}$$

where the parameters w and s are defined as

$$w_{j+1,k} = \frac{\Delta\sigma_{j-1}}{\Delta\sigma_j\left(\Delta\sigma_j + \Delta\sigma_{j-1}\right)} \qquad s_{j,k+1} = \frac{\Delta\chi_{k-1}}{\Delta\chi_k\left(\Delta\chi_k + \Delta\chi_{k-1}\right)} \tag{6}$$

$$w_{j-1,k} = \frac{\Delta\sigma_j}{\Delta\sigma_{j-1}\left(\Delta\sigma_j + \Delta\sigma_{j-1}\right)} \qquad s_{j,k-1} = \frac{\Delta\chi_k}{\Delta\chi_{k-1}\left(\Delta\chi_k + \Delta\chi_{k-1}\right)} \tag{7}$$

$$w_{j,k} = w_{j-1,k} - w_{j+1,k} \qquad s_{j,k} = s_{j,k-1} - s_{j,k+1} \tag{8}$$

and is easily observed that $w_{j,k}$ and $s_{j,k}$ become zero when the mesh is uniform.
Finally, the corrected electric field is computed as

$$E_z|_{j,k} = -\left[\frac{\partial\phi}{\partial\sigma}\frac{\partial\sigma}{\partial z} + \frac{\partial\phi}{\partial\chi}\frac{\partial\chi}{\partial z}\right]_{j,k}$$
$$E_r|_{j,k} = -\left[\frac{\partial\phi}{\partial\sigma}\frac{\partial\sigma}{\partial r} + \frac{\partial\phi}{\partial\chi}\frac{\partial\chi}{\partial r}\right]_{j,k} \tag{9}$$

where now the jacobian matrix $\mathbf{J} = \left[\frac{\partial(\sigma,\chi)}{\partial(z,r)}\right]$ of the mapping $\sigma = \sigma(z,r), \chi = \chi(z,r)$, is computed using the 5-points scheme (5).
This correction on the electric field computation reduces the error to ΔL_{cell}^{α}, where $1 < \alpha < 2$. The finite-difference scheme must involve more nodes if more accuracy is needed. But these schemes are pretty much difficult to implement because of the non-uniformities of the mesh, and computational cost is also increased.

IV. Computing with accuracy the mass and energy fluxes in a non-orthogonal mesh

The mass and energy fluxes of heavy species throughout the computational domain, $\mathbf{g}_j$ and $\mathbf{e}_j$ respectively, are obtained by the PIC subcode from the weighting formulas[3]

$$\mathbf{g}_j = \frac{1}{\Delta V}\sum_j M_j\mathbf{v}_j S(z_j, r_j) \tag{10}$$

$$\mathbf{e}_j = \frac{1}{\Delta V}\sum_j \frac{1}{2}M_j v_j^2\mathbf{v}_j S(z_j, r_j) \tag{11}$$

where subindex j stands for neutral and ion species, $S(z_j, r_j)$ is the (bi-linear) weighting function, ΔV is the volume of influence of the weighting, and M_j is the particle mass.

Parra et al.[4] showed that the time-accurate computations of these macroscopic PIC variables must involve time-centered interpolations of the position and mass of the particles. With these

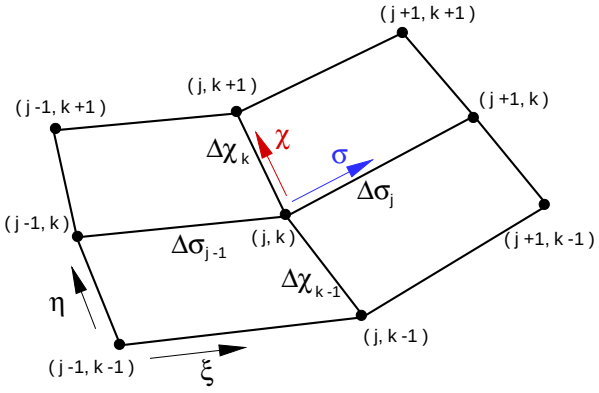


Figure 2. Sketch of the 5-points finite-difference method to compute the electric field in a non-uniform structured mesh.

corrections, the total mass flux is almost totally conserved along the channel discharge for a *standard* PIC mesh.

But the use of non-orthogonal distorted meshes, specially in the near-plume region, has revelead an important error in the conservation of the mass flux along the thruster, even for the corrected time-centered values of particle properties.

The effect of non-orthogonal distorted meshes in the PIC computations have been deeply studied in ref. 13. The authors concluded that the particle push algorithm, a 'leapfrog' integration scheme, has an error proportional to the mesh distortion magnitude.

To solve that question, a new flux-computing algorithm is proposed. It is based on measuring surface properties instead of volume properties. The *Surface PIC-flux* algorithm in the computational grid is

$$\begin{aligned} \Psi_\sigma &= \frac{1}{\Delta S_\sigma \Delta t} \sum_{j \to \Delta S_\sigma} \frac{M_j}{m_i} \psi_j \\ \Psi_\chi &= \frac{1}{\Delta S_\chi \Delta t} \sum_{j \to \Delta S_\chi} \frac{M_j}{m_i} \psi_j \end{aligned} \tag{12}$$

where σ and χ are the arch-length parameters of the curves $\xi = cont$ and $\eta = const$, as was explained in the previous section, ΔS_σ and ΔS_χ are the annular faces of the cells of the mesh, Δt is the timestep, m_i is the mass of the atoms, and ψ_j is the particle property that is measured, i.e. mass ($\psi_j = m_i$) or energy ($\psi_j = m_i v_j^2/2$).

These fluxes must be converted to the physical space Ω_P. Since the mesh is non-orthogonal, some metric corrections must be taken into account. Therefore, the mass flux $\mathbf{g}$ in axial (z) and radial (r) components is (energy flux has a similar expression)

$$\begin{aligned} g_z &= \Psi_\sigma \frac{\omega_\sigma}{|J|} \frac{\partial z}{\partial \sigma} + \Psi_\chi \frac{\omega_\chi}{|J|} \frac{\partial z}{\partial \chi} \\ g_r &= \Psi_\sigma \frac{\omega_\sigma}{|J|} \frac{\partial r}{\partial \sigma} + \Psi_\chi \frac{\omega_\chi}{|J|} \frac{\partial r}{\partial \chi} \end{aligned} \tag{13}$$

where the involved metric parameters are

$$\omega_\sigma = \sqrt{\left(\frac{\partial r}{\partial \chi}\right)^2 + \left(\frac{\partial z}{\partial \chi}\right)^2}$$
$$\omega_\chi = \sqrt{\left(\frac{\partial r}{\partial \sigma}\right)^2 + \left(\frac{\partial z}{\partial \sigma}\right)^2}$$
$$|J| = \left|\frac{\partial z}{\partial \sigma}\frac{\partial r}{\partial \chi} - \frac{\partial z}{\partial \chi}\frac{\partial r}{\partial \sigma}\right| \tag{14}$$

Figure 3(a) shows the mass flows of ions, neutrals, and heavy species (ions + neutrals) along the thruster for two different simulations. The first one using the weighting formula (11) to compute the mass flux, and the other one using the surface PIC-flux algorithm (13). Both simulations started with the same initial conditions and identical input parameters. The largest discrepancies on results are observed for the ion and neutral flows, not only in the near-plume region, but also in the near-exit region of the channel discharge. With the surface PIC-flux algorithm, the ion flow is increased about 20% in the near-exit region, and about 10% in the near-plume region. By the other way, the neutral flow is decreased about 30% in the near-exit region, and about 20% in the near-plume region. The total mass flow is basically similar in the channel discharge for both simulations, but a maximum error of 5% is detected in the near-plume region using the weighting formulas. The ion energy flow for the same two simulations is shown in fig. 3(b). Although the discrepancies are important in the near-exit region, the two methods matches the ion energy flow in the boundary-cathode region.

The observed errors have important contributions in the efficiency of the thruster. Thus for the simulation using the Surface PIC-flux algorithm, the thrust efficiency is $\eta = 0.38$, whereas for the simulation using the weighting algorithm, the thrust efficiency is $\eta = 0.31$. It is important to note that the results using the Surface PIC-flux algorithm agree well with the previous reported simulations of the HPHall code, i.e. ref. 6.

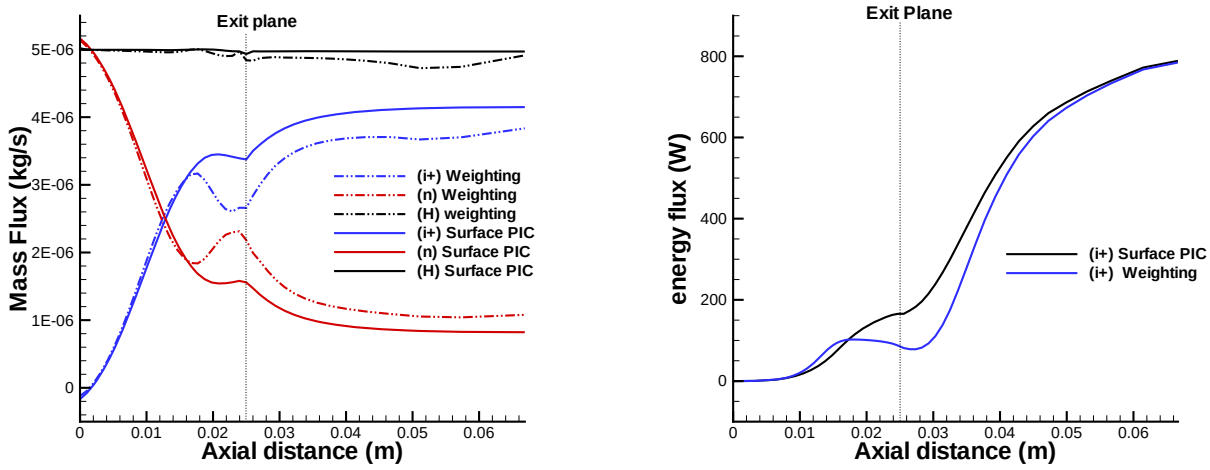
An explanation for the reduction of the ion mass flow and the ion energy flow in the near-exit region using the weighting formulas is a bad development of the acceleration region of the thruster. This is probably due to the coupling between the particle subcode and the electron fluid subcode. Since the electron subcode computes the electric potential field $\phi(\mathbf{r})$, the temperature field $T_e(\mathbf{r})$, and the electron density currents $\mathbf{j}_e(\mathbf{r})$, using the ion density, n_i, and the ion current densities, $\mathbf{j}_i$, of the PIC code, an incorrect computation of these ion variables may induce a bad simulation of the whole thruster.

V. The ion energy balance

The ions are subject to ionization and recombination processes inside the thruster. Ionization takes place by electron impact on the neutrals, whereas recombination takes place by electron impact on the ions. The HPHall code simulates single-charge and double-charge ionization events, using a simplified standard Montecarlo collision model (MCC).[9] This method is based on performing a number of probability tests to determine the real number of ion-particles to be created.

The recombination process converts ions to neutrals using a probability model too. Although recombination can be disregarded in the bulk of the thruster, is of very importance at the walls, where recombination of impacting ions is assumed. Thus in fact, the walls are a sink of ion-particles and a source of neutral-particles.

All these probabilistic processes (ionization, recombination) plus the escape of ions towards the



(a) (b)

Figure 3. (a) Mass flows of ions (blue lines), neutrals (red lines), and heavy particles (black lines) along the thruster for a mass flux rate of $5mg/s$. Solid lines are computed with the Surface PIC-flux(13) algorithm, whereas dashed-dot lines are computed whith the weighting formulas(10, 11). (b) Ion Energy flow using the Surfac PIC-flux algorithm (black line), and the weighting formulas (blue line).

injector (ion reverse current), and the vacuum (jet propulsion), makes the question of verifying the ion energy balance very important.

The ion energy balance equation for a fluid model description is

$$\frac{\partial}{\partial t}\left(\frac{3}{2}p_j + \frac{1}{2}m_j n_j u_j^2\right) + \nabla\cdot\left(\frac{3}{2}p_j\mathbf{u}_j + \frac{1}{2}m_j n_j u_j^2\mathbf{u}_j + \mathbf{P}_j\cdot\mathbf{u}_j + \mathbf{q}_j\right) = eZ_j n_j \mathbf{u}_j\cdot\mathbf{E} + \mathcal{E}_j \qquad (15)$$

where subindex j stands for the ion species (i_+ or i_{++}), $\mathbf{u}$ is the ion fluid-macroscopic velocity, n is the ion density, $\mathbf{P}$ and p are the ion pressure tensor and the ion scalar presure, respectively, $\mathbf{q}$ is the ion heat flux vector, and $\mathcal{E}$ is the ion production energy. The integration of eq. (15) over the physical space Ω_P, leads to the ion power balance equation

$$\frac{\partial U_j}{\partial t} + P_{j,\partial\Omega_P} = P_{j,elec} + P_{j,ioniz} \qquad (16)$$

where $P_{j,\partial\Omega_P}$ accounts for the ion energy flux crossing the boundaries of Ω_P, $P_{j,elec}$ is the electric power over the ions, $P_{j,ioniz}$ is the power gain (or loss) due to ionization, and $\partial U_j/\partial t$ is the variation with respect to the time of the total ion energy inside Ω_P. Since HPHall averages the results throughout the simulation time, typically of $1ms$, this variation can be disregarded.

The correct way to compute the quantities of eq. (16) is calculating one-by-one particle properties instead of using the weighted macroscopic magnitudes of the ions. Thus the total ion power crossing the boundaries in a Δt PIC-timestep is

$$P_{j,\partial\Omega_P} = \frac{1}{\Delta t}\sum_{j\to\partial\Omega_P}\frac{1}{2}M_j v_j^2, \qquad (17)$$

In a similar way, $P_{j,ioniz}$ is computed as

$$P_{j,ioniz} = \frac{1}{\Delta t}\sum_{j\ new}\frac{1}{2}M_j v_j^2 - \frac{1}{\Delta t}\sum_{j\ ioniz}\frac{1}{2}\Delta M_j v_j^2 \qquad (18)$$

where the new ionized particles represent an energy gain, and ΔM_j accounts for the mass (and energy) lost by the ion-particles ionized to produce higher charged ion-particles (i.e. i_+ to i_{++}). The ion-particle mass, M_j, and the ion-particle velocity, v_j, must be computed at the same time in order to increase the precision of the computations to $O(\Delta t^2)$. This is specially important in the calculation of the ion electric power $P_{j,elec}$, because it usually has the major contribution to the error of the ion power balance. Since the electric field $\mathbf{E}$ is known at the time t, the correct expression of $P_{j,elec}$ is

$$P_{j,elec} = \frac{1}{\Delta t} \sum_j eZ_j \frac{M_{j,t}}{m_i} \mathbf{v}_{j,t} \cdot \mathbf{E}_t \tag{19}$$

where the ion-particle velocity $\mathbf{v}_{j,t}$ is time-centered from the data obtained by the 'leapfrog' scheme

$$\mathbf{v}_{j,t} = \frac{1}{2} \left(\mathbf{v}_{j,t+\Delta t/2} + \mathbf{v}_{j,t-\Delta t/2} \right) \tag{20}$$

This correction makes the error smaller than 0.5% on the ion power balance equation, whereas previous version of HPHall has an error about 5%.

VI. Control of recombined neutrals at the walls

The recombination process at walls means, in terms of the PIC code, a conversion of the ion-particles going through the wall to new neutral-particles reinjected into the simulation domain. This process can not be performed in a single ion-to-neutral operation, since the neutral-particles are typically two order of magnitude bigger than the ion-particles. This is because two reasons. First, the neutral density inside a Hall thruster is typically $10^{19} m^{-3}$, whereas the ion density is $10^{17} m^{-3}$. And second, a similar amount of particles for both species is recommended for a good trade-off between computational cost and statistics.

The wall recombination model implemented in previous versions of HPHall,[3] takes a fixed probability P_{wall} to create new neutral-particles at the boundary panels. The process is very simple. If a generated random number is lower than P_{wall}, which typical value is 5%, a neutral-particle is created with a mass

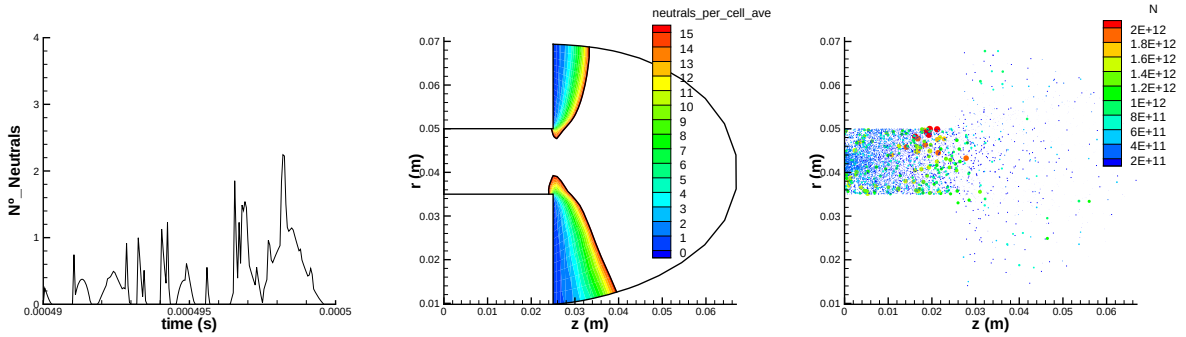
$$M_n = \frac{\Delta M_{i,out}}{P_{wall}} \tag{21}$$

where $\Delta M_{i,out}$ is the ion mass deposited at the boundary panel in the timestep Δt.

The simulation results of the HPHall code have shown that this recombination model has several objections. First, in low-density regions of the thruster, the generation of neutral-particles at the walls is very infrequent, which means that there are too few neutral-particles per cell, fig. 4(a). This effect becomes very important in the near-plume region, fig. 4(b), because the wall recombination is the only source of neutral-particles (if the neutral-ion charge-exchange collision model is turned-off). Second, the created neutral-particles may have a mass much more greater than the mean mass of the neutral population. This produces inhomogeneities in the statistics of the neutral population, as figure 4(c) shows. And third, a fixed value of P_{wall} along the walls does not take into account the plasma inhomogeneities along the thruster.

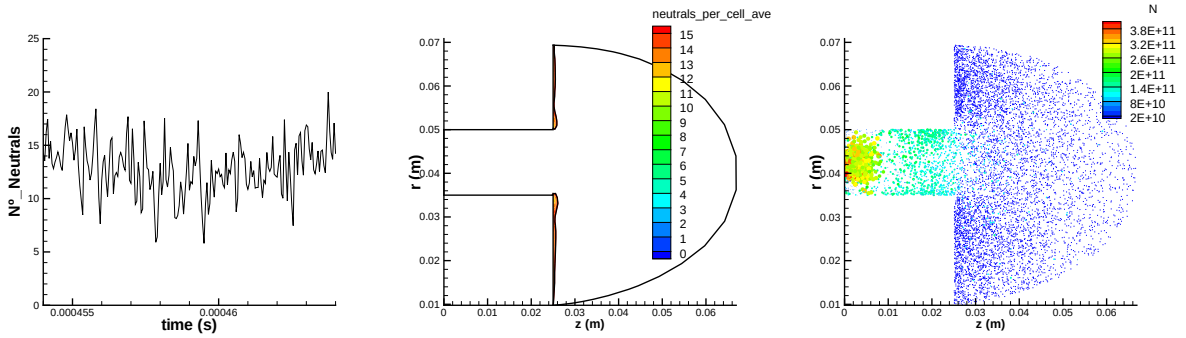
To solve those issues we propose a new recombination model based on the ionization (MCC) algorithm of Parra et al.[4] The idea of the model is to perform a number N_{prob} of probability tests to determine the number of the neutral-particles to be created, N, at a particular boundary panel and timestep. This MCC method establishes a relation between the mass of the new neutral-particle, M_n, and the available mass $\Delta M_{i,out}$ for recombination

$$M_n N_{prob} P_{wall} = \Delta M_{i,out} \tag{22}$$



(a) (b) (c)

Figure 4. Old recombination model at walls. (a) The number of neutral-particles per cell along time at a test control node in the outer wall of the near-plume region. (b) The regions where the average number of neutral-particles per cell is lower than 15. (c) Snapshot of the neutral-particle population. The particles are plotted according to their size measured as the equivalent amount of atoms that they represent.



(a) (b) (c)

Figure 5. New recombination model at walls. The parameters of the model are: $\alpha = 3$, $\beta = 1$, $N_{min} = 15$, $N_{max} = 45$. (a) The number of neutral-particles per cell along time is ever greater than 5, and the average is about 15. (b) The regions where the average number of neutral-particles per cell is lower than 15 are limited to a thin strip adjacent to the exterior walls . (c) Snapshot of the neutral-particle population, where it is shown that the are much more particles in the near-plume region, and the size of the particles is more homogeneous than before.

Therefore, only one of the three magnitudes M_n, N_{prob}, and P_{wall}, is determined, and the other two remain as free parameters. In order to control better the number of neutral-particles per cell, lower and upper limits of neutral-particles per cell are defined, N_{min} and N_{max}. Thus, depending on the number of existing neutral-particles N_n in the boundary-cell, three situations are possible:

1. If $N_n < N_{min}$: $P_{wall} = 1.0$, $N_{prob} = N_{min} - N_n + 1$, and M_n is obtained from eq. (22).

2. If $N_{min} < N_n < N_{max}$: $N_{prob} = 1$, the wall recombination probability follows the law

$$P_{wall} = (N_n - N_{min} + 1)^{-\beta}, \tag{23}$$

where β is a free parameter of the model to control the creation of neutral-particles, and M_n is obtained from eq. (22) with the constraint

$$M_n \leq \alpha M_{n,cell}, \tag{24}$$

where $M_{n,cell}$ is the averaged mass of the existing neutral-particles in the boundary cell, and α is a parameter to control the mass-dispersion of the neutral population. If the mass of the new neutral-particle is greater than $\alpha M_{n,cell}$, then N_{prob} is increased and the whole process repeated until the eq. (24) is fulfilled.

3. If $N_n > N_{max}$: $N_{prob} = 1$, the wall recombination probability follows the law

$$P_{wall} = (N_{max} - N_{min} + 1)^{-\beta}\,, \tag{25}$$

and the computation of M_n is the same as in case 2.

The parameters α and β of the model are tuned for convenience. The values $\alpha = 3$ and $\beta = 1$ have demonstrated a good control of the neutral-particle population. Typical limits of neutral-particles per cell are $N_{min} = 15$ and $N_{max} = 45$, which are similar to the limits of the ion population control algorithm.[4] Whith these values, the wall recombination probability of case 3 is $P_{wall} = 3.2\%$, a value similar to the old recombination model, whereas in case 2, P_{wall} varies from 100% to 3.2% depending on N_n. Simulation results using the new recombination model are shown in fig. 5. The number of neutral-particles per cell is significantly increased in the near-plume region, reducing also the numerical oscillations. The size of the particles, in terms of the number of atoms that they represent, is adjusted to nice values in order to obtain better statistics in the simulations.

VII. Conclusions

The non-orthogonal structured meshes of the hybrid codes are well suited to simulate complex geometries of the Hall thrusters. The use of refined meshes in the near-plume region has reveled errors on the particle algorithms implemented in our hybrid code, an updated version of HPHall. Several improvements have been reported to solve that question. First, the computation of the electric field taking into account the non-uniformities of the mesh. Second, the definition of a new algorithm to compute accurately mass and energy fluxes of the heavy species, showing better performance results of the thruster. And third, a new recombination model at walls have been presented to better control the neutral-particle population.

Acknowledgments

This research was financed by the Ministerio de Educación y Ciencia of Spain, under project ESP-2007-62694.

References

[1]J. Fife and M. Martínez-Sánchez, Comparison of results from a two-dimensional numerical SPT model with experiment, in *32nd Joint Propulsion Conference, Lak Buena Vista, FL*, AIAA-1996-3197, American Institute of Aeronautics and Astronautics, Washington, DC, 1996.

[2]J. M. Fife, *Hybrid-PIC Modeling and Electrostatic Probe Survey of Hall Thrusters*, PhD thesis, Massachusetts Institue of Technology, 1998.

[3]F. Parra, E. Ahedo, M. Fife, and M. Martínez-Sánchez. Journal of Applied Physics **100**, 023304 (2006).

[4]F. Parra, D. Escobar, and E. Ahedo, Improvements on particle accuracy in a Hall thruster hybrid code, in *42th Joint Propulsion Conference, Sacramento, CA*, AIAA-2006-4830, American Institute of Aeronautics and Astronautics, Washington, DC, 2006.

[5]I. Maqueda, D. Escobar, and E. Ahedo, Advances on a Hall thruster hybrid code, in *30th International Electric Propulsion Conference, Florence, Italy*, IEPC 2007-066, Electric Rocket Propulsion Society, Fairview Park, OH, 2007.

[6]D. Escobar and E. Ahedo, IEEE Transactions on Plasma Science **36**, 2043 (2008).

[7]D. Escobar, A. Antón, and E. Ahedo, Simulation of high-specific-impulse and double-stage Hall thrusters, in *Proc. 29th International Electric Propulsion Conference, Princeton, USA*, IEPC-2005-040, Electric Rocket Propulsion Society, Fairview Park, OH, 2005.

[8]S. Barral and E. Ahedo, Physical Review E **79**, 046401(1) (2009)

[9]F. Parra, Actualización y mejora de un código pic-fluido bidimensional para el flujo de plasma en motores de efecto Hall, Master's thesis, Escuela Técnica Superior de Ingenieros Aeronáuticos, Universidad Politécnica de Madrid, 2004.

[10]E. Ahedo, Physics of Plasmas **9**, 4340 (2002).

[11]E. Ahedo and D. Escobar, Physics of Plasmas **15**, 033504 (2008).

[12]C. K. Birdsall and A. B. Langdon. *Plasma Physics via Computer Simulation*. IOP Publishing Ltd. (1991).

[13]J. Wang, D. Kondrashov, P. C. Liewer, and S. R. Karmesin, Three-dimensional deformable-grid electromagnetic particle-in-cell for parallel computers, J. Plasma Physics, **61**, pp. 367-389, (1998).

[14]G. B. Jacobs, J. S. Hesthaven, High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids, J. of Comp. Phys., **214**, (2006).